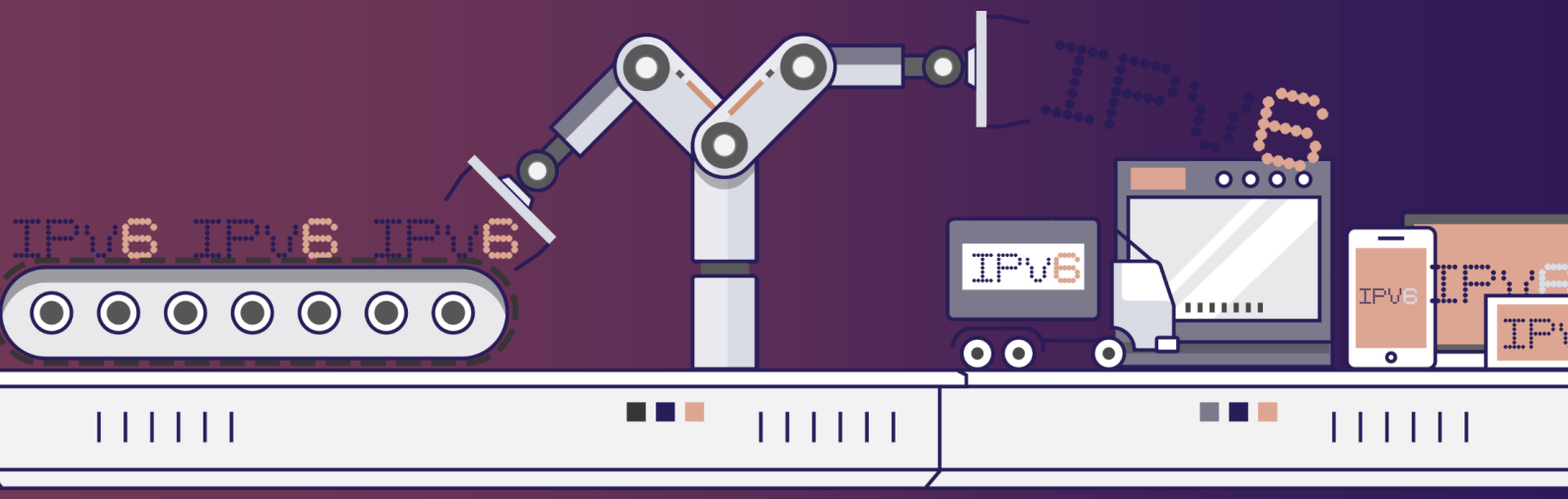
RÉPUBLIQUE
FRANÇAISE

ENTREPRISES : COMMENT DÉPLOYER IPv6 ?



DE LA RÉFLEXION AU DÉPLOIEMENT

6 juillet 2022 // **V1.3**

Ce document VOUS AIDERA Á DÉFINIR VOS BESOINS IPv6, Á PLANIFIER son implémentation et Á LE DÉPLOYER au sein de votre organisation.

Il porte sur tout type de SI.

Avant propos

Remerciements

Ce guide a été rédigé par Jean-Charles BISECCO, avec l'aide des participants de la task-force IPv6.

Ce document s'inscrit dans le cadre des travaux de la task-force IPv6 co-pilotée par l'Arcep et Internet Society France. L'Arcep et Internet Society France ont mis en place une task-force dédiée à IPv6 et ouverte à l'ensemble des acteurs de l'écosystème internet (opérateurs, hébergeurs, entreprises, secteur public, etc.). Elle a pour objectif de favoriser l'accélération de la transition vers le protocole IPv6 en permettant aux participants d'aborder des problèmes spécifiques et de partager les bonnes pratiques.

- Ce guide réalisé par la task-force IPv6 a pour objectif de donner de bonnes pratiques sur la mise en œuvre de la transition vers IPv6. Il n'a pas vocation à se substituer à l'expertise des équipes informatiques des entreprises et la task-force IPv6 ne saurait voir sa responsabilité engagée en cas de difficultés rencontrées par une entreprise qui suivrait les recommandations du guide.

Inscrivez-vous à la task-force IPv6 :

Ce document ne représente pas une prise de position de l'Arcep, mais reflète les travaux des participants à la task-force.

Retrouvez la dernière version du guide ici :

https://www.arcep.fr/la-regulation/grands-dossiers-internet-et-numerique/lipv6/task-force-ipv6.htm

Sommaire

AVANT PROPOS **2**

SOMMAIRE **3**

PREAMBULE **7**

LECTURE **8**

I. **PREHENSION** DU SUJET **10**

1. **Méthode d'analyse** **12**
   - PROJETS SIMILAIRES 12
   - DÉMÊLER LA PELOTE 12
2. **Facteurs humains** **13**
   - FORMATION 14
   - ENTRAIDE 14
3. **Besoins** **15**
   - EXPOSITION SUR INTERNET 15
   - ACCÈS AUX RESSOURCES EXTÉRIEURES 18
   - RÉSEAU INTERNE 20

II. **TECHNOLOGIES** DE MIGRATION **24**

1. **Dual-Stack** **26**
2. **Mécanismes de transport** **27**
   - INCLUSION SUR UN UNDERLAY IPV4 EXISTANT 27
   - ENCAPSULATION SPÉCIFIQUE 29
   - ET DANS L'AUTRE SENS ? 29
3. **Mécanismes de traduction** **30**
   - NAT64 + DNS64 30
4. **Quelles technologies pour chaque périmètre ?** **33**
   - CAMPUS 33
   - DATACENTER 34

◗ WAN 37

III. **ORDONNANCEMENT** DES BRIQUES **38**

1. **Avant de démarrer** **40**
2. **Réseau** **41**
   - ◗ MATURITÉ 41
   - ◗ HARDWARE 42
   - ◗ MAQUETTAGE 42
   - ◗ ROUTAGE INTERNE 45
   - ◗ FILTRAGE ET TRAÇABILITÉ 46
3. **Services d'infrastructure** **46**
   - ◗ SIEM 46
   - ◗ DNS/IPAM/DHCP 47
   - ◗ VPN, PROXY ET REVERSE PROXY 47
   - ◗ SOUCHES D'OS 47
   - ◗ SERVICES BUREAUTIQUES 50
   - ◗ APPLICATIFS 51

IV. **PLAN** D'ADRESSAGE **54**

- ◗ PUBLIC OU PRIVÉ ? 56
- ◗ PETITE STRUCTURE 56
- ◗ GRANDE STRUCTURE 58
- ◗ GROUPEMENT LOGIQUE 60
- ◗ ÉLÉMENTS CONSTITUANTS 60
- ◗ TAILLE DES PRÉFIXES 62
- ◗ ADRESSES DES SERVICES COURANTS 63
- ◗ ÉVOLUTIVITÉ TEMPORELLE 64
- ◗ USAGE DU N° D'HÔTE 0 65
- ◗ SEGMENTATION PAR INTERFACE 65
- ◗ CORRESPONDANCE V4 / V6 66
- ◗ POUR LES RÉSEAUX NATIFS V6 69
- ◗ ANNONCES PUBLIQUES 69

V. **SECURITE** ET BONNES PRATIQUES **72**

1. **Couche Accès** **75**
   - ◗ AFFECTATION DYNAMIQUE D'ADRESSES 75
   - ◗ BLOCAGE D'ICMP REDIRECT 77
   - ◗ IPV6 SNOOPING 78

- ROGUE DHCP 81
- RA GUARD 82
- RA HOP LIMIT 83
- AUTRES CONFIGURATIONS DU RA 84
- seND (NON-EXPLOITABLE ACTUELLEMENT) 85
- MLD 86
- STORM CONTROL 87
- ADRESSES MULTICAST À BLOQUER 87

2. Hôte **88**
- DHCP 88
- MÉTHODE DE GÉNÉRATION D'ADRESSE SLAAC 89
- NE PAS DÉSACTIVER LA STACK IPV6 92
- DÉSACTIVATION DES MÉCANISMES DE TRANSITION 92
- DÉSACTIVATION DES PROTOCOLES D'AUTO-DÉCOUVERTE 92
- BLOCAGE DU TRAFIC LINK-LOCAL 93
- VPN 94
- CONFIGURATION D'OS DESKTOP 94
- MOBILE ET EMBARQUÉ 96

3. Transit **98**
- URPF 98
- PROTECTION DU CONTROL PLANE 98
- SÉCURISATION OSPF 98

4. Filtrage **99**
- ICMP 99
- MÉCANISMES DE TRANSITION 101
- BOGON PREFIXES ET ROUTES 102
- EXTENSION D'EN-TÊTE 104
- POLITIQUE DE BANNISSEMENT 105

VI. ANNEXES ET AUTRES ELEMENTS 106
- URL ET IP LINK-LOCAL 108
- MULTI-PREFIXES 110
- CONTAINERS 110
- SCADA 112
- NAT64 CHEZ LES OPÉRATEURS MOBILES 112
- PARTAGE DE PORTS SUR IPV4 113
- DRAFTS RFC POUR SAUVER IPV4 114
- EXEMPLES DE PROBLÈMES D'IMPLÉMENTATION IPV6 114
- GASPILLAGE D'ADRESSES 116

◗ USAGE DE L'UNICITÉ DES ADRESSES POUR AUTRE CHOSE 116
◗ SRv6 117
◗ THREAD 117
◗ SELF-HOSTING AND RESIDENTIAL USE 118
◗ OUVERTURE AUTOMATIQUE À L'INITIATIVE DE L'HÔTE 121
◗ ÉVOLUTION DES JEUX EN LIGNE 122
◗QUE DEMANDER AUX OPÉRATEURS GRAND PUBLIC ? 122

VII. **À PROPOS DE CE DOCUMENT 123**
◗ AIDEZ-NOUS À FAIRE ÉVOLUER CE DOCUMENT 123
◗ LICENCE 123
◗ TRADUCTIONS 123

Préambule

Le déploiement d'IPv6 progresse partout dans le monde et son usage n'est plus anecdotique comme ça pouvait être le cas au début de la précédente décennie. Ce guide vous aidera à définir les périmètres sur lesquels implémenter le protocole ainsi que la marche à suivre et les bonnes pratiques.

Si une documentation abondante existe sur IPv6 et sur son déploiement, celle-ci reste la plupart du temps centrée sur la couche réseau et s'adresse typiquement à un opérateur, un transitaire ou un point d'échange.

Cependant, la fourniture des services numériques dans une entreprise s'apparente généralement à un modèle plus verticalisé avec des configurations parfois uniques, dès lors qu'on se rapproche des innombrables applicatifs en production.

Ce guide vise à apporter aux services informatiques impliqués dans la transition IPv6 en entreprise des informations destinées à faciliter opérationnellement, cette transition.

Si les aspects techniques sont abordés dans ce document, ce dernier n'a pas vocation à enseigner IPv6 et se contentera de rappels sur les points abordés. Vous pourrez approfondir ces éléments via les contenus existants, pédagogiques, documentations constructeurs, billets de blogs, MOOC, formations, etc.

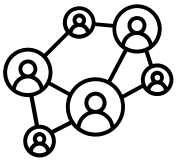
Lecture

Ce document s'adresse prioritairement aux experts des systèmes d'information en charge de la transition vers IPv6.

Il contient néanmoins des sections et des paragraphes pouvant intéresser des publics variés.

S'il est recommandé aux personnes les plus proches du projet de tout parcourir, il en va autrement pour le reste du lectorat. Afin de faciliter l'identification, des logos sont apposés en marge des sections afin d'attirer l'attention de certains lectorats.

Réseau

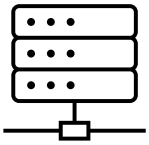


Organisationnel projet

Système

Téléphonie

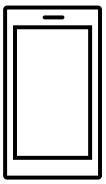
Sécurité



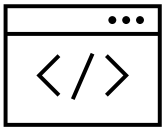
Hébergement

Traçabilité

Usage Domestique



Mobile



Applicatif Dev

Les paragraphes proposent parfois des choix différents pour les grandes structures que pour les petites. S'il est impossible de fixer une limite entre les deux, il est évident que plus une structure comporte un SI complexe et étendu plus elle tend vers ce que l'on considère être dans ce document une « grande structure », même si elle ne se classe pas comme « grande » du fait de sa masse salariale ou sa répartition géographique.

1

Préhension du sujet

1. **Méthode d'analyse** **12**
   - PROJETS SIMILAIRES 12
   - DÉMÊLER LA PELOTE 12
2. **Facteurs humains** **13**
   - FORMATION 14
   - ENTRAIDE 14
3. **Besoins** **15**
   - EXPOSITION SUR INTERNET 15
   - ACCÈS AUX RESSOURCES EXTÉRIEURES 18
     - L'arrivée de QUIC 19
   - RÉSEAU INTERNE 20

Préhension du sujet

I. Préhension du sujet

1. Méthode d'analyse

Si les aspects techniques sont une part importante du sujet, réduire le déploiement d'IPv6 à celle-ci serait une grave erreur, néanmoins la largesse du sujet rend souvent difficile le choix des méthodologies de gestion de projet et d'ingénierie, qui, de plus, peuvent varier selon les périmètres au sein d'une même organisation.

Le présent chapitre vise à éclairer le lecteur sur un certain nombre de choix méthodologiques pouvant avoir un impact direct sur la réussite du projet.

PROJETS SIMILAIRES

Le sujet est vaste, mais il n'est pas le premier, loin de là. Selon l'âge de votre organisation, d'autres projets souvent guidés par la contrainte ont pu avoir une portée similaire. Le retour d'expérience sur la planification et le déroulement de ceux-ci peuvent vous éviter de réitérer d'éventuelles erreurs.

L'exemple le plus ancien est le passage à l'an 2000, qui a parfois demandé des changements profonds notamment sur les SGBD (Bases de données) ainsi que des tests poussés pour un élément qui était codé sur 2 chiffres.

Plus récemment, l'arrivée des certificats TLS y compris sur les applications à usage interne et non exposées à internet a induit des changements sur des serveurs et des *proxys / load balancers*, etc. De la configuration de *middleware*, la création ou l'extension de PKI, l'ajout de déchiffrement sur les systèmes d'audit et de surveillance du trafic, jusqu'au contrôle du bon déploiement de l'ensemble de la chaine de certification sur les différents nœuds, de nombreux éléments étant affectés.

TLS est d'ailleurs en constante évolution et demande des changements réguliers liés à l'obsolescence d'algorithmes, à l'ajout de nouveaux mécanismes, etc.

Un dernier exemple plus ou moins fréquent selon la stratégie de la structure est le déploiement d'une nouvelle version d'un système d'exploitation qui comporte une phase de migration des postes clients avec son lot de recettes applicatives et d'accompagnement au changement.

D'autres projets larges sont répandus comme le déploiement d'un ERP, mais ils sont alors plutôt orientés métier et sont donc poussés par le cœur d'activité de l'organisation.

DÉMÊLER LA PELOTE

Le protocole IP est l'élément le plus bas de l'infrastructure qui a besoin d'une compatibilité de bout en bout entre les différents constituants du système d'information. Aucune méthode d'analyse ou de raisonnement ne permet de couvrir efficacement les impacts de l'évolution, il faut en conjuguer plusieurs.

Le cartésianisme aura sa place pour l'étude des éléments de façon unitaire : par exemple une fonctionnalité de sécurité relative à IPv6 est-elle correctement implémentée par le constructeur de mon pare-feu ? Mon *middleware* se comporte-t-il de la même façon sur son socket IPv4 et IPv6 ?

À l'opposé, la méthode d'analyse systémique est adaptée à la vision d'ensemble qu'est cette toile d'équipements interconnectés avec chacun leur système et couche applicative. La vision globale des briques du SI sur différentes couches doit être résumée via cette méthode.

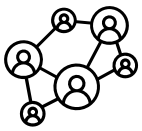
Enfin, et parce qu'il n'est pas possible de plonger en détail dans chaque élément du système d'information, le réductionnisme doit parfois être employé notamment pour aborder les éléments périphériques de l'écosystème comme la supervision ou la collecte de *logs* pour mieux y revenir lors de la phase pilote et ne pas faire 2 fois les mêmes efforts.

Essayer de se pencher sur tout en même temps, de tout inventorier ne sera sans doute pas la bonne méthode pour mener le projet à bien.

Tel un bon auteur de thriller, vous devrez amener les détails sur vos personnages au bon moment, ce type de projet est long et il ne sert à rien de plonger trop en profondeur dans tout l'écosystème alors qu'il aura probablement évolué d'ici à ce qu'il soit atteint par les pilotes et le déploiement.

En parlant d'évolution des périmètres impactés, chacune est une opportunité de déployer IPv6, faites régulièrement le tour des services pour connaitre les nouveaux projets afin de ne pas rater ces opportunités.

2. Facteurs humains



Vous vous heurterez potentiellement à des acteurs réfractaires au changement, aussi vous faudra-t-il trouver des soutiens dans chaque équipe impliquée afin de pouvoir avancer. L'implication régulière de membres de différentes équipes doit commencer au bon moment. Trop tôt les personnes perdront l'enthousiasme et le soufflé retombera, trop tard, ça impactera le projet en termes de délai. Idéalement, le phasage permettra d'élargir le nombre de participants d'une équipe dès lors qu'elle devra se pencher concrètement sur le sujet.

Les larges réunions plurientités ne doivent servir qu'à informer, et pas à brainstormer ou débattre. D'autres réunions préalables avec 1 à 2 représentants par équipe suffisent à cela. Les points de détail relatifs à une équipe doivent être abordés avec cette équipe et éventuellement les voisines d'écosystèmes concernés.

Tous ces éléments peuvent paraitre classiques, voire anodins, mais ce type de projet embarquant un très grand nombre d'acteurs, une hiérarchisation des échanges est nécessaire dans une structure de grande taille.

Pour résumer, identifiez 1 à 2 correspondants par équipe, et faites-vous informer des projets futurs qui pourraient représenter des opportunités de déploiement.

Présentez régulièrement une vue technique à haut niveau du projet aux équipes de façon conjointe, et une vue moins technique à un public plus large cette fois-ci tantôt via des réunions d'information, tantôt via des supports de communications envoyés.

Enfin, échangez avec les équipes impliquées de façon unitaire lorsqu'elles sont dans la phase d'approfondissement, cette fois de manière plus large et en suivant le phasage.

Inutile par exemple de solliciter régulièrement toute l'équipe responsable des *middlewares* pour leur demander de se préparer alors que le réseau n'a pas encore prévu de pilotes permettant de supporter des serveurs de qualification.

FORMATION

La formation des collaborateurs doit être anticipée afin de les préparer à l'arrivée d'IPv6.

La mise en place de cursus ou parcours de formation en fonctions des rôles, métiers, expertises doit être prise en compte pour accompagner cette transition.

Il convient d'interroger les collaborateurs sur leurs besoins de formation pour réussir à accomplir leurs métiers avec IPv6, pour ensuite identifier des parcours de formation. Et parfois même construire des modules adaptés à des spécificités de l'entreprise.

On peut à minima identifier une segmentation du public à former en différents groupes :

- Réseau ;
- Sécurité ;
- Développement applicatif.

Idéalement, l'équipe projet devrait s'essayer à l'ensemble des modules avec 1 ou 2 représentants de chaque public cible.

ENTRAIDE

Ne pas hésiter à contacter des entreprises, organisations de taille similaire, étudiant ou effectuant une telle transition afin de partager les bonnes pratiques.

La task-force IPv6 en France co-pilotée par l'Arcep et Internet Society France est l'occasion d'échanger sur le sujet avec des pairs, il est souhaitable d'en profiter.

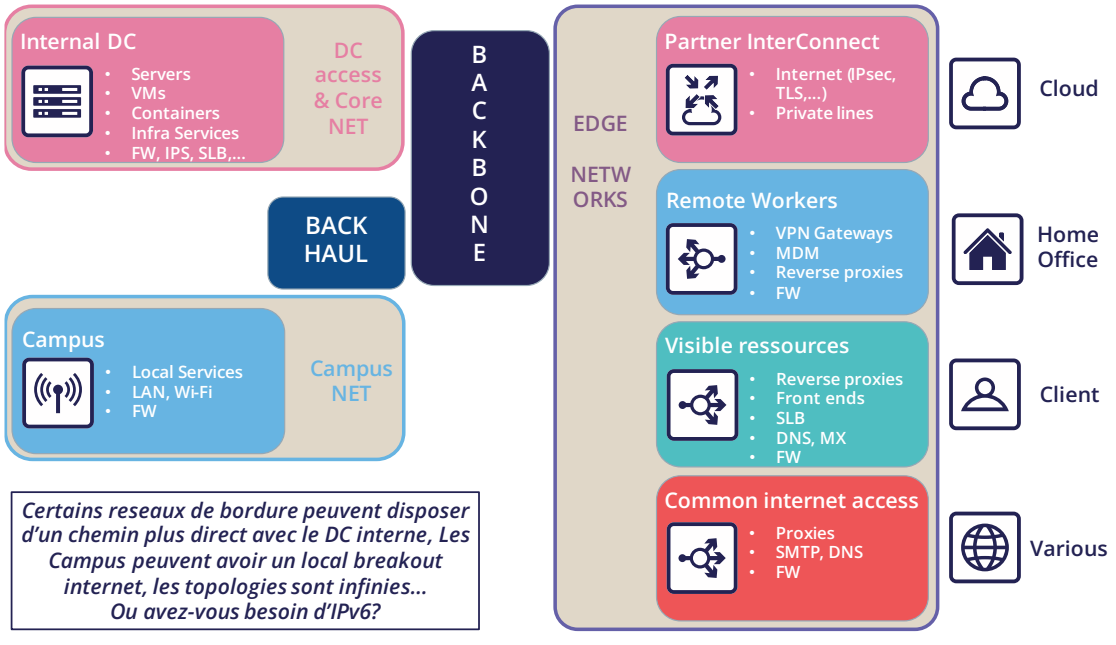
Nous comptons sur vos retours pour enrichir le guide et documenter des exemples de produits répandus en entreprise. RDV en fin de document à la rubrique retours d'expérience.

3. Besoins

Nous avons cité plus haut des exemples de projet comme TLS ou les migrations de système d'exploitation. Ces projets sont en général poussés pour résoudre une problématique de sécurité ou conserver un support technique. D'autres projets sont la résultante de l'application des lois comme la nécessaire traçabilité des actions des utilisateurs ou encore la transposition du RGPD sur ses systèmes. Évidemment la rentabilité est la source de nombreux autres projets, soit à l'initiative du métier, soit pour améliorer le niveau de service comme les projets d'orchestration.

Difficile de placer un projet IPv6 dans l'une de ces cases, et encore plus de qualifier IPv4 de « dette technique » potentielle à court terme comme peut l'être un langage de programmation désuet pour lequel on ne trouve plus de développeurs.

Cette section va donc aborder les cas d'usage d'IPv6 et exposer leurs degrés de pertinence selon la situation.



EXPOSITION SUR INTERNET

Rendre accessible en IPv6 la partie publique de son réseau est probablement l'élément prioritaire à mettre en œuvre.

Cela inclut les serveurs web, mais aussi le DNS, les passerelles VPN, etc.

Les opérateurs activent progressivement IPv6 sur leurs différents réseaux à destination du grand public. Les connexions domestiques ont ouvert le bal, puis le mobile et enfin le partage de connexion des téléphones. On trouve d'ailleurs maintenant une majorité de terminaux qui ne fonctionnent plus qu'en IPv6 sur certains réseaux mobiles, NAT64 assurant la rétrocompatibilité.

Autant de clients et de salariés qui se connectent donc aux infrastructures de l'entreprise à partir de connexions IPv6. Si la progression du déploiement opérateur atteint leurs pronostics, il est probable que plus de la moitié de la population dispose d'une connectivité native IPv6 depuis son domicile et son mobile d'ici à fin 2023. Dans certains pays (dont la France), IPv6 dépasse déjà 50%.

Pour autant, si la fourniture d'IPv4 n'est pas près de disparaitre, cette ressource devenue rare se retrouve de plus en plus souvent partagée entre de multiples abonnés via différents mécanismes. Si une connexion internet ne fournit par définition aucune garantie de service, l'incursion d'éléments intermédiaires sur la chaine IPv4 affecte statistiquement sa qualité de service. De plus, la méconnaissance de ces mécanismes par les équipes techniques peut rendre délicate la résolution de problème de connectivité, de qualité de service là où une connexion IPv6 s'établit de bout en bout sans aucune forme de supercherie protocolaire telle que le NAT44+PAT.

Notons également que les opérateurs commencent à basculer vers un monde où v6 devient le standard sur leur backbone grand public, et où v4 devient un service véhiculé de plus en plus souvent de façon encapsulée.

Si votre entreprise fournit des services dans des pays moins pourvus en stock d'IPv4 ou tout simplement progressistes par la loi, cette exposition en IPv6 peut devenir une nécessité d'alignement aux marchés en expansion ou possiblement face à une future contrainte légale. Certains pays poussent les opérateurs à fournir de l'IPv6 aux clients comme l'Inde et la France, d'autres se concentrent sur leurs propres administration comme les USA ou la Belgique, la Chine pousse une transition pour 2025, etc. Depuis le 1er janvier 2021, les opérateurs français s'étant portés acquéreurs de fréquences pour la 5G doivent fournir une connectivité IPv6, à minima en option.

Au-delà de ces aspects, un point global important est que le transit IPv6 est aujourd'hui viable et que sa maturation le rapproche qualitativement d'IPv4.

Les précurseurs ont pendant longtemps constaté qu'IPv6 était un handicap. Il y a une décennie les chemins de transit étaient moins nombreux, donc moins redondés et moins optimaux. Combien de tutoriels recommandaient, à juste titre à l'époque, de couper IPv6 afin de résoudre l'accès à tels site ou service de streaming public ? D'autant que la RFC *Happy Eyeballs* n'existait même pas et que les navigateurs ne pouvaient pas sauver la mise en quelques millisecondes comme c'est le cas aujourd'hui.

Ce « handicap » qu'était la connectivité v6 n'en est plus un aujourd'hui, c'est même tout l'inverse grâce au véritable « bout en bout ».

D'ailleurs, la latence mesurée par Google en France, au Canada ainsi que quelques autres pays sont meilleurs en IPv6, là où c'était souvent l'inverse encore en 2018. Alors que le *peering* IPv6 devient aussi bon qu'IPv4, IPv4 traverse de plus en plus souvent des CG-NAT, presque systématiquement sur mobile d'ailleurs. Un autre facteur plus minime de la diminution de latence est la suppression du checksum effectué à chaque routeur traversé en IPv6, ainsi que la non-fragmentation sur les routeurs.

À RETENIR

Il est donc primordial de se souvenir qu'IPv4 est de plus en plus souvent véhiculé de façon non native via des mécanismes d'encapsulation ou des CG-NAT, notamment sur mobile. Ce qui ajoute des points de centralisation pouvant affecter l'expérience utilisateur. Fournir un service en IPv6, c'est éviter de dépendre de ces infrastructures de traduction d'opérateur.

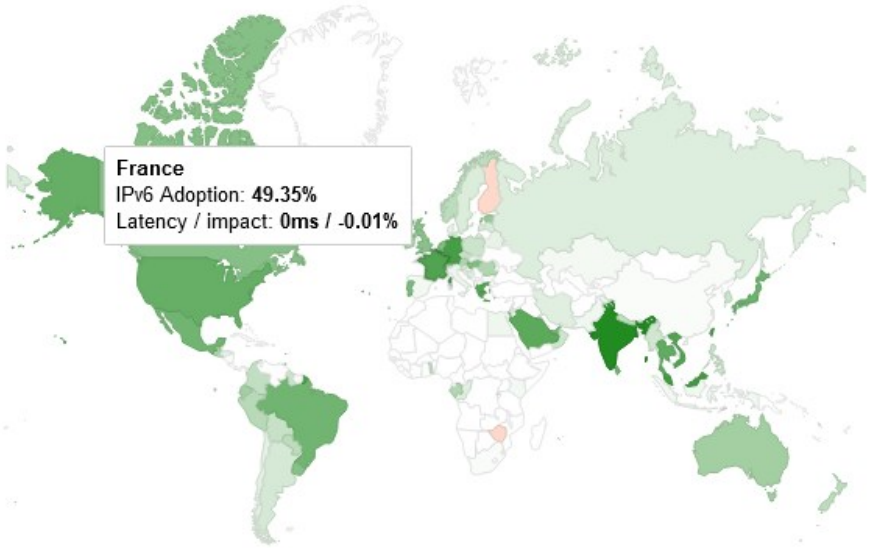


Figure 01

Statistiques France d'accès aux services de Google en IPv6 | octobre 2021

La fourniture d'IPv6 aux utilisateurs grimpe continuellement, en France, la moitié des accès aux services de Google s'opère en IPv6.

IPv4 Waiting List

LIRs in queue	579
Days that first LIR in queue has been waiting	52

We use a waiting list to allocate recovered IPv4 addresses to our members. The table above shows the number of requests already on the waiting list and the number of days that the LIR at the front of the queue has been waiting. This is also shown on the graph below, which should fluctuate over time - falling when recovered addresses become available and are allocated, and rising as new IPv4 requests are added to the waiting list. Both the table and graph are updated every three hours.

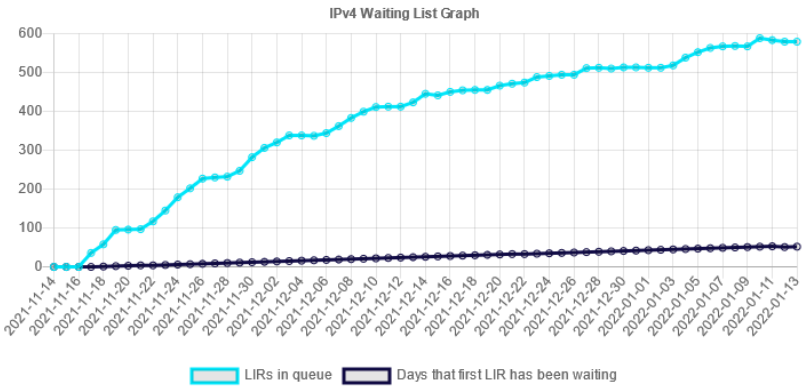


Figure 02

Liste d'attente IPv4 RIPE-NCC | janvier 2022

Obtenir des subnets IPv4 devient de plus en plus compliqué.

ACCÈS AUX RESSOURCES EXTÉRIEURES

Votre entreprise, quelle que soit sa taille, est cliente au sens protocolaire du terme, de ressources tierces. Là aussi, le nombre de sites et de services joignables en IPv6 grimpe continuellement. Le trafic généré par les utilisateurs est généralement proxisé pour des raisons de filtrage, de protection et de traçabilité. Ce proxy est la plupart du temps en IPv4 tant du côté interne qu'externe. Et ça se remarque, voyez plutôt :

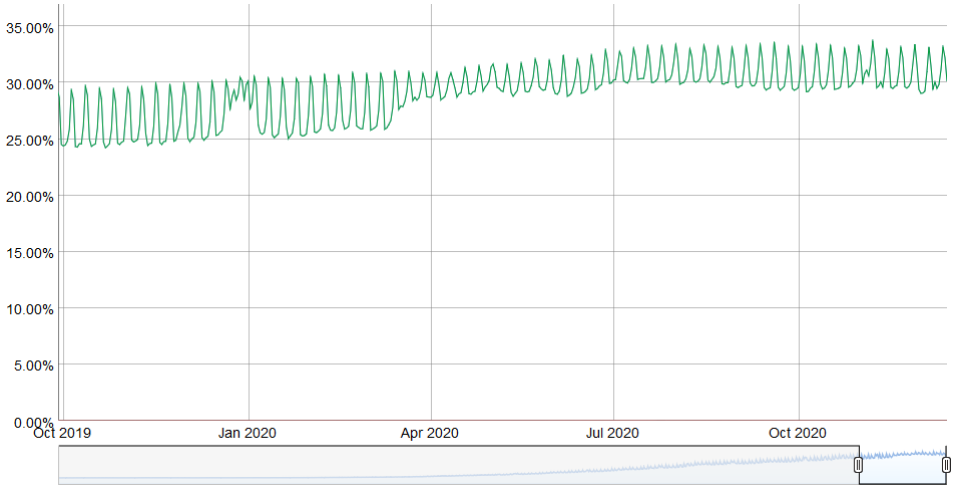


Figure 03 ***Pourcentage mondial d'accès aux services de Google en IPv6***

La période de confinement globale de Mars-Avril 2020 montre un phénomène similaire à celui de la semaine de Noël / Nouvel An. La variation de pourcentage reste dans la tranche haute, car les gens se connectent depuis chez eux où ils ont plus souvent accès à IPv6.

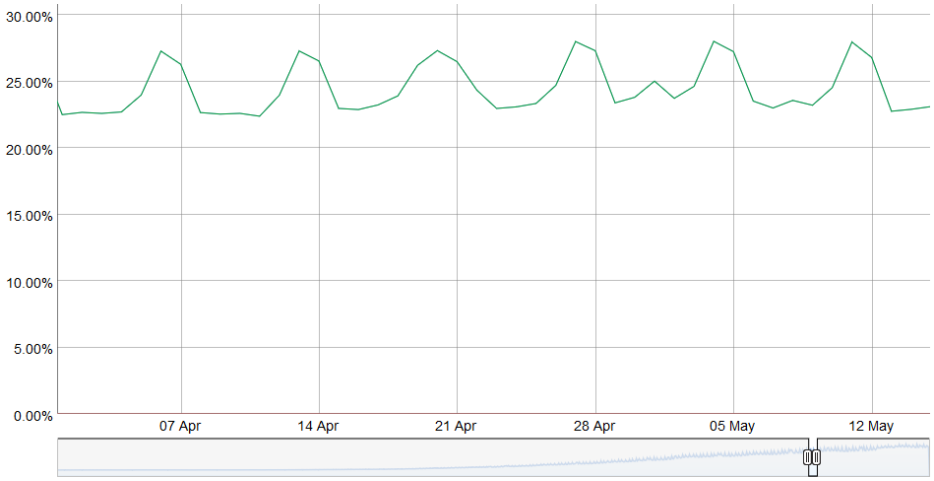


Figure 04 ***Pourcentage mondial d'accès aux services de Google en IPv6***

Ce phénomène se retrouve à l'échelle de la semaine, ici fin avril début mai 2019. Les pics se trouvent toujours le weekend. De plus le 1er et le 8 mai, fériés dans de nombreux pays, génèrent des remontées au milieu des 2 dernières semaines.

Durant l'été 2024, la moyenne d'accès IPv6 à l'échelle du globe devrait dépasser 50%. Serez-vous dans la majorité à cette date ?

L'arrivée de QUIC

Profitons-en pour aborder un point crucial dans l'accès aux ressources sur internet et parlons du règne du mode connecté. Par mode connecté on ne parle pas de l'accoutumance à l'hyperconnectivité, mais tout simplement de TCP.

TCP domine depuis longtemps grâce à ses mécanismes de contrôle, et l'UDP se cantonne généralement au temps réel où la retransmission est inutile comme la voix ou le jeu en ligne. Cependant, la connectivité est toujours plus fiable et les contrôles d'intégrités se font dans les couches hautes pour un nombre croissant d'échanges.

Ainsi, si on remonte dans les couches du modèle OSI on trouve probablement le plus gros client de TCP, HTTP. Si HTTP/1.1 est resté gravé dans le marbre depuis 1997, 20 ans après HTTP/2 a apporté la priorisation, la parallélisation, la compression et la mise en cache prédictive. HTTP/3 (RFC 9114) amène lui un schisme en se séparant de TCP pour se baser sur un nouveau protocole de transport, QUIC.

Bien qu'encapsulé dans UDP pour faciliter son déploiement, QUIC est un protocole de transport à part entière qui cherche à unifier le meilleur des 2 mondes en offrant des mécanismes réduisant considérablement le nombre d'échanges client-serveur, en plus de former une symbiose avec TLS qui lui est maintenant directement intégré. Il vise donc à offrir des connexions sécurisées, parralélisables, le tout en réduisant le nombre d'aller-retour.

Certains fournisseurs poussent déjà UDP en entreprise, notamment pour des solutions de communication. Ces fournisseurs vont parfois jusqu'à demander aux clients d'annoncer leurs IPv4 publiques du service de conférence sur leur backbone interne pour ne pas avoir à altérer le contenu du message SIP dans les couches supérieures, offrir le support d'UDP et se passer de tout traitement intermédiaire. Combien ont d'ailleurs remarqué pendant le confinement que ces solutions fonctionnaient mieux à la maison sur son poste ou sur son poste professionnel lorsqu'il offrait du split-tunneling VPN ?

Et si demain ces fournisseurs de services en Cloud poussaient QUIC et donc UDP pour le reste ? que faire ?

Et HTTP/3 n'est pas le seul à migrer vers QUIC, le très répandu protocole de partage réseau SMB est en train de franchir le cap, Microsoft travaillant à son implémentation dans Azure Files et Windows Server.

Jetez un œil à votre monitoring de flux pour voir quelle est la proportion cumulée de HTTP(S) et SMB sur votre réseau, un indice, c'est très probablement élevé…

À l'heure actuelle les éditeurs de pare-feu recommandent de désactiver QUIC, le temps que son support soit correctement implémenté. Il faudra aussi que les équipements déchiffrant le trafic s'adaptent, comme ils ciblent actuellement le couple TCP+TLS.

La réétude des chaines de sortie vers internet est une opportunité de déploiement d'IPv6, ce qui limiterait les éventuelles étapes de modifications des paquets aux seuls proxys.

Le NAT+PAT de très nombreux flux QUIC est un défi, si l'éditeur de l'équipement introduit des « Application Layer Gateways » pour appliquer des traitements spécifiques aux sessions QUIC, il risque de compromettre certaines sécurités.

Là encore, une session IPv6 supprime ces problématiques. Est-ce trivial ? Songez aux problèmes que vous avez pu rencontrer à titre personnel sur votre réseau domestique avec le fonctionnement de NAT et d'UDP

pour des besoins dynamiques comme les jeux multijoueurs, le P2P ou la VoIP à leurs débuts. Une solution est de rester en HTTP/2 sur TCP, mais pour combien de temps ? Un fonctionnement transitoire pourrait être d'autoriser QUIC sans traitement profond sur les connexions uniquement vers les offres SaaS de confiance dans un premier temps. Et n'oublions enfin pas que QUIC peut porter bien d'autres choses que HTTP.

Notez que ces éléments sont valables pour l'accès à vos ressources par d'autres également, ou par vos télétravailleurs. Ainsi la voie des solutions dites « zéro trust » amène la suppression des VPN et une exposition plus directe des ressources, qui elles aussi basculeront vers QUIC.

Ce protocole vient tout juste d'être ratifié dans les RFC 8999, RFC 9000, RFC 9001 et RFC 9002.

Note sur la proxisation : Afin de bénéficier de ses apports, la couche de proxisation doit être mise à niveau, tant côté navigateur que proxy. Deux modes existent, un mode tunnel, le plus efficace et qui est le seul à pouvoir supporter l'échange initial d'une session QUIC (avec entête longue). Et un mode *forward* où le proxy conserve un rôle de rupture protocolaire, mais seulement une fois la session établie.

À RETENIR

Ce protocole de transport va avoir une courbe de déploiement plus rapide qu'IPv6, les efforts mis en œuvre pour le supporter sur sa chaine proxy ou sur ses frontaux web dans l'autre sens sont l'opportunité de travailler au déploiement de v6 en parallèle.

RÉSEAU INTERNE

Au-delà de la bordure d'infrastructure en contact avec internet, quelles peuvent-être les motivations d'un déploiement sur le périmètre interne ?

Dans la continuité des sections précédentes, le bout en bout est évidemment un avantage à l'heure de la multiplication de l'externalisation de ressources en Cloud. Encore une fois, les fournisseurs de certains produits vont probablement encourager les solutions permettant de limiter les traitements intermédiaires sur les paquets. Notez d'ailleurs que la construction même de l'entête IPv6 amène quelques gains de temps via la suppression du checksum, l'utilisation de champs de taille fixe, ou encore *flowlabel* pour offrir un suivi plus aisé des flux lors de traitements QoS.

Pour les structures de grande taille, IPv6 c'est également la disparition des problèmes induits par la petitesse de l'adressage privé IPv4.

La RFC 1918 offre 17 891 328 IPv4, cela ne représente jamais que 70 000 réseaux en /24. De nombreuses organisations ont déjà atteint la limite du stock, pour de multiples raisons. Affectation par entité, gaspillage et surallocation, non-récupération des adresses lors du décommissionnement d'équipements ou de sites, volonté d'agréger les routes remontant à une époque où les routeurs ne supportaient qu'un faible nombre de routes, transmission à des filiales revendues, mais avec lesquelles des liens persistent, etc.

Si le NAT44 peut répondre de façon inconfortable aux liaisons vers des partenaires et des entités fraichement acquises, il est souvent impensable de découper son entreprise en différents périmètres se recouvrant ; bien que ce cas de figure existe aussi.

D'autres prennent la voie de l'usurpation, et exploitent sur leur réseau interne des IP qui appartiennent à autrui avec plus ou moins de tact. On retrouve deux camps :

- Les prudents, qui mettent en œuvre du double NAT44 et créent un véritable sas compartimentant le routage en bordure d'internet. Le trafic est natté deux fois et peut sans problème avoir la même IP en source et en destination, le NAT masquant un autre NAT, l'écran est total.

  Ces prudents se retrouvent bien dépourvus quand un fournisseur Cloud leur recommande d'annoncer l'IP publique d'un service sur leur backbone interne. Que faire si jamais cette vraie IP publique en recouvre une usurpée du LAN ? D'autant plus qu'un fournisseur peut imposer de nouvelles IP avec seulement quelques semaines de prévenance. Scénario de SF ? Du tout ! Un exemple concret parfait est l'utilisation de la solution de communication de Microsoft, Teams. L'éditeur recommande en effet d'annoncer ses IP publiques, pour des raisons expliquées plus haut dans ce document.

- Les confiants, qui exploitent des IP qui ne seront jamais annoncées sur internet comme celles du département américain de la Défense (DoD) :

  6.0.0.0/8 7.0.0.0/8 11.0.0.0/8 21.0.0.0/8 22.0.0.0/8 26.0.0.0/8 28.0.0.0/8 29.0.0.0/8 30.0.0.0/8 33.0.0.0/8 55.0.0.0/8 214.0.0.0/8 215.0.0.0/8

  Enfin ça, c'est la théorie, car fin 2019, la section 1088 de la loi de budget du DoD prévoyait de vendre ces plages dans les 10 ans. Cependant l'article n'a pas passé le Sénat. Mais qu'en sera-t-il plus tard ?

Si jamais ces adresses se retrouvaient en vente, nul doute qu'une partie finirait dans les mains des principaux fournisseurs Cloud.

Très peu de temps après l'investiture de Joe Biden, l'AS 8003 s'est mis à annoncer des IP du DoD via Hurricane Electric. Officiellement, les propos suivants ont été rapportés au Washington Post :

> *Defense Digital Service (DDS) authorized a pilot effort advertising DoD Internet Protocol (IP) space using Border Gateway Protocol (BGP). This pilot will assess, evaluate and prevent unauthorized use of DoD IP address space. Additionally, this pilot may identify potential vulnerabilities. This is one of DoD's many efforts focused on continually improving our cyber posture and defense in response to advanced persistent threats. We are partnering throughout DoD to ensure potential vulnerabilities are mitigated.*

Certains parlent de collecte de trafic vers ces plages pour de l'analyse (un *honeypot*), le DoD met de son côté en avant la lutte contre le cybersquatting de ses plages IP. Mais s'il s'agissait tout simplement de tester la mise en œuvre du scénario dit « prudent » plus haut au sein même du DoD ? Et de simuler que la vente et l'annonce de ces innombrables IP ne provoquent aucun effet de bord avant de les mettre réellement à vendre ?

En juin 2021, le DoD a annoncé que l'ensemble des nouveaux services déployés après des dates clés devraient l'être en IPv6.

Le 7 septembre 2021, l'immense majorité des préfixes ont migré vers l'AS749, appartenant au DoD, mais n'étant pas son AS usuel, le 721.

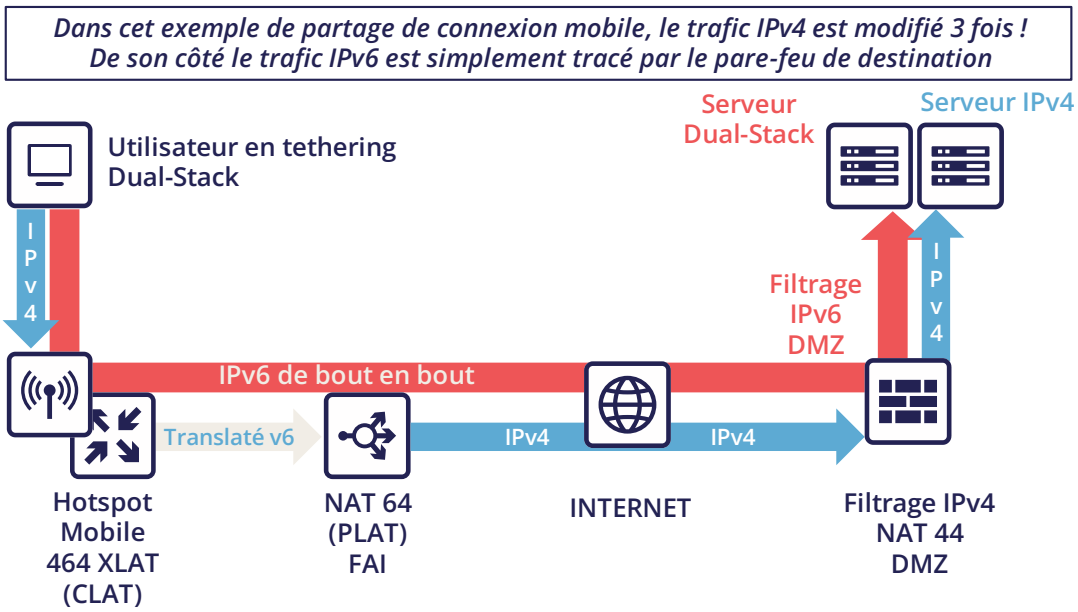
Si vous approchez de la fin du RFC 1918, vous pouvez étudier l'usage de la plage RFC 6598 100.64/10 réservée au NAT44 opérateur afin de partager des IPv4 entre abonnés avec un Carrier Grade NAT. Reste qu'il est recommandé de ne pas assigner ces adresses à des équipements opérateurs comme des routeurs MPLS ou de les exploiter sur des infrastructures Cloud, sauf après validation du fournisseur. Aucun problème en revanche à exploiter cette plage pour ses campus utilisateurs par exemple. Certaines entreprises le font déjà, y compris en France.

Attention, la plage 100.64/10 est de facto utilisée par certains systèmes d'overlay comme l'offre proxy Cloud Zscaler pour monter ses tunnels.

Enfin, si vous êtes joueur, vous pouvez tenter d'utiliser l'ex-classe E (240/4). Cette classe, située aux confins d'IPv4 après la section multicast est réservée pour un usage futur qui ne viendra jamais et inutilisable chez les équipementiers qui reconnaissent d'ailleurs que le travail nécessaire pour normaliser cette plage mettrait plus de temps à atteindre l'ensemble des parcs déployés que de migrer vers IPv6. En vrai, n'essayez pas, sauf en lab par pure curiosité. Google GCP permet de l'exploiter sur ses VPC, mais mentionne de possibles problèmes avec les OS : https://cloud.google.com/vpc/docs/vpc#valid-ranges Cependant, le fait que vous pourriez ne pas pouvoir apprendre ces préfixes sur vos routeurs BGP *on premise* n'est pas évoqué, bien qu'au moins 2 constructeurs supportent via une commande cette espace.

L'usage d'un des scénarios de « triche » décrit ci-dessus pour étendre l'adressage privé ou le simple horizon d'atteindre la fin du stock RFC 1918 semblant proche (quelques années à votre rythme de consommation) devrait vous inciter à vous pencher sérieusement sur un déploiement IPv6.

Pensez aussi au temps passé sur des projets de NAT44 et de réadressage passés et futurs correspondant à l'intégration d'entités fraichement rachetées. Avez-vous déjà vu un département informatique décréter qu'il démarrerait son adressage interne par le bloc 10.255.0.0/16 dans le sens descendant en prévision d'un rachat un jour et qu'avec un peu de chance la nouvelle entité mère aurait de son côté démarré son adressage par 10.0.0.0 ? Plus sérieusement, les conflits d'adressage IP lors de l'intégration de structures génèrent des coûts et des délais souvent importants en plus de complexifier l'exploitation à long terme dans le cas où un NAT44 perdure.



2

Technologies de migration

1. **Dual-Stack** **26**
2. **Mécanismes de transport** **27**
   - ◗ INCLUSION SUR UN UNDERLAY IPV4 EXISTANT 27
     - MPLS 27
     - VXLAN 28
     - SD-WAN 28
   - ◗ ENCAPSULATION SPÉCIFIQUE 29
   - ◗ ET DANS L'AUTRE SENS ? 29
3. **Mécanismes de traduction** **30**
   - ◗ NAT64 + DNS64 30
     - Adressage 31
     - Topologie 31
     - Attention à la MTU 32
     - Note sur le filtrage 32
4. **Quelles technologies pour chaque périmètre ?** **33**
   - ◗ CAMPUS 33
     - NAT64 + DNS64 33
   - ◗ DATACENTER 34
     - Serveurs et applicatifs en *dual-stack* 34
     - Traduction 6/4 35
     - Déploiement natif v6 36
     - Double mono-stack 36
     - Fournisseurs Cloud 36
     - Traduction externe 37
   - ◗ WAN 37
     - Plateforme NAT régionale 37

Technologies de migration

II. Technologies de migration

L'IETF travaille, avec le concours de fabricants et d'éditeurs à offrir des mécanismes permettant de gérer la transition vers IPv6. Ces mécanismes se destinent aux opérateurs et/ou aux entreprises.

Il faut tout d'abord séparer les mécanismes en 2 groupes d'usage :

- Ceux permettant de transporter un protocole sur un réseau de transport d'une autre version. Ils servent typiquement à connecter des ilots IPv6 via des réseaux de transport IPv4 (ou l'inverse). On retrouve des systèmes basés sur de l'encapsulation et d'autres sur de la translation.
- Ceux permettant de faire communiquer un hôte IPv4 avec un hôte IPv6 ou l'inverse. Ils sont forcément basés sur de la translation.

Les 2 sont complémentaires dans la plupart des cas d'usage.

1. Dual-Stack

Le *dual-stack* est simplement l'usage parallèle d'IPv4 et IPv6, il représente donc une simple cohabitation.

Cependant, exploiter parallèlement IPv4 et IPv6 provoque un surcoût, pas seulement pour le réseau avec les aspects de double configuration de routage ou règles pare-feu. L'ensemble des processus de déploiements système doit par exemple fonctionner parallèlement en v4 et v6. Le suivi de la qualité de service demande que chaque service soit monitoré à la fois en IPv4 et en IPv6, etc.

Le *dual-stack* est donc rarement viable à long terme dans une organisation de grande taille en dehors des réseaux portant des terminaux clients, typiquement des campus donc.

Pendant la phase transitoire, il permet tout de même de ne pas toucher à l'existant IPv4 et donc de n'engendrer aucune perturbation ou régression sur les services.

2. Mécanismes de transport

Au commencement les mécanismes étaient conçus pour atteindre des ilots IPv6 à partir d'autres ilots, ou tout simplement depuis un poste de travail.

Ces méthodes comprennent entre autres ISATAP (RFC 5214), Teredo (Microsoft - RFC 4380), 6over4 (RFC 2529), 6to4 (RFC 3056), 6in4 + TB/TSB (RFC 5572), 6rd (RFC 5969), IPv6 GRE (RFC 2473 / 2784), et bien d'autres encore.

La plupart de ceux basés sur des tunnels utilisent le protocole 41 d'IP et/ou de l'encapsulation dans UDP comme le ferait un VPN.

Ces mécanismes sont utiles pour une structure dont la migration d'une ou toute partie du réseau n'est pas possible pour des raisons techniques, pour les opérateurs cela a servi (ou sert encore) à pallier des protocoles ne supportant pas IPv6, c'est le cas des opérateurs câbles avant DOCSIS 3.1.

Comme dans toute technique de tunnellisation, on retrouve le principal inconvénient de la visibilité du trafic du fait de son encapsulation. En entreprise, les besoins de filtrage ainsi que la gestion de qualité de service rendent difficile l'usage de tunnels au vu de la complexité et du faible nombre de solutions compatibles, notamment les pare-feux.

La plupart de ces méthodes amènent des risques de sécurité en entreprise, et sont orientées opérateur.

Les réseaux de transport exploitent souvent des technologies en millefeuille les unes au-dessus des autres comme c'est le cas avec MPLS ou VxLAN. Déployer le *dual-stack* sur l'ensemble des couches de transport est alors rarement pertinent. Il importe cependant de le mettre en œuvre dans la couche supérieure, celle visible des utilisateurs du réseau, l'*overlay*.

INCLUSION SUR UN UNDERLAY IPV4 EXISTANT

Le transport actuel en place permet souvent d'isoler des contextes clients de bout en bout au sein même de l'organisation, tant sur le backbone qu'en datacenter.

Si les VRF lite sont encore majoritaires sur des campus, les autres périmètres utilisent massivement des technologies à base d'*underlay*. Il est alors relativement aisé de faire transiter IPv6 dans l'overlay.

MPLS

MPLS est un maillon souvent présent en entreprise, directement ou de façon externalisée via les offres d'interconnexion de sites des opérateurs professionnels.

MPLS permet de faire transiter IPv6 via 2 approches :

- 6PE (Provider Edge RFC 4798) qui fournit v6 dans la table native (GRT) des équipements, utile uniquement si on fournit des services via la GRT (comme un accès internet ou la TV pour un FAI grand public) ;
- 6VPE (RFC 4659), le V fait toute la différence, ici on transite simplement des VPNv6 aux côtés des VPNv4, c'est donc l'équivalent d'un L3VPN, méthode la plus simple et répondant à la majorité des cas d'usage.

Il est possible d'utiliser un IGP IPv6 et LDPv6 pour construire un MPLS basé sur un underlay v6, peu d'intérêt de basculer en dehors de saisir une opportunité de gros projet de refonte. Et surtout cela ne fournit pas pour autant v6 dans les L3VPN de l'overlay ce qui le sujet de ce document.

L'implémentation de 6VPE est la voie à suivre, elle sera facile à mettre en œuvre sur les équipements actuels et demandera peu de configuration.

Si votre MPLS utilise le récent MP-BGP EVPN comme plan de contrôle au lieu de MP-BGP L3VPN, le support d'IPv6 ne posera là non plus aucun problème.

Notez que sur vous pouvez utiliser un next hop IPv6 pour des routes VPN IPv4 grâce à la RFC 8950 si vous vous orientez vers un underlay IPv6.

VXLAN

Majoritairement utilisé en conjonction d'EVPN, VxLAN permet de résoudre les écueils des anciennes *Fabrics DataCenter* L2 SPB et est devenu le standard du marché. Plus rarement, on le retrouve sur des backbones qui ont abandonné MPLS pour profiter plutôt d'EVPN qui a été disponible comme *control plane* pour VxLAN avant MPLS.

Tout comme MPLS, VxLAN encapsule. La question de la compatibilité IPv6 se pose donc dans la couche supérieure (*overlay*) destinée à fournir le service aux clients. La configuration d'un *overlay* IPv6 est mure chez les grands constructeurs, vérifiez tout de même le bon support des mécanismes liés au multicast comme PIM *snooping* ou BiDir.

Si l'*underlay* peut rester en IPv4, notez que l'IETF travaille à la mise en œuvre de RIFT (Routing in Fat Tree), afin de faciliter le déploiement des *fabrics* de CLOS dans la lignée du *zero touch provisionning*. Ciblant les *fabrics* avec underlay en iBGP, il prévoit que les adresses de *loopbacks* et les *route reflectors* soient en IPv6. Difficile de dire s'il aboutira intégralement avant que les *fabrics* migrent vers SRv6 (RIFT prévoit d'ailleurs également un mécanisme d'échange de Node-SID et de préfixe global *segment routing* SRGB afin de faciliter là encore le déploiement). Voir https://datatracker.ietf.org/wg/rift/documents/.

SD-WAN

Les produits SD-WAN fonctionnent généralement avec du DPI et de la classification des flux en *ingress* pour appliquer de la QoS et éventuellement choisir un chemin de transit (internet/MPLS/...). Le trafic est ensuite souvent chiffré dans un tunnel IPsec propre au contexte client puis se retrouve encapsulé jusqu'au routeur de destination (sauf quand une analyse requiert sa décapsulation sur le hub par exemple).

L'*underlay* est conçu pour exploiter un réseau en place basé sur IPv4 afin de limiter les préparatifs à la mise en place de ce type de produit.

Ces produits ciblent principalement de grands réseaux constitués de petits et moyens sites avec des équipements de gamme dédiés et/ou intégrés à des gammes d'équipements plus classiques. En concentration sur les accès datacenter, on retrouve de gros châssis là aussi issus de gammes dédiées ou conventionnelles.

Lorsque l'on souhaite utiliser une bonne partie de solutions du marché sur de campus de plus de 2000 utilisateurs, on atteint souvent les limites des gammes dédiées, bien que les constructeurs progressent et tentent de couvrir les derniers percentiles d'usages manquants.

Reste qu'IPv6 est rarement requis par les clients puisque ces solutions sont destinées à leur réseau interne. Il en ressort que la compatibilité des solutions SD-WAN du marché varie fortement d'un constructeur à

l'autre, mais aussi d'une version à l'autre. Chez certains, elle arrive d'abord sur l'*overlay*, chez d'autre sur l'*underlay* . Il est donc important de suivre la *roadmap* du constructeur et de tester la solution avant un déploiement v6, mais aussi à chaque nouvelle *release* majeure, le code pouvant être fortement remanié vu la vitesse d'évolution de ces solutions et la concurrence.

Enfin, l'aspect *Local Breakout* de ces solutions est un autre élément qui intègre également progressivement IPv6. Souvent avec toute une couche de services locaux de sécurité couramment dénommée « SASE ».

ENCAPSULATION SPÉCIFIQUE

Il n'est parfois pas possible de faire transiter IPv6 sur un périmètre de transport, et comme vu précédemment peu de solutions techniques sont exploitables des 2 côtés sur des gammes d'équipements d'entreprise.

Il reste alors la possibilité de *tunneliser* le trafic IPv6. Cela peut se faire notamment via des solutions connues comme GRE/mGRE ou IPsec (ce dernier est toutefois moins performant au vu des ressources de chiffrement nécessaires).

Enfin, vous pouvez configurer 6in4 sur une large part des routeurs du marché si aucune autre solution évoquée précédemment ne vous satisfait, 6rd est aussi souvent disponible, mais ne cible principalement que des topologies nord-sud.

Nous vous déconseillons en revanche de vous pencher sur 6to4 (*endpoint* non configurable), 6over4 (basé IPv4 multicast), ISATAP (basé découverte DNS) et Teredo (encapsulation UDP) qui sont maintenant très peu utilisés.

La disponibilité de telle ou telle méthode sur vos équipements, en conjonction avec l'intégration à votre routage, vous guidera sur le choix à retenir.

ET DANS L'AUTRE SENS ?

Comme vu au début du chapitre, des technologies de transitions existent également pour pouvoir se passer d'IPv4 sur son backbone. Le cantonnant alors aux réseaux utilisateurs, IPv4aaS.

Certains opérateurs en sont déjà à se séparer d'IPv4 sur leur backbone, pour économiser des adresses et même partager les IP entre les abonnées en répartissant les ports. Les approches dites *Address+Port* (AP) se sont répandues. D'abord DS-Lite, puis *Lightweight 4over6* (lw4o6) et plus récemment MAP T/E et 4rd. Ces 2 dernières dominent les déploiements actuels, notamment grâce à leur capacité d'agrégation évitant de devoir terminer un nombre astronomique de tunnels et autant de routes au sein du cœur de l'ISP.

Ceux qui n'ont pas encore migré vers un backbone IPv6 et manquent d'IPv4 font du simple NAT44 sur un lot d'équipement central de type CG-NAT et utilisent le fameux adressage 100.64/10 de la RFC 6598.

Ceux qui sont en IPv6 fournissent généralement IPv4 via l'une des méthodes suivantes :

- 4rd (RFC 7600) qui fonctionne de façon opposée à 6rd et offre une méthode *stateless* efficace. Il peut fonctionner en mode *mesh* ou *hub&spoke* ;
- MAP (T ou E) (RFC 7599), disponible en mode translation et en mode encapsulation, est également *stateless* ;
- D'anciens déploiements exploitent DS-Lite et Lw4o6.

Les 2 premiers sont assez proches et exploitent des règles communes sur un domaine, des routeurs de bordure (BR), des bits EA pour définir le niveau de partage d'IP, annonce des règles de mapping via DHCP aux équipements terminaux (CPE).

Les implémentations de ces techniques côté routeur client se font en logiciel, on les retrouve dans nos routeurs domestiques. Peu probable en revanche de trouver un équipement sachant traiter MAP ou 4rd via son ASIC sur le côté client, les équipements haut de gamme ne gèrent que l'aspect *Border Router*.

Concernant MPLS et VxLAN, il est possible de remplacer IPv4 par IPv6 sur l'*underlay* de transport, mais les implémentations ne sont pas très répandues et il reste nécessaire de valider avec l'équipementier.

Pour les cas particuliers où le transport ne peut faire transiter IPv4, on retrouve la même chose que précédemment. Des tunnels spécifiques pour connecter des îlots IPv4 entre eux. On peut alors mettre en œuvre GRE/mGRE, 4in6. 4rd ne semble pas encore très présent dans les routeurs d'entreprise.

À RETENIR

Vous pouvez généralement transporter facilement du trafic IPv6 sur un underlay IPv4, et pouvez donc attendre une opportunité de projet global de refonte pour basculer l'underlay. Si vous vous lancez sur un nouvel environnement, préférez partir directement sur un underlay IPv6. De plus, réfléchissez à avoir une topologie et un adressage adapté à SRv6, ça vous fera gagner du temps plus tard si vous ne le mettez pas en œuvre immédiatement.

3. Mécanismes de traduction

La traduction/translation a pour objectif de permettre des échanges entre les clients et les serveurs utilisant des versions différentes d'IP.

Si l'on s'en tient à la logique de transition *dual-stack*, on doit déployer IPv6 partout. Or cela occasionne une double exploitation large et ne fonctionne que si tous les éléments sont compatibles *dual-stack*. Comment faire discuter des clients IPv6 avec des serveurs IPv4 (ou l'inverse) ?

NAT64 et DNS64 apportent une réponse conjointe déjà largement déployée permettant à des clients IPv6 de contacter des serveurs IPv4. À l'inverse, SIIT (Stateless IP/ICMP Translation) laisse des clients IPv4 entrer dans un réseau uniquement IPv6.

Évidemment, l'entête IPv6 étant plus long, il est plus simple techniquement de conserver l'information d'entête en faisant rentrer des clients IPv4 vers un serveur IPv6 que l'opposé. Mais le sens de déploiement est question de besoin, de stratégie, d'ordonnancement et d'homogénéité.

NAT64 + DNS64

Le NAT64 (RFC 6146) couplé à DNS64 (RFC 6147) utilise le principe de DNS « menteur » en conjonction avec un traducteur pour permettre à des terminaux IPv6 d'accéder à des ressources IPv4. L'IETF publie un guide de déploiement (RFC 7269).

Lorsqu'une ressource ne dispose pas d'enregistrement DNS AAAA, le serveur DNS va en synthétiser une à partir d'un préfixe IPv6 /96 et de l'adresse IPv4 /32 retournée dans l'enregistrement DNS A.

Le terminal initiera alors une connexion à destination d'une IPv6.

Quelque part sur le réseau (nous verrons plus loin pour les emplacements), un équipement annonçant le préfixe /96 va recevoir la connexion. Cette plateforme de NAT64 va enlever le préfixe IPv6 /96 de la destination et remplacer l'entête IPv6 par une IPv4. Ce faisant, le paquet subit un NAT avec une adresse source dans le pool NAT (ainsi qu'un port source pour le PAT) et envoie le paquet. Grâce au maintien d'une table de session, il effectuera l'opération inverse sur le paquet retour.

Notez bien que le terminal n'est à aucun moment conscient de la supercherie. En résultent des problèmes sur les protocoles P2P ainsi que ceux qui embarquent l'adresse dans le *payload* comme le SIP. Des fonctionnalités d'ALG SIP, H323, IPsec AH, SCCP, etc. peuvent être implémentées sur les plateformes de NAT64 afin de résoudre le problème, mais potentiellement au prix d'une dégradation des performances.

La validation DNSSEC effectuée par l'hôte sera également impossible avec ce scénario. Problème pouvant être résolu si l'hôte avait conscience du NAT64 (ce qui est le cas sur mobile avec la configuration APN ou encore lorsque la RFC 7050 est utilisée, mais cette dernière n'est pas très utile avec les OS desktop puisqu'ils ne le supportent pas encore. Il existe également des volontés de pouvoir informer les hôtes via DHCPv6 et PCP du préfixe NAT64)

Côté application, le NAT64 fonctionne dès lors que celle-ci peut ouvrir des sockets en IPv6 et qu'elle fait bien appelle un *hostname* et non une IP littérale.

Adressage

Sur un petit réseau, une plateforme unique suffira, elle exploitera généralement le préfixe WKP (RFC 6052 *Well Known Prefix*) ou un autre préfixe dit (*Network Specific Prefix*) défini au sein de l'adressage de l'entreprise avec un /96.

Attention, si vous piochez dans la plage ULA, le NAT64 sera toujours dépriorisé par rapport à IPv4.

N'oubliez pas dans votre projet que si 99% des connexions sont initiés par les terminaux clients, il existe des cas particuliers comme la prise en main à distance par le support. Et évidemment la téléphonie en P2P. Ces derniers requerront une compatibilité intégrale à IPv6.

Sur un grand réseau, il est préférable de disposer de plusieurs plateformes, chacune avec son préfixe. Une plage est d'ailleurs réservée à cet usage, bien que non obligatoire. La 64:ff9b:1::/48 (RFC 8215)

Topologie

Le choix de l'emplacement de ces plateformes variera selon vos contraintes.

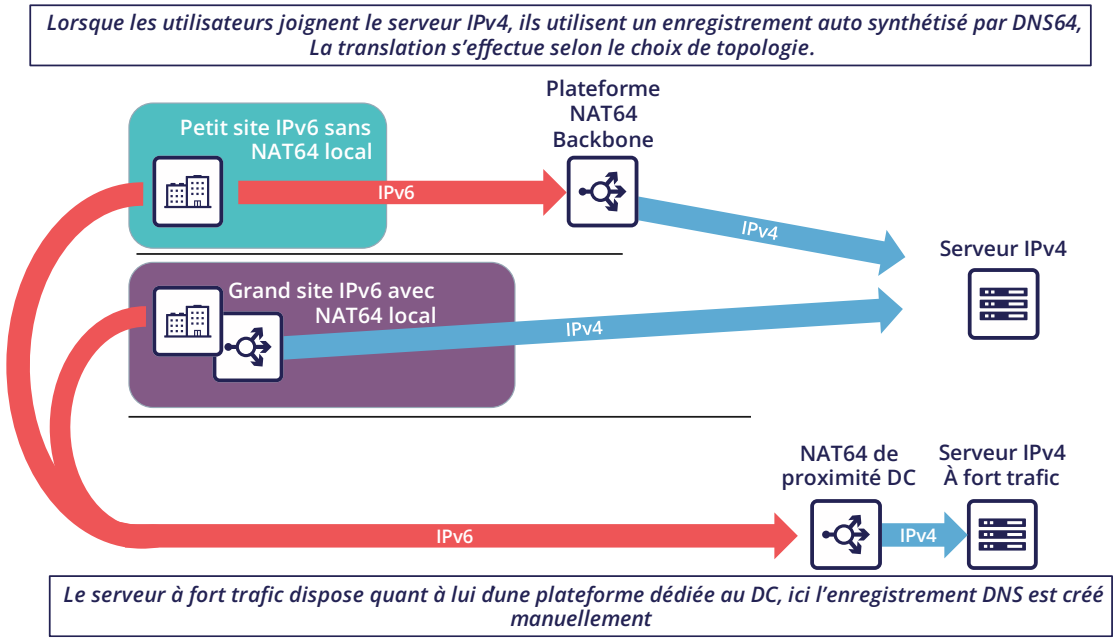
Les placer directement sur les sites évitera le *tromboning* en datacenter (aller-retour). Mais cela nécessitera l'usage d'autant de préfixes NSP que de sites, en sus d'adapter la configuration du DNS64 à chaque fois. Via un *proxyDNS* sur chaque site proprement configuré (il peut s'agir d'un Bind9 ou *Unbound* par exemple)

Il est également possible d'utiliser le même préfixe sur chaque site tant que ceux-ci sont des culs-de-sac et que les annonces de routes vers le backbone filtrent le NSP. On facilite alors la configuration DNS64.

Placer le NAT64 sur les sites implique dans tous les cas la nécessité de conserver un backbone IPv4. Remarquez qu'il sera de toute façon difficile de s'en passer rapidement, les sites n'hébergeant rarement que des postes utilisateurs. Créer des sessions NAT sur X sites signifie aussi collecter les logs de création de sessions sur l'ensemble des sites. Enfin il faudra provisionner de nombreux pools d'IPv4 de NAT et adapter les ACL de filtrage.

À l'inverse le centraliser facilite sa mise en œuvre sur tous les plans, mais n'est pas souhaitable s'il génère du *tromboning* sur des flux qui auraient pu rester internes aux sites.

Un bon compromis est de disposer de passerelles NAT64 sur les plus grands sites, notamment ceux qui hébergent des services localement et ont besoin que ces services fonctionnent même lors d'une coupure WAN. Pour les autres, le centraliser en entrée datacenter ou en bordure de backbone.



Attention à la MTU

L'en-tête IPv6 fait 20 bytes de plus qu'IPv6, un gros paquet IPv4 revenant à une plateforme NAT64 peut donc être perdu si la plateforme ne gère pas la fragmentation correctement. Comme la fragmentation ne peut se produire qu'en IPv4, avant la traduction de retour, il faut généralement définir un réglage MTU spécifique NAT64 qui touche au traitement interne et pas à un vrai MTU d'interface.

La plateforme peut également renvoyer un message ICMP "Fragmentation Needed" au serveur IPv4.

La 2e option est utile pour certains trafics, et assurément pour ceux qui ne supportent pas la fragmentation IPv4 comme TFTP. Voir le RFC 7915.

Dans l'autre sens, vous devez vous assurez que le PMUT-D envoie au moins 1280 bytes, positionnez donc toujours la MTU de l'interface IPv4 à au moins 1260 (1260 + 20 IPv6 overhead). Sans cela des attaquants pourraient profiter de la situation pour effectuer des fragmentations non désirées. Voir RFC 7269.

Note sur le filtrage

Une fois le NAT64 traversé, comment filtrer les flux ? Si le NAT64 est effectué au plus proche de l'utilisateur l'identification d'une population reste simple, s'il est centralisé cela demande beaucoup d'ACL granulaires au même endroit.

La solution réside dans la segmentation des pools IPv4 de NAT, créez des règles de correspondance afin que les machines derrière un préfixe IPv6 X ressortent avec un pool IPv4 de NAT dédié Y, et ainsi de suite. Là encore, plus il y'a de segmentation, plus il sera complexe de le mettre en œuvre sur les sites.

4. Quelles technologies pour chaque périmètre ?

Maintenant que vous avez connaissance de la technique permettant à un client d'échanger avec un serveur qui ne parle pas la même langue ainsi que les modes de transit, voyons la pertinence de chaque solution.

Un angle d'approche idéal est de se poser la question de ce qui est le plus facile à migrer.

Quels types de terminaux sont présents sur le réseau ?

CAMPUS

Côté utilisateur on trouve des postes de travail généralement homogènes, avec un écosystème identique reproduit par site/plaque géographique et d'autres briques centralisées. Cet écosystème comprend entre autres le stockage de fichiers, l'annuaire d'authentification, la messagerie et les autres outils collaboratifs, dont la téléphonie, l'impression, le proxy, l'agent de gestion du poste, celui de protection, et bien sûr les applications métiers. Ces dernières sont maintenant presque systématiquement des applications web et reposent donc souvent sur le navigateur côté client.

Les équipements réseau suivent eux aussi assez souvent des schémas d'architectures répliqués, avec 2 à 3 générations cohabitant à l'échelle de l'organisation. Malheureusement les équipements des gammes campus sont ceux le plus en retard sur la compatibilité IPv6, notamment sur les aspects sécurité.

Cependant, difficile de ne pas réaliser que, si ce périmètre est vaste, il est aussi relativement homogène. Cette homogénéité est une force. En déployant IPv6 en dual stack sur un site de chaque type en mode pilote, et en l'implémentant sur les éléments de l'écosystème « bureautique/Workplace » ; il devient ensuite possible d'industrialiser le déploiement.

Celui-ci peut survenir lors de tout remplacement d'équipement du site, de déménagement, etc.

Éventuellement, il est même envisageable de retirer IPv4 des campus afin de s'affranchir de la gestion de la double pile sur les campus. Ce scénario est d'ailleurs celui à privilégier si votre organisation manque d'espace d'adressage privé IPv4.

NAT64 + DNS64

Si cette voie correspond à vos besoins, il faudra alors étudier le ou les emplacements des fonctions de NAT64 et de DNS64. On reprend les éléments de la section topologie :

Si vos sites ne portent aucun service compatible uniquement IPv4 et/ou ne reposent que sur des serveurs en datacenter ou en Cloud, inutile d'avoir du NAT64 sur site, c'est typiquement le cas de figure des agences bancaires par exemple.

À contrario, un site industriel de grande taille aura souvent des serveurs bureautique et métiers sur place, ceci afin que la production ne dépende pas intégralement de la fiabilité du WAN. Et une partie de ces systèmes ne fonctionnera qu'en IPv4. Il faut alors pouvoir échanger localement en IPv4.

Si peu de clients doivent exploiter les applications concernées et qu'ils sont cantonnés à des réseaux bien précis, la conservation du dual-stack semble indiquée. Ce cantonnement peut être physique ou logique et appliqué à l'aide d'un serveur radius par exemple.

En revanche, si beaucoup de postes doivent pouvoir joindre une ressource locale en IPv4, l'implémentation d'un NAT64+DNS64 local devient intéressante, et est même préconisée si vous rencontrez un manque d'IPv4 privées.

Ce NAT64 sera déployé en mode *stateful* (avec tables de sessions et affectation par port).

S'il est possible de retirer IPv4 grâce à NAT64 chaque fois qu'un site est migré, un élément pose problème : la téléphonie. En effet, si l'immense majorité des flux sont émis à destination d'un serveur, la téléphonie présente la particularité de générer du trafic UDP P2P direct entre 2 utilisateurs. À moins que votre équipementier ne propose un mécanisme permettant de ségréguer la population IPv4 et IPv6 automatiquement afin de forcer la traduction via un serveur media relai dual-stack lorsqu'un appel est établi entre les 2 domaines, il vous sera nécessaire de déployer IPv6 sur l'ensemble des campus avant d'entamer le retrait d'IPv4 sur une partie des terminaux, y compris ceux en distanciel (VPN ou autre).

N'oubliez pas que certains services peuvent être amenés à initier une session IPv6 vers un poste de travail, c'est par exemple le cas du *helpdesk* pour se connecter à un poste et dépanner. Le *helpdesk* aura lui aussi donc besoin d'une connectivité IPv6. Et si ce helpdesk est externalisé, il faudra revoir vos contrats.

Cette contrainte liée au trafic SIP et RTP force donc un traitement global avant libération d'IPv4.

DATACENTER

Les ressources datacenter, qu'elles soient internes ou en Cloud, peuvent être très diverses ou relativement homogènes. Tout dépend de votre activité et de votre historique.

Si les GAFAM ont publié des ressources de transition vers IPv6, il est rare qu'elles soient applicables dans une entreprise de grande taille. Pour comprendre, il suffit d'aborder les services en termes de quantité et d'échelle de déploiement. Lorsqu'on fait tourner une cinquantaine de services sur des centaines de milliers de serveurs, on est forcément industrialisé, avec un orchestrateur qui appelle des automatisations. Il est alors faisable de tester une migration vers IPv6 en pilote, service par service, puis de généraliser. Une approche similaire à celle évoquée plus haut pour les campus, beaucoup de machines, mais avec une configuration similaire. De façon arbitraire, disons qu'un acteur majeur du net à un ratio de 100 000 machines par service, quel est celui d'une entreprise ?

Listez vos serveurs, VM, containers, et divisez par le nombre d'applications que votre SI comporte. Il est probable que le résultat se trouve entre 3 et 10. Pas vraiment de quoi parler de mise à l'échelle donc. Mais ne nous décourageons pas, ces serveurs exploitent souvent un nombre de *middleware* bien plus limité, une dizaine. La compatibilité IPv6 de ces derniers est bonne, mais il faudra tout de même qualifier le bon fonctionnement de chaque application. La section « applicatifs » vous aidera.

Serveurs et applicatifs en *dual-stack*

Comme exposé dans la section *dual-stack* des technologies de migration, conserver tout en double dans la durée amène divers surcoûts. Il est idéal de fournir la connectivité IPv6 dans vos souches serveur afin d'être prêt, quel que soit le scénario retenu sur la partie système. Ces aspects sont couverts plus loin dans le document. Le *dual-stack* reste toutefois la préconisation pour les services à fort trafic et les services critiques (DNS, annuaire, proxy, NAS,etc.)

Traduction 6/4

Le cycle de vie des applications peut s'étaler sur 2, 3, peut-être 8 ans ou plus. Difficile d'attendre autant pour offrir l'accès à celles-ci à des clients ne disposant pas d'IPv4 natif.

Si votre application est exposée sur internet, vous pouvez vous contenter de laisser faire le NAT64 côté opérateur pour les clients qui n'ont plus d'IPv4 natif, smartphone en général. Toutefois, ceci complexifie le *troubleshooting* de votre côté puisque vous n'avez pas la main, de plus le service est fourni avec une qualité dépendant de l'opérateur. Si de la latence ou des coupures de sessions surviennent, l'utilisateur mettra la responsabilité sur vous et votre image en pâtira. Il n'a aucune idée du traitement intermédiaire de son opérateur.

Deux possibilités s'offrent à vous pour exposer en IPv6, un NAT64 ou un reverse proxy.

Afin de limiter la charge de travail, vous pouvez vous reposer sur un équipement présent pour faciliter les choses. Si votre serveur de présentation est simplement situé derrière un pare-feu sans autre intermédiaire, alors le NAT64 statique semble tout indiqué. Vous ferez alors correspondre une IPv6 de NAT à chaque IPv4 de serveur de façon statique, et publierez l'enregistrement DNS AAAA correspondant. Vous pouvez même faire correspondre des préfixes IPv6 en /120 avec des réseaux IPv4 /24 par exemple, ce qui représente encore moins de règles. Le pare-feu effectuera le NAT+PAT et tracera les sessions.

Les serveurs IPv4 devront tracer le port des sessions en plus de l'IP afin de pouvoir corréler les logs du pare-feu (cf. RFC 7768).

Pour d'autres serveurs avec moins de trafic, du classique NAT64 *stateful* suffira. Rappelez-vous toujours qu'il requiert alors l'implémentation de DNS64 sur la chaine de résolution et le choix d'un *Network Specific Prefix* en /96 que vous exposerez sur internet. Même chose pour le réseau interne.

L'hybridation est une bonne réponse : un NAT64 statique avec un AAAA créé manuellement pour chaque serveur frontal très utilisé, et du NAT64 dynamique pour le reste.

Ce traitement de NAT64 s'effectue à bas niveau, avec des performances élevées sur les équipements récents. Il nécessite en contrepartie de synchroniser les tables de sessions pour garantir la haute disponibilité du mode *stateful*. Ce mode n'est pas adapté à des serveurs publiés en *anycast* puisqu'il existe une chance, bien que faible, pour que le client bascule d'une plateforme NAT64 à une autre durant la vie de la session. Une rupture se produirait alors. (Voir plus loin le SIIT)

Pour les besoins de granularité sur le trafic, par exemple car le serveur IPv4 à atteindre en interne est situé dans un autre datacenter que la plateforme d'entrée NAT64, vous pouvez utiliser des pools de SNAT IPv4 dédiés afin de respecter les principes de filtrage fin (similaire à la problématique d'ACL exposé plus haut).

Avec un SLB (*load balancer*) au niveau 4, le NAT64 est également préconisé, en revanche si celui-ci fonctionne dans les couches hautes (L7 avec ou sans pare-feu applicatif WAF, HTTP par exemple) alors la rupture protocolaire provoquera la reconstruction du trafic dans l'autre version du protocole et la question ne se pose alors plus. Reste qu'il est alors souvent utile de copier l'adresse IPv6 du client dans un champ HTTP « X-Forwarded-For » quand c'est ce dernier qui est utilisé. La visibilité du client peut ainsi remonter jusqu'au serveur.

L'entrée publique du *datacenter* étant généralement constituée de plusieurs de ces éléments, rappelez-vous d'amener IPv6 à minima jusqu'aux équipements disposant de règles fines.

Prenons l'exemple d'un trafic internet traversant un pare-feu L4 puis un pare-feu applicatif *reverse proxy* HTTP (WAF) avant d'atteindre le serveur. On serait tenté de se débarrasser d'IPv6 dès le pare-feu réseau et de faire

du NAT64. Dès lors, certaines règles de détection du reverse proxy ne fonctionneraient plus puisqu'il ne verrait toujours que le même pool d'IP SNAT du pare-feu réseau et non les IP des clients.

Concernant l'accès interne à une application incompatible IPv6, les méthodes de NAT64 ou de *reverse proxy* peuvent être là aussi employés. Enfin, pour une application interne qui ne fonctionne toujours pas avec ces méthodes, il reste possible d'utiliser un VPN interne pour joindre l'îlot IPv4 à partir d'un poste IPv6. Déplacer l'ensemble des clients concernés dans un VDI en datacenter représente une autre piste viable, mais couteuse.

Déploiement natif v6

La proportion de clients IPv6 augmentant, pourquoi ne pas envisager de fournir ses services exposés sur internet nativement en IPv6 et de mettre en œuvre une translation pour les clients IPv4 ?

C'est le principe du *Stateless* IP/ICMP Translation (SIIT), dans sa version originale il se limite à un 1 pour 1 bi directionnel entre IPv4 et IPv6. Ce qui nécessite évidemment autant d'IPv4 que d'IPv6 des 2 côtés et n'est donc exploitable que sur d'infimes périmètres bien particuliers au vu des limites qu'il impose. Par exemple pour quelques serveurs entre eux.

Dans son parfum DC, SIIT-DC propose l'accès à des serveurs IPv6 à partir de clients IPv4, sans maintenir de table d'état.

On va pour ce faire réserver un préfixe IPv6/96 qui servira à représenter les IPv4 dans les 32 derniers bits. Ainsi le système peut être multiplié sans contrainte et supporter l'anycast et la dissymétrie (puisqu'il ne se repose pas sur une table d'état). Par défaut le préfixe sera à prendre dans la plage 64:ff9b:1::/48 (RFC 8215)

Il est évidement possible d'exploiter plusieurs préfixes, par exemple pour rattacher les paquets mappés à l'entrée internet IPv4 où ils sont arrivés. Bien utile lorsque la chaine internet a des contrôles *stateful* de son côté (IPS, etc.)

Il faut cependant toujours prévoir autant d'IPv4 qu'il y'a de serveurs IPv6 à exposer.

Et lorsqu'il y'a quand même besoin d'IPv4 sur un serveur quelque part au fin fond du DC, il est possible d'utiliser de la double translation SIIT (*Dual Translation*). Le trafic internet IPv4 est traduit en IPv6, traverse le *datacenter*, puis est retraduit par un équipement au plus proche du serveur.

Bien qu'on parle ici d'internet, la même topologie peut être mise en œuvre pour des clients IPv4 internes.

Double mono-stack

Une méthode peu employée, mais viable sur d'immenses clusters est de déployer des serveurs exposant leurs services uniquement en IPv6 en parallèle des autres serveurs existants IPv4. Si cette technique ne va pas dans le sens de l'homogénéisation de la configuration, elle a l'avantage de ne pas toucher à l'existant. Ainsi les clients IPv4 en production n'ont aucun risque de perturbation ou de régression.

Fournisseurs Cloud

Si la fourniture d'IPv6 se fait sans problème dans les offres IaaS des leaders du marché, il reste en revanche du chemin à parcourir pour les offres PaaS.

Par exemple la plupart des services de *load balancing* ne sont pas encore compatibles, et quand ils le sont (comme AWS NLB depuis fin 2020), c'est uniquement sur la partie exposée au client, et pas encore sur celle en *backend* (ce qui est, il faut le reconnaitre, moins urgent).

Traduction externe

Pour les services exposés à internet, vous pouvez aussi vous reposer sur un CDN ou autre intermédiaire qui a la capacité de mettre à disposition vos ressources en *dual-stack* alors même que le *backend* reste dans l'une ou l'autre version du protocole IP seulement.

WAN

Le WAN lui-même ne fournit pas de service directement aux utilisateurs, il est là pour transporter du trafic entre les sites. Vous pouvez remonter à la section mécanismes de transport pour voir comment transporter des paquets IPv6.

Plateforme NAT régionale

Selon la taille de vos sites et la cartographie de vos flux, vous pouvez envisager d'installer des plateformes régionales de NAT64 sur votre backbone. Souvenez-vous que l'ajout de ce service statefull vous obligera à avoir des flux symétriques.

Un tel service peut également être fournit par votre MPLS opéré tant que vous ne chiffrez pas le trafic inter-sites de votre côté.

3

Ordonnancement des briques

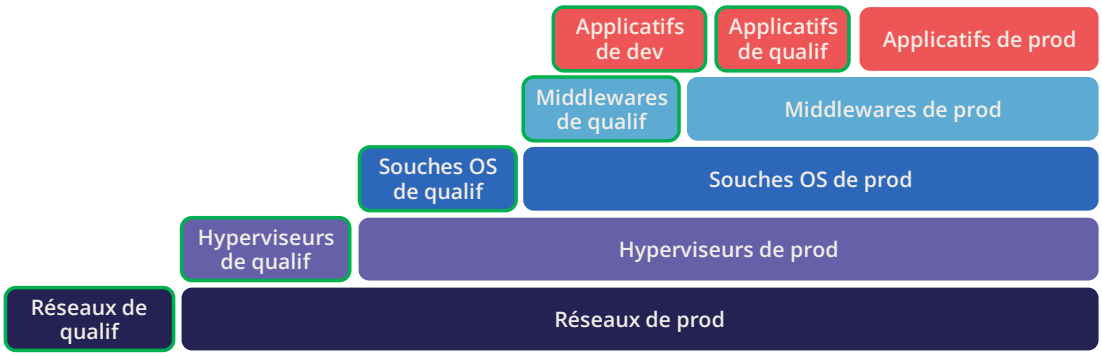
1. **Avant de démarrer** **40**
2. **Réseau** **41**
   - MATURITÉ 41
   - HARDWARE 42
   - MAQUETTAGE 42
   - ROUTAGE INTERNE 45
     - BGP 45
     - IGP 45
   - FILTRAGE ET TRAÇABILITÉ 46
3. **Services d'infrastructure** **46**
   - SIEM 46
   - DNS/IPAM/DHCP 47
   - VPN, PROXY ET REVERSE PROXY 47
     - Extérieur 47
     - Intérieur 47
   - SOUCHES D'OS 47
     - Précédence 48
     - Agents 49
   - SERVICES BUREAUTIQUES 50
     - Annuaire 50
     - Hébergement de fichiers et packages 51
     - Communication 51
   - APPLICATIFS 51
     - Cas de la manipulation d'IP 53

Ordonnancement

III. Ordonnancement des briques

Le déploiement d'IPv6 est en toute logique à effectuer à partir du bas, la couche réseau. Et avant tout déploiement, il est cohérent de maquetter le comportement de chaque élément. Très peu de structures disposent d'environnement de maquettage et de qualification de bout en bout aussi bien horizontalement dans la même couche que verticalement entre les couches. Par exemple, vos maquettes réseaux campus, datacenter et sécurité sont peut-être gérées par des équipes différentes et ne sont pas interconnectées dans une topologie proche de celle de production, c'est une rupture horizontale. Si un serveur de qualification applicative tourne sur un réseau avec des routeurs de production, on a une rupture verticale. Ce qui fait sens, sinon comment déboguer un problème si toutes les couches de pile sont en test, ça ressemblerait à un rolla-bolla multiniveau…

On retiendra donc que chaque couche dispose de ses environnements de tests, et que ceux-ci tournent sur des environnements de production des couches inférieures. En somme, tout système de qualification s'exécute lui-même sur une production sous-jacente (sauf pour la fondation qu'est le réseau). On peut représenter cela ainsi :



1. Avant de démarrer

Avant même de choisir par où commencer, commencez par vous assurer que tous vos cahiers des charges / RFP / demandes de prestation en cours de rédaction ou à venir intègrent la compatibilité IPv6 et garantissent son fonctionnement. Ces processus sont souvent longs à faire modifier, autant s'y atteler dès le début.

Cela comprend également les *process* de *build*, de *run*, de cycle de vie ainsi que tout ce qui s'y rapporte.

2. Réseau

Le réseau est la première brique à traiter, commençons par quelques questions à se poser sur les équipements :

- **Un équipement fonctionne-t-il en IPv6 ?** (Demander à l'équipementier, intégrateur, testeur, etc.) ;
- **Y'a-t-il un écart de fonctionnalité / des régressions avec IPv6 comparé à IPv4 ?** (i.e. une sonde a-t-elle les mêmes capacités de détection, la même application de règles et de signatures ?) ;
- **Y'a-t-il un écart de performance avec IPv4 ?** (i.e. nombre de paquets filtrés par seconde sur un pare-feu est-il du même ordre de grandeur ? les fonctionnalités gérées en hardware par un ASIC sont-elles équivalentes ? comme l'offloading IPsec ou TCP sur la chaine OS VM/Hyperviseur/Driver NIC ou TLS sur un *load balancer*) ;
- **L'administration dudit équipement est-elle possible via IPv6 ?** Son contrôle, sa supervision, etc. ? ou ne fait-il que du transport IPv6 sur son data plane sans gestion côté administration.

D'une façon générale, la difficulté d'implémentation augmente avec la capacité de l'équipement à monter dans les couches du modèle OSI, car de plus en plus de fonctionnalités doivent être testées et que le risque d'oubli / hétérogénéité de configuration augmente également.

On traite donc facilement les routeurs, une fois la maitrise des protocoles de routage bonne, tout en évitant de fournir immédiatement le *dual-stack* aux réseaux terminaux pour se laisser le temps de valider le fonctionnement des mécanismes de sécurité propres à v6 dans les réseaux d'hôtes.

Ensuite les réseaux d'infrastructure sans toutefois atteindre l'utilisateur.

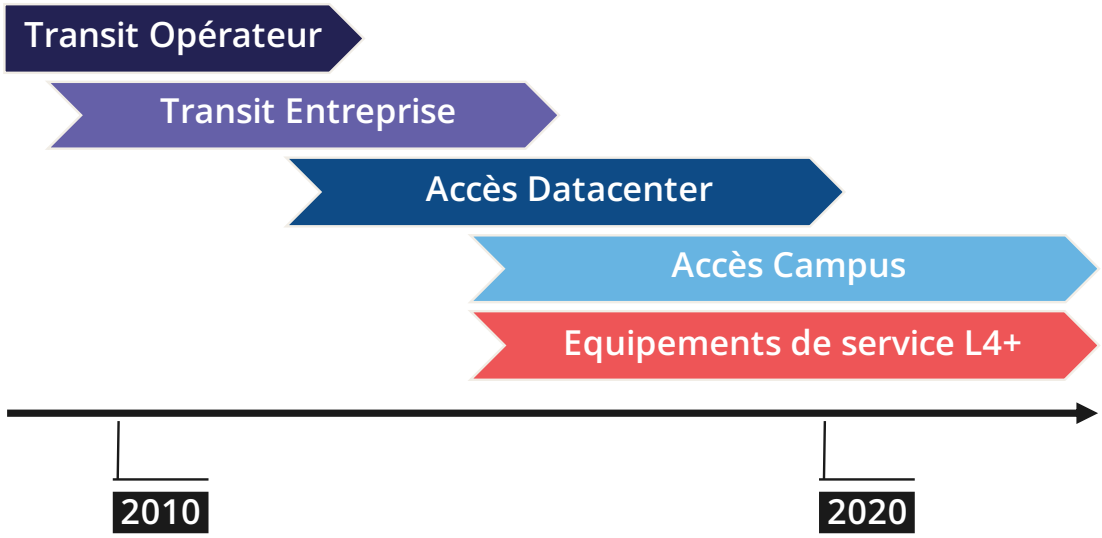
Puis on passe aux équipements de filtrage et d'optimisation de flux, où le fonctionnement par objet permet le passage au dual stack de la plupart des *policies* en agissant uniquement sur les objets pour y répercuter les *subnets*/adresses v6 en corrélation avec v4.

L'accès utilisateur ne peut être activité qu'après le déploiement de la brique sécurité tant sur les équipements réseau que sur les hôtes.

La suite requiert plus de travail, et concerne les services réseau avancés présents en *datacenter* comme les *load balancers*, WAF, sondes, etc.

MATURITÉ

La maturité de la compatibilité IPv6 varie selon les types d'équipements. De façon générale, les équipements de routage de gamme opérateur ne présentent plus de problèmes depuis des années. À l'inverse, les équipements campus rencontrent encore parfois quelques bugs, notamment sur des fonctionnalités de sécurité.



La maturité des solutions semble suivre le graphique ci-dessus, faites attention aux solutions SD-WAN et SDN Campus, cf. le paragraphe SD-WAN de le section « mécanismes de transport » dont les éléments s'appliquent aussi au SDN de campus.

Suivre les *releases notes* et *known bugs* permet de voir quand a lieu la maturation du support IPv6 en se focalisant sur les bugs spécifiques à v6. L'évolution suit généralement la densité de probabilité d'une loi normale et donc une courbe de Gauss.

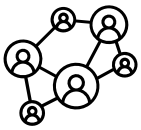
HARDWARE

Pensez à regarder la répartition des mémoires ternaires de vos routeurs, certaines configurations prévoient peu de place pour l'inscription des routes IPv6. Une partie des ASICS du marché stocke différemment les routes IPv6 /48 (et parfois d'autres tailles fréquentes) que les autres tailles de préfixes.

La *full view* IPv6 croît de manière exponentielle, prenez de la largesse sur le choix des équipements devant traiter l'interconnexion publique. Si vous êtes à l'étroit, mais avez toujours besoin des *full view* BGP, il reste possible de dédier des routeurs aux *peerings* v6 et d'autres à v4, si l'étude technicoéconomique est positive.

Comme les addresses sont plus longues, elles consomment 4x plus de mémoire. Pensez aux trables de routing, aux ACL, aux tables d'état stateful. Heureusement des optimisations permettent souvent de ne consommer que 2x plus de place qu'IPv4 en considérant les /64. C'est souvent le cas pour les décisions de routage par exemple.

MAQUETTAGE



Le maquettage des fonctionnalités, des plus basiques comme le routage aux plus évoluées comme les mécanismes de sécurité, peut se faire sur divers environnements. En autonomie ou non. Certains tests comme la validation de la prise en compte de la QoS nécessitent un châssis physique et un générateur de

trafic quand un test OSPFv3 peut, en toute vraisemblance, se limiter à une instance virtuelle. La dépendance aux ASICS étant alors limitée.

On peut imaginer répartir les tests comme suit, en sachant que les tests peuvent être décalés des colonnes de gauches vers celles de droite. Toutefois, cela complexifie leur mise en œuvre jusqu'à augmenter le risque, la dernière colonne étant le pilote sur production.

Environnement min. / Équipement	Maquette virtuelle (environnement constructeur ou type eveNG,...)	Maquette physique indépendante	Pilote sur production
Switch L2	- Validité de la configuration sans test réel - Certains tests L2 en virtuel sont peu probants selon le constructeur.	- Sécu d'accès (ex: RA guard) - MLD snooping - 802.1x - QoS - ACL - Comportement en stack	- Comportement hôtes de prod
AP wifi	NC	- Éléments précédents hors stack - Joignabilité du contrôleur - Routage local hors tunnel - ACL	- Comportement hôtes de prod
Routeur	- Protocoles (OSPFv3, IS-IS, MP-BGP) - FHRP (HSRP, VRRP) - Multicast (PIM, MLD,...) - Relayage DHCP - ACL, route-map - Voisinage de routeurs et FW - DCI - PMTU Discovery	- Éléments précédents - Sécu d'accès (RA guard,...) - QoS - BFD - ARP/ND inspect - Fourniture Dual-Stack aux réseaux d'accès - Performance	-Comportement hôtes de prod -Mise à l'échelle

FW *(en sus des fonctions routeurs)*	- Éléments précédents - Édition d'objets/règles en v6 - NAT64 - Règles de filtrage transit v6 - Tests de non-régression L7	- Éléments précédents - HA FW - Règles de filtrage transit v6 - Contrôleur constructeur - IPsec - Logs v6 + logs NAT64	- Intégration orchestration d'ACL - Intégration des logs v6 + logs NAT64 - Comportement hôtes de prod
Load Balancer (SLB)	- Édition d'objets/règles en v6 - Tests de non-régression L7 - NAT64	- Déchargement TLS - Performances - Logs v6	
IPS/IDS	- Édition d'objets/règles en v6	- Éléments précédents	- Traitement SIEM de prod
Optimisation / Compression de trafic	- Édition d'objets/règles en v6 - Tests de non-régression L7	- Éléments précédents	
Proxy	- Édition d'objets/règles et pac en v6 - Comportement hôtes	- Éléments précédents	
DNS IPAM DHCP	- DNS64 - Enregistrements AAAA - Reverse PTR - Blocs v6 IPAM - Fourniture DHCPv6 avec options	- Éléments précédents - Auto-enregistrement hôte - Fourniture IPv6	

Afin de vous aider, le RIPE a publié le RIPE-772 qui est un document contenant la liste des points de compatibilité à vérifier et à demander lors d'un appel d'offres.

Le NIST US a publié en 2020 la révision de son programme de test USGv6-rev1

ROUTAGE INTERNE

Selon la configuration de votre réseau, l'apport d'IPv6 demandera des modifications plus ou moins profondes en termes de configuration de protocoles de routage.

BGP

Si l'implémentation d'*address family* v6 dans MP-BGP facilite le travail en BGP, il faudra tout de même veiller à analyser les règles de classification de routes de type *access/prefixes lists/sets* afin que les adresses IPv6 soient prises en compte pour appliquer par la suite les route *map/policy* correctement et de façon cohérente avec IPv4. Afin de limiter les incohérences, basez vos règles sur des communautés lorsque c'est possible et marquez ces communautés dès les réseaux capillaires plutôt que de devoir maintenir des listes de préfixes v4 et v6 un peu partout. La rigueur d'une table de correspondance v4/v6 et de l'automatisation sont une autre approche valable, soit de façon distribuée sur les routeurs, soit de façon centralisée sur un *route server* comme FreeRangeRouting, Bird, Quagga (facilitant probablement d'autres aspects de vos ingénieries de routage si vous êtes du genre à souvent modifier le comportement de BGP)

IGP

Deux solutions sont à étudier en ce qui concerne l'IGP. Soit utiliser IS-IS de l'ISO qui est agnostique d'IP, plus souple qu'OSPFv3, mais rarement déployé en entreprise. C'est cet IGP qui domine aujourd'hui sur les grands réseaux opérateurs notamment pour sa convergence et son mécanisme de recalcul partiel.

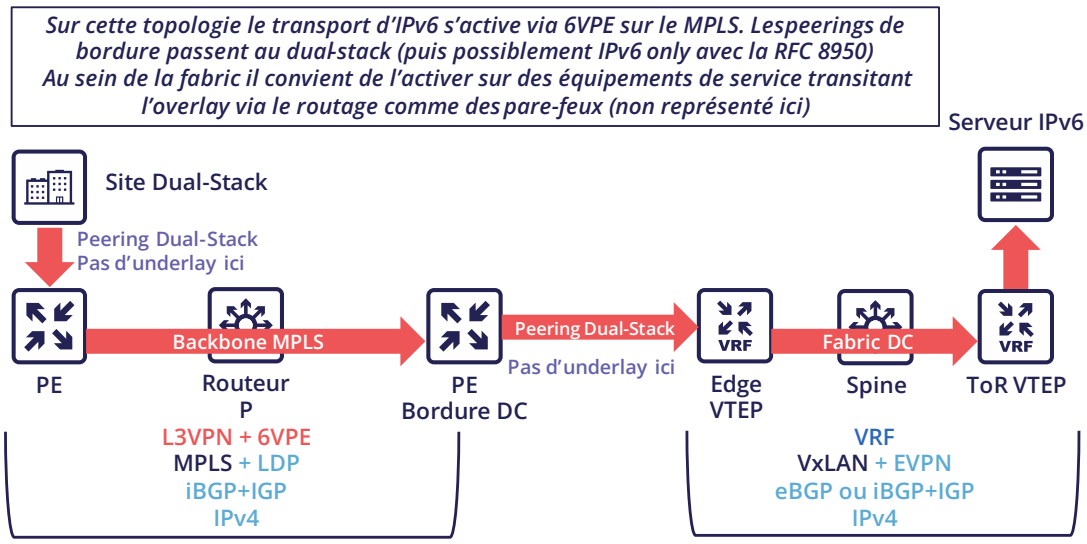
De plus, l'arrivée du Segment Routing basé IPv6 SRv6 requiert IS-IS et ses TLV, même si des LSA OSPF ont été créés pour proposer une équivalence, le marché et les constructeurs semblent se tourner principalement vers IS-IS. (vérifier avec vos fournisseurs)

L'autre solution est de migrer vers OSPFv3 et, une fois celui-ci stabilisé, d'y inclure *l'address-Family* IPv4 afin de retirer OSPFv2 périmètre après périmètre si les équipements sont compatibles avec la fourniture de routes IPv4 dans OSPFv3 RFC 5838.

Garder les 2 versions d'OSPF en parallèle amène les problèmes classiques du *dual-stack* (homogénéité de configuration entre v4 et v6, surplus de configuration, équivalence de monitoring…)

Si vous êtes une structure de grande taille, se former à IS-IS vaut probablement le cout notamment pour vous préparer à SRv6.

N'oubliez pas que seuls les IGP qui portent des réseaux clients sont concernés, généralement les capillaires donc. Il est inutile de modifier l'IGP *underlay* de votre MPLS ou de votre VxLAN EVPN puisque c'est BGP qui s'occupe de v6 dans la couche d'overlay.



FILTRAGE ET TRAÇABILITÉ

Avant de faire transiter des flux, il sera nécessaire d'atteindre le même niveau de sécurité qu'en IPv4. La section sécurité contient de nombreux éléments sur le sujet. Vous trouverez également dans le chapitre « Correspondance v4 / v6 » de la section adressage des conseils facilitant la transcription des règles.

3. Services d'infrastructure

Un grand nombre de services critiques vont de pair avec le bon fonctionnement de l'infrastructure. Certains permettent la connectivité, d'autres ciblent les aspects sécuritaires, etc.

Quel que soit le scénario de déploiement d'IPv6 que vous retiendrez pour votre organisation, l'ordonnancement d'implémentation au sein des services d'infrastructures sera similaire.

SIEM

À chaque nouveau service migré, il faudra pouvoir collecter les logs et les corréler aussi efficacement qu'avec IPv4. L'adaptation de votre SIEM est donc nécessaire tout au long du projet, prévoyez donc à long terme de disposer de ressource sur le sujet. La transcription des règles de *parsing* de logs étant plutôt chronophage. Il serait de bon usage que les principaux éditeurs proposent des mécanismes de conversion clé en main.

Vérifiez bien que les sources de log envoient l'adresse entre crochets suivie du port [IP]:port. Sans crochet difficile de séparer les deux, on peut parier sur le fait que le dernier groupe de chiffres est le port, mais certains applicatifs ne le remontent parfois pas quand le port source est le même que celui du socket serveur et qu'une fonction de simplification est appelée alors qu'elle ne devrait pas l'être (cas rare, mais pas impossible).

Attention au stockage des adresses IPV6, voir la section applicatifs quelques pages plus loin.

DNS/IPAM/DHCP

Cet ensemble est souvent confié à la même solution applicative, hormis pour des zones DNS spécifiques comme celles confiées à un environnement Microsoft Active Directory.

Dans tous les cas, les interfaces de productions de ces services accédés par les clients sont prioritaires à passer en *dual-stack*.

Les services qui interagissent avec l'interface d'administration des équipements n'ont pas besoin d'être fournis immédiatement en IPv6. C'est par exemple le cas des serveurs NTP, RADIUS, TACACS, Syslog, etc. qui peuvent patienter. Il en va autrement si votre scénario cible un déploiement de v6 sur les réseaux d'administration.

VPN, PROXY ET REVERSE PROXY

Ces services ont la particularité d'avoir à la fois des interfaces pointées vers l'intérieur et l'extérieur de la structure. La fourniture d'IPv6 peut être mise en œuvre indifféremment des 2 côtés, les cas d'usage étant différents.

Extérieur

Sans aucun doute le point à mettre en œuvre même si vous ne visez pas du tout un usage interne d'IPv6, la possibilité d'échanger sur internet permettra à vos utilisateurs et clients de vous joindre avec une connectivité native IPv6 à l'heure où les bricolages de partage d'IPv4 deviennent légion. Dans l'autre sens, il permettra à la navigation via proxy de joindre sans problème des sites en IPv6.

Ainsi, votre passerelle VPN et vos reverse proxy devraient être exposés en dual-stack dès que possible, vous évitant de voir vos flux traverser des Carrier-Grade NAT opérateur et autres joyeusetés sans aucune maitrise possible de votre part. On rappellera d'ailleurs que le *reverse proxy* peut également offrir une connectivité publique IPv6 à des serveurs IPv4. Un moyen là encore de reprendre le dessus sur cette translation côté internet.

Intérieur

L'aspect interne va de pair avec le déploiement d'IPv6 sur son LAN. Il nécessitera de s'attarder sur la bonne constitution de ses fichiers proxy PAC, ainsi que de veiller côté VPN à la transposition des règles, notamment celles de split tunneling.

SOUCHES D'OS

Alors que les piles TCP/IP des OS supportent IPv6 depuis une décennie, le support de certaines RFC comme la fourniture des IP DNS via le *router advertisement* (RDDNS) sont plus récents. Par exemple la prise en charge Windows 10 démarre avec la build 1703.

Précédence

La notion de précédence définit la priorité donnée aux différents types d'adresses, et donc notamment la priorisation de v6 sur v4 ou l'inverse.

L'ordre est normalisé, La RFC 6724 de 2012 remplace la 3484 de 2003. Voici les différences :

Adresse	Préfixe	Ancienne Précédence (RFC 3484)	Nouvelle Précédence (RFC 6724)
IPv6 loopback	::1/128	50	50
Native IPv6	::/0	40	40
IPv4	::ffff:0:0/96	10	35
6to4	2002::/16	30	30
Teredo	2001::/32	05	05
ULAs	fc00::/7	40	03
site-local	fec0::/10	40	01
6bone	3ffe::/16	40	01
IPv4compat	::/96	20	01

On peut noter qu'entre les 2 versions, IPv4 est devenu prioritaire sur les mécanismes de transition v6 (6to4, Teredo) et que les adresses site locales sont maintenant dépréciées. L'IPv6 natif conserve la tête.

Attention également aux adresses privées ULA qui deviennent moins prioritaires qu'IPv4, cela peut avoir son importance.

```
PS C:\Users\JC> netsh interface ipv6 show prefixpolicies
Recherche du statut actif...

Précédence  Libellé  Préfixe
----------  -------  ------------------------------
        50        0  ::1/128
        40        1  ::/0
        35        4  ::ffff:0:0/96
        30        2  2002::/16
         5        5  2001::/32
         3       13  fc00::/7
         1       11  fec0::/10
         1       12  3ffe::/16
         1        3  ::/96
```

Figure 05 ***Précédence sous Windows 10***

Résultat de la commande netsh interface ipv6 show prefixpolicies Ce comportement peut être modifié via la clé suivante documentée ici http://support.microsoft.com/kb/929852
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\tcpip6\Parameters

```
FILES
       /etc/gai.conf

VERSIONS
       The gai.conf file is supported by glibc since version 2.5.

EXAMPLE
       The  default  table  according  to  RFC 3484 would be specified with the following configuration
       file:

           label  ::1/128       0
           label  ::/0          1
           label  2002::/16     2
           label ::/96          3
           label ::ffff:0:0/96  4
           precedence  ::1/128       50
           precedence  ::/0          40
           precedence  2002::/16     30
           precedence ::/96          20
           precedence ::ffff:0:0/96  10

SEE ALSO
       getaddrinfo(3), RFC 3484
```

Figure 06 ***Man page Debian 10 (Buster) de GAI.CONF***

Sur de nombreuses distributions Linux la précédence se contrôle dans le fichier GetAddressInfo /etc/gai.conf

Ici le man page correspondant sur une Debian 10 (Buster), aucune trace de la nouvelle RFC de 2012… https://man7.org/linux/man-pages/man5/gai.conf.5.html

Modifier la précédence pour privilégier IPv4 (représenté par ::ffff:0:0/96) peut vous permettre d'empêcher tout dysfonctionnement sur un système en production lors du déploiement d'IPv6. En effet, sauf spécification littérale d'une IPv6 ou d'un enregistrement DNS qui ne correspond qu'à une IPv6, le système continuera alors à faire de l'IPv4 pour les requêtes dont il est client. N'oubliez pas de normaliser une fois un état stable atteint.

Attention, certaines applications comme les navigateurs implémentent leur propre priorisation de v6 sur v4, indépendante de la configuration du stack de l'OS. De plus l'implémentation du mécanisme Happy Eyeballs 2 (RFC 8305) peut varier. (Délai entre les requêtes DNS A et AAAA, temps d'attente du retour, délai de timeout du socket distant avec bascule…) Autre exemple, l'utilitaire CURL qui supporte mieux *Happy Eyeballs* que ses concurrents.

Agents

Les souches d'OS sont généralement livrées en interne avec des agents préconfigurés, plus rarement ceux-ci sont déployés au premier lancement. Dans les 2 cas, ces agents font partie de la base et permettent d'assurer sa conformité, sécurité, etc.

Il s'agit notamment des agents de sauvegarde, antivirus, télémétrie et monitoring, gestion de parc, déploiement de packages/stratégies, etc.

Tant que vous ne planifiez pas un retrait d'IPv4, il est inutile de prioriser le passage de ces services en *dual-stack*, cela peut se faire en même temps que les applications.

L'important est de vérifier que ces agents ne rencontrent pas de problème lié à la simple présence d'un adressage IPv6 routable sur la machine.

N'imaginez donc pas un ouroboros où il faudrait tout faire en même temps sans pour autant savoir par où démarrer.

Une fois les souches exploitables en *dual-stack*, vous pourrez d'une part travailler à passer en IPv6 uniquement une fois l'écosystème prêt si c'est votre scénario, d'autre part vous attaquer aux couches supérieures, les *middlewares* et applications.

SERVICES BUREAUTIQUES

Annuaire

Le service d'annuaire porte les fonctionnalités LDAP et Kerberos, en sus parfois de l'hébergement de certaines zones DNS et d'autres services annexes. Leur omniprésence au sein du SI rend leur migration incontournable. Le produit dominant le marché, Active Directory, fonctionne bien en dual-stack, il est d'ailleurs utilisé par son éditeur en interne en IPv6 depuis plusieurs années.

Note sur les SPN (Kerberos Service Principal Name)

Afin de simplifier la déclaration de chaque serveur et de son service derrière un seul nom, certaines implémentations font par défaut usage d'une requête reverse DNS. Ainsi, quand l'utilisateur demande un ticket de service pour un serveur via un CNAME plutôt que par son *hostname* original, le serveur Kerberos retrouvera l'original via reverse DNS. L'autre solution fastidieuse étant de déclarer tous les SPN possibles de chaque serveur.

Ce comportement (résolution canonique), bien que déconseillé par la RFC 4120 est employé au sein d'Active Directory pour sa simplicité. Il faut en conséquence s'assurer que le serveur Kerberos (KDC) ne va pas faire de requête reverse DNS avec une IP récupérée via un DNS64, ou à défaut, que le serveur DNS sait mentir et formuler une réponse adaptée à ces requêtes particulières.

Dernier point, il existe parfois encore des SPN basés sur l'IP plutôt que le nom d'hôte (généralement pour d'anciennes applications avec, vous l'aurez deviné, une configuration en dur ou simplement l'appel à l'IP comme on le faisait il y a fort longtemps). Cas rare ceci dit puisque Windows côté client ne supportait plus cette fonction entre Vista et Win 10 1507, forçant à rétrograder en NTLM pour lesdits services. Ce cas spécifique nécessitera l'usage de 2 SPN par machine et service (v4 et v6)

Hébergement de fichiers et packages

Qu'ils soient visibles des utilisateurs ou non, les serveurs fournissant des fichiers génèrent un fort volume de trafic. Si votre projet cible des clients en v6-*only* avec du NAT64, il serait bon de migrer ces serveurs en *dual-stack* (ou de leur dédier une plateforme de translation) ce qui soulagera considérablement la plateforme de translation centralisée.

Cela inclut notamment les serveurs SMB, NFS, WSUS, SCCM, dépôts de packages, dépôts de signatures d'EDR, GED, SharePoint, etc.

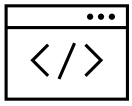
Communication

Le système de messagerie électronique peut se contenter longtemps de NAT64, mais la charge importante de trafic générée par ce système pousse à migrer au moins la couche d'accès offerte aux clients en IPv6. Pour la partie exposée en WAN, le MTA, rien d'urgent, on n'est pas près de voir des serveurs SMTP ne proposant qu'IPv6. Une migration demandera la vérification de la compatibilité et de l'efficacité de vos solutions de contrôle de contenu et d'antispam.

De la même façon, en téléphonie c'est la partie exposée aux clients qui a tout intérêt à migrer rapidement, et bien plus urgemment que la messagerie afin d'apporter la compatibilité IPv6 aux échanges, qu'ils soient P2P entre les clients ou entre client et infrastructure centralisée. Urgence renforcée par les mauvaises surprises de NAT64 avec le SIP, sauf à avoir confiance dans les ALG. Mais les flux RTP étant de plus en plus souvent chiffrés, il ne faut pas trop compter sur les ALG.

Sachez que les offres SaaS du marché sont de plus en plus nombreuses à supporter IPv6, avec quelques rares exceptions telles que l'interfaçage entre un SBC *on-premise* et son homologue SaaS, ce qui n'est pas très dérangeant.

APPLICATIFS



Plutôt que de spécifiquement lancer une fastidieuse campagne de qualification dédiée à IPv6, il est préférable d'user des opportunités offertes par les mises à niveau majeures de ces applications pour les qualifier, cette fois en IPv6, et uniquement en IPv6. Les retours de grands éditeurs montrent que qualifier une application en IPv6 suffit et qu'il est inutile de tout redérouler en IPv4, les méthodes et appels d'instructions récents étant rétrocompatibles sans travail additionnel. Forcément cela n'est pas valable pour une application exploitant un langage ancien et/ou avec des adresses codées en dur.

Voici une liste de questions à se poser sur chaque application :

- **Y'a-t-il ailleurs des utilisateurs de la solution en v6 ?** (Demander à l'éditeur, intégrateur, testeur, etc.) ;
- **Le langage utilisé est-il compatible IPv6 ?** et de façon stable et fiable (de nombreux bugs d'implémentation ont été corrigés dans différents langages jusqu'en 2015)
- **L'appel aux ouvertures de sockets dans le code est-elle agnostique de la version du protocole IP ?** Inet6Address et InetAddress dans Java par exemple ;
- **Le trafic IPv4 et v6 transite-t-il par le même socket ?** Exemple précédent vs usage de IPv4-mapped address (toujours en Java donc) ;

- **Une application gère-t-elle IPv6 côté client ? Côté frontal serveur ?** Côté *back-end* serveur dans le cas d'une application n-tiers (même si ce dernier point est moins problématique) ;
- **Une application effectue-t-elle des appels via adresse littérale plutôt que par résolution DNS ?** Champ de configuration IPv4 uniquement par exemple ;
- **Une application utilise-t-elle un protocole encapsulant l'adresse littérale ?** Comme le SIP avec la téléphonie, ou le FTP actif ;
- **Une application initie-t-elle des connexions à destination de terminaux clients ?** Exemple du FTP actif avec ses 2 sessions contrôle et data concurrentes, l'une dans chaque sens. Ou encore de la prise en main à distance, mais aussi le SIP, le DICOM, etc. ;
- **Y'a-t-il un traitement de l'adresse IP au sein de votre application ?** Par exemple l'identification du client par son IP plutôt que par son nom d'utilisateur ;
- **La RFC 8305 « *Happy Eyeballs* v2 » est-elle implémentée afin de permettre une bascule rapide entre les 2 protocoles ?** (La fonction d'appel utilisée et la configuration par défaut du langage sont à regarder en détail, il est très facile de ne pas l'implémenter correctement en java par exemple) ;
- Enfin, si l'application n'est pas compatible IPv6**, ses logs conservent-ils bien le port en plus de l'IP** (pour garantir la traçabilité NAT64) **?** cf. RFC 7768 de 2016, elle-même inspirée de la RFC 6302 de 2011 qui ne recommandait cela initialement pour les serveurs frontaux sur internet.

Divers outils d'audit existent, certains sont intégrés aux environnements de développement, d'autre indépendants comme Microsoft checkv4, PortToIPv6, IPv6 code checker, IPv6 care, etc. Ces outils permettent soit d'auditer le code, soit de détecter les appels de socket lorsque le code tourne et d'identifier la méthode utilisée.

Les applications mobiles publiées sur le Google Play Store et sur l'Apple App Store se doivent d'utiliser des méthodes et fonctions réseaux compatibles IPv6 depuis 2016, elles ont été un bon exemple d'adaptation rapide de code.

Sans attendre, intégrez IPv6 à vos cahiers des charges et dossiers d'architectures pour les nouvelles applications. Projetez également une date à partir de laquelle les mises à niveau d'une application existante devraient prévoir l'implémentation d'IPv6.

Figure 07 — ***Exemple de traitement d'une Web App***

Comment traiter un service fourni via le navigateur ?

Dans les architectures n-tiers, la priorité est de déployer IPv6 sur les frontaux, qui sont joints par les clients. Le *back-end* des applications pourra rester en v4 beaucoup plus longtemps.

L'idéal est de profiter de l'obsolescence des applications et du renouvèlement pour implémenter IPv6.

Sachez à tout hasard que l'utilitaire Curl supporte IPv6 depuis plus de 20 ans…

Cas de la manipulation d'IP

L'adresse IP est un élément important qu'on retrouve dans les référentiels, il peut s'agir entre autres des outils suivants :

- Inventaire CMDB / IPAM ;
- Configuration d'infrastructure Orchestration / Déploiement / Sauvegarde de conf ;
- Exploitation supervision / métrologie / suivit d'incident ;
- Scripts de collecte d'information ;
- Corrélation de logs (SIEM) / Audit ;
- Gestion d'accès Ouvertures de flux / Identité.

L'utilisation d'IPv6 implique de réviser le stockage et le traitement des adresses pour diverses raisons :

- L'adresse IPv6 vient parfois en sus de l'IPv4 (dual-stack) ;
- Elle est plus longue ;
- Une interface peut porter plusieurs IPv6 (link local, routable temporaire, routable stable…).

Une option de simplification peut être de tout traiter comme des IPv6, y compris les IPv4 via le préfixe de représentation ::ffff:0:0/96. Ce mode facilite la cohésion et la simplification du code applicatif.

RDV toutefois en annexe dans la section Exemples de problèmes d'implémentation pour retrouver un désagrément possible avec cette méthode.

Dans tous les cas, il faudra stocker les adresses sous leur forme canonique (raccourcie) afin de réduire leur taille. Le code effectuant la canonisation devra respecter scrupuleusement la RFC 5952 afin de s'assurer que l'on compare toujours la même chaine de caractère. Notez que les adresses doivent également être stockées avec des caractères minuscules (section RFC 4.3). Par exemple, ab01::ffff et non AB01::FFFF. Le non-respect de cette dernière préconisation peut même poser problème dans des protocoles transportant l'IP en *payload* comme le SIP.

4

Plan d'adressage

- PUBLIC OU PRIVÉ ? 56
- PETITE STRUCTURE 56
  - ULA 56
  - Préfixe Provider-Independent (PI) 57
  - Inconvénients de NPTv6 57
- GRANDE STRUCTURE 58
  - Gestion des accès directs à internet 58
- GROUPEMENT LOGIQUE 60
- ÉLÉMENTS CONSTITUANTS 60
- TAILLE DES PRÉFIXES 62
  - Standard 62
  - Interconnexion 62
- ADRESSES DES SERVICES COURANTS 63
- ÉVOLUTIVITÉ TEMPORELLE 64
- USAGE DU N° D'HÔTE 0 65
- SEGMENTATION PAR INTERFACE 65
- CORRESPONDANCE V4 / V6 66
  - Numéro de réseau / préfixe 66
  - Numéro d'hôte / ID d'interface 67
- POUR LES RÉSEAUX NATIFS V6 69
- ANNONCES PUBLIQUES 69

Plan d'adressage

IV. Plan d'adressage

La constitution de cet important référentiel que représente le plan d'adressage est un processus long à effectuer par itérations.

Le plan d'adressage est à construire en fonction des spécificités de l'organisation. Au-delà des éléments mentionnés dans cette section, il est nécessaire d'échanger avec de nombreux interlocuteurs en projetant le résultat de vos différentes propositions de plans. Dans tous les cas, pensez à long terme et gardez de la place libre tout en haut du bloc.

Avant d'entrer dans les possibilités de construction de plan, commençons par aborder le choix du préfixe à exploiter au sein de l'organisation.

PUBLIC OU PRIVÉ ?

En IPv4, l'adressage public est rare et se limite donc naturellement à l'exposition sur internet. Dans le réseau interne, les plages RFC 1918 dont la 10.0.0.0/8 dominent.

Avec IPv6 la question se pose de choisir entre les 2 options.

Selon votre taille et activité, l'idéal devient rapidement de disposer de son propre préfixe affecté par le gestionnaire régional (RIR). De même une filiale de grande taille ou disposant d'une grande indépendance en termes d'IT a tout intérêt à demander son propre bloc auprès du RIR.

Vous pouvez aussi demander un PI à un LIR sponsor, ça n'est pas cher du tout.

La politique d'attribution est au minimum /32. Il est possible d'avoir des blocs plus larges si la taille de la structure le justifie.

PETITE STRUCTURE

Pour une petite structure, il est possible d'exploiter le préfixe fourni par l'opérateur en effectuant de la délégation de préfixe. Toutefois, dépendre d'une structure tierce pour l'adressage de ses ressources n'est pas l'idéal et on atteint vite les limites de ce mode, par exemple pour l'adressage fixe de serveurs internes. Il reste la possibilité de faire de la délégation de préfixes avec DHCPv6-PD pour les téméraires adeptes du changement soudain. Dans un monde idéal, l'ensemble des outils pourraient s'adapter à une renumérotation de préfixe, mais en pratique il faudra probablement encore attendre longtemps avant de pouvoir rendre l'ensemble des configurations dynamiques.

ULA

L'adressage privé IPv6 apparait alors comme une solution, cet équivalent de la RFC 1918 IPv4 se nomme ULA pour *Unique Local Address* et correspond au préfixe FC00::/7.

Afin de limiter le risque de conflit avec une autre structure avec laquelle vous devriez échanger hypothétiquement un jour de façon privée, par exemple au travers d'un tunnel IPsec, il est recommandé de

choisir aléatoirement un préfixe dans ce /7 plutôt que de démarrer par le bas (la RFC4193 impose même cette génération aléatoire).

Prenez par exemple un /48 de façon arbitraire dans FC00::/7 et bâtissez votre adressage dessus. Pour une toute petite structure, un /56 suffit. Attention, la taille ne doit pas être trop large pour une raison que nous allons voir ci-après. La RFC4193 propose un algorithme de génération pseudo aléatoire pour obtenir un ID de 40 bits, ce qui donne un préfixe de 48.

Cependant ULA est moins prioritaire qu'IPv4 dans les règles de précédence actuelles. IPv4 sera donc privilégié dans un environnement dual-stack (sauf pour du trafic entre 2 ULA). Deux solutions s'offrent alors à vous :

- Modifier le comportement de tous les hôtes afin qu'ils privilégient ULA sur IPv4 ;
- Demander un préfixe indépendant PI et ne l'utiliser qu'en interne, c'est cette dernière méthode qui est recommandée.

Préfixe Provider-Independent (PI)

Solution préconisée, demandez un préfixe PI en /48 auprès d'un LIR. Pas besoin de l'annoncer sur internet, etc. Mais il vous permettra d'avoir un LAN avec un adressage unique et de ne pas rencontrer les problématiques de précédence ULA vs IPv4.

Les paquets utilisant cet adressage sur le LAN (PI non annoncé sur internet) doivent subir un NAT afin de sortir sur internet. Si en IPv4 on exploite du NAT44 + PAT avec table de session *stateful*, ici on va exploiter *Network Prefix Translation* v6 (NPTv6 | RFC 6296).

La Translation de préfixe change les premiers bits de l'adresse afin de faire correspondre un préfixe IPv6 à un autre de même taille. Aucun autre changement n'a lieu, tout est *stateless*.

Il suffira de mapper son préfixe privé /56 (ou autre taille) à celui public fournit par l'opérateur pour échanger sur internet tout en maitrisant son adressage interne. Il est possible de mapper un /56 interne à un morceau de /48 public routé, mais pas l'inverse évidement (d'où l'importance de ne pas prendre une plage trop grande sur son LAN)

Grâce à NPTv6, l'entreprise peut changer de fournisseur d'accès sans rien toucher en interne, de plus ça ne casse pas la découverte de taille maximale transmissible PMTU-D.

Inconvénients de NPTv6

Reste quelques ombres au tableau, premièrement les protocoles encapsulant l'adresse dans le *payload* comme SIP, H323,etc. nécessiteront toujours l'emploi d'une *Application Layer Gateway* (ALG) correspondante sur l'équipement effectuant la translation. Comme en NAT44 les ALG peuvent être un vecteur d'attaque, voir notamment les récentes méthodes de *slipstreaming* qui ont forcé les éditeurs de navigateurs à bloquer certains ports de destination.

Deuxièmement, vous aurez besoin de synchroniser vos enregistrements DNS entre la zone interne (PI ou ULA selon votre choix) et sa version externe pour les services exposés. Vous permettant de ne pas publier par erreur sur internet un enregistrement AAAA avec une adresse non joignable d'une part, et que vous n'utilisiez pas l'adresse globale routable en interne d'autre part, puisque cela impliquerait de de tromboner au travers de la plateforme NPTv6. Par exemple un client LAN devrait joindre directement un serveur en DMZ via son adresse interne (ULA ou PI non annoncée).

Oh, et n'oubliez pas de créer des reverse PTR pour les deux types d'adresses puisque c'est nécessaire pour certains services comme les MX SMTP puisque cela fait partie des contrôles antispam. Il existe heureusement des mécanismes pour générer automatiquement les PTR.

GRANDE STRUCTURE

Commencez par obtenir un préfixe public PI (Provider Independent), ou plusieurs dans le cas de filiales ou de présence géographique multicontinentale.

Certaines particularités sont à prendre en compte avant même la construction de votre plan.

Vos annonces publiques BGP ne pourront pas, par convention, être plus fines que des /48 (situation analogue aux /24 IPv4 ; Voir RFC 7454 section 6.1.3). Inutiles pour autant de dédier un préfixe annonçable qui correspondrait uniquement aux serveurs exposés, nous allons voir pourquoi.

IPv4 et l'omniprésence de NAT44+PAT ont apporté des pratiques qu'il est inutile de reproduire en IPv6, notamment le faux sentiment de sécurité offert par NAT44 en entrée. L'aspect diode est présent en raison de la nécessité d'effectuer un suivi de session, donc *stateful*. Et s'il est normal de ne pas avoir d'autotransfert de port lié à l'UPnP comme on en trouve sur un équipement grand public, il est plus difficile de se prémunir des récentes attaques par *slipstreaming* faisant appel aux ALG comme évoqué plus haut.

Un équivalent de NAT *stateful* + PAT existait en IPv6, mais son usage est loin d'être recommandé. En fait, le NAT-PT (*NAT Protocol Translator* RFC 2766, à ne pas confondre avec NPTv6) est tout simplement inexploitable et archivé, voir la RFC 4966 qui énonce les raisons de la mise au placard de ce mécanisme.

Ainsi vous trouverez parfois des préconisations de sécurité qui sont d'avoir un réseau interne en IPv6 privé, d'utiliser du NAT en sortie afin de rendre son plan d'adressage invisible à l'extérieur, etc.

Ces recommandations sont les réminiscences des habitudes en IPv4, de même faire de l'adressage privé en interne avec de la translation de préfixe NPTv6 pour sortir ne présente aucun intérêt sécuritaire pour une grande entreprise, et ne masque aucunement le détail du plan interne puisqu'on bascule simplement les premiers bits de l'adresse. Il faut bien se rappeler que le NAT ne protège pas, seuls un pare-feu avec les bonnes ACL et éventuellement de l'inspection sont efficaces.

L'ensemble du SI devrait être adressé avec le préfixe global public attribué à la société.

Gestion des accès directs à internet

La translation de préfixe NPTv6 peut toutefois servir pour des situations courantes. Prenons une entreprise qui souhaite utiliser du *local breakout* (LBO) sur ses campus afin d'atteindre des ressources internet (une solution SaaS par exemple) sans repasser par son datacenter. Le trafic va alors devoir passer d'une adresse qui appartient à l'entreprise à une qui est fournie par l'opérateur internet local du campus.

Notez que cet usage fréquent est une raison de disposer de préfixes de sites basés sur une affectation géographique. Cela permet de n'avoir qu'une seule règle de NPTv6. Si votre adressage de site est morcelé il faudra en effet faire correspondre chaque /64 local à un /64 appartenant au préfixe fournit par l'opérateur local (typiquement un /48). C'est d'autant plus de règles et donc de travail.

Déviation plus rare de ce cas d'usage de *local breakout*, si le campus est très grand et que l'opérateur local le permet, il est envisageable que le site annonce son propre /48 (ou plus) via BGP directement sur internet.

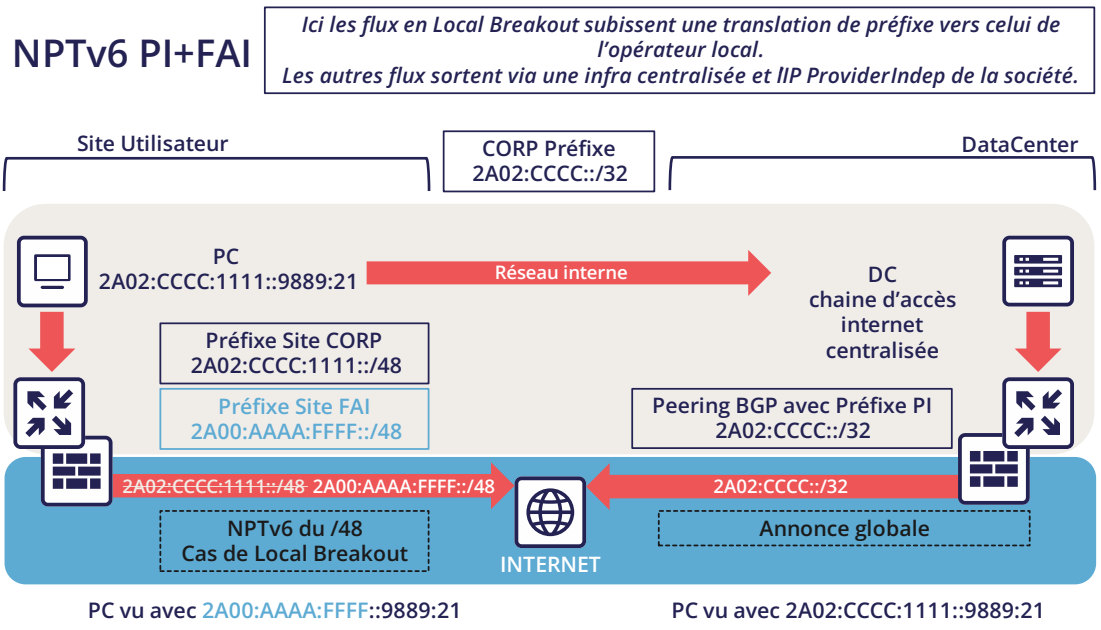
Dans ce cas les équipements du site exploitent des adresses d'un préfixe que nous nommerons « Site » /48, ce préfixe n'est pas annoncé, mais un préfixe plus large « Global » /32 qui l'englobe est annoncé par le datacenter. Enfin, le site annonce localement et directement sur internet un préfixe « LBO » /48 qui appartient aussi au global /32. En réalité ce scénario provoquerait une hausse considérable de la taille de la fullview BGP mais reste utilisable si le fournisseur agrège les routes en bordure avec un plan le permettant.

La règle locale de NPTv6 traduit le préfixe Site/48 en LBO/48 lors de la sortie internet locale. Le fonctionnement des décisions de routage de BGP privilégiant les routes fines permettra à l'ensemble de fonctionner sans conflit, avec donc cette fois des IP qui appartiennent toutes à l'entreprise. Si l'on dispose de plusieurs sites dans cette situation chez le même opérateur, il sera alors intelligent de lui demander d'agréger les annonces.

Finalement, certains trafics sortiront directement du site via le préfixe LBO, d'autres qui doivent subir des traitements plus poussés en DC sortiront via le préfixe Site (selon la configuration de proxisation des postes).

La montée en puissance des solutions dites « SASE » (Secure Access Service Edge) peuvent permettre de se passer totalement de traitement dans le DC, dans ce cas il n'est plus nécessaire d'user de deux préfixes avec du NPTv6.

Le gain en latence offert par LBO peut être important, on ne repasse plus par le DC et ses chaines de liaisons. Reste qu'il faut assurer le même niveau de sécurité en termes de filtrage, d'analyse antivirale, etc. La stratégie variera donc entre autoriser une partie des flux (destinataires d'un niveau de confiance suffisant) et tout le trafic internet selon l'équivalence de protection qu'on peut atteindre, qu'elle soit fournie localement via des VNF ou via une solution Cloud.



Pour des infrastructures qui doivent être totalement être isolées d'internet ainsi que de tout partenaire (comme un réseau SCADA), il est possible d'utiliser des adresses ULA. Ceci n'empêche aucunement les attaques par rebond depuis un autre système interne, les pare-feux suffisent de toute façon à bloquer du trafic en bordure de ces réseaux. L'apport de l'ULA est donc quasi nul et reste un choix subjectif. On rappelle là encore qu'ULA est moins prioritaire qu'IPv4 dans les règles de précédence actuelles. IPv4 sera donc privilégié dans un environnement *dual-stack*.

GROUPEMENT LOGIQUE

Historiquement, IPv4 a habitué à assigner des plages par site afin de limiter le nombre de routes via de la *route summarization*, plus récemment certains projets ont pu effectuer l'approche inverse, à savoir assigner le site géographique à partir d'un bloc dédié à un usage précis comme lors d'un projet de déploiement Wi-Fi dans des agences ou IoT.

Ce dernier cas est avantageux pour le filtrage, car on est centré usage plutôt que géographique.

Le choix est d'autant plus important que, contrairement à IPv4, on ne peut pas utiliser de masque, seulement le préfixe pour filtrer.

En IPv4, il est possible, même si rarement exploité, d'utiliser par exemple le *wildcard* 0.0.240.0 pour sélectionner les n hôtes identiques de différents *subnet*. En v6 cela disparait.

Si les équipements supportent un grand nombre de routes, les règles manuelles appliquées à des routes deviendraient complexes à mettre en œuvre avec un plan centré usage, et on sait d'ores et déjà que malgré l'automatisation et l'arrivée du SDN sur différents périmètres, BGP restera le moyen d'interconnecter les « boites noires » entre elles. Il serait néanmoins toujours possible d'utiliser des scripts et un *route server* *bird* ou *FFRouting* pour effectuer automatiquement les classifications et appliquer les stratégies ou simplement de bien utiliser les communautés sur les annonces.

Les deux options géo ou type centrées ont donc des avantages et inconvénients, qui peuvent être compensés par l'automatisation (consolidation des règles de filtrage vs consolidation des routes et des sites). Comme dit précédemment dans l'explication sur NPTv6, il est plus facile de se baser sur l'emplacement avec la notion de sites.

ÉLÉMENTS CONSTITUANTS

Le découpage peut privilégier les multiples de 4 bits (des caractères hexadécimaux), /32, /48, /52, etc. afin de faciliter la lecture, tendance qui correspond à notre habitude de découper les IPv4 par octet et qui mène au gaspillage dans le cas précis de v4.

Le groupement de 4 caractères se nomme un hextet, par exemple :A9B4:

Chaque caractère hexa peut se nommer nibble.

Dès lors, on rappellera que si v6 offre un grand nombre d'adresses, cela ne doit pas être une invitation au gaspillage, on évitera par exemple de faire du leet speak comme « c01d:c01a:cafe» / « cold cola cafe » avec les numéros de préfixes, Network ID.

À ces blocs de caractères, on peut penser tout de suite à associer :

- Entité juridique / Secteur d'activité ;
- ID de site géographique ;
- Type de réseau ;
- N° de VLAN ou de VNI ;
- Opérateur ;

- Modèle d'équipement.

Les éléments numériques peuvent être conservés tel quel, prenant plus de place, ou encodés en hexadécimal supprimant la lisibilité humaine.

Par exemple, pour stocker le n° de VLAN, de 0 à 4094 (12 bits) on a au choix

- 4 0 9 6 soit 4 caractères donc 16 bits ;
- F F E soit 3 caractères pour former 4094 en hexadécimal, avec un caractère hexa libre restant dans l'hextet x F F E.

Dans le cadre où l'on créer un nouveau référentiel, comme les types de réseau, autant les écrire directement en hexadécimal si le découpage le permet.

Si l'on reprend la liste d'éléments constituants, certains ont un cycle de vie inadapté à l'intégration dans un plan d'adressage. Par exemple, l'opérateur peut changer entre temps, la marque et le modèle d'un équipement de niveau 3 aussi. Le référentiel deviendrait donc caduc (on sait par expérience que le maintien et le changement ne seraient pas répercutés, car « ça marche bien comme ça »). Nous verrons plus loin une exception pour les interconnexions.

En datacenter la même chose se produira avec les VLAN, l'usage de technologies E-VPN + VxLAN avec un numéro de VNI sur 24 bits relèguera le VLAN au second plan, il en va de même avec les technologies propriétaires de segmentation qui intègrent des notions de tenant client, de pool de ressources, etc.

On en retient que le plan ne doit intégrer que des éléments probants et figés dans le temps, ce qui nous donne :

- La division / entité à haut niveau afin de permettre le découpage de la structure (comme on le fait dans un annuaire Active Directory) ;
- La localisation soit par arborescence continentale / plaque / site, soit par code site à plat ;
- Le type de réseau, avec des sous-catégories permettant de faciliter la gestion du filtrage et de déléguer une partie du plan d'adressage.

TAILLE DES PRÉFIXES

Standard

D'emblée, le /64 apparait comme la norme immuable pour un réseau (RFC 4291), notamment pour que les mécanismes d'autoconfiguration comme SLAAC fonctionnent.

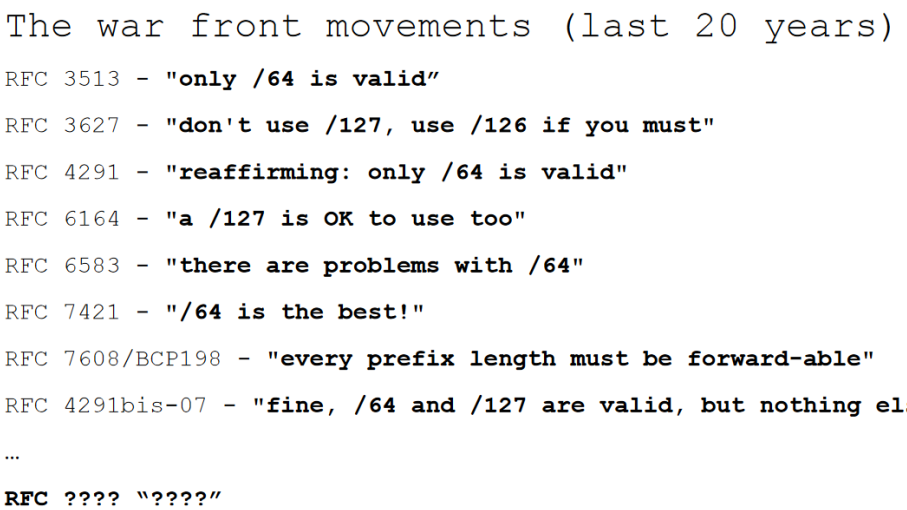


Figure 08 ***La norme n'en est pas toujours une***

Le standard est bien /64, utiliser autre chose pour le raccordement des hôtes pourra parfois vous amener à rencontrer des comportements imprévus ou des incompatibilités.

Concernant les préfixes de sites, les recommandations ont également évolué, la RFC 6177 ajuste le préfixe au besoin réel alors qu'on obligeait auparavant à user de /48.

Chez les opérateurs, la norme est donc d'assigner un /56 ou /60 aux clients domestiques, et /48 aux professionnels. Les réseaux de terminaux se font toujours en /64, sauf pour les intercos.

Interconnexion

Les opérateurs semblent préconiser des interconnexions en /125. Afin de couper entre 2 caractères hexadécimaux, il serait bon de fournir des /124 dans le plan et d'utiliser le 125ème pour des bascules lors de changement d'équipement ou de fournisseur.

Cette réservation ne vous empêche pas de paramétrer les interfaces point à point en /127

Ces réservations pour les intercos et les *loopbacks* peuvent hériter de l'adressage du site, ou au contraire d'un préfixe /64 dédié à être découpé en /124 et plus pour des intercos.

Dans ce dernier cas, vous devrez annoncer de nombreuses routes fines sur votre réseau.

Faire les interconnexions avec des *Link-local* fonctionne, mais présente des inconvénients détaillés dans la RFC 7404 (pas de retour ICMP de l'interface puisque non routable mais seulement de la *loopback*, adresse qui change en cas de remplacement hardware car auto fondée sur la MAC EUI-64, etc.). L'un des gros avantages en revanche est l'allègement des tables de routage ainsi que la réduction de la surface d'attaque. L'aspect tracé du chemin avec des *Link-local* peut être retrouvé avec la RFC 5837. Le choix sera donc généralement différent entre un réseau corporatif VS un large ISP ou un point d'échange GIX.

Vous pouvez tout à fait construire votre préfixe d'interconnexions /124 avec le n° d'AS BGP du tiers, l'ID du routeur, etc. Bref, tout ce qui vous aidera au quotidien.

Attention aux IPAM, ils refusent souvent qu'on inscrive autre chose que des /64, pourtant il n'est pas anormal d'avoir des intercos avec des préfixes longs.

En dehors des interconnexions le /64 est la norme actuelle et il serait dommage de s'aventurer à exploiter autre chose.

Focus

Des drafts de RFC visent à permettre à SLAAC de fournir autre chose que du /64, voir draft-mishra-v6ops-variable-slaac-problem-stmt et draft-mishra-6man-variable-slaac. Ces drafts cherchent à répondre à la problématique de subdivision d'un unique /64 fournis par exemple par un opérateur mobile via *3GPP link*. L'objectif est de pouvoir créer des réseaux différents sur des micro-infrastructures mobiles, typiquement un routeur avec multiples réseaux clients ou encore un véhicule connecté dont les différents réseaux internes exploitent tantôt Ethernet, tantôt des BUS et ne peuvent être bridgés. Il est même nécessaire d'avoir des réseaux d'échange direct avec les véhicules voisins (V2V). L'avenir dira si ces drafts aboutissent ou s'ils deviennent caducs dans le cas où les opérateurs se mettent tous à supporter DHCP-PD sur mobile avec des /56 via 3GPP comme c'est souvent le cas pour les connexions domestiques.

ADRESSES DES SERVICES COURANTS

Pour des raisons de facilité de saisie, il est intéressant d'attribuer des adresses courtes aux services pour lesquels l'IP doit souvent être saisie manuellement, et bien évidemment en premier lieu les serveurs DNS, mais aussi les interfaces des routeurs.

Ainsi, le lot d'adresses de tout début d'un organisme, pre:fixe:0000:0000:... devrait être dédié à des attributions fines pour faciliter le travail des exploitants/administrateurs en leur permettant notamment de les retenir de tête.

À chaque étage de votre plan, plaque régionale, site, etc., il sera bon de réserver le 0 et le 1 pour des services utilisant des adresses raccourcies. Là encore pour faciliter les tâches quotidiennes.

N'oubliez pas cependant de ne pas placer toutes les instances d'un même service dans le même préfixe. Avoir par exemple l'ensemble de ces DNS ou relais SMTP dans le même préfixe et donc dépendant de la même route n'est pas une bonne pratique. En cas d'incident de routage de ce préfixe, vous pouvez avoir autant d'instances du service physiques et/ou logiques que possible, ça sera tout de même le blackout.

ÉVOLUTIVITÉ TEMPORELLE

Afin de répondre aux migrations à différents niveaux, des bits de migration peuvent être mis en place.

Un bit de migration réseau peut faciliter les changements d'équipements, de liaisons WAN, etc. Ce bit devrait être le 64e afin d'être pris en compte dans des règles de filtrage en /63. Il permettrait des transitions de *subnet*, VLAN, équipements de façon progressive sans autre modification puisque les ACL en /63 engloberaient les 2 /64 exploitables.

Par exemple, un campus change de cœur et migre en même temps sur un MAN. Les nouveaux réseaux sont mis avec le bit de transition et routés en parallèle des anciens. Des tests métiers peuvent avoir lieu sur la nouvelle infrastructure avant la migration grâce au filtrage large incluant ce bit. Cela évite les migrations big bang et limite la découverte d'incidents postmigration.

Au changement suivant le bit est basculé. Aucun état entre le 0 et 1 n'est préféré.

Toute opération de changement d'équipement, d'opérateur, de déménagement, etc. s'en trouve donc grandement facilitée.

Il conviendra tout de même de prévenir l'annonce d'un réseau jumeau qui profiterait de règles de filtrage globales de façon involontaire. La surveillance de la provenance de routes appartenant à la même paire de migrations est une nécessité.

D'une façon plus générale, gardez de la place pour vos futurs référentiels. Cela pour permettre de vous adapter à de nouvelles architectures sans reprendre des blocs tout en haut.

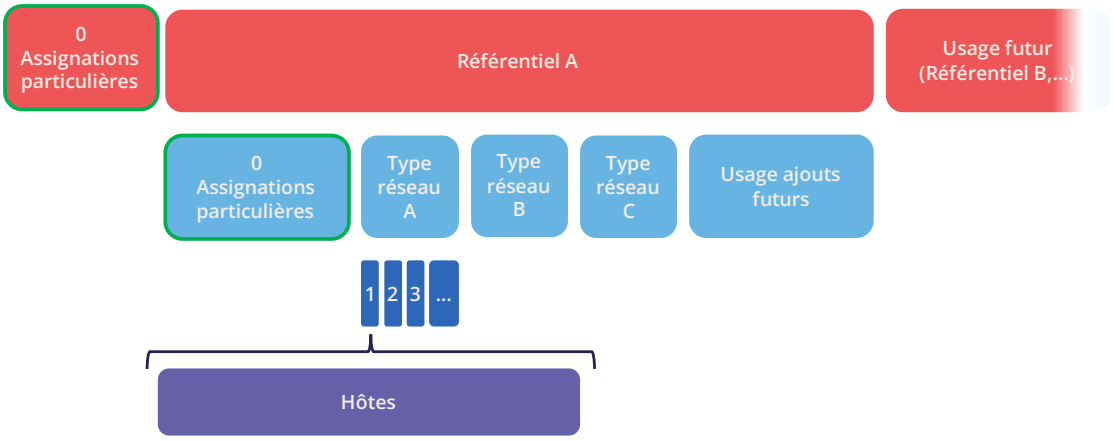


Figure 09 ***Exemple de hiérarchisation de référentiel***

USAGE DU N° D'HÔTE 0

En IPv6, il n'existe plus d'adresse de numéro de réseau ni d'adresse de *broadcast*, toutes les adresses possibles peuvent être affectées à un hôte.

Cependant, de mauvais regex sont parfois présents dans des champs de configuration d'applications. On trouve facilement des bugs de systèmes qui ne supportent pas de se voir configurer une adresse se terminant par ::0, par exemple ABCD:1234::/64. Parfois sur leur interface ou sur celle d'un élément tiers comme l'adresse du serveur DNS ou NTP.

Aussi, nous vous recommandons d'éviter les adresses avec des numéros d'hôtes en 0 à minima pour les serveurs susceptibles d'être configurés de façon littérale dans des équipements tels que des imprimantes, des caméras, et tout équipement embarqué.

L'usage de DNS limite le risque, sauf pour DNS lui-même. Gardez un 1 final aux adresses de vos serveurs DNS, ça pourra vous éviter ce type d'écueil même si cela tend à disparaitre.

L'usage de cette première adresse disponible pose également la question du risque de confusion entre adresse et préfixe. En IPv4, l'adresse de réseau ne pourra jamais servir à un hôte (sauf cas particulier des intercos en /31 RFC 3021) alors qu'en IPv6 il est possible d'avoir la même adresse indiquant un préfixe et un hôte, la taille de préfixe étant alors le seul élément délimitant. Par exemple un hôte ABCD:1234::/128 appartenant au réseau ABCD:1234::/64.

Pour cette deuxième raison de facilitation de lecture humaine, il est préférable de ne pas du tout utiliser l'adresse d'hôte 0.

SEGMENTATION PAR INTERFACE

Certaines contraintes techniques ou sécuritaires peuvent conduire à instancier plusieurs interfaces réseau sur des serveurs. Par exemple, certaines stratégies de sécurités prévoient des interfaces dédiées de management. On segmentera parfois aussi les interfaces utilisées par les agents de sauvegarde pour des raisons de performance et d'isolation.

Se pose alors la question du choix de l'interface de sortie. La pile IPv4 d'un système utilisera une métrique pour choisir l'interface portant la route 0.0.0.0/0, la ou les autres interfaces se contentant de router uniquement le *subnet* auquel elles sont chacune attachée. Charge ensuite soit à l'administrateur du serveur de poser des routes statiques, soit au réseau d'effectuer du NAT pour qu'un flux d'administration arrive via une adresse de NAT appartenant au même *subnet* que l'interface d'administration.

Qu'en est-il en IPv6 ? La courte RFC 7608 indique que la décision de routage doit se baser sur la comparaison des bits des interfaces de la machine avec l'adresse de destination, ceci par incrément de bit en bit. L'interface ayant le plus de bits en commun gagne la manche.

Ainsi, une machine disposant de deux cartes avec les adresses ABBA:CAFE::5 et ABBA:1001::5 et envoyant un paquet à destination d' ABBA:C9D6::6 utilisera la première des deux cartes.

Ce point est à prendre en compte dans votre plan d'adressage afin de réserver un préfixe de haut niveau pour l'administration ou encore pour la sauvegarde. Cela facilitera l'usage d'interfaces dédiées si nécessaire.

Existe-t-il une autre méthode pour forcer l'utilisation d'une interface spécifique vers un préfixe off-link sans modifier la configuration de l'hôte et sans avoir préétabli un plan basé sur la RF 7608 ?

En IPv4 l'usage de l'option DHCP 121 (*classless static routes*) permet de pousser des routes fines à une interface (NB : cette option écrase l'éventuelle default route annoncée, elle-même devant être recopiée dans une option 121 si on doit l'annoncer).

Rien d'équivalent en IPv6, annoncer un préfixe via le *Router Advertisement* avec le bit L (*on-link*) à 0 n'entrainera pas l'apprentissage d'une route indirecte. Quant à DHCPv6 il n'intègre pas d'équivalent à l'option 121.

La RFC 4191 propose une extension (type 24) au RA permettant d'annoncer des routes, rédigée par Microsoft elle fonctionne depuis Windows Vista, le noyau Linux l'implémente également depuis les commit 930d6ff et ebacaaa de 2006. Cependant l'option n'est pas forcément activée.

Attention, cette RFC comporte deux parties, l'une concerne la bonne prise en compte de la priorité du RA, l'autre les routes additionnelles.

S'il vous est impossible d'utiliser l'option, vous pouvez toujours essayer d'envoyer des préfixes avec l'option on-link à 1. Les hôtes ajouteront alors une route vers le routeur pour ce préfixe. On s'écarte cependant de la norme.

CORRESPONDANCE V4 / V6

Comme évoqué dans la section dual-stack, l'usage parallèle d'IPv4 et d'IPv6 amène un surplus de configuration et d'exploitation, et donc des surcouts.

De bonnes pratiques permettent de faciliter la mise en œuvre d'automatisations limitant ces efforts.

Numéro de réseau / préfixe

Il est important de disposer d'un référentiel de correspondance entre un réseau IPv4 et le réseau IPv6 correspondant. L'idéal est de pouvoir disposer de cette fonctionnalité au sein de son IPAM, ou à défaut d'utiliser un champ dans la section IPv6 de celui-ci pour indiquer le réseau IPv4 associé avec son masque.

Si l'IPAM ne peut en aucune façon, même par contournement, stocker cette information alors il faudra se tourner vers un outil d'inventaire tiers. Il peut s'agir d'un autre référentiel du SI, d'une base dédiée … L'important étant que le référentiel soit « APIsé » afin de pouvoir l'exploiter depuis d'autres systèmes.

Prenons l'exemple des règles de filtrage des pare-feux, recréer toutes les règles existantes en IPv6 lors de son déploiement puis doubler le processus de demande d'ouvertures de flux sur le SI serait bien trop lourd.

En revanche, il est possible de mettre en œuvre des automatisations passant chaque nuit regarder si chaque objet représentant un réseau IPv4 n'a pas de correspondance en IPv6, et le cas échéant modifier l'objet afin de rajouter le préfixe IPv6 associé. On évite alors les erreurs, qu'elles viennent de l'administrateur des pare-feux ou des demandeurs qui risqueraient de se tromper dans leur requête d'ouverture de flux sur le préfixe IPv6.

Avec une solution plus évoluée, il est possible de gérer les modifications de façon synchrone, sans se soucier du dual-stack.

Numéro d'hôte / ID d'interface

Dans la deuxième moitié de l'adresse, on trouve les 64 bits dédiés à la numérotation des hôtes. Là aussi, de bonnes pratiques permettent de faciliter la mise en correspondance de l'adresse IPv4 et IPv6 d'un hôte dual-stack.

Ces pratiques ne sont évidements valables qu'avec DHCPv6 *stateful* ou un adressage manuel.

Le plus simple est de conserver le numéro IPv4 et de le reporter en IPv6. Prenons le réseau 10.2.3.128/25 et un serveur 10.2.3.**239**. Après déploiement d'IPv6, ce réseau exploite dans l'exemple suivant arbitrairement le préfixe ABBA:CAFE::/64.

Numéroter le serveur ABBA:CAFE::**239** facilite l'exploitation jusqu'au niveau de la lisibilité humaine. On peut également utiliser l'hexadécimal ABBA:CAFE::**EF** si on veut que les valeurs conservent la même numérotation stricte d'un point de vue binaire. On perd cependant la lisibilité.

Autre solution, plutôt que de conserver le numéro, on peut conserver le caractère ordinal. Avec ce même réseau, on voit que le serveur utilise la 89e IPv4 exploitable du réseau 10.2.3.128/25 (239-129=110) .128 est le numéro de réseau et n'est pour rappel pas assignable ici en IPv4.

Ce mode ordinal nous donne ABBA:CAFE::**110** ou ABBA:CAFE::**6E** en hexadécimal strict.

Les plus méticuleux d'entre vous auront noté que le numéro d'hôte ::0 est utilisable en IPv6 puisqu'il n'existe pas de numéro de réseau et d'adresse de broadcast, en se basant sur ce postulat on pourrait également convertir dans le mode ordinal une adresse IPv4 .1 en ::0 IPv6. Reste que ça n'est d'une part pas pratique en termes de lecture en raison du risque de confusion avec un préfixe, et peut d'autre part poser problème sur des systèmes, par exemple en raison de contrôles de champ sont mal implémentés comme dit précédemment.

Le choix entre ces 2 méthodes et les 2 pendants (décimal ou hexadécimal) est à étudier. La première dans sa version décimale est clairement la plus pratique, mais d'autres critères peuvent entrer en jeu au fur et à mesure qu'on se rapproche d'un monde orchestré.

Voici quelques exemples :

Réseau IPv4	Hôte IPv4	N° d'hôte IPv6 - Report	N° d'hôte IPv6 - Ordinal
10.2.3.128|25	10.2.3.239	::239 déc |::EF hex	::110 déc |::6E hex | 239-129
10.2.4.0|24	10.2.4.239	::239 déc |::EF hex	::239 déc |::EF hex | 239
10.5.0.0|23	10.5.0.239	::239 déc |::EF hex	::239 déc |::EF hex | 239
10.5.2.0|23	10.5.3.239	Relatif ::1239 déc |::4D7 hex Absolu ::3239 déc |::CA7 hex	::495 déc |::1EF hex|(256+239)
10.6.0.0|16	10.6.28.239	::28239 déc |::6E4F hex 28 blocs d'un octet+239	::7407 déc |::1CEF hex (28x256) + 239
	10.6.28.3	::28003 déc |::6D63 hex 28 blocs d'un octet+003	::7171 déc |::1C03 hex (28x256) + 3
10.8.64.0|18	10.8.72.50	Relatif ::8050 dc |::1F72 hex (72-64)=8 blocs + 050 Absolu ::72050 d|::11972 hx	::2098 déc |::832 hex (8x256) + 50

On note avec le tableau en exemple que pour un réseau IPv4 coupé au niveau du dernier octet (/24), l'ordinal a la même valeur que le report, puisque le comptage démarre à 0 dans les 2 cas.

Ça devient plus complexe avec un réseau IPv4 plus large qu'un octet, dans l'exemple un /23. Que faire ici pour séparer 10.5.0.239 de 10.5.1.239 ? Ajouter un 1 pour indiquer qu'on bascule au-delà du dernier octet semble une bonne méthode. On compte alors l'ensemble des adresses des /24 constituant le réseau, adresses non assignables incluses, soit 256.

Mais la quête de lisibilité aurait aussi pu nous faire recopier l'octet précédent et établir ::3239 plutôt que ::1239, migrant ainsi d'un référentiel relatif à un absolu. D'ailleurs, on peut même recopier toute l'IPv4 de l'hôte dans son n° d'hôte IPv6, pas la plus élégante des solutions cependant.

Les exemples suivants illustrent également le besoin de conserver les 0 des octets dans le mode « Report » afin de ne pas générer de doublons. 003, 050, etc.

Vous l'aurez compris, l'important est de bien définir les règles d'ingénierie et de s'y tenir.

En résumé :

- Le mode report décimal en absolu, c'est-à-dire en copiant l'octet complet, voire 2 octets pour les réseaux plus larges que le /24 (etc.) gagne clairement pour la lisibilité de correspondance. Il implique cependant des adresses d'hôtes longues ;
- L'usage de l'hexadécimal n'est probablement avantageux que dans un environnement automatisé ;

- 2 BE or not 2 BE, l'usage de l'hexa et des puissances de 2 font travailler l'esprit ;
- Là encore, ces solutions permettront de générer des ACL, etc. sans devoir doubler le travail.

La correspondance peut également être faite via les enregistrements DNS A et AAAA de chaque serveur, ce qui requiert alors une autre forme de rigueur.

Concernant les hôtes, il ne semble pas aujourd'hui exister de produit qui permette sans configuration préalable d'assigner le même numéro d'hôte en IPv4 et IPv6 en se basant sur une correspondance IPAM intégrée.

POUR LES RÉSEAUX NATIFS V6

Lors de la mise en place d'un réseau natif IPv6, les règles précédentes concernant l'hôte n'ont pas lieu d'être. Vous pouvez alors utiliser une partie des 64 bits pour indiquer des informations sur l'hôte.

Par exemple, une lettre pour indiquer un serveur bureautique, un autre caractère pour spécifier une imprimante. Ça doit vous rappeler des règles d'ingénierie existante pour le nommage des hôtes.

En datacenter, on peut imaginer marquer le métier associé à une VM, etc.

Cela reste néanmoins complexe et redondant avec une CMDB, d'autant que l'adresse ne se change pas facilement si besoin.

L'autre solution, tout du moins sur la partir serveur est de fixer l'interface ID afin qu'il soit statique sans dépendre de l'adresse MAC (et donc ne pas changer lors d'un changement de carte, physique comme virtuelle, et ne pas exposer le fabricant dans l'adresse), en conjonction de SLAAC pour la fourniture du préfixe. Cette solution restant plus simple que la définition statique partout.

En général, on se contentera de définir une segmentation par plage dans les petits réseaux pluriusages sur de petits sites.

ANNONCES PUBLIQUES

Qu'annoncer sur internet ?

À cette question, certains répondront « Le moins de ressources que possible pardi ! ». En soit, est-ce qu'annoncer directement son /32 au lieu de quelques /44 de DMZ représente vraiment une réduction de la surface d'attaque ? Cela changera-t-il quelque chose dans l'implémentation de pare-feu *stateful* et IPS ? La cible du bout en bout finira de toute façon probablement par forcer les annonces larges.

Lorsqu'on regarde le contenu des tables BGP en IPv6, on s'aperçoit que la majorité des annonces sont des /32, /40, /44 et /48.

Adds and Wdls per Prefix Length

/28	+1	-0
/29	+41	-13
/30	+3	-2
/31	+1	-3
/32	+131	-110
/33	+49	-31
/34	+53	-58
/35	+35	-11
/36	+141	-26
/37	+4	-3
/38	+28	-3
/39	+4	-9
/40	+436	-55
/41	+8	-2
/42	+9	-9
/43	+26	-3
/44	+147	-74
/45	+42	-6
/46	+67	-126
/47	+8	-3
/48	+871	-969
/52	+2	-0
/56	+23	-27
/128	+4	-0

Figure 10 ***Variations d'annonces v6 sur une semaine***

https://cidr-report.org/v6/as2.0/

Les annonces en /48 représentent la moitié des annonces en nombre avec 54 000 routes, mais pas en volume d'adresses uniques évidement puisque chaque /32 contient 65 536 (2e16) fois plus d'adresses qu'une /48.

http://bgp.potaroo.net/v6/as6447/

https://bgp.potaroo.net/index-v6.html

https://www.cidr-report.org/v6/as2.0/

La dernière URL offre notamment le rapport hebdomadaire suivant avec les ajouts et retraits de préfixes observés.

Focus

Certaines solutions d'anti-DDOS fonctionnent en réannonçant de façon fine le préfixe contenant les ressources attaquées via un réseau de « nettoyage ». Le /48 étant le plus petit il vous faudra forcément annoncer plus large en nominal si vous exploitez ce type de solution.

Idéalement, annoncez de façon large en fonction des emplacements géographiques de vos points de sortie. La mise en place de *peering* IPv6 pourra également être l'occasion de s'essayer à la signature des routes via RPKI si vous ne le faites pas déjà, ou encore expérimenter RTBH et Flowspec.

5

Sécurité
et bonnes pratiques

1. **Couche Accès** **75**
   - AFFECTATION DYNAMIQUE D'ADRESSES 75
     - Mécanismes 75
     - Identification DHCP 77
   - BLOCAGE D'ICMP REDIRECT 77
   - IPV6 SNOOPING 78
     - Fragmentation ND 79
     - Binding 79
     - Source 80
     - Destination 80
     - Déménagement 80
     - ND suppress 80
     - Préfixe 81
     - Cache poisoning 81
   - ROGUE DHCP 81
     - Physique 81
     - Logique 82
   - RA GUARD 82
   - RA HOP LIMIT 83
   - AUTRES CONFIGURATIONS DU RA 84
   - seND (NON-EXPLOITABLE ACTUELLEMENT) 85
   - MLD 86
   - STORM CONTROL 87
   - ADRESSES MULTICAST À BLOQUER 87
2. **Hôte** **88**
   - DHCP 88
     - DHCP DUID 88
     - DHCP Identity Associations 89
     - DHCP en l'absence de RA 89
     - Prise en compte des options DHCP en Dual-Stack 89
   - MÉTHODE DE GÉNÉRATION D'ADRESSE SLAAC 89
     - Temporary address 89
     - Randomized interface ID 90

Stable privacy address 90
Synthèse SLAAC 91
Spécificité de la Link-Local 91
◗ NE PAS DÉSACTIVER LA STACK IPV6 92
◗ DÉSACTIVATION DES MÉCANISMES DE TRANSITION 92
◗ DÉSACTIVATION DES PROTOCOLES D'AUTO-DÉCOUVERTE 92
◗ BLOCAGE DU TRAFIC LINK-LOCAL 93
◗ VPN 94
◗ CONFIGURATION D'OS DESKTOP 94
Windows 94
Linux 94
Network Manager 95
Systemd Networkd 95
NETPLAN 95
Par distribution 96
◗ MOBILE ET EMBARQUÉ 96
Android 96
Autres OS 97
3. Transit **98**
◗ URPF 98
◗ PROTECTION DU CONTROL PLANE 98
◗ SÉCURISATION OSPF 98
4. Filtrage **99**
◗ ICMP 99
◗ MÉCANISMES DE TRANSITION 101
◗ BOGON PREFIXES ET ROUTES 102
◗ EXTENSION D'EN-TÊTE 104
◗ POLITIQUE DE BANNISSEMENT 105

Sécurité et bonnes pratiques

V. Sécurité et bonnes pratiques

L'arrivée de cette nouvelle version du protocole IP nécessite la mise en place de règles de sécurité à divers niveaux du SI. Pour beaucoup, ces règles sont similaires aux bonnes pratiques IPv4, d'autres sont spécifiques à IPv6.

Il est particulièrement important de maitriser les mécanismes de sécurisation d'IPv4 ainsi que le fonctionnement d'IPv6, notamment sur les modes d'affectation des adresses et le déroulement des opérations du *Neighbor Discovery Protocol* (RFC 4861) avant d'entamer cette section.

Afin de faciliter l'attribution des études de transposition de ces règles sur vos équipements dans votre organisation, celles-ci sont regroupées sous les périmètres suivants :

- TRANSIT pour tout équipement L3 (routeur, etc.) ;
- ACCES traite les spécificités de la couche réseau portant les hôtes, switchs et routeurs. Et s'attarde notamment sur NDP et les échanges avec la couche 2 du modèle OSI (ici MAC) ;
- HOTE concerne les terminaux, principalement serveurs et postes lourds ;
- FILTRAGE pour les pare-feux.

1. Couche Accès

AFFECTATION DYNAMIQUE D'ADRESSES

Le suivi des terminaux au sein de l'organisation amène à privilégier DHCPv6 *stateful* afin de tracer les équipements plutôt que l'autoaffectation basée sur SLAAC. Les serveurs peuvent évidemment utiliser de l'adressage statique manuel. Certains terminaux, principalement basés sur Android n'intègrent pas de client DHCP, amenant parfois des exceptions. cf. le chapitre dédié.

Commençons par rappeler que l'usage du *Router Advertisement* est nécessaire, même avec un DHCPv6 *stateful*. À minima ce RA donnera l'adresse *Link-local* de la passerelle et les *timers*. Seul un fonctionnement entièrement manuel et non recommandé permet de se passer du RA.

Mécanismes

Plusieurs bits permettent d'indiquer à l'hôte comment se comporter :

- A – "Autonomous Address Autoconfiguration" indique à l'hôte s'il doit s'autoaffecter son adresse (SLAAC RFC 4862) ;

- M – “Managed Address Config” informe au contraire de la nécessité d’utiliser DHCPv6 *stateful* (RFC 3315) pour obtenir son adresse IP ;
- O – “Other Config” annonce que d’autres options sont disponibles auprès d’un DHCPv6 *stateless* (RFC 3736), ces options peuvent inclure serveurs DNS, NTP, suffixe de domaine, etc. ;
- L – “On-Link” informe que le RA et le préfixe qu’il annonce se trouvent bien sur le même lien de niveau 2. Ce bit est remis à 0 si l’annonce traverse un routeur, pour éviter les effets de bord d’une mauvaise configuration d’extension L2 ou de relayage involontaire. Contrairement à IPv4, un hôte IPv6 ne considère pas nativement qu’un hôte situé dans le même préfixe que lui est forcément joignable directement en L2 (sauf pour les adresses *Link-local*), ce bit doit être à 1 pour qu’il raisonne ainsi. Voir la RFC 5942.

Notez que les bits A et L sont envoyés avec le préfixe, et non au cœur du *Router Advertisement*.

Méthode	Bit A Auto	Bit M Manag	Bit O Other	Adresses résultantes	Fourniture des options
SLAAC	1	0	0	-Temporaire SLAAC (Éphémère) -Lien-Local	RDNSS RFC 6106
Stateless DHCPv6	1	0	1	-Temporaire SLAAC (Éphémère) -Lien-Local	DCHPv6
Stateful DHCPv6	0	1	Redondant avec M	-DHCPv6 -Lien-Local	DCHPv6
DHCPv6 Stateful + SLAAC (choix OS hôte)	1	1	Redondant avec M	-Au choix de l’OS -Lien-Local	DCHPv6 ou RDNSS selon OS

La troisième méthode du tableau, *stateful* DCHPv6, est idéale grâce à sa facilité de suivi de terminaux.

Vérifiez bien que vous ne tombez pas par erreur de configuration dans le dernier cas de figure, celui où le comportement n’est pas prédictible.

En effet, lorsqu’on expose un système à un RA avec les bits A et M à 1, la majorité des systèmes configurent les 2 et exploitent seulement l’une des méthodes pour les connexions sortantes (généralement SLAAC *temporary address* + *privacy extension*), ou laissent la RFC 7608 décider (comparaison bit à bit de l’adresse d’interface la plus proche de celle de destination du paquet). À ne tester que si vous aimez la loterie donc. Ce comportement est courant sur les routeurs domestiques. La RFC 4862 évoque dans sa section 5.6 et indique que c’est l’information obtenue le plus récemment qui devrait faire foi.

Il est possible de fournir plusieurs adresses dans différents préfixes à un client via DHCP, ce cas particulier n’est pas couvert ici.

Autre point, si jamais vous déployez une infrastructure spécifique avec le deuxième cas (*stateless* DHCP), vos serveurs n’ont alors pas de baux à synchroniser, mais uniquement la configuration des options fournies pour chaque préfixe.

Notez que certains OS comme Windows enverront une requête DHCP même si le RA indique de faire du SLAAC sans option flag.

Identification DHCP

DHCPv6 ne se base pas sur l'adresse MAC comme en IPv4, à la place l'hôte fournit un identifiant nommé DUID. Une section détaille cet identifiant plus après dans la partie Hôtes du chapitre sécurité.

DHCPv6 présente des options qui existent en IPv4 sous la forme de sous options 82 et en ajoute.

- Vendor class (Option 16) permet à l'équipement client d'envoyer son fabricant, modèle, version, etc. ;
- Vendor Specific (Option 17) pour des options propriétaires ;
- Interface-ID (Option 18) qui permet d'identifier le nom d'une interface et le VLAN (circuit-ID en DHCPv4) ;
- Remote-ID (Option 37) RFC 4649 qui peut récupérer le port physique, l'identifiant utilisateur fourni à un VPN, et surtout la MAC ;
- Subscriber-ID (Option 38) est plutôt utilisé par des opérateurs pour d'autres informations d'identification.

Par abus de langage, ces options sont souvent référencées comme option 82 aussi pour DHCPv6, alors qu'option 82 est celle en DHCPv4.

Dans les équipements d'accès, il est possible de placer la MAC du client dans l'option Remote-ID. Ce point est important, c'est lui qui va permettre à coup sûr de récupérer l'adresse MAC d'un hôte.

D'autres préconisations relatives au DHCPv6 facilitant l'identification d'un terminal se trouvent dans la section Hôtes.

BLOCAGE D'ICMP REDIRECT

Neighbor Dicovery Protocol comporte 5 types de messages possibles :

- *Router Solicitation* et *Advertisement* ;
- *Neighbor Solicitation* et *Advertisement* ;
- *Redirect.*

Ce dernier type de message permet à la passerelle d'indiquer qu'un autre routeur sert à joindre une destination donnée et que l'hôte doit mettre à jour sa table de routage en conséquence.

L'ICMP *redirect* (Type 137) est à bloquer, il peut permettre à un attaquant de rediriger du trafic. L'option n'est à conserver que lorsqu'un segment réseau porte 2 routeurs permettant d'atteindre des ressources différentes ; cas particulier rarissime.

IPV6 SNOOPING

Commençons par rappeler brièvement l'usage des deux types de messages les plus fréquents au sein de NDP.

Les messages *Neighbor Solicitation* (135) et *Advertisement* (136) permettent d'établir le lien avec la couche 2 au sein d'un segment réseau, typiquement demander quelle est l'adresse MAC d'un hôte à partir de son IP et répondre. Usage similaire à ARP en IPv4 donc.

La sollicitation se fait en multicast, un mode unicast permet également de vérifier qu'un hôte est toujours joignable, on spécifie alors qui pose la question. (*Target Address*).

Lorsque cette adresse est non spécifiée (::/128) alors le message est un DAD (*Duplicate Address Detection*)

La réponse à un NS comporte un bit O "*Override*" qui est par défaut à 1 pour spécifier d'écraser toute entrée existante dans un cache ND. La RFC indique que l'usage à 0 est prévu pour les réponses proxisées à des sollicitations, ou pour des adresses de services en anycast.

En terme concret, les deux exemples suivants :

- Un proxy ND (équivalent proxy ARP) n'écrasera pas via sa réponse une réponse directe que l'hôte concerné aurait pu émettre directement ;
- Deux serveurs avec la même adresse anycast dans un segment ne chercheront pas à écraser les entrées les concernant.

Le bit S "*Solicited*" spécifie que la réponse est destinée à une requête *unicast* avec *Target Address*, donc à une demande de joignabilité.

Enfin, le bit R "Router" indique que l'hôte est un routeur, s'il passe à 0 le *Neighbor Unreachability Detection* va en déduire que l'hôte n'est plus en capacité de router. Il initiera alors une sollicitation de routeurs, et basculera sur tout autre routeur disponible (selon les priorités si plusieurs)

Avant même de reparler de *Router Solicitation* et *Advertisement*, vous aurez déjà noté tout ce qu'un attaquant peut faire avec les informations de voisinage de NDP. Aussi est-il vivement recommandé de mettre en œuvre les mécanismes d'anti spoofing adaptés à minima sur les infrastructures d'accès des campus/sites utilisateurs.


Focus

NDP exploite des groupes multicast nommés *Solicited-Node Multicast*, chaque hôte va créer un groupe multicast pour chaque adresse assignée à partir d'un préfixe standardisé FF02:0:0:0:0:1:FF00::/104 et des 24 derniers bits de l'adresse à représenter. Ce sont ces adresses multicasts qui sont utilisées pour le DAD, mais aussi pour pouvoir effectuer la correspondance MAC/IP sans déranger tout le monde comme le fait le broadcast ARP en IPv4.

Le premier contact entre deux nœuds IPv6 d'un même réseau est donc toujours un multicast.

Fragmentation ND

Les messages RA peuvent être larges s'ils comportent plusieurs préfixes et nécessiter de la fragmentation, la RFC 6980 indique qu'il vaut alors mieux envoyer plusieurs messages plutôt que de fragmenter le paquet. De toute façon, sauf configuration particulière, il n'y a pas de raisons d'avoir de nombreux préfixes et options dans un RA amenant à atteindre 1280 octets, le minimum IPv6.

En découle la préconisation de bloquer les fragments de protocole NDP.

Binding

Les mécanismes de sécurité se basent sur la constitution d'une table de relation entre IP, MAC et emplacement physique, typiquement le port d'un switch.

Le plus simple étant d'utiliser le DHCP *snooping* qui va permettre d'exploiter les messages d'affectation d'IPv6 retournés par le DHCPv6 afin de construire une table de contrôle dite de binding.

L'inspection ND, DHCPv6, etc. ne sont pas toujours implémentées de la meilleure des façons. Certaines sécurités fonctionnent parfaitement avec des extensions d'entête et même de la fragmentation. D'autres ne fonctionnent que dans le cas le plus simple. Cet écart s'explique souvent par les capacités des ASICS de l'équipement. Sur certaines gammes, le traitement remonte au control plane et est incompatible avec des options d'optimisation hardware.

Côté configuration, ces fonctionnalités sont parfois un package uniforme, parfois la somme de plusieurs options à paramétrer indépendamment, parfois même les 2 modes co-existent et l'activation de l'une coupe l'autre.

Vérifiez donc scrupuleusement la documentation constructeur et testez avec un forgeur de paquets comme Scapy.

Le déclenchement d'un évènement de sécurité lié à cet ensemble de règles doit remonter une alerte dans votre SIEM.

N'oubliez pas de mettre en œuvre également la récupération de la table de *binding* afin qu'elle se repeuple immédiatement lors d'un redémarrage de switch. Il est en général possible de l'exporter périodiquement et/ou de pouvoir requêter les baux actifs au serveur DHCP (Si vous utilisez DCHP *Stateful*).

Il est possible d'utiliser une partie de ces sécurités sans DHCP, mais la perte d'une source d'apprentissage sûre affecte le niveau de protection (RFC 6620). On voit d'ailleurs maintenant des structures où le DHCP est mis en œuvre avec des réservations au sein du datacenter afin d'offrir ce niveau de sécurité sur les raccordements de serveurs. Plus de configuration manuelle des hôtes.

Sans DHCP, l'équipement construira la table à base des messages DAD échangés lors de l'autoaffectation SLAAC.

Notez que dans les solutions à base de *fabric* L3, le protocole de signalisation transporte les informations nécessaires à la création de la table, l'inspection n'est requise que pour certaines configurations particulières. Par exemple sur une infrastructure EVPN+VxLAN, les routes EVPN de type 2 annoncent le couple MAC/IP.

Différents contrôles pourront ensuite se baser sur cette table, en voici les principaux :

Source

Un paquet dont l'adresse source est inconnue, non allouée sera détruit. Le switch pourra tenter de demander au serveur DHCP et/ou à son voisinage via NDP si l'adresse est connue avant de détruire le trafic.

Ce contrôle nécessite la présence d'une table de binding, il n'effectue pas lui-même d'inspection ND.

N'oubliez pas d'autoriser le trafic sur l'adresse de lien local, parfois une commande supplémentaire et de marquer comme sûr les ports de ressources statiques manuellement configurées comme les serveurs.

Destination

Lorsqu'un paquet arrive, l'équipement ne va le transmettre et effectuer la résolution ND si nécessaire seulement si le destinataire est connu dans la table de binding. Dans le cas contraire, le paquet sera détruit.

Ce mécanisme permet de contrer du trafic à destination d'une adresse malformée ou inexistante, à des fins de déni de service local par exemple.

Déménagement

Lorsqu'un hôte change de port, le suivi d'emplacement physique peut initier une sollicitation ND à destination de l'hôte sur la précédente position connue dans la table de binding. Si une réponse est obtenue alors le nouvel arrivant est un usurpateur.

Cela rend l'attaque inefficace tant que l'hôte original est en ligne et en capacité de répondre.

ND suppress

Afin d'optimiser le trafic et limiter le multicast, il est possible de laisser l'équipement d'accès répondre aux requêtes NS *Neighbor Solicitation* à la place de l'hôte concerné. Cette fonction peut s'activer à minima pour les requêtes multicast, mais également pour l'unicast. Le *ND/ARP suppress* est une fonctionnalité courante sur les *fabrics* EVPN/VxLAN (où l'apprentissage se fait différemment), mais on le retrouve aussi sur les gammes campus.

Cependant, souvenez-vous que l'un des usages des requêtes unicast, celui avec la *Target Address*, est de vérifier la joignabilité de l'hôte. Il n'est donc pas pertinent de répondre à la place de l'hôte pour autre chose que le multicast sauf si l'équipement fait la différence entre les requêtes *unicast* avec et sans adresse de destination, et ne prend position que pour ces dernières.

Dit autrement, un équipement ne devrait jamais avoir à envoyer un *Neighbor Advertisement* avec le bit S à 1 à la place de l'hôte concerné.

Exception possible, le Wi-Fi où le suivi de la liaison radio avec la station par le point d'accès peut autoriser à répondre à la place de celle-ci même pour un test de joignabilité. Priorité à la libération du média sous-jacent, le canal radio.

Préfixe

En se basant sur les informations obtenues à partir des sources suivantes :

- *Router Advertisement* ;
- *DHCP-Prefix-Delegation* ;
- Une configuration manuelle le cas échéant.

Le contrôle de préfixe permet de bloquer un paquet dont l'adresse routable source n'appartient pas au préfixe en cours d'utilisation dans le segment L2. On bloque ainsi l'usurpation d'adresse dès la couche d'accès avant même de recourir ultérieurement à URPF lors du routage par exemple.

Cache poisoning

Comme son ainé l'ARP *cache poisoning*, il est possible de remplir le cache ND des hôtes pour le saturer. D'autant qu'avec 2^64 adresses possibles dans un réseau, un attaquant a de quoi faire.

L'une des attaques courantes est de se faire passer pour le routeur dans un *Neighbor Advertisement* avec le bit R à 0, indiquant qu'on ne route plus. L'attaquant peut également tenter un *man-in-the-middle* en se faisant passer pour un hôte ou pour le routeur.

Les sécurités liées au binding empêchent ce comportement, mais il reste recommandé de spécifier une limite de taille de cache sur les équipements réseau. Si vous souhaitez calculer une limite fine, n'oubliez pas qu'il ne suffit pas de compter les hôtes, mais bien les adresses. Chaque hôte en ayant au minimum 2 et pouvant en avoir plus (SLAAC avec adresses temporaires par exemple). Les OS modernes ont des valeurs par défaut généralement adaptées.

Pour plus d'infos, voir la RFC 6583.

ROGUE DHCP

Physique

Décrit dans la RFC 7610, le mécanisme DHCPShield implique de définir les ports physiques pouvant recevoir le trafic du serveur DHCP. Il s'agit généralement des ports d'*uplink*. Le trafic de type DHCP server en provenance de ports non déclarés sera détruit.

L'équipement devra analyser la totalité du contenu de tout message provenant du serveur DHCP. Là encore, soyez vigilant selon l'ASIC et l'implémentation.

Si l'équipement ne supporte pas la fonctionnalité, il reste possible d'utiliser une ACL bloquant le trafic source port UDP 547 / destination UDP 546, mais ça ne fonctionne cependant pas avec un paquet fragmenté forgé.

Logique

La très longue RFC 8415 qui traite de DHCPv6 comporte une section sur la sécurisation des échanges entre le serveur et les clients et/ou relais.

IPsec peut être utilisé pour authentifier, voire chiffre les échanges DHCP entre les serveurs et les relais, RFC 8213. La configuration cryptographique peut être paramétrée manuellement ou se baser sur une PKI.

L'usage d'IPsec peut d'ailleurs servir à sécuriser d'autres échanges d'administration comme Syslog, SNMP, NTP, RADIUS, etc.

Attention, le support d'IKEv2 avec des secrets prépartagés n'est pas obligatoire dans cette RFC.

L'usage d'une simple clé partagée permet pour rappel à un attaquant le rejeu des paquets. RDM limite pourtant les risques de rejeu, mais uniquement côté client, et pas entre un relai et le serveur.

Beaucoup de RFC devenues obsolètes proposaient des mécanismes d'authentification. À ce jour la RFC 7227 traitant de la création d'options DHCP sert de base à de nombreuses propositions. Vous pouvez notamment vous documenter sur les travaux DHCPv6Sec et Secure-DHCPv6.

Dernier maillon de sécurité disponible dans la RFC 8415, RKAP (*Reconfiguration Key Authentication Protocol*) permet d'empêcher la reconfiguration d'un client par un serveur malveillant. Une clé unique est envoyée au client lors de la 1ère réponse. Le serveur utilise ensuite HMAC-MD5 pour signer ses messages.

Cependant, RKAP est récent et n'est pas encore exploitable en pratique.

Au passage, la reconfiguration est une nouveauté qui permet de forcer les clients à requêter à nouveau le DHCP (sans attendre l'approche de l'expiration de leur bail ou par suite d'un redémarrage). Ceux parmi vous qui ont déjà eu à faire redémarrer des centaines d'équipements en PoE pour leur faire prendre en compte une nouvelle option via DHCP apprécieront. N'en profitez pas pour lancer un DDoS contre vous-même en vous essayant à cette nouveauté sur un trop large périmètre…

En résumé, mettez en œuvre IPsec entre vos relais et le serveur, et laisser le volet DHCPShield traiter la sécurité de la partie relais/client uniquement d'un point de vue port d'arrivée des messages serveur DHCP autorisé.

Enfin, rappelez-vous toujours que DHCPv6 peut fournir plusieurs IPv6 au même client (DUID).

RA GUARD

Les messages Router Advertisement sont un point clé d'IPv6, il est nécessaire de s'assurer que ceux-ci sont émis par un routeur autorisé.

La RFC 6105 propose de définir manuellement un ou plusieurs des éléments suivants dans les équipements d'accès afin de valider ou de bloquer un message RA :

- le port physique ;
- l'adresse MAC du routeur ;
- l'IP de la passerelle ;
- le préfixe annoncé ;

- la priorité RA ;
- le *Hop-Count limit* ;
- La valeur des bits M - *Managed* et O – *Other*.

Le plus simple est généralement d'autoriser les interfaces d'*uplink*, notez qu'il est aussi souvent possible d'imposer une limite de TTL.

La RFC propose également un mode d'apprentissage dit *stateful*, au cours duquel l'équipement apprendrait la/les sources de RA pendant une période donnée. Au-delà il n'accepterait plus de nouvelles sources de RA.

Ce mode *stateful* commence à apparaitre dans les équipements.

Notez si le routeur bascule vers un jumeau usant d'un protocole de type NHRP, il faudra s'assurer que l'absence d'un voisin mémorisé fera retourner l'ensemble en état d'apprentissage, ou que les éléments contrôlés ne changent pas (une Virtual MAC ou IP par exemple).

Si l'équipement ne supporte pas RA guard vous pouvez à minima bloquer les RA en *ingress* avec une ACL sur les ports d'accès.

RA HOP LIMIT

Pour éviter qu'un *Router Avertisement* puisse sortir du segment, la section 6.1 de la RFC 4861 rappelle les contrôles élémentaires à faire sur les messages ND. Ceux-ci comme la destruction de RA avec un *hop-limit* inférieur à 255 doivent fonctionner d'office, sans configuration spécifique de sécurité. Le draft ND Shield https://tools.ietf.org/html/draft-gont-opsec-ipv6-nd-shield-00 propose d'aller plus loin.

Cette sécurité vous rappellera peut-être ce qui existe en BGP avec GTSM (*Generalized TTL Security Mechanisms*) RFC 5082. GTSM détruira un message BGP si son TTL/hop-limit est inférieur à 254, car cette fois c'est certain, ça ne vient pas du voisin. (Sauf lors de l'ajout de l'option BGP multihop bien évidemment).

N'oubliez pas d'adapter la configuration d'éléments intermédiaires afin qu'ils ne décrémentent volontairement pas le *hop-limit* dans certaines configurations particulières comme une extension L2 d'un réseau ou simplement si vous utilisez des switchs L3 datacenter en MLAG.

Attention, certaines documentations constructeurs parlent d'éditer la valeur du RA *hop-limit* et donnent souvent une valeur à 64 par défaut. Il s'agit en fait du champ *Current hop-limit* (CHL) qui indique aux hôtes recevant le RA la valeur de *hop-limit* à configurer de leur côté.

AUTRES CONFIGURATIONS DU RA

Après avoir vu les points spécifiques à la sécurité et aux modes d'affectation d'adresses, voyons une partie des autres réglages du router advertisement. Ces paramètres sont à configurer sur chaque interface.

- *RA interval* : délai en secondes entre 2 émissions non sollicitées de RA, avec une valeur minimale et une maximale.
    - La maximale doit se situer entre 4 et 1800. Le défaut est de 600s ;
    - La minimale entre 3 s et ¾ de la valeur de la maximale. La valeur par défaut est 1/3 du max, ou 3s si le max est inférieur à 9s.
- *RA lifetime* : durée de vie au-delà de laquelle le routeur est considéré comme à ne plus utiliser. La valeur doit se situer entre l'interval MAX et 9000 secondes. Le défaut est 3 x interval max.
    - Une valeur de 0 indique que le routeur est à ne pas utiliser par défaut ;
    - Dans le cas d'une interconnexion point à point entre 2 routeurs, par exemple un *peering* BGP, le *RA lifetime* sera normalement ignoré, l'état de vie du voisin étant supervisé via le protocole de routage lui-même.
- MTU : il est possible d'indiquer le MTU du lien aux hôtes, la valeur par défaut est 0.
    - Si vous rencontrez des problèmes avec Path-MTU-D sur un site, vous pouvez positionner temporairement cette valeur afin de traiter le problème dans la direction sortante le temps d'identifier le problème. C'est plus rapide que de configurer chaque hôte.
- Préfixe : le routeur annonce un ou plusieurs préfixes routables, avec pour chacun
    - *Lifetime* : durée de vie de la route, peut-être spécifiée en secondes depuis la dernière annonce, ou via une heure fixe. Cette dernière option peut permettre de décommissionner proprement un préfixe avant de le retirer de la configuration. La valeur par défaut est de 2592000 secondes restantes, soit 30 jours. Il est déconseillé d'utiliser la valeur 0xffffffff qui a pour effet de rendre la route valide de façon permanente, un bon moyen d'avoir un trou noir si le routeur change d'adresse de lien local.
    - *On-Link* (Bit L) : déjà évoqué plus haut, il permet d'indiquer que le routeur est sur le lien, 1 par défaut.
- SLAAC
    - *Lifetime* : la durée préférée de validité des adresses que les hôtes autoconfigurent, là encore elle peut se configurer en secondes restantes ou avec une date/heure fixe. Le défaut est de 7 jours (604800 s). Et là aussi, il est déconseillé d'utiliser infini (0xffffffff). Notez enfin que la valeur ne doit pas être plus grande que celle de validité de la route du préfixe associé.
    - Si vous n'utilisez pas DHCP *stateless* avec le SLAAC, vous pouvez spécifiez l'adresse des serveurs DNS via RDDNS. (Obligatoire pour Android)
- Priority

- Router priority peut être positionné à bas, normal (défaut) et haut. Vous pouvez l'utiliser pour remplacer un routeur en douceur sans même devoir conserver la même IP. Il est une bonne pratique de le positionner à « haut » en temps normal afin de réduire les risques de bascule involontaire voire de compromission.

D'autres champs existent dans la RFC, mais ils ne sont pas utilisés et pas configurables sur la majorité des plateformes du marché. (*Reachable Time* et *Retransmit Time*)

Bon à savoir, les constructeurs implémentent une commande de statuts permettant d'afficher l'ensemble des préfixes émis avec l'interface associée.

seND (NON-EXPLOITABLE ACTUELLEMENT)

Secure Neighbor Discovery résulte de la volonté de pouvoir authentifier les messages NDP au sein d'une organisation, il est initialement décrit dans la RFC 3971.

Le protocole se base sur :

- Adresses générées d'après une base cryptographique RSA (CGA) RFC 3972
- PKI et point d'ancrage
- Horloge et Nonce pseudoaléatoire (antirejeu)

Lorsqu'un hôte se connecte, le routeur va lui indiquer la chaine de certification et le « *trust anchor* », cela amène un 6e type de message ND, le *Certificate Path Solicitation*. cf. les RFC 6494 sur les profils et gestions de certificats et la RFC 6495, champs X.509.

Qui dit certificat dit prise de poids des messages et nouveaux risques liés à la fragmentation, voir la RFC 6980.

Quand on se penche en détail sur la RFC, on s'aperçoit que les problèmes similaires à ceux du 802.1x existent. Si la RFC démarre par rappeler qu'IPsec n'était pas viable vu que NDP est le premier contact avec le réseau, on ne trouve pas pour autant de système de remédiation comme il en existe en 802.1x.

Il faudra que l'hôte ait préconfiguré au moins un *trust anchor*.

IMPORTANT

Les équipements réseaux commencent à implémenter SeND, en revanche rien du côté des systèmes d'exploitation usuels en dehors de projets universitaires.

SeND est donc malheureusement inexploitable à ce jour, et ne pourra servir qu'au sein d'une organisation avec des postes gérés, comme le 802.1x.

MLD

IPv6 fonctionne naturellement en multicast, là où celui-ci ne sert rarement au sein d'un réseau IPv4. Se cantonnant souvent à des protocoles de découvertes comme mDNS, SSDP, LLMNR ou encore lors de la mise en place d'OSPF.

De ce fait, le multicast n'est pas toujours bien implémenté au sein d'un segment réseau. Nous ne parlons même pas ici de routage multicast, mais bien d'échanges sur le même segment L2.

MLDv1 (RFC 2710) est l'équivalent d'IGMPv2 et exploite 3 types de messages :

- *Listener Queries*, générales pour demander à l'ensemble des *nœuds* s'ils sont membres d'au moins un groupe multicast, soit spécifiques, pour identifier les membres d'un groupe en se basant sur une adresse spécifique.
- *Listener* Reports pour que les hôtes répondent aux requêtes.
- *Done* pour informer qu'ils n'ont plus besoin de faire partit d'un groupe.

MLDv2 (RFC 3810) reprend les évolutions d'IGMPv3 et apporte le filtrage de la source (SSM), de façon à permettre l'inclusion ou l'exclusion de sources.

Les hôtes envoient des rapports lors de changement d'état en plus des rapports périodiques et le type de message « *done* » disparait (repris par le changement d'état).

Les messages sont retransmis pour rendre l'ensemble robuste face à la perte d'un paquet, une variable de robustesse « *robustness* » indique combien de fois les messages doivent être retransmis. La valeur par défaut est de 2, il peut être intéressant de l'augmenter sur du wifi par exemple.

MLDv2 est rétrocompatible avec MLDv1, notez qu'il s'inscrit au-dessus d'ICMPv6, contrairement à IGMP qui directement au-dessus d'IPv4.

MLD permet donc de connaitre les besoins des clients, notamment afin de les remonter à l'agent PIM dans le cas de multicast routé. Cependant, sans autre mécanisme, le trafic multicast se comporte comme du broadcast au sein du segment réseau. Il est envoyé à tous les ports.

MLD *snooping* permet d'optimiser la remise du trafic multicast en ne l'envoyant qu'aux hôtes le requérant et aux routeurs fournissant le service. Les équipements L2 vont analyser le contenu des échanges MLD afin de construire des tables associant ports et adresse multicast. En MLDv1 cette association se base sur l'adresse multicast de destination, en MLDv2 on y ajoute la/les adresses sources, SSM oblige.

Il est donc important que la fonctionnalité de *querier* MLD du routeur soit active, et que les équipements L2 exploitent les rapports MLD afin de mettre en œuvre le *snooping*.

Il est donc important que la fonctionnalité de *querier* MLD du routeur soit active (mrouter), et que les équipements L2 exploitent les rapports MLD afin de mettre en œuvre le *snooping*. Sans « mrouter » l'état est répliqué sur l'ensemble des switchs, ce qui est lourd et non souhaitable.

Si plusieurs routeurs essayent d'être querier MLD, celui avec la plus petite IP l'emporte. Cette petite optimisation évite les problèmes que l'on peut rencontrer en IPv4 avec IGMP où le gagnant est celui qui émet le plus fréquemment.

Ne négligez pas l'optimisation qu'apporte le *snooping* et vérifiez le bon fonctionnement sur l'ensemble de la chaine. Profitez-en pour vérifier IGMP sur IPv4 en même temps.

Dans les environnements denses type datacenter, prenez le temps de réfléchir à la répartition des arbres multicast sous-jacents dans des *fabrics* EVPN+VxLAN. La bonne pratique est généralement de répartir les réseaux sur au moins 2 arbres d'*underlay*, et de créer des arbres dédiés pour les réseaux comportant des hôtes gourmands en multicast (cluster, émetteur vidéo, etc.). Cette pratique peut prévaloir aussi sur d'autres topologies à base d'overlay/*underlay*.

En résumé, bien que MLDv2 ne soit techniquement requis que lors de l'usage de SSM, sa capacité à tolérer la perte d'au moins 1 paquet est un avantage par rapport à la V1 (cf. valeur *robustness*). Le *snooping* est un impératif d'optimisation qui évite également une attaque via des adresses multicast inconnues ou sans hôtes clients.

En évoquant IPv6 et le multicast on pense de suite aux adresses de groupes de référence *Well-Know Multicast*, comme l'ensemble des routeurs (ff02::2) ou les serveurs DHCP (ff02::1:2). On oublie cependant les adresses *Solicited-Node Multicast* dont nous avons parlé plus haut.

Pour rappel chaque hôte va créer une adresse multicast pour chaque adresse configurée à partir des 24 derniers bits et du préfixe F02:0:0:0:0:1:FF00::/104. Ces adresses ne doivent pas être traitées par le MLD snooping, elles pourraient en effet rapidement saturer les tables (puisqu'on a au moins un groupe multicast par hôte). Ce bypass est parfois présent par défaut, parfois il faut utiliser une commande de type nd-workaround sur les configurations MLD snooping. Vérifiez avec le constructeur et regardez le contenu de la table MLD avec des hôtes en communication.

STORM CONTROL

Sécurité plus classique et simple, mettez en œuvre *storm control* pour le multicast et l'*unknown* à minima sur les *uplinks* des équipements d'accès. La 3e valeur broadcast également, bien qu'elle ne concerne qu'IPv4.

Sachez qu'il vaut toujours mieux une valeur élevée comme 30% du lien qu'aucune configuration en attendant de l'affiner après étude du trafic.

ADRESSES MULTICAST À BLOQUER

Il existe des adresses multicast à bloquer directement au niveau des équipements d'accès. Retrouvez-les dans la section « Désactivation des protocoles d'autodécouverte » de la partie Hôte.

2. Hôte

En dehors de rares exceptions (pare-feu avec profil), les réglages que vous appliquez à un hôte prennent effet, quel que soit le réseau auquel il est raccordé. Il est malheureusement impossible de créer des profils, par exemple pour ne pas utiliser d'adresse aléatoire lorsque le préfixe reçu dans le RA est celui de l'entreprise.

En conséquence, soyez prudent notamment pour les machines susceptibles de se connecter à des réseaux en dehors de votre organisation. Par exemple un utilisateur avec son PC portable à son domicile aura bien du mal à faire quoi que ce soit si l'administrateur a totalement désactivé SLAAC.

Vous pouvez en revanche durcir au maximum les serveurs.

DHCP

DHCP DUID

DHCP Unique IDentifier permet au serveur DHCP d'identifier le client et de suivre son bail. Il existe plusieurs méthodes de construction de cet identifiant, la plus simple étant l'adresse hardware (MAC).

Ce DUID est normalement persistant au sein d'un système, et ce quel que soit l'interface réseau. Par exemple un ordinateur portable ayant un DUID construit à partir de la MAC de sa carte Ethernet filaire utilisera la même valeur lors d'une requête via la carte wifi.

Les sources de construction possibles dans la RFC initiale 8415 sont :

- *Link-Layer Address* (DUID-LL) ;
- *Link-Layer Address Plus Time* (DUID-LLT) ;
- *Vendor Based on Enterprise Number* (DUID-EN) ;
- *Universally Unique Identifier* (DUID-UUID) RFC 6355.

Le premier est explicite, le deuxième ajoute l'horloge à la date de génération initiale, il est stocké et ne change plus, souvenez-vous.

Le troisième est au choix du constructeur.

Le quatrième, UUID, cherche à garantir la persistance pour un système démarrant à partir du réseau ou en plusieurs phases. Le démarrage d'un serveur en PXE avec un bootstrapper léger qui bascule ensuite sur un OS lourd est un cas intéressant :

Il dispose de plusieurs interfaces donc on ne peut pas garantir que le DUID-LL se base sur la même interface. Le *vendor* est différent entre le *firmware* de la carte PXE, le bootstrapper léger puis l'OS.

L'UUID pourra permettre un suivi constant si l'ensemble de la chaine se base sur la même information, par exemple le n° de série du système connu par l'UEFI.

La plupart des OS utilisent DUID-LLT par défaut, il n'existe pas de raison d'en changer.

DHCP Identity Associations

SI le DUID est unique pour un système, l'Identity Association est unique pour une interface donnée. Aucune configuration particulière ici.

DHCP en l'absence de RA

Si le *Router Advertisement* indique s'il faut utiliser ou non DHCPv6, que faire lorsqu'il n'y a pas de RA ?

La RFC 4862 indique qu'en l'absence de RA, un système peut faire du DCHP. Ceci est implémenté dans la plupart des OS. Sachez que certains OS font des requêtes même quand le routeur indique de faire uniquement du SLAAC.

Prise en compte des options DHCP en Dual-Stack

Dans la série des comportements non prédictifs, que se passe-t-il si un hôte *dual-stack* reçoit des options spécifiques à la fois en DCHPv4 et v6 et que ces options diffèrent dans leur contenu ?

Est-ce la précédence qui l'emporte ? le premier fournissant l'option ? Il peut être intéressant de le vérifier.

MÉTHODE DE GÉNÉRATION D'ADRESSE SLAAC

Initialement il était prévu que l'adresse SLAAC soit formée à partir de l'adresse MAC du système sous la forme de l'EUI-64. Toutefois cela pose de nombreux problèmes :

- La MAC étant unique, il devient possible de suivre un hôte sur internet, quel que soit le réseau depuis lequel il se connecte.
- Il est plus facile de lancer un scan d'adresses sur un réseau, l'utilisation d'EUI-64 offrant une certaine prédictibilité de ce qu'on peut trouver fréquemment sur les 1ers bits.
- Connaitre la MAC permet de connaitre le *vendor*, il devient alors par exemple possible de deviner avec quels marque et modèle d'équipement on discute en corrélant le *vendor* et le protocole utilisé lors de l'échange.
- Changer l'interface réseau va entrainer un changement de l'adresse SLAAC.

2 RFC proposent des approches limitant ces problèmes, voir :

- RFC 4941 *Privacy Extensions for Stateless Address Autoconfiguration in IPv6* ;
- RFC 7217 *A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration* (SLAAC).

Temporary address

L'adresse temporaire vient en complément de l'adresse stable (RFC 4941). Elle change plus ou moins fréquemment selon les paramètres de l'OS tout en respectant les durées de vie annoncées par le SLAAC du *Router Advertisement*.

Par exemple certains systèmes créent une nouvelle adresse toutes les 25 minutes, et déconfigurent totalement la précédente 5 min après la création de sa remplaçante si aucune session ne continue d'utiliser l'ancienne IP. Ainsi les nouvelles sessions initiés par l'hôte n'utilisent jamais une adresse plus de 30 minutes.

L'hôte reste cependant joignable en permanence via son adresse stable, d'ailleurs seule l'adresse stable fait l'objet d'un auto-enregistrement DNS.

L'usage d'adresses temporaires peut poser problème tellement leur vie est courte.

La RFC évoque le cas d'un serveur qui vérifierait qu'un enregistrement PTR reverse DNS existe pour le client avant de lui autoriser l'accès. Mais il est facile de trouver des cas bien plus courants :

Imaginons s'authentifier sur un site web pour accéder à un espace client alors que nous utilisons une adresse temporaire à sa 24ème minute d'activité.

Deux minutes plus tard, le serveur nous demande à nouveau de nous authentifier alors que nous avons navigué en continu depuis la connexion.

Ce cas est tout à fait plausible, si pour une raison de sécurité le serveur demande à ce que le client ait la même IP en plus de son cookie, il rejettera la session. De même si un *load-balancer* L4 frontal se met à rediriger le client vers un autre serveur n'ayant pas connaissance de sa session web, car pensant avoir à faire à un nouveau client, car nouvelle IP. Il n'existe aujourd'hui pas de mécanisme permettant aux navigateurs de communiquer à un serveur pour lequel un onglet de navigation serait actif (ou récemment actif) l'information de changement d'IP.

De la même façon, un jeu en ligne fonctionnant en P2P avec du *match-making* autohébergé pourrait voir ses parties interrompues au bout de quelques minutes.

Dans le cas d'un jeu, il serait souhaitable que le développeur prenne soin de faire monter les sessions via l'adresse stable, mais pour un navigateur cela reviendrait à annihiler totalement l'intérêt de la *temporary address*, le trafic web représentant la majorité des possibilités de pistage.

En prenant du recul, on peut se dire que le pistage (publicitaire par exemple) se contentera d'identifier le /64, ce qui suffit à identifier un foyer au même titre qu'une IPv4 aujourd'hui. Mais il n'est pas impossible que les publicitaires se mettent à garder en cache les IPv6 sur une semaine pour marquer comme stable celles vues plusieurs fois, exploitant donc forcément une adresse EUI-64 ou *Stable privacy*. Leur offrant finalement la possibilité de tracer l'utilisateur unique au lieu du foyer, et sans cookie ! À méditer…

Tout récemment, en février 2021, la RFC 8981 a apporté des modifications aux adresses temporaires.

Dans la liste des changements, on retrouve la possibilité de n'avoir que des adresses temporaires, plus de stable. La RFC n'impose toujours pas de mécanisme permettant d'exclure des préfixes de l'utilisation d'adresses temporaires, mais le préconise. La réponse de Microsoft risque donc par exemple de ne pas changer https://social.technet.microsoft.com/Forums/azure/en-US/e36e82e9-1911-4f4d-91a2-c62f6e04c9c1/ipv6-turn-off-privacy-extensions-temporary-addresses-for-certain-prefixes-ie-ula-in-win-10?forum=win10itpronetworking

Randomized interface ID

Plutôt que d'exploiter sa MAC, l'hôte va générer son adresse en se basant sur un identifiant pseudoaléatoire. Celui-ci change lors du redémarrage, les systèmes supportant la persistance de stockage vont se baser sur l'adresse précédente en plus du nombre pseudoaléatoire.

Stable privacy address

Ce mécanisme permet d'obtenir toujours la même adresse IPv6 tant que l'on est sur le même réseau, sans pour autant le conserver lors de connexion à d'autres réseaux. Ceci, car elle est calculée à partir de constantes intrinsèques de l'hôte conjointement au préfixe reçu.

Plus précisément, les éléments suivants :

- Préfixe reçu via RA ;
- N° d'interface (tel que vu par l'OS) ;
- Compteur DAD (0, s'incrémente si conflit) ;
- Clé secrète générée aléatoirement la première fois puis stockée ;
- Optionnellement de l'identifiant réseau, le SSID Wi-Fi typiquement.

Ainsi, il est impossible de suivre la machine lors de ses déplacements sur différents réseaux, impossible également de retrouver la MAC à partir de l'adresse. En revanche, l'aspect stable au sein de chaque réseau fréquenté facilitera le travail de l'administrateur qui a souhaité se passer de DHCPv6 *stateful*.

Synthèse SLAAC

Voici une synthèse de la traçabilité par type d'adresse. N'oubliez pas que l'adresse globale est routable et donc potentiellement visible absolument partout sur internet.

Méthode SLAAC	Traçabilité locale	Traçabilité globale	Informations sur l'équipement	Traçabilité depuis le même réseau dans la durée
EUI-64 (MAC)	OUI	OUI	OUI (vendor)	OUI
Randomized (change au reboot)	NON	NON	NON	Sur plusieurs heures/jour selon veille VS reboot
Stable Privacy (calculée selon préfixe)	OUI	NON	NON	OUI
Complément Temporary	NON (quand session initiée par l'hôte)	NON (quand session initiée par l'hôte)	NON (quand session initiée par l'hôte)	Généralement moins d'un jour (quand session initiée par l'hôte)

Dans l'idéal il convient de laisser le comportement par défaut de l'OS pour les machines susceptibles de se connecter à l'extérieur de l'entreprise. Comportement variant généralement entre *Randomized* ou *Stable Privacy*, avec ou sans *temporary*.

Pour les autres machines, il est envisageable de désactiver intégralement SLAAC, en effet l'usage de DHCPv6 *stateful* et/ou la configuration manuelle (des serveurs par exemple) rendent ce mécanisme inutile. On suit alors la logique de réduction de surface d'attaque protocolaire et referme la porte.

Spécificité de la Link-Local

Bien que n'ayant qu'une portée locale, l'adresse de lien locale bénéficie également des trois différents modes de définition automatique évoqués plus haut.

La configuration suit généralement celle de l'adresse globale sur les OS grand public, peu de systèmes offrent une granularité de configuration spécifique selon les classes d'adresses.

Les systèmes orientés serveurs et réseau se basent cependant généralement sur EUI-64.

NE PAS DÉSACTIVER LA STACK IPV6

Si pour une raison particulière vous souhaitez éviter qu'un hôte échange en IPv6, ne désactivez pas sa pile IPv6. Préférez les options suivantes :

- Modifier la précédence pour prioriser IPv4 ;
- Désactiver SLAAC sur l'hôte et le bannir du DHCP le cas échéant ;
- Paramétrer le pare-feu de l'OS pour interdire tout trafic IPv6.

Si vous désactivez la pile IPv6, vous pouvez rencontrer des anomalies avec certains programmes. Windows requiert par exemple depuis plusieurs années de ne pas désactiver totalement IPv6 au risque de ne pas pouvoir faire fonctionner certains de ses composants couramment exploités. Sous Linux, la simple absence de la *loopback* ::1 peut aussi amener son lot de surprises. Les kernels récents laissent utiliser la loopback ::1 même avec la stack désactivée.

DÉSACTIVATION DES MÉCANISMES DE TRANSITION

Des mécanismes permettent aux hôtes d'échanger en IPv6 au travers de réseaux IPv4, notamment :

- TEREDO ;
- ISATAP ;
- 6to4.

Ces mécanismes n'ont plus d'intérêt et les 2 premiers ont même disparu. Il convient donc de les désactiver.

DÉSACTIVATION DES PROTOCOLES D'AUTO-DÉCOUVERTE

Il convient de désactiver les protocoles d'autodécouverte présents au sein des OS. S'ils sont utiles dans un environnement domestique, ils représentent un vrai risque en entreprise.

Il s'agira notamment de :

- SSDP (multi-OS, ff02::c – UDP 1900) et les adresses FF0X::C suivantes, en fonction de la portée ;
    - Node-local : FF01::C (ça ne sort même pas…) ;
    - Link-local : FF02::C ;
    - Site-local : FF05::C (déprécié) ;
    - Organization-local : FF08::C (déprécié) ;
    - Global : FF0E::C.
- mDNS (multi-OS, ff02:fb – UDP 5053);
- LLMNR (Windows, ff02::1:3 – UDP et TCP 5355).

Au-delà des attaques liées à ces protocoles, leur fonctionnement avec IPv6 diffère sur un point très particulier.

En IPv4, une machine a une seule IP. Si deux machines se mettent à discuter entre elles après avoir résolu leur nom via l'un de ces protocoles, on conserve tout de même la correspondance IP/machine via les logs DHCP typiquement.

En IPv6, ces protocoles permettent à des machines de se résoudre mutuellement via leur adresse de lien-local. (FE80::/10). Allez donc retrouver dans un log a qui correspondait une FE80:: ...

Ce comportement existe en production au sein même d'organisations n'ayant même pas déployé IPv6. Il suffit par exemple d'avoir un relayage SMTP entre 2 serveurs Microsoft Exchange situés sur le même segment réseau. Si les protocoles ci-dessus ne sont pas désactivés, vous verrez dans les entêtes de messagerie une remise via FE80. Heureusement que SMTP indique tout de même le nom d'hôte.

BLOCAGE DU TRAFIC LINK-LOCAL

À la maison l'adresse de lien local peut servir à discuter avec son NAS, une imprimante, un receveur chromecast/airplay, etc. après découverte via des protocoles susmentionnés. L'auto-enregistrement DNS sur son routeur domestique fera privilégier l'adresse globale.

Mais en entreprise, un hôte n'a aucune raison de faire autre chose que de l'ICMP (et les protocoles basés dessus comme MLD) via son adresse de lien local. Aussi est-il recommandé de bloquer au sein du pare-feu de l'OS tout trafic TCP et UDP dans les 2 directions. Mais gardez bien ICMP ouvert, comme déjà dit.

Attention, dans le cas de serveurs en cluster, il est tout à fait possible qu'une solution logicielle requérant que les machines soient dans le même segment réseau exploite les adresses de lien-local pour échanger des données, où tout simplement pour le *heartbeat*.

Prévoyez une exception pour DHCP et EAPOL 802.1x sur les systèmes les exploitant.

Pour les postes nomades, il est également intéressant d'ouvrir NAT-PMP (RFC 6886) et son successeur PCP V2 (RFC 6887) afin de permettre le fonctionnement des applications nécessitant de recevoir du trafic non sollicité. Typiquement, certains systèmes de conférence. Ces 2 protocoles permettent de demander à la passerelle l'ouverture d'un port, l'équivalent de l'autoredirection de port NAT44 en IPv4 via UPnP-IGD.

NAT-PMP exploitait initialement le port 5351 des 2 côtés, mais cela posait problème aux machines à la fois clientes et serveur comme lors d'un repartage de connexion. Aussi les clients ont migré vers le port 5350. PCP exploite également 5350 côté client et 5351 pour le serveur.

On retiendra donc UDP 5350 et 5351 en écoute et 5351 en destination.

Pour moins de contraintes, vous pouvez également choisir de ne bloquer que le trafic dans le sens entrant.

VPN

L'arrivée d'IPv6 au sein des réseaux domestiques peut présenter un risque pour des sessions VPN mal configurées. Une entreprise ne pratiquant pas le split-tunneling et annonçant la route 0.0.0.0/0 pourra en effet laisser l'hôte communiquer directement avec l'extérieur s'il arrive à résoudre des ressources DNS AAAA et que le pare-feu ne le bloque pas.

Résolution possible si le serveur DNS de l'entreprise répond aux requêtes AAAA, même au travers d'une connectivité IPv4, ou si le stack de l'hôte laisse des résolutions se faire via le DNS IPv6 communiqué localement à l'hôte en VPN.

Si vous exploitez le split-tunneling, assurez-vous de la concordance entre les règles IPv4 et IPv6.

De nombreux sites permettent de faire un IPv6 VPN *leak test*.

Note pour les VPN « grand public », ceux-ci supportent rarement IPv6, mais annoncent tout de même une route par défaut IPv6 afin d'envoyer le trafic vers un *blackhole* et d'éviter un *leak*. Vous pouvez faire la même chose et annoncer ::/0 sur votre VPN même si vous n'assurez pas de connectivité réelle.

CONFIGURATION D'OS DESKTOP

Cette section donne des exemples d'éléments de configuration.

Windows

Sous Windows, bien qu'il existe des commandes *netsh*, il est recommandé d'utiliser les *cmdlets powershell*.

La majorité de la configuration est listée ici :

https://docs.microsoft.com/en-us/powershell/module/nettcpip/set-netipv6protocol?

Certaines configurations se font également directement dans le registre, c'est par exemple le cas du DUID DHCP à l'emplacement HKLM\SYSTEM\CurrentControlSet\services\TCPIP6\Parameters\Dhcpv6DUID

0001 – DUID-TTL

0002 – DUID-EN

0003 – DUID-LL

Le DUID persistant apparait sous la même clé.

Linux

Cette section liste des configurations pour GNU/Linux.

Certaines configurations se font toujours au niveau du noyau, soit directement, soit via un outil tiers.

Les autres se font selon les packages en charge des fonctionnalités associées. L'écosystème GNU étant par définition même riche et ouvert, de nombreuses façons de procéder existent ; y compris au sein de la même distribution. Les documentations officielles des distributions ne sont d'ailleurs pas toujours alignées.

Les configurations pourront se faire selon les cas via :

- Des commandes ;
- L'édition de fichiers de configuration ;
- Des outils pseudographiques comme nmtui (pour Network Manager) ;

Voici des liens vers la documentation noyau :

https://www.kernel.org/doc/Documentation/networking/ipv6.txt

https://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt

https://github.com/torvalds/linux/blob/master/net/ipv6/Kconfig

Une reprise plus lisible https://sysctl-explorer.net/net/ipv6/

Network Manager

Network Manager est un outil issu du projet Gnome relativement courant pour gérer le réseau.

https://wiki.gnome.org/Projects/NetworkManager

https://developer.gnome.org/NetworkManager/stable/settings-ipv6.html

https://developer.gnome.org/NetworkManager/stable/nm-settings-ifcfg-rh.html

https://developer.gnome.org/NetworkManager/stable/nm-settings-keyfile.html

Utilitaire CLI nmcli https://developer.gnome.org/NetworkManager/stable/nmcli.html

Utilitaire pseudo graphique https://developer.gnome.org/NetworkManager/stable/nmtui.html

Systemd Networkd

systemd-networkd (réseau) et systemd-resolved (DNS) sont omniprésents mais pas forcément actifs. Veillez bien par exemple à désactiver la gestion globale (ou celles de certaines interfaces) par un autre daemon comme Network-Manager afin d'éviter les conflits avec Networkd. L'inverse est également vrai.

https://systemd.io/

https://www.freedesktop.org/software/systemd/man/resolvconf.html#

https://www.freedesktop.org/software/systemd/man/systemd-networkd.service.html#

https://www.freedesktop.org/software/systemd/man/systemd.network.html# (le plus important)

NETPLAN

Netplan n'est pas en soit un daemon de gestion directe, mais un outil d'abstraction présent chez canonical (Ubuntu). Il configure ensuite Network Manager ou Networkd.

https://netplan.io

https://netplan.io/reference/

Netplan semble cependant ne pas encore supporter DHCP-PD, un gros point noir pour certains usages (comme la fourniture de /64 à des pods d'hyperviseurs). En attendant, vous pouvez l'utiliser avec un override de systemd sur cet élément.

https://bugs.launchpad.net/netplan/+bug/1771886

Par distribution

La documentation de chaque distribution vous indiquera quel outil est en place par défaut. Dans la majorité des cas, le choix se fait entre systemd-networkd et Network-Manager. Conman et WICD ont par exemple disparu du paysage.

La documentation d'ArchLinux est, comme souvent, très complète. Ici un lien avec les éléments de config par type de network manager https://wiki.archlinux.org/title/Network_configuration#Network_managers

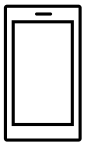
Voir également la section spéciale IPv6 https://wiki.archlinux.org/title/IPv6

Sous Ubuntu le man de netplan http://manpages.ubuntu.com/manpages/jammy/man5/netplan.5.html

Beaucoup d'éléments ici http://mirrors.deepspace6.net/Linux+IPv6-HOWTO/

et http://www.bieringer.de/linux/IPv6/.

MOBILE ET EMBARQUÉ



Les OS mobiles se retrouvent au sein d'un réseau d'entreprise sous différentes formes :

- Équipement embarqué (imprimante, système de réservation de salle, etc.) ;
- Flotte de smartphones appartenant à l'entreprise ;
- Smartphone personnel enrôlé (BYOD) ;
- Équipement non géré sur un réseau invité.

Android

Android est aujourd'hui l'acteur majoritaire sur ces segments, et il a un problème fâcheux, il ne supporte pas DHCPv6.

Étonnant ? Ce choix semble faire partie d'une stratégie de trust pour imposer la mise en œuvre de SLAAC. Les raisons sont indiquées dans la RFC 7934, DHCP ne fournit qu'une adresse et ne permet pas l'usage de *temporary* ce qui facilite le traçage. Ne disposer que d'une adresse ne permet pas de faire du *tethering*/partage de connexion …

Pourtant la demande est là, les problèmes évoqués n'ont pas de sens sur un réseau d'entreprise en Wifi. Le problème de partage de connexion n'a quant à lui de sens que derrière une liaison mobile type 3GPP.

Mais alors qui a rédigé cette RFC ? Des ingénieurs de Google et Apple, à commencer par Lorenzo Colitti.

Le problème est relaté depuis de nombreuses années :

https://www.techrepublic.com/article/androids-lack-of-dhcpv6-support-frustrates-enterprise-network-admins/

https://www.reddit.com/r/ipv6/comments/3wfpn2/i_am_getting_sick_of_lorenzos_attitude_to_ipv6/

https://www.nullzero.co.uk/android-does-not-support-dhcpv6-and-google-wont-fix-that/

https://issuetracker.google.com/issues/36949094

https://issuetracker.google.com/issues/36949085?pli=1

Que faire ? Demandez systématiquement le support de DHCPv6 dans vos appels d'offres pour des équipements. Que ça soit une flotte de smartphones ou des équipements embarqués.

Android est enrichi par les équipementiers bien au-delà du projet open-source de l'OS (AOSP), les constructeurs intègrent parfois un client DHCPv6. C'est typiquement le cas des imprimantes/copieurs sous Android, mais rarement des téléphones.

Comment tracer des équipements personnels enrôlés (BYOD) sous Android s'ils ne supportent pas DHCPv6 ? Les outils de suivi MDM (*Mobile Device Management*) pourraient apporter la réponse en traçant l'ensemble des adresses utilisées dès lors qu'elles font partie d'une liste de préfixes configurés. Par exemple, un /32 affecté par un RIR à l'entreprise. Ainsi le terminal n'est tracé que sur le réseau professionnel, sans pour autant recourir à DHCPv6.

La même chose est envisageable pour iOS, bien qu'il soit plus simple pour ces derniers de les faire se connecter à un SSID sans SLAAC et uniquement DHCPv6. Sans oublier de forcer via MDM l'usage de la véritable MAC pour ce SSID et non d'une MAC aléatoire. Les OS mobiles exploitent en effet depuis peu des adresses physiques aléatoires plus seulement lors de la recherche de SSID, mais également une fois connecté.

Concernant les réseaux invités, difficile de fournir ne serait-ce qu'un portail captif fonctionnel à un terminal en SLAAC changeant de *temporary-address* régulièrement…

Un portail captif centralisé fonctionnera avec DHCPv6, tant pis pour Android. La mise en place d'un collecteur NDPmon pourra permettre de suivre un terminal en SLAAC, mais ces solutions sont rares à l'heure actuelle.

Il est donc délicat, mais pas impossible de fournir une connectivité IPv6 SLAAC à des réseaux invités d'hôtel, d'hôpital, d'aéroport ou tout simplement au sein d'une organisation.

Autres OS

iOS supporte les 2 méthodes d'affectation d'adresse et ne pose pas de problème particulier en exploitation.

Pour d'autres équipements embarqués, il sera bon de demander à disposer du support de DHCPv6, mais également de pouvoir choisir les mécanismes d'autoaffectation d'adresse en cas d'usage de SLAAC. Typiquement, beaucoup d'équipements de type microcontrôleurs exploitent aujourd'hui du SLAAC uniquement EUI-64. Ce qui a l'inconvénient de permettre à un attaquant d'identifier la marque via l'adresse MAC, puisque cette dernière est incluse dans l'IPv6. Pensez donc à demander le support de stable privacy IPv6.

3. Transit

URPF

Unicast Reverse Path Forwarding (RFC 3704) permet d'éviter qu'un paquet dont l'adresse source ne correspond pas à une route connue dans le sens opposé ne puisse traverser un routeur ; limitant ainsi les risques de spoofing d'IP.

Plusieurs modes existent, selon que l'on se concentre sur la correspondance entre l'interface source et la meilleure route correspondante (strict), n'importe quelle route englobant l'adresse (*feasible*) ou que l'in cherche simplement à savoir si le routeur a au moins une route correspondante quel que soit l'interface (loose).

La RFC 8704 apporte des améliorations basées sur les informations BGP au mode *feasible*.

La mise en place doit se faire sur la portion périphérique du réseau, où il n'y a pas de risque d'asymétrie. Typiquement les cœurs de campus ou les routeurs de sortie de site. La configuration d'uRPF est généralement commune à IPv4 et IPv6.

Si vous routez du trafic multicast pensez également au RPF multicast.

PROTECTION DU CONTROL PLANE

Les paquets à destination du routeur lui-même ainsi que ceux disposant de certaines options d'entête entrainant une exception doivent remonter au control plane.

La RFC 6192 aborde les problématiques. L'utilisation du moteur de QoS afin de limiter le débit des trafics concernés à quelques Mbit/s permet de protéger le routeur d'une tentative de déni de service. Il conviendra évidemment d'investiguer immédiatement si la limite est atteinte ou proche de l'être. Cette sécurité ne faisant en effet pas de distinction entre trafics légitimes ou non.

De plus, le trafic destiné explicitement au routeur lui-même n'a pas de raison d'être fragmenté, vous pouvez le bloquer s'il est fragmenté.

SÉCURISATION OSPF

L'introduction d'OSPFv3 est l'occasion d'abandonner le MD5 pour exploiter IPsec afin de sécuriser les échanges. ESP doit être supporté, AH optionnellement (RFC 4552). Le tout en mode transport.

Note concernant les autres protocoles :

RIPng propose la même chose.

BGP n'étant pas spécifique à v6 suit un chemin différent au travers de l'initiative BGPsec qui vise à regrouper signature de l'origine de route et validation du chemin (AS-Path) de bout en bout. Initiative qui se focalise sur le routage public et ne semble pas comporter à l'heure actuelle de volet chiffrement et authentification

destiné aux réseaux corporatifs, basé sur une PKI privée ou tout simplement sur une implémentation manuelle des clés.

IS-IS ne reçoit pas d'évolution de ce côté-là, il est de toute façon agnostique d'IP.

4. Filtrage

Les préconisations de filtrage sont à appliquer au minimum en bordure du réseau, certaines règles peuvent être intégrées aux routeurs et pas seulement aux pare-feux, bien que l'aspect stateful soit cependant nécessaire pour une partie d'entre elles.

ICMP

Si l'on a tendance à fortement restreindre le trafic ICMPv4 autorisé, ICMPv6 nécessite une approche plus granulaire.

La RFC 4890 "Border Firewall Transit Policy" le rappelle et propose les ACL à mettre en œuvre. Vous les retrouverez ici :

Autoriser obligatoirement :

- Destination Unreachable (Type 1) - All codes ;
- Packet Too Big (Type 2) – requis pour PMTU discovery ;
- Time Exceeded (Type 3) - Code 0 only ;
- Parameter Problem (Type 4) - Codes 1 et 2 seulement.

Optionnellement :

- Time Exceeded (Type 3) - Code 1 ;
- Parameter Problem (Type 4) - Code 0.

Pour contrôler l'echo request et reply (généralement bloqués depuis internet) :

- Echo Request (Type 128) ;
- Echo Response (Type 129).

Sauf à faire usage de la mobilité IPv6, il est préconisé de bloquer :

- Home Agent Address Discovery Request (Type 144) ;
- Home Agent Address Discovery Reply (Type 145) ;
- Mobile Prefix Solicitation (Type 146) ;
- Mobile Prefix Advertisement (Type 147).

Les codes d'erreur et informatifs ICMPv6 non alloués par l'IANA devraient être bloqués sur le filtrage avec l'extérieur (internet, partenaire…) Leur blocage en interne est au choix des administrateurs.

Code d'erreur : types 5 à 99 et 102 à 126 inclus ainsi que le 150 (Seamoby).

Code informationnel : Types 154-199 et 202-254 inclus.

ICMPv6 prévoyait des mécanismes qui ne sont pas exploités en pratique, et donc à bloquer :

- Node information :
    - Node Information Query (Type 139) ;
    - Node Information Response (Type 140).
- Router Renumbering (Type 138) Ce message permet de changer le préfixe de toutes les interfaces configurées du routeur qui le reçoit. Peu de chance que vous l'utilisiez… à ne pas confondre avec la renumérotation de DHCPv6 et de Prefix Delegation.
- Codes expérimentaux (Types 100 – 101 et 200 – 201) ;
- Autres Types inutilisés (Types 127 et 255).

Si le Pare-feu est en mode L3 (routeur), il doit bloquer le transit (au-delà de la passerelle) des messages qui n'existent que dans le périmètre de l'adresse *Link-local* :

- L'ensemble de NDP y compris le reverse :
    - Router Solicitation (Type 133) ;
    - Router Advertisement (Type 134);
    - Neighbor Solicitation (Type 135) ;
    - Neighbor Advertisement (Type 136) ;
    - Redirect (Type 137) ;
    - Inverse Neighbor Discovery Solicitation (Type 141) ;
    - Inverse Neighbor Discovery Advertisement (Type 142).
- Le NDP multicast lié aux routeurs :
    - Multicast Router Advertisement (Type 151) ;
    - Multicast Router Solicitation (Type 152) ;
    - Multicast Router Termination (Type 153).
- Les messages liés à l'inexploitable protocole SeND
    - Certificate Path Solicitation (Type 148) ;
    - Certificate Path Advertisement (Type 149) ;
- Les messages MLDv1 et v2 (doivent cependant arriver via *Link-local* et avoir un hop-limit à 1)
    - Listener Query (Type 130) ;

- Listener Report (Type 131) ;
- Listener Done (Type 132) ;
- Listener Report v2 (Type 143).

À l'inverse, s'il fonctionne en bridge (L2), il doit autoriser les messages listés précédemment, en dehors de SeND (tant que ce dernier sera inexploitable).

Devraient être autorisés bien que toujours optionnels :

- Time Exceeded (Type 3) - Code 1 ;
- Parameter Problem (Type 4) - Code 0.

Même en L2 il convient de bloquer par sécurité le *Redirect* (Type 137). Sauf si celui-ci est utilisé, par exemple si un segment comporte 2 routeurs (un vers l'intérieur, l'autre vers l'extérieur) et un hôte dont la table de routage n'est pas adaptée.

Enfin, du DPI devra analyser le *payload* afin de détecter tout ICMPv6 mal formé, ou étant utilisé pour échanger des messages en créant une sorte de tunnel. Ceci à minima sur la bordure avec internet.

Le DPI pourra également permettre de bloquer un retour PMTU-D avec une valeur inférieure à 1280. Ce qui est impossible et risquerait de faire planter une pile IP mal développée.

MÉCANISMES DE TRANSITION

Si désactiver les mécanismes de transition sur les hôtes est une bonne pratique, les bloquer sur les équipements de filtrage l'est tout autant.

Ces règles sont à appliquer tantôt sur un réseau IPv4, tantôt IPv6, selon le sens de l'encapsulation. cf. RFC 7123.

Il convient donc de bloquer :

- IPv4 Protocole n°41 (6in4, 6to4, 6over4, 6rd, ISATAP) ;
- IPv4 Protocole 47 (GRE) sauf si utilisé ;
- Teredo :
  - UDPv4 destination port 3544 ;
  - Si DPI filtrer les paquets UDPv6 avec adresse Teredo (appartenant au préfixe 2001::/32) dans le *payload* ;
  - Requête DNS vers teredo.ipv6.microsoft.com. (via DPI et/ou sur les serveurs DNS).
- ISATAP :
  - Filtrer les requêtes DNS type A pour isatap.* (via DPI et/ou sur les serveurs DNS).
- 6to4 :
  - Paquets IPv4 proto 41 et sortant vers ou entrant depuis 192.88.99.0/24 :

    - Plus fin avec DPI, paquet IPv4 proto n°41 avec adresse 6to4 (appartenant au préfixe 2002::/16) dans le payload.
- 6over4 :
    - Paquet avec protocole 41 et destination 239.0.0.0/8 (bloque le NDP 6over4).
- Tunnel Broker / TSP (Tunnel Setup Protocol) :
    - TCPv4 et UDPv4 avec port de destination 3653 ;
    - Possibilité de préfiltrer avec IP proto n°41.
- AYIYA :
    - TCPv4 et UDPv4 avec port de destination 5072.

Lorsque possible soyez malin avec le DPI, filtrez d'abord sur le n° de protocole avant d'envoyer au moteur d'analyse afin d'économiser des ressources.

Tout déclenchement de l'une de ces règles à partir d'une machine située à l'intérieur du réseau doit entrainer une enquête afin de déterminer la cause de sa mauvaise configuration, particulièrement pour les mécanismes initiés par des hôtes comme Teredo et ISATAP.

Sur IPv6 vous pouvez bloquer 4rd, 4over6, etc.

BOGON PREFIXES ET ROUTES

En IPv4, il est anormal de voir certaines adresses, par exemple un paquet avec une adresse source en 127.0.0.5, ou une IP RFC1918 arriver depuis internet. Même chose en IPv6.

Il conviendra idéalement de bloquer les paquets concernés sur les pare-feux frontaux d'internet, mais également de filtrer toute annonce BGP comportant ces préfixes depuis internet ou un partenaire (sauf cas particulier)

- Blocs larges non alloués
    - 2d00::/8
    - 2e00::/7
    - 3000::/4
    - 4000::/2
    - 8000::/1
- 2001::/23 Réservé IETF
- 0::/96 Ancien préfixe de compatibilité IPv4
- ::ffff:0:0/96 Représentation d'IPv4
- 64:ff9b::/96 Well Known Prefixe NAT64
- 64:ff9b:1::/48 Plage réservée pour des plateformes NAT64 locales

- 100::/64 Préfixe RTBH (Remote triggered black hole filtering)
- 2001:2::/48 Benchmarking
- 2001:0DB8::/32 Documentation
- 5f00::/8 6bone, démantelé
- 2002::/16 6to4
- 3ffe::/16 ancien TEREDO
- 2001::/32 TEREDO
- 2001:10::/28 ORCHID Overlay Routable Cryptographic Hash Identifiers RFC 4843
- 2001:20::/28 ORCHID v2 RFC 7343
- 2001:3::/32 AMT, utilisé pour joindre un réseau multicast via un tunnel RFC 7450
- 2001:1::1/128 PCP, utilisé pour demander au pare-feu l'ouverture d'un port
- ff00::/8 Multicast
- fe00::/9 Ancien multicast
- fc00::/7 Unique Local Address
- fec0::/10 Ancien Site Local Address, déprécié
- fe80::/10 Link-local (sauf sur pare-feu bridge/L2)
- ::1/128 Loopback (à ne pas bloquer sur un pare-feu d'OS)
- ::/128 (0) Adresse non spécifiée
- ::/8 Plusieurs réservations dont les 2 précédentes

En plus des RFC, ne pas oublier les ressources de l'IANA

https://www.iana.org/assignments/iana-ipv6-special-registry/iana-ipv6-special-registry.xhtml

https://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xhtml

https://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xhtml

Il existe des listes auto-générées contenant ces préfixes ainsi que les préfixes non assignés par aucun RIR. La plus connue est https://www.team-cymru.org/Services/Bogons/fullbogons-ipv6.txt

Vous pouvez l'utiliser directement (Bogon + non alloués) ou ne garder que les informations concernant les adresses unicast routables 2000::/3

Notez que le bloc 2001:4:112::/48 AS112 permet de *blackholer* les trop nombreuses requêtes reverse DNS (PTR) associées à des IP privées. Le projet AS112 vise à soulager les root DNS, mené par l'ICANN il génère des statistiques à partir des requêtes. Vous ne devriez donc bloquer ce préfixe que si votre infrastructure DNS effectue le *blackholing* elle-même.

EXTENSION D'EN-TÊTE

IPv6 amène avec lui des extensions d'entête (EHs). Elles peuvent s'additionner et doivent toujours apparaitre sur le 1er fragment dans le cas d'un paquet fragmenté par l'hôte émetteur. Il faudra donc détruire tout 1er fragment qui ne contient pas l'entête IPv6 total.

L'un d'eux est le *HopByHop* (proto 0) qui demande un traitement à chaque routeur intermédiaire. Rendant de facto un DDoS possible notamment si l'équipement doit remonter le traitement au control plane. Plutôt que de détruire le paquet, il convient d'ignorer ce champ sur la bordure extérieure. (Ce comportement est autorisé à partir de la RFC 8200). Il reste nécessaire de l'activer pour faire du multicast ou du *jumbogram* en interne.

Une autre extension particulière est le *source routing*, *Routing Header* (proto 43), qui semble analogue à celle en IPv4. Pourtant il convient seulement de bloquer ses sous-éléments RHT 0 et RHT1 qui correspondent au *source routing* déprécié et à Nimrod. D'autres sont d'actualités comme le SRH (*Segment Routing Header*) de SRv6.

Surtout, ne bloquez pas l'extension indiquant que le paquet est fragmenté (Proto 44), et les 2 extensions liées à IPsec : Encapsulation/ESP (Proto 50) et Authentication/AH (Proto 51)

Le draft de RFC suivant détaille la politique recommandée (à partir de la section 3.3) https://datatracker.ietf.org/doc/html/draft-ietf-opsec-ipv6-eh-filtering

Ce qui est certain, c'est qu'il ne faut pas bêtement rejeter des paquets, car ils contiennent des extensions. Dans l'idéal il convient juste de filtrer certains types entre le réseau public et le réseau interne.

Vérifiez que vos fournisseurs d'accès ne détruisent pas les paquets contenant des extensions, et assurez-vous en interne du comportement de vos routeurs et pare-feux pour savoir dans quels cas un paquet remonte au control plane à cause d'extensions.

Révisez ces règles tous les 2 ans, certaines extensions risquent de disparaitre, d'autre d'arriver. À l'heure actuelle on trouve encore des équipements qui essayent de traiter les extensions même si elles ne sont pas dans l'ordre ou répétées, pouvant mener à des crashs, cf https://datatracker.ietf.org/doc/html/draft-kampanakis-6man-ipv6-eh-parsing-01

Enfin, prenez le temps de parcourir la RFC 7112 pour comprendre ce qu'il advient en cas d'enchainement à outrance d'extensions et de leur fragmentation. D'où la décision de forcer à les avoir tous dans le premier fragment.

POLITIQUE DE BANNISSEMENT

IPv6 offrant beaucoup d'adresses, il est nécessaire de changer la façon de gérer les bannissements temporaires.

Beaucoup de mécanismes se déclenchent pour bloquer un utilisateur temporairement après un certain nombre de tentatives d'authentifications infructueuses, ou imposer un captcha sur un site web après un fort trafic provenant d'une IP donnée. C'est typiquement le principe d'un outil comme Fail2Ban ou équivalent.

Une machine infectée, membre d'un réseau de botnet, aura toujours la même IPv4 tant que son FAI ne décide pas de la changer. Elle pourra en revanche utiliser les 2^64 IP qu'offre le /64 dont elle fait partie de façon aléatoire et avec des changements rapides.

De quoi rapidement saturer les listes de blocages, ou au contraire passer outre en changeant d'IP entre chaque tentative.

Pour ces raisons, il est important de toujours baser vos mécanismes de blocage sur le /64. Et idéalement de déclencher en complément un malus sur le /56 parent, permettant de gagner du temps en cas de tentative malicieuse d'un /64 voisin. Ce dernier appartenant probablement au même foyer.

Note : Cette situation s'applique bien évidemment au cas de figure inverse, redemander à un utilisateur de se réauthentifier au bout de 20 minutes, car son IPv6 temporaire a changé n'a pas de sens tant qu'il réside toujours dans le même /64.

6

Annexes et autres éléments

- URL ET IP LINK-LOCAL 108
- MULTI-PREFIXES 110
- CONTAINERS 110
  - Docker 110
  - Kubernetes 111
- SCADA 112
- NAT64 CHEZ LES OPÉRATEURS MOBILES 112
  - Découverte du service 112
  - Fonctionnement sur l'OS mobile 113
  - Partage de connexion 113
- PARTAGE DE PORTS SUR IPV4 113
- DRAFTS RFC POUR SAUVER IPV4 114
- EXEMPLES DE PROBLÈMES D'IMPLÉMENTATION IPV6 114
  - Non-décommissionnement des routes 114
  - Utilisation non contrôlée du préfixe de représentation IPv4 115
  - Champs de saisie incompatibles 115
- GASPILLAGE D'ADRESSES 116
- USAGE DE L'UNICITÉ DES ADRESSES POUR AUTRE CHOSE 116
- SRv6 117
- THREAD 117
- SELF-HOSTING AND RESIDENTIAL USE 118
  - Adressage et publication DNS 118
  - Ouverture de flux 119
  - Test de joignabilité 120
- OUVERTURE AUTOMATIQUE À L'INITIATIVE DE L'HÔTE 121
- ÉVOLUTION DES JEUX EN LIGNE 122
- QUE DEMANDER AUX OPÉRATEURS GRAND PUBLIC ? 122

Annexes

VI. ANNEXES

Les annexes contiennent des compléments techniques ainsi que des informations spécifiques liées à l'usage domestique.

URL ET IP LINK-LOCAL

L'adresse de lien local, dans la plage FE80::/10 a la particularité de disposer en sus d'un identifiant d'interface. Cet identifiant exploite le nom ou le numéro de l'interface selon le système.

Linux ajoute le nom de l'interface, par exemple %eth0, macOS de façon similaire avec %en0 typiquement.

En revanche Windows ajoute le n° d'interface, %1 et ainsi de suite.

```
C:\Users\JC>netsh interface ipv6 show address

Interface 1 : Loopback Pseudo-Interface 1

Addr Type  État DAD    Vie valide Pers. Fav. Adresse
---------  ---------- ---------- ---------- ----------------------
Autre      Préféré       infinite   infinite ::1

Interface 3 : Ethernet

Addr Type  État DAD    Vie valide Pers. Fav. Adresse
---------  ---------- ---------- ---------- ----------------------
Public     Préféré         29m51s      9m51s 2a01:cb00:83f5:[redacted]:45e1:e8c3:a8c9:739d
Autre      Préféré       infinite   infinite fe80::45e1:e8c3:a8c9:739d%3
```

Figure 11 ***netsh interface ipv6 show address***

Sous Windows la commande netsh interface ipv6 show address permet d'afficher les IPv6 assignées. En powershell les commandes Get-NetAdapter et Get-NetIPAddress affichent également l'information.

```
PS C:\WINDOWS\system32> Get-NetIPv6Protocol


DefaultHopLimit                 : 128
NeighborCacheLimit(Entries)     : 256
RouteCacheLimit(Entries)        : 4096
ReassemblyLimit(Bytes)          : 133913120
IcmpRedirects                   : Enabled
SourceRoutingBehavior           : DontForward
DhcpMediaSense                  : Enabled
MediaSenseEventLog              : Disabled
MldLevel                        : All
MldVersion                      : Version2
MulticastForwarding             : Disabled
GroupForwardedFragments         : Disabled
RandomizeIdentifiers            : Enabled
AddressMaskReply                : Disabled
UseTemporaryAddresses           : Enabled
MaxTemporaryDadAttempts         : 3
MaxTemporaryValidLifetime       : 7.00:00:00
MaxTemporaryPreferredLifetime   : 1.00:00:00
TemporaryRegenerateTime         : 00:00:05
MaxTemporaryDesyncTime          : 00:10:00
DeadGatewayDetection            : Enabled
```

Figure 12

Get-NetIPv6Protocol

En powershell Get-NetIPv6Protocol affiche la configuration générale de l'hôte relative à IPv6.

Dans le cadre d'un développement, ou simplement d'un usage personnel, on fait couramment usage d'URL avec l'IP.

En IPv6, une URL http sera par exemple sous la forme http://[AAAA:BBBB::H]:8080

Il est actuellement impossible d'utiliser les adresses de lien locales dans une URL sous un navigateur courant. La RFC 6874 force l'usage du caractère % avant l'identifiant d'interface dans une URL (%2, %eth0, etc.)

Hors l'usage du % est réservé par la communauté de maintien du HTML, WHATWG https://github.com/whatwg/url/issues/392

On doit donc toujours utiliser des adresses globales ou ULA dans un navigateur. Leur support fonctionne en revanche normalement dans d'autres contextes, comme un client SCP.

Ici le ticket demandant la réimplantation dans Firefox du support des URL Link-Local : https://bugzilla.mozilla.org/show_bug.cgi?id=700999

Et pour Chromium https://bugs.chromium.org/p/chromium/issues/detail?id=70762

À retenir, faites en sorte qu'une application à usage strictement local (comme une interface de commande industrielle) n'ai jamais besoin qu'un utilisateur s'y connecte via l'adresse de lien local dans un navigateur.

MULTI-PREFIXES

IPv6 permet d'exploiter simultanément plusieurs préfixes sur un réseau, si la mécanique fonctionne très bien à bas niveau sur le réseau et les hôtes, elle pose tout de même des problèmes.

Les flux vont quitter l'hôte à partir d'une adresse choisie par défaut ou de celle qui a le plus long préfixe commun avec la destination. Cet aspect non maitrisé complexifie la configuration.

Quelle adresse auto-enregistrer dans DNS localement ?

Les systèmes de sécurité qu'ils soient dans la couche d'accès (L2-NDP) ou de trafic (FW, ACL, ...) doivent pouvoir s'adapter à la volée.

Le multi-homing jusqu'aux hôtes peut rester intéressant dans une petite structure pour permettre des bascules lors d'un changement de fournisseurs. Pour les réseaux de taille intermédiaire, c'est un choix à opposer au couple (PI ou ULA) + NPTv6. Ce dernier étant pour rappel stateless et simple à configurer.

Note : Un mécanisme était prévu pour permettre à un client et un serveur d'échanger leurs différentes adresses via une extension d'entête et de basculer en cas de panne sans affecter la couche supérieure et donc sans timeout. Il s'agissait de Shim6. Ils pouvaient même s'authentifier via des adresses générées avec des mécanismes cryptographiques (CGA). En pratique, Shim6 est abandonné, on reste donc au royaume du timeout + établissement d'une nouvelle session en cas de perte d'un chemin, ou prise en compte par un protocole de couche supérieur. Coté modèle OSI on remarquera qu'IP n'est de toute façon jamais censé fournir ce type de mécanisme, c'est bien le rôle de TCP et maintenant de QUIC.

CONTAINERS

Docker

Docker exploite par défaut un bridge, une interface Docker0 et vient attacher des ports à des règles NAT44 pointant vers les ports publiés des containers. On créera des bridges additionnels pour isoler des containers entre eux.

Le mode overlay exploite quant à lui VxLAN et permet la communication inter hôte sans se soucier de la configuration du réseau sous -jacent (en plus d'offrir la possibilité de chiffrer, de simplifier l'administration SWARM, ...)

Difficile dès lors de faire de l'IPv6, Docker étant conçu pour fournir une abstraction totale du réseau (et du reste aussi).

Il existe plusieurs façons de contourner le problème :

- Utiliser le mode « macvlan », ce qui revient à exposer les containers au niveau 2 comme s'il s'agissait de VM. Chacun avec sa MAC. Peu pratique et surtout difficilement intégrable et exploitable dans l'écosystème.
- Plus récent mode IPvlan L2 expose les IP des containers derrière la même MAC que l'hôte via un mécanisme plus léger que le bridging classique.
- Dans sa version L3, IPvlan évacue totalement les risques de boucle et exploite des subnets IPv4 et préfixes IPv6. On doit mettre en œuvre les routes correspondantes sur les équipements réseau, chaque hôte comportant un ou plusieurs préfixes uniques.

En 2016, un développeur a initié un projet apportant du NAT66 en mode Bridge à Docker https://github.com/robbertkl/docker-ipv6nat

Il rappelle d'ailleurs que l'absence de NAT laisse tous les ports accessibles en IPv6, et qu'il faut donc penser à sécuriser les accès en amont.

Pour les gros déploiements, on recommandera le mode IPvlan L3.

A-t-on vraiment besoin d'IPv6 dans Docker ? Comme indiqué dans le document, il est intéressant de fournir le support IPv6 sur les frontaux (par exemple des containers SLB tels que traefik, hap, envoy, caddy,...). Au-delà le backend peut tout à fait rester en IPv4.

Kubernetes

Kubernetes expose par défaut une IP par Pod (regroupement de containers sur un hôte). L'hôte est nommé node. Attention au sens de Pod qui diffère ici de celui d'autres solutions. L'adresse est piochée dans le bloc assigné à la node.

L'adressage est donc exposé à plat sans overlay, facilitant la communication inter pods qu'ils soient dans la même node ou non. La vision de l'adressage est donc identique qu'on soit à l'intérieur ou à l'extérieur de la solution.

C'est donc fort similaire au mode IPvlan 3 de Docker.

La gestion du réseau incombe ensuite à l'une des nombreuses solutions tierces du marché (libre ou non).

Enfin, l'exposition depuis l'extérieur passe généralement par le combo Kubernetes services couplé à un load-balancer, ce dernier le plus souvent externe.

Docker a marqué IPv6 comme une fonctionnalité stable récemment, Kubernetes a suivi en béta dans la 1.21 et en stable dans la 1.23 https://kubernetes.io/docs/concepts/services-networking/dual-stack/

Depuis la mise à des disposition de ces 2 outils fin 2021, certains fournisseurs cloud ont activé IPv6 sur leurs offres de containers ainsi que sur d'autres services indirectement gérés par des containers.

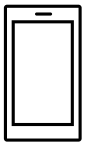
Souvenez-vous qu'à moins de faire du Headless Services, le load-balancing effectuera dans tous les cas une translation d'adresses.

Pour le trafic sortant vers internet, l'usage d'adresses IPv6 publiques évite le recours à la proxification ou au NAT.

SCADA

Un réseau SCADA est pour rappel un réseau fermé, on en retrouve souvent dans le monde industriel. L'intérêt à migrer vers IPv6 est ici relativement limité. La compatibilité des solutions industrielles avec le protocole mettra du temps à atteindre la maturité. N'hésitez cependant pas à mentionner cette compatibilité dans les questions optionnelles d'appels d'offres et n'envisagez sérieusement v6 que si l'ensemble de l'écosystème est compatible et testé. Si votre réseau SCADA est immense, car votre activité implique de très nombreux points de présence, IPv6 peut tout de même vous faire économiser de l'adressage IPv4. L'implémentation de 6LoWPAN sur du matériel embarqué peut également être un moteur. Mais à défaut vous pouvez toujours fonctionner en recouvrement/overlap d'adressage IPv4 avec le reste du SI puisque le principe même du SCADA est qu'il est fermé et non routé vers les autres ressources. Ce qui laisse le traitement de l'overlap à gérer uniquement sur les éléments d'interface entre le SI général et le SI SCADA, éléments qui sont là aussi, par sécurité, peu nombreux…

NAT64 CHEZ LES OPÉRATEURS MOBILES



Voyons ce qu'implique la mise en place de NAT64 entre les smartphones et internet.

Découverte du service

La section NAT64 du document explique son implémentation avec des postes de travail. Des méthodes permettent de fournir aux hôtes le préfixe NAT64, principalement sur les plateformes mobiles. Cela permet de s'assurer que les terminaux ont conscience de se trouver derrière un NAT64. Les principaux gains apportés par cette conscience sont de permettre à l'hôte de restaurer la validation DNSSEC ainsi que d'autoriser le fonctionnement d'adresse littérales non seulement dans la couche IP mais également lorsqu'un payload le transporte (par exemple SIP sans besoin d'ALG)

La RFC7051 aborde ce sujet, ainsi que le draft suivant :

https://tools.ietf.org/id/draft-ietf-v6ops-nat64-deployment-08.html

Une des solutions est l'enregistrement DNS ipv4only.arpa qui doit fournir une réponse connue basée sur un RFC. En l'occurrence un A record 192.0.0.170 ou 192.0.0.171.

Si la réponse est un enregistrement AAAA, par exemple 64:ff9b::192.0.0.170 (ici en notation décimale pour faciliter votre lecture, vous qui vous êtes aventuré en annexe), alors c'est qu'une plateforme de NAT64 exploitant le préfixe 64:ff9b::/96 est en production. Pour l'anecdote, Android fait la même chose avec l'enregistrement DNS ipv4.google.com.

Le protocole PCP (celui qui permet de s'ouvrir un port sur son routeur domestique) offre également la possibilité de demander l'existence préfixe NAT64.

La RFC évoque d'autres pistes, fournir l'information dans le Router Advertisement, ou via une option DHCPv6.

Enfin, la bonne vieille configuration des APN opérateurs sur les mobiles permet également de pousser le préfixe aux smartphones et est l'option la plus courante.

Les OS de PC ne supportent malheureusement aucune de ces méthodes sur leurs interfaces LAN. Laissant la main à DNS64 en entreprise pour encore longtemps…

Fonctionnement sur l'OS mobile

Pour conserver la compatibilité avec l'usage littéral d'adresses IPv4 ainsi que le support des signatures DNSsec, etc.

Si les 2 principaux OS mobiles mettent en œuvre des mécanismes afin de fournir la compatibilité IPV4, l'implémentation diffère radicalement.

Google Android se repose sur le réseau et 464 XLAT.

Le fichier clatd.conf contient les instructions relatives à la configuration CLAT du terminal, une adresse IPv6 faisant partie du /64 assigné au terminal est mappée (SIIT) avec une adresse IPv4 privée virtuelle. (Souvent 192.0.0.4). Le stack IP intercepte les paquets IPv4 et les translate en v6. Dans l'autre sens, dès qu'un paquet arrive sur l'adresse réservée au CLAT il est traduit en IPv4. Le développement peut être suivi ici https://android-review.googlesource.com/q/project:platform%252Fexternal%252Fandroid-clat

Apple iOS profite de l'aspect moins ouvert de son système pour traiter le problème dès les couches hautes. Ainsi, les frameworks (CFNetwork en bas, Cocoa URL loading system plus haut) ainsi que le moteur de rendu de navigation obligatoire WebKit convertissent directement toute adresse IPv4 en celle retournée par la synthèse du préfixe NAT64 avec ladite adresse. Ainsi aucun paquet IPv4 n'est jamais réellement créé. Ce mode est plus performant d'un point de vue énergétique.

Partage de connexion

Aussi appelé hotspot ou tethering, le partage implique de fournir un WiFi dual-stack à des hôtes n'ayant pas conscience que seul IPv6 est fourni au routeur, ici un smartphone.

464 XLAT à la rescousse, le téléphone va agir comme un CLAT en conjonction avec le NAT64 (PLAT) du réseau opérateur. Même fonctionnement sous Android et iOS :

Plutôt que de faire un NAT44 stateful suivit d'un NAT46, il va créer une règle de mapping stateless (SIIT) entre le réseau IPv4 du hotspot (/24 le plus souvent) et un morceau du /64 IPv6 dont il dispose. Ainsi pas besoin de table d'état et pas de changement de port côté téléphone. Le trafic subira ensuite le NAT64 stateful de l'opérateur pour redevenir IPv4 sur internet.

Souvenez-vous, l'entête IPv6 étant plus long, la 1ere passerelle pourra être amenée à fragmenter du trafic. Ne vous étonnez donc pas si l'upload d'un fichier s'avère ralenti par le CLAT. Les SoC ARM arrivant actuellement sur le marché offrent un support hardware de l'ensemble des opérations 464 XLAT afin d'éviter ces déconvenues.

PARTAGE DE PORTS SUR IPV4

Les approches Address + Port sont brièvement abordées dans la section mécanismes de transitions. (4rd et MAP-T/E pour les plus récentes). Les hôtes situés derrière un routeur domestique usant d'un tel mécanisme n'ont pas conscience que seule une partie des 65 535 ports est affectée à leur WAN.

Rien de très grave, sauf quand un programme requiert une ouverture de port (UPnP, NAT-PMP) et que le routeur oublie qu'il n'a pas accès à l'ensemble des ports lui aussi. Il retournera parfois un port en dehors de la plage affectée à l'abonné. Ce qui revient à jouer à la roulette russe avec certains échanges P2P.

La RFC 6269 aborde les problèmes liés au partage, dont celui évoqué ici qui se produit chez des opérateurs à l'implémentation un peu trop rapide et non rigoureuse.

Un opérateur ne devrait pas partager les IP entre plus de 16 clients.

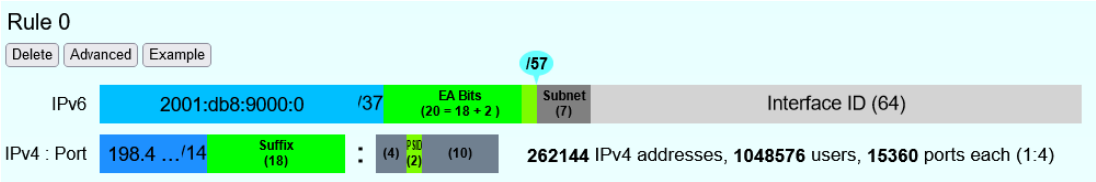


Figure 13

Simulation de partage A+P via MAP

Dans cet exemple on partage des IPv4 entre 4 abonnés http://map46.cisco.com/MAP.php

DRAFTS RFC POUR SAUVER IPV4

Certains font tout pour tenter de prolonger la vie d'IPv4 en trouvant le moyen d'augmenter ses capacités d'adressage.

Plusieurs drafts ont existé, les plus récents semblent être :

https://www.ietf.org/id/draft-schoen-intarea-unicast-0-00.html

https://www.ietf.org/id/draft-schoen-intarea-unicast-127-00.html

https://www.ietf.org/id/draft-schoen-intarea-unicast-240-00.html

Inutile de dire que mettre à jour toutes les piles IP des OS de PC, de smartphone, de routeurs, etc. pour supporter ces changements demanderait bien plus d'effort que de passer à IPv6.

Le 240/4 est tout de même supporté officiellement chez au moins 2 grands constructeurs ainsi que chez Google GCP.

Dans un autre registre, la proposition EzIP en est à sa neuvième itération, si vous aimez le NAT lisez là :

https://datatracker.ietf.org/doc/html/draft-chen-ati-adaptive-ipv4-address-space-09

EXEMPLES DE PROBLÈMES D'IMPLÉMENTATION IPV6

Ici quelques exemples de bugs d'implémentations rencontrés lors de l'usage d'IPv6.

Non-décommissionnement des routes

Sous IPv4, on a une connectivité ou on n'en a pas, tout simplement. Dès lors qu'on passe en dual-stack comment s'assurer de la disponibilité de la connectivité IPv6 ? *Happy Eyeballs* peut dépanner, mais génère un délai et n'est pas conçu pour pallier une absence prolongée de connectivité IPv6.

Exemple, les box opérateurs avec secours en 4G n'ont souvent que de l'IPv4 sur le lien de secours. Lorsque le secours se déclenche, certains routeurs continuent d'envoyer des RA pour se déclarer routeur par défaut

et annoncer un préfixe IPv6 qui n'est plus du tout exploitable puisque la connectivité IPv6 est totalement rompue.

Ce problème apparait également lors des renumérotations. En IPv4 le NAT44 rend le réseau local indépendant de l'adressage WAN. Avec IPv6 ça n'est plus le cas (sauf à user du combo ULA + NPTv6). Ainsi lors des rares occasions où un FAI grand public renumérote son réseau, les clients peuvent subir une perte de connectivité temporaire le temps que les informations des anciens RA disparaissent.

La section 6.3.5 de la RFC 4861 rappelle que les hôtes doivent purger le préfixe si le *timer* expire ou si le routeur ne s'annonce plus comme default. Mais dans notre cas le routeur existe toujours et est toujours joignable via son adresse de lien local. Les hôtes vont donc attendre que le *timer* du préfixe expire pour supprimer la ou les adresses d'interface exploitant l'ancien préfixe. Les terminaux enverront alors toujours les paquets au routeur, mais avec une adresse source appartenant à l'ancien préfixe… Difficile d'espérer une réponse, et sans réglage agressif des *timers* ça peut facilement durer 1800 secondes soit une demi-heure. On ne peut que recommander aux opérateurs d'abaisser les délais d'expiration à une valeur inférieure à la minute.

Les personnes souhaitant jouer avec le multihoming IPv6 rencontreront vite des problématiques similaires de bascule.

Utilisation non contrôlée du préfixe de représentation IPv4

Afin de simplifier votre SI, vous avez décidé de n'utiliser que la notation IPv6 dans votre CMDB. Ainsi vous utilisez le préfixe ::ffff:0:0/96 pour indiquer une IPv4 dans vos scripts de configuration, etc.

Étrangement, votre script crée une règle/ACL, mais est ensuite incapable de la retrouver dans son contrôle et clôture son exécution sur un échec. Pourtant le flux concerné fonctionne.

En fait le système configuré a simplement décidé de retraduire la notation d'une IPv4 avec ::ffff:0:0/96 vers la notation IPv4 classique.

Ce bug a par exemple existé chez F5 lors de la création des règles https://cdn.f5.com/product/bugtracker/ID669888.html

Pratique, mais à prendre en considération dans les automatisations.

```
PS C:\WINDOWS\system32> ping ::ffff:c0a8:1

Envoi d'une requête 'Ping'  192.168.0.1 avec 32 octets de données :
Réponse de 192.168.0.1 : octets=32 temps<1ms TTL=64
Réponse de 192.168.0.1 : octets=32 temps<1ms TTL=64
```

Figure 14 ***On retrouve cette facilité de conversion dans des utilitaires courants, ici le ping Windows***

Champs de saisie incompatibles

Lors de la saisie d'une IPv6, les contrôles de champ sont parfois inadaptés. On peut retrouver les bugs suivants dans des environnements graphiques et, plus rarement, en ligne de commande.

Un champ totalement incompatible rejettera une adresse qui n'est pas sous la forme IPv4, mais des subtilités peuvent passer au travers des contrôles. Par exemple il arrive que les [ ] utilisés pour séparer l'adresse du port ne soient pas pris en compte. Ainsi la saisie de [A123:8BBB::2D5E]:8443 pourra se transformer dans le logiciel en A123:8BBB::2D5E:8443

GASPILLAGE D'ADRESSES

Oui, il y'a beaucoup d'adresses IPv6 ! Internet est rempli de savants calculs pour nous expliquer que 2^E128 équivaut à 3,4 * 10^E38 adresses, soit 667 sextillions par m^2 de surface terrestre. Nombre d'ailleurs proche de la constante d'Avogadro font remarquer d'autres (~6,02*10^E23).

Alors forcément, avec des phrases comme « on pourrait adresser chaque grain de sable jusqu'à 2km de profondeur » on se dit qu'on peut faire n'importe quoi.

Pourtant, une adresse IPv6 n'est pas une plaque d'immatriculation ou un numéro de téléphone. Elle suit la plupart du temps une construction basée sur un préfixe /64. De plus ces préfixes font partie d'un sous-ensemble réservé pour le routage global et affecté par le gestionnaire continental (RIR).

Ainsi, une grande entreprise qui obtient un /29 peut en toute logique créer 34 MDs de réseaux. Si on compte maintenant le nombre de sites en /48, ça en fait 524 288.

La poste indienne avec ses 160 000 bureaux de poste est donc tranquille… Enfin, sauf si quelqu'un décide que le WiFi invité et le projet smart building IoT ont chacun besoin de leur /48 par site, car la sécurité/politique/délégation/organisation interne (rayez les mentions inutiles) le requiert. Ça vous fera sourire, mais regardez en arrière avec IPv4, ce genre de raisonnement est bien trop fréquent.

USAGE DE L'UNICITÉ DES ADRESSES POUR AUTRE CHOSE

L'immense nombre d'adresses possibles a donné à des ingénieurs des idées de détournement basées sur l'identification précise de l'utilisateur et/ou de la ressource à accéder.

Voici quelques exemples :

- Attribuer des adresses IPv6 différentes à un serveur pour chaque client s'y connectant ? En cas de DDoS on peut bloquer uniquement l'adresse concernée sans affecter les autres clients utilisant la même machine. Le futur ami de RTBH ?
- Inclure une authentification directement dans l'adresse qui évolue dans le temps ? C'est le principe du TOTP d'IPv6 que fournit ce projet de serveur SSH dont l'IP change toutes les 30 secondes. https://github.com/mikroskeem/tosh
- Assigner directement des données comme des morceaux de vidéo en streaming et non plus le serveur qui les héberge, c'est par exemple l'objet du brevet suivant https://patents.justia.com/patent/11134052

Affecter un nombre immense d'adresses à chaque serveur peut rapidement saturer le cache NDP.

Ces usages restent possibles si l'on affecte directement un préfixe /64 au serveur, comme décrit la RFC 8273. C'est d'ailleurs ce qu'on fait déjà avec les conteneurs comme décrit un peu plus haut avec l'exemple des nodes Kubernetes. Ces /64 pourraient tout aussi bien être confiés à des Load Balancers.

Pour les systèmes avec changement régulier d'adresse ça implique de remonter une session à chaque fois, mais après tout, ça ne serait jamais plus qu'un nouvel usage du 0-RTT de QUIC par exemple…

SRv6

Le *Segment Routing* se répand rapidement chez les opérateurs et les GAFAM. À l'heure actuelle SR-MPLS domine les déploiements, mais les projections montrent que son pendant basé sur un simple data-plane IPV6 prendra le relai d'ici une poignée de semestres.

La maitrise du transport IPv6 et de l'IGP dominant ce secteur, IS-IS ; seront rapidement incontournables pour tout réseau de grande taille.

Outre les apports de SR en termes de topologie dynamique et adaptive, de télémétrie ainsi que la possibilité d'inclure des champs adaptés à des traitements de service (groupe de sécurité, identifiant d'application…) au sein de l'en-tête SRH, il est indéniable qu'il sera le 1er à remplacer l'entièreté des millefeuilles existants.

Ainsi au-delà du backbone, il remplacera probablement le couple VxLAN + EVPN en datacenter, ainsi que les solutions fermées de SDN Campus. Offrant un véritable service bout en bout sans compromis.

Les champs de service permettront alors une véritable application de stratégie dynamique, non plus basée sur des plages d'adresses etc., mais bien sur les informations complémentaires. Tout ceci sans technologie propriétaire, mais pouvant être interprété par des équipements de services aussi bien physiques que virtuels (VNF).

Plus tard l'ajout même de ces champs de fera probablement au sein de l'hôte lui-même, afin de pouvoir y reporter des informations fournies directement par l'applicatif. Laissant toujours au routeur de 1er saut la charge d'ajouter le chemin retenu. Côté serveur, on a pu voir l'intégration de terminaison VTEP (VxLAN, et parfois GENEVE) descendre des Switchs Top of Rack vers les serveurs eux-mêmes. De la même façon on assistera probablement à un traitement intégral SRv6 sur les serveurs, y compris à terme pour la gestion de la topologie grâce notamment à l'arrivée des Network Processeur Unit (NPU, à ne pas confondre avec Neural Processor Unit) et des IPU (Infrastructure Processing Unit)

Les constructeurs poussent donc actuellement les entreprises à la transition vers SR-MPLS pour mieux revenir un peu plus tard avec SRv6. Nous devrions cependant bientôt assister à des migrations directes vers SRv6 sur des réseaux d'entreprises et plus seulement d'opérateurs.

THREAD

THREAD est un protocole réseau conçu pour l'IoT par le Thread Group https://www.threadgroup.org/

Il a vocation à fournir un système d'échange en mesh entre les équipements domotiques en se basant sur 6LoWPAN. Il exploite donc IPv6 avec des notions de scope, de nodes routeurs et enfants. Cf la page du projet open source OpenThread https://openthread.io/guides/thread-primer/ipv6-addressing

Le projet « Matter » de connectivité standardisée pour la domotique l'exploite d'ailleurs.

SELF-HOSTING AND RESIDENTIAL USE

La mise en œuvre d'IPv6 sur un simple réseau domestique permet d'appréhender facilement une partie des différences avec IPv4. Nous verrons notamment ici l'exposition de services à l'extérieur.

Bien que ces exemples puissent être utilisés dans une petite structure, nous rappelons qu'il est primordial de disposer d'une véritable couche de filtrage et d'analyse en entrée d'internet sur un système en production, même petit.

Adressage et publication DNS

Les opérateurs grand public ne fournissent la plupart du temps qu'un /64 sans possibilité d'exploiter les autres préfixes assignés au routeur (souvent dans un /56).

Il est également impossible de s'assurer de la stabilité du préfixe dans le temps (sauf engagement contractuel).

L'adresse de chaque machine à exposer est donc à publier indépendamment, là où on publiait l'adresse WAN IPv4 et jouait avec les ports du NAT44.

On commencera par s'assurer que les machines exploitent une adresse stable (Typiquement basée sur la MAC ou *stable privacy*, ce qui est préférable).

On aura ensuite recours à un service de DNS dynamique IPv6, par exemple Dynu, DuckDNS, etc.

Plusieurs méthodes permettent de remonter le couple IP/enregistrement DNS AAAA directement sur une machine :

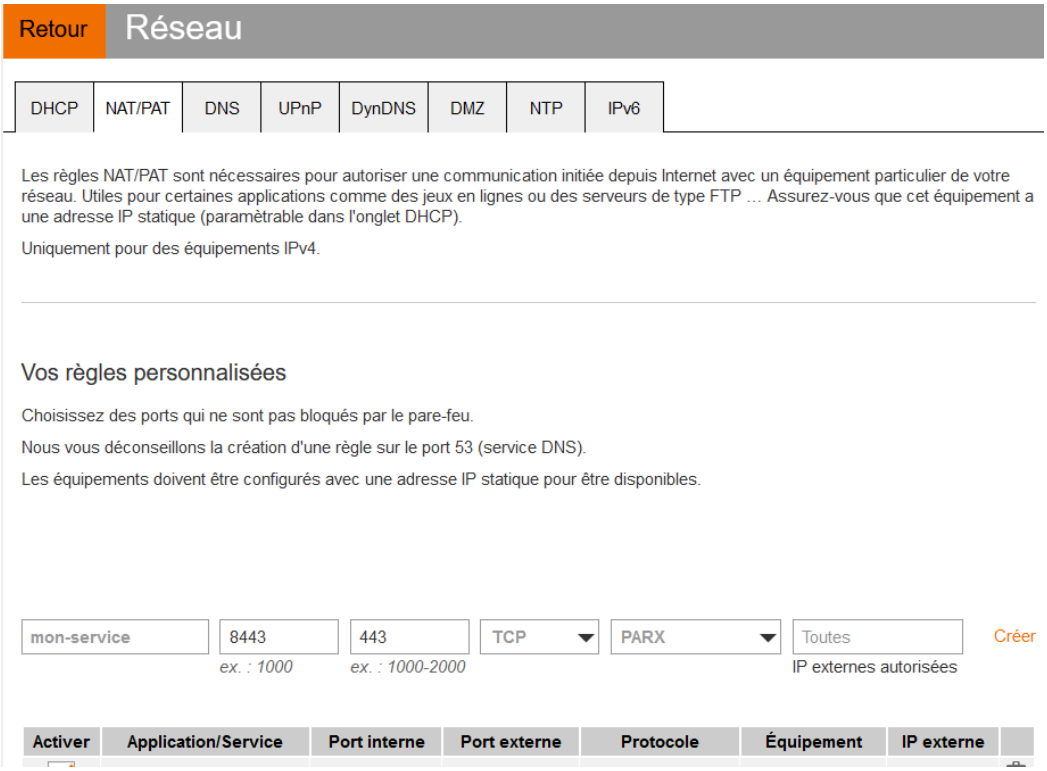
- Script de requête avec auto-déduction de l'adresse par le serveur API du service DNS
- Script récupérant l'IP publique via une API tierce (ex api6.ipify.org) puis envoyant au service DNS
- Script récupérant l'IP depuis l'interface système (attention à bien prendre la publique stable)
- Agent logiciel du service

Il est également possible de se baser sur un routeur et ses infos NDP, mais on sort alors de l'utilisation simple du matériel opérateur.

Ouverture de flux

La fourniture d'un pare-feu en IPv6 fait l'objet d'un traitement inégal selon les opérateurs. Certains l'ont implémenté très tardivement en mode tout ou rien, d'autres offrent une granularité analogue à ce qu'on trouve en IPv4.

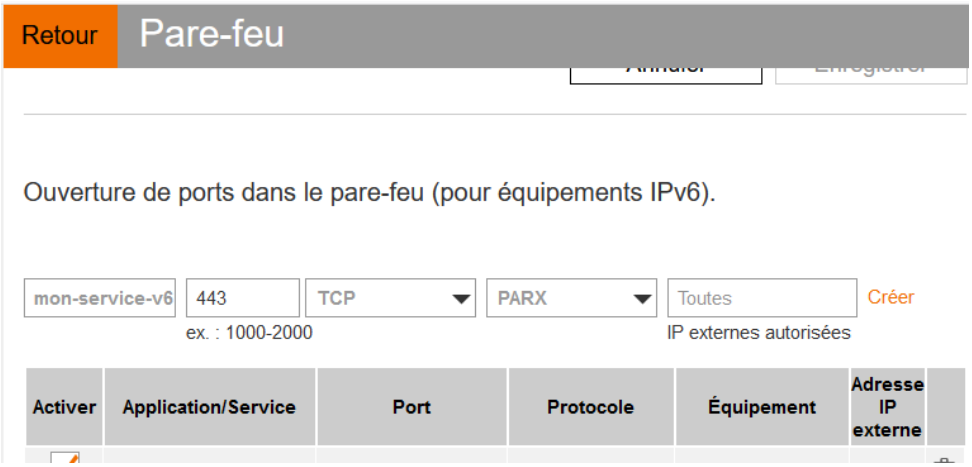
Prenons l'exemple d'une LiveBox 4 Orange. En IPv4 l'ouverture se fait dans la section réseau.



En IPv4 on aura l'habitude d'avoir des ports différents entre l'interne et l'externe, ce qui évite de devoir changer les ports sur les serveurs, mais empêche de publier plusieurs machines sur le même port externe (à moins de passer au travers d'un reverse proxy intermédiaire)

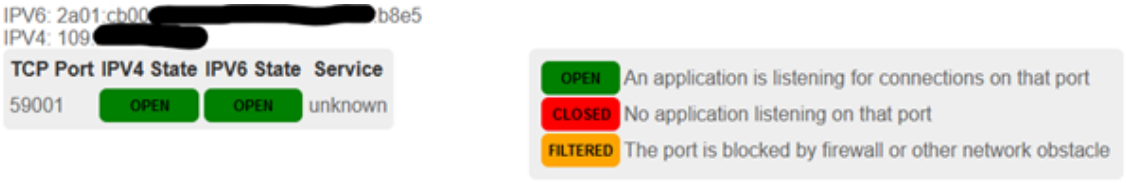
En IPv6 la situation est l'exact opposé, chaque machine a son IP et donc ses 65535 ports, mais on doit forcément utiliser le même n° de port en interne et en externe en raison de l'absence de translation (PAT).

Chez Orange la configuration se trouve dans la section pare-feu.



Test de joignabilité

Le test peut se faire via un scanneur de port en ligne tel que http://www.ipv6scanner.com/



Ici tout est en ordre, dans le cas contraire souvenez-vous qu'Happy-Eyeballs V2 rebasculera la connexion en IPv4 en l'absence de réponse v6.

Certains fournisseurs ne proposent pas de pare-feu fin, c'est le cas de Free qui s'est longtemps retranché derrière le fait que la RFC sur les CPE recommande, mais n'impose pas de filtrage stateful. Free ne propose de pare-feu IPv6 que depuis 2020 et celui-ci est très léger. Nombreux sont les clients à demander la mise en œuvre d'un véritable pare-feu sur le bugtracker https://dev.freebox.fr/bugs/index.php?string=ipv6&project=9&type%5B%5D=&sev%5B%5D=&pri%5B%5D=&due%5B%5D=&reported%5B%5D=&cat%5B%5D=&status%5B%5D=open&opened=&dev=&closed=&duedatefrom=&duedateto=&changedfrom=&changedto=&openedfrom=&openedto=&closedfrom=&closedto=&do=index

OUVERTURE AUTOMATIQUE À L'INITIATIVE DE L'HÔTE

Discuté plus haut dans le document, PCP V2 permet une ouverture de port par le routeur à la demande d'une application. Généralement pour des usages P2P.

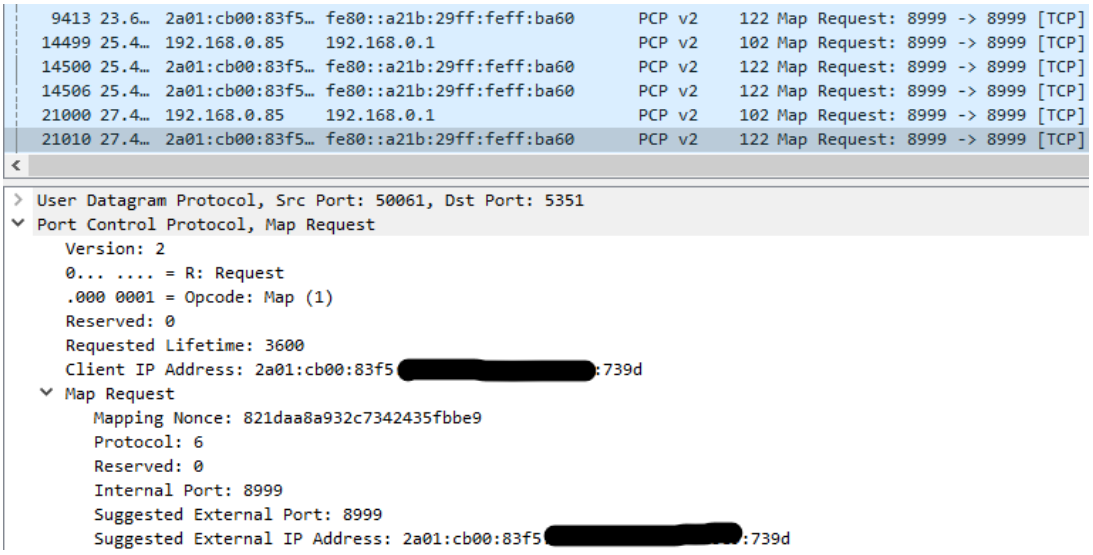


Figure 15

Wireshark PCP v2 IPv6

Exemple de capture Wireshark de PCP V2 avec le filtre « udp.port eq 5351 ». On retrouve des demandes d'ouverture en IPv4 et IPv6.

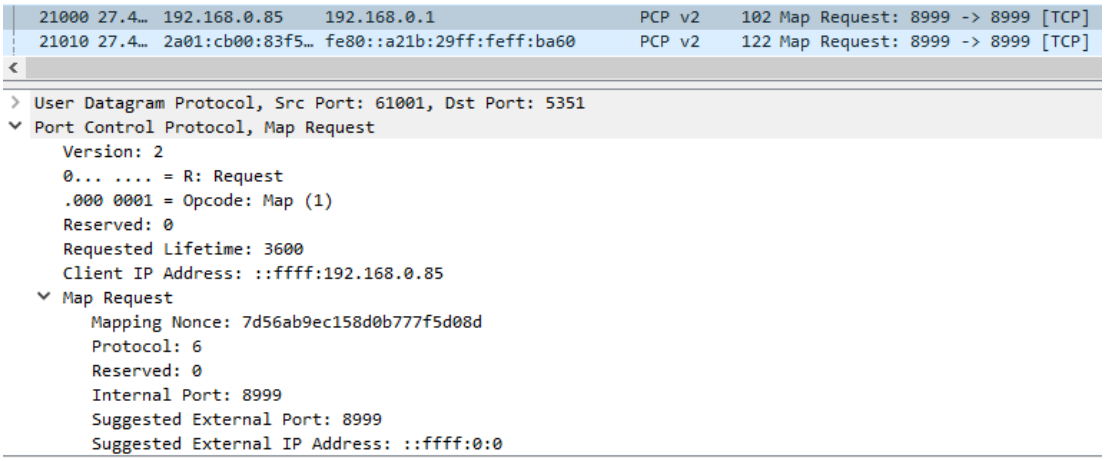


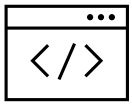
Figure 16

Wireshark PCP v2 IPv4

Notez que la version IPv4 de la requête voit son IP interne écrite sous la forme d'une IPv6 représentant une IPv4, et que l'adresse WAN est mise à 0.0.0.0 puisque c'est de toute façon la WAN IPv4 du routeur (là aussi sous la même forme avec ::ffff:)

On est bien loin du lourd XML d'UPnP-IGD requérant l'échange de nombreux paquets.

ÉVOLUTION DES JEUX EN LIGNE



À l'heure actuelle le secteur du jeu vidéo n'intègre pas IPv6 dans ses communications entre joueurs et serveurs. L'impact des CG-NAT IPv4 et autres mécanismes d'IPv4aaS pourrait pourtant être évité avec un effort des studios.

Les jeux où la partie est gérée par un serveur dédié devraient passer leur serveur en dual stack et privilégier IPv6 lorsqu'il est disponible.

Pour les jeux en P2P où l'un des joueurs héberge la partie, il serait bon d'inclure dans les algorithmes de choix de l'hôte un élément de pondération basé sur la disponibilité du dual-stack si par exemple au moins 40% des joueurs de la partie ont IPv6 actif.

QUE DEMANDER AUX OPÉRATEURS GRAND PUBLIC ?

Les régulateurs devraient demander aux opérateurs de mettre en œuvre les mécanismes suivants en complément d'IPv6 sur les connexions fixes (xDSL, FTTh, 4/5G fixe, Low Orbit SAT…) :

- Un pare-feu à réglage granulaire, basé dynamiquement sur le suivi de l'ensemble des adresses de chaque hôte et la correspondance avec l'adresse MAC dans la table NDP.
- La fourniture d'au moins 2 préfixes /60 en plus du préfixe par défaut sur simple requête DHCPv6-PD à partir d'un autre routeur. Il serait d'ailleurs de bon usage que les opérateurs offrent également la possibilité d'implémenter des routes statiques à minima sur une portion IPv4 RFC1918 documentée de leur part.
- Une gestion des renumérotations IPv6 évitant des blackouts, typiquement en ajustant les timers RA.
- Une information claire dans l'interface du modem sur le mode d'accès IPv4 et IPv6, ainsi que la plage port mappée dans le cas d'une approche de partage d'IPv4 A+P (4rd, MAP-x,…)
- La possibilité d'utiliser un routeur tiers à l'heure où les mécanismes A+P de partage IPv4 rendent le routeur de l'opérateur encore plus captif.

Sur les connexions mobiles, il serait pertinent de travailler à faire fonctionner PCP v2 sur les terminaux, notamment sur l'APN de partage de connexion. Ceci permettrait aux clients de profiter pleinement du bout en bout IPv6 lors d'usage hotspot. Le support de DCHP-PD serait également pratique pour les cas spécifiques de partage multi réseaux avec plusieurs /64.

À propos de ce document

Ce document a été rédigé dans le cadre de la task-force IPv6, son rédacteur principal Jean-Charles BISECCO tient à remercier l'ensemble de ses membres pour leurs apports et corrections.

AIDEZ-NOUS À FAIRE ÉVOLUER CE DOCUMENT

Ce document a vocation à être mis à jour dans la durée, vos retours sont précieux.

Vous pouvez nous transmettre vos idées pour enrichir le guide ainsi que vos retours d'expériences :

- Rejoindre la task-force IPv6
- Contacter l'auteur à l'adresse IPv6@arcep.fr/ IPv6@jclb.net

La dernière version est disponible à l'adresse suivante :

https://www.arcep.fr/la-regulation/grands-dossiers-internet-et-numerique/lipv6/task-force-ipv6.html

LICENCE

TRADUCTIONS

Ce document est initialement publié en Français et Anglais. Nous sommes ouverts à des traductions dans d'autres langues afin de faciliter le déploiement d'IPv6 dans un maximum de lieux.

Les traductions pourront être ajoutées à la liste officielle.

L'accès aux deltas entre cette version et les futures sera fourni aux traducteurs.

Task -force IPv6 ARCEP