

IPv6

Hainaut Patrick 2017

But de cette présentation

- Vous permettre de connaître et comprendre IPv6

Pénurie d'adresses IPv4

- **Histoire:**

- Les premiers RFC (Request For Comments) traitant d'un protocole Internet datent de 1977 et il était déjà fait mention d'une version 4 en 1979 dans l'IEN 123
- C'est néanmoins le RFC 790 de septembre 1981 qui décrit vraiment pour la première fois la structure sur 32 bits et la répartition des adresses telles que nous les connaissons aujourd'hui avec les classes A, B et C
- Pour mémoire, la classe A contient 126 réseaux, dont plus d'un tiers sont directement affectés à des sociétés ou des organisations
- À l'époque, seules des classes A (et encore, seulement les 44 premières) étaient assignées à des réseaux et il était inimaginable que ce réseau s'étende un jour au monde entier en offrant la connectivité à chacun des habitants de la planète

Pénurie d'adresses IPv4

- **Maintenant:**
 - La situation a bien changé car on considère qu'il ne reste plus que 2 % de l'espace à allouer
 - En fait, la situation varie un peu d'une région à l'autre, car si l'IANA (Internet Assigned Numbers Authority) a alloué toutes les plages (les 256 /8) aux différents registres (épuisement intervenu en février 2011), ceux-ci n'ont pas obligatoirement distribué toutes les adresses transmises par l'IANA

Pénurie d'adresses IPv4

- **Maintenant:**
 - Sur les cinq registres régionaux, quatre sont maintenant arrivés en phase de pénurie :
 - En avril 2011 pour l'APNIC (Asia-Pacific Network Information Center),
 - En septembre 2012 pour le RIPE (Reseaux IP Europeens Network Coordination Centre),
 - En juin 2014 pour le LACNIC (Latin America and the Caribbean Network Information Center),
 - En septembre 2015 pour l'ARIN (American Registry for Internet Numbers).
 - Seul le continent africain géré par le registre AFRINIC (African Network Interformation Center) dispose encore de plages d'adresses non allouées. La situation de pénurie devrait y être déclarée autour de juillet 2018 selon les dernières simulations (données de septembre 2016)

Pénurie d'adresses IPv4

- **Pourquoi:**
 - Parmi les raisons ayant contribué à l'accroissement de la demande en adresse IP nous pouvons citer :
 - Le nombre croissant et exponentiel d'abonnés à un fournisseur d'accès Internet (croissance spectaculaire dans les pays dits émergents), abonnés nécessitant à chaque fois au moins une adresse
 - La pérennité des connexions car, contrairement au temps de la connexion par RTC (Réseau Téléphonique Commuté), les connexions Internet sont souvent ininterrompues et donc la remise au pot commun lors de l'expiration des baux DHCP des adresses inutilisées se fait beaucoup plus rare
 - L'offre des opérateurs de disposer d'une adresse fixe pour chaque box limite les possibilités de réutilisation des adresses dormantes

Pénurie d'adresses IPv4

- **Pourquoi:**
 - Parmi les raisons ayant contribué à l'accroissement de la demande en adresse IP nous pouvons citer :
 - La multiplication des mobiles et tablettes connectés en permanence à Internet fait qu'une personne peut consommer à elle seule plusieurs adresses IP
 - L'apparition et la multiplication des objets connectés (IoT - Internet of Things) va probablement faire exploser notre consommation d'adresses IP, même si tous les objets n'auront pas forcément une adresse IP publique
 - Les estimations du nombre d'objets connectés en 2020 varient de 20 à 50 milliards pour s'établir autour de 500 milliards en 2030

Pénurie d'adresses IPv4

- **Pourquoi:**
 - Parmi les raisons ayant contribué à l'accroissement de la demande en adresse IP nous pouvons citer :
 - L'allocation par blocs des adresses IPv4 est particulièrement gaspilleuse d'adresses IP, de même que la notion d'adresse de réseau et de broadcast
 - Ainsi, quand un FAI (Fournisseur d'Accès Internet) attribue à un client deux adresses (l'une pour le firewall du client, l'autre pour le routeur du FAI par exemple), il lui attribue généralement un sous-réseau de 4 adresses minimum puisque l'une d'entre elles doit représenter le réseau et l'autre l'adresse de diffusion (broadcast)

Pénurie d'adresses IPv4

- **Les attributions d'adresses IPv4 en situation de pénurie:**
 - La pénurie déclarée sur 4 des 5 registres régionaux entraîne automatiquement des mesures de restrictions pour les attributions de nouvelles plages d'adresses
 - Ces restrictions sont variables d'un registre à l'autre mais le principe général est que lorsque l'on commence à puiser dans le dernier /8, il n'est plus attribué qu'un seul et unique /22 par demandeur
 - Et certains registres excluent même toute entrée d'un nouvel opérateur. Une réserve (/10, /12 ou /16 selon les registres) est également constituée pour parer à des demandes inattendues ou pour assurer des services de transition

Pénurie d'adresses IPv4

- **Les attributions d'adresses IPv4 en situation de pénurie:**
 - Les clients ne ressentent pas toujours ces restrictions si l'opérateur auquel ils s'adressent dispose de réserves confortables mais cela est de moins en moins vrai et il est maintenant fréquent d'avoir à remplir des formulaires de justification pour une demande de 4 ou 8 adresses

Pénurie d'adresses IPv4

- Mesures appliquées ou applicables pour limiter la consommation d'adresses IPv4 publiques:
 - Pour essayer de retarder la date fatidique où plus aucune adresse ne sera disponible, plusieurs séries de mesures sont appliquées :
 - La translation d'adresse ou NAT (Network Address Translation) est probablement la fonctionnalité qui a évité l'explosion apocalyptique de la demande en adresses IP publiques puisque plusieurs voire des centaines de machines peuvent accéder à Internet en utilisant une seule adresse IPv4 publique
 - Cette translation va de pair avec l'utilisation intensive de l'adressage privé (RFC1918) à l'intérieur des entreprises

Pénurie d'adresses IPv4

- Mesures appliquées ou applicables pour limiter la consommation d'adresses IPv4 publiques:
 - Pour essayer de retarder la date fatidique où plus aucune adresse ne sera disponible, plusieurs séries de mesures sont appliquées :
 - Enfin, les techniques de virtual-hosting pour les serveurs web (techniques permettant d'accueillir plusieurs sites sous une même adresse IP, l'URL d'appel faisant la différence) ont grandement limité les besoins pour les entreprises mais aussi pour les hébergeurs

Les limites d'IPv4

- **1. Épuisement des adresses disponibles:**
 - Donc, s'il ne doit y avoir qu'une seule raison au passage en IPv6, c'est bien celle-ci
 - Même si pour les utilisateurs finaux elle semble un peu abstraite et déconnectée de notre quotidien, c'est pourtant elle qui va vraiment justifier cette transition

Les limites d'IPv4

- **2. Accès direct aux périphériques limité:**
 - *a. Difficultés accrues pour la voix sur IP:*
 - La façon la plus logique d'établir une communication vocale entre deux postes est de faire un appel direct d'un poste à l'autre
 - Mais les postes téléphoniques n'ont généralement pas une adresse IP publique mais une adresse privée. Il faut donc passer par une passerelle et mettre en œuvre le mécanisme de NAT
 - Cela induit évidemment du délai dans la transmission, une baisse de la qualité et un point de fragilité. Cela fonctionne mais de façon peu optimale
 - En résumé, le vrai peer-to-peer n'est plus possible avec l'IPv4 d'aujourd'hui

Les limites d'IPv4

- *b. Difficultés accrues pour la visioconférence:*
 - mêmes difficultés, souvent accrues, que pour la VOIP
 - Tout se passe bien si tous les systèmes à interconnecter disposent chacun d'au moins une adresse IPv4 fixe et publique (vrai peer-to-peer)
 - Si on utilise des adresses privées, ce qui est souvent le cas, le NAT complique les choses car les protocoles de visioconférence ont besoin d'ouvrir des connexions en utilisant des ports négociés dynamiquement entre les deux extrémités
 - Il faut décoder le flux visio pour extraire les ports demandés et modifier dynamiquement les règles du pare-feu ou du routeur pour permettre ce type de trafic
 - C'est possible sur le papier, mais devient un casse-tête en pratique
 - Cela se termine souvent par l'installation de boîtiers complémentaires pour établir le lien entre l'extérieur et l'intérieur du réseau, ce qui engendre des coûts supplémentaires

Les limites d'IPv4

- c. *Limitations sur les accès aux serveurs web internes pour les particuliers:*
 - Si nous voulons accéder depuis Internet aux multiples serveurs, notamment web, qui vont être présents sur un réseau domestique (imprimantes, serveurs, webcam permettant de surveiller les locaux, serveurs NAS), il est indispensable de mettre en place soit des VPN (Virtual Private Network ou réseau privé virtuel), soit des techniques de port forwarding, puisque nous disposons généralement d'une seule adresse IP publique
 - Dans le premier cas, la gestion est lourde au quotidien, nécessitant à chaque fois de monter le VPN avant toute connexion vers les machines internes

Les limites d'IPv4

- c. *Limitations sur les accès aux serveurs web internes pour les particuliers:*
 - Dans le deuxième cas, cela nécessite une certaine aisance avec la technologie TCP/IP puisqu'il va falloir paramétrer le firewall de telle sorte qu'il transfère vers l'adresse IP interne A toute requête HTTP sur le port 8081 (par exemple), vers l'adresse IP de B quand le flux arrive sur le port 8082...
C'est à la fois fastidieux à paramétrer et à utiliser
 - Il existe certes des solutions pour éviter de devoir entrer dans notre réseau comme, dans le cas de caméras IP, un site central auquel les caméras viennent se connecter et auquel on peut accéder à distance
 - Cela revient à la technique de la passerelle déjà évoquée plus haut
 - Là encore, si chacun des matériels (serveurs, caméras...) avait une adresse directement routable, cela serait préférable

Les limites d'IPv4

- *d. Difficultés pour établir des VPN avec de la translation:*
 - Un cas classique est un accès Internet grand public dans lequel nous avons le dispositif suivant : Internet → Routeur ou box du FAI → Firewall → LAN
 - Si nous voulons établir un VPN entre un site (ou un nomade) extérieur et notre site local, nous avons deux solutions :
 - Supprimer la fonction routeur de la box (passer en mode bridge, en fait)
Avantage : l'adresse IP publique peut être affectée directement à notre firewall et nous pouvons donc établir tranquillement le VPN
Inconvénient : plus de fonction téléphone sur IP ou de TV Internet si la box remplissait cette fonction
 - Rediriger les ports ou protocoles liés au VPN vers notre firewall
Avantage : nous pouvons garder la fonction téléphone et TV de la box
Inconvénients : du port forwarding intervient avec les inconvénients liés aux NAT

Les limites d'IPv4

- e. *Difficultés d'utiliser des communications cryptées dans les applications:*
 - Si une machine à l'extérieur de notre réseau a besoin de se connecter à des serveurs privés avec des applications ou des protocoles qui négocient les ouvertures de port de façon dynamique (FTP, SQL, H323...), il sera très difficile ou même impossible d'envisager de crypter ces communications
 - C'est fort dommage si les données à transmettre ou à consulter sont très sensibles
 - Bien sûr, des solutions existent, mais c'est à chaque fois un contournement plus ou moins élégant du problème
 - Une fois encore, la suppression de la translation permet de voir les choses différemment

Les limites d'IPv4

- **3. Conflits d'adressages sur les adresses privées:**
- Les entreprises ont depuis longtemps recours aux adresses privées à la fois pour limiter le besoin d'adresses publiques et aussi parce que cela permet de masquer dans une certaine mesure le réseau interne
- C'est le RFC 1918 qui détermine les adresses utilisables :
10.0.0.0/8, 172.16.0.0/12 et 192.168.0.0/16
- L'ensemble des entreprises utilisant cet adressage privé ont donc recours à ces mêmes adresses
- Il est donc inévitable que des conflits surviennent à un moment ou à un autre

Les limites d'IPv4

- a. *Conflits d'adressages privés lors de la mise en place de VPN entre sites:*
 - Pour illustrer ce problème, nous allons prendre un exemple concret:
 - Un site A utilise le réseau 192.168.1.0/24
 - Un autre site B de cette entreprise souhaite établir un VPN avec le site A mais son réseau est aussi en 192.168.1.0/24
 - Il sera donc impossible aux routeurs/firewalls gérant le tunnel de savoir quand un poste A cherche à joindre l'adresse 192.168.1.25 (par exemple) s'il doit chercher cette adresse en local ou via le tunnel
 - Il faudra donc soit procéder à une renumérotation de toutes les machines sur le site A ou B, soit mettre en place de la translation d'adresses bidirectionnelle dans le tunnel entre A et B
 - La solution est techniquement faisable (sur certains matériels) mais assez lourde dans les faits

Les limites d'IPv4

- b. *Conflits d'adressages privés lors de la fusion de réseaux:*
 - Le même problème peut se poser lorsque deux entreprises cherchent à mutualiser leurs moyens et donc à connecter leurs réseaux via un lien direct ou un VPN opérateur
 - C'est aussi assez fréquemment le cas d'entreprises qui font héberger une partie de leur infrastructure dans un datacenter doté d'un adressage privé
 - Le risque de conflit est assez élevé et dans ce cas la renumérotation étant souvent impossible d'un côté comme de l'autre, nous nous retrouvons à faire des doubles translations, ce qui est toujours contraignant en termes de performances, de règles d'accès, et de résolutions DNS, entre autres

Les limites d'IPv4

- c. *Conflits d'adressages privés dans les VPN fournis par les opérateurs*
 - Les opérateurs utilisent intensivement l'adressage privé, tant au niveau de leurs clients que pour les besoins de leur propre infrastructure
 - La cohabitation de toutes ces adresses ne se fait pas sans risque de confusion et de conflits, que ce soit sur le routage, les VLAN, les règles d'accès

Les limites d'IPv4

- **4. Broadcasts intempestifs et inefficaces**
 - IPv4 est notoirement basé sur des broadcasts (trames de diffusion), par exemple pour :
 - les résolutions IP-adresses MAC par le protocole ARP
 - la recherche de serveurs DHCP
 - un certain nombre d'échanges NetBIOS
 - Comme, par définition, les broadcasts s'adressent à tous les postes d'un réseau local, cela signifie que ceux-ci doivent traiter chaque broadcast arrivant sur leur carte Ethernet avant de décider si la trame leur est adressée ou pas
 - Cela peut faire peser une charge non négligeable sur le poste si le taux de broadcast est élevé

Les limites d'IPv4

- **5. IPv4 n'est plus le protocole de travail par défaut pour certains OS**
 - Il est maintenant établi que les OS modernes, notamment les versions de Windows récentes, ne font plus appel en interne à IPv4 mais à IPv6
 - Cela implique trois choses :
 - Les flux IPv6 vont probablement être privilégiés par rapport à IPv4
 - Lors des choix de routes, les adresses IPv6 seront préférées aux adresses IPv4
 - Dans un futur plus ou moins proche, au moins une partie des réseaux et/ou des applications ne communiqueront plus qu'en IPv6.

Les apports d'IPv6

- **Espace d'adressage "infini":**
 - L'adresse IP passe de 32 bits à 128
 - Multiplier par 4 le nombre d'octets paraît peu mais, si l'on raisonne en nombre d'adresses potentielles, nous passons de 4 294 967 296 adresses (un peu plus de 4 milliards d'adresses, soit bien moins d'une adresse par être humain) à 340 282 366 920 938 463 463 374 607 431 768 211 456 adresses (soit environ $3,4 \times 10^{38}$)
 - Il est habituel d'indiquer que cela fait environ $6,6 \times 10^{23}$ adresses par mètre carré de surface de la Terre (environ $5,1 \times 10^{14} \text{ m}^2$)

Les apports d'IPv6

- Cela permet donc d'affecter une adresse publique à tout appareil devant être connecté sur un réseau IP
- Et cela supprime potentiellement tout conflit d'adresses, sauf erreur grossière de configuration, et toute nécessité de recourir à la translation d'adresses
- Par contre, cela nécessitera une grande vigilance sur les configurations des firewalls protégeant l'entreprise pour que cette accessibilité ne se fasse pas au détriment de la sécurité

Les apports d'IPv6

- **Autoconfiguration des postes:**
 - En IPv4, pour configurer un poste, il n'y avait pas d'autre solution que de laisser le poste en DHCP ou de passer sur le poste pour lui affecter une IP fixe
 - IPv6 permet une autoconfiguration beaucoup plus efficace avec deux options :
 - Laisser le poste s'autoconfigurer de façon autonome
 - Indiquer au poste qu'il doit faire appel à un serveur DHCP

Les apports d'IPv6

- Cette autoconfiguration permet de rendre le poste connectable au réseau quasiment dès son déballage
- Au cours de cette autoconfiguration, le poste découvre, par les annonces des routeurs ou du serveur DHCP, le ou les préfixes en vigueur sur son interface (et peut donc calculer sa propre adresse locale), la passerelle par défaut et éventuellement les serveurs DNS
- Cela peut soulager grandement la tâche de l'administrateur notamment lors d'un changement de fournisseur puisqu'une simple diffusion de préfixes et d'annonces de routeur peut permettre de reconnecter plus ou moins progressivement tous les postes sur le nouveau FAI

Les apports d'IPv6

- **Adresses multiples par interface:**
 - C'était déjà un peu le cas en IPv4 mais cela se généralise en IPv6 : il est possible de spécifier plusieurs adresses IP par interface. Cela peut faciliter la renumérotation, l'hébergement de multiples services...
 - Adressage privé unique:
 - L'utilisation d'adresses Unique Local Unicast ou Link-Local permet de générer des adresses automatiquement différentes les unes des autres, ce qui rend quasiment improbable tout conflit d'adresses sur un lien (sauf erreur grossière de configuration)
 - Cela est renforcé par le mécanisme de découverte des adresses dupliquées

Les apports d'IPv6

- **Remplacement des broadcasts par du multicast:**
 - Dès que des diffusions sont à effectuer, il est possible de mettre en place des groupes de multicast (par exemple tous les routeurs du lien ou du site, tous les serveurs DHCP...), ce qui rend beaucoup plus efficaces ces diffusions puisqu'elles seront plus ciblées. Quand un poste cherche un serveur DHCP, il n'inonde pas tous les réseaux et tous les postes de sa requête.
 - Intégration obligatoire d'IPsec dans IPv6:
 - Cela veut simplement dire que tout matériel ou logiciel compatible IPv6 doit permettre l'utilisation d'IPsec dans les communications. IPsec n'est pas une nouveauté
 - C'est le fait que nous n'ayons plus à nous poser de question sur la disponibilité de ce protocole sur un matériel ou un OS qui est le point clé ici

Les apports d'IPv6

- **En-têtes IP moins gourmands:**
 - Les en-têtes IPv6 ont été simplifiés et rendus plus performants pour accélérer le traitement des paquets par les nœuds intermédiaires.
 - Facilité de renumérotation d'un réseau:
 - Comme nous le verrons dans quelques chapitres, il est aisément de renuméroter un réseau puisque le préfixe propriété est facilement séparable de l'identifiant d'interface réseau
 - Il est donc possible d'en changer tout en conservant l'adressage

Les apports d'IPv6

- **Diffusion multimédia facilitée:**
 - La richesse du multicast IPv6 permet d'envisager assez facilement la diffusion de programmes audio ou vidéo en IPv6 jusqu'au moindre récepteur (TV, radio, micro-ordinateur).
 - Mobilité facilitée:
 - IPv6, par sa richesse en adresses IP et par les protocoles qu'il inclut, va considérablement faciliter la mobilité des postes, que ce soit de travail ou de téléphone, entre la maison, le bureau et d'autres implantations encore

Les apports d'IPv6

- **Pour aller plus loin**
 - <http://www.ietf.org> qui est le site de référence pour les documents RFC.
 - <http://www.rfc-editor.org> pour obtenir ces mêmes documents avec une interface de recherche plus souple et avec la possibilité d'avoir les versions PDF des RFC, ce qui est parfois bien utile pour les imprimer correctement.
 - <http://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xml> pour la liste des préfixes IPv6 attribués par le IANA.
 - <http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml> qui est son équivalent pour les adresses IPv4.
 - <http://www.iana.org> d'une façon générale pour tout ce qui est assignation d'adresses, de protocoles.
 - <http://www.potaroo.net/tools/ipv4/> qui est un des sites de prévisions des adresses IPv4 restantes et de leur épuisement.
 - <http://www.ipv4depletion.com/?cat=4> qui en est un autre.
 - <http://www.worldipv6launch.org/> pour le site consacré aux journées annuelles de tests IPv6.
 - <https://www.google.com/intl/en/ipv6/statistics.html> pour des statistiques sur les requêtes vers Google.
 - <http://6lab.cisco.com/index.php> pour des statistiques compilées par Cisco sur la pénétration d'IPv6.
 - <http://stats.labs.apnic.net/ipv6/>

Syntaxe des adresses IPv6

- Alors qu'en IPv4 une adresse se compose de 32 bits répartis en quatre octets (par exemple 192.168.254.22), une adresse IPv6 comporte 128 bits, soit 16 octets.
- Pour faciliter la manipulation de telles adresses, celles-ci sont divisées en huit blocs de 16 bits séparés par le caractère :, comme dans certaines notations d'adresses MAC
- De même, la notation est également basée sur des caractères hexadécimaux
- Nous trouverons donc uniquement les chiffres de 0 à 9 ainsi que les lettres de A à F.
- Par exemple, nous pourrons rencontrer l'adresse IPv6 suivante : 2001:0db8:0000:0000:0101:abcd:def1:1234

Syntaxe des adresses IPv6

- Une même adresse peut revêtir différentes formes car plusieurs mécanismes de simplification de l'écriture d'une adresse IPv6 existent
- Nous allons maintenant les décrire et donner quelques exemples pour chacun d'entre eux
- **1. Suppression des zéros de tête**
 - Les zéros figurant en tête de chaque bloc peuvent être supprimés, ce qui transformera notre exemple 2001:0db8:0000:0000:0101:abcd:def1:1234 en : 2001:db8:0:0:101:abcd:def1:1234

Syntaxe des adresses IPv6

- **2. Utilisation d'un double ::**
 - Lorsqu'un ou plusieurs blocs consécutifs ne contiennent que des zéros, il est possible de les abréger en utilisant un double :: comme dans notre exemple 2001:0db8:0000:0000:0101:abcd:def1:1234 qui devient alors 2001:db8::101:abcd:def1:1234
 - De même, si nous avions une adresse de base en 2001:db8:1234:101:0:0:0:5678 il serait possible de l'abréger en 2001:db8:1234:101::5678
 - ou bien encore 2001:db8:202:101:abcd:1234::5678 au lieu de 2001:0db8:0202:0101:abcd:1234:0000:5678

Syntaxe des adresses IPv6

- **2. Utilisation d'un double ::**
 - Dans chaque cas, les logiciels et matériels devant interpréter une telle adresse rajouteront autant de blocs de zéros que nécessaire pour obtenir 128 bits
 - Cette dernière précision explique également pourquoi il est interdit d'utiliser plus d'une fois cette abréviation :: dans une adresse
 - En effet, si une adresse comme 2001:db8::1234::5678 se présentait, il serait impossible de savoir combien de blocs représentent chaque :: et donc de trancher entre une adresse originale 2001:db8:0:0:1234:0:0:5678, 2001:db8:0:1234:0:0:0:5678 ou bien encore 2001:db8:0:0:0:1234:0:5678

Syntaxe des adresses IPv6

- **3. Les préfixes en IPv6**

- Il n'y a plus ici de notion de classes d'adressage (A, B ou C) ou de masque de sous-réseau comme cela pouvait se rencontrer en IPv4
- Il faut donc trouver un autre moyen de préciser quelle est la partie d'adresse qui désigne le réseau et celle qui correspond à l'interface elle-même
- C'est le rôle de la longueur de préfixe qui précise combien de bits (en partant de la gauche) représentent le préfixe. C'est un peu l'équivalent des /8, /16, /24... couramment rencontrés en IPv4
- Un préfixe en IPv6 s'exprime donc avec la syntaxe suivante :
adresse IPv6/longueur de préfixe

Syntaxe des adresses IPv6

- **3. Les préfixes en IPv6**

- Nous pouvons utiliser ici aussi les abréviations évoquées plus haut (à quelques nuances près), ce qui donne pour un préfixe de 60 bits tel que 2001:0db8:0000:ba3, les possibilités suivantes de notation :
2001:db8::ba30:0:0:0/60 ou
2001:db8:0:ba30::/60 ou
2001:0db8:0000:ba30:0000:0000:0000:0000/60
- Par contre, il n'est pas permis d'utiliser une notation telle que 2001:0db8::ba30/60 car dans ce cas les règles d'interprétation développeront cette adresse en 2001:0db8:0000:0000:0000:0000:ba30, ce qui ne correspond pas du tout à l'adresse de départ (commençant par 2001:0db8:0000:ba3)

Syntaxe des adresses IPv6

- **3. Les préfixes en IPv6**

- Comme en IPv4, il est possible d'écrire simultanément l'adresse de l'interface et le préfixe comme dans l'exemple suivant :
2001:0db8:0000:ba30:1234:5678:9abc:def0/60
- Les préfixes permettent de déterminer le type d'une adresse (un peu comme les premiers bits d'une adresse IPv4 permettaient, à l'origine, de déterminer la classe de cette adresse (A,B,C...))

Syntaxe des adresses IPv6

- Nous allons trouver ainsi les valeurs suivantes:

Type d'adresse	Préfixe binaire	Notation en IPv6
Unspecified	0000....0 (128 bits)	::/128
Loopback	0000....1 (128 bits)	::1/128
Multicast	1111 1111	ff00::/8
Link-Local Unicast	1111 1110 10	fe80::/10
Unique Local Unicast	1111 1100 et 1111 1101	fc00::/7
Global Unicast	Tout le reste	

Syntaxe des adresses IPv6

- **4. Recommandations d'écriture pour faciliter le traitement des adresses IPv6**
 - La variabilité dans la représentation d'une même adresse IPv6 peut compliquer considérablement certaines tâches car comment comparer, classer, vérifier des adresses alors que leur syntaxe est variable
 - Par exemple, doit-on chercher 2001:db8::1:0:0:1 ou 2001:0db8:0:0:1:0:0:1 ou bien encore 2001:db8:0:0:1::1 ?

Syntaxe des adresses IPv6

- **4. Recommandations d'écriture pour uniformiser la représentation des adresses IPv6 (RFC 5952 d'août 2010)**
 - Suppression des zéros de tête
 - Compresser au maximum les champs en utilisant les ::
 - Les :: doivent remplacer le plus grand nombre possible de doubles octets à 0
 - En cas d'égalité du nombre de 0 remplacés, c'est la première série de 0 qui doit être compressée
 - Ainsi pour 2001:db8:0:0:1:0:0:1, on écrira 2001:db8::1:0:0:1 de préférence à 2001:db8:0:0:1::1
 - Les adresses doivent être écrites en minuscules
 - En cas de combinaison des adresses avec des ports, il faut utiliser les crochets : [2001:db8::1:0:0:1:80]

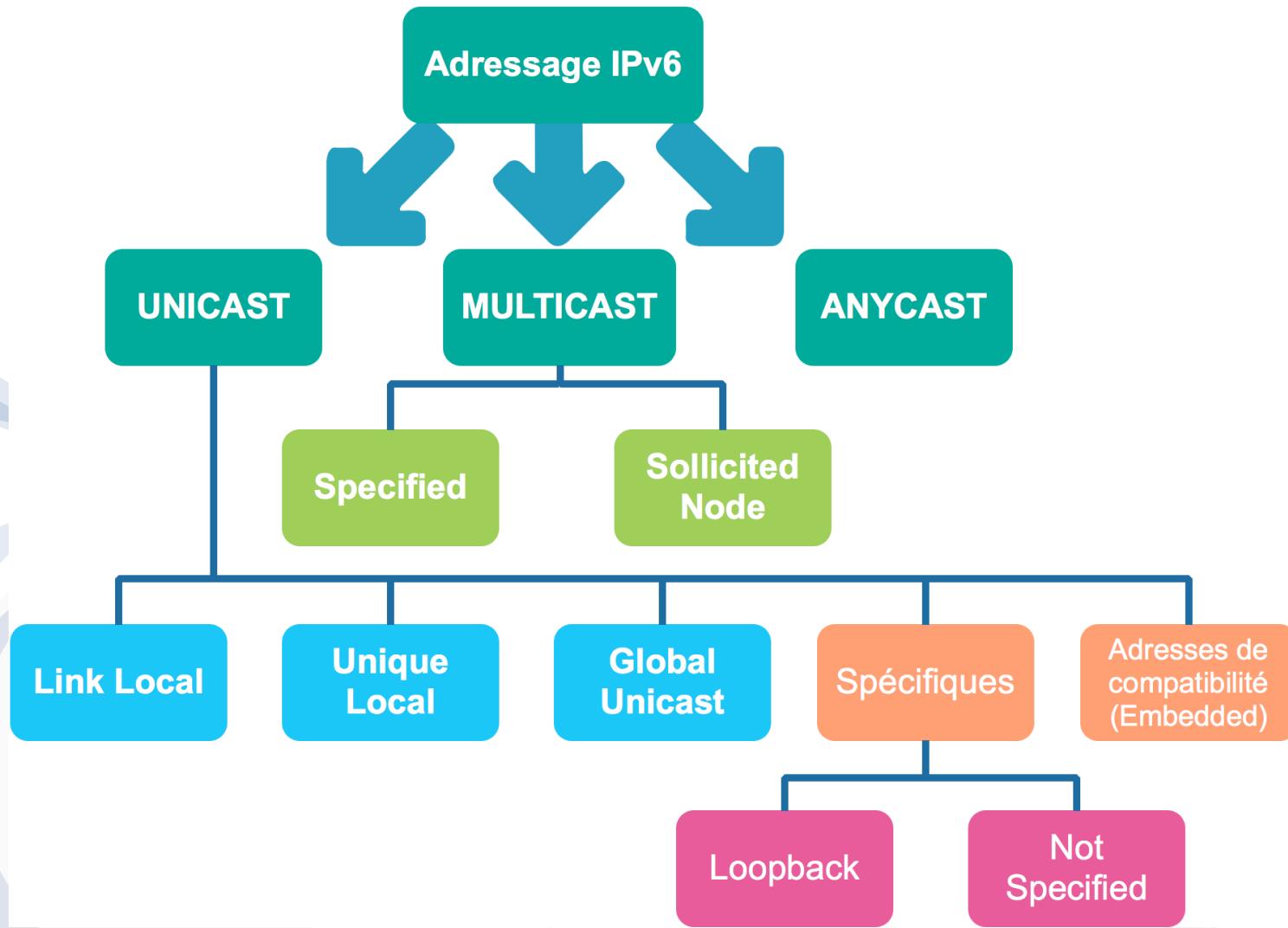
Types d'adresses IPv6 - Généralités

- En IPv6, il existe de nombreux types d'adresses
- Il y a tout d'abord trois catégories principales :
 - **Unicast** - c'est l'adresse la plus classique, qui désigne une interface unique en IPv6
Tout paquet ayant pour destination cette adresse est délivré uniquement à l'interface détentrice de cette adresse
 - **Multicast** - comme en IPv4, tout paquet envoyé à une adresse de ce type est reçu et traité par l'ensemble des interfaces appartenant au groupe de diffusion désigné par cette adresse

Types d'adresses IPv6 - Généralités

- **Anycast** - il s'agit de désigner une adresse pouvant être détenue par plusieurs interfaces (sur un même matériel ou sur des matériels différents)
Dans ce cas, un paquet envoyé à une adresse Anycast est traité seulement par une de ces interfaces, souvent celle qui est la plus proche topologiquement
- À l'intérieur de ces principaux types, notamment le type Unicast, nous allons retrouver d'autres distinctions

Types d'adresses IPv6 - Généralités



Types d'adresses IPv6 – adresses Unicast

- Elle se décompose généralement en un identifiant d'interface (interface ID) et un préfixe de sous-réseau (subnet prefix) selon le schéma suivant :

n premiers bits	128 -n derniers bits
-> Préfixe du sous-réseau	-> Interface ID

- L'interface ID est la plupart du temps sur 64 bits
- Elle est couramment dérivée de l'adresse MAC de l'interface mais elle peut aussi être définie manuellement ou de façon aléatoire

Types d'adresses IPv6 – adresses Unicast

- À l'intérieur du type Unicast, nous allons encore trouver la distinction entre :
 - **Global Unicast** - ces adresses sont routables au travers de l'ensemble d'un réseau IPv6, que ce soit sur Internet ou sur des liens privés et sont donc uniques au monde
 - **Link-Local Unicast** - ces adresses n'ont qu'une signification locale et ne sont pas routables en dehors du lieu local
C'est un peu une extension de la notion d'adresses privées telles que définies par le RFC 1918 en IPv4
 - **Unique Local Unicast** - adresses routables dans un réseau privé (sur un ou plusieurs sites) mais pas sur Internet
- Il existait également un autre type d'adresse dénommé Site-local Unicast mais il a été rendu obsolète en 2004

Types d'adresses IPv6 – adresses Unicast

- a. **Adresse de type Link-Local Unicast**

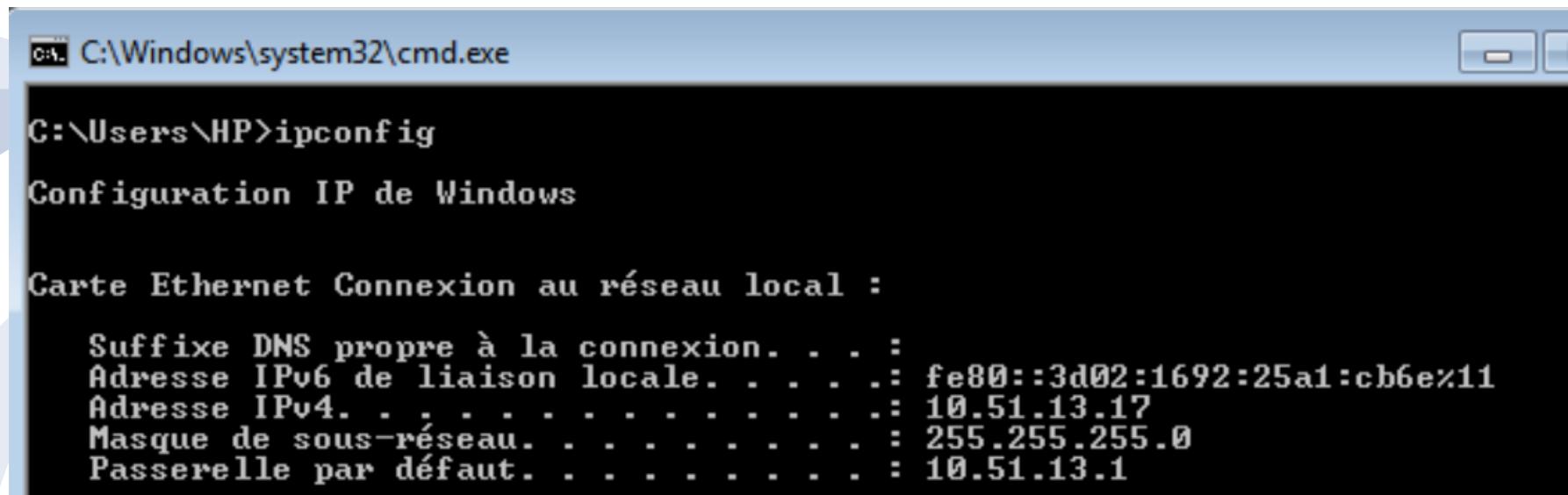
- Sa structure est des plus simples et permet d'adresser uniquement des systèmes présents sur un même lien sans possibilité de routage
- Sa structure est la suivante :

10 premiers bits	54 bits suivants	64 derniers bits
1111 1110 10	0	Interface ID

- Ce qui nous explique pourquoi les adresses de ce type commencent systématiquement par fe80::/10
- Ce sont souvent les seules adresses générées sur une interface par les mécanismes d'autoconfiguration

Types d'adresses IPv6 – adresses Unicast

- Exemple sur un PC sous Windows 7 en autoconfiguration :



```
C:\Windows\system32\cmd.exe
C:\Users\HP>ipconfig
Configuration IP de Windows

Carte Ethernet Connexion au réseau local :

    Suffixe DNS propre à la connexion. . . . . : 
    Adresse IPv6 de liaison locale. . . . . : fe80::3d02:1692:25a1:cb6e%11
    Adresse IPv4. . . . . : 10.51.13.17
    Masque de sous-réseau. . . . . : 255.255.255.0
    Passerelle par défaut. . . . . : 10.51.13.1
```

Types d'adresses IPv6 – adresses Unicast

- L'adresse elle-même est ici fe80::3d02:1692:25a1:cb6e, le suffixe %11 indiquant le numéro (index) de l'interface sur laquelle est présente cette adresse
- Les routeurs ne doivent pas transférer des paquets ayant comme source ou comme destination une adresse de type link-local

Types d'adresses IPv6 – adresses Unicast

- **b. Adresses de type Global Unicast**

- C'est l'équivalent de l'adresse IPv4 publique, routable sur Internet et unique mondialement
- L'adresse se décompose en trois zones :

n premiers bits	m bits suivants	128-n-m derniers bits
Préfixe de routage global	Subnet ID	Interface ID

- Le préfixe de routage global est la valeur permettant de router les paquets depuis Internet vers un site précis
- Le subnet ID permet d'identifier, sur ce site, le lien portant le sous-réseau ainsi désigné

Types d'adresses IPv6 – adresses Unicast

- L'URL <http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xhtml> permet d'obtenir la liste des préfixes affectés aux différents registres
- Pour l'instant, seul le préfixe 2000::/3 est utilisé pour ces affectations
- La plupart des autres préfixes sont seulement réservés (cf. <http://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xml>)
- Il est à noter que le préfixe 2001:0db8/32 n'est pas routable bien qu'appartenant à la zone APNIC car il n'est destiné qu'à être utilisé dans les documentations (comme nous l'avons fait au début de ce chapitre), comme spécifié par le RFC 3849 de juillet 2004

Types d'adresses IPv6 – adresses Unicast

- **c. Adresses de type Unique Local Unicast**

- Voici maintenant un type d'adresses intermédiaire entre l'adresse locale et l'adresse globale
- Pour répondre au besoin de routage intrasite ou entre sites via des tunnels ou des réseaux privés, fonction mal remplie par les adresses maintenant obsolètes de type Site-local, une catégorie d'adresses locales a été ajoutée : Unique Local Unicast
- Ces adresses ne sont pas supposées être routées directement sur Internet mais sont par contre conçues pour être générées par un algorithme de telle sorte que des réseaux identiques ne puissent pas exister sur deux entités différentes
- Ainsi deux entreprises qui fusionnent ou établissent des liens directs entre elles n'ont qu'un infime risque de devoir renuméroter leurs réseaux (ou mettre en place des translations d'adresses), comme c'est souvent le cas actuellement en IPv4

Types d'adresses IPv6 – adresses Unicast

- Le préfixe réservé pour ces adresses est fc00::/7. La structure des adresses, définie par la RFC 4193 (octobre 2005), est la suivante :

7 premiers bits	1	40 bits suivants	16 bits suivants	64 derniers bits
Préfixe 1111 110	bit L	Global ID	Subnet ID	Interface ID

- Le préfixe fc00::/7 permet d'identifier les adresses uniques de type Local Unicast
- Le bit L est positionné à 1 si le préfixe est fixé localement
- La valeur 0 est réservée pour un usage futur
- Cela induit que les adresses de ce type commencent actuellement systématiquement par FD

Types d'adresses IPv6 – adresses Unicast

- Pour générer ces adresses, il faut utiliser l'algorithme pseudo-aléatoire décrit dans le RFC 4193
- Pour nous faciliter la tâche, il existe plusieurs sites web supposés suivre ces spécifications
- Nous pouvons par exemple citer www.ultratools.com
- Sur ce site, nous cliquons sur l'onglet **UltraTools** puis sur le sous-onglet **IPv6 Tools**, puis **Local IPv6 Range Generator**

Types d'adresses IPv6 – adresses Unicast

- Nous pouvons également directement utiliser l'URL:
<https://www.ultratools.com/tools/rangeGenerator>

[IPv6 Tools](#)

[IPv4 to IPv6 Conversion](#)

[IPv6 CIDR to Range](#)

[Range to IPv6 CIDR](#)

[IPv6 Compress](#)

[IPv6 Expand](#)

[IPv6 Info](#)

Local IPv6 Range Generator >

[IPv6 Compatibility](#)

The Local IPv6 Range Generator tool can be used to generate global IDs, subnet IDs, and the valid IPv6 range of addresses. Both the global ID and the subnet ID should always be filled in if you are operating on an existing network and existing subnet.

If you are deploying an entirely new network you will need a new global ID and new subnet ID - leave both fields blank and press "Go."

If you are deploying a new subnet to an existing network, fill in the global ID and leave the subnet ID blank.

If you need to generate a new local IPv6 range for an existing subnet, fill in your global ID and subnet ID, and press "Go."

Enter a Global ID and / or a Subnet ID, respectively:

34fe567891 2f3a Go »

Related Tools: [IPWHOIS Lookup](#) [Decimal IP Calculator](#) [Traceroute](#) [Vector Trace](#) [IPv4 to IPv6 Conversion](#) [IPv6 Expand](#) [IPv6 CIDR to Range](#) [IPv6 Compress](#) [CIDR/Netmask](#) [IPv6 Info](#) [IPv6 Compatibility](#)

- Il suffit alors de laisser vides les champs **Global ID** et **Subnet ID** puis de cliquer sur **Go**

Types d'adresses IPv6 – adresses Unicast

- Une plage est générée:

Enter a Global ID and / or a Subnet ID, respectively:

Related Tools: [IPWHOIS Lookup](#) [Decimal IP Calculator](#) [Traceroute](#) [Vector Trace](#) [IPv4 to IPv6 Conversion](#) [IPv6 Expand to Range](#) [IPv6 Compress](#) [CIDR/Netmask](#) [IPv6 Info](#) [IPv6 Compatibility](#)

Prefix/L: fd
Global ID: 69859b43ee
Subnet ID: 5fd2
Combine/CID: fd69:859b:43ee:5fd2::/64
IPv6 addresses: fd69:859b:43ee:5fd2::/64:XXXX:XXXX:XXXX:XXXX
Start Range: fd69:859b:43ee:5fd2:0:0:0:0
End Range: fd69:859b:43ee:5fd2:ffff:ffff:ffff:ffff
No. of hosts: 18446744073709551616

- Avec par ordre d'affichage: les 8 premiers bits: **fd**; le global ID: **fcced56d56**; le subnet ID: **5763**
- Ces trois éléments combinés nous donnent le préfixe de 64 bits de notre réseau : **fdfc:ced5:6d56:5763::**

Types d'adresses IPv6 – adresses Unicast

- Si nous voulons disposer d'un autre sous-réseau, tout en restant dans le réseau global fcced56d56, il suffit de revenir sur la même page en remplissant seulement la zone **Global ID** et de cliquer sur **Go**

69859b43ee

Go »

- Ce qui conduit au résultat suivant:

69859b43ee

ba63

Go »

Related Tools: [IPWHOIS Lookup](#) [Decimal IP Calculator](#) [Traceroute](#) [Vector Trace](#) [IPv4 to IPv6 Conversion](#) [IPv6 Expand](#) [IPv6 CIDR to Range](#) [IPv6 Compress](#) [CIDR/Netmask](#) [IPv6 Info](#) [IPv6 Compatibility](#)

```
Prefix/L: fd
Global ID: 69859b43ee
Subnet ID: ba63
Combine/CID: fd69:859b:43ee:ba63::/64
IPv6 addresses: fd69:859b:43ee:ba63::/64:XXXX:XXXX:XXXX:XXXX
Start Range: fd69:859b:43ee:ba63:0:0:0:0
End Range: fd69:859b:43ee:ba63:ffff:ffff:ffff:ffff
No. of hosts: 18446744073709551616
```

Types d'adresses IPv6 – adresses Unicast

- Nous pouvons également citer d'autres liens offrant le même type de service :
 - <http://www.simpledns.com/private-ipv6.aspx>
 - <http://www.dnsstuff.com/tools/>
 - <http://unique-local-ipv6.com/>
- Là encore, l'utilitaire **ipv6calc** permet de décoder de telles adresses comme dans l'exemple ci-dessous qui fait bien ressortir la notion d'adresse Unique Local Unicast (cf. deuxième ligne) :

```
root@logs:~# ipv6calc -q -i fdfc:ced5:6d56:d9c4::c0a8:c801
Address type: unicast, unique-local-unicast, iid, iid-local
Registry for address: reserved(RFC4193#3.1)
Address type has SLA: d9c4
Interface identifier: 0000:0000:c0a8:c801
Interface identifier is probably manual set
root@logs:~#
```

Types d'adresses IPv6 – adresses Unicast

- **d. Adresses de type Site-local Unicast**
 - Ces adresses ne doivent plus être employées dans les nouvelles implémentations d'IPv6 car elles ont été rendues obsolètes par le RFC 3879 (décembre 2004)
 - Elles peuvent néanmoins continuer à être utilisées par les anciennes implémentations
 - Leur structure de base ne diffère que très peu du type étudié précédemment :

10 premiers bits	54 bits suivants	64 derniers bits
1111 1110 11	Subnet ID	Interface ID

- Ce type d'adresses commence donc toujours par un préfixe compris entre fec etfef

Types d'adresses IPv6 – adresses Multicast

- Comme nous l'avons déjà évoqué, tout paquet envoyé à une adresse de ce type est reçu et traité par l'ensemble des interfaces appartenant au groupe de diffusion désigné par cette adresse
- Chaque adresse de type multicast identifie un groupe d'interfaces (donc généralement un groupe de matériels)

Une même interface peut appartenir à différents groupes de multicast

Types d'adresses IPv6 – adresses Multicast

- **a. Syntaxe**

- La structure de l'adresse multicast est la suivante :

8 premiers bits	4 bits suivants	4 bits suivants	112 derniers bits
1111 1111	drapeaux (flags)	portée (scope)	identifiant de groupe (group ID)

- Le premier groupe de 8 bits (1111 1111) identifie l'adresse comme étant une adresse de type multicast
 - Une adresse multicast est donc dotée d'un préfixe ff00::/8
 - Les drapeaux (flags) sont au nombre de quatre avec les valeurs symboliques suivantes (de gauche à droite) : 0 R P T
 - Le premier bit est donc systématiquement à zéro
 - Le bit T, quand il est à zéro, désigne une adresse affectée de façon permanente (well-known address) par l'IANA (Internet Assigned Numbers Authority)

Types d'adresses IPv6 – adresses Multicast

- S'il est à 1, nous avons alors affaire à une adresse transitoire ou affectée dynamiquement (transient or dynamically assigned)
- Le bit P, quand il est à zéro, signifie que l'adresse multicast n'est pas basée sur le préfixe du réseau (cas classique, notamment dans le cas où le bit T est à zéro) contrairement à ce qui se passe si le bit P est à 1 (qui implique obligatoirement un bit T à 1 donc une adresse affectée dynamiquement)
- Le RFC 3306 détaille ce dernier cas.
- Quant au bit R, il est à zéro dans le cas le plus courant et à 1 dans le cas où l'adresse de multicast inclut une adresse d'un point de rendez-vous (notions détaillées ultérieurement)
- Dans ce dernier cas, les bits P et donc T sont obligatoirement positionnés à la valeur 1

Types d'adresses IPv6 – adresses Multicast

8 premiers bits	4 bits suivants	4 bits suivants	112 derniers bits
1111 1111	drapeaux (flags)	portée (scope)	identifiant de groupe (group ID)

- Le champ portée (scope), comme le nom le laisse supposer, indique l'étendue sur laquelle porte cette adresse multicast
- Les valeurs principales sont :
 - 1 - portée limitée à l'interface (Interface-local scope) : pour tester en bouclage la transmission des multicast
 - 2 - portée limitée au lien (Link-local scope), comme pour une adresse link-local unicast
 - 5 - portée limitée au site (Site-local scope)
 - E - portée globale (Global scope)

Types d'adresses IPv6 – adresses Multicast

- Ainsi, si l'on suppose qu'un service particulier (NTP par exemple) se voit assigner l'identifiant de groupe 123 en hexadécimal, nous pourrons alors trouver les adresses de multicast suivantes:
 - ff01:0:0:0:0:0:123 pour tous les serveurs NTP situés sur la même interface que l'expéditeur
 - ff02:0:0:0:0:0:123 pour tous les serveurs NTP situés sur le même lien réseau que l'expéditeur
 - ff05:0:0:0:0:0:123 pour tous les serveurs NTP situés sur le même site que l'expéditeur
 - ff0e:0:0:0:0:0:123 pour tous les serveurs NTP du réseau Internet

Types d'adresses IPv6 – adresses Multicast

- **b. Règles**
 - Des contraintes particulières s'appliquent aux adresses multicast :
 - Une adresse multicast ne peut être utilisée comme adresse source
 - Les routeurs ne doivent pas propager les adresses multicast au-delà de leur portée, tenant compte pour cela du champ scope figurant dans l'adresse
 - Et quelques autres encore, détaillées dans le RFC 4291

Types d'adresses IPv6 – adresses Multicast

- **c. Exemples**
 - Parmi les adresses multicast prédéfinies, nous pouvons citer « All Routers » :
 - ff01:0:0:0:0:0:2 ou ff01::2 en version abrégée pour tous les routeurs de l'interface
 - ff02:0:0:0:0:0:2 ou ff02::2 en version abrégée pour tous les routeurs du lien
 - ff05:0:0:0:0:0:2 ou ff05::2 en version abrégée pour tous les routeurs du site
 - ou bien « All Nodes » :
 - ff01:0:0:0:0:0:1 ou ff01::1 en abrégé
 - ff02:0:0:0:0:0:1 ou ff02::1 en abrégé

Types d'adresses IPv6 – adresses Multicast

- **d. Cas particulier : adresses sollicited-node**
 - Ces adresses de type multicast sont un peu particulières
 - Elles permettent de cibler un ensemble de nœuds (poste, équipements, serveurs...) dont les 24 derniers bits de la partie interface ID sont identiques
 - Le but est de remplacer les broadcasts couramment utilisés en IPv4 par le protocole ARP pour découvrir l'adresse MAC correspondant à une IP donnée
 - Ce mécanisme est utilisé intensivement par les mécanismes de découverte des voisins qui en IPv6 prennent le relais d'ARP

Types d'adresses IPv6 – adresses Multicast

- **d. Cas particulier : adresses sollicited-node**
 - Une adresse multicast est donc créée à partir de l'adresse IPv6 recherchée en combinant le préfixe multicast ff02::1:ff00:0/104 avec les 24 derniers bits de l'adresse IP
 - Chaque équipement présent sur un réseau rejoint obligatoirement le groupe multicast sollicited-node correspondant à chacune des adresses IPv6 présentes sur ses interfaces

Types d'adresses IPv6 – adresses Multicast

- La commande Windows permettant de visualiser les adhésions aux différents groupes est **netsh int ipv6 show joins**.
- Voici un exemple pour un poste :

```
C:\Windows\system32\cmd.exe
C:\Users\HP>netsh int ipv6 sh joins
Interface 1 : Loopback Pseudo-Interface 1
Étendue    Références   Dern   Adresse
-----    -----   -----   -----
0          3   Oui   ff02::c

Interface 12 : isatap.{DE845CCF-5465-471D-9E35-7A29C6368C87}
Étendue    Références   Dern   Adresse
-----    -----   -----   -----
0          1   Oui   ff02::1:ffa8:8

Interface 13 : Connexion au réseau local* 4
Étendue    Références   Dern   Adresse
-----    -----   -----   -----
0          0   Oui   ff01::1
0          0   Oui   ff02::1
0          1   Oui   ff02::1:ff00:0
```

Types d'adresses IPv6 – adresses Anycast

- Ces adresses ne sont en fait pas différentes des adresses unicast déjà rencontrées
- Il est impossible de les distinguer d'après leur syntaxe et seules les interfaces auxquelles elles ont été assignées sont conscientes de leur nature anycast
- Le principe de fonctionnement est qu'une même adresse peut être affectée à des interfaces différentes (dans la plupart des cas sur des matériels différents) et qu'en cas de diffusion d'un paquet ayant une telle adresse comme destination, seule l'interface la plus proche dans la topologie du réseau va prendre en charge ce paquet (différence essentielle par rapport au multicast, dans lequel toutes les interfaces dotées de la même adresse prennent en compte le paquet ; nous sommes donc bien dans une utilisation unicast)

Types d'adresses IPv6 – adresses Anycast

- Le RFC 4786 de décembre 2006 détaille les bonnes pratiques pour l'utilisation de telles adresses
- Leur principal usage est de proposer une adresse IP unique pour les différents routeurs pouvant être reliés à Internet ou à des réseaux distants de l'entreprise
- Dans ce cas, seul le routeur le plus pertinent géographiquement va prendre en charge une demande de connexion effectuée avec l'adresse anycast comme destination

Types d'adresses IPv6 – adresses Anycast

- **Subnet-router anycast:**
 - Cette adresse est un cas particulier permettant de s'adresser à n'importe quel routeur d'un sous-réseau
 - Dans ce cas, seul le préfixe correspondant au réseau concerné est renseigné, l'identifiant d'interface étant à zéro :

n premiers bits	128 -n derniers bits
Préfixe du sous-réseau	0000000000...000

- Un exemple d'utilisation de préfixe anycast était celui défini pour les routeurs servant de relais pour le mécanisme 6to4 permettant l'interconnexion d'IPv4 à IPv6 mais ce préfixe a été rendu obsolète par le RFC 7526 de mai 2015

Types d'adresses IPv6 – adresses spécifiques

- **a. Loopback**
 - L'adresse de type loopback est une adresse de bouclage : 0:0:0:0:0:0:0:1 ou ::1/128. Elle est utilisée par une interface pour s'envoyer des paquets
 - C'est un peu l'équivalent de 127.0.0.1 en IPv4
- **b. Adresse non spécifiée (unspecified address)**
 - Cette adresse en 0:0:0:0:0:0:0:0 ou ::/128 indique tout simplement l'absence d'adresse IPv6 sur une interface
 - Elle ne doit jamais être utilisée sur une interface ou comme adresse de destination
 - Par contre, elle peut être utilisée comme adresse source, par exemple lors d'une requête DHCP

Types d'adresses IPv6 – adresses de compatibilité IPv4

- Ces adresses permettent de véhiculer des adresses IPv4 dans les 32 bits les plus à droite des adresses IPv6
- Le RFC 4038 traite en détail leur utilisation
- a. **IPv4-Compatible IPv6 address**
 - Les adresses de ce type se présentent sous la forme suivante :

80 premiers bits	16 bits suivants	64 derniers bits
00.....00	0	adresse IPv4

- Ces adresses ont été rendues obsolètes par les mécanismes de transition actuels

Types d'adresses IPv6 – adresses de compatibilité IPv4

- **b. IPv4-Mapped IPv6 address**
 - Ces adresses ont le format suivant :

80 premiers bits	16 bits suivants	64 derniers bits
00.....00	ffff	adresse IPv4

- Ces adresses ne sont utilisées qu'en interne, par le noyau, pour permettre aux applications de ne traiter que des adresses au format IPv6, alors que les communications TCP/IP avec le réseau se font indifféremment en IPv4 ou IPv6
- C'est alors le mécanisme de translation dual-stack étudié plus loin qui fait les translations nécessaires entre ces adresses mappées et les adresses IPv4 d'origine

Types d'adresses IPv6 – adresses de compatibilité IPv4

- **c. ISATAP address**

- Les adresses ISATAP (pour Intra-Site Automatic Tunnel Addressing Protocol) sont conçues pour permettre l'interconnexion par des connexions uniquement IPv4 de systèmes travaillant en dual-stack (c'est-à-dire possédant une couche IPv4 et une couche IPv6)
- Ces systèmes créent alors un tunnel entre eux pour peu que leur système d'exploitation soit capable de le faire, ce qui est possible depuis Windows XP
- Le format de ces adresses est le suivant :

64 premiers bits	32 bits suivants	32 derniers bits
Préfixe	00 00 5e fe ou 02 00 5e fe	adresse IPv4

Types d'adresses IPv6 – adresses de compatibilité IPv4

- Les 64 premiers bits correspondent à une adresse unicast globale
- Les 32 bits suivants peuvent prendre deux valeurs selon que l'adresse IPv4 véhiculée est de type privé (dans ce cas les 32 bits commencent par la valeur 00), ou public (avec une valeur commençant par 02)
- La chaîne 00:00:5e correspond à l'OUI (Organizationally Unique Identifier) attribué à l'IANA (Internet Assigned Numbers Authority) par l'IEEE (Institute of Electrical and Electronic Engineers, autorité enregistrant notamment les identifiants de constructeurs pour attribuer les adresses MAC)
- Enfin, la valeur fe figurant à la fin de ces 32 bits signale que cette adresse IPv6 contient en fait une adresse IPv4
- Si nous souhaitons donner un exemple à partir du préfixe 2001:0dB8:1234:5678 ainsi que l'adresse IPv4 192.168.89.22, nous aboutissons à l'adresse ISATAP 2001:0dB8:1234:5678:0000:5efe:192.168.89.22 que nous pouvons également noter 2001:0dB8:1234:5678:0000:5efe:c0a8:5916 dans la forme hexadécimale complète.

Types d'adresses IPv6 – adresses de compatibilité IPv4

- L'utilitaire `ipv6calc` déjà mentionné nous permet de décoder facilement une telle adresse :

```
droopy:~# ipv6calc -q -i 2001:0db8:1234:5678:0000:5efe:c0a8:5916
Address type: unicast, global-unicast, ISATAP
Address type has SLA: 5678
Registry for address: APNIC
Interface identifier: 0000:5efe:c0a8:5916 IPv4
registry for ISATAP client address:
IPv4 address: 192.168.89.22
IPv4 registry[192.168.89.22]: reserved(RFC1918)
droopy:~#
```

Types d'adresses IPv6 – adresses de compatibilité IPv4

- Ce même utilitaire peut nous servir à faire la conversion IPv4 ↔ IPv6 comme dans les exemples ci-dessous :

```
admin@jerry ~]$ ipv6calc -q --action conv6to4 -in ipv4  
192.168.200.1  
2002:c0a8:c801::
```

```
[admin@jerry ~]$ ipv6calc -q --action conv6to4 --in ipv4  
192.168.89.22  
2002:c0a8:5916::
```

- La conversion inverse est également possible

Types d'adresses IPv6 – adresses de compatibilité IPv4

- **d. Teredo address**
 - Ces adresses correspondent à un autre cas, dans lequel des systèmes IPv6 se retrouvent derrière des routeurs ou des pare-feu opérant une translation d'adresse (NAT) et doivent emprunter des réseaux uniquement IPv4 pour communiquer entre eux
 - Le trafic sera encapsulé dans de l'UDP et devra passer par un serveur relais

Types d'adresses IPv6 – adresses de compatibilité IPv4

- **d. Teredo address**
 - Cela explique le format un peu compliqué de l'adresse Teredo :

Préfixe (32 bits)	Adresse du serveur (32 bits)	Flags (16 bits)	Ports (16 bits)	Adresse du client (32 bits)
-------------------	------------------------------	-----------------	-----------------	-----------------------------

Types d'adresses IPv6 – pour aller plus loin

- 1. Un outil : **ipv6calc (Linux)**
 - L'utilitaire **ipv6calc** présent dans certaines distributions Linux et BSD ou téléchargeable depuis l'URL
<http://www.deepspace6.net/projects/ipv6calc.html> permet des opérations de décodage et/ou de conversion sur ces adresses IP

Types d'adresses IPv6

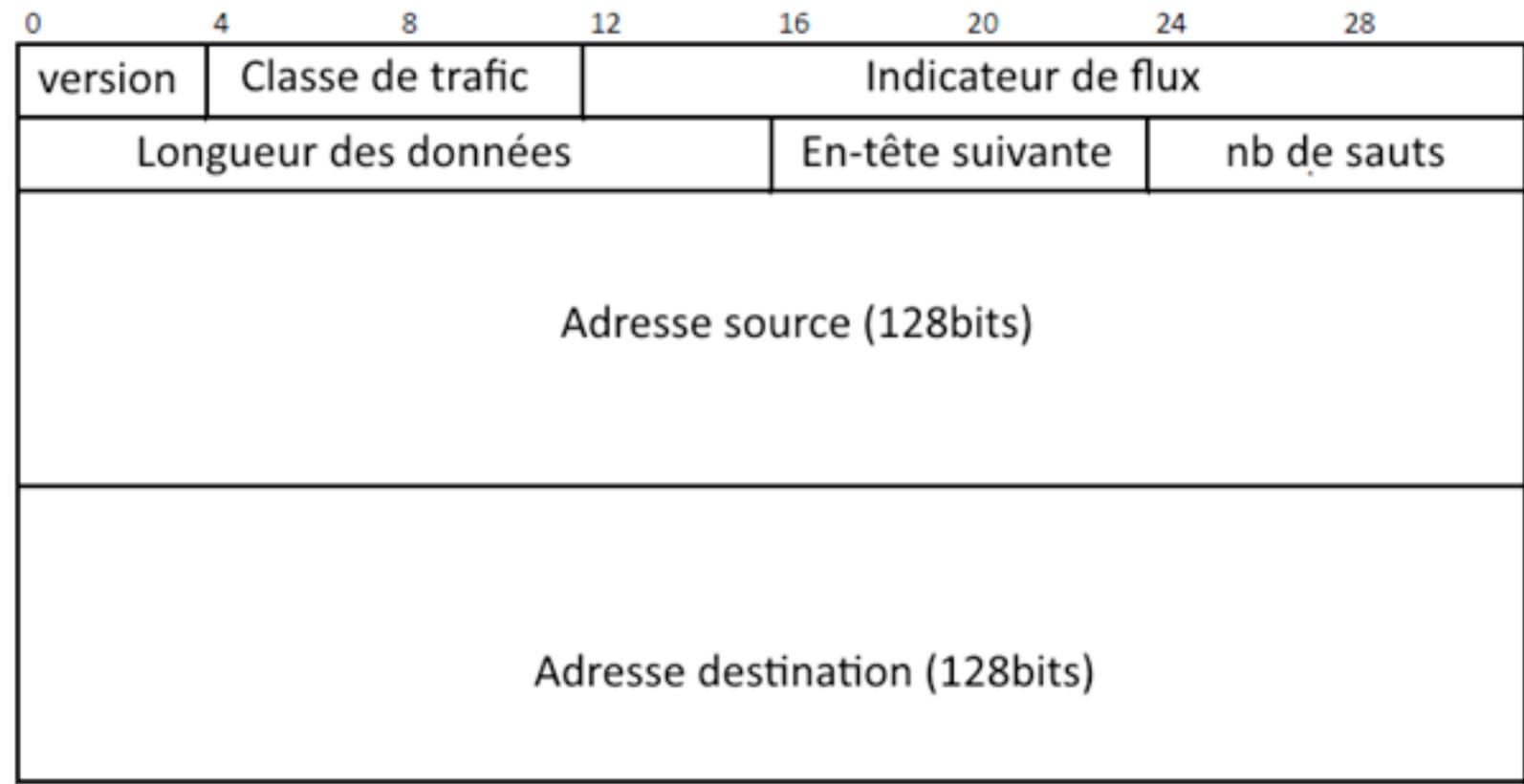
- **2. Quelques documents**
 - RFC 4291 pour les principes de base des adressages IPv6
 - RFC 3306, RFC 3956, RFC 4489 pour les adresses multicast
 - RFC 4038 pour les adressages compatibles IPv4
 - RFC 5214 (et 4214 pour l'ancienne version) pour les adresses ISATAP
 - RFC 4380, RFC 5991 et RFC 6081 pour les adresses Teredo
 - RFC 0791 et 1349 pour les champs des paquets IPv6

Types d'adresses IPv6

- **3. Quelques liens**
 - <http://www.ietf.org> qui est le site de référence pour les documents RFC.
 - <http://www.rfc-editor.org> pour obtenir ces mêmes documents avec une interface de recherche plus souple et avec la possibilité d'avoir les versions PDF des RFC, ce qui est parfois bien utile pour les imprimer correctement.
 - <http://www.iana.org/assignments/protocol-numbers/> pour récupérer la liste la plus récente des protocoles reconnus.
 - <http://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xml> pour la liste des préfixes IPv6 attribués par l'IANA.
 - <http://www.ultratools.com> pour un site de calcul d'adresses IPv6 et notamment d'adresses uniques.
 - <http://www.simpledns.com/private-ipv6.aspx> pour accéder à un autre calculateur d'adresses IPv6 uniques.
 - <http://www.dnsstuff.com/tools/> pour accéder à un ensemble d'outils (hélas payants pour la plupart) dont un autre calculateur d'adresses IPv6.
 - <http://www.deepspace6.net/projects/ipv6calc.html> pour trouver la documentation et les sources de l'utilitaire **ipv6calc**.

Structure des adresses IPv6

- Voici l'entête de base d'un paquet IPv6



Structure des adresses IPv6

- Examinons maintenant les différents champs le constituant :
 - **Version (4 bits)** désigne évidemment la version d'IP utilisée
 - Ici, ce sera 6 au lieu de 4 dans les versions antérieures
 - **Traffic Class (8 bits)** remplace le champ TOS (Type of Service) qui classe les données à faire circuler, et donc les priorités qui leur sont affectées. Les valeurs sont définies par les RFC 0791 et 1349
 - Son utilisation est la même qu'en IPv4

Structure des adresses IPv6

- **Flow Label (20 bits)** est un champ nouveau
 - Son rôle est de permettre aux équipements intermédiaires de routage de pouvoir identifier un type de flux et le traiter en conséquence sans avoir à analyser les flux en détail (notamment sans ouvrir les en-têtes de la couche IP ni les en-têtes de la couche transport sur chaque routeur)
 - Il permet donc potentiellement d'améliorer les performances des routeurs tout au long du parcours
 - De nombreux RFC documentent l'utilisation de ce champ, principalement le RFC 6437

Structure des adresses IPv6

- **Payload Length (16 bits)** : contrairement à IPv4, l'en-tête est de taille fixe (20 octets), ce qui implique que ce champ longueur désigne la longueur totale du datagramme (en dehors de l'en-tête IP de base)
 - Les 16 bits permettent donc d'envisager une longueur maximale de 65536 octets
- **Hop Limit (8 bits)** : remplace le TTL (Time To Live) de la version 4
 - Le fonctionnement est le même qu'en IPv4 puisque ce champ désigne le nombre de sauts (autrement dit de franchissements de routeurs) qu'un datagramme peut effectuer avant de voir sa valeur atteindre zéro, ce qui est synonyme de rejet (avec message ICMP Time Exceeded)
 - La valeur décroît en effet de 1 à chaque traversée de routeur

Structure des adresses IPv6

- **Next Header (8 bits)** : supplante le champ Protocol d'IPv4
 - Il permet de connaître le type de données contenues dans le datagramme (TCP, UDP, ESP, AH...)
 - Les valeurs possibles sont les mêmes qu'en IPv4 (par exemple 6 pour TCP, 17 pour UDP, 58 pour ICMP...)
- **Source et Destination Address (2 fois 128 bits)** : adresses source et destination du datagramme

Structure des adresses IPv6

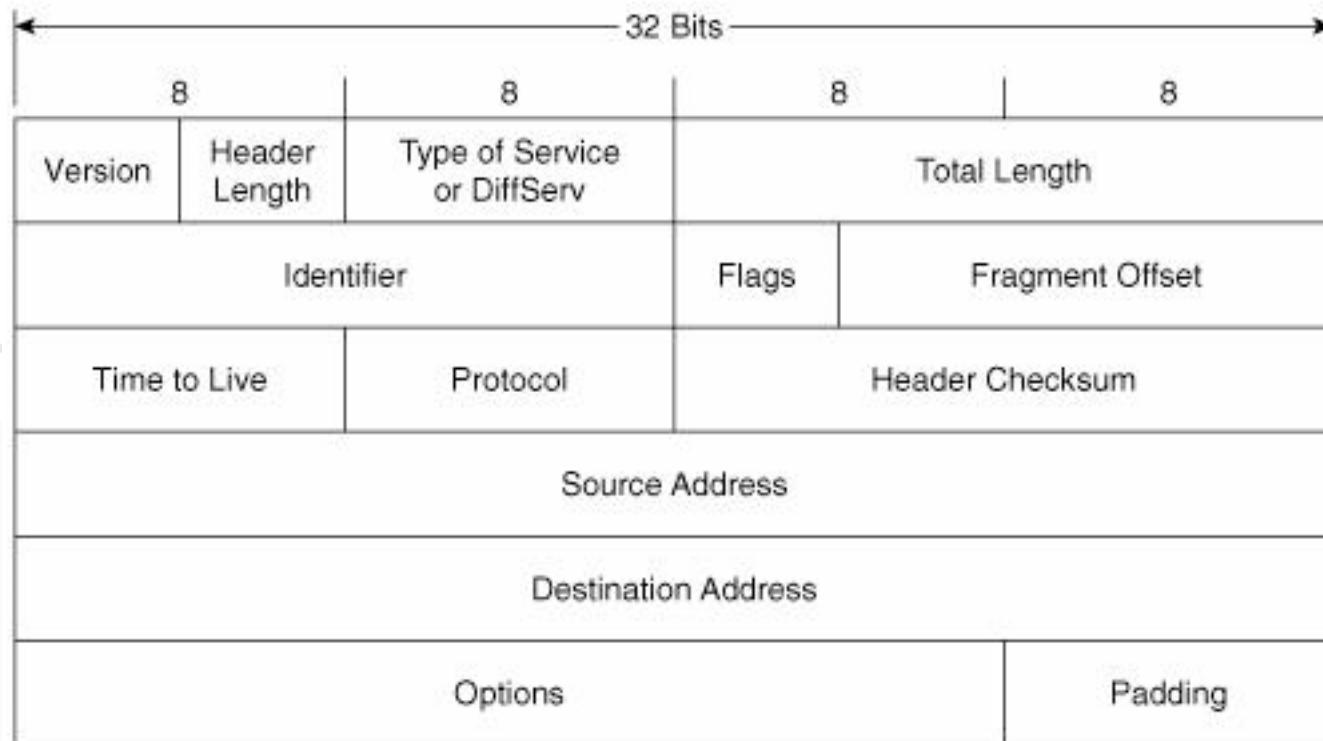
- Après cet en-tête de base, nous allons pouvoir rencontrer deux autres types d'éléments :
 - **Extension Header** : au-delà des 40 octets de l'en-tête de base peuvent être ajoutés des en-têtes spécifiques à des protocoles ou des options
 - Chacun se présente avec une longueur multiple de 8 octets (64 bits) pour faciliter le traitement matériel par les routeurs et les commutateurs
 - Les champs présents varient selon leur nature mais il y a toujours en début d'en-tête, un champ Next Header qui permet de pointer sur l'en-tête suivant
 - Quand on arrive au dernier en-tête, la valeur de Next Header est 59 (No Next Header)
 - Ces en-têtes s'enchaînent donc en cascade (daisy-chain en anglais)

Structure des adresses IPv6

- Après cet en-tête de base, nous allons pouvoir rencontrer deux autres types d'éléments :
 - **Données** (longueur variable) : comme nous l'avons évoqué plus haut, la longueur maximale prise en charge est de 64 Ko (soit 65536 octets), sauf si nous décidons de faire appel à des Jumbograms

Structure des adresses IPv4

- Par comparaison, voici le format d'un paquet IPv4 :



Structure des adresses IPv4

- Rappelons le devenir de chacun des champs qui le composent dans la nouvelle version IPv6:
 - **Version (4 bits)** est identique à celui d'IPv6 si ce n'est que sa valeur est 4 au lieu de 6
 - **Header Length (4 bits)** permet de déterminer en IPv4 à partir d'où commencent les données puisque l'en-tête dans cette version peut varier de 20 à 60 octets en fonction des options présentes
 - Ce champ n'a plus lieu d'être en IPv6
 - **ToS (8 bits)** (Type of Service) a été renommé Traffic Class
 - **Total Length (16 bits)** est renommé Payload Length

Structure des adresses IPv4

- **Identification (16 bits), Flags (3 bits), Fragment Offset (13 bits)** ont été transférés dans l'en-tête Fragment Extension quand la fragmentation est présente
- **Time To Live ou TTL (8 bits)** a été renommé Hop Limit
- **Protocol (8 bits)** a été renommé Next Header
- **Checksum (16 bits)** n'existe plus en IPv6 puisque la fonction de checksum est confiée aux couches supérieures
- **Source et destination adresses (32 bits chacune)** sont les mêmes mais leurs longueurs diffèrent (32 bits pour les adresses en IPv4, 128 pour celles en IPv6)

Structure des adresses IPv6

- A rajouter: entêtes d'extension (présentation des principaux headers)

Structure des adresses IPv6

- A rajouter: MTU (fragmentation des paquets)

Exemples de capture en IPv6

- Pour illustrer à la fois l'adressage et le format des paquets IPv6, voici quelques exemples de paquets capturés lors d'un dialogue entre un poste et une imprimante en IPv6
- D'abord, un simple ping depuis l'adresse fe80::129a:ddff:fe57:90e7 vers l'adresse de notre imprimante fe80::21b:a9ff:fe3a:4066

```
MacMini:~ root# ping6 FE80::21B:A9FF:FE3A:4066%en0
PING6(56=40+8+8 bytes) fe80::129a:ddff:fe57:90e7%en0 -->
fe80::21b:a9ff:fe3a:4066%en0
16 bytes from fe80::21b:a9ff:fe3a:4066%en0, icmp_seq=0 hlim=64
time=964.409 ms
16 bytes from fe80::21b:a9ff:fe3a:4066%en0, icmp_seq=1 hlim=64
time=1.793 ms
```

Exemples de capture en IPv6

- Ce qui, si l'on analyse avec **tshark** les paquets transmis, donne le résultat suivant avec les commentaires en *gras et en italique*

```
Frame 1 (86 bytes on wire, 86 bytes captured)
```

```
Arrival Time: Jan 29, 2012 19:17:36.810489000
```

```
.../...
```

À l'origine, nous cherchons à atteindre une adresse IPv6 précise, mais comme notre poste ne connaît pas l'adresse MAC à utiliser, il est nécessaire de passer par un multicast pour trouver celle-ci.

```
Ethernet II, Src: 10:9a:dd:57:90:e7 (10:9a:dd:57:90:e7), Dst:
```

```
IPv6multicast_ff:3a:40:66 (33:33:ff:3a:40:66)
```

```
    Destination: IPv6multicast_ff:3a:40:66 (33:33:ff:3a:40:66)
```

```
    Address: IPv6multicast_ff:3a:40:66 (33:33:ff:3a:40:66)
```

```
    .../...
```

```
Source: 10:9a:dd:57:90:e7 (10:9a:dd:57:90:e7)
```

```
    Address: 10:9a:dd:57:90:e7 (10:9a:dd:57:90:e7)
```

```
    .../...
```

Exemples de capture en IPv6

Commence alors le décodage du paquet Internet Protocol Version 6

0110 = Version: 6

Puis le champ Traffic Class et Flow label (tous les deux à zéro car pas de traitement particulier nécessaire en termes de priorité ou de services).

Traffic class: 0x00000000

Flowlabel: 0x00000000

Payload length: 32

Le contenu de ce datagramme est bien ICMPv6 - 3a vaut 58 en décimal et nous avons encore la valeur maximale pour la limite de sauts puisque nous n'avons pas traversé de routeur

Next header: ICMPv6 (0x3a)

Hop limit: 255

Puis viennent les adresses source et destination en format ipv6

Source: fe80::129a:ddff:fe57:90e7 (fe80::129a:ddff:fe57:90e7)

Destination: ff02::1:ff3a:4066 (ff02::1:ff3a:4066)

Exemples de capture en IPv6

et enfin le contenu du paquet ICMP qui consiste ici à trouver le voisin recherché

Internet Control Message Protocol v6

Type: 135 (**Neighbor solicitation**)

Code: 0

Checksum: 0x528e [correct]

Target: fe80::21b:a9ff:fe3a:4066 (fe80::21b:a9ff:fe3a:4066)

ICMPv6 Option (Source link-layer address)

Type: Source link-layer address (1)

Length: 8

Link-layer address: 10:9a:dd:57:90:e7

Exemples de capture en IPv6

Puis vient la réponse de l'imprimante

Frame 2 (86 bytes on wire, 86 bytes captured)

.... / ...

avec l'adresse Ethernet de la carte réseau de notre imprimante Brother

Ethernet II, Src: BrotherI_3a:40:66 (00:1b:a9:3a:40:66), Dst:

10:9a:dd:57:90:e7 (10:9a:dd:57:90:e7)

 Destination: 10:9a:dd:57:90:e7 (10:9a:dd:57:90:e7)

 Address: 10:9a:dd:57:90:e7 (10:9a:dd:57:90:e7)

.... . . . 0 = IG bit: Individual address
(unicast)

.... . . 0 = LG bit: Globally unique
address (factory default)

Source: BrotherI_3a:40:66 (00:1b:a9:3a:40:66)

.... / ...

Type: IPv6 (0x86dd)

Internet Protocol Version 6

.... / ...

Source: fe80::21b:a9ff:fe3a:4066 (fe80::21b:a9ff:fe3a:4066)

Destination: fe80::129a:ddff:fe57:90e7

(fe80::129a:ddff:fe57:90e7)

Exemples de capture en IPv6

Ici, le voisin (en l'occurrence l'imprimante) annonce sa présence
Internet Control Message Protocol v6

Type: 136 (**Neighbor advertisement**)

ainsi que son adresse physique sur le lien (adresse MAC)

Link-layer address: 00:1b:a9:3a:40:66

Nous pouvons alors voir la demande d'écho liée au ping

Frame 3 (70 bytes on wire, 70 bytes captured)

Arrival Time: Jan 29, 2012 19:17:36.811205000

.../...

Internet Protocol Version 6

0110 = Version: 6

.../...

Internet Control Message Protocol v6

Type: 128 (Echo request)

Code: 0

Checksum: 0x113b [correct]

ID: 0x3c26

Sequence: 0x0000

Data (8 bytes)

0000 4f 25 8d 3f 00 0c ee 4b

Data: 4F258D3F000CEE4B

0%..?....K

Exemples de capture en IPv6

et la réponse de l'imprimante

```
Frame 4 (70 bytes on wire, 70 bytes captured)
    ...
Ethernet II, Src: BrotherI_3a:40:66 (00:1b:a9:3a:40:66), Dst:
    10:9a:dd:57:90:e7 (10:9a:dd:57:90:e7)
        Destination: 10:9a:dd:57:90:e7 (10:9a:dd:57:90:e7)
    ...
        Source: BrotherI_3a:40:66 (00:1b:a9:3a:40:66)
        ...
Internet Protocol Version 6
    0110 .... = Version: 6
    ...
        Next header: ICMPv6 (0x3a)
        Hop limit: 64
        Source: fe80::21b:a9ff:fe3a:4066 (fe80::21b:a9ff:fe3a:4066)
        Destination: fe80::129a:ddff:fe57:90e7 (fe80::129a:ddff:fe57:90e7)
Internet Control Message Protocol v6
    Type: 129 (Echo reply)
    Code: 0
    Checksum: 0x103b [correct]
    ID: 0x3c26
    Sequence: 0x0000
    Data (8 bytes)

0000  4f 25 8d 3f 00 0c ee 4b
        Data: 4F258D3F000CEE4B
    0% .?...K
```

Mise en œuvre d'IPv6 sous CentOS7

- Deux fichiers existent pour la configuration d'IPv6 :
 - `/etc/sysconfig/network` : qui régit les paramètres réseau du système d'une façon globale
 - `/etc/sysconfig/network-scripts/ifcfg-nom_interface` : qui régit les paramètres propres à une interface
 - Nous pouvons également régler les paramètres du noyau avec les commandes **sysctl** ou le fichier `sysctl.conf`

Mise en œuvre d'IPv6 sous CentOS7

- Si IPv6 n'est pas déjà activé sur le système, il faut donc le faire en ajoutant la ligne NETWORKING_IPV6=yes au fichier /etc/sysconfig/network
- Cela suffit à activer IPv6 de façon globale sur le système, et donc à permettre une autoconfiguration de ce dernier
- Il peut néanmoins être utile d'ajouter des lignes de configuration telles qu'une passerelle par défaut avec l'instruction IPV6_DEFAULTGW= *adresseipv6*

Mise en œuvre d'IPv6 sous CentOS7

- Le fichier /etc/sysconfig/network pourrait ressembler à cela :

NETWORKING=yes

HOSTNAME=garcia.isat.lan

GATEWAY=192.168.10.1

NETWORKING_IPV6=Yes

IPV6_DEFAULTGW= 2001:41D0:2:657d:0:185:9:2017

Mise en œuvre d'IPv6 sous CentOS7

- Dans le fichier /etc/sysconfig/network-scripts/enp0s8, on pourrait trouver:

DEVICE=enp0s8

ONBOOT=yes

BOOTPROTO=static

HWADDR=00:1C:42:5A:BD:F1

IPADDR=192.168.10.185

NETMASK=255.255.255.0

IPV6INIT=yes

IPV6ADDR=2001:41D0:2:657d:0:185:9:1234/64

Mise en œuvre d'IPv6 sous CentOS7

- En réalité, certains de ces paramètres peuvent être placés dans l'un ou l'autre des fichiers en fonction de la portée qu'on veut leur donner (globale ou propre à une interface)
- Pour que ces modifications soient prises en compte, il faut généralement relancer le service réseau par la commande **/sbin/service network restart**

Mise en œuvre d'IPv6 sous CentOS7

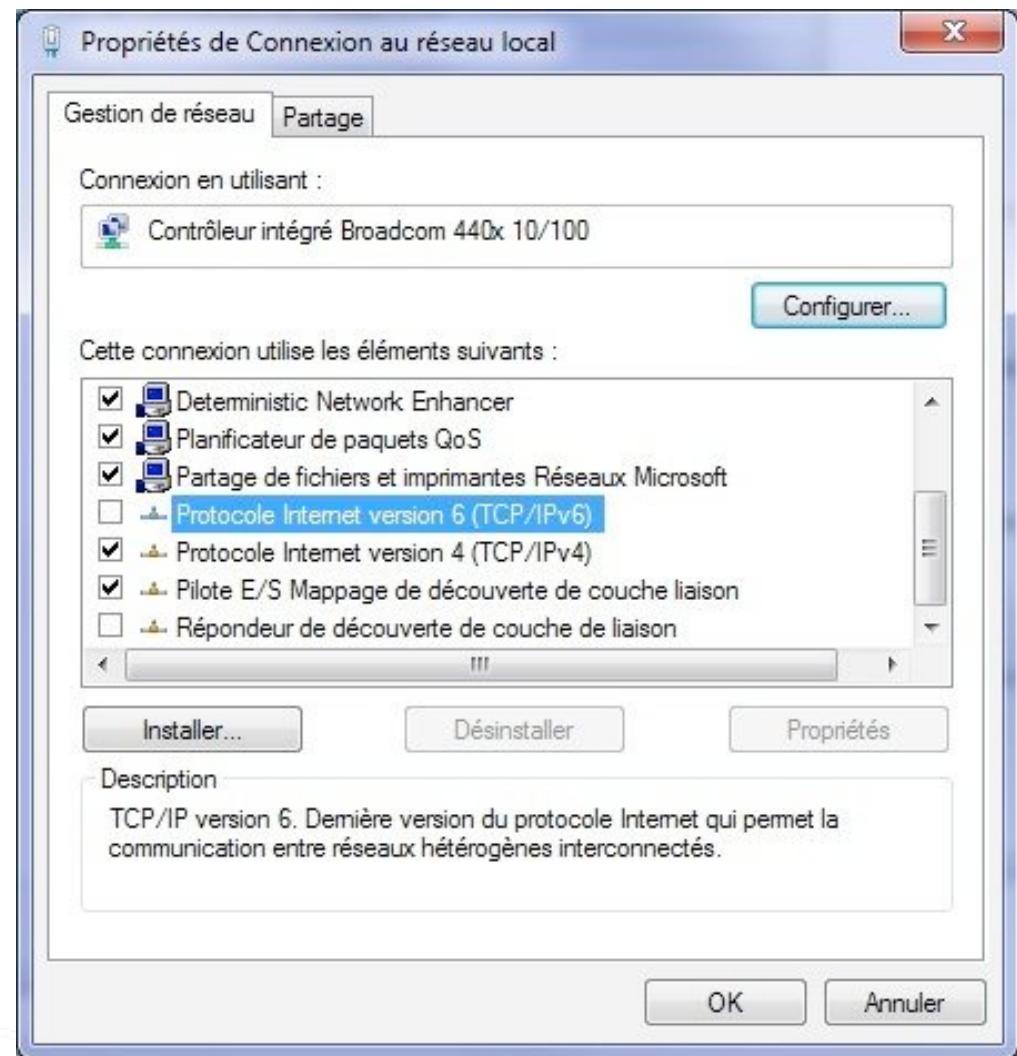
- Nous pouvons ajouter de façon temporaire une adresse à une interface, avec la commande :
ip -6 addr add 2001:41d0:1008:2697::167/64
- Nous pouvons ensuite vérifier sa présence par la commande :
ip -6 addr
- De la même façon, nous pouvons ajouter une route par défaut de façon temporaire par la commande **ip -6 route add default** ou **ip -6 route add ::0**
Exemple:
ip -6 route add default via fe80::223:f8ff:fe15:bac8 dev eth0

Mise en œuvre d'IPv6 sous CentOS7

- Il est possible aussi d'ajouter une route vers un réseau précis comme dans l'exemple ci-dessous dans lequel nous passons par un autre routeur situé sur le lien local pour atteindre notre objectif :
ip -6 route add 2001:41d0:1:1234::/64 via fe80::223:f8ff:fe15:5678 dev enp0s3
- La commande **ip -6 route** permet d'obtenir la table de routage
- La commande **ping6** permet de tester une adresse IPv6
Exemple: **ping6 2a01:240:fe00:1c6::1**

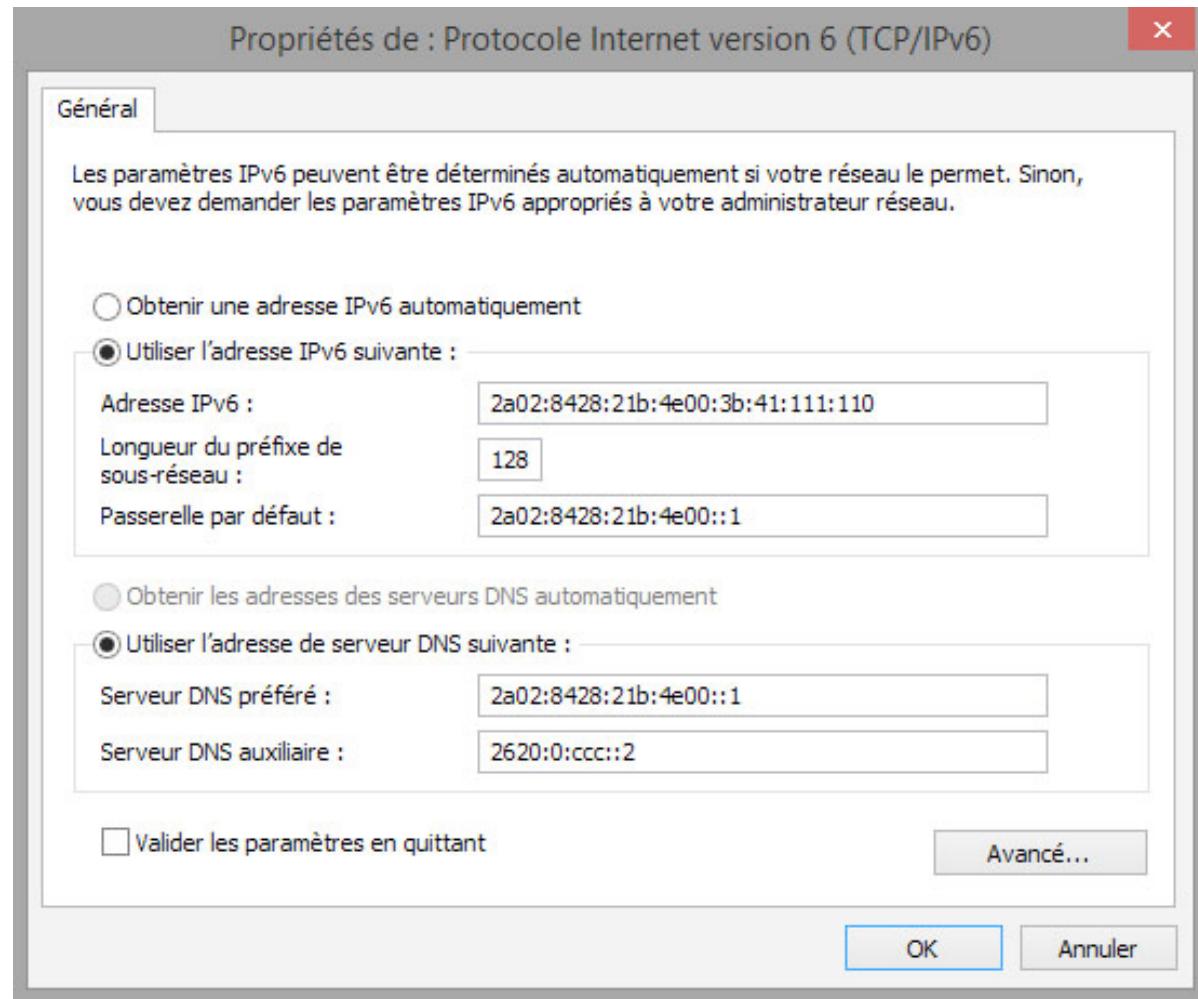
Mise en œuvre d'IPv6 sous Windows

- IPv6 est présent de base sur Windows 7, 8 et 10
- Il est par défaut en autoconfiguration mais il est possible via la fenêtre de propriétés des cartes réseau de le désactiver ou de modifier manuellement les paramètres



Mise en œuvre d'IPv6 sous Windows

- Ecran de réglage IPv6



Mise en œuvre d'IPv6 sous Windows

- Au niveau des commandes, on trouve la commande **netsh interface ipv6 show** qui permet d'avoir divers renseignements
- Exemples:
netsh int ipv6 show interface permet d'afficher les interfaces avec leur numéro d'index et leur MTU

netsh interface ipv6 show neighbors permet de découvrir les voisins sur les différentes interfaces
C'est un peu l'équivalent de la commande **arp -a** d'IPv4

Mise en œuvre d'IPv6 sous Windows

netsh int ipv6 show address affiche non seulement les adresses présentes sur les différentes interfaces mais aussi leur durée de vie et le type d'assignation : automatique ou manuelle

netsh int ipv6 show route se charge d'afficher les routes mais aussi pour chacune d'elles l'interface de sortie et la métrique

netsh int ipv6 show destinationcache affiche la table des routes effectives mises en cache avec pour chacune le prochain saut à utiliser ainsi que le Path MTU en vigueur

Mise en œuvre d'IPv6 sous Windows

- La plupart de ces commandes permettent de préciser l'interface sur laquelle porte notre demande, ce qui modifie légèrement la sortie pour certaines d'entre elles
- Pour cela nous pouvons employer le nom ou, ce qui est plus simple, l'index désignant l'interface

Exemple: **netsh int ipv6 show address 12**

Mise en œuvre d'IPv6 sous Windows

- Les versions les plus récentes de Windows (8,10, 2012 et 2016) disposent d'un ensemble de commandes PowerShell pour remplacer la commande **netsh**
- Cette dernière fonctionne encore sur ces versions mais elle est maintenant considérée comme obsolète
- Il est possible d'obtenir la liste des commandes disponibles par **gcm -module nettcpip**

Mise en œuvre d'IPv6 sous Windows

- Voici quelques-unes de ces commandes:

get-netipinterface affiche les propriétés des différentes interfaces IPv6 ou IPv4, dont l'Index et le MTU

get-netipinterface -addressfamily ipv6 affiche les propriétés des seules interfaces avec IPv6, dont l'Index et le MTU

get-netipaddress -addressfamily ipv6 affiche chaque adresse IPv6 avec ses caractéristiques détaillées

Mise en œuvre d'IPv6 sous Windows

get-netneighbor -addressfamily ipv6 affiche tous les voisins IPv6, y compris les groupes multicast, avec leur état et l'index des interfaces où ils résident

get-netroute -addressfamily ipv6 permet d'afficher la table de routage IPv6

Mise en œuvre d'IPv6 sous Windows

- Au niveau des commandes **ping** et **tracert**, Il existe maintenant une option **-4** ou **-6** à préciser pour forcer à utiliser une version IPv4 ou IPv6 de la commande
- Sinon la commande utilise le protocole qui lui semble le plus pertinent, souvent IPv6 quand cela est possible

Mise en œuvre d'IPv6 sous Cisco

- Pour activer IPv6 sur l'ensemble d'un routeur, il suffit de passer la commande **ipv6 unicast-routing**, ce qui permet le routage des paquets IPv6 entre les différentes interfaces du routeur
- Ensuite, la commande **ipv6 enable** au niveau d'une interface permet de passer celle-ci en mode autoconfiguration avec donc une attribution automatique d'une adresse de type link-local
- Si nous souhaitons affecter une adresse précise (locale ou non), il faudra utiliser la commande **ipv6 address**
Exemple: **ipv6 addr 2a01:240:fedd:2017::/64 eui-64**

Mise en œuvre d'IPv6 sous Cisco

- **ipv6 neighbor discovery** permet de définir le comportement de notre routeur en matière d'annonces et de découverte des voisins
- **ipv6 nd ?** permet de voir la liste des possibilités
- **show ipv6 int f0/0** permet de vérifier le bon paramétrage d'IPv6
- **sh ipv6 traffic** permet de visualiser un résumé du trafic IPv6

Mise en œuvre d'IPv6 sous Cisco

- **show ipv6 neighbors** permet d'afficher l'état des voisins
- Les commandes **debug ipv6 packet** et **debug ipv6 nd** (entre autres) sont disponibles pour nous aider à diagnostiquer d'éventuels problèmes
- Au niveau des switchs, les commandes sont essentiellement les mêmes que pour les routeurs

Mise en œuvre d'IPv6

- **Pour aller plus loin**
- Voici quelques liens, parmi des centaines, permettant d'obtenir plus d'informations sur le paramétrage IPv6 de certains OS ou matériels
- <http://guide.ovh.com/Ipv4Ipv6> qui balaie de nombreuses versions de Linux
- <http://www.freebsd.org/doc/fr/books/handbook/network-ipv6.html> pour FreeBSD
- http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/12_4/ipv6_12_4_book.html et http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html pour les routeurs Cisco (pour une des dernières versions)
- <http://www.centos.org/docs/5/> pour Centos
- <http://wiki.debian.org/DebianIPv6> pour Debian

Autoconfiguration en mode stateless

- Cette possibilité de configuration automatique et autonome d'un poste dès son branchement sur le réseau est une des grandes fonctionnalités apportées par IPv6
- Elle pourrait favoriser son adoption en dehors de toute nécessité liée à la pénurie d'adresse
- En effet, aucun serveur n'est nécessaire (notamment pas de serveur DHCP) pour que cette configuration puisse être menée à bien

Autoconfiguration en mode stateless

- Par contre, les routeurs (ou firewalls faisant fonction de routeurs) peuvent participer à cette configuration
- Ce mécanisme s'applique aux postes au sens large (PC, serveurs, imprimantes, PDA...) à l'exclusion des routeurs qui doivent être configurés manuellement (à l'exception éventuellement de leurs adresses de type link-local qui peuvent faire l'objet d'une autoconfiguration) comme nous le verrons plus loin

Autoconfiguration en mode stateless

Calcul de l'adresse

- La première opération que doit faire un poste pour se configurer est de déterminer son adresse IPv6
- Comme nous l'avons déjà vu, celle-ci se compose généralement d'un préfixe sur 64 bits et de l'identifiant d'interface sur les 64 bits restants

Autoconfiguration en mode stateless

Calcul de l'adresse

- Dans le cas d'une adresse de type link-local, le préfixe est toujours fe80::/
- Il ne reste plus alors qu'à dériver l'identifiant d'interface à partir de l'adresse MAC de l'interface concernée
- De cette façon, nous sommes quasiment certains que l'adresse résultante sera unique puisque l'adresse MAC est unique car fixée par le constructeur de l'interface (sauf modification faite manuellement par l'utilisateur)

Autoconfiguration en mode stateless

Calcul de l'adresse

- L'adresse MAC ayant une longueur de seulement 48 bits, il est nécessaire d'ajouter 2 octets supplémentaires
- Cela se fait en insérant la chaîne fffe (réservée par l'IEEE à cet usage) au milieu de ces 48 bits
- Ainsi, l'adresse MAC d'un de nos postes étant 10:93:e9:0f:00:18, nous obtenons l'adresse provisoire 1093:e9ff:fe0f:0018 (en utilisant la présentation par 2 octets classique en IPv6)

Autoconfiguration en mode stateless

Calcul de l'adresse

- Nous parlons d'adresse provisoire car un dernier changement va être nécessaire
- En effet, pour distinguer les adresses MAC uniques (celles qui n'ont donc pas été modifiées par l'utilisateur, ce qui est le cas le plus courant) de celles qui pourraient ne pas être uniques, nous allons avoir recours au 7^e bit de l'octet le plus à gauche de cette adresse provisoire
- Nous lui attribuerons la valeur 1 si l'adresse est unique et 0 dans le cas contraire

Autoconfiguration en mode stateless

Calcul de l'adresse

- Ainsi, l'octet 10 (valeur binaire 0001 0000) de notre adresse devient 12 (0001 0010)
- L'identifiant d'interface résultant de ce calcul est donc 1293:e9ff:fe0f:18, ce qui conduit à l'adresse IPv6 fe80::1293:e9ff:fe0f:18

Autoconfiguration en mode stateless

Calcul de l'adresse

```
MacBookAir-JPA:~ root# ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    inet 127.0.0.1 netmask 0xff000000
    inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu
1500
    ether 10:93:e9:0f:00:18
    inet6 fe80::1293:e9ff:fe0f:18%en0 prefixlen 64 scopeid 0x4
        inet 192.168.194.104 netmask 0xffffffff broadcast
192.168.194.255
    media: autoselect
    status: active
.....
MacBookAir-JPA:~ root#
```



- Nous pouvons noter le suffixe %en0 à la fin de l'adresse IPv6. Cela rappelle à quelle interface cette adresse est rattachée
- Un poste en autoconfiguration dispose toujours d'une adresse Link-Local

Autoconfiguration en mode stateless

Calcul de l'adresse

- Pour ce qui concerne les adresses globales, le même mécanisme d'autoconfiguration s'applique pour obtenir les 64 bits de droite (ceux correspondant à l'identifiant d'interface)
- Par contre, le préfixe s'obtient par écoute des diffusions de préfixes effectuées sur le lien puis il sera combiné à l'identifiant pour aboutir à l'adresse IPv6 globale

Autoconfiguration en mode stateless

Calcul de l'adresse

- Cela nous donne alors, par exemple pour l'interface eth0 d'un de nos serveurs, deux adresses IPv6, l'une de type link-local et l'autre de type global unicast, toutes les deux ayant le même identifiant d'interface :

```
[root@pluto admin]# ifconfig -a
eth0      Link encap:Ethernet    HWaddr 00:16:3E:FD:62:CC
          inet addr:173.246.101.46   Bcast:173.246.103.255
          Mask:255.255.252.0
                  inet6 addr: 2604:3400:dc1:41:216:3eff:fed:62cc/64
          Scope:Global
                  inet6 addr: fe80::216:3eff:fed:62cc/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500 Metric:1
          RX packets:182 errors:0 dropped:0 overruns:0 frame:0
          TX packets:108 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:41123 (40.1 KiB)  TX bytes:15956 (15.5 KiB)
          Interrupt:9
...
[root@pluto admin]#
```

Autoconfiguration en mode stateless

Calcul de l'adresse

- Ces préfixes globaux sont généralement diffusés par des routeurs puisque ce sont ces équipements qui vont permettre le routage à l'extérieur du site
- Il reste ensuite deux éléments à intégrer pour que la configuration réseau du poste soit complète : la passerelle (routeur) par défaut et le(s) serveur(s) DNS
- Les routeurs participant à un réseau IPv6 peuvent s'annoncer sur le lien et les postes n'ont plus qu'à écouter les annonces « publiées » sur le lien pour les intégrer à leurs configurations

Autoconfiguration en mode stateless

Vérification de l'unicité de l'adresse

- Avant toute affectation d'une adresse unicast à une interface et quelle que soit l'étendue de cette adresse, le poste doit s'assurer que cette adresse n'est pas déjà utilisée sur le lien
- Pour cela, il a recours au mécanisme de découverte des voisins (Neighbor Discovery) qui n'est pas sans rappeler l'ARP gratuit (Gratuitous ARP) intervenant en IPv4

Autoconfiguration en mode stateless

Vérification de l'unicité de l'adresse

- Mais ici, ce mécanisme passera par des messages ICMPv6
- Dans le chapitre ICMPv6, nous reviendrons plus en détail sur le fonctionnement d'ICMPv6 mais nous introduisons dès maintenant les types de messages nécessaires au bon fonctionnement de l'autoconfiguration

Autoconfiguration en mode stateless

Vérification de l'unicité de l'adresse

- a. **Envoi d'un message de type Neighbor Solicitation**
 - Le principe employé est d'envoyer un message de type multicast (puisque nous ne pouvons préjuger de l'adresse physique du nœud ayant potentiellement la même adresse que celle de notre interface en cours de paramétrage) ayant pour cible l'adresse pressentie afin d'observer si nous obtenons une réponse
 - Si c'est le cas, c'est qu'un autre nœud possède déjà notre adresse et l'autoconfiguration ne peut alors aboutir
 - Dans le cas contraire, c'est que notre adresse est probablement unique (au moins au moment de notre demande) et que nous pouvons donc l'affecter à l'interface

Autoconfiguration en mode stateless

Vérification de l'unicité de l'adresse

- Par contre, avant d'envoyer ce message et afin de pouvoir recevoir les réponses éventuelles à ce trafic multicast, il faudra que notre interface se joigne à deux groupes multicast :
 - l'adresse de diffusion à tous les nœuds (adresse FF02::1)
 - l'adresse solicited-node multicast dérivée de l'adresse cible recherchée

Autoconfiguration en mode stateless

Vérification de l'unicité de l'adresse

- Si nous n'avons pas obtenu de réponse après plusieurs requêtes, c'est que notre adresse potentielle (fe80::a03d:d71:9555:20ef dans l'exemple ci-dessus) n'est pas utilisée et peut donc être affectée à l'interface
- Ce mécanisme est en fonction pour toutes les adresses qu'elles soient de type link-local ou global

Autoconfiguration en mode stateless

Vérification de l'unicité de l'adresse

- **b. Réponse reçue si adresse déjà existante**
 - Si un autre élément sur le réseau possède déjà l'adresse demandée, nous allons recevoir un message « Neighbor Advertisement »
 - Tout d'abord, voici le message de sollicitation provenant d'un système cherchant à acquérir l'adresse globale 2a01:240:fedd:2015::1 :

```
Frame 11274: 78 bytes on wire (624 bits), 78 bytes captured  
(624 bits) on interface 0  
.../  
Ethernet II, Src: Shuttle_9d:08:98 (80:ee:73:9d:08:98),  
Dst: IPv6mcast_ff:00:00:01 (33:33:ff:00:00:01)  
Destination: IPv6mcast_ff:00:00:01 (33:33:ff:00:00:01)  
.../  
Internet Control Message Protocol v6  
Type: Neighbor Solicitation (135)  
Code: 0  
Checksum: 0x2f72 [correct]  
[Checksum Status: Good]  
Reserved: 00000000  
Target Address: 2a01:240:fedd:2015::1  
.../...
```

Autoconfiguration en mode stateless

Vérification de l'unicité de l'adresse

- Puis le message parvenant d'un autre système ayant déjà l'adresse demandée :

```
Ethernet II, Src: Watchgua_9e:c0:f7 (00:90:7f:9e:c0:f7),  
Dst: IPv6mcast_01 (33:33:00:00:00:01)  
.../  
    Source: Watchgua_9e:c0:f7 (00:90:7f:9e:c0:f7)  
    Address: Watchgua_9e:c0:f7 (00:90:7f:9e:c0:f7)  
.../  
Internet Protocol Version 6, Src: 2a01:240:fedd:2015::1, Dst: ff02::1  
.../  
Internet Control Message Protocol v6  
    Type: Neighbor Advertisement (136)  
    Code: 0  
    Checksum: 0xff0e [correct]  
        [Checksum Status: Good]  
    Flags: 0xa0000000  
.../  
    Target Address: 2a01:240:fedd:2015::1  
    ICMPv6 Option (Target link-layer address : 00:90:7f:9e:c0:f7)  
        Type: Target link-layer address (2)  
        Length: 1 (8 bytes)  
        Link-layer address: Watchgua_9e:c0:f7 (00:90:7f:9e:c0:f7)  
.../
```

Autoconfiguration en mode stateless

Vérification de l'unicité de l'adresse

- Ce message fournit aussi, comme nous le voyons dans la capture, l'adresse MAC du poste possédant actuellement l'adresse pressentie
- Il faut noter que des « Neighbor Advertisement » non sollicités peuvent également intervenir lorsqu'un poste change une de ses adresses IPv6

Autoconfiguration en mode stateless

Mise en œuvre

- La mise en œuvre est généralement des plus simples puisqu'il n'y a le plus souvent rien à faire sur les matériels récents à part activer IPv6, quand ce n'est pas fait par défaut
- C'est notamment le cas des postes de travail

Autoconfiguration en mode stateless

Mise en oeuvre

- Par contre, pour les routeurs ou les firewalls, pour lesquels l'autoconfiguration de leurs propres IP n'est pas l'option la plus courante ni la plus logique pour les adresses globales, il est souvent nécessaire d'activer l'option stateless de façon explicite (instructions ou cases à cocher)

Autoconfiguration en mode stateless

Mise en oeuvre

- **a. Postes**
 - Par défaut, la simple activation d'IPv6 dans les réglages réseaux permet l'autoconfiguration, notamment en tenant compte des préfixes qui sont diffusés par les routeurs présents sur le réseau

Autoconfiguration en mode stateless

Mise en oeuvre

- **b. Routeurs Cisco**
 - Pour qu'une interface physique ou de type vlan s'autoconfigure sur les routeurs Cisco, il faut utiliser en plus de **ipv6 enable**, la commande **ipv6 address autoconfig** comme dans l'extrait de configuration ci-dessous
 - Sinon, il y a juste autoconfiguration de l'adresse link-local

```
interface Vlan88
  ipv6 address autoconfig
  ipv6 enable
```

Autoconfiguration en mode stateless

Mise en oeuvre

- Et nous obtenons alors les adresses suivantes :

```
R100#sh ipv6 int vlan88
Vlan88 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::FE99:47FF:FEFA:E7E
  No Virtual link-local address(es):
  Stateless address autoconfig enabled
  Global unicast address(es):
    2A01:240:FEDD:2015:FE99:47FF:FEFA:E7E, subnet is
    2A01:240:FEDD:2015::/64 [EUI/CAL/PRE]
      valid lifetime 21400 preferred lifetime 7000
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FFFA:E7E
  MTU is 1500 bytes
  .../...
R100#
```

Autoconfiguration en mode stateless

Avantages

- Il est évident qu'une autoconfiguration gérée de manière complètement autonome permet une grande simplicité dans l'administration d'un réseau local au quotidien puisqu'il suffit de brancher les postes pour qu'ils puissent accéder aux ressources de ce réseau
- Cette autoconfiguration facilite également grandement une renumérotation éventuelle du réseau, par exemple en cas de changement de fournisseur d'accès Internet

Autoconfiguration en mode stateless

Inconvénients et risques

- Nous pouvons néanmoins y trouver un premier inconvénient, en raison de la difficulté à maîtriser les paramètres adoptés (adresses, routeurs, DNS...)
- Nous pouvons aussi évoquer les attaques potentielles facilitées par les mécanismes d'autoconfiguration et plus particulièrement ceux liés aux découvertes/annonces des voisins et des routeurs

Autoconfiguration en mode stateless

Inconvénients et risques

- Parmi ces menaces, nous pouvons mentionner par exemple :
 - Des dénis de service basés sur une réponse systématique aux sollicitations des voisins avant la mise en œuvre d'une adresse
Cela empêchera alors toute affectation d'adresses aux postes légitimes
 - Des usurpations d'adresses permettant par exemple de définir un faux routeur ou bien de mettre hors d'état le routeur officiel
 - Des attaques par inondation

Autoconfiguration en mode stateless

Inconvénients et risques

- Il est quand même utile de préciser que les attaques évoquées nécessitent d'avoir un accès physique au lien
- Le risque est donc plus grand sur un réseau difficile à contrôler comme un réseau Wi-Fi ou un réseau extérieur que sur le réseau local

Autoconfiguration en mode stateless

Diffusions effectuées par les routeurs

- Chaque routeur situé sur un lien envoie régulièrement (par défaut toutes les 5 minutes) les informations permettant aux postes de compléter leur configuration
- Parmi ces informations figurent leur adresse et le ou les préfixes vers lesquels ils peuvent router du trafic
- Ces annonces sont diffusées vers l'adresse de multicast ff02::1 (tous les nœuds du lien)

Autoconfiguration en mode stateless

Diffusions effectuées par les routeurs

- Il faut noter que toutes les notifications, contrairement à ce qui se passe en DHCP IPv4, sont prises en compte immédiatement
- Donc si un nouveau préfixe ou un nouveau routeur est annoncé, tous les postes situés sur le même lien l'incorporent de suite à leur configuration

Autoconfiguration en mode stateless

Diffusions effectuées par les routeurs

- Si un poste démarre (ou se connecte sur le réseau), il ne va pas attendre qu'un routeur s'annonce puisque cela peut prendre jusqu'à cinq minutes (si ce sont les valeurs par défaut qui sont conservées), donc il envoie un message de type « Router Solicitation » à l'adresse de multicast spécifiant tous les routeurs du lien (FF02::2)
- Ce message est également un message de type ICMPv6

Autoconfiguration en mode stateless

Diffusions effectuées par les routeurs

- Sur un routeur Cisco, le simple fait d'avoir activé IPv6 et paramétré une adresse de type global ou de type unique local, active la diffusion des avertissements de routeurs qui vont permettre aux postes de s'autoconfigurer
- Nous pouvons visualiser le réglage actuel de ces diffusions pour une interface donnée avec la commande
sh ipv6 int *nom_interface*

Autoconfiguration en mode stateless

Découverte et gestion des voisins (Neighbor Discovery)

- La découverte des voisins se fait soit en testant activement leur présence en envoyant un message de type Neighbor Solicitation et en observant les réponses, soit par écoute des annonces Neighbor Advertisement diffusées spontanément par les voisins
- Il existe deux caches utilisés pour situer des postes en IPv6 et communiquer avec eux : Neighbor cache et Destination cache

Autoconfiguration en mode stateless

Découverte et gestion des voisins (Neighbor Discovery)

- a. **Neighbor cache**
 - Ce cache contient les adresses des voisins vers lesquelles du trafic a été récemment émis. Il est donc alimenté par toutes les connexions IPv6 émanant du poste et contient les informations suivantes pour chaque entrée présente :
 - L'adresse unicast locale des interfaces des voisins connectés sur le même lien
 - L'adresse MAC (généralement Ethernet) de ces interfaces
 - Un indicateur (drapeau) indiquant le type de nœud (routeur)
 - Le statut de l'adresse

Autoconfiguration en mode stateless

Découverte et gestion des voisins (Neighbor Discovery)

- Les statuts possibles sont :
 - INCOMPLETE : une résolution d'adresse est en cours mais n'a pas encore abouti
 - REACHABLE : le voisin a pu être contacté très récemment (moins d'une dizaine de secondes)
 - STALE : on ne sait pas si le voisin est atteignable et, tant qu'aucun trafic n'est à envoyer vers lui, aucun test d'accessibilité ne doit être lancé
 - DELAY : on ne sait pas si le voisin est atteignable et du trafic a été acheminé vers lui. Pour l'instant, les tests par sollicitation sont différés pour permettre aux couches de niveau supérieur de vérifier éventuellement l'accessibilité de ce voisin
 - PROBE : on ne sait pas si le voisin est atteignable et des tests d'accessibilité via des messages Neighbor Solicitation vont être effectués

Autoconfiguration en mode stateless

Découverte et gestion des voisins (Neighbor Discovery)

- **b. Destination cache**
 - Ce cache est un peu différent dans son objectif puisqu'il ne répertorie pas seulement les voisins accessibles sur le lien mais aussi tous ceux qui le sont au travers de liens externes
 - Le Neighbor cache est donc un sous-ensemble de ce cache

Autoconfiguration en mode stateless

Découverte et gestion des voisins (Neighbor Discovery)

- Sous Linux, les commandes pour gérer les voisins sont:
 - **ip -6 neighbour show** qui permet d'afficher les voisins découverts
 - **ip -6 neighbour add *adresse_ipv6* lladdr *adresse_MAC* dev *nom_interface*** permet d'ajouter des entrées (par défaut permanentes)
- Exemple: **ip -6 neighbour add fe80::223:f8ff:fe15:1234 lladdr 10:12:22:33:44:56 dev eth0**
- **ip -6 neighbour del *adresse_ipv6* lladdr *adresse_MAC* dev *nom_interface*** permet de les supprimer

Autoconfiguration en mode stateless

Découverte et gestion des voisins (Neighbor Discovery)

- Sous windows,
 - **netsh interface ipv6 show neighbours** permet d'afficher le Neighbor Cache
 - **netsh interface ipv6 show destinationcache** permet d'afficher le Destination Cache

Autoconfiguration en mode stateless

Découverte et gestion des voisins (Neighbor Discovery)

- Sous Cisco,
 - `show ipv6 neighbors [interface]` affiche le cache
 - `clear ipv6 neighbors` le vide

ICMPv6

Introduction

- Rappelons que ICMP signifie *Internet Control Message Protocol*
- Son rôle, aussi bien en version 6 qu'en version 4, est donc bien de véhiculer non pas des données utilisateur mais des informations permettant de gérer les communications entre les différents composants d'un réseau (postes, routeurs, imprimantes, switches...)
- Le RFC 4443 de mars 2006 décrit le fonctionnement d'ICMPv6

ICMPv6

Format des messages

- Il existe deux grandes catégories de messages ICMP : les messages d'erreur et les messages d'information
- Le format général est le suivant :
 - **Type** sur un octet
 - **Code** sur un octet
 - **Checksum** (somme de contrôle) sur deux octets
 - **Message Body** (corps du message) de taille variable selon le message

ICMPv6

Format des messages

- La taille totale d'un paquet ICMP ne doit pas excéder la taille minimale du MTU (*Maximum Transmission Unit*) qui est de 1280 octets en IPv6
- Cela assure une bonne transmission du message, sans aucune fragmentation, dans toutes les circonstances, ce qui est primordial pour des messages de contrôle

ICMPv6

Champs Type et Code

- Ci-après, quelques valeurs des champs Type et Code différenciant les messages
- Il faut noter que pour les messages de type erreur, ce champ va de 0 à 127, alors que pour les messages d'information, il va de 128 à 255 (cela correspond en fait à la valeur du bit le plus à gauche qui est de 0 pour les erreurs et de 1 pour les informations)

ICMPv6

Format des messages

Type	Message	Code
1	Destination Unreachable	0 - no route 1 - communication administratively prohibited
2	Packet Too Big	0
3	Time Exceeded	0 - hop limit exceeded 1 - fragmentation reassembly time exceeded
4	Parameter Problem	0 - erroneous header field 1 - next header unknown 2 - IPv6 option unknown
100	Private experimentation	
101	Private experimentation	

ICMPv6

Format des messages

Type	Message	Code
128	Echo request	
129	Echo Reply	
130	Multicast Listener Query	
131	Multicast Listener Report	
132	Multicast Listener Done	
133	Router Solicitation	
134	Router Advertisement	
135	Neighbor Solicitation	
136	Neighbor Advertisement	
137	Redirect Message	
138	Router Renumbering	0 - router renumbering command 1 - router renumbering result 255 - sequence number reset

ICMPv6

Format des messages

Type	Message	Code
151	Multicast Router Advertisement	
152	Multicast Router Solicitation	
153	Multicast Router Termination	

ICMPv6

Multicast Listener Discovery (MLD)

- Ce protocole, décrit dans le RFC 2710 pour sa version 1 et RFC 3810 pour sa version 2, vise principalement à informer les routeurs présents sur un lien des postes en écoute, pour un groupe de multicast donné, sur ce même lien
- Le but visé est que seuls les routeurs ayant des auditeurs avérés pour un groupe diffusent les informations qui leurs sont destinées

ICMPv6

Multicast Listener Discovery (MLD)

- 1. Adhésion à un groupe
 - Un nœud (appelé listener dans les documentations) souhaitant se joindre à un groupe de multicast donné envoie un message de type Multicast Listener Report à tous les routeurs du lien
 - L'adresse de destination est le groupe dans le cas de MLD version 1 ou l'adresse ff02::16 (tous les routeurs MLDv2) dans le cas de la version 2

ICMPv6

Multicast Listener Discovery (MLD)

- À partir du moment où au moins un listener existe sur le lien, les routeurs vont retransmettre à ce dernier le trafic destiné à ce groupe
- Pour s'assurer de la présence de ces auditeurs, les routeurs vont également vérifier périodiquement celle-ci en diffusant régulièrement des requêtes de vérification
- Mais pour éviter que tous les routeurs d'un lien fassent chacun leurs propres vérifications, un mécanisme d'élection d'un routeur est prévu dans MLD

ICMPv6

Multicast Listener Discovery (MLD)

- 2. Élection du routeur Querier
 - Lorsqu'ils démarrent, tous les routeurs possèdent le statut de Querier
 - Ils vont donc diffuser les requêtes de vérification décrites un peu plus loin
 - Si un routeur reçoit des requêtes émanant d'un routeur disposant d'une adresse link-local de valeur inférieure, il devient Not Querier et cesse de diffuser ces requêtes (au moins tant qu'il reçoit toujours celles de son collègue)
 - Donc un seul routeur reste élu en tant que Querier sur un segment réseau

ICMPv6

Multicast Listener Discovery (MLD)

- 3. Vérification des adhésions à un groupe
 - Chaque routeur élu diffuse un message de type Query pour vérifier la présence de membres appartenant à un groupe de multicast

ICMPv6

Multicast Listener Discovery (MLD)

- 4. Abandon par un nœud d'un groupe de multicast
 - Dans ce cas, le système concerné envoie un message de type MLD Done à tous les routeurs présents sur le lien (adresse de destination ff02::2). Cela n'est valable que pour MLDv1
 - Le routeur désigné comme Querier doit alors s'assurer qu'il reste ou non des listeners pour le groupe en question
 - Pour cela, il enverra une requête de type Query
 - Si le nœud ayant quitté le groupe était le dernier abonné au groupe, le routeur cessera toute diffusion de trafic à destination du groupe sur le lien concerné
 - Les messages de type Done n'étant pas systématiquement envoyés par les nœuds quittant les groupes, un mécanisme de timers intervient pour que le routeur Querier fasse régulièrement cette vérification

ICMPv6

Multicast Router Discovery (MRD)

- Ce mécanisme, décrit dans le RFC 4286, permet à des postes de découvrir quels sont les routeurs susceptibles de faire du routage multicast. Il existe en IPv4 et en IPv6
- Dans le cadre d'IPv6, le mécanisme MRD se base sur les trois types de messages ICMPv6 déjà évoqués plus haut : 151, 152 et 153
- Les routeurs annoncent leur participation à du routage multicast en envoyant un message de type 151 (Multicast Router Advertisement) à l'adresse spéciale ff02::6A (All-snoopers multicast address) désignant tous les auditeurs potentiels

ICMPv6

Multicast Router Discovery (MRD)

- Les postes souhaitant connaître les routeurs intervenant dans le routage multicast émettent à destination de l'adresse ff02::2 (tous les routeurs) un message de type 152 (Multicast Router Solicitation) pour que ceux-ci répondent par un message de type Advertisement
- Enfin, les routeurs cessant d'accomplir cette fonction de routage multicast envoient un message de type 153 (Multicast Router Termination) à l'adresse ff02::6A