Adressage IP
Quatrième partie

Hainaut Patrick 2024

But de cette présentation

- Vous permettre de connaitre et comprendre IPv6
- Ce cours constitue une introduction à IPv6
- Nous avons omis à dessein les aspects avancés qui relèvent d'un niveau bachelier

Pénurie d'adresses IPv4

- **Histoire:**
  - On parle pour la première fois d'un protocole Internet en1977 et il était déjà fait mention d'une version 4 en 1979
  - C'est en 1981 que la structure sur 32 bits et la répartition des adresses en classes A, B et C sont définies
  - 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

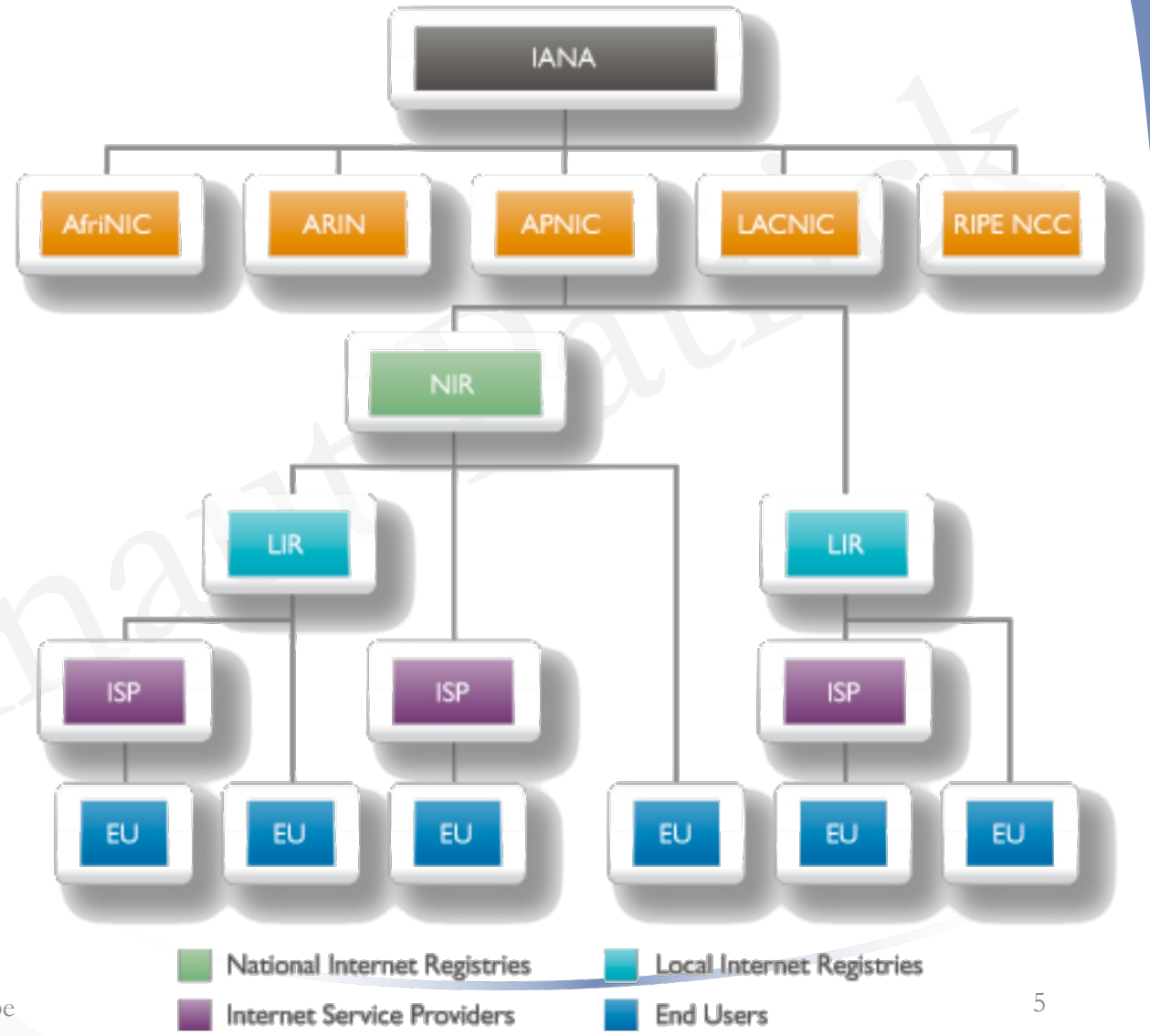
Pénurie d'adresses IPv4

- **Maintenant:**
  - Les 3,7 milliards d'adresses publiques IPv4 ont toutes été distribuées par l'IANA depuis février 2011
  - En fait, la situation varie un peu d'une région à l'autre, car les différents registres n'ont pas obligatoirement distribué toutes les adresses transmises par l'IANA, mais il en reste moins de 2%

Pénurie d'adresses IPv4

- Organisation de la distribution des adresses IP publiques (IPv4 et IPv6)

- Les organisations citées sont des ASBL



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 Internet qui utilisent parfois plusieurs adresses IP publiques (une avec leur abonnement Internet et au moins une avec leur smartphone)
    - La multiplication d'objets connectés (*IoT*) augmente encore cette consommation d'adresses même si tous les objets n'ont pas forcément une adresse publique
    - L'accès Internet est permanent, les adresses ne sont donc plus libérées pour être redistribuées
    - L'utilisation des adresses de réseau et de diffusion qui augmente encore le nombre d'adresses publiques utilisées

Pénurie d'adresses IPv4

- **Mesures appliquées ou applicables pour limiter la consommation d'adresses IPv4 publiques:**
  - 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

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

- **2. Accès direct aux périphériques limité:**
  - *a. Difficultés accrues pour la voix sur IP:*
    - Le fait que les téléphones IP ne disposent pas d'adresses IP publiques, la communication se fait par passerelles et NAT interposé, ce qui induit des délais et une baisse de qualité
  - *b. Difficultés accrues pour la visioconférence:*
    - mêmes difficultés, souvent accrues, que pour la *VOIP*

Les limites d'IPv4

- *c. Limitations sur les accès aux serveurs web internes pour les particuliers:*
  - Si nous voulons accéder depuis Internet aux serveurs, notamment Web, qui sont 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*, soit des techniques de *port forwarding*, puisque nous disposons généralement d'une seule adresse IP publique
  - Les deux techniques ne sont pas faciles à mettre en œuvre

- On pourrait encore citer d'autres cas où le *NAT* complique les échanges

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
  - L'ensemble des entreprises utilisant cet adressage privé a donc recours à ces mêmes adresses
  - Il est donc inévitable que des conflits surviennent à un moment ou à un autre
    - *a. Conflits d'adressages privés lors de la mise en place de VPN*
    - *b. Conflits d'adressages privés lors de la fusion de réseaux*
    - …

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 à 128 bits
  - 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 d'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 $3,4x10^{38}$)
  - Il est habituel d'indiquer que cela fait environ $6,6x10^{23}$ adresses par mètre carré de surface de la Terre (environ $5,1x10^{14}$ m^2)
  - Il y a tellement d'adresses IPv6 disponibles que des billions d'adresses pourraient être attribués à chaque personne sur la planète

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 -> **plus de *NAT***

- 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*
  - 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*

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. -> **Plus de *broadcast* !**

- **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.

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

- **Diffusion multimédia facilitée:**
  - La richesse du *multicast* IPv6 permet de diffuser facilement des programmes audio ou vidéo en IPv6 jusqu'au moindre récepteur (TV, radio, micro-ordinateur).

Syntaxe des adresses IPv6

- Une adresse IPv6 comporte 128 bits, soit 16 octets, notés en hexadécimal (0 -> 9, A -> F)

- Elle est divisée en huit blocs de 16 *bits* séparés par le caractère **:** , comme dans certaines notations d'adresses *MAC*

- Par exemple, nous pourrons rencontrer l'adresse IPv6 suivante :

  2001:0db8:0000:0000:0101:abcd:0000:1234

  1er : 2ème : 3ème : 4ème : 5ème : 6ème :7ème : 8ème bloc

Syntaxe des adresses IPv6

- On peut simplifier l'écriture des adresses IPv6:

- **1. Suppression des zéros de tête**
  - Les zéros figurant en tête de chaque bloc peuvent être supprimés
    Exemple: :0db8: devient :db8: et :0000: devient :0:

    ce qui transformera notre exemple
    2001:0db8:0000:0000:0101:abcd:0000:1234 en :
    2001:db8:0:0:101:abcd:0:1234

Syntaxe des adresses IPv6

- **2. Utilisation d'un double ::**
  - Des blocs remplis de 0 continus peuvent être représentés par **::** comme dans notre exemple
    2001:0db8:0000:0000:0101:abcd:0000:1234
    qui devient alors 2001:db8**::**101:abcd:0:1234

  - 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 2001:0db8:0202:0101:abcd:1234:0000:5678
    qui devient 2001:db8:202:101:abcd:1234::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 : **préfixe 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::ba30:0:0:0:0/60 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 2001:0db8:0000:0000:0000:0000:0000: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	2000::/8 à 3fff::/8

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**
  - 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 lettres peuvent être écrites en minuscules ou majuscules
  - En cas de combinaison des adresses avec des ports, il faut utiliser les crochets : [2001:db8::1:0:0:1:80]

Syntaxe des adresses IPv6

- **Exemples:**

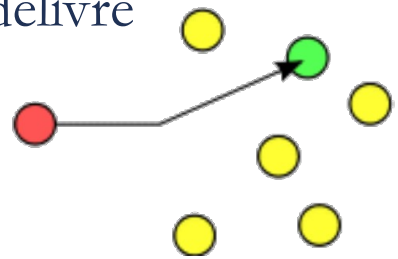
  Adresse IPv6: 2001:db8:25::/48

  Adresse de boucle locale (*loopback adress*): 0:0:0:0:0:0:0:1 -> ::1

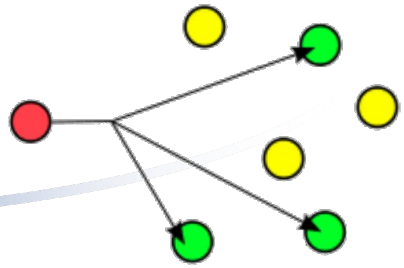
  Adresse indéterminée: 0:0:0:0:0:0:0:0 -> ::

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 qui désigne une interface unique en IPv6
    Tout paquet ayant pour destination cette adresse est délivré uniquement à l'interface détentrice de cette adresse

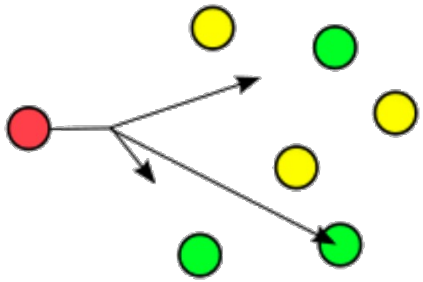


  - ***Multicast*** - comme en IPv4, tout paquet envoyé à une adresse de ce type est reçu et traité par l'ensemble des interfaces appartenant au groupe de diffusion désigné par cette adresse

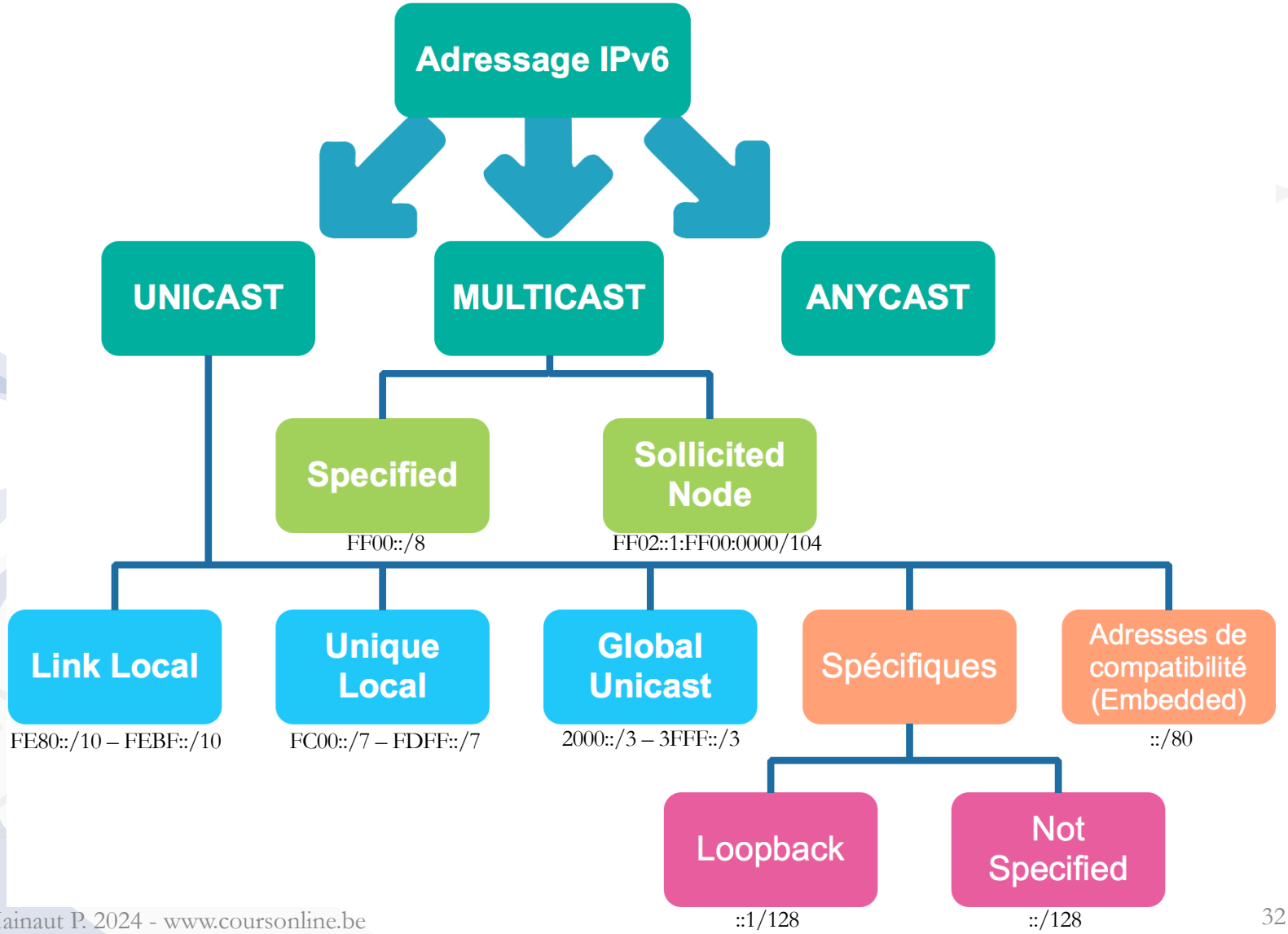


Types d'adresses IPv6 - Généralités

- ***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



Types d'adresses IPv6 - Généralités



Types d'adresses IPv6 – adresses Unicast

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

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

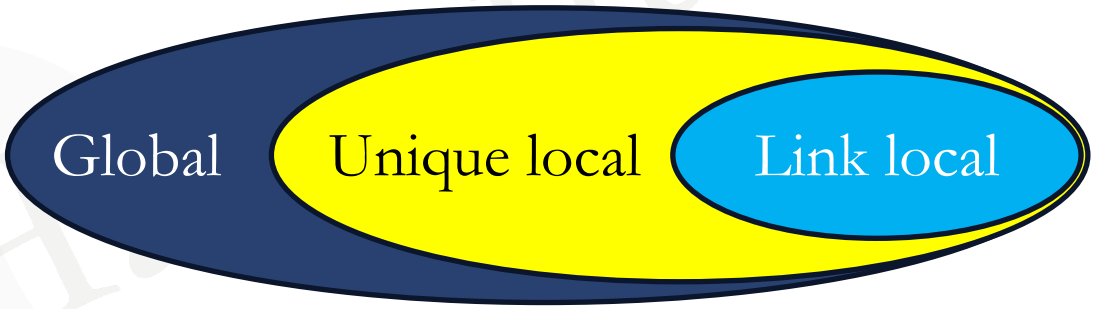
- L'interface ID est la plupart du temps sur 64 bits

Types d'adresses IPv6 – adresses Unicast

- On trouve 3 grands types d'adresses unicast:
  - ***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 en IPv4
  - ***Unique Local Unicast*** - adresses routables dans un réseau privé (sur un ou plusieurs sites) mais pas sur Internet
  - ***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

Types d'adresses IPv6 – adresses Unicast

- Champ d'action des adresses IPv6 *unicast*
  - *Link local*: segment lan / apipa
  - *Unique local*: site / privé
  - *Global*: world wide



Adresses Link-Local Unicast

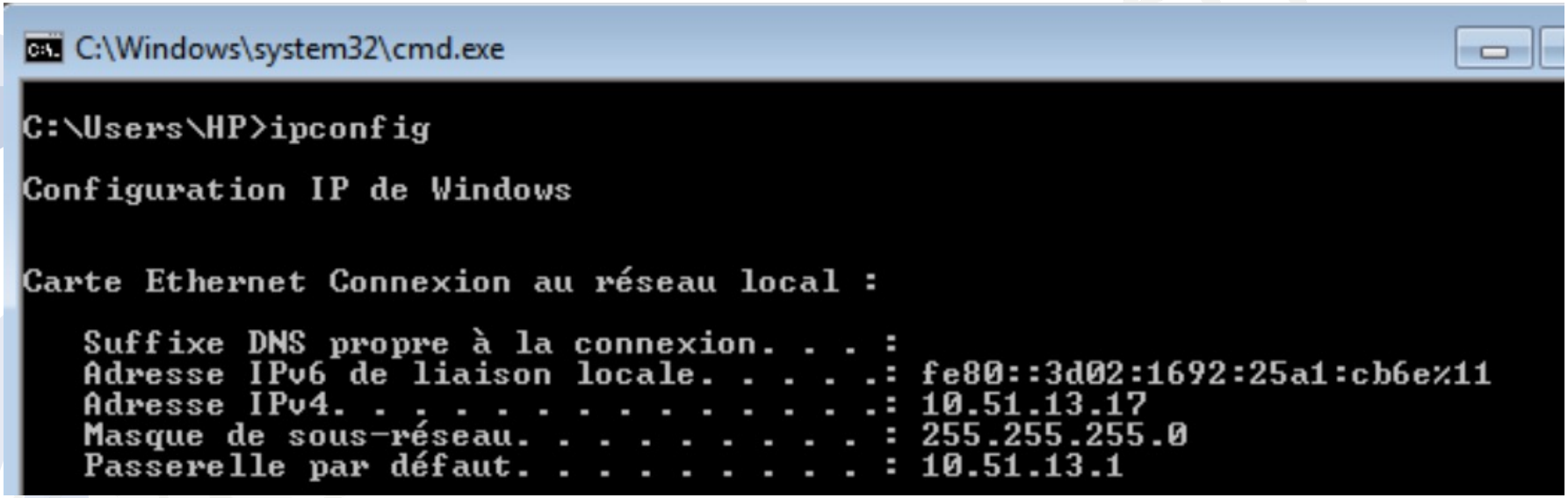
- **a. Adresse de type *Link-Local Unicast***
  - Adresse obligatoire pour communiquer entre deux *devices* IPv6 sur un même lien sans possibilité de routage (idem *APIPA* IPv4)
  - Sa structure est la suivante :

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

  - Ce qui explique pourquoi les adresses de ce type commencent systématiquement par **fe80::/10**
  - Générées et assignées automatiquement sur une interface dès que IPv6 est actif
  - Plage possible: de fe80::/10 à feb::/10

Adresses Link-Local Unicast

- Exemple sur un PC sous Windows en autoconfiguration :



Adresses Link-Local 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 transfèrent pas des paquets ayant comme source ou comme destination une adresse de type *link-local*

Adresses Link-Local Unicast: assignation

- Les 64 derniers bits sont les bits machines et peuvent être assignés:
  - Statiquement:
    - Bits choisis manuellement
  - Dynamiquement, c'est généralement le cas, car ces adresses sont créées automatiquement pour chaque interface:
    - Via EUI-64 pour Cisco
    - Via un *token* pour Microsoft

Adresses Global 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*

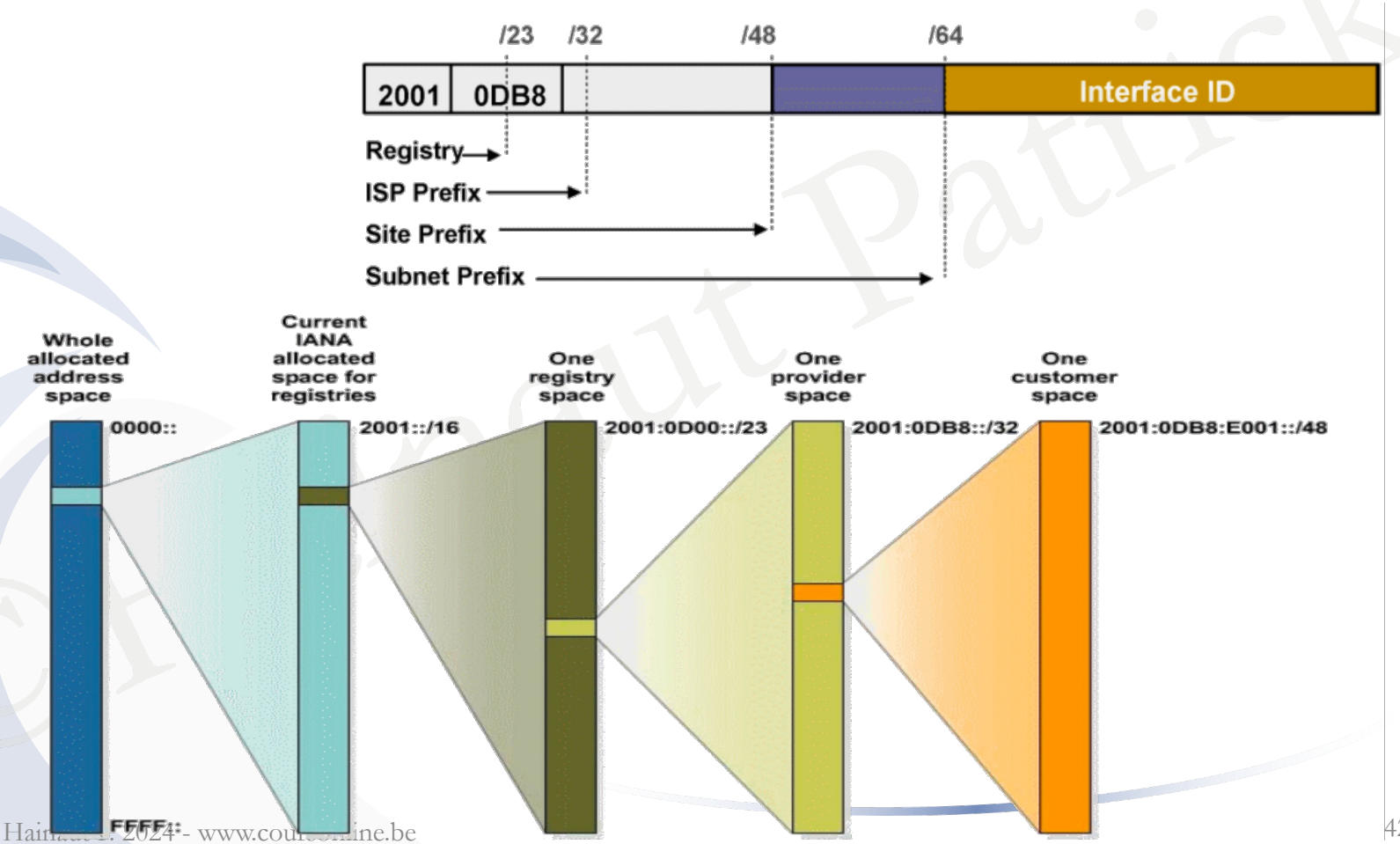
  - Le préfixe de routage global est la valeur permettant de router les paquets depuis Internet vers un site précis
  - Le *subnet ID* permet d'identifier, sur ce site, le sous-réseau
  - L'*interface ID* est la partie hôte (machine) de l'adresse

Adresses Global Unicast

- Pour l'instant, seul le préfixe 2000::/3 est utilisé pour ces affectations
- La plupart des autres préfixes sont seulement réservés
- Plage possible: de 2000::/3 à 3FFF::/3
- 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)

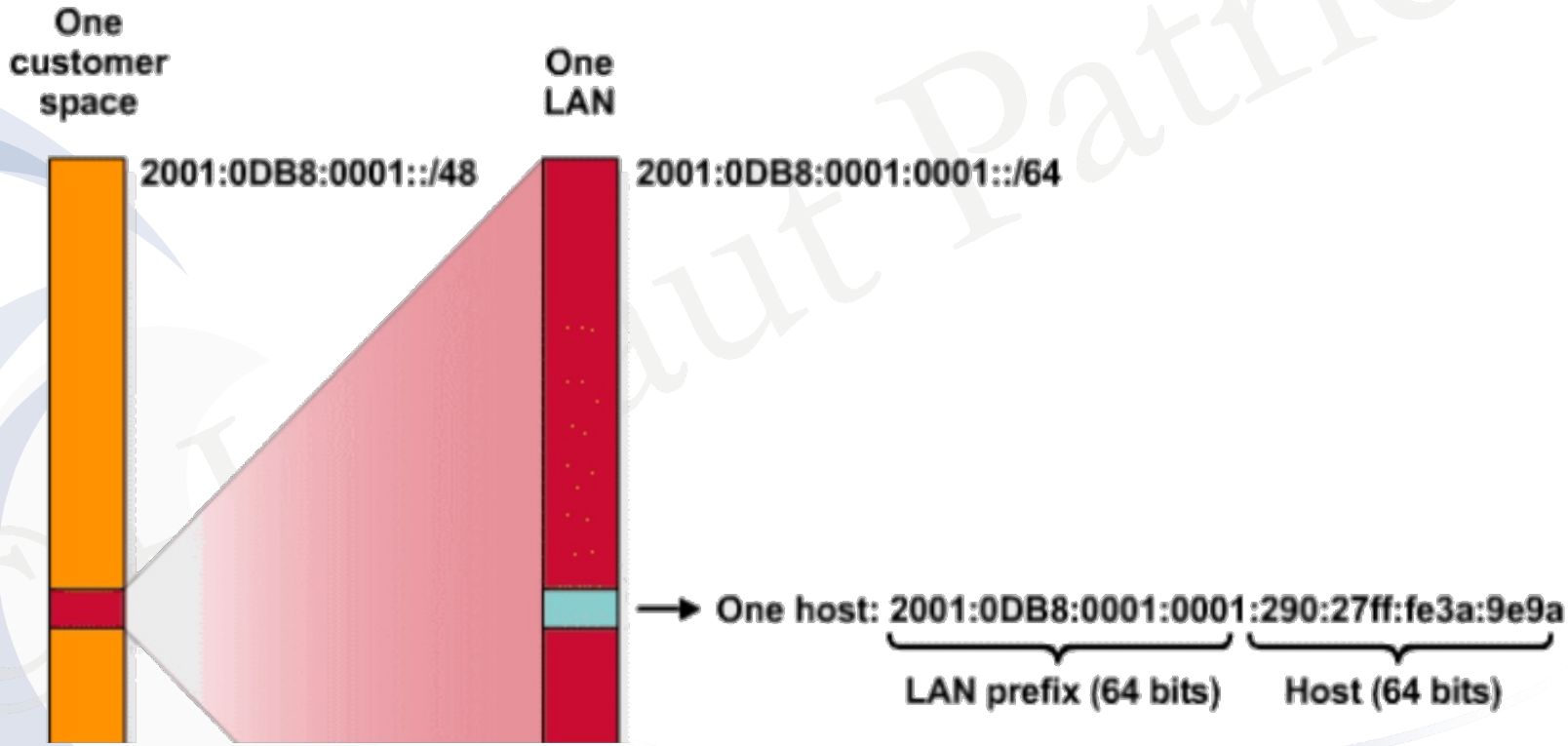
Adresses Global Unicast

- Exemple avec la portion allouée actuelle



Adresses Global Unicast

- Exemple, à partir du préfixe client, du préfixe d'un (sous-)réseau et d'une adresse hôte



Adresses Global Unicast: assignation

- Les 64 derniers bits sont les bits machines et peuvent être assignés:
  - Statiquement:
    - Bits choisis manuellement
    - Via *EUI-64* (voir plus loin)
  - Dynamiquement:
    - Par *DHCPv6*
    - Par I*CMPv6* (*stateless*)

Adresses Unique Local 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, cette catégorie d'adresses locales a été ajoutée
  - Ces adresses ne sont pas routables directement sur Internet mais sont générées par un algorithme pour que des réseaux identiques ne puissent pas exister sur deux entités différentes
  - Ainsi deux entreprises qui fusionnent ou établissent des liens directs entre elles ont peu de risques de devoir renuméroter leurs réseaux (ou mettre en place des translations d'adresses), comme c'est souvent le cas en IPv4

Adresses Unique Local 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::/8

Adresses Unique Local Unicast: assignation

- 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

Types d'adresses IPv6 – adresses Multicast

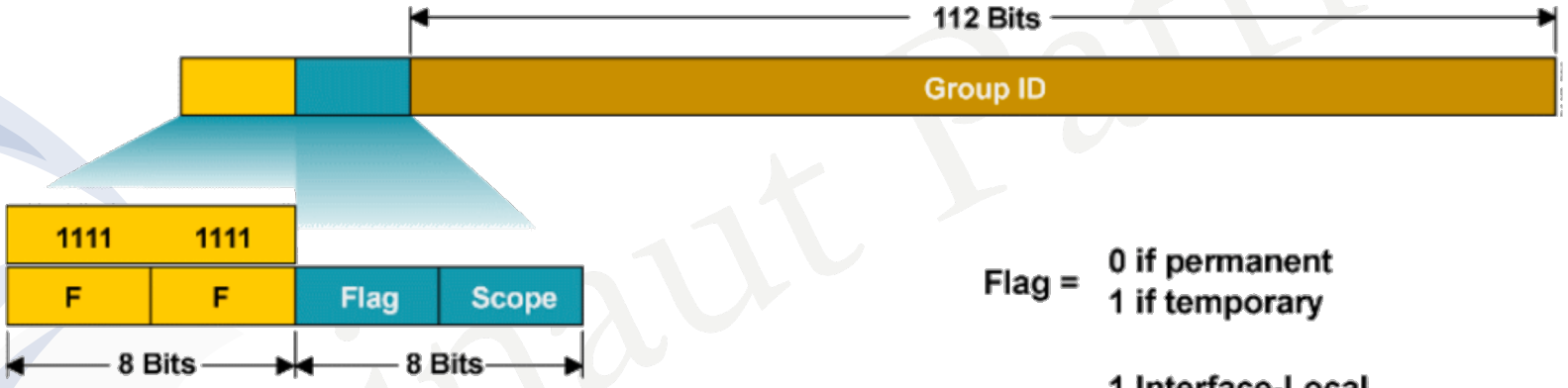
- Comme nous l'avons déjà évoqué, tout paquet envoyé à une adresse de ce type est reçu et traité par l'ensemble des interfaces appartenant au groupe de diffusion désigné par cette adresse

- 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 :



Flag =
0 if permanent
1 if temporary

Scope =
1 Interface-Local
2 Link-Local
3 Subnet-Local
4 Admin-Local
5 Site-Local
8 Organization
E Global

Types d'adresses IPv6 – adresses Multicast

- **a. Syntaxe**
  - Une adresse *multicast* est donc dotée d'un préfixe **ff00::/8**
  - Si le *flag* (4bits) vaut zéro, l'adresse est affectée de façon permanente par l'IANA sinon elle est temporaire
  - Le champ portée (scope), indique l'étendue de 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::123 pour tous les serveurs *NTP* situés sur la même interface que l'expéditeur
  - ff02::123 pour tous les serveurs *NTP* situés sur le même lien réseau que l'expéditeur
  - ff05::123 pour tous les serveurs *NTP* situés sur le même site que l'expéditeur
  - ff0e::123 pour tous les serveurs *NTP* du réseau Internet

Types d'adresses IPv6 – adresses Multicast

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

Types d'adresses IPv6 – adresses Multicast

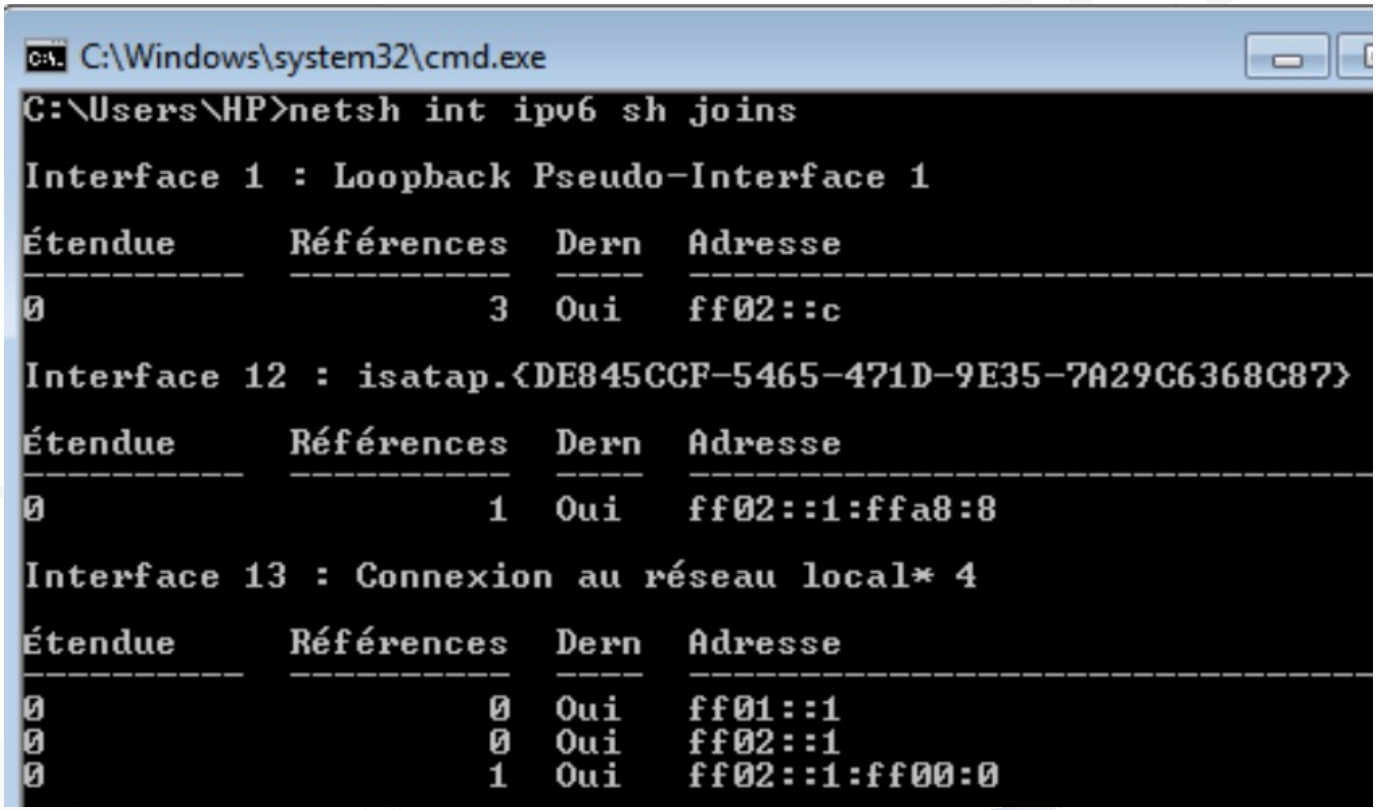
- **d. Cas particulier : adresses *sollicited-node***
  - Ces adresses de type *multicast* sont un peu particulières
  - Elles permettent de cibler un ensemble de nœuds (poste, équipements, serveurs…) dont les 24 derniers bits de la partie *interface ID* sont identiques
  - Le but est de remplacer les *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 :



Types d'adresses IPv6 – adresses Anycast

- Ces adresses ne sont pas différentes des adresses unicast déjà rencontrées

- La syntaxe est pareille et seules les interfaces auxquelles elles ont été assignées sont conscientes de leur nature *anycast*

- Une même adresse *anycast* peut être affectée à des interfaces différentes (dans la plupart des cas sur des matériels différents) et en cas de diffusion d'un paquet vers cette adresse, seule l'interface la plus proche 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)

Types d'adresses IPv6 – adresses Anycast

- Un exemple est d'affecter une adresse IP unique pour les différents routeurs d'une entreprise reliés à Internet ou à des réseaux distants

- 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 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*

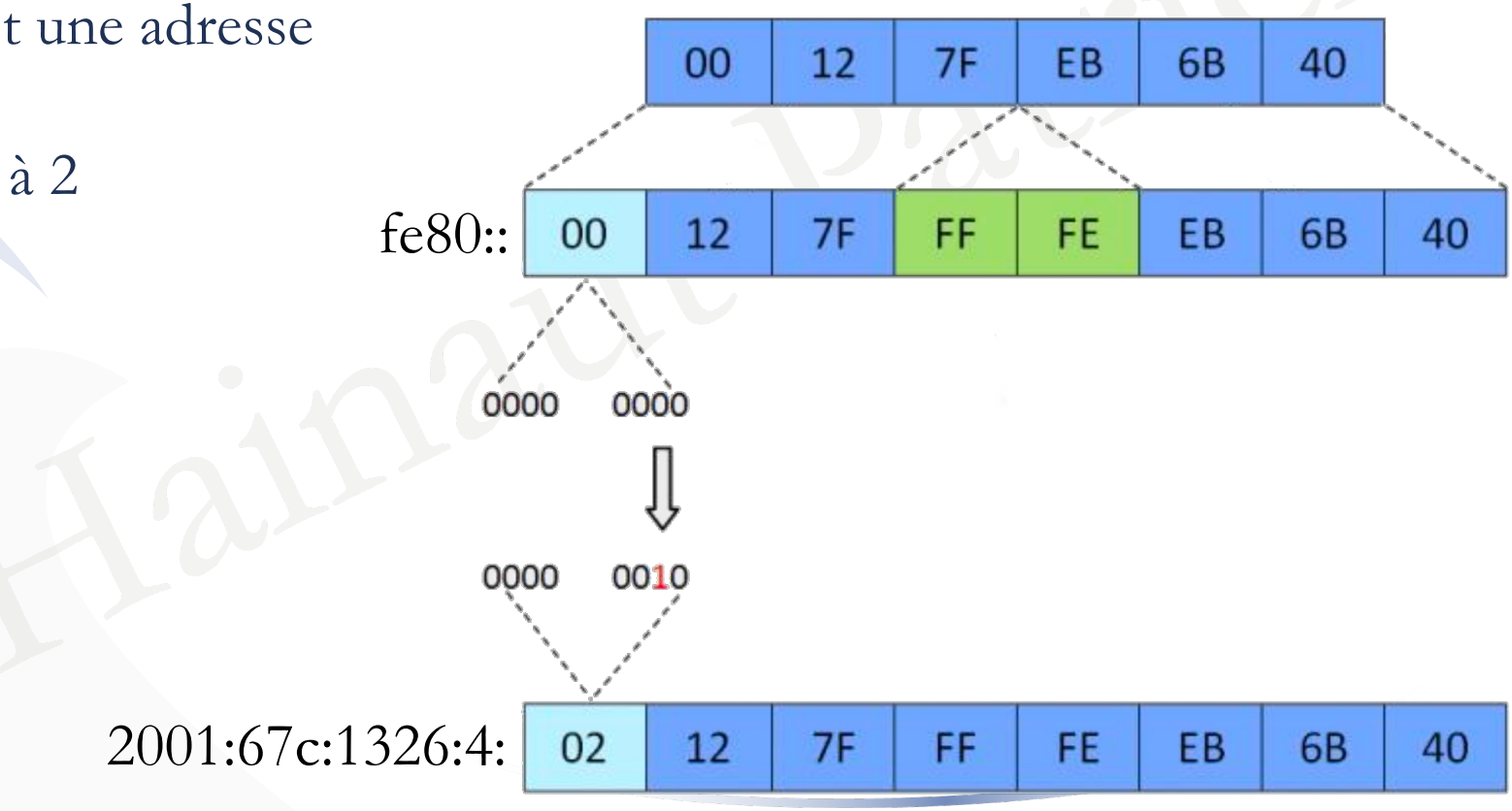
Processus d'allocation d'adresse IPv6: EUI-64

- Cisco utilise le format *EUI-64* pour l'auto-configuration d'une adresse *link-local* ou *global*, pour les 64 bits de la partie *host* (machine)

- Ce format étend les 48 bits de l'adresse *MAC* en ajoutant la valeur 0xFFFE au milieu pour obtenir les 64 bits requis

Processus d'allocation d'adresse IPv6: EUI-64

- Si c'est une adresse *link-local*, le premier byte (de la partie *host*) sera à 0
- Si c'est une adresse *global*, il sera à 2



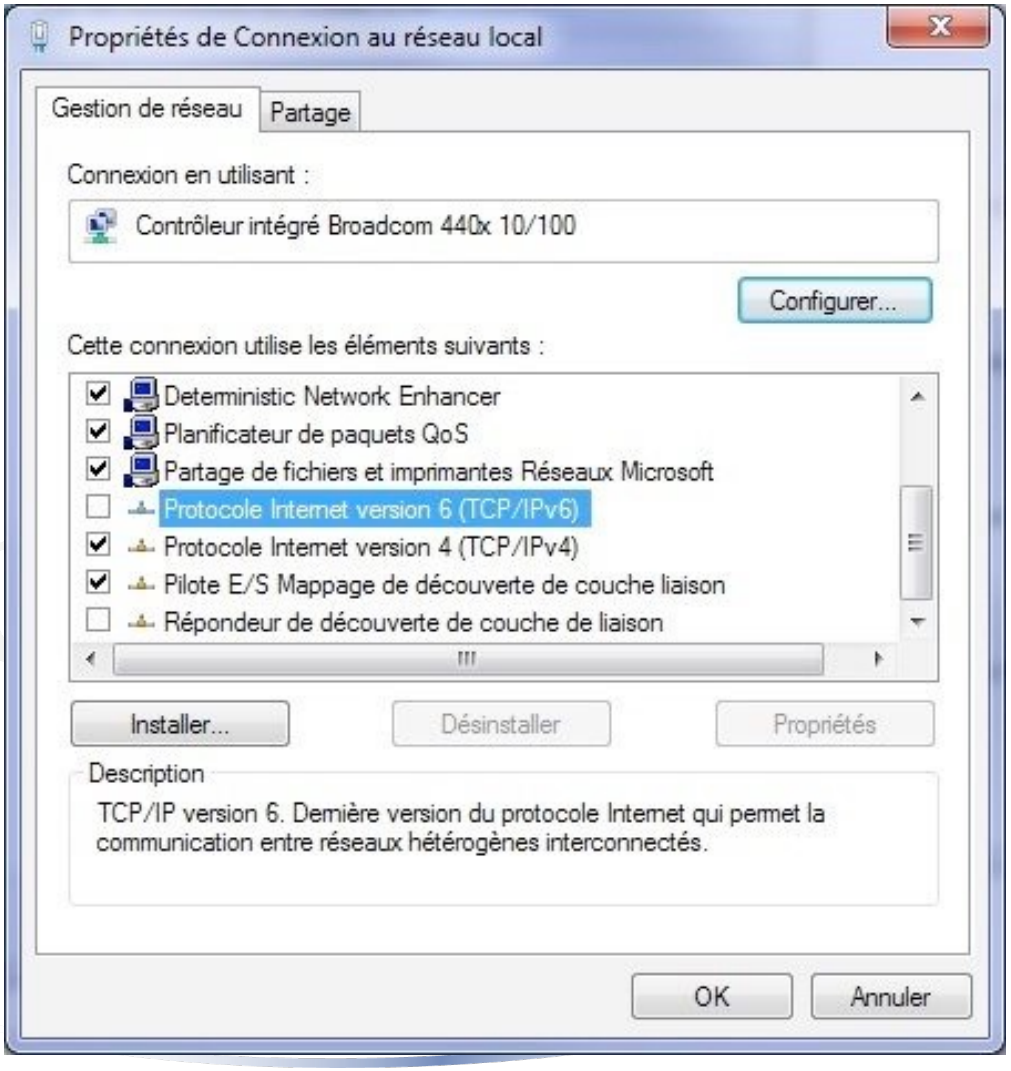
Processus d'allocation d'adresse IPv6: ICMPv6

- Rappelons que *ICMP* signifie *Internet Control Message Protocol*
- La version IPv6 d'*ICMP* permet d'échanger des messages de découverte d'*host* avec *NDP* (*Neighbor Discovery Protocol*)
- Cela permet de déterminer l'adresse *link-local* d'un voisin sur le même segment réseau
- *NDP* permet également de générer une adresse IPv6 *global* à partir des informations reçues d'un routeur configuré au préalable

Mise en œuvre d'IPv6 sous Windows

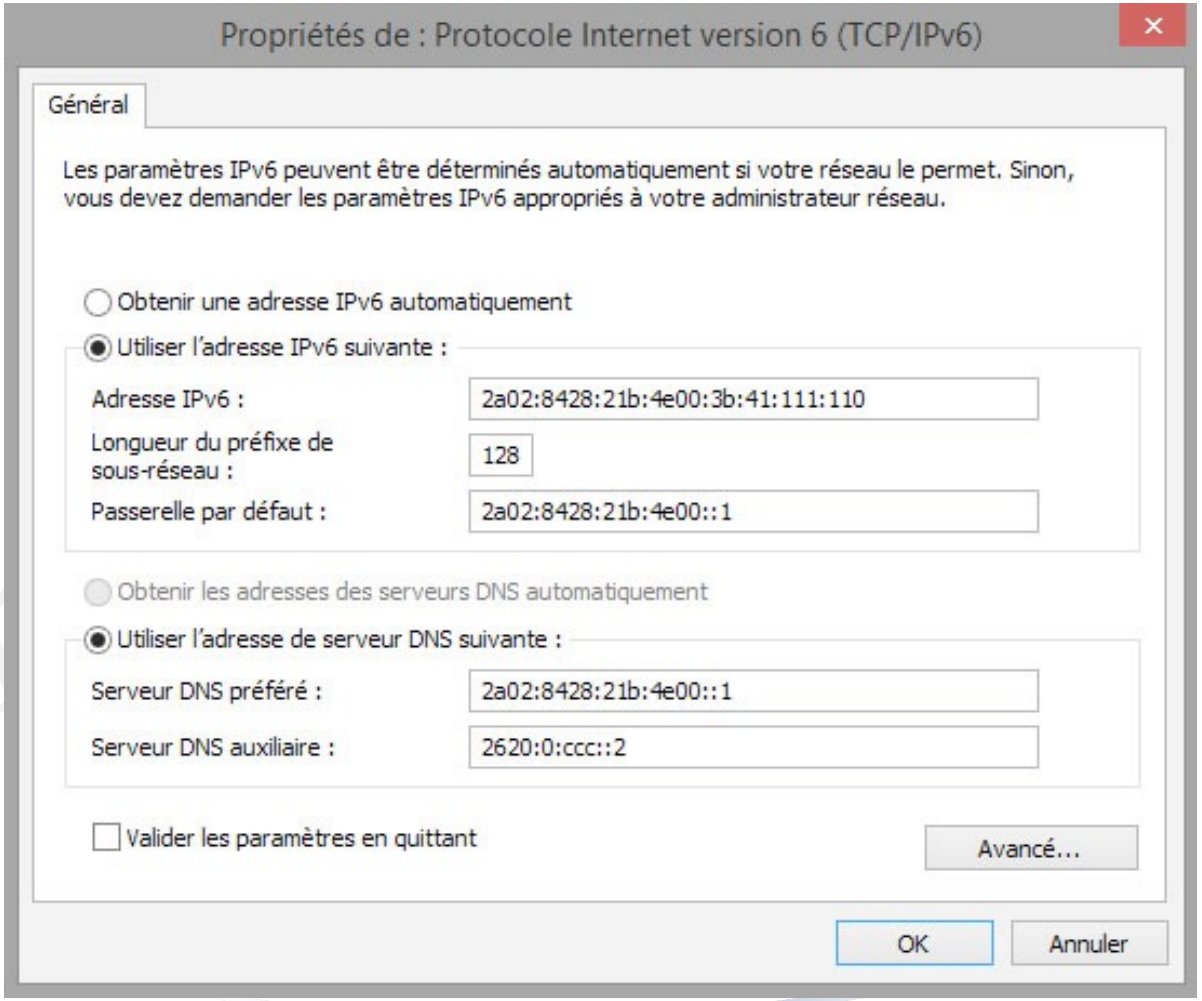
- IPv6 est présent de base sur Windows 7 à 11

- 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 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 PowerShell:

  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
- **show ipv6 int br** pour avoir un résumé de la config IPv6
- **show ipv6 route** permet de voir la table de routage 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

Transition d'IPv4 vers IPv6

- 3 catégories de techniques de transition:
  - ***Dual-stack***: IPv4 et IPv6 coexiste sur les mêmes périphériques/réseaux
  - ***Tunneling***: permet d'encapsuler le protocole IPv6 dans IPv4
  - ***Translation***: NAT-PT solution orientée *NAT*

Transition d'IPv4 vers IPv6: dual-stack

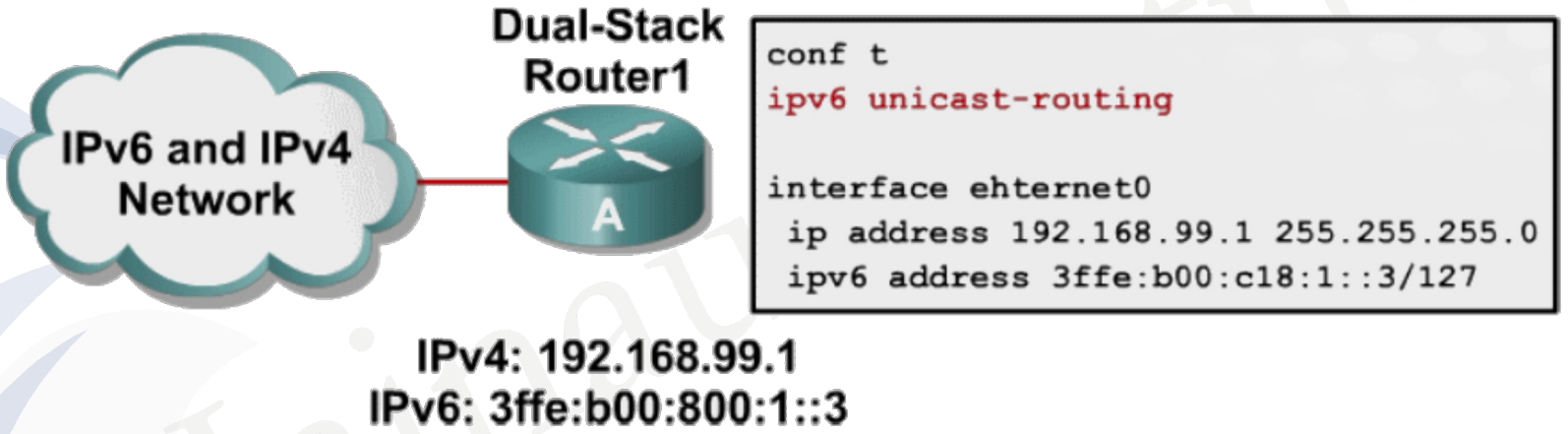
- Quand le réseau est configuré en *dual-stack*, chaque périphérique du réseau reçoit une adresse IPv4 et une (ou plusieurs) adresse(s) IPv6

- Cela permet de convertir progressivement les périphériques d'IPv4 à IPv6

- C'est la méthode la plus employée car la plus simple à mettre en œuvre, elle est de plus transparente pour le *end user*

Transition d'IPv4 vers IPv6: dual-stack

- Exemple sous Cisco




```
conf t
ipv6 unicast-routing

interface ehternet0
 ip address 192.168.99.1 255.255.255.0
 ipv6 address 3ffe:b00:c18:1::3/127
```

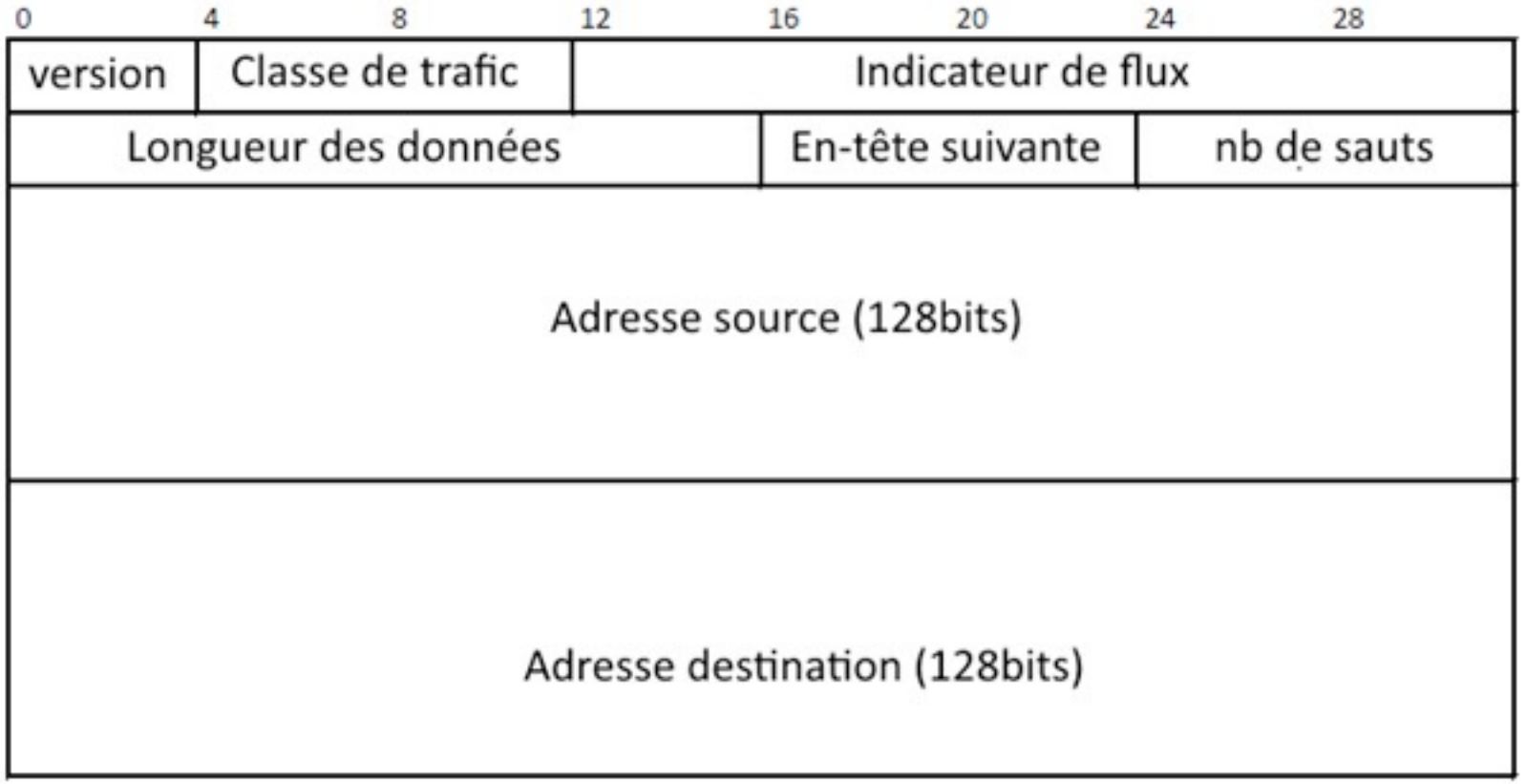
Transition d'IPv4 vers IPv6: tunneling

- Il y a plusieurs méthodes de tunneling qui ne seront pas développées ici

Méthode	Usage
manuelle	Fournit une liaison point à point IPv6 sur un réseau IPv4 existant, prend en charge uniquement le trafic IPv6
GRE	Fournit une liaison point à point IPv6 sur un réseau IPv4 existant, prend en charge plusieurs protocoles, dont IPv6
6to4	Fournit une liaison point à multipoint IPv6 sur un réseau IPv4 existant, les sites doivent utiliser des adresses IPv6 de la plage 2002::/16
6RD	Fournit une liaison point à multipoint IPv6 sur un réseau IPv4 existant, les sites peuvent utiliser des adresses IPv6 de n'importe quelle plage
ISATAP	Fournit une liaison point à multipoint IPv6 sur un réseau IPv4 existant, conçu pour être utilisé entre périphériques d'un même site

Annexe: structure des adresses IPv6

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



Annexe: 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 classifie les données à faire circuler, et donc les priorités qui leur sont affectées
    Son utilisation est la même qu'en IPv4

Annexe: 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

Annexe: 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

Annexe: 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

Annexe: structure des adresses IPv6

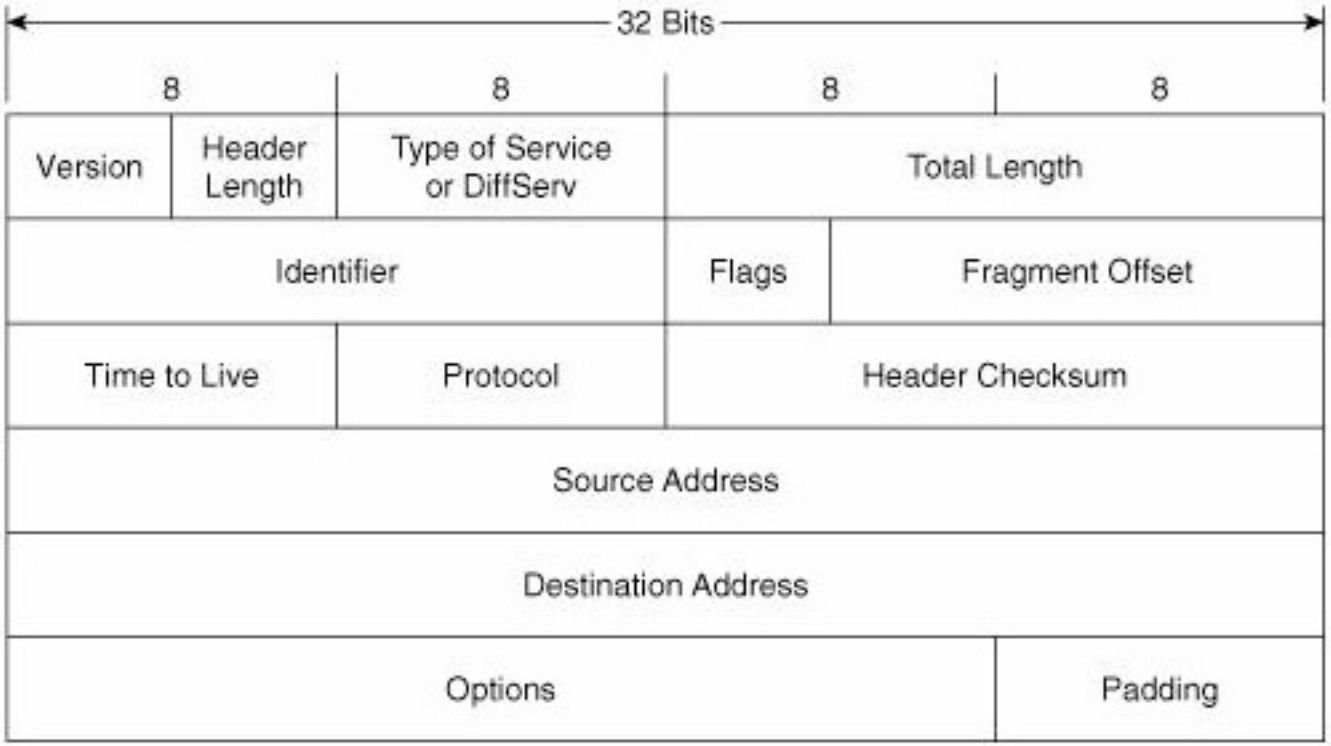
- Après cet en-tête de base, nous allons pouvoir rencontrer deux autres types d'éléments :
  - ***Extension Header*** : au-delà des 40 octets de l'en-tête de base peuvent être ajoutés des en-têtes spécifiques à des protocoles ou des options
    - Chacun se présente avec une longueur multiple de 8 octets (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)

Annexe: 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*

Annexe: structure des adresses IPv4

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



Annexe: 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*

Annexe: 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 address* (32 bits chacune)** sont les mêmes mais leurs longueurs diffèrent (32 bits pour les adresses en IPv4, 128 pour celles en IPv6)

Annexe: 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
```

Annexe: 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:
IPv6mcast_ff:3a:40:66 (33:33:ff:3a:40:66)
    Destination: IPv6mcast_ff:3a:40:66 (33:33:ff:3a:40:66)
        Address: IPv6mcast_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)
        .../...
```

Annexe: 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

```
    Next header: ICMPv6 (0x3a)
```

et nous avons encore la valeur maximale pour la limite de sauts puisque nous n'avons pas traversé de routeur

```
    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)
```

Annexe: 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
```

Annexe: 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)
```

Annexe: 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                           O%.?...K
        Data: 4F258D3F000CEE4B
```

Annexe: 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                          O%.?...K
        Data: 4F258D3F000CEE4B
```

Annexe: 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

Annexe: autoconfiguration en mode stateless: mise en œuvre

- 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)

Annexe: autoconfiguration en mode stateless: mise en œuvre

- **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

Annexe: autoconfiguration en mode stateless: mise en œuvre

- **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
```

Annexe: autoconfiguration en mode stateless: mise en œuvre

- 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#
```

Annexe: 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

Annexe: 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*

Conclusion

- Voilà un tour assez complet de la théorie IPv6 avec quelques exemples de mise en œuvre

- Sources:
  - IPv6, principes et mise en œuvre – Jean-Paul Archier, éditions ENI
  - Formation à Technobel Ciney