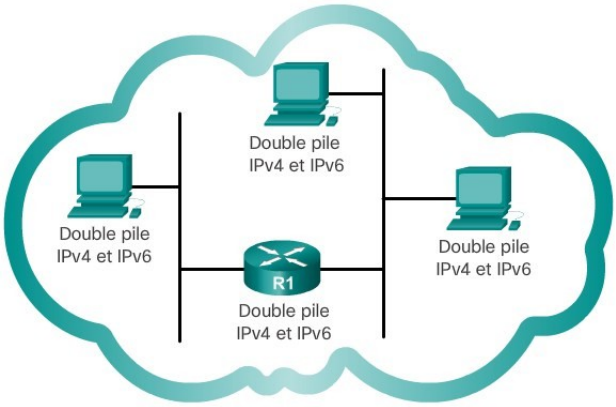
IPV6

Objectif

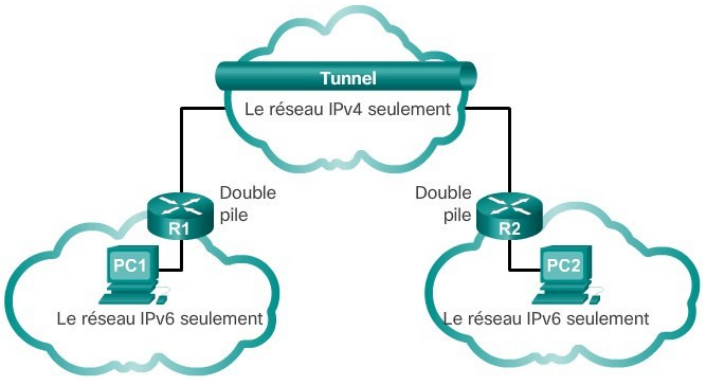
Les bases d'IPV6
transition Ipv6 Ipv4
routage Ipv6

La transition Ipv6 n'ayant pas eut lieu a ce jour, l'ipv4 et ipv6 doivent cohabiter.
L'IETF a classé les techniques de migration en 3 catégorie,

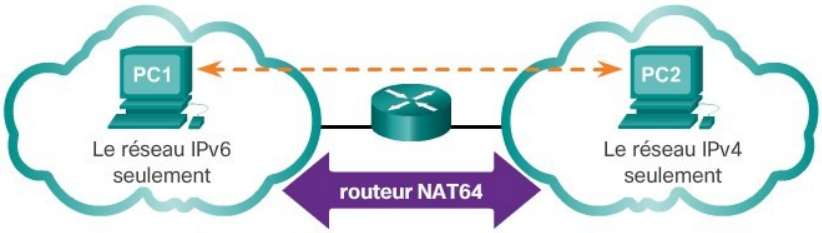
Introduction :



Double-pile : elle permet à l'ipv4 et a l'ipv6 de coexister sur le même réseau. Les équipements double pile exécutent les piles de protocole ipv4 et ipv6 simultanément.



Tunneling : Méthode de transport des paquets ipv6 via un réseau ipv4. Les paquets ipv6 sont encaspulés dans des paquets ipv4.



Traduction : Les équipements ipv6 peuvent utiliser la traduction d'adresses réseaux 64- NAT 64 pour communiquer avec les périphériques ipv4 à l'aide d'une technique similaire au NAT ipv4, un paquet ipv6 sera traduis en paquets ipv4 et inversement.

IPV4 rappel

Adresse sur 32 bits

Les classes :

A utilisé pour les grands réseaux
B moyen réseaux
C petit réseaux

A – 0xxxxxx – 255,0,0,0 – plage de valeur réseau : 1,x,x,x a 126,x,x,x
B – 10 – 255,255,0,0 – 128,0,x,x a 191,255,x,x
c – 110 – 255,255,255,0 – 192,0,0 a 223,255,255,0

Les adresses réservées : 111xxxxx,xxxx,xxxx,xxxx, elles st découpées en 2 groupes

Classe D : 1110 – 224 @ multidiffusion, envoi a un groupement d'équipements utilisant le même protocole Ospf par exemple

Classe E : 1111 – 240 @ expérimentale

Adresses de loopback de type 127,x,x,x elle se situe dans la couche 1 physique du modèle OSI,
mais n'est reliée a aucune interface physique, permet de tester le fonctionnement de votre carte réseau.

0,x,x,x : adresse d'acheminement par défaut permet de limiter les informations d'acheminement, elle correspond a votre adresse de passerelle sur votre Pc, ou 0,0,0,0 utilisée dans les tables de routage pour redirigé les paquets dont le réseau n'ai pas mentionnée dans celle-ci, on l'appel la route par défaut

Adresse réseau : tous les bits d'hôtes st positionnés a 0, de signée pour adresser tous les postes du réseau, l'adresse réseau est utilisée dans les tables de routage

@ de diffusion – 255,255,255,255 envoyé a tous le monde

@ de diffusion réseau – 10,255,255,255 , destiné a tous les hôtes du réseau 10,0,0,0

Espace d'adressage ipv4 - ipv6

Ipv4 : contient 4,3 milliards d'IP utilisables

Ipv6 : 2^128 adresses – 667 millions de milliards d'adresses

La disparition du NAT :

Avec le nombres d'adresses Ipv6 disponible le NAT n'a plus lieu d'être

En Ipv6 chaque machine est directement visible sur internet

Le format IP

format IPV4 - 192.168.1.13

format Ipv6 - fe80::135:2a01:a45:cbbe

L'écriture Ipv6

Elle est écrite en hexadécimale et plus en décimal

elle contient 128 bits – 8 groupes de 2 octets

Le masque de sous-réseau est un **/64** maximum et se nomme le **préfixe de sous-réseau**

Le format d'adresse Ipv6

2 formats :

preferred format - adresse complète

2b05:012:fe3e:0:402c:c185:e9ac:128f

compressed format - format compressé

le - **:0:** - peut être résumé en - **::** -

2b05:012:fe3e::402c:c185:e9ac:128f

ou avec plusieurs **groupe consécutifs de 0 peut être résumé en** **::**

2b05:012:fe3e:0:0:0:e9ac:128f
2b05:012:fe3e::e9ac:128f

Les sous-réseaux sous ipv6

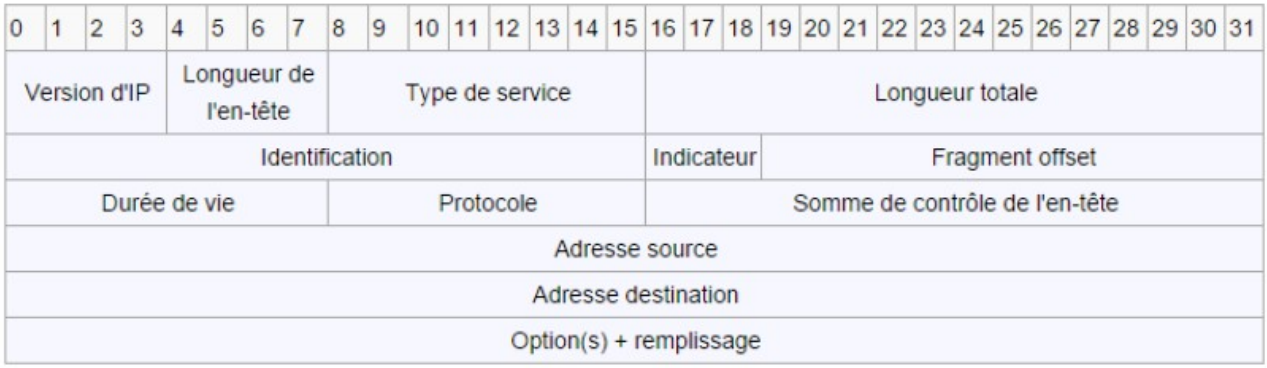
Les **masques de sous-réseaux** sont écrits uniquement en **notation CIDR**

2b05:012:fe3e::402c:c185:e9ac:128f /64 préfixe Ipv6

Toutes les adresses sont utilisables donc **plus de notions de classes d'adresses** et de **broadcast ip**

Comparaison des datagrammes

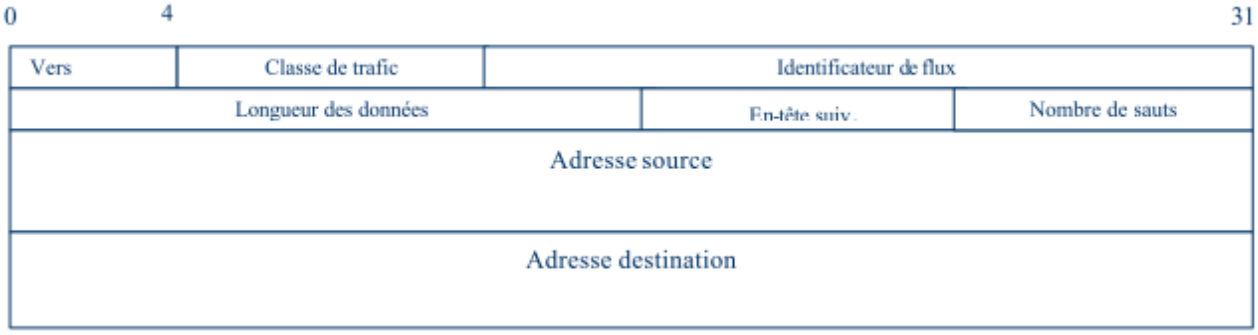
datagramme ipv4



datagramme Ipv6

on trouve par rapport a l'ipv4 une simplification de l'entête , les champs sont moins nombreux , 7 au lieu de 14 champs , cela permet une **meilleur commutation/routage des équipements de routage** et qui par conséquents ont moins de données a empiler pour router.

Entête IPV6



version :

version 6 pour ipv6

TOS - traffic of class - classe de traffic – – :

type de service - **Voip – data - mgmt** ... **nature du trafic** , permettant d'**offrir un niveau de priorité aux paquets** suivant leur type et suivant le type d'équipement traversé

QOS - Quality of service - flow label (20 bits) - Identificateur de flux – - :

gestion de la QOS permet d'**optimiser le routage** , on peut donner un **identifiant a la communication** selon sa valeur , les routeurs du chemin reconnaissent la connexion et ne dépilent pas les informations et transmettent directement.

,

Longueur des données :

longueur des paquets sans l'entête exprimé en octet

en-tête suivant – next Header :

protocol / extension / options

hop limits - nombres de sauts – :

TTL : nb de routeurs traversés, valeur sur **8 bits** , elle est **décrémentée** a chaque traversée d'un routeur , quand la valeur atteint **0** , le paquet est détruit et **un message d'erreur est émis en ICMPv6**

Le NEXT HEADER -en-tete suivant

il est codé sur **8 bits** , il permet de connaître **le type de data ou options qui se trouve derrière l'entête du paquet ipv6** , il correspond au champ protocole de l'ipv4 RFC 1700

liste des protocoles

- 01 - 00000001 - ICMP
- 02 - 00000010 - IGMP
- 06 - 00000110 - TCP
- 17 - 00010001 - UDP
- 58 - 00111010 - ICMPV6

Ipv6 les options des paquets sont gérés grâce a des extensions

Les extensions sont posées après les **header** et avant les **data**

exemples :

routage
fragmentation //découpage des paquets en fonction de la quantité des données a envoyées et du PMTUD
ipsec
mobilité : losqu'un client change d'emplacement il garde la même passerelle , utilisation du binding update – triangular routing
etc...

la fragmentation ipv6

ipv4 , les routeurs fragmente les paquets en fonction de la MTU

Ipv6 , si le paquet est trop volumineux , il envoi un paquets icmpv6 avec un message **"packet too big"**

-L'émetteur est devenu le responsable de la fragmentation

Tialle maximun d'une MTU IPV6 : **4352 octets**

MTU minimal pour les liens : **1280 octets IPV6** – contre **576 octets en ipv4**

le PMTUD – path MTU discovery:

repose sur des messages ICMPv6 pour déterminer le MTU minimum entre 2 équipements

Les différentes adresses utilisé sur la carte réseau

Premier hextet (à l'extrême gauche)	Type d'adresse IPv6
0000 à 00FF	Adresse de bouclage, n'importe quelle adresse, adresse non spécifiée ou adresse compatible IPv4
2000 à 3FFF	Adresse de monodiffusion globale (adresse routable dans une plage d'adresses actuellement distribuée par l'IANA [Internet Assigned Numbers Authority])
FE80 à FEBF	Liaison locale (adresse de monodiffusion qui identifie l'ordinateur hôte du réseau local)
FC00 à FCFF	Adresse locale unique (adresse de monodiffusion qui peut être attribuée à un hôte pour l'identifier comme faisant partie d'un sous-réseau spécifique du réseau local)
FF00 à FFFF	Adresse de multidiffusion

Les type d'adresses ipv6 :

IPV6 supporte 3 types d'adresses : Unicast, Multicast et Anycast

Les adresses unicast :

Elles **désignent une et une seule machine**
Elles comportent une partie réseau « préfixe » et une partie hôte « suffixe »
La partie réseau ou préfixe est codées sur 64 bits : les 48 bis public « global routing prefix » et les 16 bits de site définissant le sous-réseau

La partie hôte ou siffixe est codée sur 64 bits , fabriquée a partir de l'adresse MAC de l'interface, elle permet d'identifier la machine dans un réseau donné.

Example :

L'adresse **fe80::20d:61ff:fe22:3476**

fe80:: ,en réalité **fe80:0000:0000:0000** correspond au préfixe ou partie réseau

20d:61ff:fe22:3476 correspond au **suffixe ou partie hôte**

Les adresses multicast :

Le protocole IPv6 généralise l'utilisation des adresses multicast qui remplacent les adresses de type "broadcast" (diffusion) qui n'existent plus en IPv6. La raison de cette disparition est que l'émission d'un paquet broadcast était très pénalisante pour toutes les machines se trouvant sur un même lien.

Une adresse **multicast** est une adresse **désignant un groupe d'interfaces** donné. Une interface est libre de s'abonner à un groupe ou de le quitter à tout moment, c'est donc moins pénalisant qu'en IPv4.

Le format des adresses multicast est le suivant :

ff01 : noeud local, les paquets ne quittent pas l'interface.

ff02 : lien local, les paquets ne quittent pas le lien .

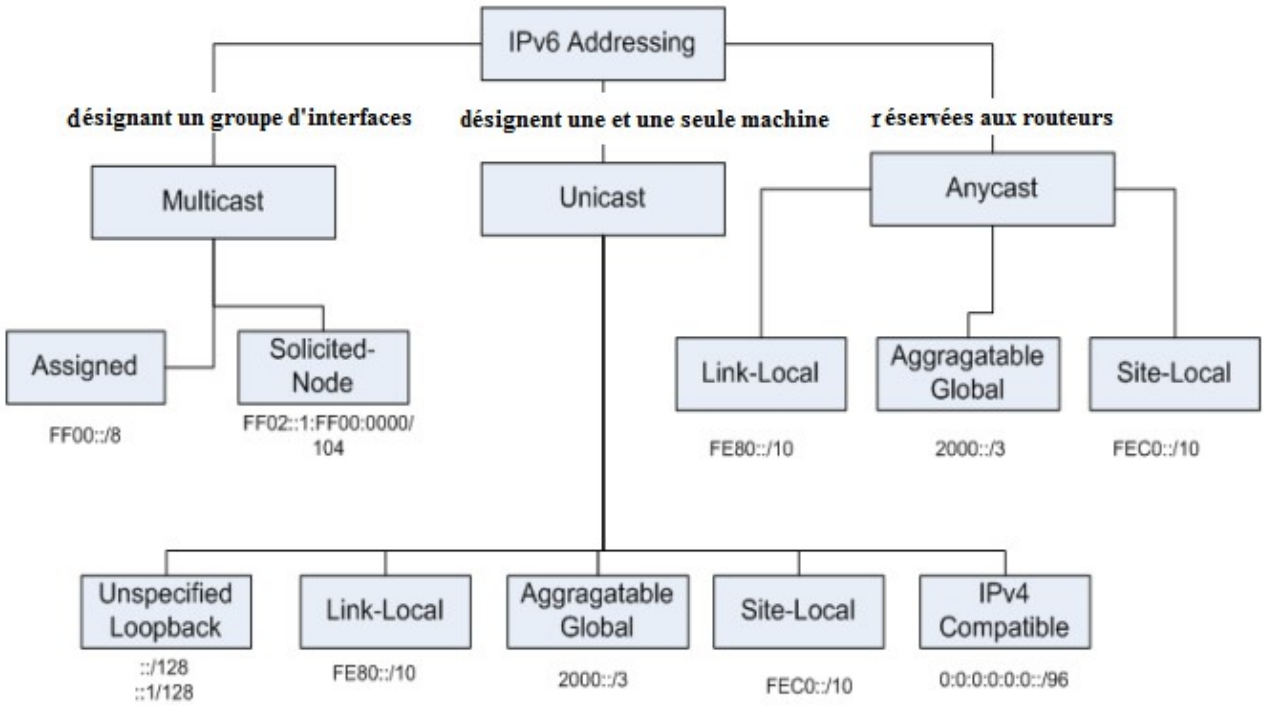
ff05 : site local, les paquets ne quittent pas le site .

Les adresses anycast

Anycast est un nouveau type d'adressage. Il **identifie qu'un noeud**, parmi un groupe de noeuds, doit recevoir l'information.

Une adresse anycast, comme une adresse multicast, désigne un **groupe d'interfaces**, à la différence qu'un paquet émis avec comme destinataire **une adresse anycast ne sera remis qu'à un seul membre du groupe**, par exemple **le plus proche au sens de la métrique des protocoles de routage**, même si plusieurs interfaces ont répondu au message. L'interface de destination doit spécifiquement être configurée pour savoir qu'elle est anycast.

Pour l'instant, **une seule adresse anycast** est utilisée, elle est **réservée aux routeurs** mais dans l'avenir, d'autres pourraient être définies.



1. Adresse lien local - link-local FE80::/10 – AUTOCONF STATELESS

Une adresse de lien local est une adresse automatique unique (cf chapitre AUTOCONF) affectée a un pc câblé ou non , on peut la faire correspondre aux adresse APIPA -169.X.X.X que Windows affecté automatiquement lorsque votre carte réseau est configurée en client dhcp et qu'aucun serveur dhcp n'ai présent sur le réseau cela permettra a 2 pc switchés de pouvoir communiquer entre eux et d'accéder aux partage windows par exemple .

construction : FE80+**EUI-64**

sous linux cette adresse sera constituée d'un premier hextet de type **FE80** et d'un **EUI qui est constitué de l'adresse MAC de la carte réseau**

sous windows , idem pour le premier hextet par contre la partie **EUI-64 sera générée de façon aléatoire**

cette **adresse non routable** donc **ne peut traverser un routeur** sera utilisé que sur la LAN de la société , elle est utilisé que pour communiquer entre les équipements sur un même sous-réseau

elle est **utilisée dans certains mécanisme de ipv6 par le protocole NDP** en envoyant des paquets de découverte et d'avertissement (**discovery / advertissement du NDP**), **pour surveiller l'unicité de cette adresse**

Grâce au protocole NDP et la fonction router discovery , le PC pourra récupérer l'adresse de **link-local du routeur** qui sera utilisée comme **default gateway** pour celui-ci

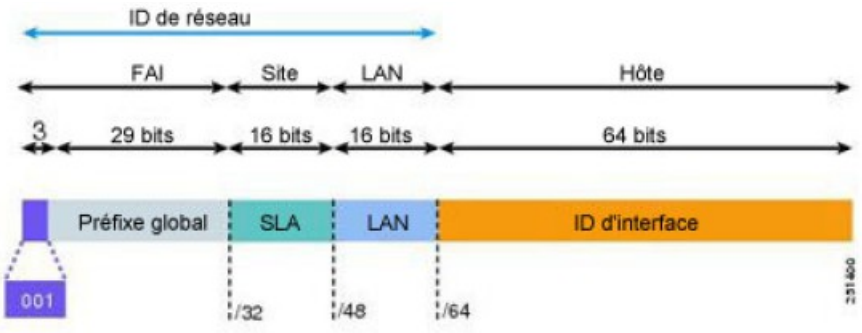
2. adresses de Site-local – RFC 3879

préfixe : FEC0::/10
les adresses sont restreintes au site local de l'entreprise , correspond aux adresses privées ipv4 qui utilise la notion de classes

10 bits 54 bits 64 bits
FECO: + identifiant-sous-réseau +identifiant-interface
1111111011

-adresse de site local ne sont plus utilisées et ont étaient remplacée par des adresse locales uniques

3. address global unicats



préfixe : 2000::/3
adresse publique de l'équipement/pc vu de l'internet
adresse divisée en 3 partie

001 – 3 bits = 200::/12 : défini que c une adresse routable sur internet
préfix global – 29 bits

Iana défini **les types d'adresse**, cette adresse de type global unicats : **200::/12**
et les **bits /12 a /23 définissent quel organisme gére les mapping par pays** ,cad les organismes **d'enregistrements locaux RIR**, après adresse du **fai** , **adresse du site client** , **adresse sous-réseau**, **identité machine** .

adresses de Multicast – broadcast

plus de broadcast , remplacé par les adresses de multicast
utilisé sur les sous-réseaux locaux

commence par FF

bits	8	4	4	112
champs	111111	**flag**	**scope**	ident-groupe

préfixe FF00::/8

les bits de **FLAG : O R P T**

les bits **SCOPE** – étendue

0	réservé
1	noeud
2	lien
4	administration
5	site
8	organisation
E	global
F	réservé

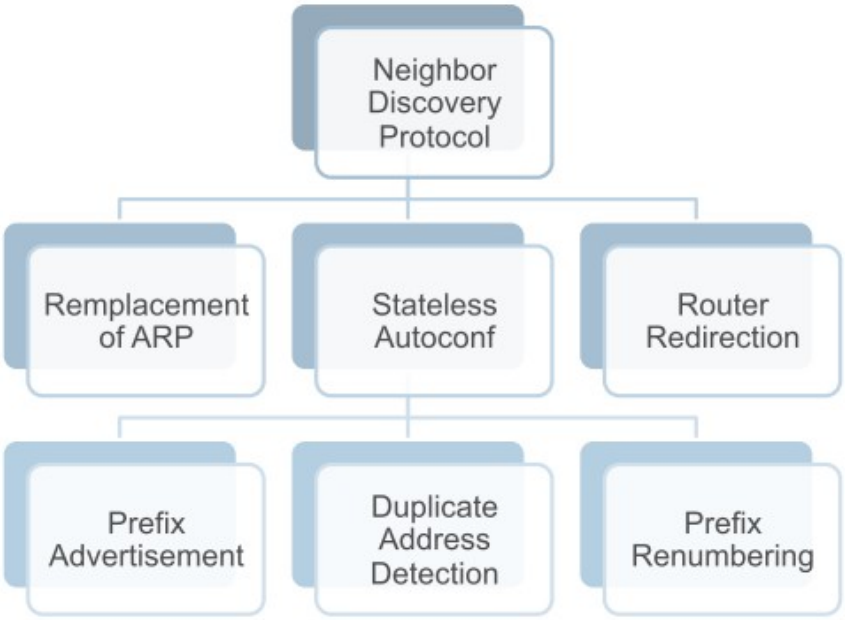
exemple :

FF02::1	destiné a tous les équipements des sous réseaux
FF02::2	destiné a tous les routeurs

mapping ethernet

33:33xx:xx:xx:xx ou xx étant les 32 derniers bits de l'adresse ipv6

le protocole NDP – neighbor discovery protocol



Neighbor Discovery Protocol (NDP) est un protocole utilisé par IPV6 Il **opère en couche 3** et est responsable de la **découverte des autres hôtes** sur le même lien physique, de la **détermination de leur adresse et de l'identification des routeurs présents.**

NDP fournit à IPv6 des services similaires à **Address Resolution Protocol (ARP)**, ICMP Router Discovery et Router Redirect pour IPv4. Il fournit cependant certaines améliorations comme le ***Neighbor Unreachability Detection*** **(NUD)** qui permet de **détecter des systèmes inaccessibles**. D'autre part, NDP est moins dépendant d'un type de média qu'ARP. utilise le protocole ICMP V6

les différents types de messages NDP – icmpV6

type 135 : **neighbor solicitation NS** **//demande d'adressage ipv6 auprès d'un router**
envoi a l'adresse de multicast **FF02::2**

type 136 : **neighbor advertissement** **//reponse du router**
envoi a l'adresse de multicast **FF02::1**
dans les 2 cas l'**adresse source** d'envoi et l'adresse de lien local du pc

envoi a l'adresse de multicast FF02::1:FFXX:XXXX ou XX sont les derniers 32 bits de l'adresse Ipv6 cible

AUTOCONF

L'auto configuration : L'auto-configuration met en œuvre en un certain nombre de nouveaux protocoles associés à IPv6 : protocole de découverte des voisins, nouvelle version d'ICMPV6, etc. **L'auto-configuration permet à un équipement de devenir complètement « plug-and-play ».** Il suffit de connecter physiquement la machine pour qu'elle acquière automatiquement une adresse IPv6 et une route par défaut. Ceci facilite la renumérotation (ré-adressage des équipements et des machines).

trois types d'autoconf existe en Ipv6

1. **autoconfiguration stateless – automatique sans état**
2. **autoconfiguration stateful – automatique plein état – DHCP v6**
3. **autoconfiguration stateless- automatique sans état – DHCP v6**

1. autoconf stateless

les équipements se configurent tout seuls et obtiennes leur adresse de lien local
le routeur annonce un préfixe sur le réseau et les clients se configure en utilisant
le préfixe et EUI-64

il existe plusieurs normes quand a cette auto-configuration de lien local pour les équipements

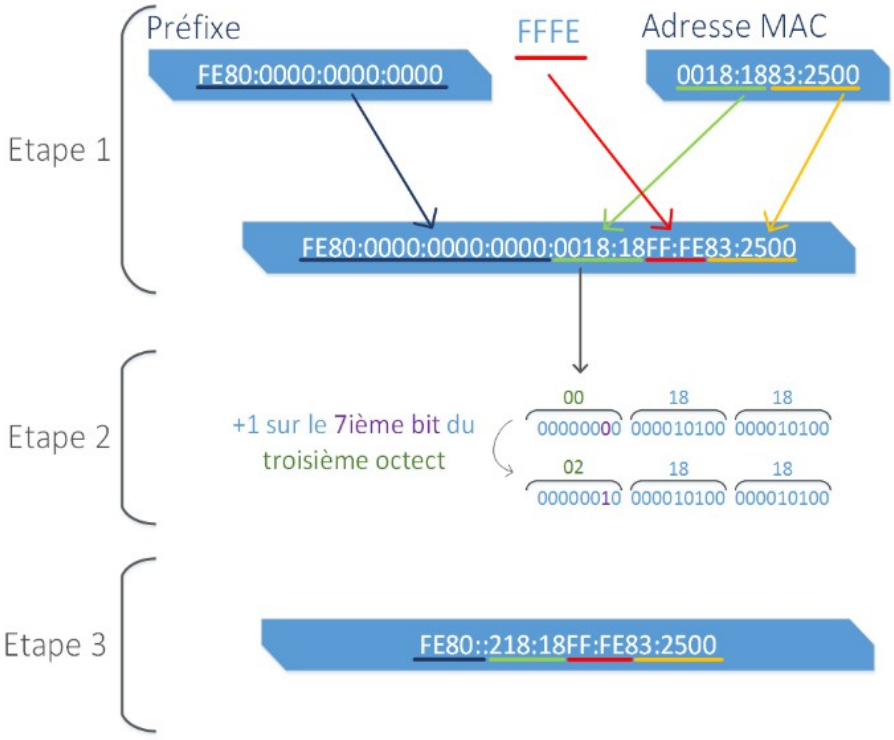
-RFC 4862 : appelé configuration sans état basée sur l'adresse Mac pour la construction de EUI-64 (norme) qui utilise le protocole NDP, utilisé par les systèmes linux
-RFC 4941 : auto configuration avec une valeur aléatoire pour la création de EUI-64 utilisé par les clients système Windows
-RFC 3315: DHCPv6 -cf chapitre autoconf statefull

L'EUI-64 :

EUI-64 pour **"Extended Unique Identifier"** ou "**identifiant unique étendu**" et **64 = /64** qui **correspond aux 64 bits de poid faible de l'adresse ipv6** identifiant l'interface est une façon de former les adresses IPv6 de type **unicast – lien local**. Cette méthode de formation des adresses est unique car elle se base, pour se former, de l'adresse MAC de la carte réseau qu'elle utilise. Pour rappel, les adresses MAC sont des identifiants uniques à chaque carte réseaux.

cela permet à un hôte de s'attribuer à lui même une adresse IPv6. C'est un plus par rapport à l'IPv4 qui nécessitait aux postes, pour avoir une IP afin de communiquer, de **repérer un serveur DHCP et de lui demander un IP.**

Le processus de construction de l'adresse ipv6 en EUI-64 se fait en trois étapes



En prend le préfixe **FE80:0000:0000:0000** et l'adresse MAC **0018:1883:2500** de la carte réseau , on combine le préfixe + premiers octets de l'@ Mac + FFFE + 3 derniers octets de l'adresse MAC

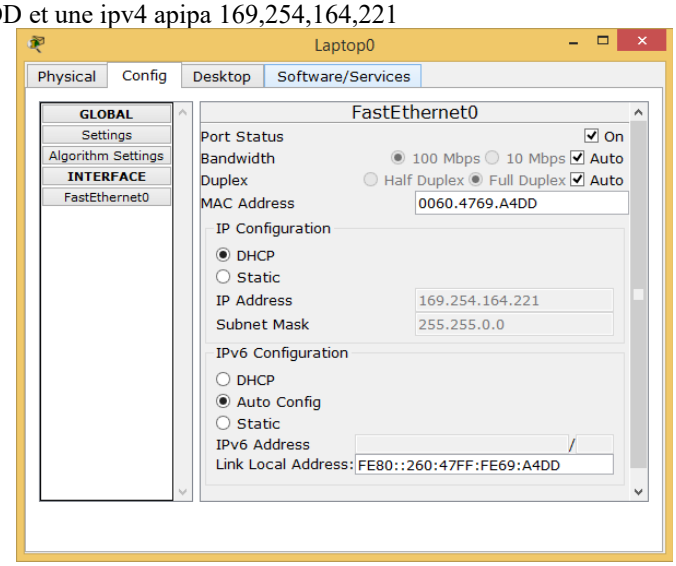
On effectue une modificaton sur le septième bits du premiers octet de l'adresse Mac sur lequel on va faire +1 qui modifie sa valeur décimal

En enléve les 0 inutiles

Gâce au protocole NDP et notamment a la fonction Duplicate Address Detection - DAD , elle s'assure de l'unicité de l'adresse généré par le message "Neighbor solicitation ICMPv type 135

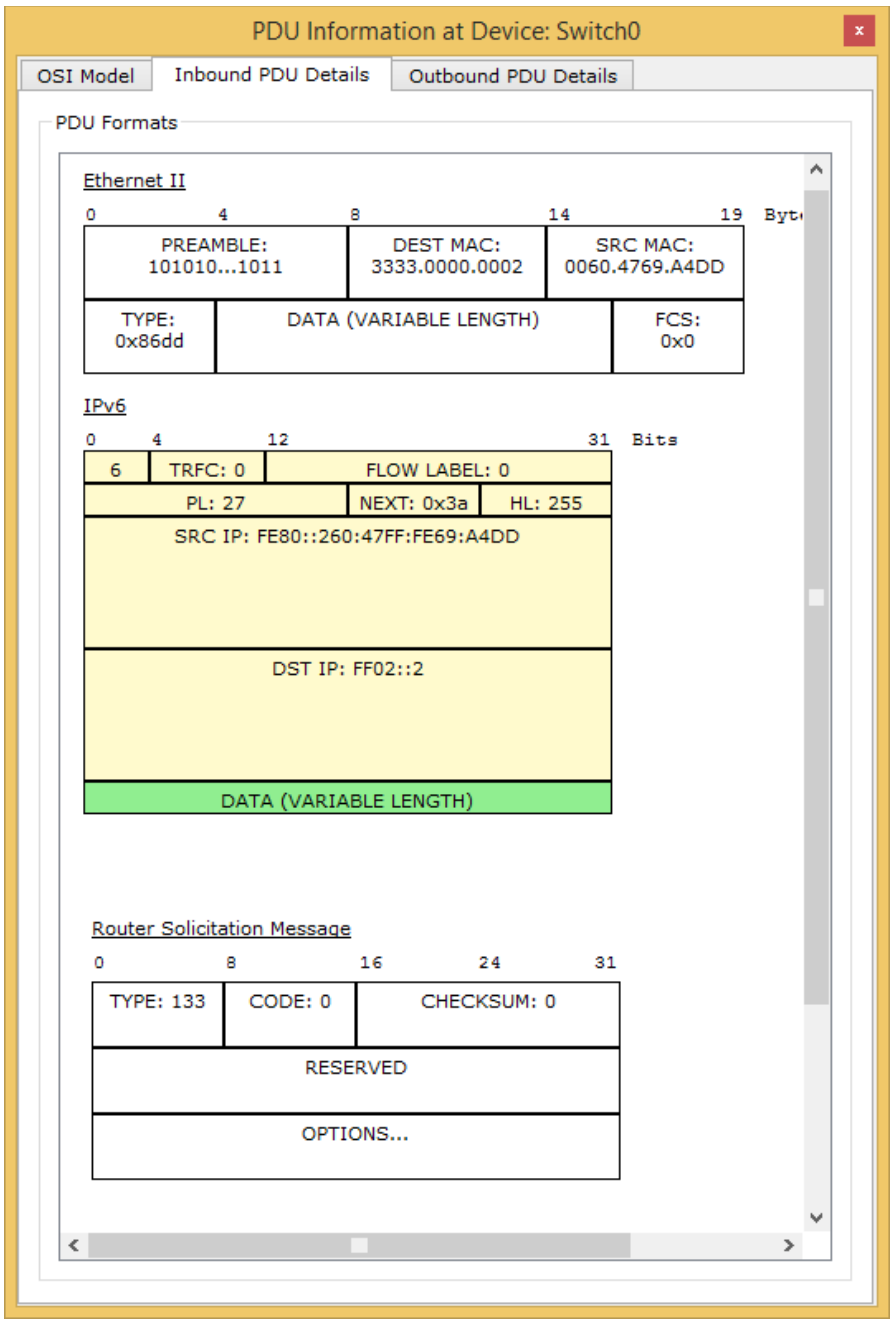
Exemple :

Mon Pc n'ai pas connecté mais a l'option de configuration double-pile , il possède une adresse de lien local unique FE80::260:47FF:FE69:A4DD et une ipv4 apipa 169,254,164,221

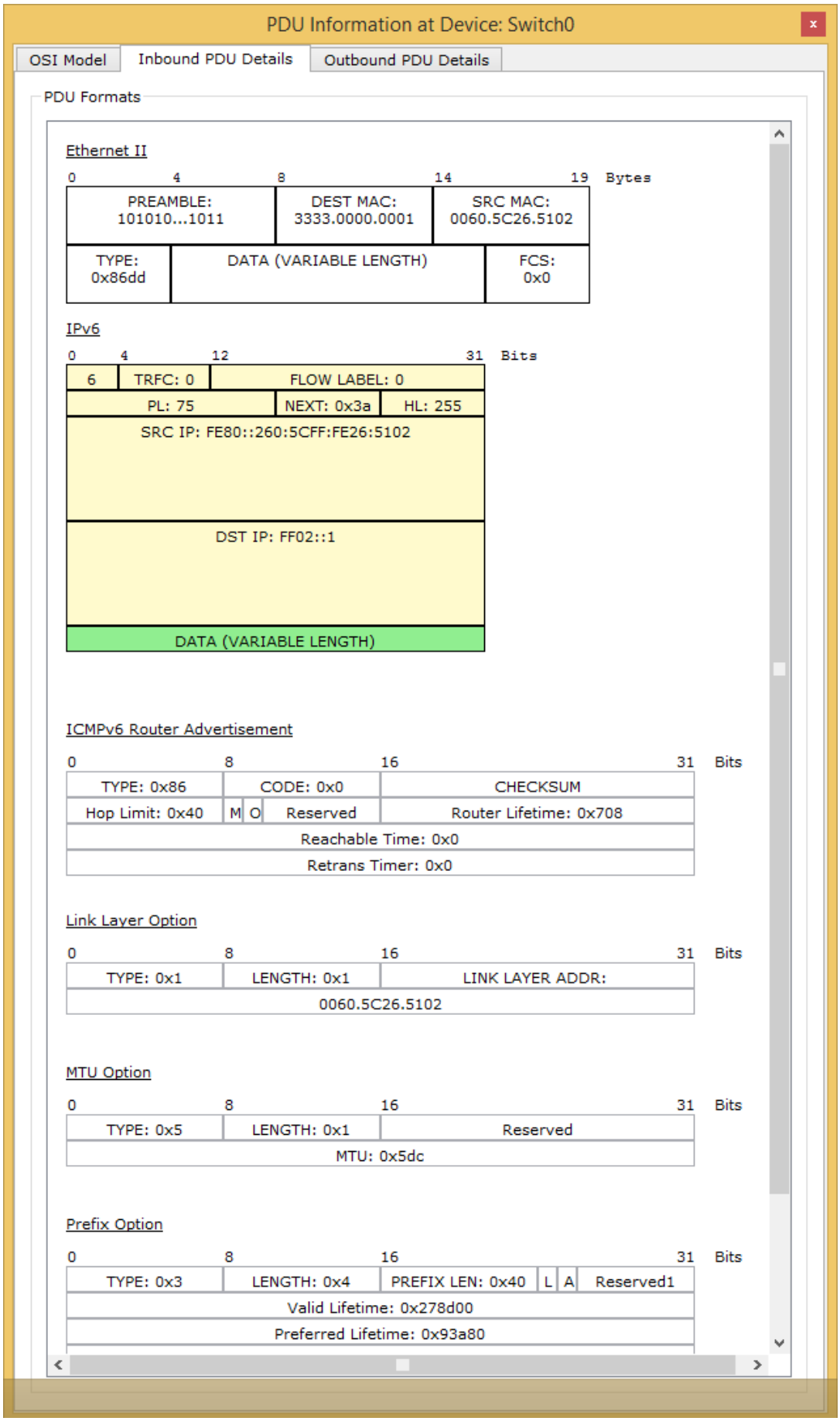


Je le connecte sur un switch , aucune modification même adressage lien local et ipv4
Je connecte mon router sur le swicth via sa giga0/1

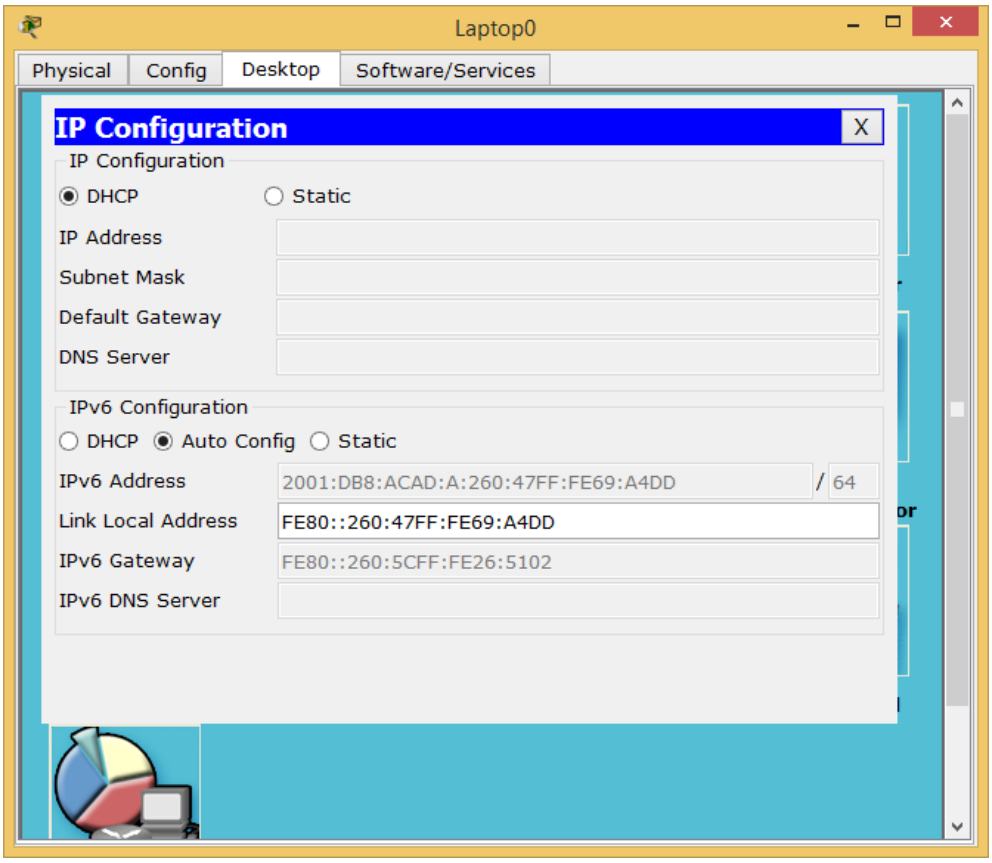
ipv6 unicast-routing
!
interface GigabitEthernet0/1
no ip address
duplex auto
speed auto
ipv6 address 2001:DB8:ACAD:A::/64 eui-64
!
le client PC envoi une requête NDP de **sollicitation** pour trouver un router


PDU Information at Device: Switch0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
0 4 8 14 19 Byt
PREAMBLE: 101010...1011
DEST MAC: 3333.0000.0002
SRC MAC: 0060.4769.A4DD
TYPE: 0x86dd
DATA (VARIABLE LENGTH)
FCS: 0x0
IPv6
0 4 12 31 Bits
6
TRFC: 0
FLOW LABEL: 0
PL: 27
NEXT: 0x3a
HL: 255
SRC IP: FE80::260:47FF:FE69:A4DD
DST IP: FF02::2
DATA (VARIABLE LENGTH)
Router Solicitation Message
0 8 16 24 31
TYPE: 133
CODE: 0
CHECKSUM: 0
RESERVED
OPTIONS...

le router retourne un message NDP **advertissement**



et voila notre Pc équipé d'une adresse ipv6 et gateway ipv6 obtenu grâce au protocole NDP en 2 échange



un test de ping a partir du pc sur giga 0/1 du router

PC>ping 2001:DB8:ACAD:A:260:5CFF:FE26:5102

Pinging 2001:DB8:ACAD:A:260:5CFF:FE26:5102 with 32 bytes of data:

Reply from 2001:DB8:ACAD:A:260:5CFF:FE26:5102: bytes=32 time=1ms TTL=255

configuration des interfaces du routeur

Router#sh ipv6 interface brie
GigabitEthernet0/0 [administratively down/down]
GigabitEthernet0/1 [up/up]
FE80::260:5CFF:FE26:5102
2001:DB8:ACAD:A:260:5CFF:FE26:5102
Serial0/0/0 [administratively down/down]
Serial0/0/1 [up/up]
FE80::260:5CFF:FE26:5101
FC00::1
Vlan1 [administratively down/down]

ping de l'interface lien local du router

PC>ping FE80::260:5CFF:FE26:5102

Pinging FE80::260:5CFF:FE26:5102 with 32 bytes of data:

Reply from FE80::260:5CFF:FE26:5102: bytes=32 time=0ms TTL=255

notre configuration du pc est valide et peut donc accéder a internet

2. autoconf statefull – serveur DHCPv6

utilise un serveur DHCP

même concept que l'Ipv4

utilisation d'adresse de multicast pour les requêtes FF02::1:2 et FF05::1:3

les avantages

permet de passer de nombreuses options : NTP, DNS ...

gestion centralisée via un serveur

3. autoconf stateless DHCPv6

Mélange des configurations précédentes

l'équipement utilise l'autconf stateless et s'affecte leur IP- lien local, puis **récupèrent uniquement les options de configuration type DNS , Ntp ... via un serveur DHCP**

Notion de découpe de sous réseau dans un réseau LAN– ipv6

Pour accéder a internet chaque poste-pc se voit attribuer une adresse ipv6 global unique , dans le cadre de sous réseaux locaux on procédera a une découpe sur l'adresse global

bits	3+45	16	64
champ	<topologie public>	<sous-réseaux>	<ident-interface>

on pourra donc jouer entre **/48 et /64** pour faire les sous-réseaux

Traduction Ipv6 / IPv4

Le DUAL STACK

Mécanisme de coexistence Ipv4 et Ipv6 est le plus déployé a l'heure actuelle, il nécessite une double configuration de tous les composants de communication, tous les systèmes d'exploitation utilise une double pile.
Le terme de double pile fait référence s'explique par le fait que chaque système gère une double pile de protocole de la **couche 7 applicative a la couche réseau 3** , par opposition au contexte de migration prévue ou seul le protocole de la couche réseau devait être concerné par la double pile.

les routeurs dual stack support l'ipv6 et ipv4
chaque équipement possède une Ipv4 et Ipv6
pour les requêtes Dns , l'ipv6 est toujours prioritaire

Si un hôte est configuré avec 2 adresse ipv4/v6, et qu'une application se connecte a un hôte distant l'adresse ip utilisée dépendra de la réponse du serveur DNS a savoir si la résolution de nom est de type ipv4 ou ipv6 (enregistrement AAAA)

Exemple :

Dans ce cadre le serveur DNS nous a bien retourner l'adresse Ipv6 de google suite a un ping ipv6,
nous observons l'enregistrement AAAA local du PC , les commande ne marche pas car le pc gère le dual-stack et non le routeur que ce soit en local ou côté WAN (TEST 1), ce pourquoi le PC n'a pas d'adresse globale attribuée par le routeur et routable sur internet.

Concernant le TEST 2 tout est valide.
Dans le cadre d'un réseau local gérant le dual-stack le choix de ipv6 aurait était prioritaire, dans la cas ou l'application gére l'ipv6, si une application doit choisir entre une adresse ipv4 ou ipv6 il préférera ipv6 idem pour les navigateurs qui utilisent l'algo "happy eyeballs" (algo publié par l'IETF qui permet rendre les applications compatible double stack, permettant d'éviter les problèmes de connexions ipv6 imparfaites.

TEST 1

C:\Windows\System32>ping -6 www.google.com

Envoi d'une requête 'ping' sur www.google.com [2a00:1450:400c:c03::69] avec 32 octets de données :
Délai d'attente de la demande dépassé.

C:\Windows\System32>ipconfig /displaydns

Configuration IP de Windows

www.google.com

Nom d'enregistrement. : www.google.com
Type d'enregistrement : 28
Durée de vie . . . . : 112
Longueur de données . : 16
Section . . . . . : Réponse
Enregistrement AAAA . : 2a00:1450:400c:c03::69

C:\Windows\System32>tracert -6 2a00:1450:400c:c03::69

Détermination de l'itinéraire vers we-in-x69.1e100.net [2a00:1450:400c:c03::69]
avec un maximum de 30 sauts :

  1   *   *   *   Délai d'attente de la demande dépassé.

TEST 2

C:\Windows\System32>ipconfig /flushdns

Configuration IP de Windows

Cache de résolution DNS vidé.

C:\Windows\System32>ping www.google.com

Envoi d'une requête 'ping' sur www.google.com [173.194.45.83] avec 32 octets de données :
Réponse de 173.194.45.83 : octets=32 temps=35 ms TTL=56

C:\Windows\System32>ipconfig /displaydns

Configuration IP de Windows

    www.google.com

    Nom d'enregistrement. : www.google.com
    Type d'enregistrement : 1
    Durée de vie . . . . : 175
    Longueur de données . : 4
    Section . . . . . : Réponse
    Enregistrement (hôte) : 173.194.45.83

@réseau Ipv6 – int @ipv6====(R)==== int @ipv4 ------- int @ipv4 ====(R)==== – int @ipv6 ---------@réseau ipv6

le tunneling 6 in 4

il permet de relié 2 réseaux natifs ipv6 isolés

@réseau Ipv6 – int @ipv6-R- @ipv4 –<-----tunnel ipv4-----> – R – int @ipv6 ---------@réseau ipv6

il encapsule ipv6 dans des paquets ipv4

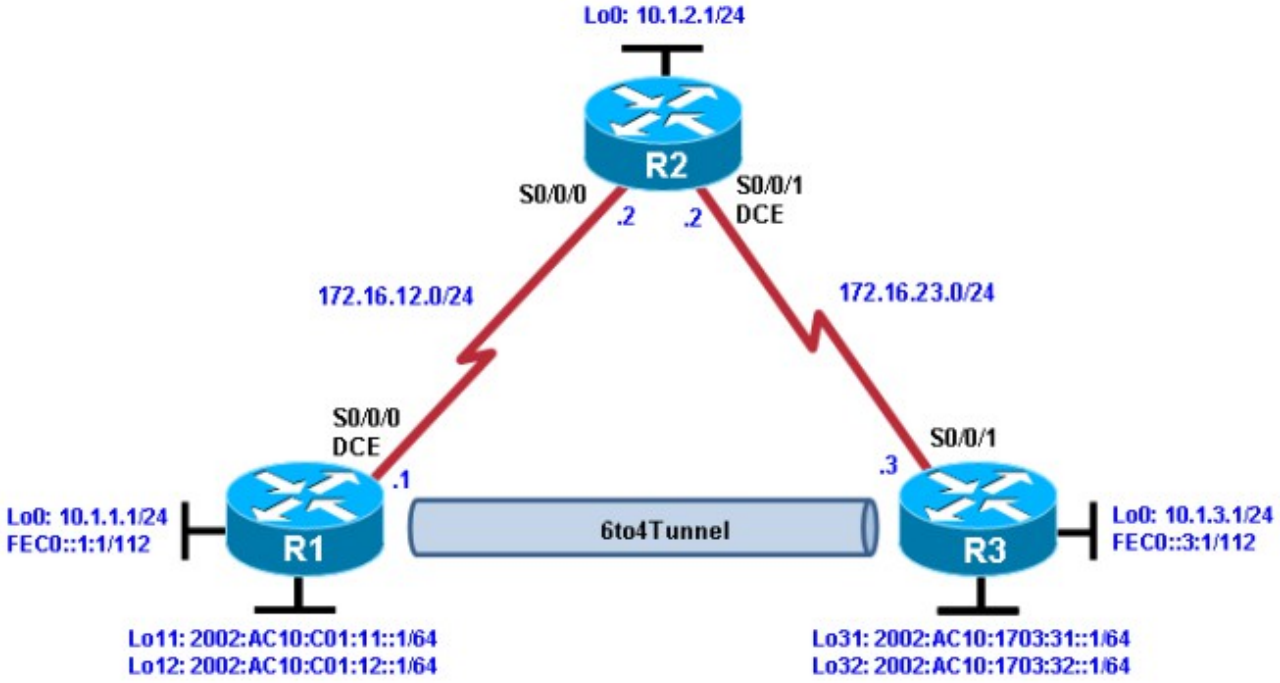
prefix 2002::/16 + ipv4 router du point d'entrée du réseau

mécanisme automatique de tunneling entre ipv4 et le réseau ipv6

3 étape pour la mise en place

assigner un block ipv6 a un réseau local qui a une global ipv4
encapsuler ipv6 dans ipv4 – 6in4
router le traffic entre 6to4 et le réseau ipv6 natif

Développement



un tunnel est une interface logique – loopback qui agit comme une connexion logique entre 2 points d'extrémités , il n'y a pas d'interface physique correspondant , mais des routeurs reste impliqués, un tunnel ipv6 utilise des adresses ipv6 spéciales dans l'espace d'adressage 2002::/16 ,
les premiers 16 bits sont égales en hexa a 2002 et les 32 bits suivants sont l'adresses ipv4 de l'interface Wan public d'entré du réseau. Un tunnel ipv6 ne nécessite pas d'adresse de destination , il n'est pas une liaison point a point .

1. Sur R1 cfg vos périphériques du réseau local en ipv6 2002:AC10:0c01:11::1/64 et 12::1/64
sachant que AC10:0C01 correspondant a votre adresse ipv4 Wan public d'entré de votre réseau 172.16.12.1 et sur R2 – 2002:AC10:1703:31::1/64 et 2002:AC10:1703:32::1/64 et 172.16.23.3 est égale a AC10:1703

Déclaration des loopback 0 affectant adresse du lien local utilisé pour le routage statique

```
R1(config)# interface loopback0
R1(config-if)# ip address 10.1.1.1 255.255.255.0
R1(config-if)# ipv6 address FEC0::1:1/112
R1(config-if)# interface serial0/0/0
R1(config-if)# ip address 172.16.12.1 255.255.255.0
R1(config-if)# clockrate 64000
R1(config-if)# bandwidth 64
R1(config-if)# no shutdown
```

```
R3(config)# interface loopback0
R3(config-if)# ip address 10.1.3.1 255.255.255.0
R3(config-if)# ipv6 address FEC0::3:1/112
R3(config-if)# interface serial0/0/1
R3(config-if)# ip address 172.16.23.3 255.255.255.0
R3(config-if)# bandwidth 64
R3(config-if)# no shutdown
```

//dans cette configuration on affectera aux loopback / équipements / pc des adresse ipv6 globales 6to4 dit six to for , globale parce qu'elles doivent être routable sur internet et 2002: car ce sont des adresses réservées pour l'utilisation des piles 6to4

```
R1(config-if)# interface loopback11
R1(config-if)# ipv6 address 2002:AC10:0C01:11::1/64          //utilisation d'adresse global / PC
R1(config-if)# interface loopback12
R1(config-if)# ipv6 address 2002:AC10:0C01:12::1/64
R3(config-if)# interface loopback31
R3(config-if)# ipv6 address 2002:AC10:1703:31::1/64
R3(config-if)# interface loopback32
R3(config-if)# ipv6 address 2002:AC10:1703:32::1/64
```

2. création du tunnel

```
R1(config)# interface tunnel 0
R1(config-if)# tunnel mode ipv6ip 6to4
R1(config-if)# ipv6 address 2002:AC10:0C01:1::1/64
R1(config-if)# tunnel source serial0/0/0
R1(config-if)# exit
R1(config)# ipv6 unicast-routing              //activation du routage ipv6
R1(config)# ipv6 route 2002::/16 tunnel0      //déclaration du réseau 2002::/16
                                              //lan interne connecté au tunnel0
```

```
R3(config)# interface tunnel 0
R3(config-if)# tunnel mode ipv6ip 6to4
R3(config-if)# ipv6 address 2002:AC10:1703:1::3/64
R3(config-if)# tunnel source serial0/0/1
R3(config-if)# exit
R3(config)# ipv6 unicast-routing
R3(config)# ipv6 route 2002::/16 tunnel0
```

3. routage statique

après avoir déclaré le réseau local attaché au tunnnel 0, il faut configurer les réseaux a atteindre sur R3 , on utilisera en passerelle l'adresse ipv6 du tunnel a atteindre

```
R1(config)# ipv6 route FEC0::3:0/112 2002:AC10:1703:1::3
R3(config)# ipv6 route FEC0::1:0/112 2002:AC10:C01:1::1
```

cf: Chapter 8 Lab 8-3, Configuring 6to4 Tunnels

ANNEXE

clear ndp
ndpmonitor : NDPMon implémenté et disponible sous licence LGPL

MAC Address (Vendor)	Link Local Address	IPv6 Global Address
0:30:b6:51:d4:1c (Cisco)	fe80:0:0:0:230:b6ff:fe51:d41c	2001:660:4501:1:0:0:0:1
0:f:1f:bb:5f:aa (WwPcbaTest)	fe80:0:0:0:e906:a182:1faa:bcb5	2001:660:4501:1:1181:5db0:b069:e9d8 2001:660:4501:1:6027:a6ce:72e4:7ee0 2001:660:4501:1:4dc1:ee67:2824:cd53
0:1b:63:a1:bf:62 (Apple)	fe80:0:0:0:21b:63ff:fea1:bf62	2001:660:4501:1:21b:63ff:fea1:bf62
0:17:f2:c6:53:db (AppleCompu)	fe80:0:0:0:217:f2ff:fec6:53db	2001:660:4501:1:217:f2ff:fec6:53db
0:14:51:21:aa:4e (AppleCompu)	fe80:0:0:0:214:51ff:fe21:aa4e	2001:660:4501:1:214:51ff:fe21:aa4e
0:12:3f:76:e3:bd (Dell)	fe80:0:0:0:212:3fff:fe76:e3bd	2001:660:4501:1:212:3fff:fe76:e3bd 2002:9851:190:7:212:3fff:fe76:e3bd
0:13:72:35:7:45 (Dell)	fe80:0:0:0:213:72ff:fe35:745	2001:660:4501:1:213:72ff:fe35:745 2002:9851:190:7:213:72ff:fe35:745
0:14:22:b6:5e:0 (Dell)	fe80:0:0:0:214:22ff:feb6:5e00	2001:660:4501:1:214:22ff:feb6:5e00
8:0:46:db:18:5d (Sony)	fe80:0:0:0:a00:46ff:fedb:185d	2001:660:4501:1:a00:46ff:fedb:185d
0:c:f1:83:57:58 (Intel)	fe80:0:0:0:20c:f1ff:fe83:5758	2001:660:4501:1:20c:f1ff:fe83:5758
0:18:f3:f6:a1:1b (AsustekCom)	fe80:0:0:0:218:f3ff:fef6:a11b	2001:660:4501:1:218:f3ff:fef6:a11b 2002:9851:190:7:218:f3ff:fef6:a11b
0:b:db:d3:ee:95 (DellEsgPcb)	fe80:0:0:0:20b:dbff:fed3:ee95	2001:660:4501:1:20b:dbff:fed3:ee95
0:c:f1:72:87:aa (Intel)	fe80:0:0:0:20c:f1ff:fe72:87aa	2001:660:4501:1:20c:f1ff:fe72:87aa
0:11:11:4c:c6:42 (Intel)	fe80:0:0:0:211:11ff:fe4c:c642	2001:660:4501:1:211:11ff:fe4c:c642 2002:9851:190:7:211:11ff:fe4c:c642