Laboratoire

Ipv6

<Lienhard Steve / TR>

<Burkihalter Arnaud / TR>

# Objectifs du laboratoire

Ce laboratoire étudie le protocole IP Version 6.

Les objectifs de ce laboratoire sont:

* Connaître les différents types d'adresses IPv6:
* adresses link-local
* adresses globales
* adresses multicast
* Comprendre l'auto-configuration sans état d'IPv6
* Comprendre le fonction des messages « Neighbor solicitation » et « Neighbor advertisement » qui remplacent le protocole ARP
* Comprendre les mécanismes les plus important pour l'interoperabilité entre IPv4 et IPv6:
* la double pile IPv4 / IPv6
* les tunnels manuels entre IPv6 et IPv4
* les tunnels 6to4
* Savoir configurer des interfaces réseau pour IPv6 sous Linux

# Prérequis

Le laboratoire nécessite les connaissances pratiques suivantes, acquises dans le laboratoire LAN1:

* Configuration d'interfaces réseau avec ifconfig
* Utilisation de l'analyseur de paquets ethereal
* Utilisation de ping

En plus de ces connaissances, ce laboratoire demande des connaissances théoriques sur le protocole IPv6, présenté dans le Chapitre La couche réseau du cours.

Outils et commandes

Commandes à utiliser dans ce laboratoire:

|  |  |
| --- | --- |
| **Ajouter une adresse IPv6 à une interface** | |
|  | ifconfig *interface* add *adresse\_ipv6/len\_préfixe*  **Exemple**: ifconfig eth0 add fe80::1/64 |
| **Effacer une adresse IPv6 d'une interface** | |
|  | ifconfig *interface* add *adresse\_ipv6/len\_préfixe*  **Exemple**: ifconfig eth0 add fe80::1/64 |
| **Activer l'acheminement IPv6 sous Linux** | |
|  | echo 1 > /proc/sys/net/ipv6/conf/all/forwarding |
| **Configurer la passerelle par défaut d'une station IPv6** | |
|  | route -A inet6 add default gw *adr\_ipv6\_passerelle* |
| **Afficher les routes IPv6** | |
|  | Ancienne commande : route -A inet6  Nouvelle commande : ip -6 route |
| **Création d'une interface IPv6 virtuel de tunnel** | |
|  | ip tunnel add *if* mode sit ttl *v\_ttl* remote *adr\_dst* local *adr\_locale*  **Exemple**: ip tunnel add iftun mode sit ttl 64 remote any local 1.1.1.1 |

Questions théoriques

Répondez aux questions suivantes en utilisant le support du cours, Chapitre Réseau.

Questions I

* Une interface IPv6 a en général plusieurs adresses. Citez les 2 différents types d'adresses IPv6 qui sont définies actuellement et expliquer leur utilisation. (3p)

**Préfixe Type d'adresse Utilisation**

2003 ::/3 Adresse globale

Ces adresses sont les adresses publiques utilisées pour adresser les interfaces de toutes les machines de l'Internet globale.

FE80 ::/10 et FEC0 ::/10 Adresse locale de lien et de site

Les adresses locales de lien sont des adresses restreintes à un LAN et les adresses locales de site sont restreintes à un site (plus haut dans la hiérarchie).

* Considérez les adresses IPv6 2001:1:4:1::2.

a) Quel est le type de cette adresse (locale, globale, ...) ? (1p)

Adresse globale, donc routable sur internet.

b) Quelles sont les valeurs du préfixe de routage, identificateur de sous-réseau et identificateur d'interface de cette adresse ? (3p)

Préfixe de routage : 2001:1:4

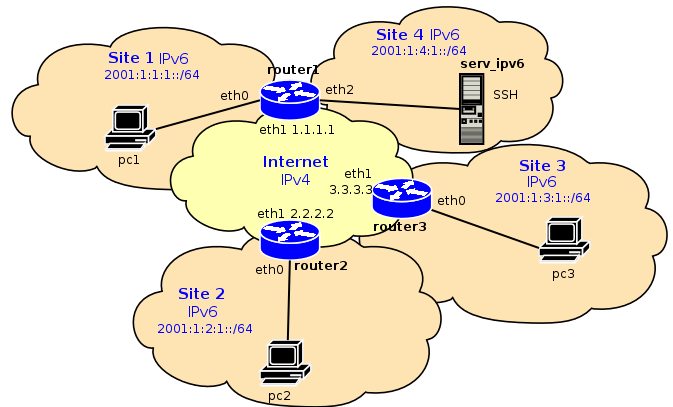
Identificateur de sous-réseau : 1

Identificateur d'interface : 0:0:0:2

Pratique

# Réseau

Pour ce laboratoire, nous allons utiliser le réseau montré à l'Illustration 1.

Illustration 1: Réseau de test IPv6 et IPv4

Les adresses IPv4 et IPv6 des différentes interfaces sont:

| Noeud | Interface | Type | Adresse |
| --- | --- | --- | --- |
| router1 | eth0 | IPv6 globale | 2001:1:1:1::1/64 |
|  | eth1 | IPv4 (Internet) | 1.1.1.1 netmask 255.0.0.0 |
|  | eth2 | IPv6 globale | 2001:1:4:1::1/64 |
| router2 | eth0 | IPv6 globale | 2001:1:2:1::1/64 |
|  | eth1 | IPv4 (Internet) | 2.2.2.2 netmask 255.0.0.0 |
| router3 | eth0 | IPv6 globale | 2001:1:3:1::1/64 |
|  | eth1 | IPv4 | 3.3.3.3 netmask 255.0.0.0 |
| pc1 | eth0 | IPv6 globale | 2001:1:1:1::/64 (auto-configuration) |
| pc2 | eth0 | IPv6 globale | 2001:1:2:1::2/64 |
| pc3 | eth0 | IPv6 globale | 2001:1:3:1::2/64 |
| serv\_ipv6 | eth0 | IPv6 | 2001:1:4:1::2/64 |

La configuration du réseau est décrite dans le fichier ipv6.conf. Démarrez le réseau avec la commande :

>>> nemo ipv6.conf

Pour faciliter l'orientation, minimisez les consoles de tous les noeuds sauf pc1 et router1 pour la première expérience.

# Adresses IPv6

Contrairement à une interface IPv4, une interface IPv6 a normalement plusieurs adresses IP. Elle a au minimum une adresse du type *link-local* pour la communication avec les autres noeuds dans son LAN. Normalement elle a également une adresse globale, pour la communication avec le réseau public (Internet)

Comme vu au cours, les adresses locales sont configurées avec un préfixe fixe et l'adresse MAC de l'interface comme Interface ID.

## Auto-configuration sans état

IPv6 offre de nouveaux mécanismes pour la configuration des adresses IP des machines : l'auto-configuration sans et avec état. L'auto-configuration sans état est similaire à la configuration avec DHCP et utilise un DHCPv6 pour assigner des adresses IPv6. Dans l'expérience suivante nous allons étudier l'auto-configuration sans état.

###### Configuration

Lancez ethereal comme tâche de fond sur router1. Capturez les messages sur l'interface eth0 (connectée à pc1).

Puis désactivez et réactivez l'interface eth0 sur pc1 avec les commandes :

pc1> ifconfig eth0 down  
pc1> ifconfig eth0 up

Questions II

* Expliquez le message « Neighbor solicitation » utilisé lors de la configuration de l'adresse link-local. (3p).

L'interface après avoir construit une adresse locale de lien par rapport à son adresse mac va envoyer le message « Neighbor Solicitation » en multicast sur le réseau local. Si elle ne reçoit aucune réponse cela veut dire que cette adresse est unique et qu'elle peut donc l'utiliser. Si au contraire elle reçoit une réponse, cela signifie que cette adresse est déjà utilisée par une autre machine et qu'elle ne peut de ce fait pas être utilisée.

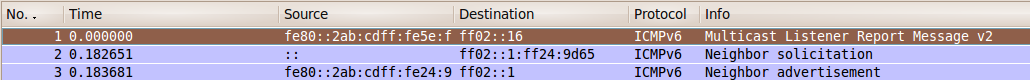
* Expliquez la fonction des adresses IPv6 multicast utilisées par les messages « Neighbor solicitation » et « Router solicitation » (RFC 2373, section 2.7.1) (2p)

Les adresses multicast utilisées sont des adresses permettant d'avoir des communications dans un LAN sans risquer une diffusion des datagrammes dans tout l'Internet.

* Tout en capturant les messages sur router1, configurez manuellement la même adresse IPv6 link-local que pc1 sur l'interface eth0 de router1:

routeur1> ifconfig eth0 add fe80::2ab:cdff:fe5e:f

N'oubliez pas le 'add' qui est nécessaire pour IPv6 !  
Cela crée un conflit d'adresses. Qu'observez-vous ? (2p)



Nous remarquons que le routeur 1 teste si la nouvelle adresse de eth0 est unique sur le réseau local par le message « Neighbor solicitation ». Comme il obtient une réponse par « Neighbor advertisement », il en conclu que cette adresse est déjà utilisée par une autre machine. Tout comme dans IPV4, il est indispensable dans ce cas là d'assigner une autre adresse IP à cette interface afin de ne pas créer de conflit.

Effacez l'adresse configurée manuellement sur router1 avec la commande :

routeur1> ifconfig eth0 del fe80::2....

## Auto-configuration sans état d'une adresse globale

L'obtention d'une adresse link-local permet à une station de communiquer avec tous les autres noeuds du même LAN. Comme nous l'avons vu dans l'expérience précédente, le pc1 essaye de contacter un routeur (message Router solicitation) pour demander le préfixe de l'adresse globale. Comme aucun des routeurs n'est configuré pour répondre, cette sollicitation n'obtient pas de réponse.

###### Configuration de router1

La première chose à faire sur un routeur IPv6 est d'activer le routage. La commande pour ceci est similaire à l'activation du routage IPv4:

router1> echo 1 > /proc/sys/net/ipv6/conf/all/forwarding

Le router1 sera responsable pour le réseau avec le préfixe 2001:1:1:1::/64. Le '/64' à la fin a la même signification qu'un masque de sous-réseau pour IPv4. Il indique que le préfixe du réseau comprend les premiers 64 bits de l'adresse. Le reste des bits identifie une machine à l'intérieur de ce réseau. Si lors de la configuration d'une adresse IPv6 sur un routeur, la gestion des routes ne sera pas correcte.

Nous allons configurer l'interface eth0 sur router1 avec une adresse du réseau 2001:1:1:1::/64:

router1> ifconfig eth0 add 2001:1:1:1::1/64

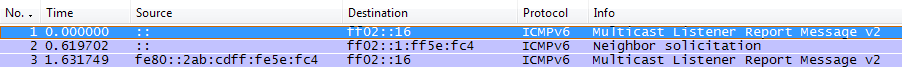
Finalement, il faut démarrer le démon Linux qui répond aux sollicitations de routeurs. Ce démon est déjà configuré correctement à l'aide du fichier /etc/init.d/radvd/radvd.conf:

router1> /usr/sbin/radvd -u radvd

Vérifiez avec la commande 'ps aux' que le démon a démarré correctement.

Questions III

* Sur pc1 désactivez et réactivez l'interface eth0, tout en capturant le trafic sur router1. Expliquez les messages échangés (avec une capture d'écran). (3p)



Nous pouvons voir sur la capture ci-dessus, que lors de la réactivation de l’interface, le pc va à nouveau effectuer un  « neighbor solicitation » pour vérifier que personne ne possède la même adresse, cependant il n’y a pas de « routeur solicitation » car la machine connaît déjà le préfixe utilisé.

- Neighbor solicitation :

Le pc forme une adresse IPV6 à l'aide de son adresse MAC et vérifie ensuite si cette adresse est disponible. Comme mentionné précédemment, si elle ne reçoit pas de « Neighbor Advertisement », l'adresse peut être utilisée.

Les messages qui suivent ne sont pas présents dans notre cas (puisque la machine connait déjà le prefixe du réseau) mais pourraient survenir dans le cas contraire :

- Router solicitation :

Le pc demande aux routeurs de s'annoncer afin qu'il puisse connaitre le préfixe de l'adresse réseau.

- Routeur Advertisement :

Le routeur répond au pc1 ce qui permet à ce dernier de construire l'adresse globale en concaténant son identificateur d'interface à au préfixe du réseau reçu par le routeur.

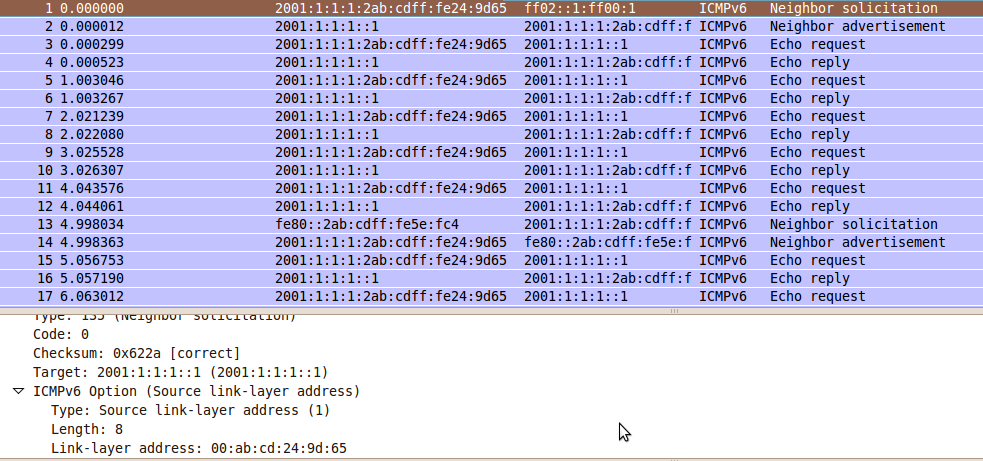
## Résolution d'adresses MAC

Dans IPv4, le protocole ARP s'occupe de la résolution d'adresses MAC en adresses IPv4. Ce protocole n'est plus utilisé dans IPv6 et la fonction de résolution d'adresses MAC est réalisée par des messages ICMPv6.

Depuis pc1, effectuez un **ping6** sur l'interface eth0 de router1 et capturez les messages échangés.

Questions IV

Présentez et commentez les principaux paramètres des messages ICMPv6 observés. En particulier, mettez en évidence l'obtention d'une adresse MAC distante (équivalent du protocole ARP). (6p)



La capture ci-dessus nous montre que le premier message envoyé est celui de pc1 qui effectue un « Neighbor solicitation » en multicast afin de déterminer qui a l'adresse IP de destination du ping (2001:1:1:1::1 dans notre cas). Le routeur 1 répond ensuite par un Neighbor advertisement que c'est lui qui détient cette adresse. Ces deux messages sont l’équivalent d'une requête ARP pour IPV4. Une fois que le pc1 sait quelle adresse MAC correspond à l'adresse IPV6, il peut envoyer son ping.

* Effectuez deux ping6 vers la même station. D’abords très rapprochés puis, attendez 3 minutes et effectuez à nouveau le même ping6. Existe-t-il en IPv6 aussi un cache ? (1p)



La capture ci-dessus nous permet de constater qu'il existe effectivement un cache ICMPv6 mais que celui-ci est très court, environ 5 secondes dans notre cas.

# Format des datagrammes IPv6

IPv6 a été conçu pour fonctionner dans des réseaux de prochaine génération à très haut débit. IPv4 impose une charge considérable aux routeurs pour manipuler les paquets. Dans IPv6, l'entête des paquets a une structure différente, dans le but de réduire les opérations à effectuer sur les routeurs.

Questions V

* A partir des mesures effectuées lors de l'expérience précédente, illustrez les principales différences entre les datagrammes IPv6 et IPv4, en particulier le checksum et le Time To Live. (2p)

Le champ **Time To Live** d’IPv4, est conservé dans IPv6, mais a toutefois été renommé en Nombre de sauts afin de mieux refléter l’usage véritable du champ.

Le champ **checksum** présent dans l’en-tête des datagrammes IPv4 a été considérablement modifié pour IPv6. Dans la nouvelle version d’IP, ce champ est dorénavant nommé ‘longueur des données’est indique le nombre d’octets de la charge utile du datagramme (en-tête non comptabilisé). Cela a été fait dans le but de ne plus devoir recalculer ce champ à chaque saut, du fait que le champ N° de sauts est modifié, et ainsi accélérer le processus de transmission.

* Effectuez un ping6 –s packetsize (exemple : ping6 –s 3000 2001:1:1:1::1). Déterminez la charge utile maximum (longueur des données encapsulées) d'un datagramme IPv6 sans fragmentation. (1p).

La taille maximum des données est de 1448 bytes.

* Comment IPv6 indique-t-il les informations concernant la fragmentation dans l'entête, les champs More Fragments, Offet et Identifcation d'IPv4 n'étant plus présents dans l'entête ? (1p)

Chaque fragment doit être un multiple de 8 octets. L'en-tête de fragmentation contient le champ « déplacement fragment » qui indique (en multiple de 8 octets), la position du fragment dans la charge utile du datagramme complet. Enfin, un bit M indique par la valeur 0 si ce fragment est le dernier ou par la valeur 1 si d'autres fragments suivent.

# Acheminement IPv6

L'acheminement de datagrammes IPv6 fonctionne d'après les mêmes principes que dans IPv4. Seules les commandes de configuration sous Linux sont légèrement différences.

* Configurer la passerelle par défaut sur pc1 et serv1:

pc1> route **-A inet6** add default gw 2001:1:1:1::1  
serv\_ipv6> route **-A inet6** add default gw 2001:1:4:1::1

Après cette configuration, vous devez pouvez pinguer serv1 depuis pc1. Vérifiez que le ping fonctionne.

Vous pouvez visualiser les routes sur un noeud avec la commande:

>>> route -A inet6

Mais il y a une nouvelle commande Linux pour la configuration du réseau, la commande 'ip', qui affiche les routes de manière plus lisible:

>>> ip -6 route

# Interopérabilité IPv4 et IPv6

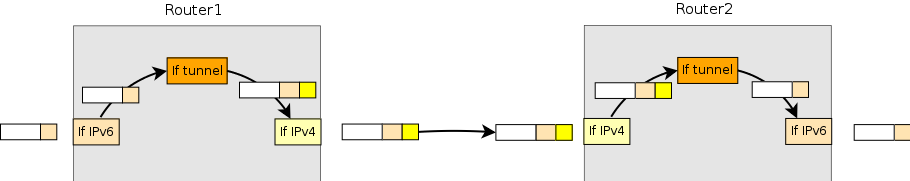
Il est impossible de penser passer à IPv6 sans avoir une cohabitation entre les deux versions du protocole. Il a donc été nécessaire de prévoir des mécanismes qui permettent la transition et la coexistence entre les deux versions du protocole IP.

Il n’est pas toujours possible d’avoir une connectivité IPv6 directe, il est donc nécessaire d’avoir recours aux réseaux IPv4 actuels pour transporter les paquets IPv6.

## Tunnels manuels

La création d'un tunnel peut sembler quelque chose de difficile et exotique, mais en pratique, la configuration se fait de manière aisée. L'Illustration 2 montre le principe d'un tunnel.

Une interface virtuelle ('If tunnel') effectue l'encapsulation et la dé-encapsulation. Les seules informations nécessaires pour pouvoir effectuer cette tâche sont les adresses IPv4 source et destinataire à utiliser dans les entête IPv4 ajoutés.

Illustration 2: Utilisation d'une interface virtuelle pour l'établissement d'un tunnel

Dans notre réseau, nous aimerions interconnecter les réseaux IPv6 à travers un tunnel Ipv6-IPv4 entre router1 et router2. Sur router1, la configuration d'une interface virtuelle se fait de manière suivante:

router1> ip tunnel add sit1 mode sit ttl 48 remote 2.2.2.2 local 1.1.1.1  
router1> ip link set dev sit1 up

Ensuite, il faut encore ajouter une route IPv6 pour dire que tous les paquets vers le réseau 'Site 2' (préfixe 2001:1:2:1::2/64) doivent être routé vers cette interface virtuelle:

router1> route -A inet6 add 2001:1:2:1::2/64 dev sit1 metric 1

Questions VI

* Quelle route IPv4 est nécessaire pour que tous les paquets vers le réseau IPv4 2.0.0.0 soient routés vers l'interface eth1, connectée à Internet ? (2p)

Il est nécessaire de définir les routes suivantes : (mais celles-ci sont déjà configurées !)

router1> route add -net 2.0.0.0 netmask 255.0.0.0 dev eth1

router2> route add -net 1.0.0.0 netmask 255.0.0.0 dev eth1

* Quelles commandes configurent la terminaison du tunnel et les routes correspondantes sur router2 ? (3p)

Nous avons utilisé les mêmes commandes que celles nécessaires pour le routeur1, il a juste fallu changer le réseau ainsi que l’adresse IP.

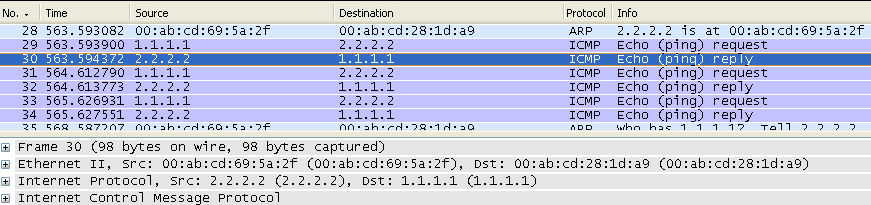
Il s’agit donc des commandes suivantes :

router2> ip tunnel add sit1 mode sit ttl 48 remote 1.1.1.1 local 2.2.2.2

router2> ip link set dev sit1 up

router2> route -A inet6 add 2001:1:1:1::2/64 dev sit1 metric 1

* Testez la configuration en effectuant un ping6 entre pc1 et pc2. Mettez en évidence l'encapsulation des paquets IPv6 à l'aide d'une capture de trafic avec Ethereal sur l'interface eth1 de router1. (2p)



Remarque : Suite à un problème de lisibilité du fichier capturé, nous avons repris la capture venant du groupe de Loic Augsburger et Stéphane Besuchet.

* Comment l'entête IPv4 du paquet indique-t-il qu'il contient un paquet encapsulé et non pas un segment TCP ou autre ? (1p)

Le protocole encapsulé doit être 0x29, ce qui indique une encapsulation ipv6 dans ipv4.

## Tunnels 6to4

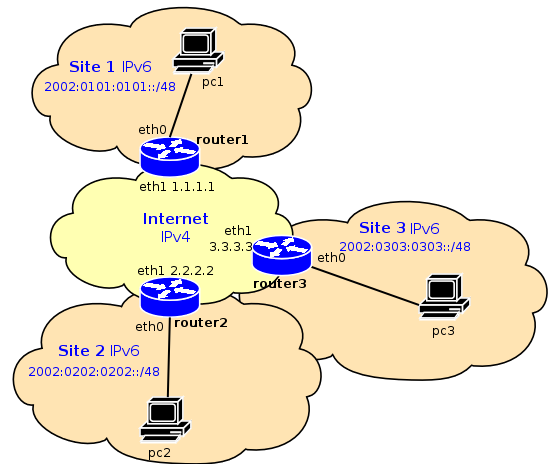
Bien que la configuration d'un seul tunnel soit aisée, cette méthode devient fastidieuse lorsqu'il s'agit d'interconnecter beaucoup de sites à travers des tunnels. En particulier, deux cas de figure se présentent souvent dans des réseaux réels:

1. Communication en IPv6 avec des stations se trouvant dans un réseau IPv4.
2. Interconnexion de beaucoup d'îlots IPv6 à travers Internet en IPv4.

Dans le deux cas une multitude de tunnels serait nécessaire (vers chaque îlot IPv6 respectivement chaque station IPv6). L'IETF a développé des méthodes pour minimiser la configuration des tunnels dans ces cas. Les tunnels automatiques avec des adresses IPv6 compatibles avec IPv4 permettent de résoudre le premier cas de figure.

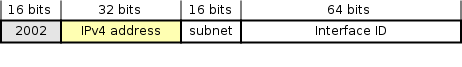
Dans l'expérience suivante, nous allons étudier les tunnels 6to4.

Le scénario d'utilisation des tunnels 6to4 est montré à l'Illustration 3.

Illustration 3 : Scénario d'utilisation de tunnels 6to4.

Chacun des trois routeurs fonctionne comme routeur d'accès d'un site IPv4. Les sites sont interconnectés par l'Internet global en IPv4. Au lieu de configurer des tunnels manuellement, on aimerait que un paquet en destination d'un des sites soit encapsulé en IPv4 et acheminé en IPv4 vers le routeur d'accès approprié.

Mais comment connaître l'adresse IPv4 du router d'accès si on ne connaît que l'adresse IPv6 du réseau pour lequel il est responsable ? La solution est de 'coder' l'adresse IPv4 du routeur d'accès dans l'adresse IPv6 de son site. Le RFC 3056 définit un type d'adresses IPv6, montrée à l'Illustration 4, qui a justement cette fonction.

Illustration 4: Format des adresses 6to4

Ces adresses possèdent le préfixe spéciale 2002 sur 16 bits. Ensuite suit l'adresse IPv4 du routeur d'accès, sur 32 bits. Ce préfixe sur 48 bits constitue le préfixe des adresses du site. Dans notre scénario d'utilisation, le routeur3 a l'interface IPv4 avec l'adresse 3.3.3.3 et le site qui est atteignable à travers lui a le préfixe 2002:0303:0303::/48. Un routeur qui doit acheminer un paquet ver une telle adresse IPv6 sait qu'il doit le 'tunneler' vers l'adresse 3.3.3.3. Ainsi, il n'est plus nécessaire de configurer les tunnels manuellement.

###### Configuration

Il suffit de configurer sur chaque routeur d'accès une interface virtuelle avec une adresse 6to4 correspondante :

router1> ip tunnel add tun6to4 mode sit ttl 64 remote any local 1.1.1.1  
router1> ip link set dev tun6to4 up  
router1> ifconfig tun6to4 add 2002:0101:0101::1

router2> ip tunnel add tun6to4 mode sit ttl 64 remote any local 2.2.2.2  
router2> ip link set dev tun6to4 up  
router2> ifconfig tun6to4 add 2002:0202:0202::1

router3> ip tunnel add tun6to4 mode sit ttl 64 remote any local 3.3.3.3  
router3> ip link set dev tun6to4 up  
router3> ifconfig tun6to4 add 2002:0303:0303::1

Essayez d’effectuer un ping IPv6 entre les routeurs en utilisant leurs adresses 6to4. Cela doit fonctionner.

Questions VII

* Quelles routes IPv4 sont nécessaires pour que les tunnels fonctionnent correctement ? (2p)

Les routes entre chaque routeur, et dans les deux sens. En d’autre terme, dans le cas d’un ping entre les PC1 et PC2, les routes entre router 1 et router 2 et inversement, sont nécessaires.

* Dans quelle situation 6to4 peut-il être utilisé ? (2p)

- Lors de la migration d'un réseau conséquent sur IPv6.

- Lorsqu'une application communique sur IPv6 et qu'elle se trouve dans un réseau IPv4.

* Quel est l’inconvénient majeur du tunneling en ce qui concerne les performances ? (1p)

Il est nécessaire de «réencapsuler » les datagrammes IPv6 dans un datagramme IPv4. Le rajout à la source et l'interprétation à la destination de l'en-tête IPv4 en plus de l'en-tête IPv6 prend du temps

Résumé

Résumez les concepts appris, des questions ouvertes ainsi que d'éventuelles suggestions.

**Concepts appris**

**Types d’adresse**: plusieurs types d’adresses peuvent être définis sur une interface (adresse Globale ainsi qu’adresse local de lien ou de site).

**Auto configuration**: Mécanisme permettant à une machine de se créer une adresse locale.

**Neighbor solicitation** : message multicast envoyé par une machine, et lui permettant de savoir si une adresse est présente sur le réseau. Avec la réponse *Neighbor advertisement*, il s’agit du mécanisme remplaçant ARP pour IPv6.

**Format des datagrammes IPv6** : L’en-tête des datagrammes dans IPv6 ont été considérablement allégé par rapport à la version précédente. Différents champs ont été supprimé afin de facilité le travail à effectuer par les routeurs et ainsi accélérer la transmission des paquets.

**Tunnels**: Des tunnels manuels ainsi que des tunnels automatiques peuvent être mis en place afin de permettre la cohabitation de réseaux IPv6 avec des réseaux IPv4. Ces tunnels encapsulent les paquets IPv6 dans des paquets IPv4 afin de traverser les parties IPv4 du réseau.