**Questions/réponses réseau avancé wilfart**

***Q1 : Pourvoir décrire l’utilité des différents champs présent dans le record SOA d’un fichier de zone***

Un fichier de zone contient des enregistrements de ressource pour la zone gérée par le serveur (maitre ou esclave) dans un format ASCII ce qui fait qu’il est facilement modifiable. Le 1er fichier dans n’importe quel fichier de zone est l’enregistrement SOA sous la forme suivante :

IN SOA <hôte source> <adresse mail de contact> <num de série> <tmp de rafraichissement> <intervalle d’essai> <tmp d’expiration> <TTL>

*Hôte source* : l’hôte sur lequel le fichier est maintenu.

*Adresse mail de contact* : mail internet pour la personne responsable du fichier de zone pour le domaine

*Num de série* : numéro de version du fichier de zone, incrémenté à chaque modification, sert pour déterminer si un transfert de zone est nécessaire entre le serveur maitre et le ou les serveurs de sauvegarde.

*Tmp de rafraichissement* : écart de temps en seconde qu’un serveur de nom secondaire doit attendre en 2 vérifications auprès du serveur principal pour vérifier si le fichier de zone a été modifier et qu’un transfert soit demandé.

*Intervalle d’essai* : intervalle de temps en seconde qu’un serveur secondaire doit attendre avant de réessayer un transfert de zone qui a raté.

*Tmp d’expiration* : intervalle de temps en seconde qu’un serveur secondaire doit continuer à essayer de changer de zone. Après expiration, les infos de vieille zone seront écartées.

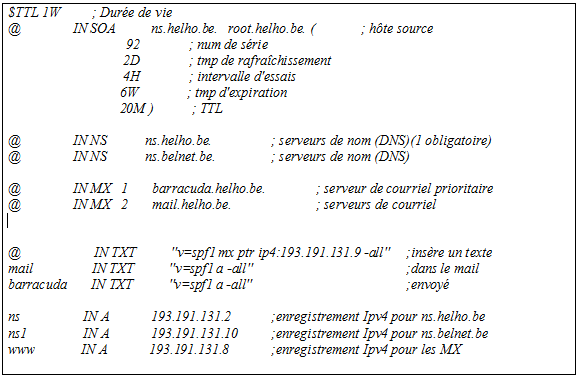
*TTL* : temps de vie, la valeur TTL reprise dans l’enregistrement SOA est utilisée comme valeur de cache négative.

***Q2 : Sur base d’une topologie réseau données, comprenant des serveurs avec leur nom, adresses IP et fonctions (ex serveur smtp), être à même de donner le contenu d’un fichier de zone avec l’ensemble des différents enregistrements (records)***

A la suite d’un enregistrement SOA obligatoire en en-tête de fichier, que ce soit pour les fichiers relatifs à la résolution directe ou inverse, nous retrouverons les éventuels enregistrements suivants :

* **A** : enregistrement d’adresse IPV4 qui fait correspondre un nom d'hôte à une adresse IPv4 de 32 bits. Le nom d’hôte peut être sous la forme d’un FQDN en étant terminé par le « . » associé au TLD.
* **AAAA** : enregistrement d’adresse IPV6 qui fait correspondre un nom d'hôte à une adresse IPv6 de 128 bits.
* **CNAME** : enregistrement de nom canonique qui permet de créer un alias vers un nom d’hôte. Ces alias sont utiles lorsque des noms de domaines multiples sont résolus vers une même adresse IP ;
* **MX**: enregistrement d’échange de courriel (Mail Exchange) qui définit les serveurs de courriel pour ce domaine. Cet enregistrement est associé à une préférence dans le cas où plusieurs serveurs sont définis pour le même domaine.
* **NS**: enregistrement de serveur de nom qui définit les serveurs DNS de ce domaine. Il est important d’y renseigner le serveur primaire est les serveurs de sauvegardes pour que les avis de notifications puissent être envoyés lorsque le numéro de série du fichier de zone change et que le serveur est redémarré.
* **TXT**: enregistrement de type texte et qui permet à un administrateur d'insérer un texte quelconque. Nous retrouvons notamment dans cet enregistrement les informations de type SPF (Sender Policy Framework) permettant de valider les serveurs de messagerie utilisés pour un domaine donné. Ce mécanisme est utilisé dans les solutions de lutte contre le spam.

*Exemple de fichier de zone :*



***Q3 : Décrire les mécanismes permettant de sécuriser un serveur DNS au niveau des types de requêtes qu’il est à même de pouvoir accepter : requêtes récursives, itératives, demandes de transfert de zone. Mettre en évidence les options de configuration avec les options correspondantes : allow-transfert, allow-recursion, allow-query.***

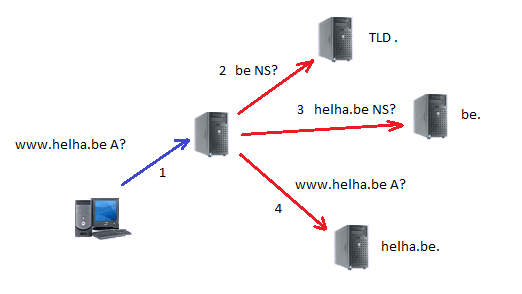
Il y a 2 types de demande vers un serveur DNS : récursives, itératives et inverse. Il faut garder à l’esprit qu’un serveur DNS peut être client d’un autre serveur DNS.

Le serveur DNS recevant la demande est responsable de la gestion du processus de résolution. Ce système doit envoyer la requête à un serveur différent pour chaque domaine.

La charge sur le serveur original provient d’une requête récursive : celle-ci place le moins possible de charges sur le client. Le serveur DNS recevant la requête peut retourner une réponse contenant l’adresse IP associée au nom ou un message d’erreur expliquant son incapacité à résoudre le nom. Le serveur est responsable de tous les appels vers les secteurs de plus haut niveau jusqu’à résolution du nom.

Une requête itérative nécessite un plus grand effort de la part du client. Lorsqu’un serveur reçoit une requête itérative, il ne retourne que l’information en sa possession sans effectuer de requête supplémentaire. S’il ne possède pas d’information, il peu fournir l’adresse de l’un des serveurs de noms racine.

Ces types de requêtes ne sont pas réservés aux clients. Quand un serveur DNS envoie une requête à un autre serveur DNS, il fonctionne alors comme un client et peut être configuré pour envoyer l’un des types de requêtes.



*Le résolveur envoie une requête récursive vers le serveur DNS paramétré dans sa pile TCPIP. S’il ne peut résoudre ce nom, le serveur envoie différentes requête itératives pour obtenir la réponse. Les serveurs DNS sont en général configurés pour accepter les requêtes itératives pour leurs zones mais uniquement les requêtes récursives pour leur client.*

Pour améliorer les performances lors des demandes de résolution, les résolveurs et les serveurs récursifs possèdent une cache interne. Dans un réseau local, on peut utiliser un serveur qui n’a autorité sur aucune zone mais dont le but est de servir de cache en relayant les demandes récursives vers un autre serveur externe.

Un serveur TLD doit fournir les informations au serveur de noms capable de répondre aux requêtes itératives. Un serveur récursif n’a pas la possibilité de déterminer de façon dynamique la liste des adresses de serveur TLD. Ces informations doivent être mise à jour lors de l’installation du serveur DNS.

Les serveurs racines sont au nombre de 13 et appartiennent au même domaine root-servers.net. Ces serveurs sont gérés par 12 organisations : 2 européennes, 1 japonaise et 9 américaines. Ces serveurs sont répartit à travers le monde.

***Q4 : Décrire les mécanismes permettant de mettre en place un serveur de backup et d’assurer les transferts de zones : création d’une zone secondaire, lien avec la zone primaire, utilité des enregistrements NS multiple dans la zone maitre, utilité des options de configuration de type notify, als-notify…***

Il y a différents types de zone :

Maitre : Le serveur a une copie maitre des données et autorité sur la zone.

Esclave : réplique d’une zone maitre, contient la liste appelé « master » et contient une ou plusieurs adresses IP que l’esclave peut contacter pour mettre à jour la copie de sa zone.

Stub : semblable à une zone esclave sauf qu’elle ne contient que les enregistrements NS de la zone maitre.

Forward : utilisée pour rediriger toutes les demandes vers un autre serveur. Si aucune close « forwarders » n’est présente dans la zone ou qu’une liste vide est renseignée, aucun transfert ne sera effectué pour les demandes vers un autre serveur.

Hint : comprend l’ensemble des serveurs de noms racines.

Nous retrouvons dans le répertoir /etc ou /etc/bind les fichiers suivants :

Named.conf : fichier principal comprenant des inclusions des deux autres fichiers, comprend aussi la définition de zones locales.

Named.conf.local : fichier de définition des différentes zones que l’on souhaite ajouter au serveur.

Named.conf.option : option de configuration globale du serveur et valable par défaut pour certaines d’entre elles pour toutes les zones définies.

Une des options correspond à l’emplacement où se trouveront les fichiers individuels pour chaque zone définie. Le nom du fichier est donné par l’administrateur du serveur.

Option {

Directory « /var/cache/bind » ;

}

Cette option précise que les fichiers de zone sont dans le répertoire mentionné. Dans le fichier named.conf, on retrouvera l’instruction suivante :

Include « /etc/bind/named.conf.local » ;

**Serveur maitre :**

Zone « helho.be » in {

Type master ;

File « helho.be.zone »

} ;

Le serveur a autorité sur la zone helho.be. On utilise le type master comme type de zone. Le nom du fichier est « helho.be.zone » et le fichier devra se trouver dans le répertoire défini dans l’option « directory » dans le fichier named.conf.options.

**Serveur esclave ou de sauvegarde :**

Zone »hehla.be » in {

Type slave ;

File « slave.helha.be.zone » ;

Masters{

193.192.13.4 ;

} ;

} ;

Serveurs de sauvegarde, ne possède pas d’autorité sur la zone, on utilise donc le type slave. Le nom du fichier est « slave.helha.be.zone » mais son contenu ne sera pas défini sur le serveur même puisqu’il en recevra une copie à partir du serveur maitre et dont l’adresse est fournie dans la rubrique « master ». Dans le fichier named.conf, on retrouve la zone suivante :

Zone « . » {

Type hint ;

File « etc/bind/db.root » ;

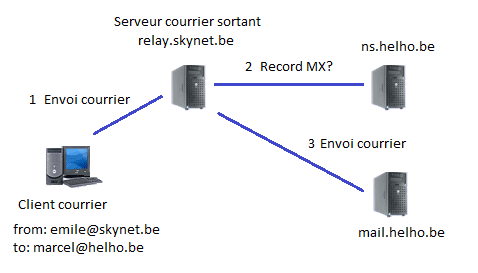
} ;

C’est le fichier qui contient la liste des serveurs de noms du top level domain.

Enregistrement NS est l’enregistrement de serveur de nom qui définit les serveurs DNS de ce domaine. Il est important d’y renseigner le serveur primaire et les serveurs de sauvegarde pour que les avis de notifications puissent être envoyés lorsque le numéro de série du fichier de zone change et que le serveur est redémarré.

***Q5 : Décrire le mécanisme par lequel un serveur DNS intervient dans le mécanisme de transfert de mails entre domaines différents. Mise en évidence de l’utilité du record MX.***

Enregistrement MX (mail exchange) définit les serveurs de courriel pour un domaine. Cet enregistrement est associé à une préférence dans le cas où plusieurs serveurs sont définis pour le même domaine.



1. Un courriel est envoyé part [emile@skynet.be](mailto:emile@skynet.be) à [marcel@helho.be](mailto:marcel@helho.be). Ce courriel arrive sur le serveur de courrier sortant configurer dans le client. Le serveur va vérifier si le destinataire est un domaine qu’il gère, si oui, le courrier est alors placé dans la boite de réception du destinataire.
2. Le destinataire appartient au domaine helho.be. Le serveur de courrier va donc interroger le serveur DNS qui a autorité sur la zone helho.be pour en connaitre l’enregistrement MX.
3. Ayant reçu l’information sur le serveur MX, le serveur de courrier sortant peut alors rentrer en contact avec le serveur de courrier du destinataire.
4. Le serveur mail.helho.be vérifie que le destinataire est bien géré par lui et le courrier est placé dans la boite de réception correspondante.

***Q6 : Expliquer par un exemple commenté la structure hiérarchisée de DNS***

Serveur DNS

Configuration TCP/IP

* TLD (top level domaine)
* Be (dns.be)
* Helha.be
* [www.helha.be](http://www.helha.be) (adresse IP)

Le DNS travaille en divisant l’espace de travail internet en un ensemble de domaines ou réseaux qui peuvent être à leur tour divisé en sous domaines.

Au sommet de cette arborescence se trouve le top level domain (TLD). Ce niveau est géré par Internic. Il comprend des domaines pour les organisations (org, com, …) et un domaine par pays (be, fr, …).

Un domaine peut contenir des hôtes (host) ou des sous domaines. Chaque domaine s’identifie dans l’arborescence par rapport à son parent en étant séparé par un point. Par exemple le site de la Haute Ecole est un sous domaine de « be », ce qui donne « helha.be ». Un sous domaine peut également contenir des hôtes ou un sous domaine. On pourra donc trouvé des ordinateurs tels que [www.helha.be](http://www.helha.be) ou ac.helha.be, sous domaine comprenant l’ordinateur extranet.ac.helha.be.

***Q7 : Décrire le rôle de chacune des clés DNSKEY de type ZSK et KSK. Expliquer les différents mécanismes permettant d’effectuer une rotation de ces clés.***

Afin d’assurer l’authenticité et l’intégrité des réponses, DNSSEC est basé sur un modèle de cryptographie à clé publique.

L’enregistrement DNSKEY est utilisé pour transmettre une clé publique entre le résolveur et le serveur de nom. Cette clé publique est associée à la clé privée que va utiliser le serveur d’autorité pour signer les hash des enregistrements RRSET. Les RRSET (ressources record set) sont un regroupement d’enregistrement en fonction du nom et du type de ces enregistrements.

Une clé publique est associée à une zone non à un serveur. Le résolveur utilisera cette clé publique écrite dans l’enregistrement DNSKEY pour vérifier la signature du serveur d’autorité.

On retrouve 2 types de clé : KSK et ZSK. La clé KSK signera seulement les DNSKEY RRSET tandis que la clé ZSK signera les autres RRSET dans la zone.

**ZSK (zone signing key)** : correspond à la clé privée utilisée pour signer une zone, c’est-à-dire les enregistrements demandés par les serveurs récursifs. Une clé n’expire jamais, seules les signatures RRSIG ont une période de validité. Mais pour des questions de sécurité, il est conseillé de renouveler la clé tous les mois. Les serveurs DNSSEC intègre un mécanisme de rotation de clé de sorte que chaque mois, la clé est changée.

Une 1ère technique consiste à générer une nouvelle clé et à signer la zone avec cette clé en laissant l’ancienne clé présente dans le fichier de zone le temps de l’expiration des données dans la cache mais cette technique est pénalisante du fait d’une double signature pour chaque enregistrement.

Une 2ème technique consiste à pré publier la nouvelle clé. On retrouve une clé active et une clé passive dans le sens où une seule des 2 est utilisée pour signer les enregistrements de la zone. On peut ajouter la nouvelle clé juste avant la valeur TTL de la clé ou les 2 clés en même temps au début du de la période du mois. En fin de période, la zone est signée avec la clé passive, elle-même signée avec la clé KSK.

**KSK (key signing key)** : clé privée destinée à signer une clé de zone (ZSK), n’est utilisée que pour signer un enregistrement DNSKEY. Les signatures des clés ZSK ont, en général, une durée de validité de 1 an. Le remplacement de cette clé est plus complexe du fait que l’on doit également intervenir dans la zone parent.

1. On ne peut controler quand le parent changera ou publiera le nouvel enregistrement DS.
2. On doit maintenir l’ancienne KSK le temps que le TTL de l’enregistrement DS arrive à expiration pour éviter des problèmes de caches.

Il faut aussi tenir compte du fait que le parent puisse accepter ou pas plusieurs enregistrements DS pour une même zone enfant.

Pour les parents n’acceptant qu’un seul enregistrement DS :

1. Introduire une nouvelle KSK dans la zone enfant.
2. Signer les ZSK avec la nouvelle KSK en maintenant les anciennes signatures.
3. Attendre que les caches aient la nouvelle version de la zone.
4. Publier le nouvel enregistrement DS chez le parent.
5. Attendre que le parent publie ce nouvel enregistrement.
6. Attendre le plus long temps entre la TTL de l’ancien et du nouveau DS.
7. Déprécier d’ancienne KSK et supprimer les anciennes signatures ZSK.

Pour les parents acceptant plusieurs enregistrements DS :

1. Introduire une nouvelle KSK dans la zone enfant
2. Signer les ZSK avec la nouvelle KSK en maintenant les anciennes signatures
3. Publier le nouvel enregistrement DS en maintenant l’ancien
4. Attendre que le parent publie ce nouvel enregistrement DS
5. Attendre le plus long temps entre la TTL de l’ancien et du nouveau DS
6. Déprécier l’ancien KSK et supprimer les anciennes signatures ZSK
7. Supprimer l’ancien enregistrement DS du parent

***Q8 : Etre à même de décrire le mécanisme d’obtention des enregistrements RRSIG qui sont les signatures des codes de hashage des RRSET. Expliquer ce que sont les RRSET. Décrivez la structure d’un record RRSIG et évoquer la durée de vie d’un tel record.***

L’enregistrement RRSIG (ressource record signature) contient la signature de l’enregistrement envoyé par le serveur d’autorité. Il s’agit de la signature obtenue en signant le hash du RRSET avec la clé privée du serveur d’autorité, c’est-à-dire à celle contenue dans l’enregistrement DNSKEY. C’est cette signature que le résolveur vérifiera par la suite. Il existera un enregistrement RRSIG pour chaque enregistrement de la zone dans le fichier de zone signé.

Les RRSET (ressources record set) sont un regroupement d’enregistrement en fonction du nom et du type de ces enregistrements.

86400 A 193.168.131.10

86400 RRSIG A 8 2 86400 20140710102050(

20140610092050 6228 esit.be.

)

A : le champ « type covered » correspond au type d’enregistrement qui a été signé. Ici, il s’agit d’un RRSET couvrant le type A.

8 : algorithme utilisé.

2 : le champ « label » indique le niveau de hiérarchie dans le nom contenu dans le RRSET signé. Ici, esit.be contient 2 champs (be et esit), donc la valeur 2.

86400 : le TTL d’origine

20140710102050 : date d’expiration de la signature

20140610092050 : date de la signature

6228 : identifiant de la clé utilisée

Esit.be : nom de la zone détentrice de la clé utilisée pour la signature, utile pour la vérification de la signature.

***Q9 : Comment assurer l’authenticité et l’intégrité des réponses lorsque le serveur DNS n’a aucune réponse à fournir (ex d’un nom d’hôte qui n’existe pas). Expliquer comment fonctionnent les mécanismes NSEC et NSEC3, avantage et désavantage de l’un par rapport à l’autre.***

Assuré l’authenticité et l’intégrité des réponses est possible temps qu’une réponse peut être fournie. Mais que se passe-t-il lorsque la requête envoyée vers le serveur DNS ne correspond à aucune réponse ? Voilà la raison d’exister des enregistrements NSEC et NSEC3.

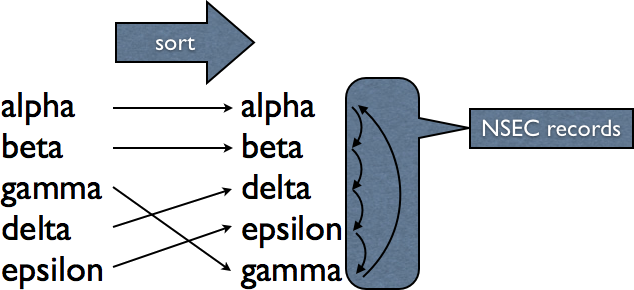
NSEC : un enregistrement NSEC contient les informations nécessaires à l’identification des enregistrements existants pour un nom donné, ainsi que le prochain nom existant dans la zone. Ces 2 informations suffisent pour prouver qu’un enregistrement ou qu’un domaine n’existe pas.

Exemple :

Ns1.esit.be. 3600 NSEC [www.esit.be](http://www.esit.be). A AAAA RRSIG NSEC

[www.esit.be](http://www.esit.be). Correspond au nom de domaine suivant.

A AAAA RRSIG NSEC correspond à une liste de RSET correspondant au domaine suivant. Le NSEC du dernier nom de domaine pointe vers le 1er nom de domaine de la liste.



Le problème de ce type d’enregistrement est la possibilité sans que le transfert ne soit validé pour une zone, de récupérer l’ensemble des différent noms de domaine. C’est pour éviter ce genre de problème que 2 autres types d’enregistrement sont apparus : NSEC3PARAM et NSEC3.

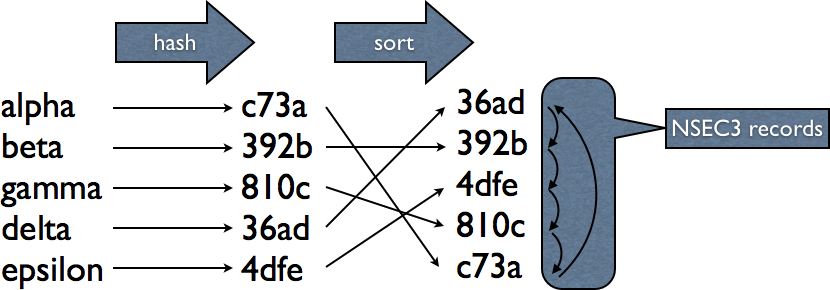
NSEC3 : permet la gestion des réponses négatives tout en luttant contre des attaques de type « énumération de zone ». L’enregistrement NSEC3 utilise une fonction de hachage appliquée aux noms renvoyés. Ainsi, sera envoyé le hash du prochain nom de la zone, ne permettant pas de reconstruire la zone entière.

NSEC3PARAM : utilisé par les serveurs DNS pour calculer et déterminer quels sont les enregistrements NSEC3 à inclure pour les réponses de non existence.

Il comprendra les enregistrements suivants :

* Algorithme de hashahe
* Flag « opt-out » indiquant si les délégations sont signées ou pas
* Itérations : combien de fois l’algo de hashage doit-il être appliqué
* Salt : valeur de salt pour le calcul de hashage

La hashage fonctionne comme ceci : lorsque le résolveur voudra vérifier la réponse négative, il va générer le hash du nom demandé, exemple : h2 et vérifier que h2 est compris entre h1 et h3 (indiqué dans l’enregistrement NSEC3). Il est conseillé de remplacer régulièrement le salt.



***Q10 : Expliquer comment l’enregistrement DS assure la chaine de confiance dans les liaisons entre les serveurs DNS parents et enfants. Donner un exemple graphique dans une hiérarchie complète du DNNSEC. (ex pour esit.be avec chaque niveau de hiérarchie)***

L’enregistrement DS (delegation signer) assure la chaine de confiance dans les liaisons entre les serveurs DNS parents et enfants. Il sera placé dans la zone parent et sera lié à l’enregistrement DNSKEY de la zone enfant. Il contiendra également un hash de la clé KSK de la zone enfant, stockée avec l’enregistrement NS indiquant la délégation vers la zone enfant. Cet enregistrement DS est signé avec le reste des données de la zone parente.

Cet enregistrement est demandé par le résolveur au serveur autoritaire dès qu’il a besoin de vérifier une signature reçue par la zone fille. La zone mère lui indique qu’il peut faire confiance à ce serveur en lui renvoyant un hash de la clé signé par elle-même. Le résolveur sait ainsi qu’il peut lui faire confiance.

A chaque maillon de la chaine, on retrouve le même schéma : un enregistrement DS permet d’avoir la confiance en une KSK, la KSK permet d’avoir confiance en l’ensemble des clés de la zone et les ZSK permettent alors d’avoir confiance dans les autres enregistrements.

Exemple graphique DNSSEK avec esit.be :

ROOT « . »

fr

…

be

…

helha

esit

esit.be

**IPv4**

**Intro**

Adresses en 32 bits (de 0 à 255 décimales => exemple : 192.168.103.14)

* Permet 2^32 adresses (4 294 967 296) -> 4 milliards. Hôtes qui ont besoin d’une IP double chaque année -> nombre d’adresses dispo diminue beaucoup.

En 2011, 5 derniers blocs d’IP distribués par IANA, les IP restantes dispo sont gérés par les RIR. Maintient de l’IPv4 en libérant les adresses non utilisées des : adresses en format CCIDR, affection d’adresses réservées, le NAT et la demande de clients ayant reçu des IP en /8.

* Quand même obligé de trouver un nouveau moyen pour se libérer des limites imposés par IPv4 => routage plus efficace.

Architecture IPv6 définie dans RDC 2373 (IPv6 Addresing Architecture) = Adresses en 128 bits (16 octets). Pas facile de représenter une IPv6

**Représentation de l’IPv6**

* Pour éviter les décimaux énormes, IPv6 est en hecadécimal avec des « : » entre les blocs. 8 blocs de 2 octers :



* Chaque bloc est séparé des autres, les 0 non significatifs peuvent être éliminés :



* Si plusieurs blocs consécutifs sont composés de 0, ils peuvent être abrégés :

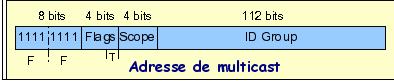




* **Utilisation du « :: » qu’une seule fois dans une adresse IPv6**

**IPv6 et les types d’adresses**

* Unicast : Identification d’une seule interface. Si un paquet est envoyé à une adresse Unicast, elle est reçue par l’interface dépositaire de l’adresse.
* Multicast : Groupe de plusieurs interfaces de plusieurs nœuds. Paquet envoyé vers une adresse multicast est reçue par tous les membres du groupe. Adresse multicaast comment par : « ff00 ::/8 »



Bit T = permanence de l’adresse, T=0 (adresse permanente), T=1 (adresse temporaire)

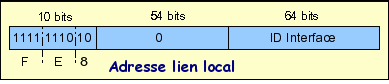
Scope = portée de diffusion , 0(réservé), 1(nœud), 2(lien), 3(sous-réseau),5(site),8(organisation),E(global),F(réservé)

* Certaines adresses sont prédéfinies  -> considérées comme permanentes :
  + Ff02 ::1 -> toutes les machines du lien
  + Ff02 ::2 -> tous les routeurs du lien
  + Ff05 ::2 -> tous les routeurs du site
  + Ff02 ::1 :ffxx :xxx -> tous les noeufs se terminent par xx :xxx dans leur IP
* Anycast : Identifie un groupe d’interfaces de plusieurs nœuds (comme Multicast). Quand un paquet est envoyée à une adresse Anycast -> reçue par un des membres du groupe anycast (souvent le plus proche en terme de routage).

En IPv6, type d’adresse broadcast n’existe plus.

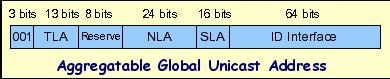
**ADRESSES PARTICULIÈRES EN IPv6 :**

* + Adresse de bouclage :
    - ::1
  + Adresse inconnue composée de 0
    - ::
  + Adresse de lien local
    - Donnée aux cartes d’interface dynamiquement. Valide uniquement dans un espace local qui ne traversent pas de routeur. Préfixe = « fe80 ::/10 » Structure :



Sous windows, ipconfig apparaît les adresses de link local, une par interface. IP indépendante de toute config. Sert à adhérer aux divers groupes multicasts sur le lien, remplacent le broadcast IPv4 et ARP.

* + Adresse de site local :
    - Adresses privées IPv4. Routables mais ne peut pas être utilisé sur Internet. Préfixe = fec0 ::/10. Après les 10 premiers bits, les 54 suivants définissent l’adresse de network. La RFC qui a introduit est passé en deprecated -> chaque équipement aura son adresse publique -> adresse IPv6 locales -> POUBELLE
    - En 2005, une RFC introduit les adresses de type Unique Local Address. Préfixe = fc00 ::/7. 8ème bit L est assigné à la valeur 1. Ensuite ID global sur 40 bits, ID de sous-réseau sur 16 bits + interface sur 64 bits. => Possibilité de pouvoir utiliser un ID global qui permet l’interconnexion de sites sans changer l’ID des réseaux.
  + Adresse utilisable sur Internet
    - Structure :

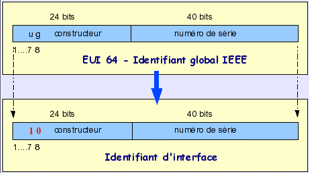


* + - Préfixe sur 3 bits, valeur correspond à son adresse unicast publique.
    - Top Level Agegator -> différents blocs libérés par IANA -> premier porte sur numéro 1. 16 bits correspondants seront donc : 0010000000000001 et 2001 en hexa. 32 bits suivants dont 8 de poids forts se trouvent répartis comme :
      * 13 bits comme SUB-TLA, 19 bits comme NLA.
    - **CETTE RFC EST MAINTENANT ARCHIVÉE**
    - IPv6 distribuées dans le premier bloc 2001 :: aux RIR sont des blocs /23 -> RIR distribueront les IP aux LIR en /32 => Belnet reçoit un /32.
* Belnet dispose de 16 bits pour répartir les adresses aux users. = > Adresses /48 distribuées aux gros users.
  + - * EXEMPLE : L’HELHa Tournai a reçu 2001:06a8:3480::/48. Sur une adresse de 128 bits, le site final gère 80 bits dont 16 pour identifier un réseau et 64 bits restant l’host sur le réseau. La HELHa pourra donc connecter 2^16 réseaux et chaque réseau 2^64 hosts. Les IP sont attribués via un serveur DHCP.

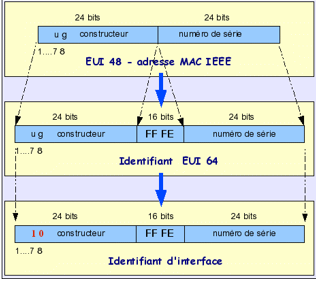
**Création des adresses de lien**

Dans ce chapitre, on analyse la façon dont les 64 bits de l’ID interface seront choisis (adresses de lien local sont attribués de façon automatique au démarrage de toute interface).

1ère solution = se baser sur une adresse physique dans un format EUI-64 de l’IEEE. On insère donc l’adresse physique dans les 64 bits de l’IPv6.



2ème solution = basé sur les adresses physiques sur 48 bits comme les adresses MAC des cartes réseaux => Format EUI-48 de l’IEEE. Pour combler les 64 bits, on fait cette opération :



**Les adresses anonymes**

Adresses fabriquées à-p-d l’EUI-64 n’était plus anonyme. Sous XP, (avec un routeur qui annonce un réseau), l’adresses normale est auto configurée mais une autre adresse l’est aussi. Cette adresse est dite « anonyme » dont la partie hôte est choisie au hasard (remplace EUI-64). Fonction désactivable.

Chaque préfixe, chaque interface reçoit une IPv6 supplémentaire :

<préfixe><EUI-64>

Sous XP/2003, on ajoute une IPv6 anonyme en plus :

<préfixe><64 bits au hasard>

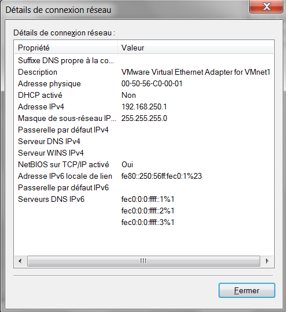
Pour les connexions sortantes, on utilise l’IP anonyme.

Pour les connexions entrantes, on peut joindre sur les 2 IP.

Quel que soit le réseau de la machine, l’EUI-64 reste toujours le même => Quelque soit le réseau et la période de connexion, on peut identifier la machine elle-même.

L’adresse anonyme de Microsoft est censé palier à ce « problème », l’anonymat en matière de réseau est largement théorique.

EXEMPLE :



Adresse MAC de la carte réseau : 00-50-56-C0-00-01

Adresse de lien : 00-50-56 FFFE C0-00-01

On a donc 0250:56FF:FEC0:0001 (2 pour bit ‘u’ positionné à 1)

Adresse de lien local sera donc : fe80:250:56ff:fec0:1%23 (%23 représente l’interface associée a l’adresse)

**Solicited Node Address, Neighbor Discovery, Router Advertissement**

Pour chacune des IPv6, chaque interface adhère à un groupe multicast calculée en fonction de l’interface. Adresse = solicited node address => adresse multicast de lien local se présente comme ceci : **FF02:0:0:0:0:1:FFXX:XXXX**



Les X dans l’adresse représente la partie variable de l’adresse. Contient les 24 derniers bits de l’IP d’origine. (3 octets ou 6 nibbles de droite). Donc, si l’adresse à-p-d EUI-64 est utilisé, toutes les IPv6 de l’interface se lient donc à une seule solicited node address (car EUI-64 ne change pas même si la partie réseau change).

En connaissant l’IPv6, on peut directement reconnaître la solicited node address. Pour découvrir l’adresse MAC relative à une IPv6 connue, il faut envoyer un message ND (Neighbor Discovery) vers cette adresse en utilisant l’adresse de lien local.

**Intérêt de l’utilisation de la solicited node address :**

La méthode fait appel à plusieurs groupes multicast auxquelles peu de nodes adhèrent en même temps. Multicast sur un segment Ethernet de base = broadcast niveau 2. **MAIS** niveau équipement niveau 2 et 3 qui gère le multicast (chaque adhésion multicast IPv6 crée une adhésion niveau 3, par ND, mais aussi au niveau 2 par broadcast Ethernet), utilisation du ND et donc du multicast permet de réduire **ÉNORMÉMENT** le nombre de broadcasts sur le réseau et de limiter le trafic de résolution des adresses MAC, convertit le broadcast en quasi-unicast.

Au démarrage, une interface a une adresse de lien local (avec EUI-64) et se lie à la solicited node address correspondante. Si il n’y a pas de routeur sur le lien, il n’y a pas d’adresse attribuée automatiquement. Le node cherche la présence d’un routeur avec un multicast approprié : Le Router Solicitation.

**Si un routeur est présent et configuré**, il répondra au message par un RA (router advertisement). Puisque le message a été expressément demandé, il s’agira d’un Solicited Router Advertisement. Il contient les préfixes disponisbles sur le réseau au format CIDR ainsi que sa propre adresse. Avec l’adresse du réseau, la machine peut accoler le réseau et son EUI-64 pour former une IPv6 routable et unique. Le routage est possible car la machine utilisera l’adresse du routeur fournie dans le RA. Si il y’a plusieurs routeurs, plusieurs IP seront fabriquées et plusieurs routes par défaut seront rajoutées.

**L’adresse est fabriquée comme ceci :** <64 bits du réseau, annoncé par le routeur><EUI-64 : 64 bits hosts, définis à-p-d de la MAC>

Tout routeur correctement configuré émet régulièrement sur le réseau des RA pour annoncer les informations. Puisque personne ne les réclame, il s’agit d’Unsolicited Router Advertisement.

Il est possible de configurer des IP statiques et indépendantes de l’EUI-64, on peut aussi désactiver l’auto-config des adresses et de la route par défaut. **PRATIQUEMENT INDISPENSABLE POUR LES ROUTEURS.**

Quand un host détecte un routeur, une route par défaut est automatiquement ajoutée. Cette adresse est une adresse de lien (pas une adresse globale). Choix justifié car une adresse de lien ne change pas. Une adresse globale peut être modifiée par l’administrateur du réseau.

Exemple de la création de la SNA :

2 hosts sur un segment Ethernet, A et B. Host A a une link local fe80:100:100:100::A/64 . Tandis que Host B a une link local fe80:100:100:100::B/64. A veut rentrer en contact avec B.

1. A doit connaitre la MAC de B. En IPv4, la récupération est faite avec le protocole ARP (broadcast). Host A peut envoyer un paquet vers l’adresse de multicast à laquelle Host B est joint. Il suffit pour l’Host A de prendre le préfixe **ff02:0:0:0:0:1:ff00::/104** et d’ajouter les 24 bits de poids faible de l’adresse de lien d’Host B = **ff02:0:0:0:0:1:ff00:000B.**
2. Host B est déjà configuré au niveau de son solicited node address et a rejoint le groupe **ff02:0:0:0:0:1:ff00:000B** en utilisant MLD (substitut IPv6 d’IGMP). Host B, comme membre du groupe, recevra un NS (Neighbor Solicitation) de Host A.
3. Host B répond avec sa MAC dans la partie payload du Neighbor Advertisement Message.

**Attribution des IP globales**

Le mécanisme d’auto configuration des IP en mode stateless prend aussi en charge les adresses globales grâce à des messages de type Router Solicitation et Router Advertisement avec les routeur voisins.

En stateless, on peut configurer les IP de façon automatique au travers de la découverte des préfixes. Pour chaque préfixe renseigné sur le routeur pour une interface donnée, on retrouve deux flags :

* + - Flag Autonomous : Instruit l’host qui reçoit la donnée en générant une adresse automatique à-p-d du préfixe. Si le flag = false, pas d’adresse générée sur base du préfixe.
    - Flag OnLink : Instruit l’host qui reçoit la donnée que le préfixe est OnLink et qu’il doit être utilisé dans la config automatique, e.g. : dans la table de routage de l’host.

Dans cette solution, aucune config n’est envisagée pour les mécanismes de résolution de nom. On doit donc soit configurer les informations manuellement soit utiliser un serveur DHCPv6 qui se limite à attribuer des IPv6 stateless et d’attribuer des options comme : serveur DNS, serveur de temps, etc…

Dans la config des messages Router Advetisement sur un routeur, on peut retrouver deux flags Managed Configuration et Other Parameters Configuration -> permet de définir la façon dont les IP et paramètres sont obtenus automatiquement par un serveur DHCPv6. Les deux flags sont **M flag** et **O flag**.

Effet des flag envoyées d’un routeur vers un host :

* Flag M et O à 0 : Correspond à un réseau sans serveur DHCPv6. Paramètres devront être définies manuellement.
* Flag M et O à 1 : DHCPv6 utilisé pour fournir les paramètres et l’IPv6 -> DHCPv6 statefull.
* Flag M 0 et O 1 : DHCPv6 utilisé pour assigner les paramètres, IPv6 attribuée automatiquement par les mécanismes de découverte des routeurs voisins -> DHCPv6 stateless
* Flag M 1 et O 0 : Adresse attribuée par DHCPv6, paramètres non définies.
* On peut donc se retrouver avec deux IPv6 pour le même préfixe. Si on définit sur le routeur un préfixe avec un flag autonomous avec un flag M à 1, deux adresses sont générées. L’une fournit par le DHCPv6 et l’autre pour le préfixe fourni.

**Tables de routages sur les hosts**

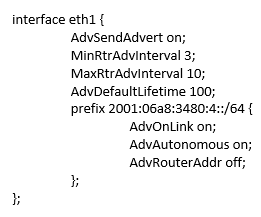
En DHCPv4, il est possible de définir des routes et une passerelle par défaut comme options à fournir par le client. Dans les premières normes DHCPv6, les options ne sont pas présentes. Les options ont étés ajoutées mais pas implémentées dans tous les serveurs et/ou clients DHCP.

Passerelle par défaut est ajoutée automatiquement lors de la découverte des routeurs. Pour les tables de routage, elles sont générées sur base des préfixes définis sur les routeurs. Il suffit de mettre le flag autonomous à 0 et le flag OnLink à 1 pour chacun des préfixes.

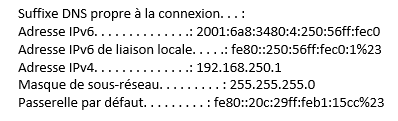
S’il y’a plusieurs routeurs sur le même segment, on peut définir un ordre de préférence pour les différentes routes fournies si des routes peuvent être redondantes. L’Host peut donc voir un routeur comme routeur primaire ou routeur secondaire. On donne donc trois valeurs : High, Medium et Low.

Pour la passerelle par défaut, il est possible, s’il y’a plusieurs routeurs, de supprimer la passerelle de l’host pour un routeur déterminé. Il suffit de placer le default lifetime à 0 pour le routeur concerné. Un host qui recevra la propriété écartera l’adresse de lien du routeur si le lifetime de celui-ci = 0.

Pour aider dans les test de config auto (cas où un routeur est présent sur le lien local), on configurera un Debian avec le package radvd. Voilà la config du routeur (/etc/radvd.conf)



Quand le service sera démarré, on obtiendra pour cette interface, les informations suivantes :

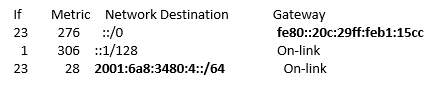


L’adresse de liaison locale existe toujours. On retrouve, en plus des autres, une adresse générée automatiquement sur base du préfixe renseigné par le routeur. La génération de l’adresse est possible grâce au flag ‘AdvAutonomous’ mis à 1. On rajoute au préfixe l’EUI-64 (identique à celui qui est utilisé pour l’adresse de lien)

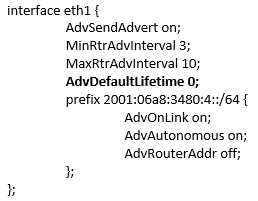
Une adresse de passerelle est ajoutée = adresse de liaison locale du routeur

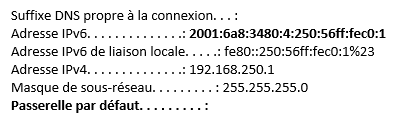


Si on analyse la table de routage, on peut voir :



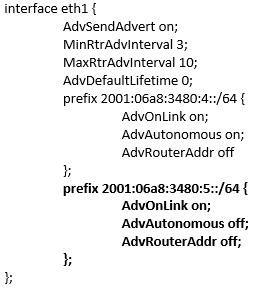
Route par défaut qui correspond à l’adresse de liaison du routeur. On retrouve aussi la route associée au préfixe fourni par le routeur. S’il y’a plusieurs routeurs présents et qu’on souhaite supprimer une route par défaut pour un routeur, on peut supprimer cette route en définissant le temps de vie de l’information fournie par le routeur à 0.





L’adresse de scope global existe toujours mais la passerelle par défaut est maintenant absente. Si l’on veut ajouté des routes supplémentaires associées à d’autres préfixes (sans générer d’adresses automatique basé sur ce dernier), on peut jouer sur le flag AdvAutonomous que l’on peut placer à 0

Exemple :



Si on regarde les adresses automatiques, on remarque qu’aucune adresse automatique n’a été générée pour 2001:06a8:3480:5::/64 mais on peut retrouver cette adresse dans les tables de routage.

