M3102-- TD 1

La résolution de nom et le service DNS

1. Rappel : machine, réseau, DNS

Dans un même sous-réseau informatique, plusieurs machines sont connectées avec un même support physique. Les machines utilisent leurs adresses MAC pour communiquer l'une avec l'autre. En dehors du réseau, toutefois, les machines s'identifient l'une à l'autre en utilisant leurs adresses IP. Les adresses IP sont utilisés parfois également dans un même réseau également.

L'Internet connecte des réseaux informatiques plus petits d'une façon structurée. Pour que les utilisateurs puissent plus facilement se connecter à une autre machine, au lieu d'utiliser son adresse IP, on utilise plutôt un nom associé. Le service DNS s'occupe de l'association dans les deux sens :

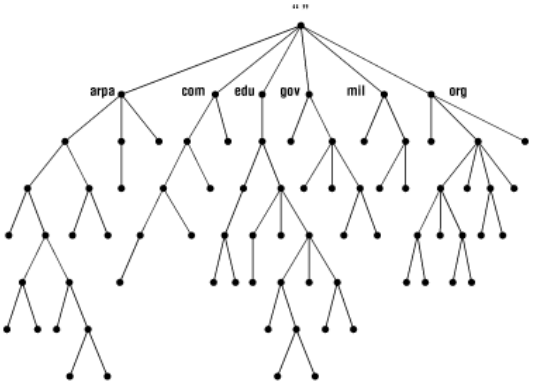
1. **La résolution directe** : à partir d'un nom, trouver une adresse IP
2. **La résolution inverse** : à partir d'une adresse IP, trouver le nom de la machine

Dans un petit réseau la résolution directe et inverse peut se réaliser en modifiant à la main le fichier ***hosts***. A grande échelle, toutefois, l'utilisation du fichier ***hosts*** n'est pas pratique. Sur les grands réseaux on utilise plutôt le système ***Domain Name System (DNS)***.

Il est essentiel que toutes les machines utilisent la même association entre les noms et les adresses IP. De plus cette association doit être mise à jour régulièrement. C'est pourquoi une organisation mondiale, l'IANA, est chargée de gérer les noms de domaine. Elle délègue une partie de ces responsabilités à quelques organisations comme par exemple NIC France, RIPE, etc.

2. Les noms de domaine et leur hiérarchie

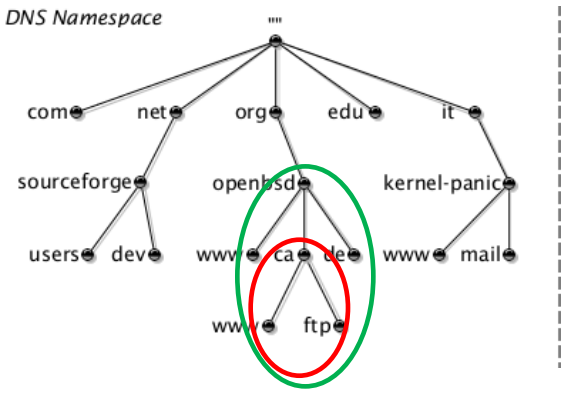
Le système DNS utilise une base de données distribuée, qui est structurée en un arbre inverse qui s'appelle un «espace de nommage » (*namespace* en anglais). Cet espace de nommage DNS est visible sur le schéma à côté (***Source : docstore.mik.ua***).



La structure d'arbre indique une hiérarchie. Chaque nœud de l'arbre est identifié par un nom d'au max 63 caractères. Le nœud racine n'a pas de nom, sinon il est identifié seulement par un point (« . »).

La hiérarchie de nommage inclut au plus 127 niveaux (un nom a en total 253 caractères ASCII).

Chaque domaine est structuré à plusieurs niveaux. Le nom d'un domaine est la collection des noms de plusieurs nœuds, séparés par des points (par exemple ***mail.unilim.fr***). L'ordre de noms est inversée par rapport à la hiérarchie de l'arbre : on met en premier les noms aux niveaux inférieurs et seulement après les noms aux niveaux supérieurs. Un domaine forme un sous-arbre, dont le nom commence au nom du nœud le plus haut dans la hiérarchie.



Dans l'exemple ci-dessus (***Source : kernel-panic.it***) on se concentre sur un sous-arbre sur 6 niveaux (y compris la racine). On peut distinguer par exemple un sous-domaine ***ca.openbsd.org.*** et un sous-domaine plus grand au nom de ***openbsd.org.***

Disons qu'on se trouve sur une machine ***ftp.ca.openbsd.org*** et on tape alors *ping www*. La machine cherchera dans le sous-domaine indiqué et trouvera ***www.ca.openbsd.org***.

Pour avoir un espace de nommage utilisable, il faut que chaque domaine ait un nom unique. C'est pourquoi deux nœuds d'un même sous-arbre ne peuvent pas avoir le même nom. Ceci est toutefois possible dans deux domaines différents.

L'organisation d'un nom de domaine

Les noms de domaines se structurent en plusieurs niveaux. Juste en dessous du niveau de la racine se trouvent les niveaux les plus hauts (Top Level Domains -- TLD). En plus de son TLD, un nom de domaine aura également un nom de deuxième niveau et pourra avoir des noms d'un niveau plus bas encore (troisième, quatrième…).

Les TLDs peuvent être de plusieurs types :

- Le domaine spécial ***.arpa*** utilisé pour résoudre des problèmes d'infrastructure;
- Les domaines représentant des pays (niveau national) -- country-code TLD ou ccTLD -- comme par exemple ***.fr***, ***.de***, ***.ca***, ***.be***, etc.
- Des domaines génériques de premier niveau -- generic TLD ou gTLD -- comme par exemple ***.com***, ***.edu*** ou ***.mil***.

- Des domaines génériques restreints dont les sous-domaines font l'objet de quelques règles, notamment ***.biz***, ***.name*** et .***pro***.
- Des domaines sponsorisés de premier niveau, sTLD, dont les noms sont sponsorisés par de certaines organisations : par exemple ***.post*** est un domaine sponsorisé par Universal Postal Union.
- Le domaine de premier niveau de test, tTLD, ***.test***. Ni le niveau de test ni ses sous-domaines ne peuvent être enregistrés, donc ils peuvent être utilisés librement dans chaque autre domaine.

En général un domaine a plus qu'un seul niveau. Le niveau inférieur au TLD est le deuxième niveau (sous-domaine). Plus bas on trouve un domaine de troisième niveau (machine), et on peut continuer ainsi.

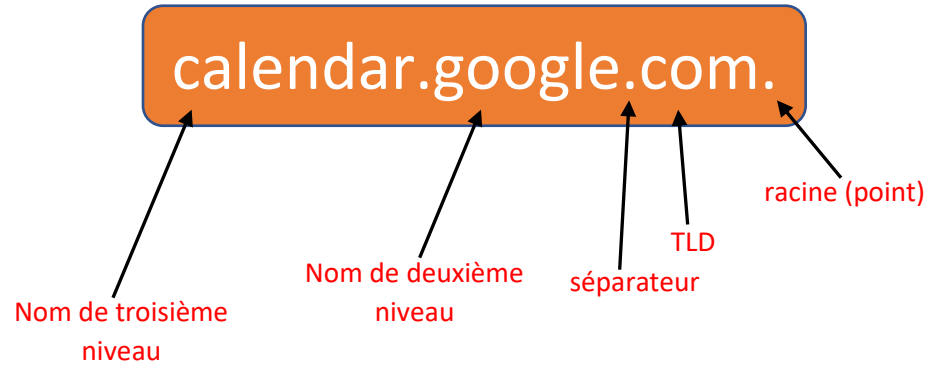
Les machines d'un domaine ne doivent pas forcément être réparties dans une même zone géographique : deux machines qui font partie de deux réseaux différents et sont localisées dans deux pays différents peuvent, néanmoins, faire partie d'un même domaine.

Nom absolu, nom relatif

Le nom absolu d'un domaine est un nom complet (Fully Qualified Domain Name, FQDN), qui finit toujours par un point. Même si à l'écriture on omet parfois le point final, ceci n'est pas possible lorsqu'on fait la configuration des serveurs DNS.

Une adresse IP s'écrit de gauche à droite. La partie la plus significative est celle de gauche, qui indique la classe de réseau à laquelle elle appartient.

Au contraire, le nom d'un domaine s'écrit de droite à gauche, avec la racine tout à gauche. Prenons l'exemple du domaine calendar.google.com. :

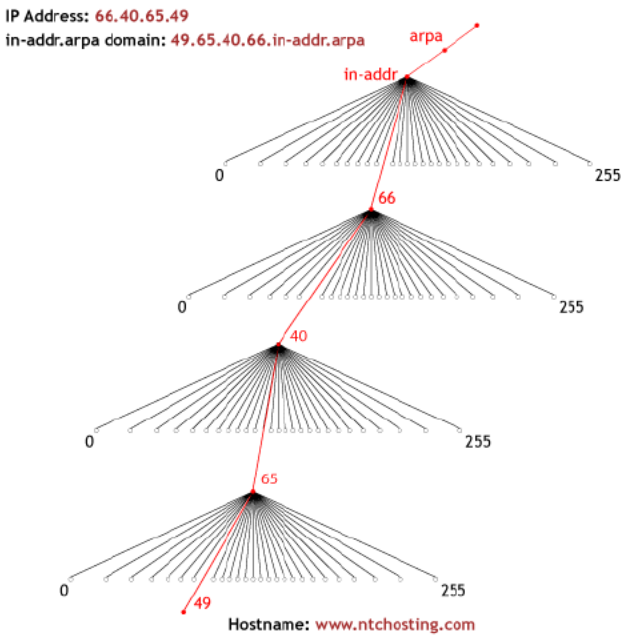


La résolution DNS

La résolution directe DNS consiste à trouver l'adresse IP d'une machine à partir de son nom. La résolution inverse consiste à trouver le nom d'une machine à partir de son adresse IP. Pour réussir cela, un domaine spécial, ***in-addr.arpa*** a été mis en place. Ce domaine stocke les adresses des machines dans un arbre, avec

la partie machine en bas. Les octets de l'adresse forment des nœuds qui sont parcourus en ordre inverse pour la résolution inverse.

Nous voyons dans la figure ci-dessous un exemple de résolution inverse. Ici nous cherchons l'adresse IP 64.40.65.49. Les octets sont parcourus d'en bas de l'arbre vers le haut, et chaque nœud a comme nom un des octets de l'adresse. Dans ce cas-ci, le FQDN associé à l'adresse 64.40.65.49 est 49.65.40.64.in-addr.arpa.



Exercice I

1. Donnez quelques avantages pour l'utilisation d'un fichier *hosts* pour faire la résolution directe et inverse. Puis, donnez quelques avantages dans l'utilisation d'un serveur DNS.

2. Dans quel cas peut-il être utile de faire une résolution inverse de nom ?

3. Quel est le FQDN de la machine ***mail*** du domaine ***unilim.fr*** ?

4. Quel est le FQDN de la machine 164.81.20.22 lorsqu'on fait la résolution inverse ?

5. Quelle est la cible de la commande ***ping www*** sur la machine ***pc1.mondomaine.fr*** ?

6. Qu'est-ce qui se passe si on essaie de trouver la machine **www.** (avec le point final) ?

2. L'application DNS

Les machines qui gèrent les informations pertinentes aux espaces de noms s'appellent serveurs de nom (***name server***). Un serveur de nom contient des informations propres à une partie de l'espace complet de noms -- on appelle ce sous-espace la ***zone d'administration*** dont le serveur de nom est responsable. Un serveur de nom qui est responsable d'une certaine zone d'administration a l'autorité concernant toute

requête et réponse sur des domaines inclus dans sa zone d'administration (***Authoritative answers***). Lorsque le propriétaire d'un domaine fait enregistrer son domaine, il reçoit de l'administrateur de la zone contenant son domaine une liste de serveurs de nom autoritaires sur la zone d'administration en question.

Pour être fiable et efficace, le système DNS doit être décentralisé, et doit utiliser la délégation pour s'assurer que la responsabilité administrative peut le diviser en sous-domaines plus petits, qui peuvent être dirigés par des organisations indépendantes.

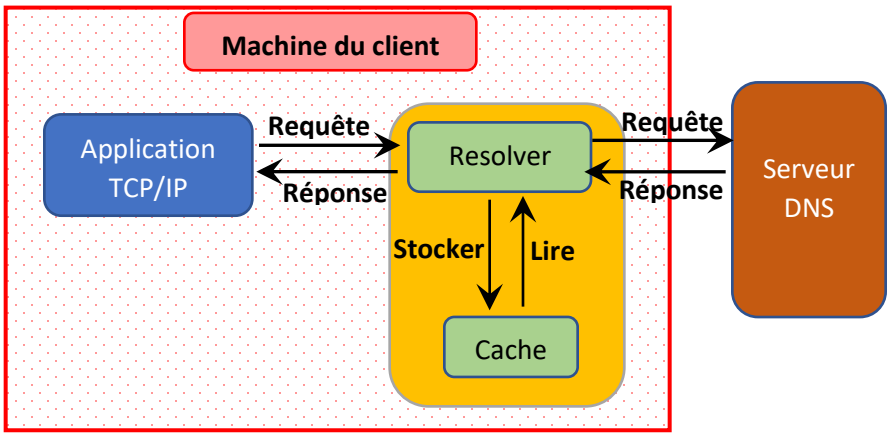
Les zones d'administration de l'espace de noms ne sont pas définies en fonction des domaines existants. C'est pourquoi un domaine divisé en plusieurs sous-domaines peut être administré par plusieurs serveurs de nom avec des zones d'administration différentes.

Si un serveur DNS reçoit une requête pour un domaine hors de sa zone d'autorité, alors il peut demander la résolution à un autre serveur DNS plus haut dans la hiérarchie de délégation.

2.1 Un service client-serveur

Le service DNS fonctionne en mode client-serveur. Une machine client fait une demande DNS (soit pour une résolution directe, soit pour une résolution inverse de nom), et une autre machine, jouant le rôle d'un serveur, donne la réponse à sa requête. La machine jouant le rôle du serveur est toujours un serveur DNS, tandis que la machine client peut être soit une machine d'un utilisateur, soit un serveur DNS demandant une résolution plus haut dans sa hiérarchie.

Lorsqu'une application TCP/IP sur une machine client a besoin d'une résolution DNS, elle contacte une application existante sur la même machine. Cette application s'appelle le ***resolver*** DNS. Celui-ci peut être configuré en modifiant un fichier ***/etc/resolv.conf***.

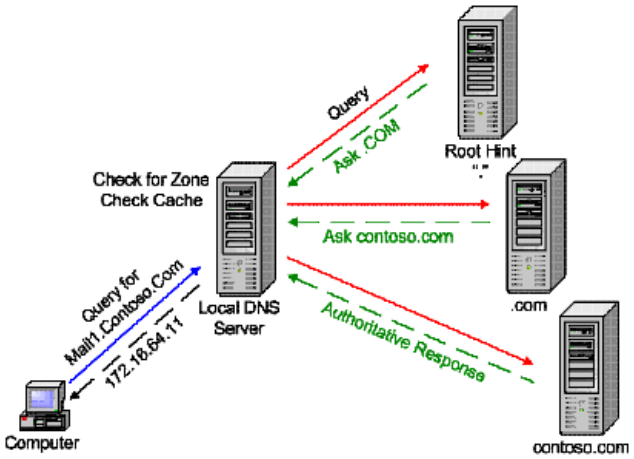


Dans l'illustration ci-dessus on voit l'interaction entre la machine du client et celle du serveur. Sur la machine du client, l'application TCP/IP fait premièrement sa requête vers le resolver. Le resolver cherche premièrement dans son cache pour savoir si une valeur existe déjà pour l'adresse ou le domaine cherché. Si cette valeur n'existe pas, le resolver contacte le serveur DNS.

Dans le cas ci-dessus on suppose que le serveur, étant autoritaire sur la requête du resolver, peut bien répondre à cette requête. Dans ce cas-ci, la réponse du serveur sera stockée dans le cache du resolver, puis relayée à l'application TCP/IP.

Si le serveur DNS n'est pas autoritaire sur la requête et ne connait pas la réponse, alors il doit demander plus loin. S'il connait un serveur qui est autoritaire sur le sujet de la requête, alors il fait sa demande directement. Sinon, il va devoir trouver d'abord un tel serveur.

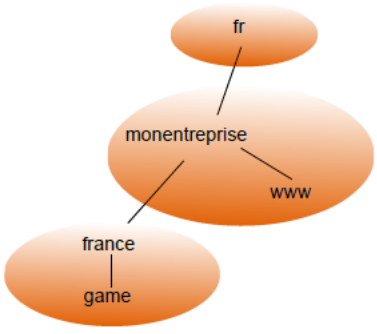
Dans ce dernier cas, le serveur va d'abord demander au serveur qui fait autorité sur le premier niveau du domaine dont on cherche l'adresse. Puis on demande plus loin, niveau par niveau.



Comme on peut voir dans cette figure, le rôle du serveur racine est essentiel pour trouver les informations que le client cherche. C'est pourquoi il est nécessaire d'avoir plusieurs serveurs de ce type, avec des redondances qui les aident à gérer les requêtes reçues.

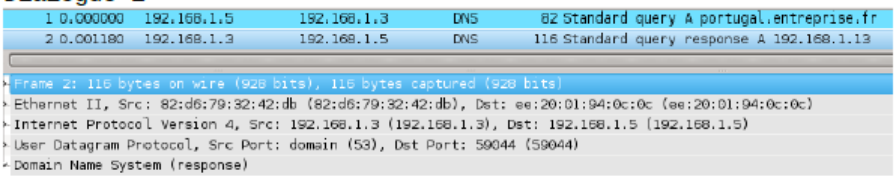
Exercice II

Nous partirons sur une machine ***www.monentreprise.fr*** dans la zone d'administration ***monentreprise.fr***.

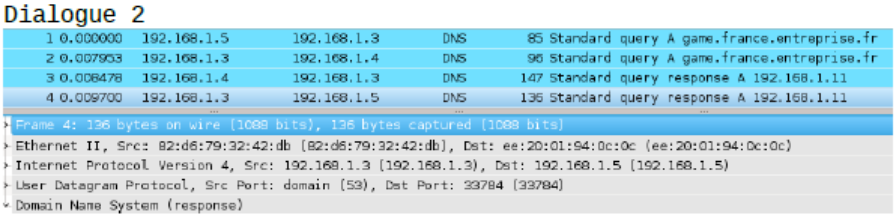


Les échanges captés ci-dessous sont des dialogues DNS. Etudiez-les en détails, puis répondez aux questions ci-dessous.

Dialogue 1



Dialogue 2



1. Regardez le premier dialogue DNS, puis complétez le tableau ci-dessous :

Participant	**Adresse IP**	**Port utilisé**	**Client/serveur**
1			
2			

2. Quel type de résolution (directe/inverse) est demandée dans le premier dialogue ? Justifiez votre réponse.

3. Quel est le nom et l'adresse IP de la machine recherchée dans le premier échange ?

4. Quelles machines participent au deuxième dialogue DNS ? Donnez leurs adresses IPs ainsi que leurs rôles (client/serveur) dans chaque échange (chaque ligne de la capture).

   Ligne 1 :

   Ligne 2 :

   Ligne 3 :

   Ligne 4 :

5. Expliquez ce qui se passe dans le deuxième dialogue DNS ci-dessus.

6. Décrivez précisément la fonctionnalité et la nature de la machine dont l'adresse IP est 192.168.1.4.

7. Comment la machine 192.168.1.3 a-t-elle su à qui s'adresser ?

3. Configuration d'un serveur DNS

Nous pouvons installer un serveur DNS à partir de plusieurs implémentations. L'implémentation la plus utilisée a été créée par l'Université de Berkeley et s'appelle Berkeley Internet Name Domain (BIND). Le daemon correspondant s'appelle ***bind*** ou ***named*** selon les versions (sur les machines netkit que vous allez utiliser il s'appelle ***bind9***).

Une fois l'installation réalisée on peut installer un ***name server*** primaire (un serveur qui sert de DNS pour son propre domaine local). Pour ce faire il faut utiliser le fichier ***/etc/bind/named.conf***. Ce fichier indique où se trouvent certains de fichiers, notamment :

- Le fichier utilisé pour la résolution directe : db.DOMAIN, qui donne la correspondance nom -> adresse IP.
- Le fichier utilisé pour la résolution inverse : db. ADDR, qui donne la correspondance adresse IP -> nom.

Voici un exemple du fichier named.conf :

```
zone “exemple.fr” in {
      type master;
      file “/etc/bind/db.exemple.fr”;
};
```

Le type indiqué est « master » . Ceci indique une référence vers le serveur DNS primaire, plutôt que par exemple un serveur secondaire.

La base de données DNS

Un serveur de DNS possède une base de données qui stocke des informations sur plusieurs noms de domaine. Plusieurs informations peuvent être associées à chaque nom, au-delà de juste son adresse IP. Si un resolver fait une requête concernant un certain domaine, alors le serveur DNS lui donnera toutes les informations (ou enregistrements -- « records » en anglais) qu'il possède concernant ce nom de domaine.

On appelle les informations associées à un domaine les enregistrements d'une ressource (« resource records » en anglais). Un ***resource record*** consiste en cinq éléments : le nom du domaine, la durée de vie, la classe, le type et finalement une valeur. Nous détaillons ces valeurs ci-dessous.

On peut avoir plusieurs entrées correspondant à un seul nom de domaine dans la base de données du serveur DNS. Les informations pertinentes associées à un nom pourraient être :

- Le nom du domaine indexe les ressources stockées dans la base de données du serveur DNS. C'est à partir de ce nom de domaine que le serveur cherchera des données à retourner aux requêtes du client. Un serveur DNS peut contenir plusieurs entrées avec un même nom de domaine.
- La durée de vie indique la stabilité des informations associées à chaque domaine. Si l'information est très stable, alors sa durée de vie est mise à une valeur très large (comme par exemple 86400. Si l'information est susceptible à être beaucoup modifiée dans le temps, alors la valeur de la durée de vie est bien plus basse, comme par exemple 60 (le nombre de secondes dans une minute).
- La classe d'un ***resource record*** consiste en un code qui indique la classe d'un domaine. En pratique on utilise presque toujours le code IN, pour Internet. Parfois, les DNS BIND utilisent le code CH pour indiquer par exemple la version d'un serveur.
- Le type d'un ***resource record*** peut prendre une valeur parmi les suivantes :
    - SOA (Start of Authority) : déclare le serveur de noms ayant l'autorité sur la zone, ainsi que les données (adresse mail) de l'administrateur de la zone et de plus quelques informations concernant la potentielle expiration et les mises à jour de la base de données.

- A (address) : une adresse IP associée au domaine en question. Si un domaine est associé en réalité avec plusieurs adresses, alors plusieurs entrées existeront dans la base de données, ayant le même nom de domaine. Le serveur DNS enverra alors l'adresse IP dans le premier RR pour une première requête, puis à la deuxième requête il enverra l'adresse du deuxième RR, etc.
- CNAME (canonical name) : Dans cette entrée on peut définir des alias, par exemple pour s'assurer qu'une personne qui tente de joindre une adresse presque pareille à la nôtre, peut en fait nous joindre. Par exemple le sous-domaine informatique.iut.limousin.fr pourrait se faire un alias info.iut.limousin.fr pour que les gens tapant info.iut.limousin.fr puissent le joindre. De plus, un serveur qui héberge plusieurs services doit souvent joindre une même machine physique pour joindre des domaines différentes.
- PTR (pointer) : Comme le CNAME, le PTR sert à associer un nom de domaine à une autre entrée. En pratique le PTR est utilisé pour la résolution inverse dans le domaine in-addr.arpa. Une feuille du domaine in-addr.arpa est un pointeur sur le nom de domaine correspondant.
- NS (name server) : un serveur de noms pour le domaine en question.
- MX (Mail exchanger) : ce champ indique quels serveurs de mail sont capables de recevoir de l'email pour un domaine donné. Si on configure mal ce champ, alors le mail ne pourra pas être livré.
- HINFO (host info) et TXT : deux champs qui permettent d'ajouter des informations concernant la machine qui hôte le domaine. Le premier de ces champs indique des paramètres de la machine comme le CPU et le système d'exploitation. Le deuxième indique des informations arbitraires.

- La valeur est associée à la valeur du type indiquée ci-dessus. Son format diffère en fonction du type : il peut s'agir d'un entier à 32 bits (pour une adresse IP), mais aussi d'un nombre de paramètres (pour le champ SOA). Les valeurs correspondant à chaque type de RR sont listées dans le tableau ci-dessous.

Type	Signifiance	Valeur
SOA	Start of Authority	Des paramètres pour la zone d'autorité en question
A	Address	Entier à 32 bits
MX	Mail Exchanger	Un domaine qui acceptera des mails
NS	Name server	Le nom d'un serveur de noms pour ce domaine
CNAME	Canonical Name	Alias pour le nom du domaine
PTR	Pointer	Pointer vers une adresse IP
HINFO	Host info	Des informations en hardware : CPU, OS
TXT	Text	D'autres informations sur la machine hôte

Exemple

Voici un exemple de base de données d'un serveur DNS. Le domaine ciblé est le domaine cs.vu.nl. Dans un premier bloc nous avons des informations sur le domaine et le serveur d'autorité sur le domaine. Puis,

on donne des informations sur une machine particulière, qui utilise un serveur avec un système d'exploitation Sun Unix, avec deux adresses IP. Finalement nous avons des informations sur une imprimante dont le nom est printer.

```
cs.vu.nl      86400  IN  SOA    Boss(9500,7200,7200,241920,86400)
cs.vu.nl      86400  IN  TXT    "Wiskunde en Informatica"
cs.vu.nl      86400  IN  TXT    "V.U. Amsterdam"
cs.vu.nl      86400  IN  MX     mailbox.cs.vu.nl.

mch.cs.vu.nl  86400  IN  HINFO  Sun Unix
mch.cs.vu.nl  86400  IN  A      130.37.16.20
mch.cs.vu.nl  86400  IN  A      192.31.231.157
mch.cs.vu.nl  86400  IN  MX     mailbox.cs.vu.nl

printer              IN  A      192.31.231.200
```

On peut également donner le contenu du fichier ***db.exemple.fr*** pour un domaine exemple.fr.

```
exemple.fr. IN SOA dns.exemple.fr  root.dns.exemple.fr {
      2019060201 ; numéro de serie AAAAMMJJNN
      10800 ; rafraîchissement
      3600 ; nouvel essai
      604800 ; obsolète au bout d'une semaine
      86400 ; TTL 1 jour
}

exemple.fr.   IN NS dns.exemple.fr

dns.exemple.fr.   IN   A       83.1.2.3
www.exemple.fr.   IN   A       83.1.2.4
ftp               IN   A       83.1.2.5

web.exemple.fr.   IN   CNAME   www.exemple.fr.
```

Le mot clé $ORIGIN et l'utilité de @

Pour la configuration des fichiers comme db.domaine.org, on peut utiliser le mot cle $ORIGIN. Ce mot clé est défini dans l'RFC 1035. Il définit une « origine », notamment un domaine (qui finit avec un point). Jusqu'à une prochaine définition, tout nom relatif sera interprété par rapport à l'origine définie.

On pourrait par exemple voir l'exemple suivant :

```
; exemple.fr
…
$ORIGIN exemple.fr.
…
mon A 192.168.1.35
; ceci définit l'association entre mon.exemple.fr et 192.168.1.35
```

Dans l'exemple ci-dessus le mot « mon » indique un nom de domaine relatif par rapport à l'origine et se réfère au domaine mon.exemple.fr.

Le même RFC 1035 définit le caractère « @ » comme étant une façon de faire une référence à l'origine définie. Si, dans l'exemple ci-dessus on remplaçait « mon A 192.168.1.35 » par « @ A 192.168.1.35 », alors l'adresse donnée serait celle de exemple.fr.

Exercice 3

Regardez l'exemple détaillant le fichier ***db.exemple.fr*** ci-dessus.

Utilisez les mots clé $ORIGIN et @ pour réécrire les contenus de ce fichier.

```
_____________________________________
_________  IN SOA ____________________________________ {
      2019060201 ; numéro de serie AAAAMMJJNN
      10800 ; rafraîchissement
      3600 ; nouvel essai
      604800 ; obsolète au bout d'une semaine
      86400 ; TTL 1 jour
}

________   IN NS _________________

_________________   IN   A       83.1.2.3
_________________   IN   A       83.1.2.4
___                 IN   A       83.1.2.5

_________________   IN   CNAME   ___________________
```

Exercice 4

Nous considérons un domaine dont le nom sera ***domaine.org***. Celui-ci contient un nombre de machines, listées ci-dessous avec leurs adresses IP respectives.

Type de machine	**Adresse IP**
firewall	192.168.56.254
dns	192.168.56.11
http	192.168.56.15
www	192.168.56.15
files	192.168.56.18

Complétez ci-dessous le contenu du fichier ***db.domaine.org*** correspondant à ce domaine.

```
_____________________________________

_________ IN SOA _____________________________ {
        2019060201 ; numéro de serie AAAAMMJJNN
        10800 ; rafraîchissement
        3600 ; nouvel essai
        604800 ; obsolète au bout d'une semaine
        86400 ; TTL 1 jour
}

; serveurs de noms

_______________   IN    NS   ____________________

_______________   IN    A    ____________________


; hôtes declares
```

4. Fonctions supplémentaires DNS

Une des fonctions essentielles du service DNS est la résolution inverse, c'est-à-dire pouvoir trouver le nom d'une machine à partir d'une adresse IP. Pour ce faire il faudrait définir un deuxième fichier de type db, qui portera le nom de l'adresse IP correspondant au domaine qu'on décrit. Ce fichier devra être créé et puis déclaré dans le fichier named.conf.

Dans ce deuxième fichier db, on devra utiliser le domaine in-addr.arpa. Pour la résolution inverse, on se rappelle que les octets des adresses IP, en ordre inversé, sont interprétés comme des domaines. Prenons par exemple un domaine 192.168.1 (attention, on n'a mis que les premiers 3 octets), le fichier db utilisé s'appellera db.192.168.1 et le domaine concerné sera 1.168.192.in-addr.arpa.

La délégation de zone

Il est essentiel pour un serveur DNS d'être capable de déléguer une partie de sa zone d'autorité. Ceci peut être réalisé en indiquant, dans la configuration DNS du serveur parent, les références du serveur de nom de niveau inférieur. On utilise alors un enregistrement de type NS et un de type A.

Exercice 5

On reprend le domaine indiqué dans l'exercice 4. On veut faire la résolution inverse, en écrivant un fichier db.192.168.56.

```
_____________________________________


_____________________  IN SOA
_____________________________ {
        2019060201 ; numéro de serie AAAAMMJJNN
        10800 ; rafraîchissement
        3600 ; nouvel essai
        604800 ; obsolète au bout d'une semaine
        86400 ; TTL 1 jour
}

; serveurs de noms

_________________  IN   NS   ____________________

_________________  IN   A    ____________________


; hôtes declares
```

Exercice 6

On considère le domaine domaine.org, qu'on avait vu précédemment. Pour le service DNS, nous aurons une zone d'administration serveurs.domaine.org, qui aura deux sous-zones d'administration, notamment prive.serveurs.domaine.org et public.serveurs.domaine.org. Dans cet exercice nous voulons justement faire la délégation de l'autorité par rapport à cette division. A chaque fois le nom du serveur DNS pour un sous-domaine sera le nom du sous-domaine, préfixé par dns. Le tableau ci-dessous indique les adresses IP des serveurs DNS responsables pour chaque zone d'administration.

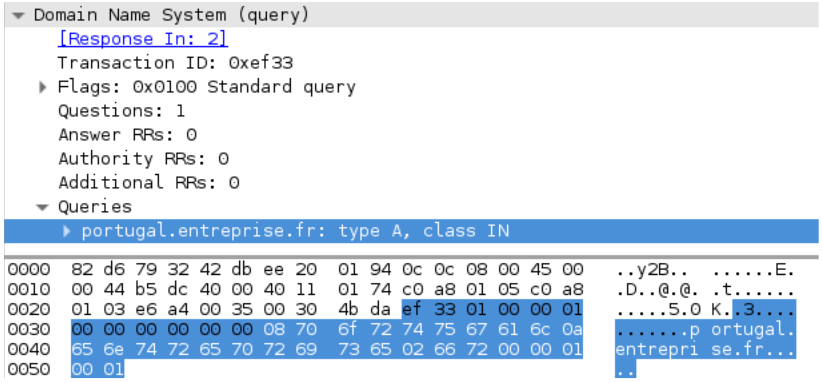
Zones d'administration	Adresse IP du serveur DNS responsable
serveurs.domaine.org	192.168.33.200
prive.serveurs.domaine.org	192.168.1.100
public.serveurs.domaine.org	192.168.2.100

1. Indiquez ce qu'il faut ajouter au fichier db.domaine.org pour indiquer le sous-domaine serveurs.domaine.org avec son serveur de noms.

2. Indiquez ce qu'il faut ajouter au fichier db.serveurs.domaine.org pour indiquer ses deux sous-domaines avec leurs serveurs DNS correspondants.

Exercice 7

Voici une capture d'écran pour une certaine requête DNS. Cette requête est reçue sur le port UDP/53.



En regardant cette requête, indiquez quel est le format des messages DNS.