TCP/IP - DNS

Roger Yerbanga
contact@yerbynet.com

Transfer 1.3 - Bobo-Dioulasso - Décembre 2012

Pourquoi les noms ?

Les ordinateurs utilisent des adresses IP.
Pourquoi avons nous besoin des noms?

- Faciles à mémoriser par les humains
- Les ordinateurs peuvent être déplacés entre réseaux, dans ce cas leurs adresses IP changent

DNS, fondamentale pour Internet

- Les adresses IP représentent une partie de l'infrastructure fondamentale de l'Internet.
  - Adressage IP
  - Routage IP (le paquet est routé selon sa destination)
- Une autre partie fondamentale de l'Internet est le DNS.
- Les applications Internet sont souvent dépendantes des DNS :
  - Comment joindre des sites Web (URL) ?
  - Comment échanger du courrier électronique ?
  - Comment gérer les changements d'adresses IP ?

But du DNS

- Les équipements communiquent grâce à leurs IP
- Les applications utilisent les noms des équipements
- A une adresse IP peut correspondre un ou plusieurs nom (alias)
- Un nom doit être unique au monde
- **But** : Identifier les ressources sur le réseau par des noms

Domain Name System

- DNS : ***Domain Name System.***
- Base de données hiérarchique et distribuée dans Internet.
- Contient des correspondances entre :
  - Nom ordinateur -> adresse IP
  - Passerelle de courrier d'un domaine -> adresse IP
  - Traduction inverse : IP -> nom ordinateur
  - Bien plus que seulement nom/adresse IP
  - Applications (navigateurs, Telnet, ftp, ssh, etc.) utilisent les adresses IP pour joindre les serveurs.
- Système d'exploitation responsable de la résolution des noms de domaine (client DNS).

Ancienne solution : hosts

- Le fichier est maintenu de façon centralisée et distribué à toutes les machines sur Internet

```
128.4.13.9          SPARKY
4.98.133.7          UCB-MAILGATE
200.10.194.33       FTPHOST
... etc
```

Cette rubrique existe encore:
/etc/hosts [Unix]

Fichier Hosts (Lmhosts)

- ARPAnet : 1er réseau IP avec quelques centaines d'ordinateurs.
- Chaque ordinateur possédait un fichier /etc/hosts pour faire la correspondance

  Exemple de fichier /etc/hosts :

  ```
  127.0.0.1     localhost.localdomain localhost
  10.0.1.25     gate.thetueur.bj gate
  ```

- Problèmes : Collision des noms, déploiement à grande échelle, changement d'adresse IP

hosts est inadapté à grande échelle

- Fichier volumineux
- Nécessite d'être copié fréquemment sur toutes les machines
- Pas uniforme
- Toujours dépassé
- Pas d'unicité des noms
- Un seul point d’administration

Le “Domain Name System” est né

- Le DNS est une base de données distribuée qui fait correspondre le nom à une adresse IP (et/ou à d’autres informations)
- Distribuée:
  - Partage l’administration
  - Partage la charge
- Robustesse et performance à travers :
  - La réplication
  - Le système cache

- Une pièce ***critique*** de l’infrastructure Internet

Résolution de Noms

- Processus consistant à mapper un nom d'hôte sur une adresse IP
- Méthodes standard de résolution des noms d'hôte :
  - Nom d'hôte local
  - Fichier Hosts
  - Serveur DNS
- Les méthodes de résolution des noms d'hôte sont configurables

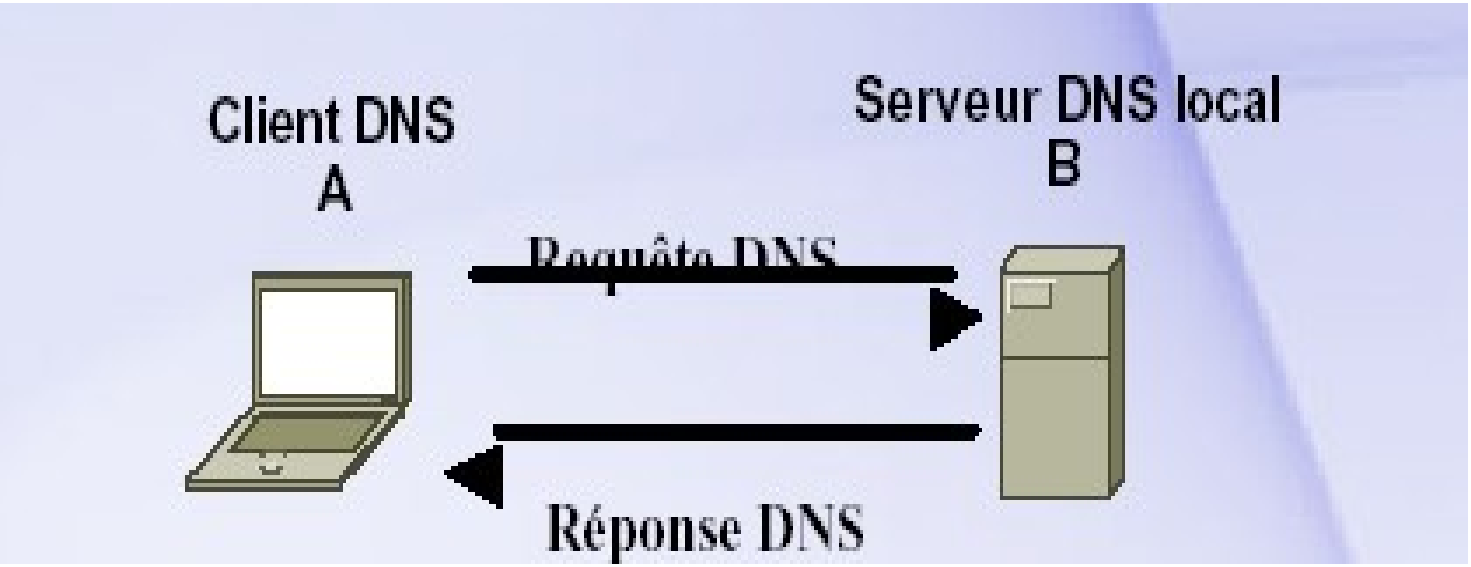
Requête DNS simple (1)

- Client DNS : Accéder à un ordinateur en utilisant son nom d'hôte et son domaine
- Serveur DNS : Contient les correspondances pour un domaine, répond aux requêtes pour ce domaine, interroge d'autres serveurs DNS pour d'autres domaines
- FQDN (*Fully Qualify Domain Name*) =

  Nom d'ordinateur+Domaine (kalifa.afribone.net.ml.)
- Ports UDP/TCP 53 ont été réservés par l'IANA pour le DNS.

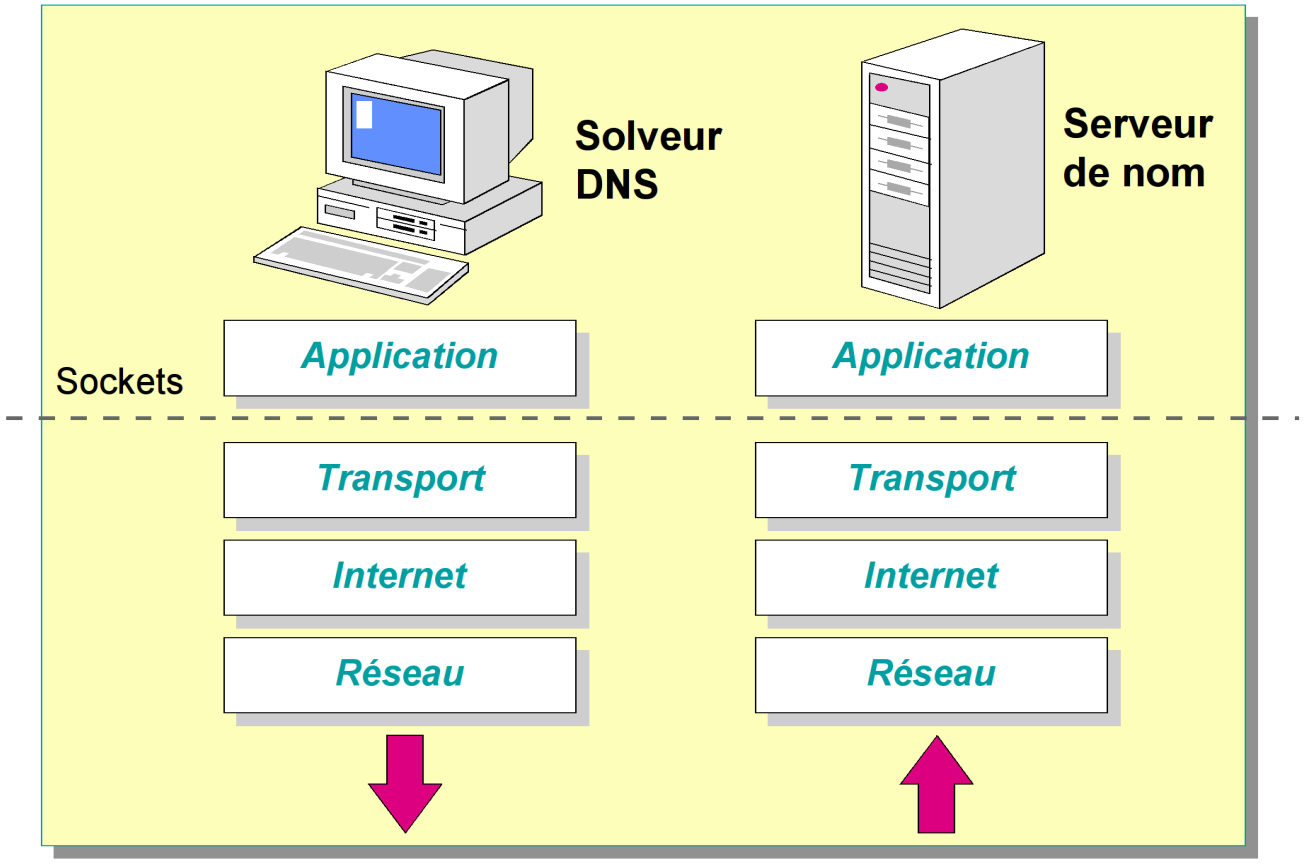
Requête DNS simple (2)

- Le «résolveur» DNS d'un ordinateur (client DNS) interroge le serveur DNS local avec UDP sur le port 53.



DNS & Résolution de noms

- DNS : Domain Name Service = Base de données répartie



Gestion des domaines Internet (1)

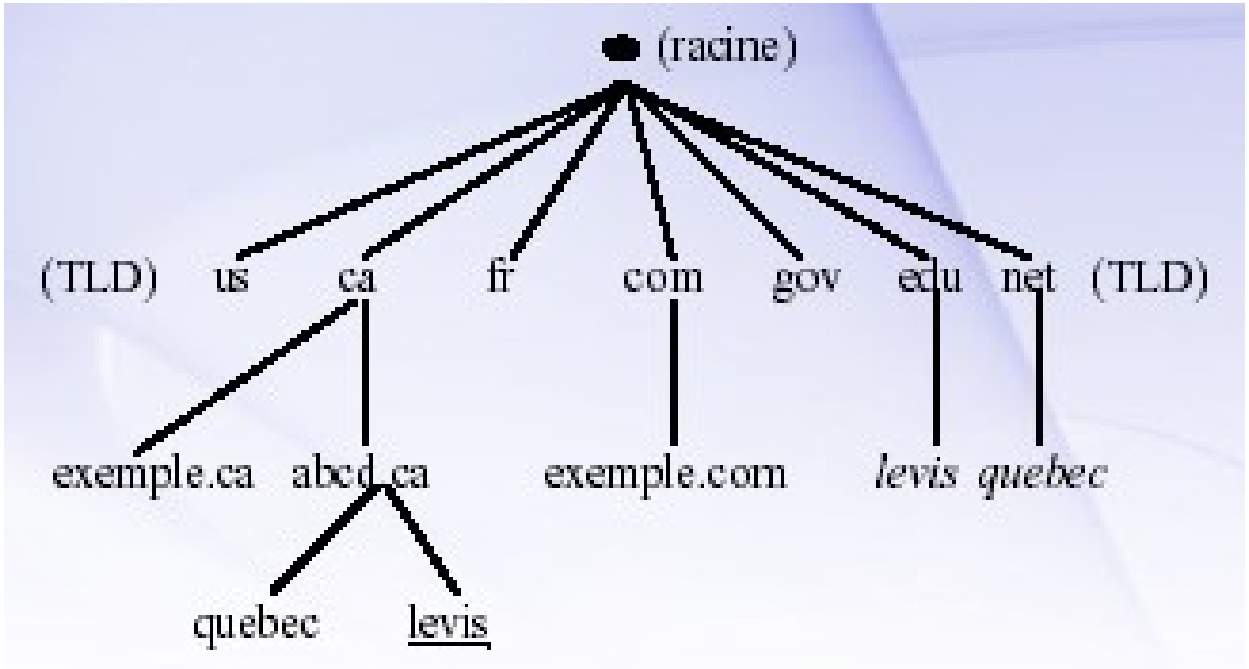
- L’administration des noms de domaines est hiérarchisée :
  - IANA (Internet Assigned Numbers Authority) : responsable de la coordination mondiale
- ET décentralisée :
  - IANA délègue à ONATEL la gestion des noms de domaines au Burkina Faso (.bf)
    - ONATEL délègue à l'UPB la gestion des noms se terminant par upb.bf.

Gestion des domaines Internet (2)

- Au Mali (.ml) : Gestion assurée par la SOTELMA
- En règle générale : Le gestionnaire du domaine X est responsable :
  - de la délégation des noms de domaines de la forme Y.X
  - de la désignation d'un administrateur du domaine Y.X
- Exemple : refer.bj.

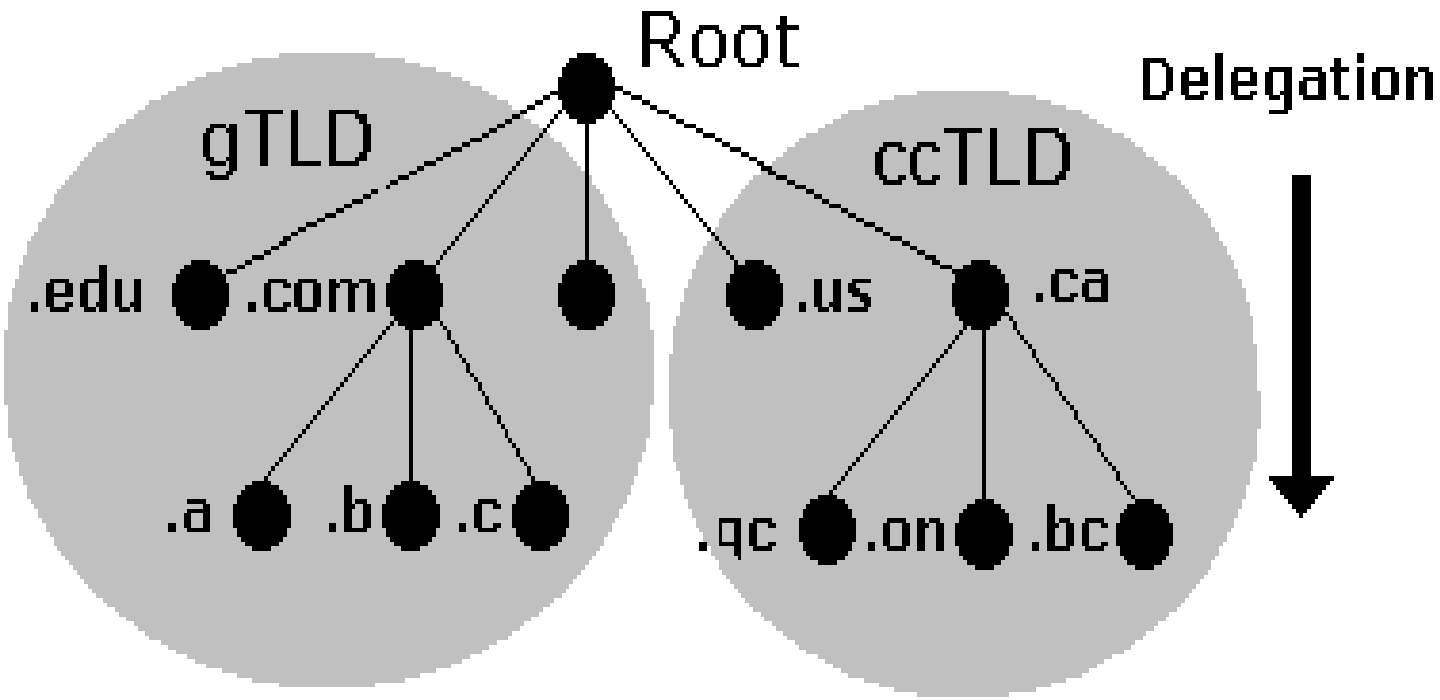
Hiérarchie des domaines Internet (1)

- Structure hiérarchique des noms de domaines Internet



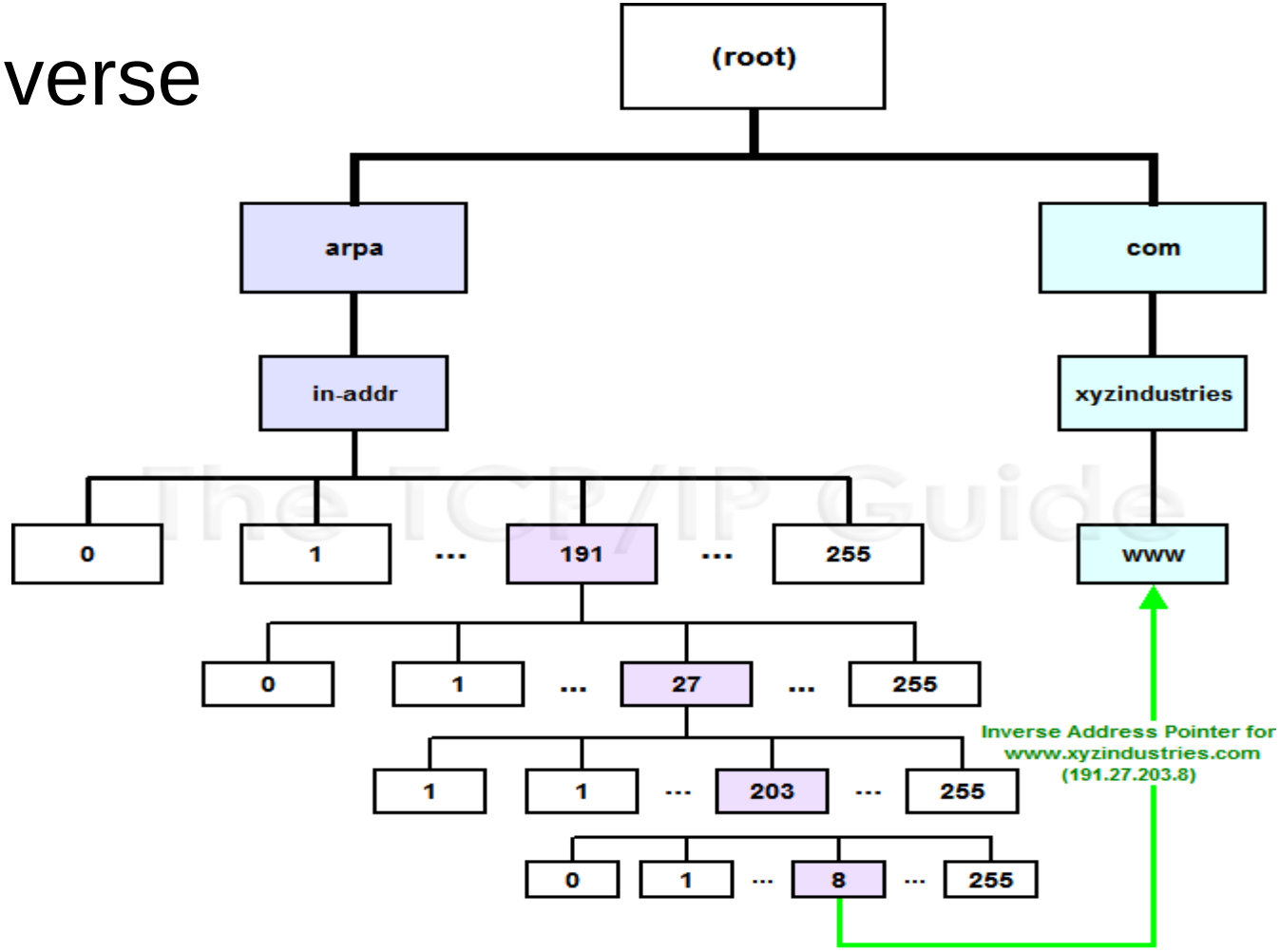
Hiérarchie des domaines Internet (2)

- Generique et Country Code TLD



Hiérarchie des domaines Internet (3)

- Reverse



Hiérarchie des domaines Internet (4)

- L'espace des noms est arborescent
- Divisé en plusieurs niveaux :
  - Root (.)
  - Top Level Domain (com, net, org, ml, fr, ca, arpa)
  - Secondary Level Domain, …
- A chaque nœud de l'arbre est associé :
  - un ensemble de ressources
  - et un Nom
- Nom de domaine d'un nœud = Suite de domaines en remontant vers la racine séparés par des « . »

DNS root servers (1)

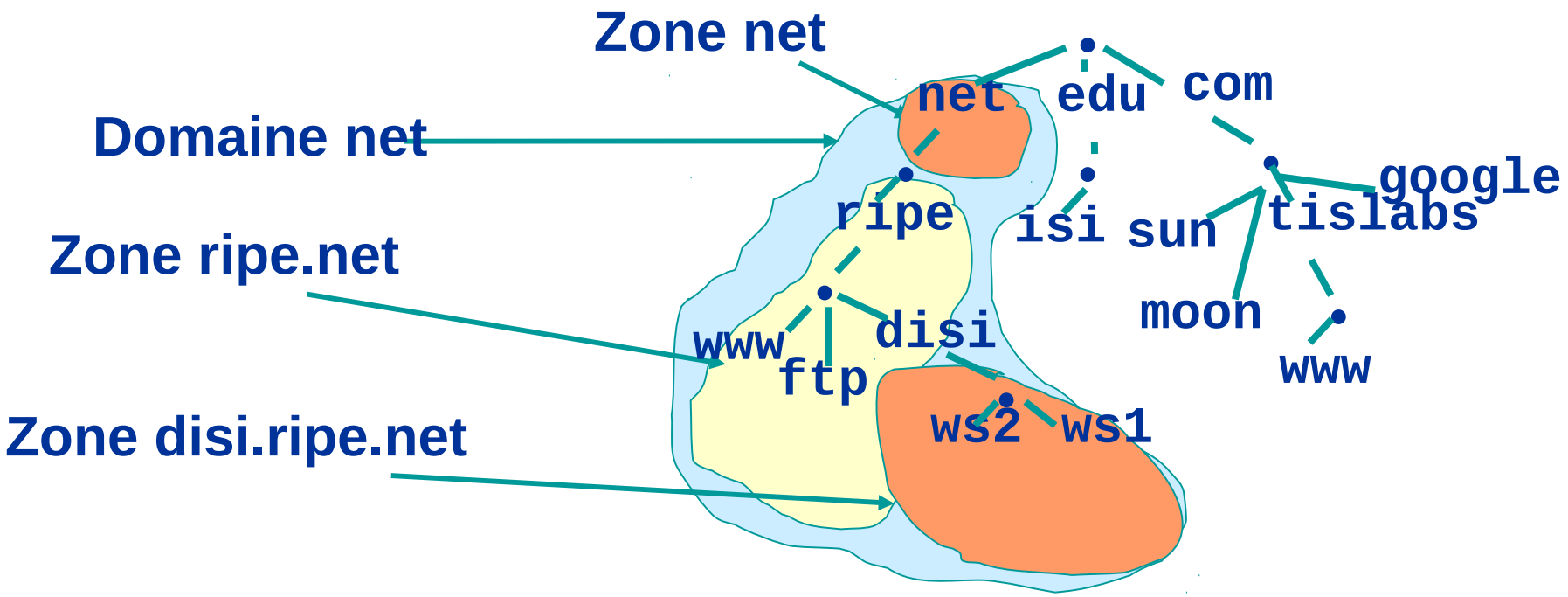
- 13 serveurs racines : x.*root-servers.net (A,..., M)*
- Dispersion géographique des serveurs en Asie, en Europe, aux USA.
- Anycast => C,F,I,J,K,L,M éparpillés sur plusieurs continents.
- Diversité de systèmes d'exploitation et de plate-formes matérielles.
- Exploite des concepts avancés de distribution de charge sur Internet.
- Contient la liste de tous les NS (*name servers*) reconnus par NIC

DNS Root Servers (2)

- Lorsqu'un serveur DNS local recherche un nom de domaine Internet qu'il ne connaît pas, il interroge l'un de ces 13 serveurs racines.
- Chaque serveur DNS dispose d'un fichier de configuration contenant les adresses IP de ces 13 serveurs DNS racines (*root servers*).
- Extrait d'un fichier **named.root**

; formerly NS.INTERNIC.NET
. 3600000 IN NS A.ROOT-SERVERS.NET.
*A.ROOT-SERVERS.NET. 3600000 A **198.41.0.4***
;
. 3600000 NS B.ROOT-SERVERS.NET.
*B.ROOT-SERVERS.NET. 3600000 A **128.9.0.107***
;

DNS est hiérarchique (1)



Structure de l'arborescence

DNS est hiérarchique (2)

- Donne globalement des noms uniques
- Administration par zones (des parties de l'arborescence)
- Vous pouvez donner (" déléguer ") le contrôle d'une partie de l'arborescence sous votre autorité
- Exemple:
  - .net est sur un ensemble de serveurs de noms
  - ripe.net est sur un ensemble différent
  - disi.ripe.net est sur un autre ensemble

Les noms de domaine sont (presque) illimités

- Longueur totale de 255 caractères maximum
- 63 caractères maximum dans chaque partie (RFC 1034, RFC 1035)
- Quelques restrictions à respecter :
  - RFC 1123
  - a-z, 0-9 et tiret (-) uniquement
  - Pas d'underscores (_)
  - Le tiret ne peut commencer ni terminer un nom de domaine
  - IDN va lever pas mal de restrictions

Utilisation du DNS

- Un nom de domaine (comme www.trstech.net) est une CLEF pour rechercher une information
- Le résultat est un ou plusieurs Enregistrements de Ressources (Record Resources ou RR)
- Il y a différents RR pour différents types d'information
- Vous pouvez demander le type spécifique que vous voulez, ou demandez " n'importe quel '" RR associé au nom de domaine

Vue générale des RRs

- A (adresse): associe le nom d'hôte à l'adresse IPv4
- AAAA : Adresse IPv6
- PTR (pointer): associe l'adresse IP au nom
- MX (Mail eXchanger): où délivrer le courrier pour l'adresse user@domain
- CNAME (Canonical NAME): associe un nom alternatif au nom réel
- TXT (text): tout texte descriptif
- NS (Name Server), SOA (Start Of Authority): sont utilisés pour la délégation et la gestion du DNS lui-même

Exemple Simple

- Query: www.afribone.net.ml
- Query type: A
- Result:

www.afribone.net.ml. IN A 196.200.57.132

Dans ce cas, juste un RR est trouvé,
mais en général, de multiples RR peuvent être retournés

IN est la “class” d’INTERNET, utilisé par DNS

Résultats Possibles

- Positifs (un ou plusieurs RR sont trouvés)
- Négatifs (aucun RR ne correspond à la requête)
- Échec de serveur (ne peut pas trouver la réponse)

Comment utilisez une adresse IP comme clef pour une requête DNS

- Convertir l'adresse IP au format décimal
- Intervertir les quatre parties
- Ajouter '" in-addr.arpa. " à la fin
- TLD spécial réservé à cette fin

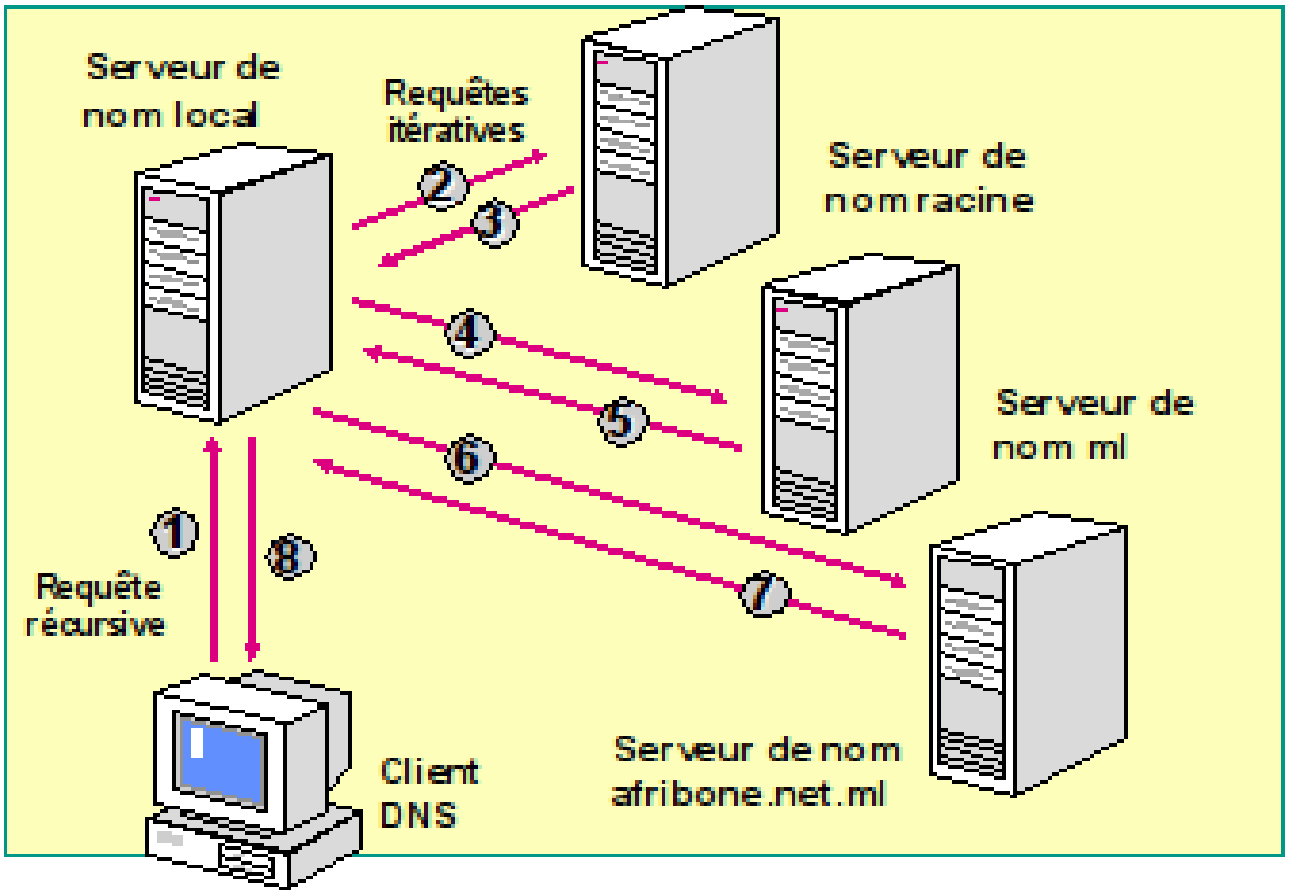
Exemple pour chercher le nom de 212.74.101.10
10.101.74.212.in-addr.arpa. est PTR www.tiscali.co.uk

Connue comme " une consultation du DNS inverse "

Parce que nous recherchons le nom pour une adresse IP, plutôt que l'adresse IP pour un nom

Exemples résolution de noms

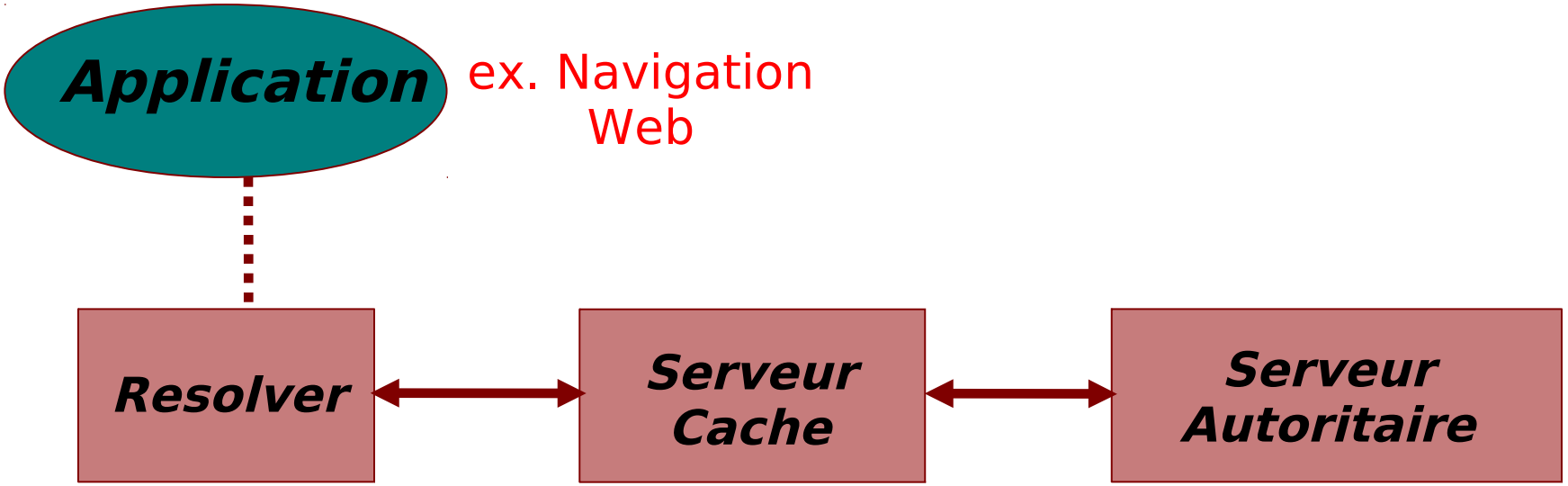
- Résolution de noms de DNS



Le DNS est une application Client/Serveur

- Fonctionne à travers un réseau
- Les requêtes et les réponses sont normalement envoyées dans des paquets UDP(port 53)
- De temps en temps utilise TCP(port 53)
  - pour les requêtes très grandes: transfert de zone,etc...

Il y a trois rôles impliqués dans le DNS



Trois rôles du DNS(1)

- Le **RESOLVER**
  - prend la demande de l'application,
  - formate la demande dans le paquet UDP
  - Envoie la demande au cache DNS
- **SERVEUR CACHE**
  - Renvoie la réponse si elle est déjà connue
  - Autrement,recherche un serveur autoritaire qui a la donnée
  - Met le résultat en cache pour de requêtes futures
  - Egalement connu sous le nom de Serveur RECURSIF
- **SERVEUR AUTORITAIRE**
  - Contient l'information réelle mise dans le DNS par l'administrateur du domaine

Trois rôles du DNS(2)

- Le MEME protocole est utilisé pour la communication du resolver <--> cache et du cache <-->serveur autoritaire
- Il est possible de configurer un même serveur de nom en tant que serveur cache et serveur autoritaire à la fois
- Mais il exécute toujours un seul rôle pour chaque requête entrante
- Ceci est courant, mais N'EST PAS RECOMMANDE

Rôle 1: LE RESOLVER

- Un morceau de logiciel qui formate une requête DNS dans un paquet UDP, l'envoie à un serveur cache, et décode la réponse
- Habituellement une bibliothèque partagée (ex. libresolv.so sous Unix) parce que beaucoup d'applications ont besoin de lui
- Chaque machine a besoin d'un resolver – ex chaque poste de travail ubuntu en a au moins un.

Comment le resolver trouve-t-il le serveur cache?

- Il doit être explicitement configuré (manuellement ou par l’intermédiaire du DHCP, etc...)
- Il doit être configuré avec l'ADRESSE IP du serveur cache (pourquoi pas le nom?)
- Bonne idée de configurer plus d'un cache

Quel serveur cache utiliser ?

- Vous devez avoir la PERMISSION d'utiliser le serveur cache
  - Ex. serveur cache de votre ISP, ou le vôtre
- Préférer un serveur cache proche
  - Réduit au minimum le temps d'aller-retour et les pertes de paquets
  - Peut réduire le trafic sur votre liaison externe, puisque souvent le serveur cache peut répondre sans contacter d'autres serveurs
- Préférer un serveur cache fiable
  - Peut-être votre propre serveur cache

Exemple: La configuration du resolver de Linux

/etc/resolv.conf

search refer.tg
nameserver 196.28.9.1
nameserver 212.74.112.67

C'est tout ce dont vous avez besoin pour configurer un resolver

Les tests du DNS

- Saisir dans la zone adresse de votre navigateur : "www.yahoo.com" ?

- Pourquoi ceci n’est pas un bon essai?

Tester le DNS avec "dig"

- "dig" est un programme qui effectue des requêtes DNS et affiche les résultats
- Mieux que "nslookup", "host" parce qu'il montre l'information DNS complète

```
dig tiscali.co.uk.
  -- par défaut pour demander le type "A"
dig tiscali.co.uk. mx
  -- indique le type de requête
dig @196.200.90.99 tiscali.co.uk. mx
  -- Envoyé à un cache DNS particulier (Bypass
/etc/resolv.conf)
```

Le point à la fin d'un nom de domaine

dig tiscali.co.uk. 

- Empêche l'ajout des domaines par défaut
- Prendre l'habitude de l'utiliser au cours des tests du DNS

→ **seulement sur des noms de domaine, pas sur les adresses IP**

dig @ben02.gouv.bj. www.gouv.bj.

```
; <<>> DiG 9.5.1-P3 <<>> @ben02.gouv.bj. www.gouv.bj.
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21208
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 1
;; WARNING: recursion requested but not available
;; QUESTION SECTION:
;www.gouv.bj.           IN  A
;; ANSWER SECTION:
www.gouv.bj.       86400  IN  CNAME   waib.gouv.bj.
waib.gouv.bj.      86400  IN  A   81.91.232.2
;; AUTHORITY SECTION:
gouv.bj.        86400  IN  NS  ben02.gouv.bj.
gouv.bj.        86400  IN  NS  nakayo.leland.bj.
gouv.bj.        86400  IN  NS  ns1.intnet.bj.
gouv.bj.        86400  IN  NS  rip.psg.com.
;; ADDITIONAL SECTION:
ben02.gouv.bj.     86400  IN  A   81.91.232.1
;; Query time: 227 msec
;; SERVER: 81.91.232.1#53(81.91.232.1)
;; WHEN: Wed Nov 11 15:15:37 2009
;; MSG SIZE  rcvd: 178
```

Interprétation des résultats : header

- STATUS
  - NOERROR: 0 ou plusieurs RRs sont trouvés
  - NXDOMAIN: domaine inexistant
  - SERVFAIL: le serveur cache ne pouvait pas localiser la réponse
- FLAGS

  AA: Réponse de serveurs autoritaires

  Vous pouvez ignorer les autres

  QR: Query/Response (1 = réponse)

  RD: Recursion Desired (résursion désirée)

  RA: Recursion Available (résursion disponible)

Interprétation des résultats

- Answer section (Les RRs demandés)
  - Chaque enregistrement a un temps de vie (TTL)
  - Dit combien de temps le cache gardera la donnée
- Authority section
  - Quels serveurs de noms sont autoritaires pour ce domaine
- Additional section
  - Plus d'enregistrements (RRs) : généralement des adresses IP pour les serveurs de noms autoritaires
- Total query time
- From
  - Vérifier quel serveur a donné la réponse!
  - Si vous faites une faute de frappe, la requête peut aller à un serveur par défaut

Exercices Pratiques

- Configurer le resolver Unix
- Faire des requêtes DNS en utilisant 'dig'
- Utiliser ‘tcpdump' pour afficher les requêtes émises qui sont envoyées au cache

Serveurs DNS autoritaires

- Au moins un serveur DNS autoritaire pour assurer la diffusion du domaine.
  - Appelé serveur autoritaire primaire ou principal
  - Contient les données (enregistrements) à jour sur le domaine dans un fichier *zone*
- Second serveur DNS autoritaire :
  - serveur autoritaire secondaire
  - contient une copie de tous les enregistrements du domaine téléchargée à partir d'un serveur maître
  - Répond également aux requêtes DNS

REPLICATION DNS

- Pour chaque domaine, nous avons besoin de plus d'un serveur autoritaire avec la même information (RFC 2182)
- Les données sont enregistrées sur un seul serveur
- (maître) et répliquées sur les autres (les esclaves)
- Le monde extérieur ne "peut faire la différence" entre le maître et le slave
  - Les enregistrements NS sont retournés de façon aléatoire pour le partage de charge égal

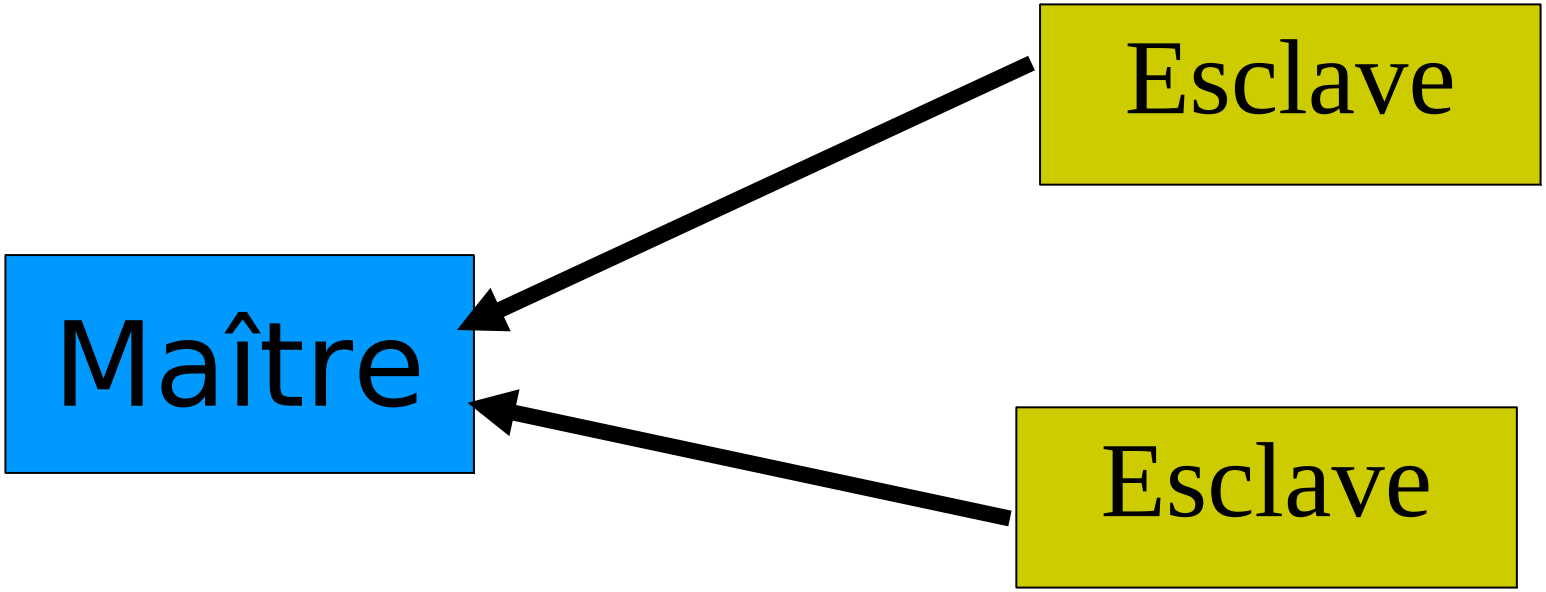
Sont aussi appelés "primaire" et "secondaire"

Serveur « esclave »

- Un serveur n'est pas forcement secondaire (ou esclave) pour tous les domaines d'un autre serveur.
- Un serveur est esclave pour un domaine et peut même être maître pour d'autres domaines.
- Serveur A peut être maître de Serveur B pour certains domaines, et Serveur B peut être maître de A pour d'autres domaines.

Serveur « esclave »

Les esclaves se connectent au maître pour rechercher la copie des données de la zone



- Le maître n'envoie pas automatiquement les données aux esclaves

Quand est-ce que la replication a lieu ?

- L'esclave scrute le maître périodiquement– appelé **``Temps de rafraîchissement``**
  - A l'origine c'était le seul mécanisme

- Une extension du protocole permet maintenant au maître d'informer les esclaves quand les données ont changé
  - Aboutit à des mises à jours plus rapides
  - DNS NOTIFY

- La notification est incertaine (ex. le réseau peut perdre le paquet); ainsi nous avons toujours besoin de contrôler l'intervalle de rafraîchissement

Fichier de zone (1)

- Fichier qui contient l'en-tête et les enregistrements d'un domaine.
- En-tête du fichier de zone contient :
  - Nom du domaine et durée de vie pour les enregistrements de ce domaine (sert à la cache DNS).
  - 1 enregistrement SOA (*Start of Authority*) par domaine :
    - Identifie un serveur DNS comme **autoritaire primaire.**
    - Identifie une adresse de courrier électronique.
    - Numéro de série : à changer à chaque modification/ajout/retrait d'enregistrements.
    - Intervalle pour le rafraîchissement du fichier par les serveurs autoritaires secondaires.
    - Intervalle pour l'expiration du fichier de zone (durée de vie limitée).
    - Intervalle pour la cache négative pour les enregistrements de ce domaine.

Fichier de zone (2)

- Exemple d’en-tête de fichier de zone

```
$TTL 3D                 ; Cache DNS = 3 jours
@    in   SOA   ns1.test.ml. root.test.ml. (
                2004011600 ; Serial yyyymmdd##
                10800 ; Refresh (3h)
                1H ; Retry (une heure)
                604800 ; Expire (1 sem) 1W
                43200 ) ; Minimum (12 heures)
```

Fichier de zone (3)

- ns1.test.ml : DNS maître pour la zone test.ml
- root.test.ml = root@test.ml = responsable
- Les DNS secondaires téléchargent les informations de la zone à partir d'un DNS primaire.
  - Ils vérifient toutes les 3H (*refresh=10800*) si une nouvelle version est disponible en comparant le *serial*.
  - En cas d'echec → nouvelle tentative toutes les heures *(retry=1H)* jusqu'à la péremption des informations au bout d'une semaine (*expire=1W*).
  - Informations conservées dans un cache DNS au moins ½ jour (*minimum=43200=12h*).

Types d'enregistrements dans un fichier de zone (1)

- **SOA** : *Start of Authority*
  - Identifie l'autorité et définit les paramètres pour un domaine.
- **NS** : *Name Server* (serveur DNS)
  - Liste les serveurs DNS autoritaires pour ce domaine.
- **MX** : *Mail eXchanger*
  - Identifie le(s) serveur(s) de courrier électronique responsable(s) de recevoir le courrier pour ce domaine.
  - Il est possible d'identifier plusieurs serveurs SMTP donc plusieurs enregistrements MX et d'accorder des préférences.
- **A** : *Name-to-Address Mapping*
  - Correspondance entre un FQDN et une adresse IP.
- **CNAME** : *Canonical Name*
  - Alias pour un FQDN.
  - 2 FQDN ayant la même correspondance.

Types d’enregistrements dans un fichier de zone (2)

- Exemples d’enregistrements NS, MX, A, CNAME

```
$TTL 3D                 ; Cache DNS = 3 jours
@    in     soa     ns1.test.ml. root.test.ml. (
                    2002092400 ; numéro de série yyyymmdd##
                    10800 ; Refresh (3h)
                    900 ; Retry (15 min)
                    604800 ; Expire (1 sem)
                    43200 )  ; Negative cache (12 heures)
                    in              ns      ns1.test.ml.
                    in              mx      10      mail.test.ml.
                    in              mx      20      mail2.test.ml.
ns1         in      a       10.0.0.1
proxy       in      a       10.0.0.2
www         in      cname   ns1.test.ml.
mail        in      cname   ns1.test.ml.
cache       in      cname   proxy.test.ml.
```

Configuration d'un serveur DNS sous Linux (1)

- Editer le fichier « **/etc/named.conf** »
- Ajouter la zone à créer
- Exemple :

```
zone "test.ml"  {
        type master ;
        file "db.test.ml" ;
} ;
```

Configuration d'un serveur DNS sous Linux (2)

- Créer le fichier de zone « db.test.ml »

```
$TTL 86400
@       IN      SOA     ns1.test..ml.   root.test.ml.       (
                        2004011601 ; Serial
                        28800      ; Refresh
                        14400      ; Retry
                        3600000    ; Expire
                        3600 )     ; Minimum

        IN      NS      ns1.test.ml.
        IN      MX      10      ns1.test..ml.
ns1     IN      A       10.0.0.3
mail    IN      CNAME   ns1.test.ml.
cache   IN      CNAME   ns1.test.ml.
ftp     IN      CNAME   ns1.test.ml.
```

Configuration de l'esclave

- /etc/named.conf pointe vers l'adresse IP du maître et l'emplacement du fichier de zone
- Le fichier de zone sont transféré automatiquement

```
zone "example.com" {
        type slave;
        masters { 192.188.58.126; }
        file "s/example.com";
        allow-transfer  { none;};
};
```

Maître et esclave

- Il est maintenant clair, qu'un serveur peut-être maître pour certaines zones et esclave pour d'autres au même moment

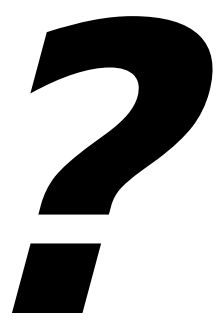
- C'est pourquoi nous recommandons de maintenir les fichiers dans des sous-répertoires différents
  - <chemin>/local
  - <chemin>/slave

allow-transfer { ...; }

- Les machines à distance peuvent demander le transfert du contenu entier d'une zone
- Par défaut, ceci est autorisé à n'importe qui
- Vaut mieux limiter ceci
- Vous pouvez en fixer un par défaut, et passer les autres dans la configuration de chaque zone s'il y a lieu.

```
Options {
        allow-transfer  { 127.0.0.1; };
};
```

Questions ???



Sous-domaine et délégation

- En principe simple : juste insérer les ERs NS pour le sous-domaine, pointant vers les serveurs autoritaires pour le sous-domaine
- Si vous faites attention, vous devriez en premier *vérifier* que les serveurs sont autoritaires pour les sous-domaines.
  - En utilisant "dig" sur tous les serveurs
  - Si le sous-domaine est mal géré, ceci n'est pas bon pour l'image du parent

Fichier de zone pour "example.com"

```
$TTL 1d
@   1h   IN   SOA     ns1.example.net.     Brian.nsrc.org. (
     2004030300      ; Serial
     8h   ;Refresh
     1h   ; Retry
     4w   ; expire
     1h ) ; Negative
     IN   NS  ns1.example.net.
     IN   NS  ns2.example.net.
     IN   NS  ns1.othernetwork.com.
; Les données de ma propre zone
     IN   MX 10  mailhost.example.net.
www  IN   A    212.74.112.80

; Sous domaine délégué
Subdom    IN   NS  ns1.othernet.net.
Subdom    IN   NS  ns2.othernet.net.
```

Cas de nécessité du glue record

- Mais et si "subdom.example.com" est délégué à ns1.subdom.example.com

- Quelqu'un qui est en cours de résolution de www.subdom.example.com doit d'abord résoudre ns1.subdom.example.com

- Mais il ne peut pas résoudre ns1.subdom.example.com sans résoudre en premier subdom.example.com!!!

Dans ce cas vous avez besoin de "glue record"

- L'enregistrement "glue record" est un ER de type A ou AAAA pour les noms de serveur placés hors de leurs zones autoritaires
- Exemple :

```
; zone .com
example          NS  ns.example.com.
                 NS      ns.othernet.net.
ns.example.com.     A    192.0.2.1    ; GLUE RECORD
```

Exemple où un « glue record » est nécessaire

```
; Les  données de ma propre zone
            IN   MX      10 mailhost.example.net.
www         IN   A       212.74.112.80

; Le sous-domaine délégué
Subdom              IN   NS  ns1.subdom          ; nécessaire
                    IN   NS  ns2.othernet.net.   ; non nécessaire
ns1.subdomIN   A    192.0.2.4
```

Le reverse

- Si vous avez au moins un /24 de l'espace d'adressage, alors votre fournisseur se chargera de la délégation à votre serveur de noms.
- Ex. Votre bloc réseau est 192.0.2.0/24
- Créer la zone 2.0.192.in-addr.arpa.
- Si vous avez plus qu'un /24 (Ex. Un /22) alors chaque /24 sera une zone séparée
- Si vous avez assez de chance d'avoir /16 alors il sera une zone unique.
  - 172.16.0.0/16 est 16.172.in-addr.arpa

Exemple : 192.0.2.0/24

```
Zone "2.0.292.in-addr.arpa" {
        type master:
        file "m/192.0.2";
        allow-transfer { . . . };
};

/etc/named/m/192.0.2
@IN     SOA         .......
        IN      NS      ns0.example.com
        IN      NS      ns0.otherwork.com.

1       IN      PTR         router-e0.example.com.
2       IN      PTR         ns0.example.com.
3       IN      PTR         mailhost.example.com.
4       IN      PTR         www.example.com.
; etc...
```

Comment fonctionne le DNS inverse?

- Ex. pour 192.0.2.4, l'hôte distant consultera 4.2.0.192.in-addr.arpa. (PTR)
- La requête suit l'arborescence de la délégation comme la normale. Si tout est correct, Il atteint vos serveurs et aura la réponse.
- Les octets sont placés dans l'ordre inversé
  - Poids plus faible en premier.
- Le propriétaire du grand bloc réseau (192/8) peut déléguer le DNS inverse dans de gros morceaux de /16. le propriétaire d'un /16 peut déléguer des /24

Les mêmes principes que pour le DNS normal

- Vous avez toujours besoin du maître et des esclaves
- Il ne fonctionnera pas à moins que vous obteniez la délégation du parent
- S'assurer que si vous avez un enregistrement PTR pour une adresse IP, le nom peut se résoudre à la même adresse IP
  - Ceci n'est pas obligatoire, mais certains sites l'utilise comme critère de filtrage.

Que faire si vous avez moins d'un /24 ?

- Le DNS inverse pour le /24 est délégué à votre fournisseur
- **Option 1:** demandez à votre fournisseur d'insérer les ERs PTR pour votre bloc dans la zone du /24.
  - **Problème**: vous devez leur demander à chaque fois que vous voulez faire un changement
- **Option 2** : Suivez la procédure décrite dans le RFC2317
  - Utilisez l'astuce avec le CNAME pour rediriger les requêtes PTR pour vos adresses IP vers vos serveurs de noms.

IDN/IDNA (RFC3490)

- Internationalized Domain Names (IDN)
- Internationalizing Domain Names in Applications (IDNA)
- IDN utilise les caractères unicode / or DNS => ASCII
- IDNA permet de représenter les caractères non-ascii utilisés dans IDN en ASCII (déjà autorisé dans les noms d'hôtes)
- IDNA sert uniquement pour le traitement des noms de domaine

IDN/IDNA (RFC3490)

- N'implique aucun changement sur les serveurs DNS et les resolvers
- IDNA demande juste les mises à jour des applications clientes
- IDNA résout un grand problème : l'augmentation du nombre de caractères pouvant être utilisé dans le nommage DNS.
- 2 opérations :
  - ToASCII : avant d'envoyer vers quelque chose qui attend de l'ASCII (resolver ou fichier de zone)
  - ToUnicode : avant d'afficher à l'utilisateur

Petit test

- Installez le paquet ***idn***
- Puis tapez la commande : ***idn tèst***
- Dans votre fichier de zone, faîtes un CNAME de la chaîne de caractère obtenu avec le nom de votre site web

  xn--tst-6la IN CNAME www
- Lancer votre navigateur et taper tèst.votredomaine dans la barre d'adresse, et voyez si cela marche ou pas

Questions ???