DNS Module 1: Les Fondamentaux

Basé sur un document de Brian Candler
Revu et traduit par Alain Patrick AINA
Atelier CCTLD ISOC

Les ordinateurs utilisent les adresses IP. Pourquoi a-t-on besoin de noms?

- Faciles à mémoriser par les humains
- Les ordinateurs peuvent changer de réseau, et donc d'adresses IP

Ancienne solution: hosts.txt

- Un fichier centralisé distribué à toutes les machines sur l'Internet
- Cette fonctionalité existe toujours
  - `/etc/hosts` [Unix]
  - `c:\windows\system32\drivers\etc\hosts` [Windows]

```
128.4.13.9          SPARKY
4.98.133.7          UCB-MAILHOST
200.10.194.33       FTPHOST
```

hosts.txt est inadapté à grande échelle

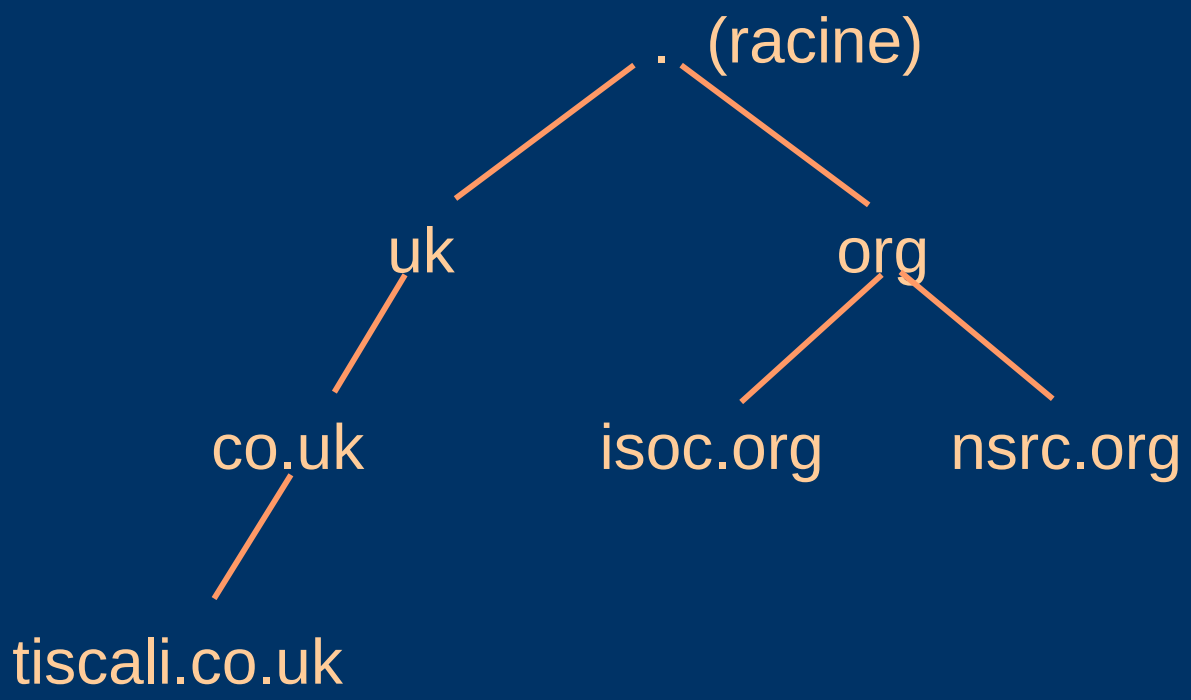
- ✗ Fichier volumineux
- ✗ Nécessite d'être copié fréquemment sur toutes les machines
- ✗ Pas uniforme
- ✗ Toujours dépassé
- ✗ Pas d'unicité des noms
- ✗ Un seul point d' administration

Le système de nom de domaine était né

- DNS est une base de donnée distribuée pour faire correspondre des noms aux adresses IP(et autres informations)
- Distribuée:
  - Administration partagée
  - Charge partagée
- Robustesse et performance à travers:
  - La duplication
  - Le cache
- Une pièce critique de l'infrastructure Internet

DNS est Hiérarchisé

- Forme la structure d'un arbre



DNS est Hiérarchisé (2)

- Donne des noms globalement uniques
- Administré en "zones" (parties de l'arbre)
- Vous pouvez donner ("déléguer") le contrôle d'une partie de l'arbre sous votre autorité
- Exemple:
  - isoc.org est sur un ensemble de serveurs
  - dnsws.isoc.org sur un ensemble différent
  - foobar.dnsws.isoc.org sur un autre ensemble

Les noms de domaines sont (presque) illimités

- Une longueur totale de 255 caractères maximum
- 63 caractères maximum dans chaque partie
  - RFC 1034, RFC 1035
- Si un nom de domaine est utilisé comme nom de machine, vous devez respecter certaines restrictions
  - RFC 952 (vieux!)
  - a-z 0-9 et tiret (-) seulement
  - Pas d'underscores (_)

Utilisation du DNS

- Un nom de domaine (comme www.tiscali.co.uk) est une clé de recherche d'informations
- Le resultat est un ou plusieurs enregistrements de ressources (ER)
- Il y a différents ER pour différents types d'informations
- Vous pouvez rechercher un type spécifique, ou rechercher "tous" les ER associés à un nom de domaine

ER courants

- A (adresse IP): associe les noms aux adresses IP
- PTR (pointer): associe les adresses IP aux noms
- MX (mail exchanger): où délivrer les courriers pour utilisateur@*domaine*
- CNAME (canonical name):associe des alias au nom réel
- TXT (text):n'importe quel texte descriptif
- NS (Name Server), SOA (Start Of Authority): Utilisés pour les délégations et le fonctionnement du DNS

Exemple simple

- Requête: www.tiscali.co.uk
- Type de requête : A
- Resultat:

```
www.tiscali.co.uk.  IN   A   212.74.101.10
```

- Dans ce cas, un seul ER a été trouvé, mais en général, plusieurs ER peuvent être retournés
  - IN est la "classe" pour INTERNET, utilisée par le DNS

Résultats possibles

- Positif
  - 1 ou plusieurs EE trouvés
- Négatif
  - Définitivement aucun ER ne correspond à la requête
  - Définitivement le nom recherché n'existe pas
- Echec de serveur
  - Ne peut contacter "quelqu'un" qui connait la réponse

Comment utiliser une adresse IP comme clé pour une requête DNS?

- Convertir l'adresse IP au format décimal(A.B.C.D)
- Inverser les quatre parties
- Ajouter ".in-addr.arpa" à la fin (domaine spécial réservé à cette fin)
- e.g. Pour trouver le nom de 212.74.101.10

```
10.101.74.212.in-addr.arpa.
                → PTR www.tiscali.co.uk.
```

  - Connue comme "une requête DNS inverse"
    - Parce que nous cherchons le nom pour une adresse IP, au lieu d'une adresse IP pour un nom

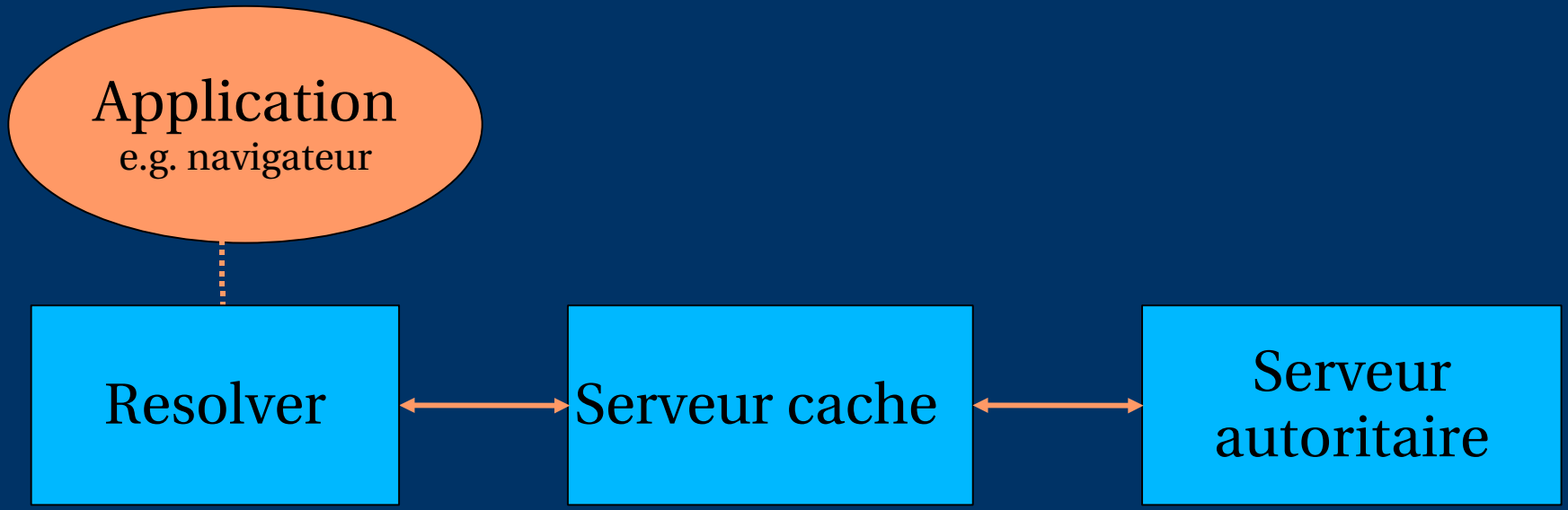
Questions?

?

Le DNS est une application Client-Serveur

- (Naturellement – Il tourne sur un réseau)
- Requêtes et réponses sont normalement envoyées dans des paquets UDP, port 53
- Utilise occasionnellement TCP, port 53
  - Pour les transferts de zones du maître aux esclaves et pour les grandes requêtes, e.g. > 512 octets

Il y a trois rôles dans le DNS



Trois rôles dans le DNS

- *RESOLVER*
  - Prend la requête de l'application, le formate dans un paquet UDP, et l' envoie au serveur cache
- *Serveur CACHE*
  - Renvoie la réponse si déja connue
  - Dans le cas contraire, recherche le serveur autoritaire qui a l'information
  - Met le resultat en cache pour les requêtes futures
  - Egalement connu comme serveur récursif
- *SERVEUR AUTORITAIRE*
  - Contient les informations actuelles placées dans le DNS par le propriétaire du domaine

Trois rôles dans le DNS (2)

- Le même protocole est utilisé pour les communications resolver↔cache and cache↔NS autoritaire
- Il est possible de configurer un serveur de nom comme cache et autoritaire en même temps
- Mais il joue un seul rôle pour chaque requête entrante
- Classique, mais “NON RECOMMANDE” (voir plus loin)

ROLE 1: LE RESOLVER

- Un morceau de logiciel qui formate une requête DNS dans un paquet UDP, l'envoie au serveur cache et décode le resultat
- Généralement une librairie partagée (e.g. `libresolv.so` sous Unix) parce que beaucoup d'applications en ont besoin
- Chaque machine a besoin d'un resolver - e.g. Chaque poste de travail windows a en un

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

- Doit être configuré explicitement (statique, ou via DHCP etc)
- Doit être configuré avec l'adresse IP du cache (pourquoi pas le nom?)
- C'est une bonne idée de configurer plus d'un cache, dans le cas où le premier n'est pas disponible

Comment choisir quel cache utilisé?

- Vous devez avoir la permission de l'utiliser
  - e.g. Cache de votre FAI, ou le vôtre
- Préférer un cache proche
  - Minimise les temps d'aller-retour et les pertes de paquets
  - Peut réduire le trafic sur votre lien externe, puisque le cache peut souvent répondre sans contacter d'autres serveurs
- Préférer un cache fiable
  - Pouvez -vous avoir un mieux que celui de votre FAI ?

“Resolver” peut être configuré avec des domaines par défaut

- Si"foo.bar" échoue, alors éssayer la requête avec "foo.bar.mydomain.com"
- Peut faire économiser des saisies, mais ajoute de la confusion
- Peut générer du trafic extra non nécessaire
- A éviter dans le meilleur des cas

Exemple: Configuration d'un resolver unix

- `/etc/resolv.conf`

```
Search cctld.sn
nameserver 196.216.0.21
```

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

Tests DNS

- Mettre simplement "www.yahoo.com" dans un navigateur
- Pourquoi ceci n’est pas un bon test?

Tests DNS avec "dig"

- "dig" est un programme qui fait simplement des requêtes DNS et affiche les résultats
  - Mieux que "nslookup" et "host" pour le débogage, parce qu'il montre les messges DNS au complet

```
dig tiscali.co.uk.
   - Par défaut recherche le type "A"
dig tiscali.co.uk. mx
   - specifier le type recherché
dig @212.74.112.66 tiscali.co.uk. mx
   - Envoie la requête à un cache spécifique
     (outrepasse /etc/resolv.conf)
```

Le point final

```
dig tiscali.co.uk.
```

- Empêche l'ajout des domaines par défaut
- Prenez l'habitude de l'utiliser pendant les tests DNS
  - Mais uniquement sur les noms de domaine, pas sur sur les adresses IP ou les adresses électroniques

```
# dig www.gouv.bj. a
; <<>> DiG 9.3.0 <<>> www.gouv.bj a
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2462
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 4
;; 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     rip.psg.com.
gouv.bj.                86400   IN   NS     ben02.gouv.bj.
gouv.bj.                86400   IN   NS     nakayo.leland.bj.
gouv.bj.                86400   IN   NS     ns1.intnet.bj.

;; ADDITIONAL SECTION:
ben02.gouv.bj.          86400   IN   A      81.91.232.1
nakayo.leland.bj.       18205   IN   A      81.91.225.1
ns1.intnet.bj.          18205   IN   A      81.91.225.18
rip.psg.com.            160785  IN   A      147.28.0.39

;; Query time: 200 msec
;; SERVER: 212.74.112.67#53(212.74.112.67)
;; WHEN: Tue Dec 28 19:50:01 2004
;; MSG SIZE  rcvd: 237
```

Interprétation des resultats: en-tête

- STATUS
  - NOERROR: 0 ou plusieurs ER returnés
  - NXDOMAIN: Domaine non-existent
  - SERVFAIL: cache ne peut trouver la réponse
- FLAGS
  - AA: Authoritative answer(réponse autoritaire)
  - Vous pouvez ignorer les autres
    - QR: Requête or Réponse (1 = Réponse)
    - RD: Recursion Desired(récursion désirée)
    - RA: Recursion Available(récursion disponible)
- ANSWER: nombre de ER dans la réponse

Interprétation des resultats

- Answer section (ER recherchés)
  - Chaque enregistrement à un TTL (Time To Live)
  - Qui indique pendant combien de temps le cache peut le gardé
- Authority section
  - Les NS autoritaires pour ce domaine
- Additional section
  - Plus de ERs (généralement des adresses IP des NS autoriatires)
- Total query time
- Vérifier quel serveur a donné le resultat!
  - En cas d'erreur de frappe, la requête peut aller au serveur par défaut

Travaux pratiques

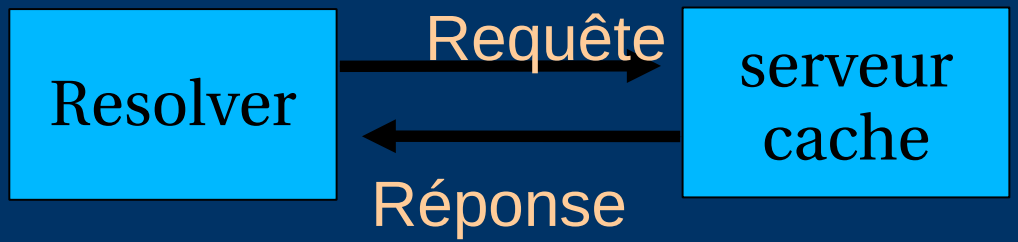
- dig
- Téléchargement et installation de BIND
- Configuration de rndc

DNS Session 2: Cache DNS
Fonctionnement et débogage du DNS

Basé sur un document de Brian Candler
Revu et traduit par Alain Patrick AINA
Atelier CCTLD ISOC

Comme fonctionne un cache DNS(1)

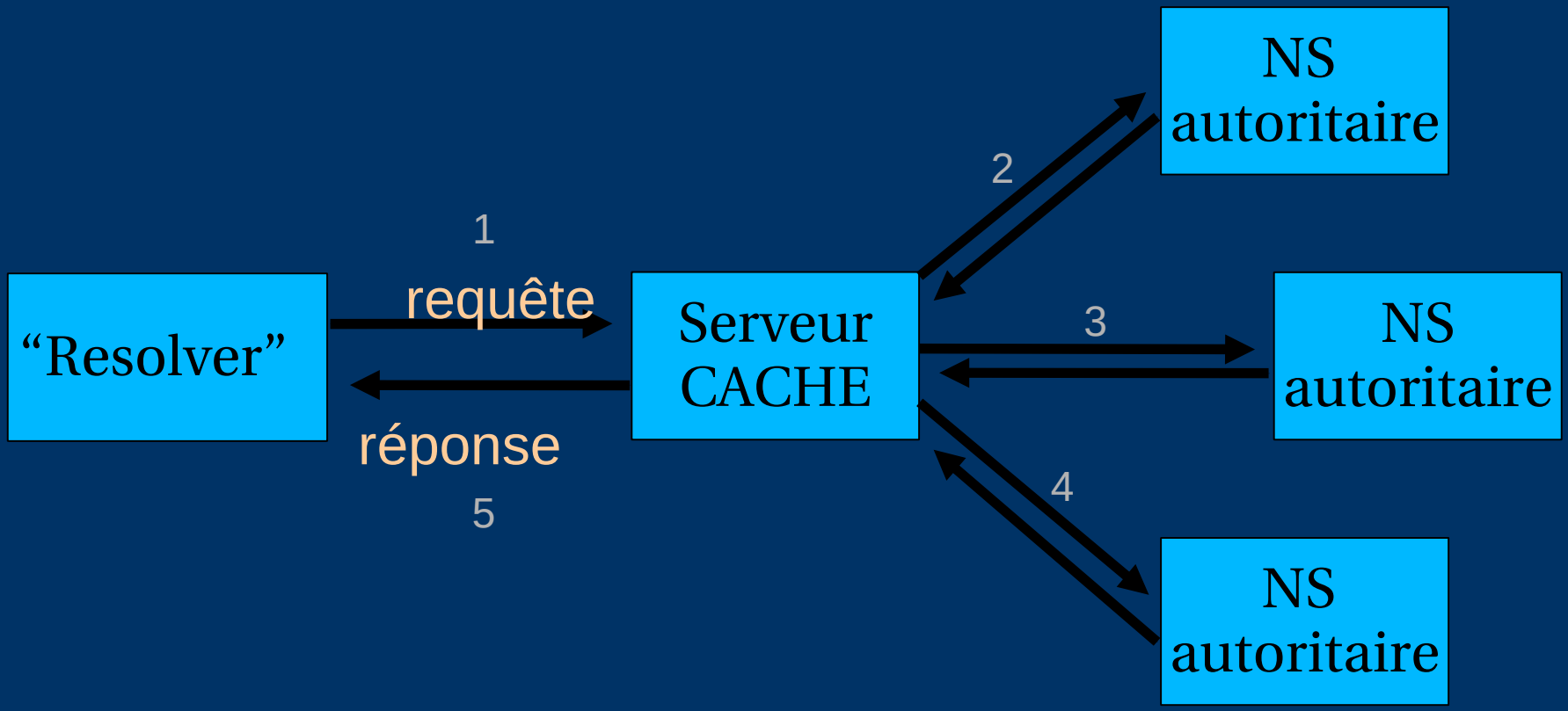
- Si nous avons géré cette requête récemment, la réponse est déjà dans le cache - facile!



Que se passe-t-il si la réponse n'est pas le cache?

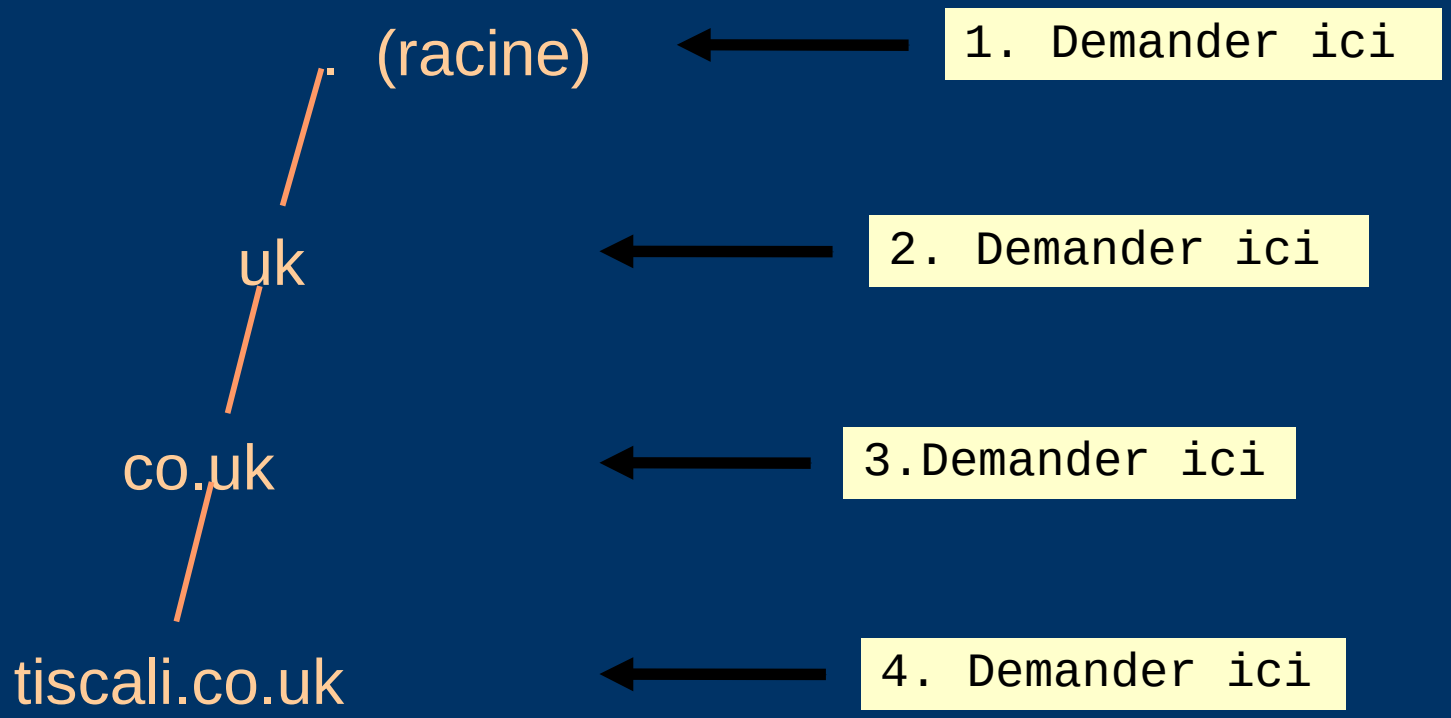
- Le DNS est une base de donnée distribuée : Des parties de l'arborescence (appelée "zones") sont tenues par des serveurs différents
- Ils sont appelés "autoritaires" pour leur part de l'arbre
- C'est le travail du serveur cache de trouver le bon serveur autoritaire et obtenir le resultat
- Il peut avoir besoin de demander à d'autres serveurs de trouver d'abord celui dont il a besoin

Comme fonctionne un cache DNS (2)



Comment trouve-t-il le NS autoritaire à qui demander ?

- Il suit la structure hiérarchique de l'arbre
- e.g. Pour trouver "www.tiscali.co.uk"



Les serveurs intermédiaires renvoient les enregistrements de ressources "NS"

- "Je n’ai pas la réponse, mais essaye ces autres NS ."
- Appelé une référence
- Vous fait descendre l’arbre d’un ou plusieurs niveaux

Ce processus pourra éventuellement soit:

- Trouver un serveur autoritaire qui connait la réponse (positive ou négative)
- Ne pas trouver un serveur fonctionnel: *SERVFAIL*
- Se terminer sur un serveur fautif – pas de réponse et pas de référence, ou fausse réponse!
    - NB: Il peut arriver qu'un serveur cache soit également autoritaire pour une requête particulière. Dans ce cas, il répond immédiatement sans chercher ailleurs. Nous verrons plus tard, pourquoi il faut avoir des machines séparées pour le cache et les serveurs autoritaires

Comment commence ce processus?

- Chaque serveur cache est doté d'une liste de serveurs racines

/usr/local/etc/named.conf

```
zone "." {
        type hint;
        file "named.root";
}
```

named.root

```
.                       3600000      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

.                       3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.     3600000      A     192.33.4.12
;... etc
```

D'où vient le fichier named.root ?

- ftp://ftp.internic.net/domain/named.cache
- Vérifier pour des mises à jour environ chaque semestre

Démonstration

- **`dig +trace www.tiscali.co.uk.`**
- Au lieu d'envoyer la requête au cache, "dig +trace" traverse l'arborescence depuis la racine et affiche les resultats obtenus
  - `dig +trace` est une fonctionnalité de bind 9
  - Utile pour des démo, mais pas pour le débogage

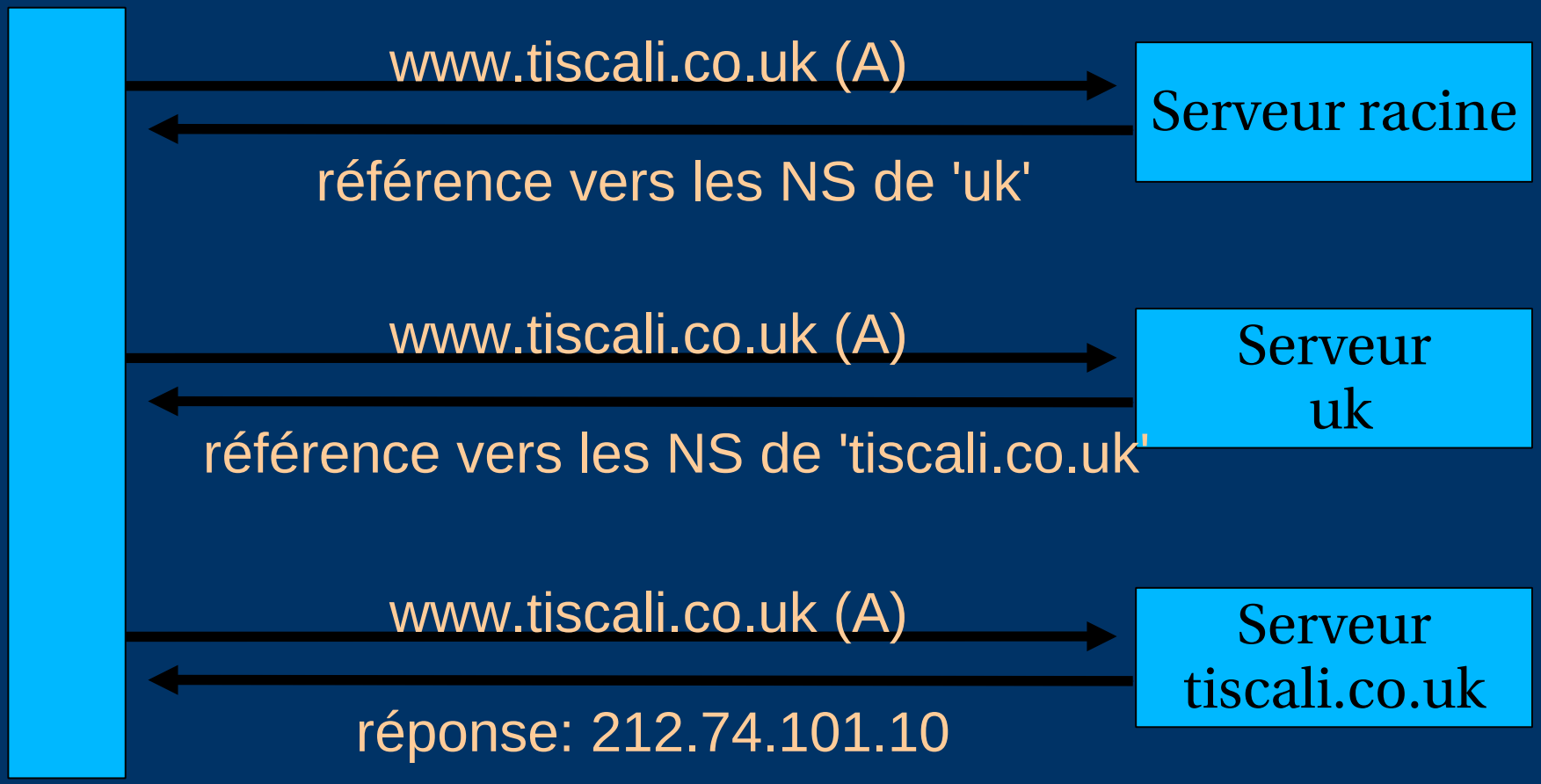
Les systèmes distribués ont plusieurs points de dysfonctionnement!

- Ainsi chaque zone a deux ou plusieurs serveurs autoritaires pour la redondance
- Ils sont tous équivalents et peuvent être essayés dans n'importe quel ordre
- Les essais s'arrêtent dès que l'un donne une réponse
- Aide également à partager la charge
- Les serveurs racines sont très occupés
  - Ils sont présentement 13 (Avec de nombreuse copies ANYCAST un peu partout dans le monde.)
  - http://www.root-servers.org

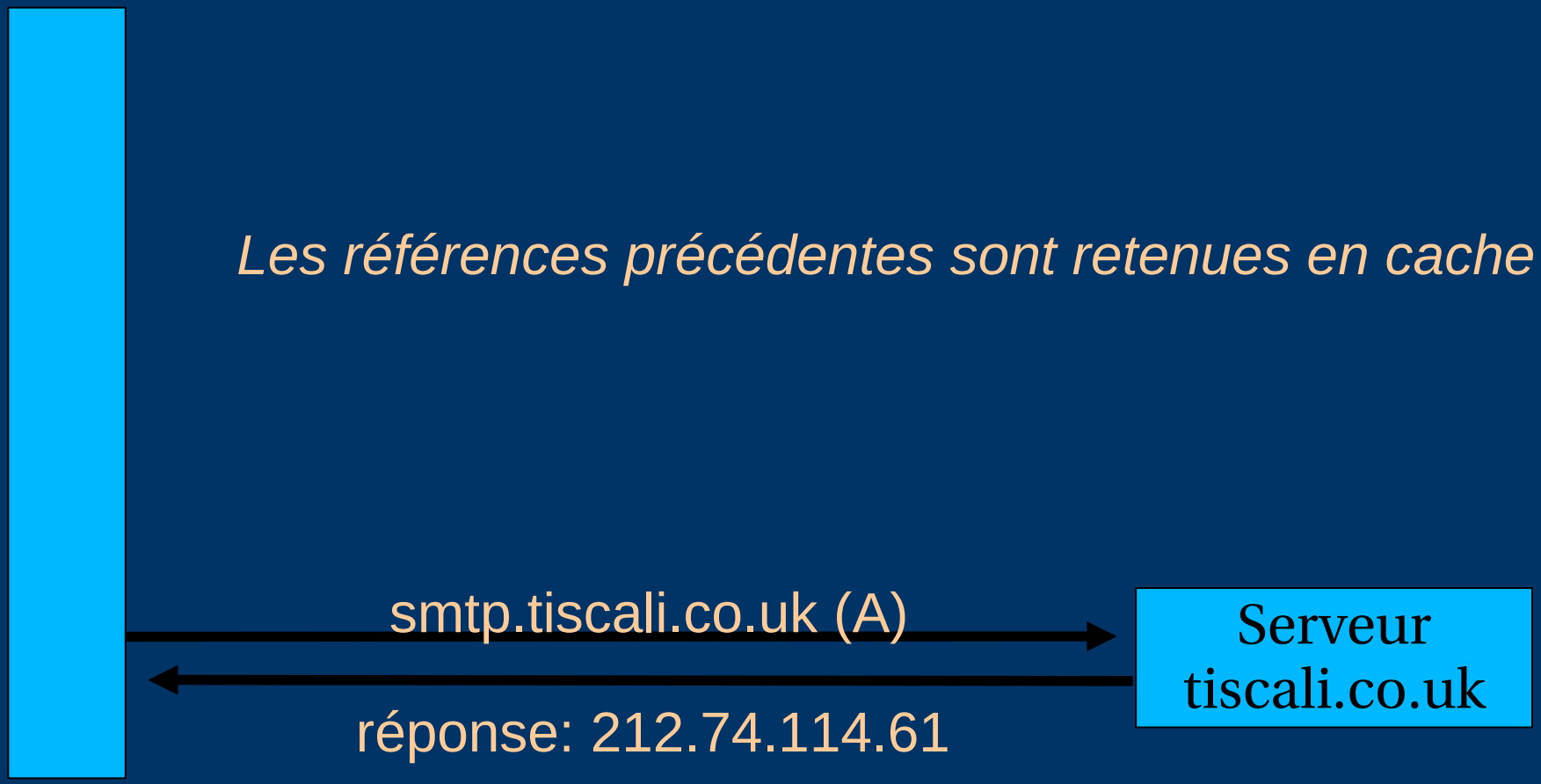
Le cache réduit la charge sur les serveurs autoritaires

- Spécialement important aux niveaux supérieurs: serveurs racines, serveurs TLD (GTLDs, ccTLDs, etc.....)
- Toutes les informations intermédiaires sont mises en cache comme la réponse finale -ainsi les enregistrements NS des références sont aussi mis en cache

Exemple 1: www.tiscali.co.uk (sur un cache vide)



Exemple 2: smtp.tiscali.co.uk (après l'exemple précédent



Les caches peuvent être un problème si les données vieillissent

- Si les caches gardent les données pendant un long moment, ils peuvent donner de fausses réponses si les données autoritaires changent
- Si les caches gardent les données pour un temps très court, ceci augmentera le travail des serveurs autoritaires

Les administrateurs de zone contrôlent comment les données sont mises en cache

- Chaque enregistrement de ressources(ER) a un TTL (Time To Live) qui indique pendant combien de temps il peut être gardé en cache
- L'enregistrement SOA indique pendant combien de temps une réponse négative peut être gardée en cache (i.e. La non-existence d'un nom ou d'un ER)
- NB: les administrateurs de cache n'ont aucun contrôle -mais dans tous les cas, ils n'en n'auraient pas besoin

Un compromis

- Définir un TTL relativement long - 1 ou 2 jours
  Quand vous vous préparez à faire des changements, reduire le TTL à 10 minutes
- Attendre 1 ou 2 jours AVANT de faire le changement
- Après le changement, ramener le TTL à la valeur initiale

Questions?

?

Quel genre de problèmes peuvent se produire pendant le résolution de noms?

- Se rappeler que suivre les références est en général un processus à plusieurs étapes
- Se rappeler du cache

(1) Un NS autoritaire est inopérationnel ou inaccessible

- Pas un problème: timeout and essayer le NS suivant
  - Se rappeler qu'il y a plusieurs NS autoritaires pour une zone, ainsi les références renvoient plusieurs ER NS

*(2) *Tous les* NS autoritaires sont inopérationels ou inaccessibles!*

- Ceci est mauvais; Les requêtes ne peuvent aboutir
- S'assurer que les NS ne sont pas sur le même sous-réseau (problème de switch/routeur)
- S'assurer que tous les NS ne sont pas le même bâtiment (problème d'électricité)
- S'assurer que tous les NS ne sont pas sur le même backbone Internet (problème de lien du fournisseur)
- Pour plus de détails, lire RFC 2182

(3) Référence vers un serveur qui n'est pas autoritaire pour la zone

- Mauvaise erreur. Appelée "Lame Delegation" (délégation douteuse)
- Les requêtes ne peuvent pas continuer – serveur ne pouvant donner ni la bonne réponse, ni la bonne délégation
- Erreur classique: Enregistrement NS pour une zone pointant vers un serveur cache non configuré en tant que autoritaire pour la zone
- Ou: une erreur de syntaxe dans le fichier de zone ayant conduit le serveur a ignoré la zone

(4) Contradictions entre NS autoritaires

- Si les NS autoritaires n'ont pas les mêmes informations, ils donneront différentes réponses en fonction de celui choisi (aléatoire)
- A cause du cache, ces problèmes peuvent être difficiles à deboguer. Problèmes intermittents.

(5) Contradictions dans les délégations

- Les enregistrements NS dans la délégation ne correspondent pas aux enregistrements NS dans le fichier de zone(nous écrirons les fichiers de zone plus loin)
- Problème: si les deux ne sont pas les mêmes, lequel est correct ?
  - Conduit à des comportements imprévisibles
  - Les caches pourraient utiliser un jeu ou l'autre, ou une union des deux

(6) Couplage cache et autoritaire

- Considérer les cas où un cache contient un ancien fichier de zone, alors que le client a transféré son DNS ailleurs
- Les caches répondent immédiatement avec les anciennes informations, bien que les NS pointent vers les serveurs autoritaires d' autre FAI qui tiennent les informations correctes!
- Ceci est une raison importante pour séparer les caches des autoritaires
    - Autre raison est que les serveurs uniquement autoritaires ont un besoin mémoire fixe

(7) Choix inapproprié des paramètres

- e.g. TTL défini trop court ou trop grand

Ces problèmes ne sont pas la faute des serveurs cache!

- Ils viennent de la mauvaise configuration des serveurs autoritaires
- Beaucoup de ces erreurs sont faciles à faire, mais difficile à deboguer, spécialement à cause du cache
- Faire fonctionner un serveur cache est facile; faire bien fonctionner un serveur autoritaire demande une plus grande attention au détail

Comment déboguer ces problèmes?

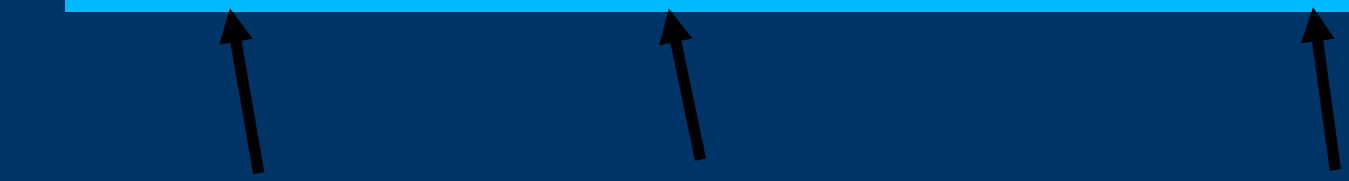
- Nous devons outrepasser le cache
- Nous devons essayer **tous les** N serveurs pour une zone (un serveur cache s'arrête après un)
- Nous devons outrepasser la récursion pour tester toutes les références intermédiaires
- "dig +norec" est votre ami

```
dig +norec @1.2.3.4 foo.bar. a
```

Comment interpréter les réponses (1)

- Rechercher le "status: NOERROR"
- "flags ... **aa**" indique une réponse autoritaire (pas d'un cache)
- "ANSWER SECTION" donne la réponse
- Si vous n'avez que des NS: c'est une référence

```
;; ANSWER SECTION
foo.bar.      3600   IN   A   1.2.3.4
```



Nom de domaine TTL Réponse

Comment interprèter les réponses (2)

- "status: NXDOMAIN"
  - OK, négatif (le domaine n'existe pas). Vous devez avoir en retour un SOA
- "status: NOERROR" avec zéro ER
  - OK, négatif (Le domaine existe, mais pas de ER du type recherché). Vous devez avoir en retour un SOA
- d'autres status peuvent indiquer des erreurs
- Rechercher également le "status : *Refused*" (Le serveur DNS rejette vos requêtes) ou "*Timeout*" (pas de réponse)

Comment déboguer un domaine avec "dig +norec" (1)

1. Partir d'un serveur racine: [a-m].root-servers.net.

```
dig +norec @a.root-servers.net. www.tiscali.co.uk. a
```

Attention aux points à la fin des noms!

   1. Pour une référence, noter les NS obtenus
   2. Répéter la requêtes pour **tous les* NS
   3. Retourner à l'étape 2, jusqu'à l'obtention de la réponse finale

Comment déboguer un domaine avec "dig +norec" (2)

1. Vérifier que les résultats d'un groupe de serveurs autoritaires correspondent l'un à l'autre
2. Vérifier que toutes les réponses finales ont le "flags: aa"
3. Noter que les NS pointent vers des noms, et non des adresses IP. Ainsi il faut vérifier que chaque enregistrement NS vu correspond à la bonne adresse IP en utilisant le même processus!!

Comment déboguer un domaine avec "dig +norec" (3)

- Fastidieux, demande de la patience et de l'exactitude, mais paye
- Apprendre ceci avant de commencer à s'amuser avec les outils automatisés
  - Comme:
    - http://www.squish.net/dnscheck/
    - http://dnsecheck.se/
  - Tous ont des limites, aucun n'est parfait

Travaux pratiques

Débogage avec dig

Débogage de domaine avec

http://www.squish.net/dnscheck/
http://dnsecheck.se/

DNS module 3: Configuration de services de nom autoritaires

Basé sur un document de Brian Candler
Revu et traduit par Alain Patrick AINA
Atelier CCTLD ISOC

Récap

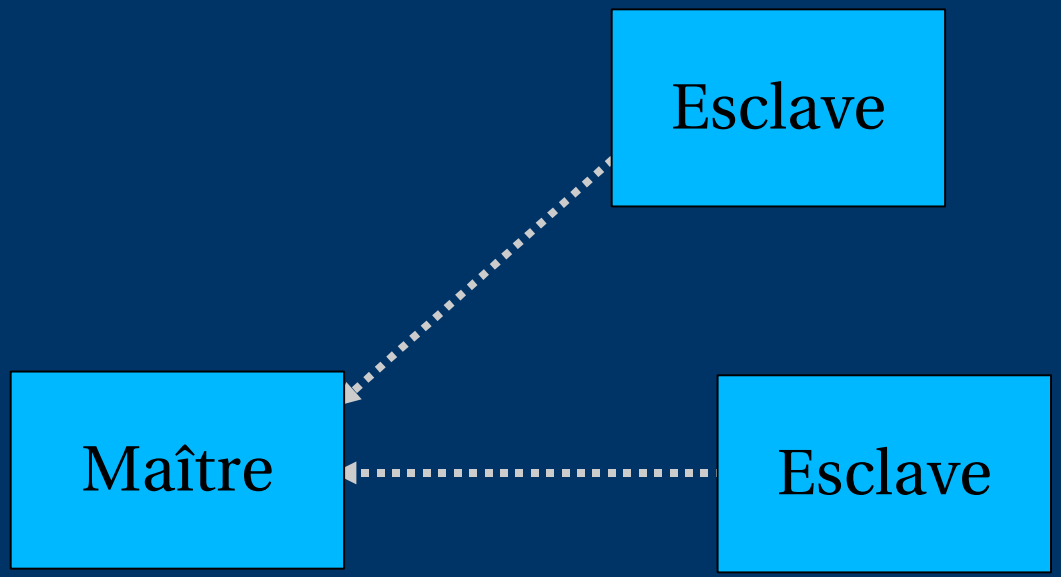
- Le DNS est une base de donnée distribuée
- Le“Resolver” demande des informations au cache
- Le cache traverse l’arborescence du DNS pour trouver quel serveur autoritaire a l’information recherchée
- La mauvaise configuration des serveurs autoritaires peuvent conduire à des domaines défectueux

Duplication DNS

- Pour chaque domaine, nous avons besoin de plus d'un serveur autoritaire avec les mêmes informations (RFC 2182)
- Les données sont enregistrées sur un serveur (Maître) et copiées vers les autres(esclave(s))
- Le monde extérieur ne peuvent pas dire la différence entre maître et esclave
  - Les enregitrements NS sont retournés dans un ordre aléatoire pour une répartition de charge équitable
- Communément appelés "primaire" et "secondarire"

Les esclaves se connectent au maître pour copier les données de la zone

- Le maître “n’envoie” pas les données aux esclaves



Quand a lieu la duplication?

- Les esclavent scrutent le maître périodiquement – appelé “intervalle de rafraîchement" - pour vérifier s’il y a de nouvelles données
  - Seul mécanisme au départ
- Avec les nouveaux logiciels, le maître peut informer les esclaves si les données ont changé (notify)
  - Des mises à jour plus rapides
- Cette notification n’est pas fiable (e.g. Le réseau peut perdre un paquet).Ainsi nous avons toujours besoin de vérifier à “l’intervalle de rafraîchement”

Numéro de série

- Chaque zone a un numéro de série
- Les esclaves copient les données uniquement si le numéro de série a augmenté
  - Requête UDP périodique pour vérifier le numéro de série
  - s'il a augmenté, un transfert TCP des données de la zone
- C'est votre responsabilité d'augmenter le numéro de série après chaque changement, sinon esclaves et maîtres seront incohérents

Format de numéro de série recommandé: YYYYMMDDNN

- YYYY = année
- MM = mois (01-12)
- DD = jour(01-31)
- NN = Nombre de changements par jour (00-99)
  - e.g. Si vous changez le fichier le 5 Mars 2004, le numéro de série sera 2004030500. Si vous le changez de nouveau le même jour, il sera be 2004030501.

Numéro de série: Danger 1

- Si vous dimunuer le numéro de série, les esclaves ne se mettront plus à jour jusqu'à ce que le numéro de série dépasse sa valeur précédente
- RFC1912 section 3.1 explique une méthode pour corriger ce problème
- Au pire des cas, vous pouvez contacter tous vos esclaves et leur faire supprimer leur copie de la zone

Numéro de série: Danger 2

- Le numéro de série est un nombre unsigned sur 32bits
- Intervalle: 0 à 4,294,967,295
- Toute valeur plus grande que ceci est silencieusement tronquée
- e.g. 20040305000 (noter le digit extra)
  = 4AA7EC968 (hex)
  = AA7EC968 (32 bits)
  = 2860435816
- Si vous faites cette erreur et la corriger plus tard, le numéro de série aura dimunié

Configuration du Maître

- /usr/local/etc/named.conf pointe vers les fichiers de zone (créés manuellement) contenants vos ER
- Choisir une place logique pour les garder
  - e.g. /var/cctld/master/cctld.sn
  - or /var/cctld/master/sn.cctld

```
zone "example.com" {
        type master;
        file "/var/cctld/master/example.com";
        allow-transfer { 192.188.58.126;
                         192.188.58.2; };
};
```

Configuration d'esclave

- named.conf pointe vers l'adresse IP du maître et où créer le fichier de zone
- Les fichiers de zone sont transférés automatiquement
- Ne pas les toucher!

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

Maître et esclave

- Un serveur peut bien être maître pour certaines zones et esclave pour d'autres
- C'est pourquoi, nous recommandons de garder les fichiers dans des répertoires différents
  - /var/cctld/master/
  - /var/cctld/slave/
    - (également, le répertoire esclave doit avoir les droits appropriés pour permettre au démon de créer des fichiers)

allow-transfer { ... }

- Les machines distantes peuvent demander un transfert de tout le contenu de la zone
- Par défaut, ceci est autorisé à n'importe qui
- Mieux vaut le restreindre
- Vous pouvez définir une option par défaut globale et le redéfinir pour chaque zone si nécessaire

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

Structure d'un fichier de zone

- Options globales
  - $TTL 1d
  - Définit le TTL par défaut pour tous lesER
- ER SOA
  - "Start Of Authority"
  - Information de maintenance de la zone
- ER NS
  - Liste tous les serveurs autoritaires pour cette zone, maître et esclaves
- Autres ER
  - Les données actuelles que vous voulez publier

Format d'un ER

```
www        3600   IN        A         212.74.112.80
Domaine    TTL    Classe    Type      Donnée
```

- Un par ligne (sauf pour le SOA qui peut couvrir plusieurs lignes)
- Si vous omettez le nom de domaine, c'est le même qu'à la ligne précédente
- Raccourcis de TTL : e.g. 60s, 30m, 4h, 1w2d
- Si vous omettez le TTL, utiliser la valeur de $TTL
- Si vous omettez la classe, se réfèrer à la classe du SOA
- Type et donnée ne peuvent pas être omis
- Les commentaires commencent avec point-virgule (;)

Raccourcis

- Si un nom de domaine ne se termine pas par un point, le domaine de la zone("origine") est ajouté
- "@"signifie l’ origine elle-même
- e.g. dans le fichier de zone pour example.com:
  - `@` signifie `example.com.`
  - `www` *signifie* `www.example.com.`

Si vous écrivez ceci ...

```
$TTL 1d
@                               SOA ( ... )
                                NS  ns0
                                NS  ns0.as9105.net.

www                             A   212.74.112.80
                                MX  10 mail
```

... il devient ceci

```
example.com.          86400  IN  SOA ( ... )
example.com.          86400  IN  NS  ns0.example.com.
example.com.          86400  IN  NS  ns0.as9105.net.
www.example.com.      86400  IN  A   212.74.112.80
www.example.com.      86400  IN  MX  10 mail.example.com.
```

Format de l'enregistrement SOA

```
$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.
```

Format de l'enregistrement SOA

- `ns1.example.net.`
  - Nom du serveur maître
- `brian.nsrc.org.`
  - Adresse E-mail de la personne responsable, avec"@" changé en point, et le point à la fin
- Numéro de série
- Intervalle de rafraichissement
  - A quel rythme les esclaves doivent scruter le numéro de série sur le maître
- Intervalle de nouvelle tentative
  - A quel rythme les esclaves doivent essayer de contacter de nouveau le maître si la tentative précédente a échoué

Format de l'enregistrement SOA (suite)

- Temps d'expiration
  - Si l'esclave n'arrive pas à contacter le maître pour cette période de temps, il supprime sa copie des données de la zone
- Négatif / Minimum
  - Les vieux logiciels utilisent ceci comme valeur minimale de TTL
  - Il est maintenant utilisé pour le cache négatif: pendant combien de temps un cahe peut garder la non non-existence d'un nom ou d'un ER
- RIPE-203 a des valeurs recommandées
  - http://www.ripe.net/ripe/docs/dns-soa.html

Format des enregitrements NS

- Liste tous les serveurs autoritaires pour une zone – maître et esclave(s)
- Doit pointer vers des noms et non des adresses IP

```
$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.
```

Format des autres ER

- `IN A 1.2.3.4`
- `IN MX 10 mailhost.example.com.`
  - Le nombre est " valeur de préférence ". les messages seront d'abord délivrés au MX ayant la préférence la plus faible
  - Doit pointer vers des noms et non des adresses IP
- `IN CNAME host.example.com.`
- `IN PTR host.example.com.`
- `IN TXT "n'mporte quel texte de votre choix"`

Quand vous changez des données dans un fichier de zone:

- Se rappeler d'augmenter le numéro de série!
- `named-checkzone example.com \`
  `/var/cctld/master/example.com`
  - Fonctionalité de bind 9
  - Rapporte les erreurs de syntaxe; corrigez les!
- `named-checkconf`
  - Rapporte les erreurs dans named.conf
- `rndc reload`
  - ou: `rndc reload example.com`
- `tail /var/log/messages`

Ces vérifications sont essentielles

- Si vous avez une erreur dans named.conf ou dans un fichier de zone, named peut continuer à fonctionner sans prendre en compte les mauvaises zone(s)
- Vous serez douteux(lame) pour la zone sans s'en rendre compte
- Les esclaves ne pourront pas contacter le maître
- Eventuellement (e.g. 4 semaines plus tard) les esclaves vont faire expirer la zone
- Votre domaine va s'arrêter de fonctionner

Autres vérifications que vous pouvez faire

- `dig +norec @x.x.x.x example.com. soa`
  - Vérifier si la réponse est autoritaire
  - Répéter pour le maître et tous les esclaves
  - Vérifier si les numéros de série sont les mêmes
- `dig @x.x.x.x example.com. axfr`
  - " Transfert Autoritaire"
  - Demande une copie de toute la zone par TCP, comme le font les esclaves
  - Ceci ne marchera que si vous remplisssez les conditions listées par la section allow-transfer {...}

Ainsi, vous avez des serveurs autoritaires opérationnels!

- Mais aucun d'eux ne fonctionnera jusqu'à l'obtention de la délégation du domaine parent
- Ceci est fait en plaçant les enregistrements NS pour votre domaine pointant vers vos serveurs autoritaires
- Vous devez aussi mettre les enregistrements NS dans le fichier de zone
- Les deux ensembles doivent être les mêmes

Questions?

?

Les 10 plus importantes erreurs avec les serveurs autoritaires

- Tous les opérateurs de serveurs autoritaires doivent lire le RFC 1912
  - Les erreurs courantes de configuration et de fonctionnement du DNS
- Et aussi le RFC 2182
  - Sélection et fonctionnement de serveurs DNS secondaires

1. Les erreurs du numéro de série

- Oublier d'augmenter le numéro de série
- Incrementer le numéro le numéro de série, puis le décrémenter
- Utiliser un numéro de série supérieur à 2^{32}
- Impact:
  - Les esclaves ne se mettent pas à jour
  - Maître et esclaves n'ont pas les mêmes données
  - Les caches auront tantôt la nouvelle donnée, tantôt l'ancienne donnée – problème intermittent

2. Commentaires dans fichier de zone commencant par '#' au lieu de ';'

- Erreur de syntaxe dans le fichier de zone
- Maître plus autoritaire pour la zone
- Les esclaves ne peuvent plus vérifier le SOA
- Les esclaves vont eventuellemnt faire expirer la zone, et votre domaine s’arrête de fonctionner entirement
- Utiliser "named-checkzone"
- Utiliser"tail /var/log/messages"

3. D’autres erreurs de syntaxe dans le fichier de zone

- e.g. Omission de la valeur de préférence de l’enregistrement MX
- Même impact

4. Point final manquant

```
; zone example.com.
@  IN  MX 10  mailhost.example.com
```

devient

```
@  IN  MX 10  mailhost.example.com.example.com.
```

```
; zone 2.0.192.in-addr.arpa.
1  IN  PTR    host.example.com
```

devient

```
1  IN  PTR    host.example.com.2.0.192.in-addr.arpa.
```

5. NS ou MX pointant vers des adresses IP

- Ils doivent pointer vers des noms, non vers des adresses IP
- Malheureusement, quelques serveurs de messagerie acceptent les addresses IP dans les enregistrements MX . Ainsi vous pouvez ne pas avoir des problèmes avec tous les sites distants

6. Les esclaves ne peuvent pas transférer le zone du maître

- Restriction des accès par allow-transfer {...} et les esclaves n'y sont pas lister
- Ou filtres mal configurés
- Les esclaves seront non autoritaire (douteux: lame)

7. Délégation douteuse

- Vous ne pouvez pas lister n'importe quel serveur dans les enregistrements NS de votre domaine
- Vous devez avoir un accord avec les administrateurs des serveurs, et ils doivent se configurer en tant que esclaves pour la zone
- Au Mieux: Résolutions DNS lentes et manque de redondance
- Au pire: échecs intermittents dans la résolution de votre domaine

8. Pas de délégation du tout

- Vous pouvez configurer "example.com" sur vos serveurs, mais le monde extérieur ne leur enverra pas de requêtes jusqu'à l'obtention de la délégation
- Le problème est caché si votre serveur jouele rôle de serveur cache et de serveur autoritaire
- Votre propres clients peuvent résoudre ww.example.com, mais le reste du monde ne peut pas

9. Enregistrement “glue” dépassé

- Voir plus loin

10. Mauvaise gestion du TTL pendant les changements

- e.g. Si vous avez un TTL de 24 heures, et vous faites pointer www.example.com vers un nouveau serveur, alors il y aura une période de temps ou certains utilisateurs vont atteindre le nouveau et d'autres l'ancien
- Suivre la procédure:
  - Réduire le TTL à 10 minutes
  - Attendre au moins 24 heures
  - Faire le changement
  - Remettre le TTL à 24 heures

Travaux pratiques

- Création de nouveaux domaines
- Configuration de serveurs maître et esclave
- Demande de délégation du parent
- Tests

DNS module 4: Délégation

Basé sur un document de Brian Candler
Revu et traduit par Alain Patrick AINA
Atelier CCTLD ISOC

Comment déléguer un sous-domaine?

- En principle simple: insérer simplement les enregistrements NS du sous-domaine pointants vers les serveurs d'autrui
- Si vous êtes pointilleux, vous devrez d'abord vérifier que ces serveurs sont autoritaires pour le sous-domaine
  - avec "dig +norec" sur tous les serveurs
- Si le sous-domaine est mal géré, il vous donne une mauvaise image!
  - Et vous ne voulez pas être comptable pour des problèmes qui sont ailleurs

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.

; vos propres données
       IN  MX  10  mailhost.example.net.
www    IN  A   212.74.112.80

; un sous-domaine délégué
subdom IN  NS  ns1.othernet.net.
       IN  NS  ns2.othernet.net.
```

Il y a un problème ici:

- Les enregistrements NS pointent vers des noms, et non des adresses IP
- Que se passe-t-il si "example.com" est délégué à "ns.example.com"?
- Quelqu'un qui essaye de résoudre www.example.com doit d'abord résoudre ns.example.com
- Mais pour résoudre ns.example.com , il faut d'abord résoudre ns.example.com !!

Dans ce cas vous avez besoin de "glue"

- Un “enregistrement glue" est un enregistrement A pour le serveur de nom, tenu plus haut dans l’arborescence (hors de sa zone)
- Exemple: considerer les serveurs du .com, et une délégation pour

```
; ceci est la zone com.

example           NS   ns.example.com.
                  NS   ns.othernet.net.

ns.example.com.   A    192.0.2.1      ; enregistrement glue
```

Ne mettez pas de “glue” si vous n’en avez pas besoin

- Dans l’exemple précédent, "ns.othernet.net" n’est pas dans le sous-domaine "example.com". Par conséquent pas besoin de “glue”.
- Les enregistrements “glue” dépassés sont une grande source de problèmes
  - e.g. Après la renumérotation d’un serveur
  - Résulte en des problèmes intermittents, difficiles à déboguer

Exemple où un enregistrement “glue” est nécessaire

```
; mes données
            IN  MX  10  mailhost.example.net.
www         IN  A   212.74.112.80

; un sous-domaine délégué
subdom      IN  NS  ns1.subdom          ; needs glue
            IN  NS  ns2.othernet.net.   ; doesn't
ns1.subdom  IN  A   192.0.2.4
```

Recherche d'enregistrement "glue"

- dig +norec ... *et répeter plusieurs fois*
- Rechercher les enregistrements A dans la section additionnelle dont le TTL ne diminue pas

```
$ dig +norec @a.gtld-servers.net. www.as9105.net. a
 ...
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 1
;; QUERY SECTION:
;;      www.as9105.net, type = A, class = IN

;; AUTHORITY SECTION:
as9105.net.          172800  IN    NS        ns0.as9105.com.
as9105.net.          172800  IN    NS        ns0.tiscali.co.uk.

;; ADDITIONAL SECTION:
ns0.as9105.com.      172800  IN    A         212.139.129.130
```

Travaux pratiques

- Délégation de sous-domaine

Lectures

- "DNS and BIND" (O'Reilly)
- BIND 9 Administrator Reference Manual
  - /usr/share/doc/bind9/arm/Bv9ARM.html
- http://www.isc.org/sw/bind/
  - includes FAQ, security alerts
- RFC 1912, RFC 2182
  - http://www.rfc-editor.org/