DNS (*Domain Name System*)

Anne WEI
CNAM Paris

Bibliographie

- **Gérard Florin**, support de cours RSX102 « Technologie des applications Client-serveur », 2000
- P. Mockapetris, « Domain Names – Conepts and Facilities», RFC 1034, IETF, 1987,
- P. Mockapetris, «Domain Names – Implementation and Specification», RFC 1035, IETF, 1987
- P. Vixie, « Extension Mechanisms for DNS », RFC 2671, IETF, 1999
- P. Vixie and al, « Dynamic Updates in the Domain Name System (DNS UPDATE) », RFC 2136, IETF, 1997
- enwikipedia.org

Plan

1. Introduction
2. Structure et Base de données
3. Opérations principales
4. Conclusion

Introduction – DNS (1)

- DNS (*Domain Name System*) est un système « annuaire » (phone book) mettant une correspondance entre un nom logique et un identifiant « numérique ». Par exemple, la correspondance entre une @IP et un nom de hôte connecté à l'Internet, *deptinfo.cnam.fr* par exemple.

- DNS consiste en deux parties: une hiérarchie de noms logiques et un système d'adresses IP

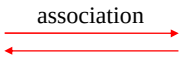
- DNS utilise la technologie *client/serveur* . Un *serveur* DNS stocke les informations telles que des @IP, les serveurs de nom et les échanges de message; un *client* interroge le serveur pour obtenir les informations nécessaires

Introduction – DNS (2)

- Dans un réseau Internet, chaque entité est identifiable par une adresse IP. Cependant, l'utilisateur préfère d'appeler une entité par son nom. Donc, il faut un système « annuaire »

- Exemples:

Adresse courrier : gerard@cnam.fr
Plutôt que :gerard@163.173.128.60
Nom de site web : http://www.cnam.fr
Plutôt que : http://163.173.128.28

gerard@cnam.fr 

gerard@163.173.128.60

Introduction – DNS (3)

- Les défis techniques:
- Comment regrouper toutes les adresses IP et leurs noms de hôte correspondants?
- Comment regrouper toutes les adresses IP *locales* et leurs noms de hôte correspondants? (main libre?)
- Quels sont les formats de données?
- Comment faire la mise à jour du fichier *hôte (fichier HOSTS.TXT)* quand le nombre de hôtes devient très élevé?
- Comment chercher une correspondance entre une adresse IP et un nom de domaine?
- …..

Historique

- Au début de l'Internet (ARPANET), les noms étaient définis *localement* dans un fichier « *hosts.txt* » (sous Unix, /etc/hosts, par exemple)
- ce fichier met la correspondance entre un nom et une adresse IP

127.0.0.1 localhost
163.173.212.2 cisco-for-acces35

- Ce fichier étais mis à jours par FTP de nuit automatiquement ou manuellement à partir d'une version récente. Cependant, le nombre de hôtes connectés à l'Internet augmente rapidement. Il faut trouver *une autre technique*.

- en 1983, les concepts (RFC 882) et les spécifications de DNS (RFC 883) ont été proposées. Elles sont finalisées en 1987 (RFC 1034 et RFC 1035)
- en 1984, la premier implémentation a été réalisé par 4 étudiants de Berkeley
- Depuis, DNS devient un outil fondamental de l'Internet…

Rappel – adresses IPv4 (1)

- **Adresses IPv4**
  - sous format de 4 nombres décimaux de 0 à 255. C'est-à-dire, la taille de l'adresse IPv4 est de 4 octets (32 bits). Ces 4 nombres sont séparés par « . »
  - **une adresse** contient deux parties suivantes: « réseau » et « hôte »
  - **un masque** sert à séparer la partie *réseau* et la partie *hôte*. On retrouve l'adresse réseau en effectuant une opération (bit à bit) *ET* entre l'adresse complète et le masque
  - **une adresse de diffusion** (192.168.1.255) permet au réseau de diffuser un paquet à l'ensemble des hôtes du réseau

adresse IPv4	192.	168.	1.	1
adresse de réseau	192.	168.	1.	0
adresse de masque	255.	255.	255.	0
adresse de diffusion	192.	168.	1.	255

adresse réseau

Rappel – adresses IPv4 (2)

- La taille de la partie réseau est variée selon les classes A, B, C , D et E
- La classes **A**: *le premier octet désigne le numéro de réseau* et le reste désigne l'adresse de hôtes. Le numéro du premier octet est de 1 à 126
- La classes **B**: *les deux premiers octets désignent le numéro de réseau* et le reste désigne l'adresse de hôtes. Le numéro du premier octet est de 128 à 191
- La classe **C**: *les trois premiers octets désignent le numéro de réseau* et le reste désigne l'adresse de hôtes. Le numéro du premier octet est de à 192 à 223
- La classe **D**: une adresse « multidiffusion » vers des groupes de hôtes. Le numéro du premier octet est de à 224 à 239
- La classe **E**: une adresse d'expérimentation. Le numéro du premier octet est de à 240 à 255

Rappel – adresses IPv4 (3)

classe	1er octet	nb d'octets réseau
A	1 à 126	1
B	128 à 191	2
C	192 à 223	3
D	224 à 239	multidiffusion
E	240 à 255	expérimentation

Rappel – adresses IPv6 (1)

- **Adresses IPv6**
- sous format de 8 mots de 16 bits. Chaque mot est un nombre hexadécimal. Autrement dit, la taille de l'adresse IPv6 est de 128 bits. Ces 8 mots sont séparés par « **:** »
- **4 types d'adresses: unicast, anycast, multicast et broadcast**
- le séparateur « **::** » permet d'abréger plusieurs mots nuls consécutifs

3201**:** 001A**:** 0000**:** 0000**:** 0000**:** 12BA**:** FFE0**:** 0002

3201**:**1A**::** 12BA**:** FFE0**:** 0002

abréviation de 3 mots « 0000 »

Rappel – adresses IPv6 (2)

- **Adresses IPv6 –** ***unicast (un à un)***
- Adresses lien local est une adresse restreinte à un lien (une interface)
- l'adresse lien local est utilisée par protocoles de détection (DAD – Duplication Address Detection), de configuration globale et de découvrir du voisinage
- Adresse site local est une adresse restreinte à un site
- Adresses unique local est une adresse restreinte dans un sous-réseau

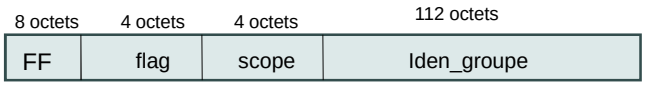
	10 octets	54 octets	64 octets
Adresses lien local	FE80	0000….0000	adresse lien local
Adresses site local	FECO	Iden_sous_réseau	adresse site local

	8 octets	40 octets	16 octets	64 octets
Adresses unique local	FC00	Iden_global	Iden_sous_réseau	adresse unique local

Rappel – adresses IPv6 (3)

- **Adresses IPv6**
- Adresses *multicast* (un à plusieurs) commencent par FF; scope = 1: interface-locale; scope = 2: link –local; scope = 5: site-local
- Adresses *anycast* (un à un parmi plusieurs) peut identifier un ensemble d'interfaces. Un paquet adressé à une adresse « *anycast* » doit être acheminé par une des interfaces. Généralement, l'interface du voisinage. Le format « *anycast* » est le même format « unicast »

Adresses multicast

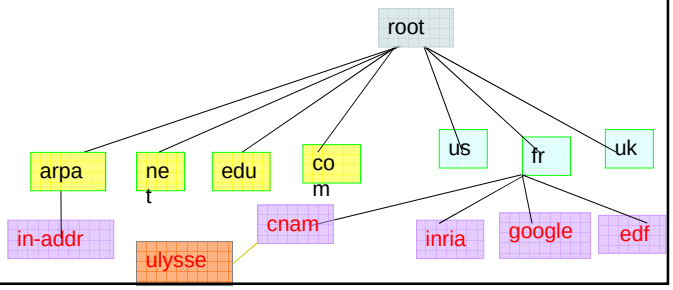


Plan

1. Introduction
2. Structure et Base de données
3. Opérations principales
4. Conclusion

Structure de Domain Name Space

- L'hiérarchie de nom logique (*Domain Name Space*) est une architecture de l'*arborescence*.
- Au niveau le plus haut: plusieurs centaines de nom de domaines.
  - *com*: nom génériques de domaines
  - *fr*: nom génériques de domaines
  - *arpa*: les info d'administration correspondant d'adresses IP vers noms
- aux niveaux intermédiaires: les noms de domaines sont les sous-domaines
- au niveau des feuilles: les sous-domaines composés d'hôtes ou définissant des services



Structure – Domain Name Formulation (1)

- Les formats de noms logiques (définis par RFC 1035, RFC 1123 et RFC 2181) consistent en une ou deux parties/labels (63-253 caractères, souvent en ASCII). Un nom est conçu en suivant l'arbre des feuilles vers la racine (*cnam.fr*, par exemple).

- niveaux de domaines
  - TLD (*Top Level Domains*)
  - TLD non ouverts

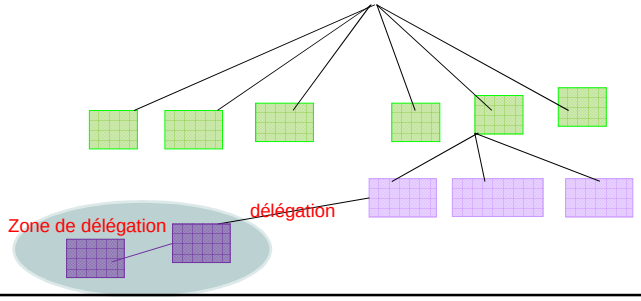
Structure – Domain Name Formulation (2)

- TLD (Top Level Domains) représente les domaines génériques (organisationnels)
  - .com: Organismes commerciaux (VeriSign)
  - .net: Prestataires réseaux (VeriSign)
  - .aero: Industries aéronautiques (SITA)
  - .museum: musées
  - .org: organismes publics

- TLD non ouverts
  - .edu: Institutions d'éducation US
  - .gov: Organisations gouvernementales US
  - .int: Organisations internationales
  - .fr: France
  - .uk: Angleterre
  - …

Structure – désignation arborescente

- L'arbre de désignation du *DNS* n'a pas de relation avec la topologie physique du réseau Internet.
- les niveaux intermédiaires sont des noms de domaines composés d'ensemble de ressources
- les feuilles sont des noms d'hôtes ou de services (www, par exemple); un nom d'hôte (*hostname*) doit être associé au moins une @IP.
- la désignation arborescente permet de déléguer l'administration des noms. Pour créer un nom à l'intérieur d'un domaine, il faut avoir l'autorisation de ce domaine (son administrateur).



Structure – noms

- Les noms absolus FQDN (*Fully Qualified Domain Name*) sont les noms de domaines qui dépendent de la localisation dans l'arbre. Un nom est terminé par un point final (.). Un nom complet a moins de 255 oct.
  - Par exemple, un nom complet: *cnam.fr.*
  - le domaine cnam appartient au domaine fr qui appartient à la racine
  - Notons que cette approche est différente de noms de fichier (/fr/cnam, par exemple),
- La casse est non significative: cnam=Cnam=CNAM
- les noms relatifs n'ont pas de point final et doivent être complétés par une chaîne de caractère pour former un nom absolu utilisable
- la différence entre FQDN et URL
  - FQDN (cnam.fr, par exemple) sont les noms de domaines
  - URL (Uniform Resource Locator) est la méthode d'accès à un document à distance par un lieu Hypertexte (, par exemple)

Structure – noms

- Les noms absolus FQDN (*Fully Qualified Domain Name*) sont les noms de domaines qui dépendent de la localisation dans l'arbre.
  - Par exemple, *cnam.fr*
  - le domaine cnam appartient au domaine fr qui appartient à la racine
  - Notons que cette approche est différente de noms de fichier (/fr/cnam, par exemple),
- La casse est non significative: cnam=Cnam=CNAM
- les noms relatifs n'ont pas de point final et doivent être complétés par une chaîne de caractère pour former un nom absolu utilisable
- la différence entre FQDN et URL
  - FQDN (cnam.fr, par exemple) sont les noms de domaines
  - URL (*Uniform Resource Locator*) est la méthode d'accès à un document à distance par un lieu Hypertexte (http://cnam.fr, par exemple)

**URI: (Universal Resource Identifier) soutenu par World Wide Web Consortium est ressemble de l'URL*

Base de données distribuées

- DNS est un système distribué en mode client-serveur. Chaque domaine possède au moins un serveur.

- L'unité d'informations s'appelle « Resource Record », l'enregistrement de ressources avec le format suivant: RR = (nom_de_domaine, durée_de_vie, classe, type, long_valeur, valeur)

Enregistrement de données - RR

- RR:
  - le nom de domaine identifie un nœud de l'arborescence; un nom absolu avec *taille variable*
  - la durée de vie (TTL *Time To Live*) donne la durée de validité de l'information dans un cache (nombre entier de secondes) avec *4 oct.*
  - la classe identifie *le protocole* utilisateur (IN pour Internet) avec *2 oct.*
  - le type (A, par exemple) indique le type de donnée avec *2 oct. ;* A signifie une adresse IPv4 de hôte;
  - la valeur (*taille variable*) contient les données associées au type
  - la longueur de la valeur avec *2 oct.*

Types principaux

- **Type**
  - A *(Address) :* signifie une adresse IPv4 de hôte
  - NS *(Name Serveur):* signifie un serveur de noms
  - MX *(Mail Exchanger)* : signifie un serveur de courriers
  - SOA (*Start of Authority*): indique les informations générales d'un domaine
  - PTR *(Pointer):* un pointeur sur un autre nom pour associer à une adresse IP et un nom
  - CNAME (*Canonical NAME*): un alias pour un autre nom

- **Classe** : la seule classe IN est utilisé en pratique
  - IN: Internet (@IP, nom de hôte ou de serveur)
  - CH: Choas utilisé par BIND (*Berkeley Internet Name Domain*)
  - HS (*Hesiod*): un service de nom permet d'accès aux bases de données

** l'utilisation de PTR est expliquée dans un transparent plus tard.*

Base de données distribuées - exemples

- Exemple du type SOA: {cnam.fr, 43200 (12 heures), IN, SOA,
  origin = asimov.cnam.fr
  mail addr = hostmaster.cnam.fr
  serial = 2000031515
  refresh = 21600 (6 heures)
  retry = 3600 (1 heure)
  expire = 1814400 (21 jours) }

- Exemple du type A: {ulysse, 86400, IN, A, 163.173.136.6}

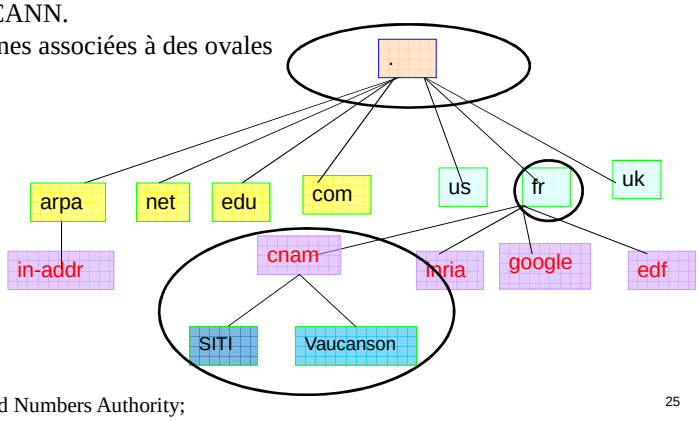
- Exemple du type NS: {cnam.fr, 86400, IN, NS, asimov.cnam.fr}

- Exemple du type MX: {cnam.fr, 86400, IN, MX, 10, fermi.cnam.fr}

** La liste des type RR se trouve dans les normées RFC 1035, RFC 1876…*

Serveurs DNS - Zone

- **Zone:** une unité d'administration (tous les membres d'une zone sont servies par **un même serveur**)
- *root zone* signifie à l'unité du niveau le plus haut (TLD). Elle est géré par IANA et maintenant par ICANN.
- Exemple de zones associées à des ovales



IANA: Internet Assigned Numbers Authority;
ICANN: Internet Corporation for Assigned Names and Numbers

Serveurs DNS - vocabulaire

- un domaine définit un ensemble de noms qui ont un même suffixe. C'est un découpage syntaxique de l'espace de nommage Internet

- une zone est un ensemble de noms qui ont un même suffixe et servis par le même serveur de nom. C'est un découpage administratif définissant la portée d'action des serveurs de noms.

- un serveur de noms pour une zone peut servir différents sous-domaines.

- une zone est servie par un (ou plusieurs) serveur primaire alimenté directement en informations par l'administrateur DNS. Les fichiers d'informations des serveurs primaires font l'autorité (*Authoritative Answer*)
- pour des raisons de performances et de fiabilité (tolérance aux pannes), il faut créer des serveurs secondaires. Les serveurs secondaires recopient périodiquement en utilisant TCP la base d'informations DNS d'un serveur primaire.

Serveurs DNS primaires et secondaires

- un domaine définit un ensemble de noms qui ont un même suffixe. C'est un découpage syntaxique de l'espace de nommage Internet
- une zone est un ensemble de noms qui ont un même suffixe et servis par le même serveur de nom. C'est un découpage administratif définissant la portée d'action des serveurs de noms.
- un serveur de noms pour une zone peut servir différents sous-domaines.
- une zone est servie par un (ou plusieurs) serveur primaire alimenté directement en informations par l'administrateur DNS. Les fichiers d'informations des serveurs primaires font l'autorité (*Authoritative Answer*)
- pour des raisons de performances et de fiabilité (tolérance aux pannes), il faut créer des serveurs secondaires. Les serveurs secondaires recopient périodiquement en utilisant TCP la base d'informations DNS d'un serveur primaire.

Plan

Opérations principales

- le serveur DNS traite les requêtes reçues par la méthode « récursive » ou la méthode « itérative » dans l'ordre de file d'attente.

- pour une raison de performances, la technique «cache» est introduite dans systèmes DNS. Un cache contient les réponses à des requêtes récentes durant une période TTL afin de garantir la cohérence d'informations.

- un « résolveur » est une procédure invoquée sur un client pour un accès DNS.
- Le résolveur s'adresse tout d'abord à un serveur DNS local en présentant une question concernant un nom de domaine
- Si le serveur local connaît la réponse dans sa base de données ou dans son cache, il retour au client de la réponse.
- sinon, la recherche dans les sous domaines et/ou la recherche sur d'autres serveurs DNS

Opérations principales

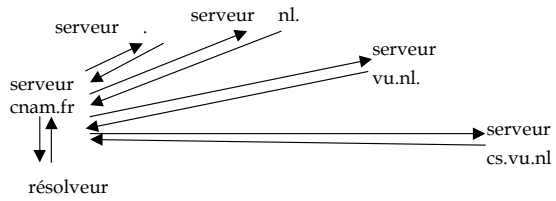
- le serveur DNS traite les requêtes reçues par la méthode « récursive » ou la méthode « itérative » dans l'ordre de file d'attente.

- pour une raison de performances, la technique « cache » est introduite dans systèmes DNS. Une cache contient les réponses à des requêtes récentes durant une période TTL afin de garantir la cohérence d'informations.

- un «*résolveur*» est une procédure invoquée sur un client pour un accès DNS.
- Le résolveur s'adresse tout d'abord à un serveur DNS local en présentant une question concernant un nom de domaine
- Si le serveur local connaît la réponse dans sa base de données ou dans son cache, il retourne au client de la réponse.
- sinon, la recherche dans les sous domaines et/ou la recherche sur d'autres serveurs DNS

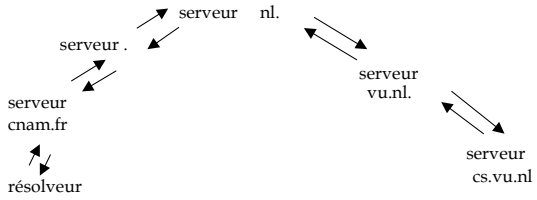
Recherche itérative

- On interroge successivement les serveurs
- Exemple: au CNAM on cherche le nom *star.cs.vu.nl*
- Le résolveur interroge d'abord le serveur local *cnam.fr*
- Le serveur local répond par l'@IP du serveur racine. Le résolveur interroge ensuite le serveur racine. Le serveur racine répond par l'@IP du serveur *nl*….
- Chaque serveur visité donne l'adresse IP du serveur suivant à interroger



Recherche récursive

- Chaque serveur visité prend l'initiative d'interroger le serveur suivant
- Exemple: au CNAM on cherche le nom *star.cs.vu.nl*
- le serveur local interroge le serveur racine
- le serveur racine interroge le serveur «.nl»
- …

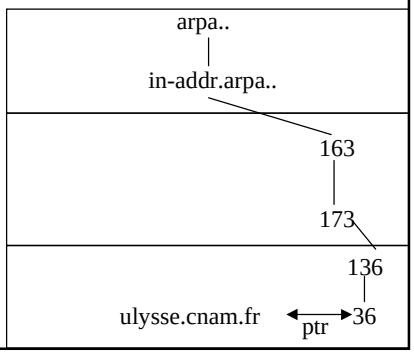


Requêtes inverses

- Une requête inverse (*reverse lookup*) est une requête formulée à partir de la valeur d'un attribut (une adresse IP, par exemple)
- RR (*Resource Record*) est le type PTR (pointer) qui indique un nom associé à une adresse IP
- les serveurs DNS de ARPA stocke les adresses IP
  - les adresse IPv4 se trouve dans le serveur *in-addr.arpa*
  - les adresse IPv6 se trouve dans le serveur *ip6.arpa*

- Exemple:

Une enquête de DNS:
{IN, PTR, 36.136.173.163. in-addr.arpa}
Réponse:
nom de domaine = ulysse.cnam.fr



Rappel - systèmes DNS

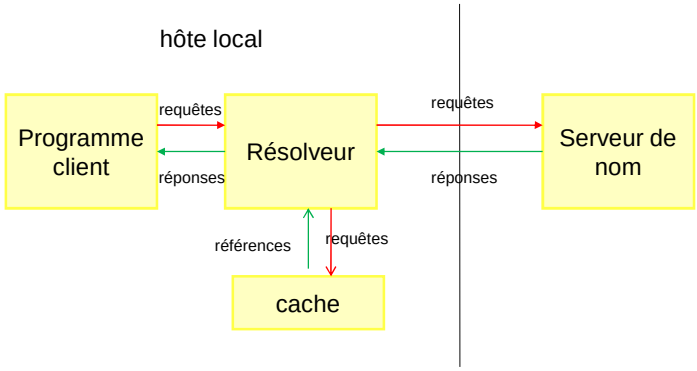
- **Un système DNS consiste en trois principales parties:**
  - L'hiérarchie de noms de domaines est une **architecture d'arborescence** (la racine, noms de domaines, noms de sous-domaines et hôtes)
  - **Serveur de nom** est un serveur qui possède toutes les informations de ses sous-domaines, les informations concernant ce serveur et les pointeurs vers d'autres serveurs de nom. Un serveur est une autorité désignée par l'administrateur.
  - **Résolveur** est une procédure permettant de retirer des informations qu'un client demande

- la technique « cache » est utilisée afin d'éviter à interroger des informations trop volumineuses.

** La mise à jour de la base de données du cache est très importante.*

Rappel - configuration

- une configuration commune:



=> Comment former une requête?

Requête – réponse (1)

- **le message utilisé dans un protocole de transport (UDP ou TCP):**

en-tête
Questions (taille variable)
Réponses (taille variable)
Serveurs faisant autorité (taille variable)
Info supplémentaires (taille variable)

- **l'en-tête:**

Identifiant
drapeaux
Nb de questions
Nb de réponses
Nb de serveurs faisant autorité

Requête – réponse (2)

- les champs « requête»:
- Exemple: *un e-mail* adressé à *Mockapetris@ISI.EDU* doit être spécifié comme QNAME = *ISI.EDU*; QTYPE=*MX*, QCLASS = *IN*

- Les champs «réponse » (Resource Record) :

ISI.EDU MX 10 VENERA.ISI.EDU
MX 10 VAXA.ISI.EDU

* Les informations supplémentaires pourront être:

VAXA.ISI.EDU A 10.2.0.27
A 128.9.0.33
VENERA.ISI.EDU A 10.1.0.52
A 128.9.0.32

* QTYPE signifie querttype. Les détails des champs se trouvent dans RFC 1035

Requête – réponse (3)

- **le message utilisé dans un protocole de transport:**

en-tête
QNAME = *ISI.EDU*; QTYPE=*MX*, QCLASS = *IN*
ISI.EDU MX 10 VENERA.ISI.EDU MX 10 VAXA.ISI.EDU
Vide
VAXA.ISI.EDU A 10.2.0.27 A 128.9.0.33 VENERA.ISI.EDU A 10.1.0.52 A 128.9.0.32

- Le ports utilisé dans les protocoles UDP et TCP est 53
- TCP est souvent utilisé si le message dépasse le format de UDP (512oct)
- TCP est utilisé pour transférer périodiquement les informations relatives à une zone entre serveurs primaires et secondaires

DNS dynamique

- DNS est un système dynamique permettant de modifier automatiquement des informations dans la base de données du côté de serveur et du côté de client
- Message UPDATE ci-dessous utilise la même en-tête d'un message DNS

en-tête
Zone
Condition préalable
modifications
Informations supplémentaires

- Tous les serveurs « autorités» peuvent demander les modifications, en particulier, les serveurs primaires

Plan

1. Introduction
2. Structure et Base de données
3. Opérations principales - outil
4. Conclusion

Utilisation du DNS – *nslookup*

- *nslookup (Name Server Lookup)* est un outil qui permet de
- connaître le nom de serveur local et son adresse associée
- rechercher d'adresses IP et de noms

- Exemples:

$ nslookup
Default server: asimov.cnam.fr *(serveur DNS local)*
Adresse: 163.173.128.6

Recherche d'adresse IP et de nom:
$ set q=A // (qtype=A)
$ ulysse.cnam.fr
Server: asimov.cnam.fr
Address: 163.173.128.6
Name: ulysse.cnam.fr
Address: 163.173.136.36
$exit

Recherche inverse d'un nom par l'adresse (1)

$ set q=A
$ 163.173.136.6
Server: asimov.cnam.fr
Address: 163.173.136.6

Name:asimov.cnam.fr
Address: 163.173.136.6

$ 36.136.173.163.in-addr.arpa..
Server: asimo.cnam.fr
Address: 163.173.136.6
***asimo.cnam.fr can't fond 36.136.173.163.in-addr.arpa:
Non-existent host/domain

Pourquoi la requête ne peut pas être résolue?

Recherche inverse d'un nom par l'adresse (2)

```
$ set q=ptr
$ 36.136.173.163.in-addr.arpa..
Server:  asimov.cnam.fr
Address:  163.173.128.6

36.136.173.163.in-addr.arpa        name = ulysse.cnam.fr

173.163.IN-ADDR.ARPA    nameserver = asimov.cnam.fr
173.163.IN-ADDR.ARPA    nameserver = fermi.cnam.fr
asimov.cnam.fr  internet address = 163.173.128.6
fermi.cnam.fr    internet address = 163.173.128.60
```


Recherche d'un domaine

```
$ set q=SOA
$ escpi.cnam.fr
Server: asimov.cnam.fr
Address: 163.173.136.6

escpi.cnam.fr
        origin = pollux.escpi.cnam.fr
        mail addr = Postmaster.escpi.cnam.fr
        serial = 2000011016
        refresh = 21600 (6 hours)
        retry   = 3600 (1 hour)
        expire  = 1209600 (14 days)
        minimum ttl = 86400 (1 day)
escpi.cnam.fr   nameserver = pollux.escpi.cnam.fr
escpi.cnam.fr   nameserver = bellatrix.escpi.cnam.fr
escpi.cnam.fr   nameserver = asimov.cnam.fr
pollux.escpi.cnam.fr    internet address = 163.173.112.100
bellatrix.escpi.cnam.fr internet address = 163.173.112.4
asimov.cnam.fr  internet address = 163.173.128.6
```

Recherche de serveur de courrier

$ set q=mx
$ escpi.cnam.fr
Server: asimov.cnam.fr
Address: 163.173.136.6

escpi.cnam.fr preference = 20, mail exchanger = fermi.cnam.fr
escpi.cnam.fr preference = 10, mail exchanger = postman.escpi.cnam.fr
escpi.cnam.fr nameserver = pollux.escpi.cnam.fr
escpi.cnam.fr nameserver = bellatrix.escpi.cnam.fr
escpi.cnam.fr nameserver = asimov.cnam.fr
fermi.cnam.fr internet address = 163.173.128.60
postman.escpi.cnam.fr internet address = 163.173.112.5
pollux.escpi.cnam.fr internet address = 163.173.112.100
bellatrix.escpi.cnam.fr internet address = 163.173.112.4
asimov.cnam.fr internet address = 163.173.128.6

Recherche de toutes les informations d'une zone

$ set q=any
$ escpi
Server: asimov.cnam.fr
Address: 163.173.128.6
escpi.cnam.fr internet address = 163.173.112.5 escpi.cnam.fr preference = 10,
mail exchanger = postman.escpi.cnam.fr escpi.cnam.fr preference = 20,
mail exchanger = fermi.cnam.fr escpi.cnam.fr text = "Unauthorized access to this network is prohibited"
escpi.cnam.fr text = "Ecole Supérieure de Conception et Production Industrielle "
escpi.cnam.fr nameserver = pollux.escpi.cnam.fr
escpi.cnam.fr nameserver = bellatrix.escpi.cnam.fr
escpi.cnam.fr nameserver = asimov.cnam.fr
escpi.cnam.fr
origin = pollux.escpi.cnam.fr mail addr = Postmaster.escpi.cnam.fr
serial = 2000042020 refresh = 21600 (6H)
retry = 3600 (1H) expire = 1209600 (2W)
minimum ttl = 86400 (1D) escpi.cnam.fr nameserver = pollux.escpi.cnam.fr
escpi.cnam.fr nameserver = bellatrix.escpi.cnam.fr
escpi.cnam.fr nameserver = asimov.cnam.fr
postman.escpi.cnam.fr internet address = 163.173.112.5
fermi.cnam.fr internet address = 163.173.128.60
pollux.escpi.cnam.fr internet address = 163.173.112.100
bellatrix.escpi.cnam.fr internet address = 163.173.112.4
asimov.cnam.fr internet address = 163.173.128.6

*** l'autres outils client DNS existent comme host sous Unix**

Plan

1. Introduction
2. Structure et Base de données
3. Opérations principales - sécurité
4. Conclusion

Sécurité de DNS

- Quelques mots sur la sécurité de DNS:
  - les premières spécifications (RFC 1034 et RFC 1035) n'assurent pas la sécurité de DNS
  - une requête avec SOA permet d'obtenir toutes les informatiques génerales
  - la requête et la réponse ne sont pas cryptés
  - les mise à jours de base de données (cache, serveur primaire, serveur secondaire…) ne sont pas protégées

- le standard *RFC 3833* «Threat Analysis of the Domain Name System (DNS) » en 2004 et le standard *RFC 4033* « DNS Security Introduction and Requirements » en 2005 proposent les solutions

Plan

1. Introduction
2. Structure et Base de données
3. Opérations principales - configuration
4. Conclusion

Administration du DNS

- Du côté du serveur DNS, deux façons de construire un serveur DNS:
  Utiliser le serveur DNS de son fournisseur (FAI *Fournisseur d'Accès Internet* ou ISP *Internet Service Provider*)
  - construire son propre serveur

- Créer son propre serveur DNS permet
  - de protéger les données personnelles et l'architecture de réseau
  - d'offrir une performance plus élevée (le délai de recherche, par exemple) par rapport d'un serveur de FAI surchargé
  - de mettre à jour les informations de son propre domaine sans en référer en permanence à son fournisseur

Configuration du serveur DNS avec un fournisseur d'accès Internet

- Enregistrer un nom de domaine auprès de l'autorité compétente (un nom *.com .fr*)

- Le FAI fournit l'adresse IP de ses serveurs primaires et secondaires
  - configurer les piles TCP/IP des sites clients DNS
  - ou utiliser le serveur DHCP (*Dynamic Host Configuration Protocol*)

- le FAI insère dans la base de données les enregistrements (Resource Records) à rendre publics
  - pour recevoir du courrier sur le serveur de courrier (Mail Exchange MX)
  - pour associer les adresses IP aux noms de domaines (type A) serveur de courrier, serveur FTP et serveur WEB

Configuration de serveur DNS – fichiers (1)

- En Unix/Linux, les fichiers concernés
  - du site client DNS: /etc/resolv.conf
  - du site serveur DNS: /etc/named.boot; /var/named/root.cache; /etc/conf.linuxconf

- Comment connaître les serveurs DNS qui savent répondre pour le noms de domaines « intermédiaires» afin de résoudre le nom de domaine recherché?
  - ftp://ftp.rd.internic.net/named.root
  - l'InterNIC est l'organisme qui s'occupe de noms de domaines TLD et leurs adresses IP associées

Configuration de serveur DNS – fichiers (2)

- **le ficher named.root**

```
;last update: Jun 17, 2010
; related version of root zone: 2010061700
 ; formerly   NS.INTERNIC.NET
 ; .                    3600000           IN     NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000                A      198.41.0.4
A.ROOT-SERVERS.NET.      3600000                AAAA       2001:503:BA3E::2:30
; FORMERLY NS1.ISI.EDU
 ; .                    3600000                NS    B.ROOT-SERVERS.NET. B.ROOT-SERVERS.NET.     3600000
                A       192.228.79.201
;  FORMERLY C.PSI.NET
 ; .                   3600000                 NS    C.ROOT-SERVERS.NET. C.ROOT-SERVERS.NET.     3600000
                A       192.33.4.12
 ; FORMERLY TERP.UMD.EDU
 ; .                    3600000                NS     D.ROOT-SERVERS.NET. D.ROOT-SERVERS.NET.     3600000
                A       128.8.10.90
;  FORMERLY NS.NASA.GOV
 ; .                    3600000                NS    E.ROOT-SERVERS.NET. E.ROOT-SERVERS.NET.     3600000
                A       192.203.230.10
….
```


Configuration du serveur DNS (1)

- Les outils de serveur DNS:
  - BIND (*Berkeley Software Distribution*) est largement utilisé comme un standard sous systèmes Unix/Linux.
  - Microsoft DNS avec Windows Server
- construction d'un cache avec BIND; installer BIND correctement
- configuration avec *linuxconf*
  - vérifier si « module.list 1 dnsconf » est dans le fichier conf.linuxcon
  - choisir l'option « Configurateur réseau/Domain Name Server (DNS) »
  - choisir l'option « Configurateur DNS/fonctionnalités »
  - Dans « Fonctionnalités DNS », choisir « sécurité/contrôle d'accès » où on peut limiter d'accès par une classe de réseau, 192.168.0.0, par exemple
  - Dans « Fonctionnalités DNS », choisir « sécurité/contrôle d'accès», on règle l'autorité de requêtes.

** l'autre outil équivalent est Webmin*

Configuration du serveur DNS (2)

- Vérifier les fichiers de configuration
  - vérifier le fichier *named.conf*. On trouve la zone « . » qui permet à BIND d'interroger les serveurs racines; on trouve la zone « 0.0.127.IN-ADDR.ARPA » qui permet de la recherche inverse par l'adresse IP de l'hôte
  - Vérifier le fichier *root.cache* dans /etc/var si le fichier est mise à jour

- Vérifier le Daemon « named » est en service (*processus)* à l'aide de l'outil *SysV.* C'est-à-dire, « named » doit être affiché à la fois dans « Services » et « Runlevel 3 Démarrer »

Plan

Conclusion

- Un système d'annuaire distribué au niveau mondial (mode client-serveur)
- Une organisation en arbre simple et efficace
- La recherche d'un nom (ou la recherche inverse) est simple
  - *une adresse IP* peut être associée avec plusieurs noms de hôte dans le but d'économiser la ressource, un exemple, deux sites Web localisés par une même adresse (c'est un type *virtually-hosted website*)
  - un nom de hôte peut être associé avec plusieurs adresses IP dans le but d'assurer la tolérance aux pannes
- Les problèmes de performances et de sûreté sont résolus par les serveurs primaires et secondaires mais aussi par l'utilisation de cache
- Les mesures de sécurité du DNS n'a pas abordée dans ce cours

….