

TD4 - Applications de l'internet

Proposition de correction

Désignation et liaison dans le WEB

1 - Les identificateurs

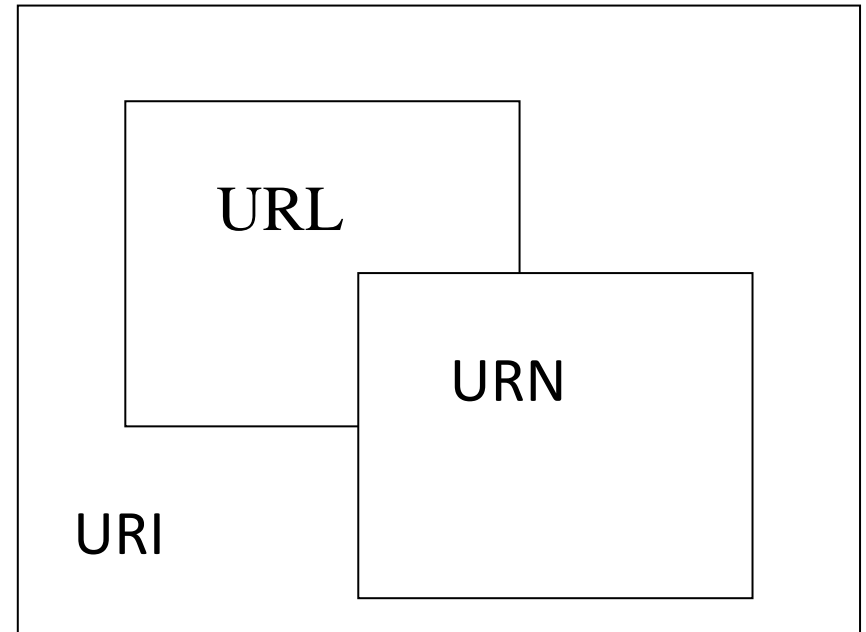
URI 'Uniform Resource Identifiers'

2 - Les noms

URN 'Uniform Resource Names'

3 - Les localisateurs

URL 'Uniform Resource Locators'



Les identificateurs - URI

Un URI (*identifiant uniforme de ressource*) doit permettre d'identifier une ressource de manière permanente, même si la ressource est déplacée ou supprimée.

Un **URI** peut être de type :

- « *locator* »
- « *name* »
- les deux.

Les localisateurs - URL

URL 'Uniform Resource Locators' (RFC 1738 - 12/1994)

La désignation avec les URL est placée dans le cadre d'un ensemble de **schémas** associés à des applications Internet:

ftp	File Transfer protocol
http	Hypertext Transfer Protocol
mailto	Adresse de courrier électronique
news	USENET news
telnet	Référence session interactive
file	Noms de fichiers spécifiques
...	

Les localisateurs - URL

Les URL HTTP définissent un adressage absolu **pratique pour la localisation des ressources** mais uniquement pour cela ...

Les URL comportent essentiellement des informations **d'adresses ('codées en dur') :** Protocole, Adresse DNS, No Port

Ainsi, lorsqu'une ressource est déplacée, par exemple mise sur un autre serveur Web, toutes les URL l'identifiant sont rendues obsolètes.

Les noms - URN

URN 'Uniform Resource Names'

- Les URN ont été définis au départ pour résoudre le problème **de persistance** des noms que ne possèdent pas les URLs.
- Les noms choisis doivent donc être **indépendants de la localisation** des ressources: les URN sont en fait les **noms symboliques** du WEB.
- La résolution doit faire appel à **des services d'annuaires** qui font correspondre noms symboliques et adresses.

Les noms - URN

URN selon la RFC 2141 Mai 1997

Structure générale :

<URN> ::= "urn:" <NID> ":" <NSS>

Exemple : urn:cnam.fr:CoursInfos

→ Ces URN sont peu utilisés.

Les noms - URN

Exemple :

http://etud.cnam.fr/chemin/symbole

adresse de serveur d'annuaire



The diagram consists of two blue arrows. The first arrow originates from the text 'adresse de serveur d'annuaire' and points diagonally upwards to the 'etud.cnam.fr' portion of the URL 'http://etud.cnam.fr/chemin/symbole'. The second arrow originates from the text 'nom symbolique : le résolveur transforme en une URL effective par consultation d'annuaire' and points diagonally upwards to the 'symbole' portion of the same URL.

nom symbolique : le résolveur transforme en une URL effective par consultation d'annuaire

Les noms - URN

Dans la pratique, les URN ne sont guère utilisés.

Les problèmes de localisation de ressource sont généralement résolus avec un moteur de recherche.

On peut noter la fonctionnalité de « document en cache » qui conserve un certain temps une version du document référencé, indépendamment de son accessibilité à son URL originale.

RFC

Spécification et RFC

Les RFC sont rédigées sur l'initiative d'experts techniques, puis sont revues par la communauté Internet dans son ensemble.

Network Working Group
Request for Comments: 2616
Obsoletes: 2068
Category: Standards Track

R. Fielding
UC Irvine
J. Gettys
Compaq/W3C
J. Mogul
Compaq
H. Frystyk
W3C/MIT
L. Masinter
Xerox
P. Leach
Microsoft
T. Berners-Lee
W3C/MIT
June 1999

Plusieurs liens
existent :

<http://www.rfc.fr/>

<http://abcdrfc.free.fr/>

<http://www.ietf.org/rfc.html>

Hypertext Transfer Protocol -- HTTP/1.1

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

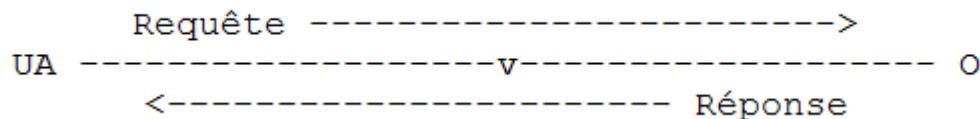
Spécification et RFC

Dans les RFC on trouve :

1 – La façon d'accéder au service (URL)

```
http_URL      = "http:" "://" host [ ":" port ] [ chem_abs ]  
  
host          = <un nom Internet d'ordinateur valide ou une adresse IP  
              (sous forme numérique), comme définie en Section 2.1 de la  
              RFC 1123>  
  
port          = *DIGIT
```

2 – La structure générale des échanges



Spécification et RFC

3 – La liste des méthodes

```
Méthode          = "GET"                ; Section 8.1  
                  | "HEAD"              ; Section 8.2  
                  | "POST"              ; Section 8.3  
                  | nom_de_méthode
```

4 – La description des méthodes

8.1 GET

La méthode GET signifie "récupérer" le contenu quel qu'il soit de la ressource (sous forme d'une entité) identifiée par l'URI-visée. Si l'URI-visée identifie un processus générant dynamiquement des données, ce sont les données produites qui sont renvoyées dans l'entité au lieu du source de l'exécutable appelé, sauf si ce texte lui-même est la sortie du processus.

Spécification et RFC

5 – Les réponses

Les codes réponses informent de la suite donnée à la requête
→ Code + libelle

Le premier chiffre définit la catégorie:

1xx un message informatif.

2xx une opération réussie.

3xx une redirection vers une autre URL

4xx une erreur du client

5xx une erreur du serveur

Spécification et RFC

6 – La liste des réponses

Code_état	= "200" ; OK	OK
	"201" ; Created	Créé
	"202" ; Accepted	Accepté
	"204" ; No Content	Pas de contenu
	"301" ; Moved Permanently	Changement définitif
	"302" ; Moved Temporarily	Changement temporaire
	"304" ; Not Modified	Non modifié
	"400" ; Bad Request	Requête incorrecte
	"401" ; Unauthorized	Non autorisé
	"403" ; Forbidden	Interdit
	"404" ; Not Found	Non trouvé
	"500" ; Internal Server Error	Erreur interne serveur
	"501" ; Not Implemented	Non implémenté
	"502" ; Bad Gateway	Erreur de routeur
	"503" ; Service Unavailable	Indisponible

7 – La description des réponses

9.2 Succès 2xx

Cette classe précise que la requête du client a été correctement transmise, interprétée, et exécutée.

200 OK

La requête a abouti. L'information retournée en réponse dépend de la requête émise, comme suit:

Spécification et RFC

8 – Le format détaillé des requêtes et des réponses

```
Requête_complète      = Ligne_requête                ; Section 5.1  
                      *( En-tête_générale              ; Section 4.3  
                        | En-tête_requête               ; Section 5.2  
                        | En-tête_entité )             ; Section 7.1  
                      CRLF  
                      [ Corps_entité ]                 ; Section 7.2  
  
Réponse_complète     = Ligne-état                     ; Section 6.1  
                      *( En-tête_générale              ; Section 4.3  
                        | En-tête_réponse               ; Section 6.2  
                        | En-tête_entité )             ; Section 7.1  
                      CRLF  
                      [ Corps entité ]                 ; Section 7.2
```


Spécification et RFC

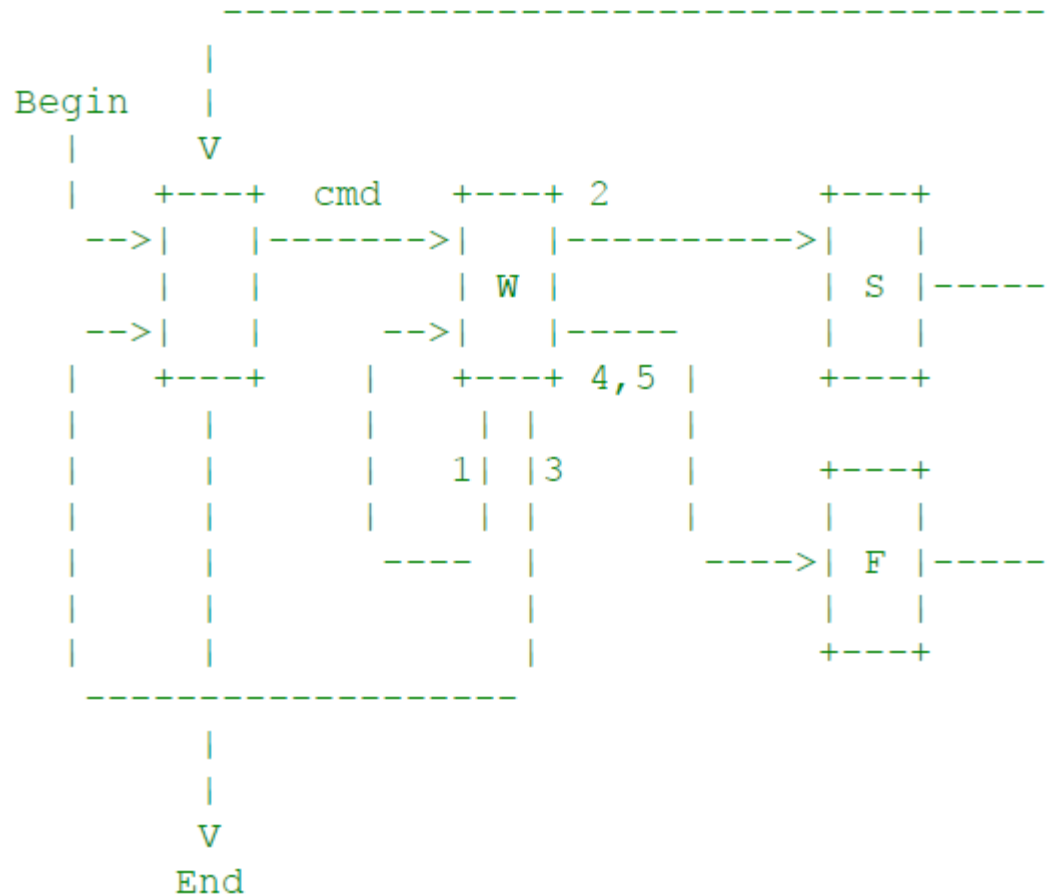
9 – Le format des données

Descripteur	Compte
code=16	= 6

Marqueur	Marqueur	Marqueur
8 bits	8 bits	8 bits

Spécification et RFC

10 – Les échanges types – Diagrammes d'états



HTTP

Le protocole HTTP

Protocole de communication client-serveur défini à partir de 1989 pour supporter les échanges de données du 'World Wide Web' :

- conçu à l'origine pour transporter des textes HTML.
- a évolué considérablement sous la pression d'utilisateurs très nombreux d'une version simple et rustique 0.9 à une version assez puissante 1.1

Le protocole HTTP

Caractéristiques de base :

HTTP se définit comme un protocole

- de niveau application
- dédié aux applications distribuées
- hypermédia

HTTP permet à des systèmes conçus indépendamment de dialoguer :

- typage des données échangées.
- négociation de la représentation des données.

Le protocole HTTP

Caractéristiques de base :

- utilise le **transport fiable TCP** (d'autres réseaux sont possibles).
- protocole de niveau application utilisé pour transporter des données baptisées **ressources**
- une ressource est un **ensemble d'informations** identifié par une URL ou plus généralement une URI.
(Le plus souvent un **fichier**)

Le protocole HTTP

Caractéristiques de base :

- **HTTP : un protocole client serveur basé sur le mode message**
- Tous les échanges sont basés sur un couple de **messages requête réponse**.
- A chaque couple requête réponse est associé une **‘méthode’**.
- **Approche asynchrone : un client n’est pas bloqué par une requête** (il peut émettre plusieurs requêtes en parallèle).

La version HTTP/1.0

Structure des échanges HTTP/1.0

- Le client HTTP/1.0 ouvre une **connexion TCP** (le **numéro de port par défaut de http** est 80, un autre numéro est possible).
- Le client envoie **un message de requête**(exemple type : obtenir une ressource en fonction de différents paramètres).
- Le serveur répond par **un message de réponse** (la **ressource demandée**).
- Le client **ferme la connexion TCP**

Remarque : En HTTP/1.0 chaque ressource nécessite sa propre connexion TCP.

La version HTTP/1.0

La liste des méthodes HTTP/1.0

Les trois méthodes principales

GET : Pour lire la valeur d'une ressource.

HEAD : Pour lire seulement l'entête.

POST: Pour faire exécuter un programme distant.

Les méthodes annexes

PUT : Pour envoyer une ressource sur un site distant.

DELETE : Pour détruire une ressource distante.

LINK : Pour établir un lien entre deux ressources.

La version HTTP/1.0

Le format de la ligne initiale des requêtes

Trois parties séparées par des espaces

- Un **nom de méthode**,
- Le **chemin d'accès à la ressource**,
- La **version du protocole HTTP**.

Exemple : GET /chemin HTTP/1.0

La version HTTP/1.0

Les codes réponses informent de la suite donnée à la requête = Code + libelle

Le premier chiffre définit la catégorie:

- 1xx un message informatif.
- 2xx une opération réussie.
- 3xx une redirection vers une autre URL
- 4xx une erreur du client
- 5xx une erreur du serveur

La version HTTP/1.0

Les codes réponses les plus courants

"200" ; OK

"201" ; Created

"202" ; Accepted

"204" ; No Content

"301" ; Moved Permanently

"302" ; Moved Temporarily

"304" ; Not Modified

"400" ; Bad Request

"401" ; Unauthorized

"403" ; Forbidden

"404" ; Not Found

"500" ; Internal Server Error

La version HTTP/1.0

Le format de la ligne initiale des réponses

Trois parties séparées par des espaces

- . La version du protocole HTTP,**
- . Le statut de la réponse (code réponse),**
- . Un texte expliquant le code réponse.**

Exemples caractéristiques de réponses:

HTTP/1.0 200 OK

HTTP/1.0 404 Not Found

La version HTTP/1.0

Exemple

Requête : GET /index.html HTTP/1.0

Réponse

HTTP/1.1 200 OK

Date: Wed, 25 Oct 2000 10:02:12 GMT

Server: Apache/1.3.6 (Unix) PHP/3.0.7

Last-Modified: Wed, 18 Oct 2000 11:07:00 GMT

ETag: "ac-221e-39ed8454"

Accept-Ranges: bytes

Content-Length: 8734

Connection: close

Content-Type: text/html

<html>

<head>

<link rel="stylesheet"

.....

La version HTTP/1.0

Exemple d'échange

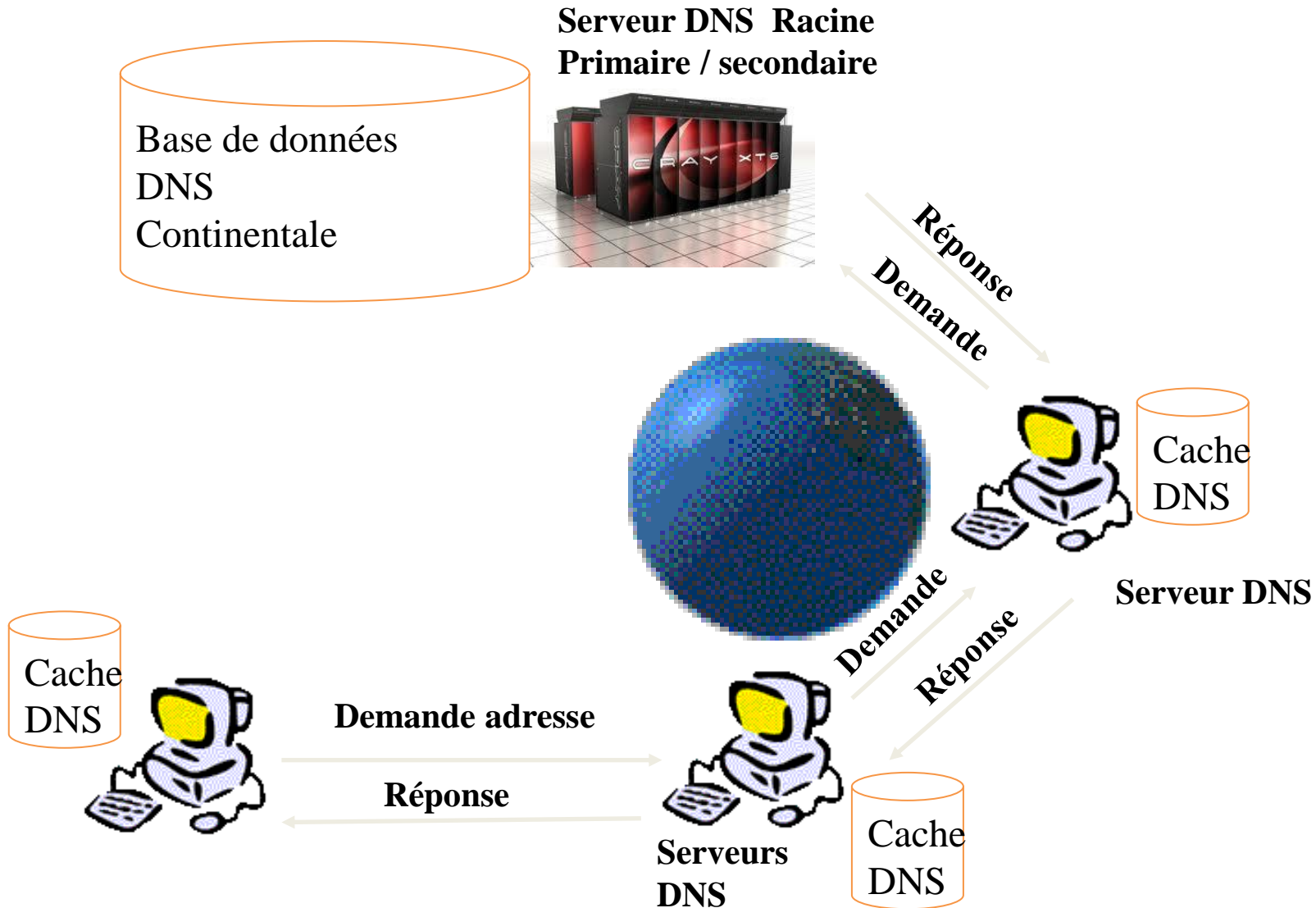
10.20.1.1

66.249.85.99

```
TCP 33667 > http [SYN] Seq=0 Ack=0 Win=5840 Len=0      →
    ←      TCP http > 33667 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0
TCP 33667 > http [ACK] Seq=1 Ack=1 Win=5840 Len=0      →
HTTP GET / HTTP/1.0                                  →
    ←      TCP http > 33667 [ACK] Seq=1 Ack=426 Win=7765 Len=0
    ←      HTTP HTTP/1.0 200 OK (text/html)
TCP 33667 > http [ACK] Seq=426 Ack=1431 Win=8580 Len=0 →
    ←      HTTP Continuation
TCP 33668 > http [ACK] Seq=520 Ack=2062 Win=11440 Len=0 →
    ←      HTTP Continuation
    ←      HTTP Continuation
```

DNS

Protocole DNS (Domain Name Service)



Protocole DNS (Domain Name Service)

Un serveur DNS primaire et 12 serveurs DNS secondaires.

Serveur DNS primaire : A.ROOT-SERVERS.NET (198.41.0.4).

Serveurs DNS secondaires : B.ROOT-SERVERS.NET à M.ROOT-SERVERS.NET, adresse IP sont les suivantes :

B.ROOT-SERVERS.NET	192.228.79.201
C.ROOT-SERVERS.NET	192.33.4.12
D.ROOT-SERVERS.NET	199.7.91.13
E.ROOT-SERVERS.NET	192.203.230.10
F.ROOT-SERVERS.NET	192.5.5.241
G.ROOT-SERVERS.NET	192.112.36.4
H.ROOT-SERVERS.NET	128.63.2.53
I.ROOT-SERVERS.NET	192.36.148.17
J.ROOT-SERVERS.NET	192.58.128.30
K.ROOT-SERVERS.NET	193.0.14.129
L.ROOT-SERVERS.NET	198.32.64.12
M.ROOT-SERVERS.NET	202.12.27.33

```
C:\Users\Francis>nslookup
Serveur par défaut : dns1.univ-montp2.fr
Address: 162.38.101.51

> set type=ns
>
Serveur : dns1.univ-montp2.fr
Address: 162.38.101.51

Réponse ne faisant pas autorité :
(root) nameserver = h.root-servers.net
(root) nameserver = i.root-servers.net
(root) nameserver = j.root-servers.net
(root) nameserver = k.root-servers.net
(root) nameserver = l.root-servers.net
(root) nameserver = m.root-servers.net
(root) nameserver = a.root-servers.net
(root) nameserver = b.root-servers.net
(root) nameserver = c.root-servers.net
(root) nameserver = d.root-servers.net
(root) nameserver = e.root-servers.net
(root) nameserver = f.root-servers.net
(root) nameserver = g.root-servers.net

a.root-servers.net internet address = 198.41.0.4
a.root-servers.net AAAA IPv6 address = 2001:503:ba3e::2:30
b.root-servers.net internet address = 192.228.79.201
b.root-servers.net AAAA IPv6 address = 2001:500:200::b
c.root-servers.net internet address = 192.33.4.12
c.root-servers.net AAAA IPv6 address = 2001:500:2::c
d.root-servers.net internet address = 199.7.91.13
d.root-servers.net AAAA IPv6 address = 2001:500:2d::d
e.root-servers.net internet address = 192.203.230.10
e.root-servers.net AAAA IPv6 address = 2001:500:a8::e
f.root-servers.net internet address = 192.5.5.241
f.root-servers.net AAAA IPv6 address = 2001:500:2f::f
g.root-servers.net internet address = 192.112.36.4
>
```

Protocole DNS (Domain Name Service)

Serveur racines



Structure du service DNS

Le DNS est un rouage essentiel du fonctionnement d'Internet. Véritable annuaire géant, le DNS est, par ailleurs, subdivisé en répertoires plus petits appelés « domaines ». Pouvant être très vastes, ces derniers se subdivisent également en annuaires plus petits : les zones DNS.

Ainsi, il n'y a pas de serveur DNS unique qui centralise tous les répertoires. Il existe un grand nombre de serveurs DNS conçus pour stocker tous les enregistrements DNS du web.

Tout ordinateur cherchant à connaître une adresse IP ou un nom peut alors interroger son serveur DNS.

Si ce dernier a besoin d'un enregistrement, il peut interroger d'autres serveurs DNS via l'envoi d'une requête. Le cas échéant, le serveur DNS émet une requête en amont, jusqu'à ce qu'elle remonte à l'autorité du domaine. Le serveur faisant autorité est celui où les administrateurs fixent les noms de serveur et les adresses IP.

C'est dans ce « **serveur DNS maître** » qu'un administrateur peut changer le nom d'un serveur ou une adresse IP.

Référencement DNS

Le dépôt d'un [nom de domaine](#) se fait dans un « bureau d'enregistrement », nommé « registrar » en anglais.

Celui-ci fait alors la médiation entre les demandeurs de noms de domaine et l'Internet Corporation for Assigned Names and Numbers. Plus connue sous l'acronyme d'**ICANN**, cette société attribue des adresses IP à tous les sites du web.

Les étapes d'une recherche DNS

L'envoi d'une requête DNS se déroule comme suit :

- Une requête DNS est envoyée lorsque vous essayez de vous connecter à un site internet quelconque.
- L'ordinateur effectue l'analyse du cache du DNS local. Quand un site est visité, son adresse se trouve à cet endroit.
- Si l'adresse IP n'est pas dans le cache, le DNS consulte un serveur DNS récursif (de la zone DNS associée).
- Le serveur DNS récursif a son propre cache. Si ce dernier a un l'adresse IP, il l'envoie immédiatement.
- Si l'adresse IP n'est pas trouvée, des serveurs de noms TLD peuvent envoyer la requête DNS dans la bonne direction.
- Les serveurs de noms TLD identifient l'emplacement du serveur de nom gérant le site recherché. Ce dernier répond avec l'adresse IP demandée. Le serveur DNS récursif la stocke dans le cache DNS local, avant de l'envoyer à votre ordinateur.
- Le DNS enregistre l'adresse IP dans le cache local, avec une durée de vie prédéfinie (appelée TTL). Cette durée de validité fixe l'enregistrement de l'adresse IP dans le cache. Quand celle-ci est expirée, la requête pour le même site suit de nouveau ce cheminement.

Les limites du DNS

Même s'il est indispensable au bon fonctionnement des connexions internet, le DNS présente quelques faiblesses :

- La première est que les serveurs DNS internes stockent les noms et les adresses IP des serveurs de leur domaine. Ainsi, chaque serveur DNS partage ces informations avec tout utilisateur qui en fait la requête. Cela fait donc de lui (et de toute zone DNS) une véritable source d'informations pour les pirates informatiques.
- Les caches DNS sont manipulables car ils ne font pas autorité. Ainsi, si votre serveur DNS est piraté, votre ordinateur peut être envoyé dans de mauvaises directions. Il y a donc risque d'orientation vers des sites malveillants.
- Le DNS transmet des informations de requêtes de stations internes vers des serveurs externes. Sachant cela, des pirates informatiques créent des « canaux cachés », pour exfiltrer les données privées des utilisateurs.