

SERVEUR DNS

(Domain Name System)

Encadré par:

☐ Mr MEHDI MOUKHAFI

Réalisée par:

☐ RANIA DIYANI
☐ FATIMA-ZAHRAE
OUANDOURI

Le plan

➤ Introduction

- 0 C'est quoi un réseau informatique?
- 0 La transmettre des informations
- 0 Les couches d'un réseau informatique

➤ Le serveur DNS

➤ L'histoire du serveur DNS

➤ L'hierarchie du serveur

DNS

➤ Le fonctionnement du serveur DNS

➤ La cache DNS

➤ Les enregistrements

Introduction au réseau informatique

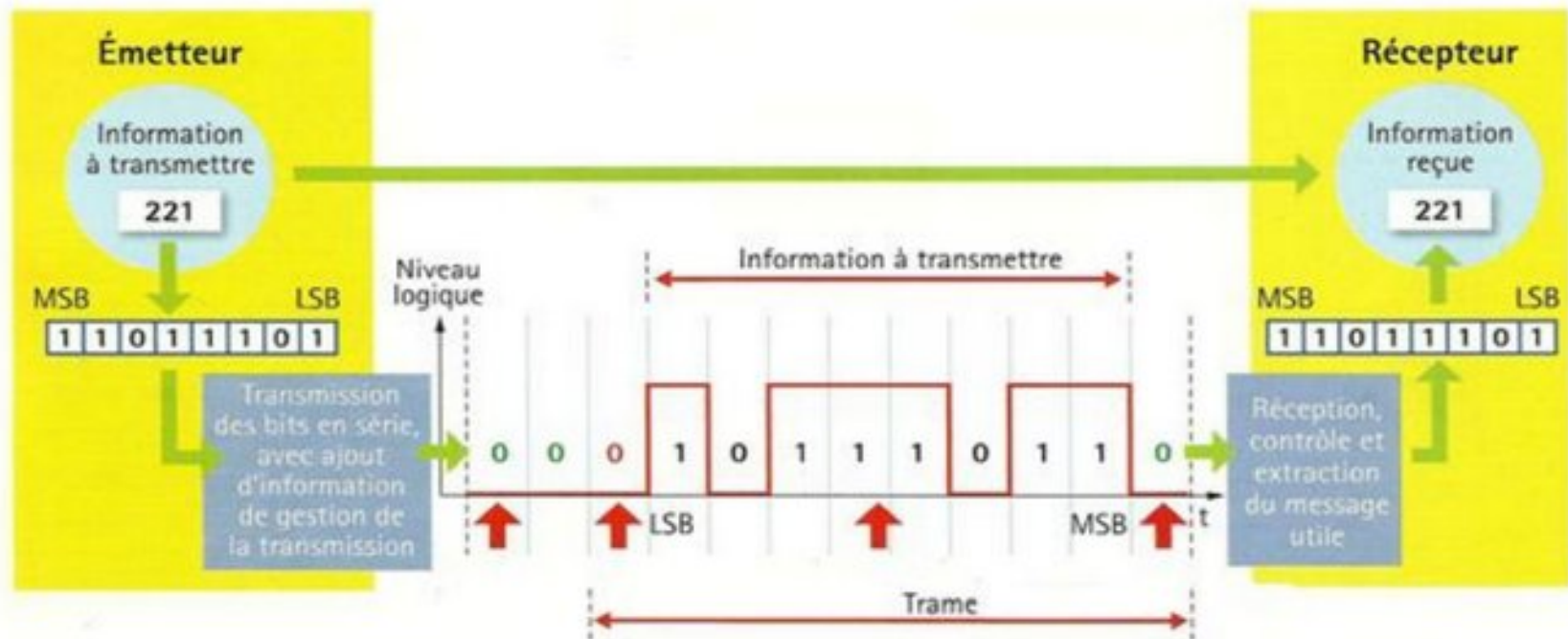
Définition d'un réseau informatique:

Le réseau informatique désigne les appareils informatiques interconnectés qui peuvent échanger des données et partager des ressources entre eux. Ces appareils en réseau utilisent un système de règles, appelées protocoles de communication, pour transmettre des informations sur des technologies physiques ou sans fil.

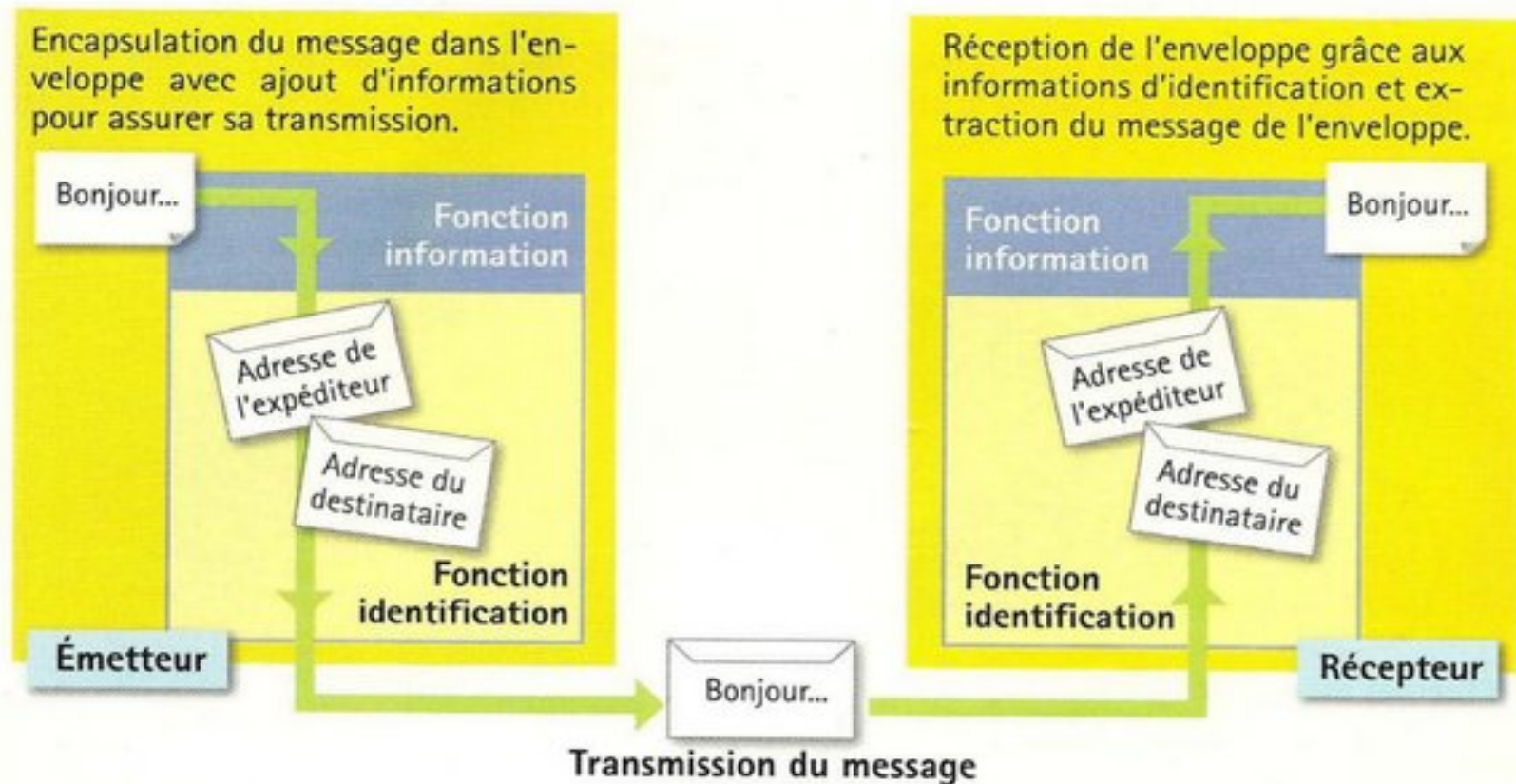


TRANSMETTRE DES INFORMATIONS:

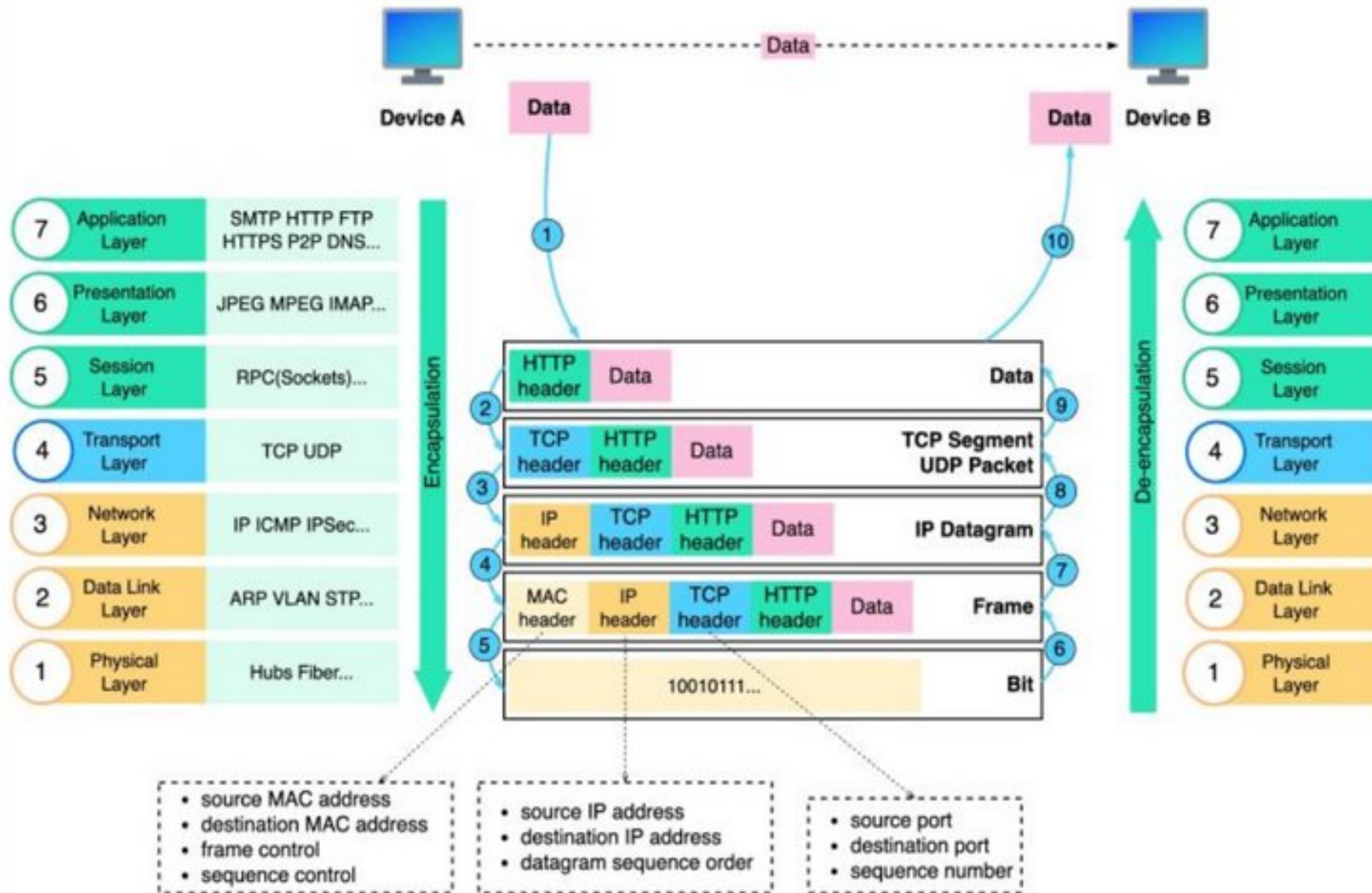
Quelque soit le réseau, le principe de transmission de l'information reste le même : les données numériques sont envoyées les une après les autres en série, sous forme d'une succession de bits, à une cadence donnée.



A cette information que l'on doit envoyer, il est nécessaire d'en ajouter d'autres qui vont gérer la bonne marche de la transmission. On rajoutera au minimum : le début de la transmission, bonne transmission et fin de transmission. Ces informations là ne seront prises en compte que par les interfaces qui prennent charge l'émission et réception des données .



Les couches de modèle OSI:



Les couches d'un réseau informatique :

Le modèle OSI :

Les couches du modèle OSI (Open System Interconnections) sont conçues de telle sorte qu'une application puisse communiquer sur un réseau avec une autre application sur un appareil différent, quelle que soit la complexité de l'application et des systèmes sous-jacents.

Le modèle TCP/IP :

Le modèle TCP/IP définit la manière dont les périphériques doivent transmettre des données entre eux et facilite la communication sur les réseaux et les grandes distances. Le modèle décrit la manière dont les données sont échangées et organisées sur les réseaux.

Modèle OSI

Couche 7 : application

Couche 6 : présentation

Couche 5 : session

Couche 4 : transport

Couche 3 : réseau

Couche 2 : liaison de données

Couche 1 : physique

Modèle TCP/IP

Application

Transport

Internet

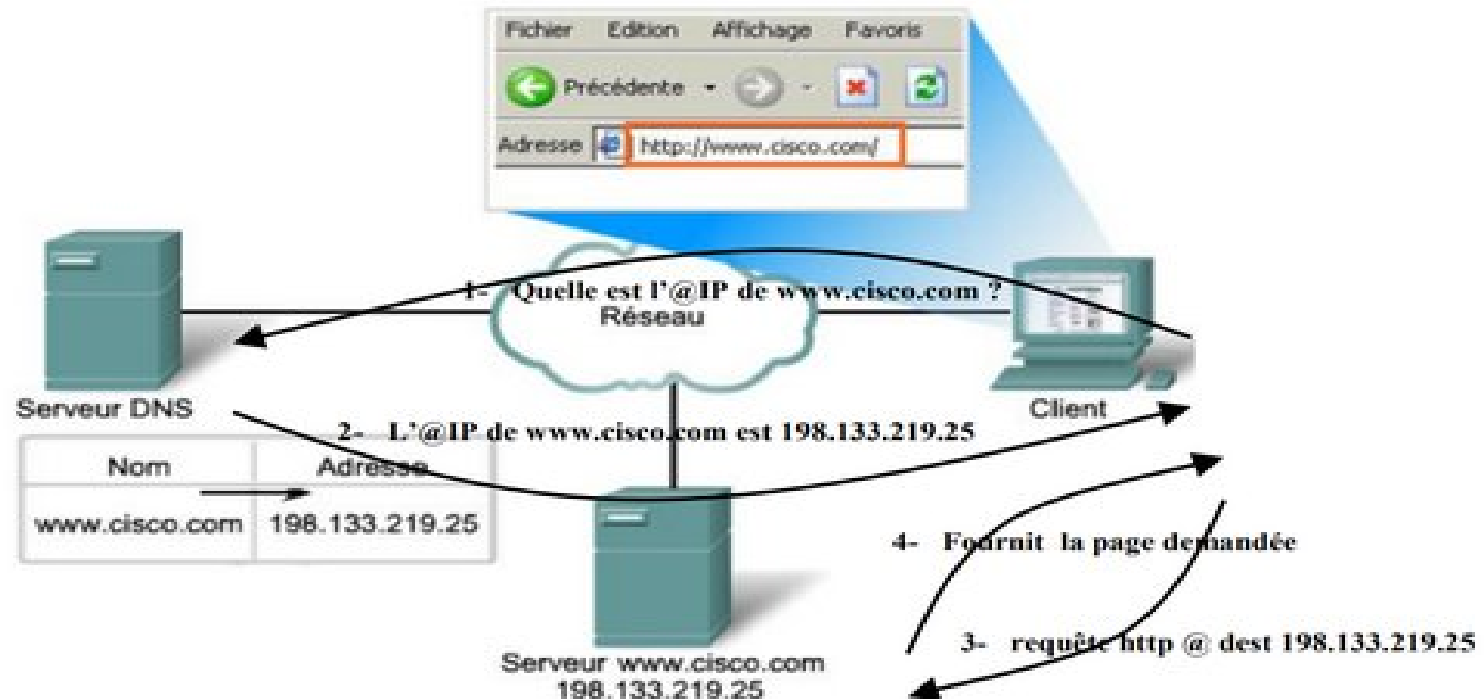
Accès réseau

LE SERVEUR DNS

C'est quoi un serveur DNS:

Un serveur DNS, abréviation de "Domain Name System", est un composant crucial d'Internet. Il agit comme un annuaire qui traduit les noms de domaine compréhensibles par les humains (comme example.com) en adresses IP, qui sont les identifiants numériques utilisés par les ordinateurs pour localiser les serveurs et les sites web sur Internet.

En résumé, il convertit les noms de domaine en adresses IP pour permettre la navigation sur le web.



Historique du serveur DNS :

Au fil des années, l'histoire du serveur DNS a été marquée par une évolution constante pour répondre aux besoins croissants et à la complexité grandissante d'Internet :

- ❑ **Années 1980 :** Le DNS est conçu pour remplacer le fichier "HOSTS.TXT" qui contenait manuellement les correspondances entre noms de domaine et adresses IP. Le DNS a été déployé pour la première fois en 1983, apportant une solution décentralisée et plus évolutive pour résoudre les noms de domaine en adresses IP.
- ❑ **Années 1990 :** Avec l'essor rapide d'Internet et l'explosion du nombre de noms de domaine, des améliorations ont été apportées pour optimiser les performances du DNS. Des concepts tels que les serveurs de cache ont été introduits pour réduire la charge des serveurs principaux et accélérer la résolution des requêtes DNS.

- ❑ **Années 2000** : La sécurité du DNS est devenue une préoccupation majeure. Des protocoles comme DNSSEC (Domain Name System Security Extensions) ont été développés pour renforcer la sécurité et empêcher les attaques de détournement ou de falsification de données DNS.
- ❑ **Années 2010** : L'adoption de IPv6 a demandé des ajustements au DNS pour prendre en charge cette nouvelle version du protocole Internet, offrant un plus grand espace d'adressage et un support pour la croissance continue d'appareils connectés à Internet.
- ❑ **Années récentes** : L'optimisation des performances, l'amélioration de la résilience et la mise en œuvre de nouvelles technologies pour accélérer la résolution des noms de domaine font partie des préoccupations actuelles. De plus, des efforts continus sont déployés pour garantir la sécurité et l'intégrité des données DNS, en particulier face à des menaces de plus en plus sophistiquées.

Hiérarchie du DNS :

La hiérarchie des serveurs DNS (Domain Name System) suit une structure en couches organisée de manière pyramidale pour faciliter la résolution des noms de domaine en adresses IP.

- ❑ **Serveurs Racine (Root Servers) :** Ces serveurs constituent le sommet de la hiérarchie. Ils sont au nombre de 13 dans le monde et portent des lettres de A à M. Les serveurs racine contiennent des informations sur les serveurs de domaines de premier niveau (TLD) pour chaque extension de domaine (.com, .org, .net, etc.). Ils redirigent les requêtes vers les serveurs de domaines de premier niveau appropriés.

- ❑ **Serveurs de Domaines de Premier Niveau (Top-Level Domain Servers - TLD) :** Ces serveurs gèrent les domaines de premier niveau (.com, .org, .net, etc.). Chaque extension de domaine a ses propres serveurs TLD. Ils stockent des informations sur les serveurs autoritaires pour les domaines de deuxième niveau.

Serveurs Autoritaires de Domaines (Authoritative Name Servers) :

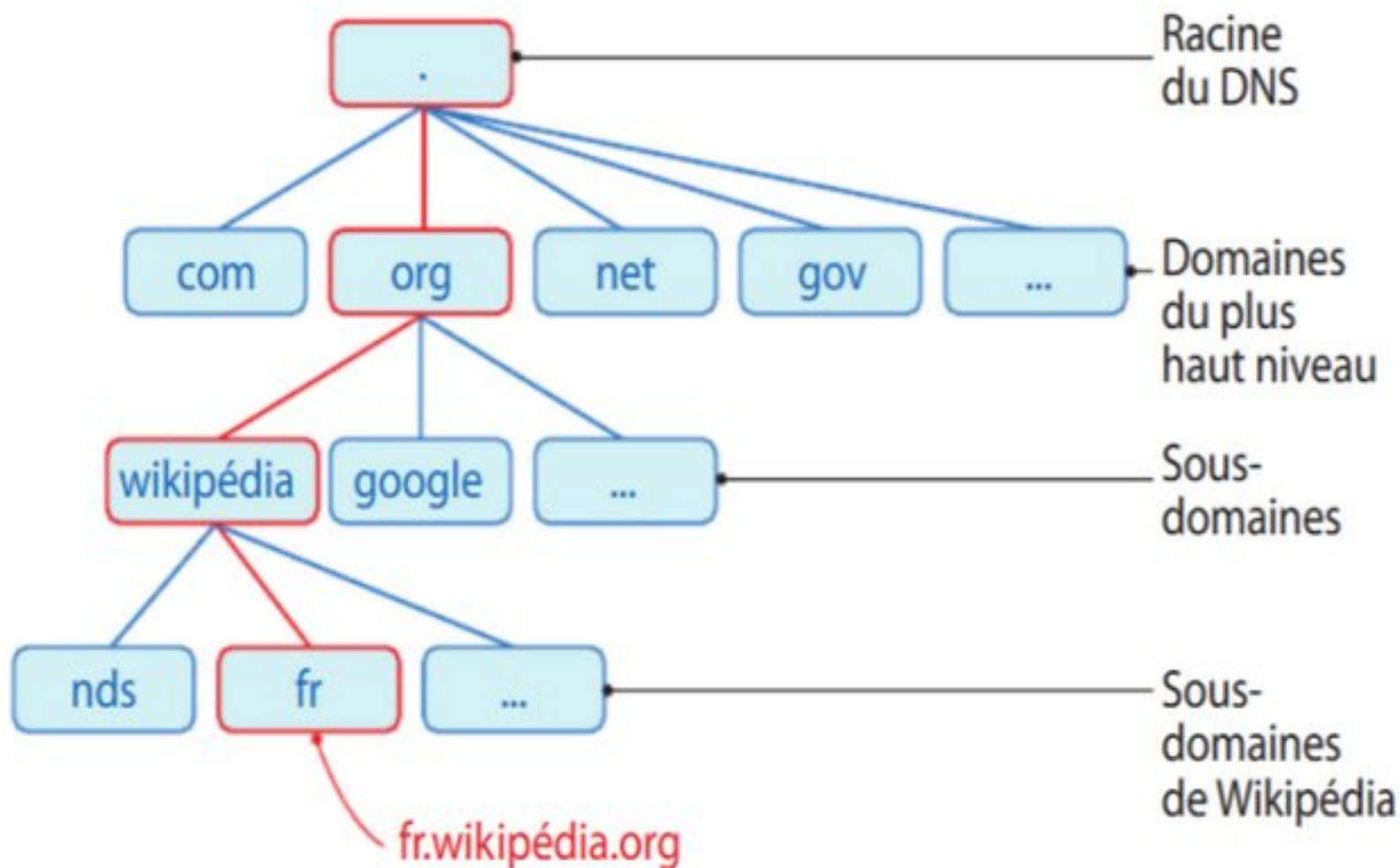
Ces serveurs sont responsables des domaines spécifiques. Chaque domaine possède ses propres serveurs autoritaires qui contiennent des informations détaillées sur ce domaine, telles que les adresses IP associées à des noms de domaine spécifiques.

Serveurs de Cache ou Résolveurs DNS (Recursive DNS Servers) :

Bien qu'ils ne soient pas formellement dans la structure hiérarchique, ces serveurs jouent un rôle crucial dans la résolution des noms de domaine. Ils reçoivent les requêtes des utilisateurs, effectuent des recherches en interrogeant d'autres serveurs DNS, et stockent temporairement ces informations pour accélérer les futures requêtes.

Lorsqu'un utilisateur envoie une requête pour accéder à un site web en utilisant un nom de domaine, cette requête est acheminée à travers cette hiérarchie, en commençant par les serveurs racine, puis en descendant vers les niveaux inférieurs jusqu'à ce que la réponse soit trouvée. Cette organisation hiérarchique permet une résolution efficace des noms de domaine sur Internet.

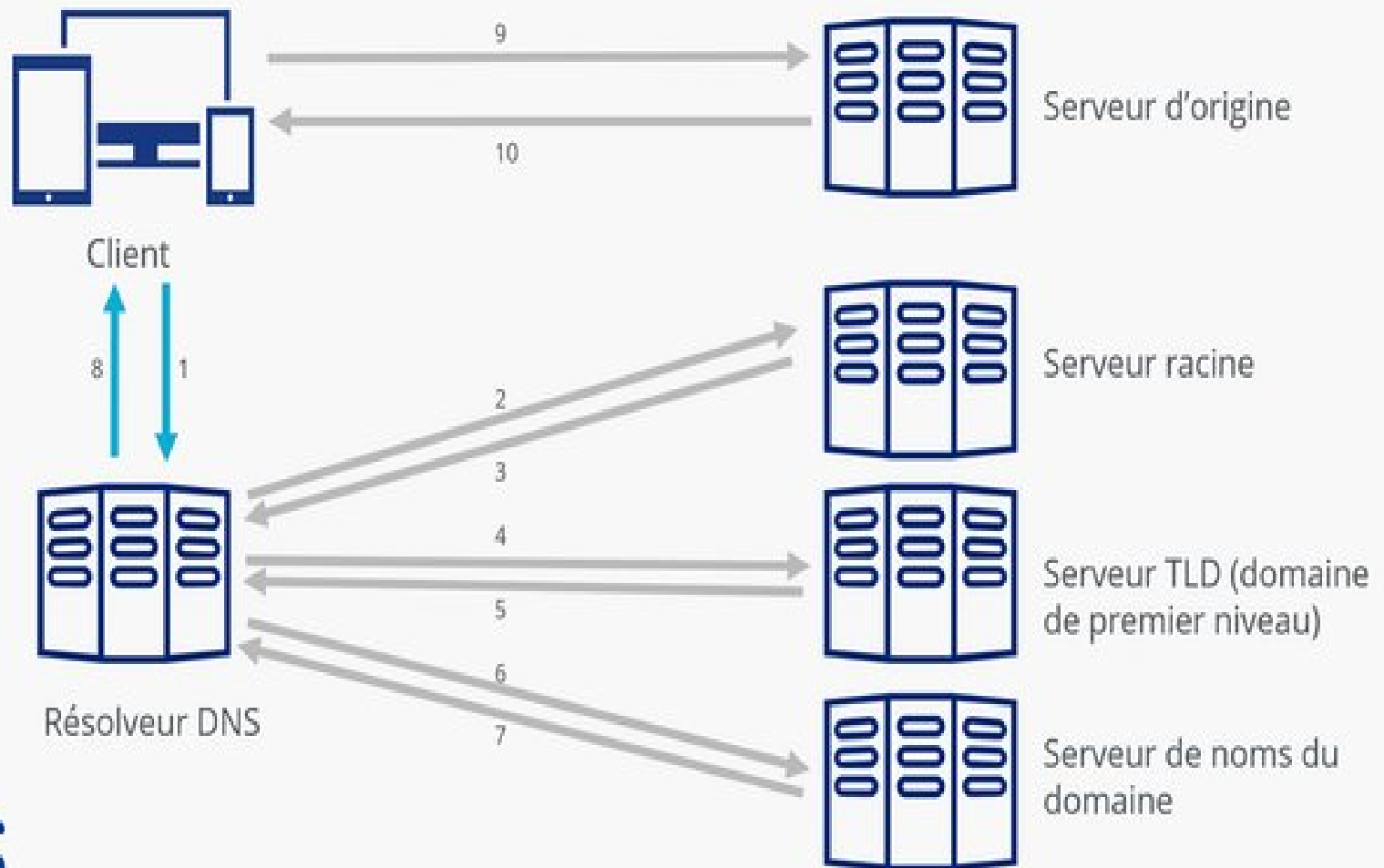
Organisation hiérarchique de l'annuaire DNS



Le fonctionnement de serveur DNS:

- ❑ **Requête DNS** : Lorsqu'un utilisateur tape une URL dans son navigateur (par exemple, www.example.com), son appareil envoie une requête DNS pour traduire ce nom de domaine en une adresse IP utilisable par les machines.
- ❑ **Interrogation du DNS** : La requête DNS est d'abord envoyée aux serveurs DNS récursifs gérés par le FAI (Fournisseur d'Accès Internet) ou d'autres fournisseurs de services Internet. S'il a déjà l'information en cache, il la renvoie directement à l'appareil. Sinon, il initie une recherche pour trouver la réponse à la requête.
- ❑ **Hiérarchie DNS** : La requête est envoyée aux serveurs racine, qui la redirigent vers les serveurs de domaines de premier niveau (TLD - Top-Level Domains). Ces derniers la renvoient vers les serveurs autoritaires pour le domaine spécifique recherché.
- ❑ **Réponse DNS** : Les serveurs autoritaires pour le domaine fournissent l'adresse IP associée au nom de domaine demandé, et cette réponse est renvoyée par le serveur DNS récursif au résolveur DNS de l'appareil de l'utilisateur.
- ❑ **Mise en cache** : Une fois la réponse obtenue, elle est temporairement stockée en cache au niveau des serveurs DNS récursifs pour accélérer les futures requêtes pour ce domaine.
- ❑ **Transmission de la réponse** : Enfin, l'adresse IP obtenue est transmise à l'appareil de l'utilisateur.

Comment un serveur DNS résout-il une requête DNS ?



La cache DNS :

La cache DNS est une mémoire temporaire utilisée par les serveurs DNS pour stocker les enregistrements de recherche de noms de domaine auxquels ils ont récemment accédé.

Il existe deux types de caches DNS :

- **Cache du côté client** : Les systèmes d'exploitation et les navigateurs web peuvent également conserver des informations DNS en cache pour accélérer les futures requêtes DNS. Ces caches sont situés sur les appareils des utilisateurs.
- **Cache du côté serveur** : Chaque serveur DNS, qu'il soit autoritaire (qui stocke les informations sur des zones spécifiques) ou récursif (qui recherche des informations pour les clients), possède son propre cache DNS. Ce cache stocke temporairement les informations des requêtes DNS récentes.

Les enregistrements:

- **A record** ou **address record** qui fait correspondre un nom d'hôte à une adresse IPv4 de 32 bits distribués sur quatre octets ex: 123.234.1.2 .
- **AAAA record** ou **IPv6 address record** qui fait correspondre un nom d'hôte à une adresse IPv6 de 128 bits distribués sur seize octets.
- **CNAME record** ou **canonical name record** qui permet de faire d'un domaine un alias vers un autre. Cet alias hérite de tous les sous-domaines de l'original.
- **MX record** ou **mail exchange record** est utilisé pour spécifier les serveurs de messagerie responsables de recevoir les courriers électroniques destinés à un domaine particulier. Cet enregistrement indique la priorité et l'adresse du serveur de messagerie associé.

Les enregistrements:

- **SOA record** ou **Start Of Authority record** qui donne les informations générales de la zone : serveur principal, courriel de contact, différentes durées dont celle d'expiration, numéro de série de la zone.
- **SRV record** qui généralise la notion de **MX record**, mais qui propose aussi des fonctionnalités avancées comme le taux de répartition de charge pour un service donné. **PTR record** ou **pointer record** qui associe une adresse IP à un enregistrement de nom de domaine, aussi dit « *reverse* » puisque il fait exactement le contraire du A record.
- **NS record** ou **name server record** qui définit les serveurs DNS de ce domaine.

C'est quoi le Bind9

BIND9 (Berkeley Internet Name Domain version 9) est un logiciel serveur DNS open-source populaire et largement utilisé.

IL agit comme une infrastructure de base pour la résolution des noms de domaine sur Internet. Il prend en charge la résolution des requêtes DNS en convertissant les noms de domaine (comme www.example.com) en adresses IP et gère la distribution des enregistrements DNS.

L'installation de serveur DNS:

```
[liveuser@localhost-live ~]$ sudo su
[root@localhost-live liveuser]# rpm -aq | grep -i bind
bind-license-9.18.13-1.fc38.noarch
bind-libs-9.18.13-1.fc38.x86_64
bind-utils-9.18.13-1.fc38.x86_64
rpcbind-1.2.6-4.rc2.fc38.x86_64
keybinder3-0.3.2-15.fc38.x86_64
```

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=112 time=59.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=112 time=62.2 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=112 time=62.3 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 3004ms
rtt min/avg/max/mdev = 59.405/61.312/62.284/1.348 ms
```

```
[root@localhost-live liveuser]# dnf install bind
Fedora 38 openh264 (From Cisco) - x86_64      137 B/s | 2.5 kB      00:18
Fedora Modular 38 - x86_64                   350 kB/s | 2.8 MB      00:08
Fedora 38 - x86_64 - Updates                  566 kB/s | 36 MB       01:04
Fedora Modular 38 - x86_64 - Updates          88 kB/s | 2.1 MB       00:24
Last metadata expiration check: 0:00:01 ago on Wed 13 Dec 2023 09:53:33 AM EST.
Dependencies resolved.
=====
Package                        Architecture Version                Repository            Size
=====
Installing:
bind                           x86_64      32:9.18.20-1.fc38      updates               528 k
Upgrading:
bind-libs                      x86_64      32:9.18.20-1.fc38      updates               1.3 M
bind-license                   noarch      32:9.18.20-1.fc38      updates               14 k
bind-utils                     x86_64      32:9.18.20-1.fc38      updates               224 k
Installing weak dependencies:
bind-dnssec-utils              x86_64      32:9.18.20-1.fc38      updates               148 k

Transaction Summary
=====
Install 2 Packages
Upgrade 3 Packages

Total download size: 2.2 M
Is this ok [y/N]: Y
Downloading Packages:
(1/5): bind-dnssec-utils-9.18.20-1.fc38.x86_64.rpm      76 kB/s | 148 kB      00:01
(2/5): bind-license-9.18.20-1.fc38.noarch.rpm           48 kB/s | 14 kB        00:00
(3/5): bind-utils-9.18.20-1.fc38.x86_64.rpm            194 kB/s | 224 kB      00:01
(4/5): bind-9.18.20-1.fc38.x86_64.rpm                  153 kB/s | 528 kB      00:03
(5/5): bind-libs-9.18.20-1.fc38.x86_64.rpm             200 kB/s | 1.3 MB       00:06
-----
Total                                                    271 kB/s | 2.2 MB       00:08
Fedora 38 - x86_64 - Updates                            115 kB/s | 1.6 kB       00:00
Importing GPG key 0xEB10B464:
  Userid      : "Fedora (38) <fedora-38-primary@fedoraproject.org>"
  Fingerprint: 6A51 BBAB BA3D 5467 B617 1221 809A 8D7C EB10 B464
  From        : /etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-38-x86_64
Is this ok [y/N]: Y
```

```

Is this ok [y/N]: Y
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Upgrading      : bind-license-32:9.18.20-1.fc38.noarch 1/8
  Upgrading      : bind-libs-32:9.18.20-1.fc38.x86_64  2/8
  Upgrading      : bind-utils-32:9.18.20-1.fc38.x86_64 3/8
  Installing     : bind-dnssec-utils-32:9.18.20-1.fc38.x86_64 4/8
  Running scriptlet: bind-32:9.18.20-1.fc38.x86_64      5/8
  Installing     : bind-32:9.18.20-1.fc38.x86_64      5/8
  Running scriptlet: bind-32:9.18.20-1.fc38.x86_64      5/8
  Cleanup        : bind-utils-32:9.18.13-1.fc38.x86_64 6/8
  Cleanup        : bind-libs-32:9.18.13-1.fc38.x86_64 7/8
  Cleanup        : bind-license-32:9.18.13-1.fc38.noarch 8/8
  Running scriptlet: bind-license-32:9.18.13-1.fc38.noarch 8/8
  Verifying      : bind-32:9.18.20-1.fc38.x86_64      1/8
  Verifying      : bind-dnssec-utils-32:9.18.20-1.fc38.x86_64 2/8
  Verifying      : bind-libs-32:9.18.20-1.fc38.x86_64 3/8
  Verifying      : bind-libs-32:9.18.13-1.fc38.x86_64 4/8
  Verifying      : bind-license-32:9.18.20-1.fc38.noarch 5/8
  Verifying      : bind-license-32:9.18.13-1.fc38.noarch 6/8
  Verifying      : bind-utils-32:9.18.20-1.fc38.x86_64 7/8
  Verifying      : bind-utils-32:9.18.13-1.fc38.x86_64 8/8

Upgraded:
  bind-libs-32:9.18.20-1.fc38.x86_64      bind-license-32:9.18.20-1.fc38.noarch
  bind-utils-32:9.18.20-1.fc38.x86_64

Installed:
  bind-32:9.18.20-1.fc38.x86_64      bind-dnssec-utils-32:9.18.20-1.fc38.x86_64

Complete!
[root@localhost-live liveuser]#

```

```
[root@localhost-live liveuser]# rpm -aq | grep -i bind
rpcbind-1.2.6-4.rc2.fc38.x86_64
keybinder3-0.3.2-15.fc38.x86_64
bind-license-9.18.20-1.fc38.noarch
bind-libs-9.18.20-1.fc38.x86_64
bind-utils-9.18.20-1.fc38.x86_64
bind-dnssec-utils-9.18.20-1.fc38.x86_64
bind-9.18.20-1.fc38.x86_64
[root@localhost-live liveuser]#
```

```
[root@localhost-live liveuser]# hostname
localhost-live
[root@localhost-live liveuser]# hostname serveurDNS
[root@localhost-live liveuser]# hostname
serveurDNS
[root@localhost-live liveuser]#
```

```
[root@localhost-live liveuser]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:46:bd:ea brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 81985sec preferred_lft 81985sec
    inet6 fe80::27eb:2660:c63b:ab95/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@localhost-live liveuser]#
```

```
[root@localhost-live liveuser]# vi /etc/named.conf
```

```
//
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind-*/sample/ for example named configuration files.
//

options {
    listen-on port 53 { 127.0.0.1; 10.0.2.15; };
    listen-on-v6 port 53 { ::1; };
    directory "/var/named";
    dump-file "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    secroots-file "/var/named/data/named.secroots";
    recursing-file "/var/named/data/named.recursing";
    allow-query { localhost; 10.0.2.15/24; any; };

    /*
     * If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.
     * If you are building a RECURSIVE (caching) DNS server, you need to enable
     * recursion.
     * If your recursive DNS server has a public IP address, you MUST enable access
     * control to limit queries to your legitimate users. Failing to do so will
     * cause your server to become part of large scale DNS amplification
     * attacks. Implementing BCP38 within your network would greatly
     * reduce such attack surface
     */
    recursion yes;

    dnssec-validation yes;

    managed-keys-directory "/var/named/dynamic";
    geoip-directory "/usr/share/GeoIP";

    pid-file "/run/named/named.pid";
    session-keyfile "/run/named/session.key";
}
```

-- INSERT --

1,3

Top


```
logging {  
    channel default_debug {  
        file "data/named.run";  
        severity dynamic;  
    };  
};  
  
zone "." IN {  
    type hint;  
    file "named.ca";  
};  
  
include "/etc/named.rfc1912.zones";  
include "/etc/named.root.key";
```

```
[root@localhost-live liveuser]# vi /etc/named.conf
[root@localhost-live liveuser]# vi /etc/named.rfc1912.zones
```

```
// named.rfc1912.zones:  
//  
// Provided by Red Hat caching-nameserver package  
//  
// ISC BIND named zone configuration for zones recommended by  
// RFC 1912 section 4.1 : localhost TLDs and address zones  
// and https://tools.ietf.org/html/rfc8363  
// (c)2007 R W Franks  
//  
// See /usr/share/doc/bind-*/sample/ for example named configuration files.  
//  
// Note: empty-zones-enable yes; option is default.  
// If private ranges should be forwarded, add  
// disable-empty-zone "."; into options  
//  
  
zone "localhost.localdomain" IN {  
    type primary;  
    file "named.localhost";  
    allow-update { none; };  
};  
  
zone "localhost" IN {  
    type primary;  
    file "named.localhost";  
    allow-update { none; };  
};  
  
zone "1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa" IN {  
    type primary;  
    file "named.loopback";  
    allow-update { none; };  
};  
  
zone "1.0.0.127.in-addr.arpa" IN {  
    type primary;  
    file "named.loopback";  
    allow-update { none; };  
};
```

La déclaration de zone directe:

```
zone "." IN {  
    type hint;  
    file "named.ca";  
};  
zone "unv.ens.info" IN {  
    type primary;  
    file "f.unv.ens.info.dir";  
    allow-update { none; };  
};
```

```
[root@localhost-live liveuser]# cd /var/named/  
[root@localhost-live named]# touch f.unv.ens.info.dir  
[root@localhost-live named]# ls  
data      f.unv.ens.info.dir  named.empty      named.loopback  
dynamic   named.ca             named.localhost  slaves  
[root@localhost-live named]# cat named.localhost  
$TTL 1D  
@          IN SOA  @ rname.invalid. (  
                                0          ; serial  
                                1D         ; refresh  
                                1H         ; retry  
                                1W         ; expire  
                                3H )       ; minimum  
  
    NS      @  
    A       127.0.0.1  
    AAAA    ::1  
[root@localhost-live named]# vi f.unv.ens.info.dir
```

```

$TTL 1D
@      IN      SOA      serveurDNS.univ.ens.info.      admin.univ.ens.info. (
                                0      ; serial
                                1D     ; refresh
                                1H     ; retry
                                1W     ; expire
                                3H )   ; minimum
@      IN      NS       serveurDNS
serveurDNS  IN      A      10.0.2.15
pc         IN      A      10.0.2.1
pc1        IN      A      10.0.2.2
serveur-web IN      A      10.0.2.5
www        IN      CNAME   serveur-web

```

```

[root@localhost-live named]# vi f.unv.ens.info.dir
[root@localhost-live named]# vi f.unv.ens.info.dir
[root@localhost-live named]# ll
total 32
drwxrwx---. 2 named named 4096 Nov 15 19:00 data
drwxrwx---. 2 named named 4096 Nov 15 19:00 dynamic
-rw-r--r--. 1 root root 312 Dec 13 11:01 f.unv.ens.info.dir
-rw-r-----. 1 root named 3312 Nov 15 19:00 named.ca
-rw-r-----. 1 root named 152 Nov 15 19:00 named.empty
-rw-r-----. 1 root named 152 Nov 15 19:00 named.localhost
-rw-r-----. 1 root named 168 Nov 15 19:00 named.loopback
drwxrwx---. 2 named named 4096 Nov 15 19:00 slaves
[root@localhost-live named]# chgrp named f.unv.ens.info.dir
[root@localhost-live named]# ll
total 32
drwxrwx---. 2 named named 4096 Nov 15 19:00 data
drwxrwx---. 2 named named 4096 Nov 15 19:00 dynamic
-rw-r--r--. 1 root named 312 Dec 13 11:01 f.unv.ens.info.dir
-rw-r-----. 1 root named 3312 Nov 15 19:00 named.ca
-rw-r-----. 1 root named 152 Nov 15 19:00 named.empty
-rw-r-----. 1 root named 152 Nov 15 19:00 named.localhost
-rw-r-----. 1 root named 168 Nov 15 19:00 named.loopback
drwxrwx---. 2 named named 4096 Nov 15 19:00 slaves
[root@localhost-live named]#

```

```
[root@localhost-live named]# named-checkconf /etc/named.conf
[root@localhost-live named]# systemctl status named.service
* named.service - Berkeley Internet Name Domain (DNS)
   Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; preset: dis>
   Drop-In: /usr/lib/systemd/system/service.d
           └─10-timeout-abort.conf
   Active: failed (Result: exit-code) since Wed 2023-12-13 11:43:14 EST; 18mi>
     CPU: 36ms

Dec 13 11:43:14 localhost-live systemd[1]: Starting named.service - Berkeley In>
Dec 13 11:43:14 localhost-live bash[35843]: /etc/named.conf:19: '10.0.2.15/24':>
Dec 13 11:43:14 localhost-live systemd[1]: named.service: Control process exite>
Dec 13 11:43:14 localhost-live systemd[1]: named.service: Failed with result 'e>
Dec 13 11:43:14 localhost-live systemd[1]: Failed to start named.service - Berk>

[root@localhost-live named]# named-checkzone unv.ens.info f.unv.ens.info.dir
zone unv.ens.info/IN: loaded serial 0
OK
```

```
[root@localhost-live named]# systemctl enable named.service
[root@localhost-live named]# systemctl status named.service
```

```
[root@localhost-live named]# systemctl status named.service
● named.service - Berkeley Internet Name Domain (DNS)
   Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/service.d
            └─10-timeout-abort.conf
   Active: active (running) since Wed 2023-12-13 12:05:49 EST; 7h ago
     Main PID: 36559 (named)
        Tasks: 4 (limit: 2257)
       Memory: 5.7M
          CPU: 1.770s
       CGroup: /system.slice/named.service
              └─36559 /usr/sbin/named -u named -c /etc/named.conf
```

```
Dec 13 19:05:50 localhost-live named[36559]: network unreachable resolving './DNSKEY/IN': 198.51.100.100
Dec 13 19:05:50 localhost-live named[36559]: network unreachable resolving './NS/IN': 198.51.100.100
Dec 13 19:05:50 localhost-live named[36559]: network unreachable resolving './DNSKEY/IN': 199.192.192.100
Dec 13 19:05:50 localhost-live named[36559]: network unreachable resolving './NS/IN': 199.192.192.100
Dec 13 19:05:50 localhost-live named[36559]: network unreachable resolving './DNSKEY/IN': 192.168.1.100
Dec 13 19:05:50 localhost-live named[36559]: network unreachable resolving './NS/IN': 192.168.1.100
Dec 13 19:05:50 localhost-live named[36559]: network unreachable resolving './DNSKEY/IN': 192.168.1.100
Dec 13 19:05:50 localhost-live named[36559]: network unreachable resolving './NS/IN': 192.168.1.100
Dec 13 19:05:50 localhost-live named[36559]: managed-keys-zone: Unable to fetch DNSKEY set '192.168.1.100'
Dec 13 19:05:50 localhost-live named[36559]: resolver priming query complete: failure
```



```
[root@localhost-live named]# nslookup
```

```
> serveurDNS.univ.ens.info.
```

```
Server:          127.0.0.53
```

```
Address:         127.0.0.53#53
```

```
Non-authoritative answer:
```

```
Name:   serveurDNS.univ.ens.info
```

```
Address: 10.0.2.15
```