

IUT DE COLMAR

R303

ANNÉE 2022-23

Services réseaux avancés

MARTIN BAUMGAERTNER

16 septembre 2022

Table des matières

1	CM 1 - 6 septembre 2022	3
1.1	Rappels	3
1.2	Les DNS	4
1.2.1	Fonctionnement du DNS	4
1.3	Créer des raccourcis (alias)	5
1.3.1	Petit exercice pour vendredi prochain	5
2	CM 2 - 9 septembre 2022	6
2.1	Les Mails	6
2.1.1	Contenu d'un mail	6
2.1.2	Pertes des données	6
3	TD 1 - 9 septembre 2022	7
3.1	Exercice 1	7
3.1.1	Algorithme itératif (str, @R)	8
3.1.2	Algorithme récursif (str, @R)	8
4	TP 1 - 13 septembre 2022	9
4.1	Environnement	9
4.1.1	Exercice 1	9
4.2	Configuration du client	9
4.2.1	Exercice 2	9
4.2.2	Exercice 3	9
4.3	Configuration du serveur	9
4.3.1	Exercice 4	9
4.3.2	Exercice 5	9
4.3.3	Exercice 6	9
4.3.4	Exercice 7	10
4.3.5	Exercice 8	11
5	TP 2 - 16 septembre 2022	12
5.1	Environnement	12
5.1.1	Exercice 1	12
5.2	Configuration du client	12
5.2.1	Exercice 2	12
5.3	Mise en place du serveur	12
5.3.1	Exercice 3	12
5.3.2	Sur le serveur	12
5.3.3	Sur le client	13

5.3.4	Exercice 4	14
5.3.5	Exercice 5	14
5.3.6	Exercice 6	14
5.4	Mise en place du serveur SMTP	15

Table des figures

1	Le modèle OSI	3
2	Nombre de site web et d'internautes en fonction des années	4
3	La recherche itérative	7
4	La recherche récursive	7
5	fichier db.rt.iutcolmar.uha.fr	12
6	fichier named.conf.local	13
7	Fichier /etc/hostname	13
8	fichier /etc/hosts	13
9	fichier /etc/resolv.conf	13
10	Ping du client vers le serveur	14
11	fichier /etc/bind/db.rt.iutcolmar.uha.fr	14
12	Commande host -t MX rt.iutcolmar.uha.fr.	14

1 CM 1 - 6 septembre 2022

1.1 Rappels

TCP : permet d'avoir une communication fiable de bout en bout, on a une session qui permet de faire du recouvrement, s'il y a un paquet perdu on le retransmet, pour ça on utilise un numéro de séquence, et on a un mécanisme d'acknowledgment qui nous dit quel octet est arrivé et quel octet est perdu.

UDP : est utilisé lorsque l'accusé de réception des données n'a aucune signification, est un bon protocole pour les données circulant dans une seule direction, est simple et adapté aux communications basées sur des requêtes, n'est pas orienté connexion, ne fournit pas de mécanisme de contrôle de congestion.

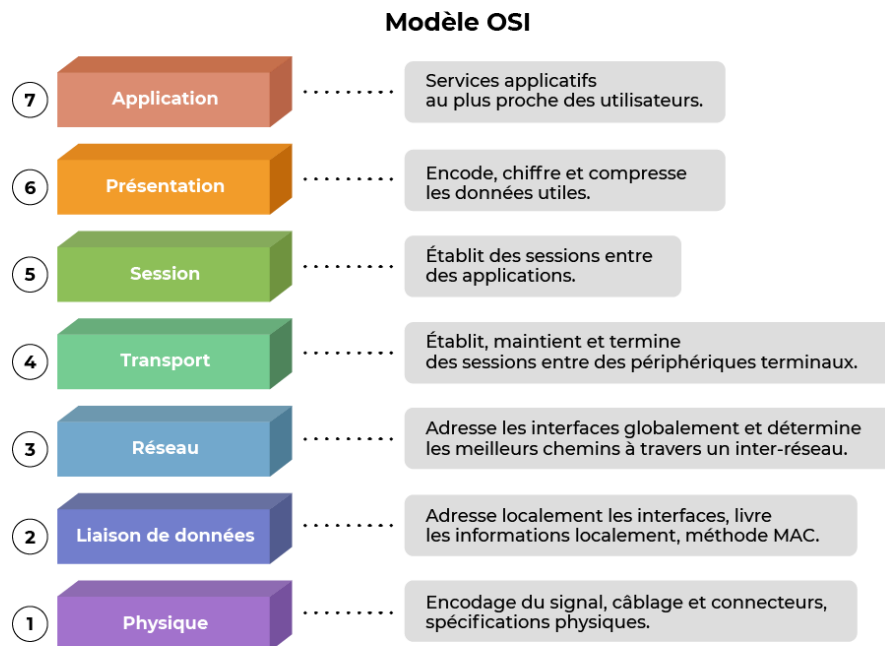


FIGURE 1 – Le modèle OSI

1.2 Les DNS

Le Domain Name System (DNS) est une base de données distribuée organisée de manière hiérarchique, un protocole applicatif qui permet à un hôte de l'interroger. Le port utilisé par défaut pour DNS est le port 53. L'objectif de ce système est d'identifier les hôtes grâce à un nom.

Le système DNS est souvent utilisé en amont des protocoles applicatifs comme HTTP ou les mails.

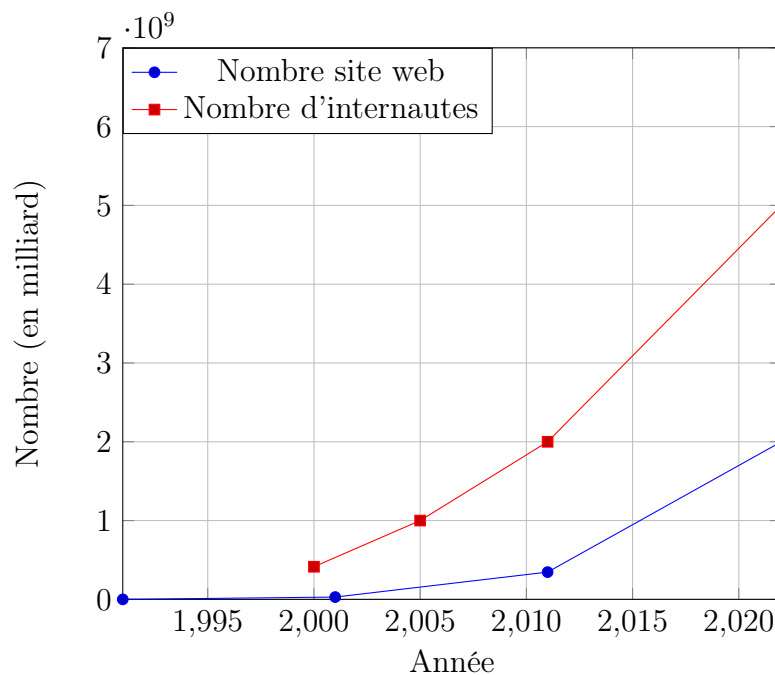


FIGURE 2 – Nombre de site web et d'internautes en fonction des années

1.2.1 Fonctionnement du DNS

Le DNS gère la distribution du trafic : distribue les informations reçus sur le site voulu sur les 2 adresses IP du site. Un serveur racine du DNS est un serveur DNS qui répond aux requêtes qui concernent les noms de domaines du premier niveau et qui les redirige vers le serveur DNS de premier niveau concerné. Bien qu'il puisse exister d'autres hiérarchies de systèmes de noms de domaine (DNS) avec des serveurs racine alternatifs, "serveur racine du DNS" est généralement utilisé pour désigner l'un des serveurs racine du DNS d'internet géré sous l'autorité de L'ICANN.

1.3 Créer des raccourcis (alias)

Un alias est un raccourcis dans l'URL d'un site web, on peut donc passer de `www.iutc078.uha.fr.iut.colmar.fr` à `www.iutc078.uha.fr`. La version complète se nomme version canonique.

1.3.1 Petit exercice pour vendredi prochain

Comment on gère un cache

Un cache est un ensemble de réponses de requêtes faites précédemment sur une page web et qui ont été préalablement recueillies. Une clé d'identification est rattachée à chacune de ces réponses, chaque clé est différente. Dans le cas où une requête correspond à une clé d'identification déjà identifiée, alors il fournit une réponse instantanée sans refaire appel au serveur d'origine. Il facilite donc votre navigation et augmente sa rapidité. Le gestionnaire de cache prend en charge les mises à jour ainsi que la suppression des réponses dans le stockage cache. Il est mis à la disposition des utilisateurs dans l'action lancée, avant de répandre la demande vers le serveur.

Considérons une liste de trois éléments (x_1, x_2, x_3) montrer que le traitement de la séquence par un algorithme optimal hors-ligne se fait avec un coup de 8.

2 CM 2 - 9 septembre 2022

2.1 Les Mails

- MUA : Mail User Agent, c'est le logiciel qui permet à l'utilisateur de consulter ses mails.
- MTA : Mail Transfer Agent, c'est le logiciel qui permet d'envoyer des mails.
- POP : Post Office Protocol, c'est un protocole qui permet de récupérer les mails sur un serveur.
- IMAP : Internet Message Access Protocol, c'est un protocole qui permet de récupérer les mails sur un serveur.
- MDA : Mail Delivery Agent, c'est un agent de distribution de mail.

2.1.1 Contenu d'un mail

Dans l'en-tête du mail, on retrouve le Return Push, la liste des serveurs SMTP. Ensuite, il y a une partie blanche en ASCII puis les données.

2.1.2 Pertes des données

Nous avons aussi des pertes dans un mail.
Les formats sans pertes de données pour un son, est le FLAC. Et pour les images c'est le PNG. Mais sur le texte il n'y a aucun algorithme de perte de données.

3 TD 1 - 9 septembre 2022

3.1 Exercice 1

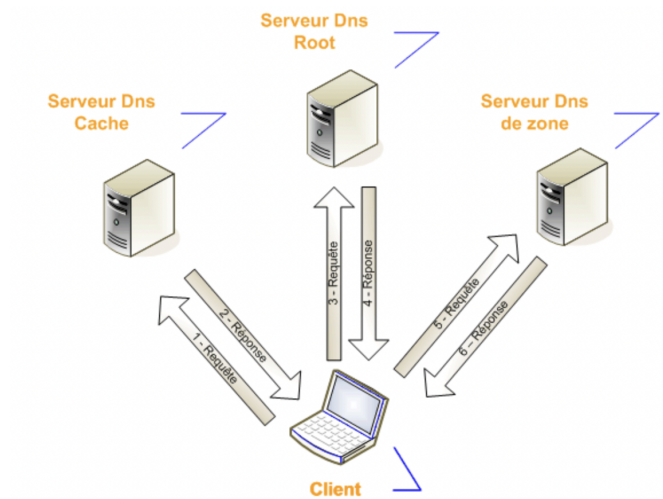


FIGURE 3 – La recherche itérative

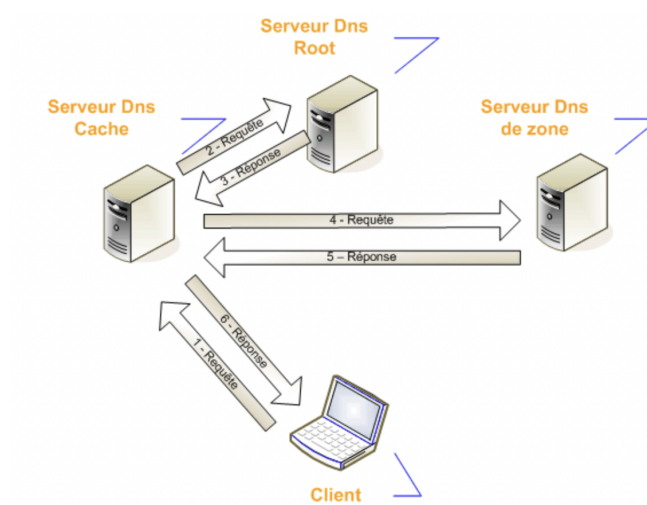


FIGURE 4 – La recherche récursive

Dans la version itérative : www.uha.fr/index.html on fait dans l'ordre la recherche de ces éléments : (on va venir demander au résolveur ces informations dans l'ordre)

- 1. `www`
- 2. `uha`
- 3. `fr`
- 4. `index.html` : ressource

3.1.1 Algorithme itératif (`str`, `@R`)

```
liste <- decoupage(str)
IP <- @R
pour index allant de 0 à longueur-1 :
    IP <- résoudre(liste[longueur-index], IP)
retrouver IP
```

Code 1 – Algorithme pour la méthode itérative

3.1.2 Algorithme récursif (`str`, `@R`)

```
si l(str) = 0
    retourne IP
sinon
    IP <- str([ln(str)], IP)
    supprime dernier élément(str)
    fonction récursive(str, IP)
```

Code 2 – Algorithme pour la méthode itérative

4 TP 1 - 13 septembre 2022

4.1 Environnement

4.1.1 Exercice 1

Les adresses IP des deux machines sont dans le même réseau.

— serveur : 192.168.242.130

— client : 192.168.242.129

Les deux machines arrivent bien à se ping.

4.2 Configuration du client

4.2.1 Exercice 2

Dans la machine client, j'ai modifié l'adresse présente dans le fichier */etc/resolv.conf* par l'adresse IP du serveur.

4.2.2 Exercice 3

"Multi on" signifie de dire au résolveur, le système qui traduit les noms d'hôtes en IP, de renvoyer plusieurs listes si elles sont trouvées dans le fichier *host*.

4.3 Configuration du serveur

4.3.1 Exercice 4

La commande *echo master > etc/hostname* permet de changer le nom de domaine en *master*

4.3.2 Exercice 5

Pour pouvoir faire en sorte que le serveur ping son propre nom, avec la commande *ping master*, il faut ajouter dans le fichier */etc/hosts*

la ligne suivante : 192.168.242.130¹ *master*

4.3.3 Exercice 6

/etc/bind/conf.d/ est le dossier où se trouve les fichiers de configuration de *bind*.

1. adresse IP du serveur

4.3.4 Exercice 7

Tout fonctionne. J'arrive à ping depuis le client en fonction cette commande *ping master.rt.iutcolmar.uha.fr* Au préalable, j'ai écrit les commandes suivantes dans les fichiers suivants :

```
zone "rt.iutcolmar.uha.fr." {  
    type master;  
    notify yes;  
    file "/etc/bind/db.rt.iutcolmar.uha.fr";  
};
```

Code 3 – Fichier : /etc/bind/named.conf.local

```
;  
; BIND data file for broadcast rt.iutcolmar.uha.fr  
;  
$TTL      604800  
@         IN      SOA      master.rt.iutcolmar.uha.fr. root.rt.iutcolmar.uha.fr. (  
    1                      ; Serial  
    604800                 ; Refresh  
    86400                  ; Retry  
    2419200                ; Expire  
    604800 )               ; Negative Cache TTL  
;  
@         IN      NS       master.rt.iutcolmar.uha.fr.  
master    IN      A        192.168.X.X (adresse IP du serveur)
```

Code 4 – Fichier : /etc/bind/db.rt.iutcolmar.uha.fr

Il faut aussi rajouter dans le fichier */etc/resolv.conf* de la machine client les lignes suivantes :

```
search localdomain  
nameserver 192.168.242.130
```

Code 5 – Fichier : /etc/resolv.conf

4.3.5 Exercice 8

Pour rajouter un alias qui pointe vers `master.rt.iutcolmar.uha.fr`, il faut ajouter dans le fichier `/etc/bind/db.rt.iutcolmar.uha.fr` la ligne suivante :

`dns1 IN CNAME master.rt.iutcolmar.uha.fr.` tout à la fin.

5 TP 2 - 16 septembre 2022

5.1 Environnement

5.1.1 Exercice 1

Les adresses IP des deux machines sont dans le même réseau.

— serveur : 10.211.55.5

— client : 10.211.55.6

5.2 Configuration du client

5.2.1 Exercice 2

J'ai donc comme dans le TP 1 remis dans le dossier */etc/resolv.conf* l'adresse IP du serveur

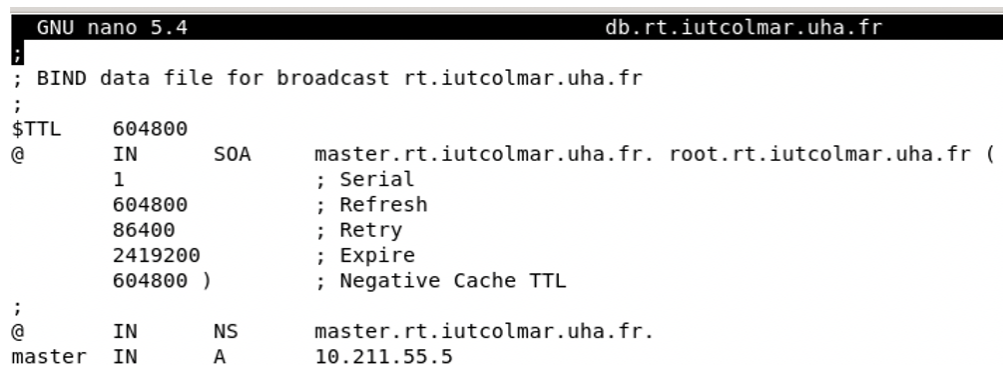
5.3 Mise en place du serveur

5.3.1 Exercice 3

J'ai donc repris exactement les mêmes configuration que le TP 1 pour configurer le serveur dns. Tout fonctionne, j'arrive même à *ping master*

Voici au total tous les fichiers que j'ai modifié avec ce que j'ai écrit. Ne pas oublier de supprimer **network-manager**.

5.3.2 Sur le serveur



```
GNU nano 5.4 db.rt.iutcolmar.uha.fr
;
; BIND data file for broadcast rt.iutcolmar.uha.fr
;
$TTL      604800
@          IN      SOA      master.rt.iutcolmar.uha.fr. root.rt.iutcolmar.uha.fr (
    1              ; Serial
    604800         ; Refresh
    86400          ; Retry
    2419200        ; Expire
    604800 )        ; Negative Cache TTL
;
@          IN      NS       master.rt.iutcolmar.uha.fr.
master     IN      A        10.211.55.5
```

FIGURE 5 – fichier db.rt.iutcolmar.uha.fr

```

GNU nano 5.4                                named.conf.local
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "rt.iutcolmar.uha.fr." {
    type master;
    notify yes;
    file "/etc/bind/db.rt.iutcolmar.uha.fr";
};

```

FIGURE 6 – fichier named.conf.local

```

GNU nano 5.4                                /etc/hostname
master

```

FIGURE 7 – Fichier /etc/hostname

```

GNU nano 5.4                                /etc/hosts
127.0.0.1      master
127.0.1.1      debian-gnu-linux-11.localdomain debian-gnu-linux-11

# The following lines are desirable for IPv6 capable hosts
::1           localhost ip6-localhost ip6-loopback
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters

10.211.55.5    master

```

FIGURE 8 – fichier /etc/hosts

5.3.3 Sur le client

```

GNU nano 5.4                                /etc/resolv.conf
domain localdomain
search localdomain
nameserver 10.211.55.5

```

FIGURE 9 – fichier /etc/resolv.conf

Les pings sont bien fonctionnels entre machines et serveurs :

```

root@debian-gnu-linux-11:/home/parallels# ping master.rt.iutcolmar.uha.fr
PING master.rt.iutcolmar.uha.fr (10.211.55.5) 56(84) bytes of data.
64 bytes from 10.211.55.5 (10.211.55.5): icmp_seq=1 ttl=64 time=0.610 ms
64 bytes from 10.211.55.5 (10.211.55.5): icmp_seq=2 ttl=64 time=0.791 ms
^C
--- master.rt.iutcolmar.uha.fr ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.610/0.700/0.791/0.090 ms

```

FIGURE 10 – Ping du client vers le serveur

5.3.4 Exercice 4

Les variables *SOA*, *NS* et *A* ont déjà été définies comme les captures d'écrans précédentes peuvent le démontrer.

5.3.5 Exercice 5

Pour configurer le service de mails, j'ai décalaré de cette manière les variables :

```

rt.iutcolmar.uha.fr.      IN      MX      10      smtp
smtp      IN      A      10.211.55.5

```

FIGURE 11 – fichier /etc/bind/db.rt.iutcolmar.uha.fr

5.3.6 Exercice 6

Toutes les commandes sont fonctionnels et en plus, voici ce que donne la commande `host -t MX rt.iutcolmar.uha.fr` :

```

root@debian-gnu-linux-11:/home/parallels# host -t MX rt.iutcolmar.uha.fr.
rt.iutcolmar.uha.fr mail is handled by 10 _smtp.rt.iutcolmar.uha.fr.

```

FIGURE 12 – Commande `host -t MX rt.iutcolmar.uha.fr`.

Tout est donc normal et fonctionnel.

5.4 Mise en place du serveur SMTP