

# DNS SMTP piliers de l'Internet

BUT 2 R3.03

Jean-marc Pouchoulon

Septembre 2022

# Demandez le programme !

## 1.3.3. Ressource **R3.03** : Services réseaux avancés

### Compétence ciblée :

- Administrer les réseaux et l'Internet

### SAÉ au sein de laquelle la ressource peut être mobilisée et combinée :

- SAÉ 3.Cyber.03 | Concevoir un réseau informatique sécurisé multi-sites

### Descriptif :

#### Contexte et ancrage professionnel :

Les professionnels R&T doivent être capables d'installer et configurer le serveur DNS d'une zone ainsi que le serveur de courrier correspondant.

#### Contenus :

L'architecture globale du système de nommage internet ainsi que le fonctionnement de la messagerie électronique seront étudiés. Les interactions entre les deux services seront détaillées.

Des clients de tests (nslookup, dig, host, mutt, ...) seront utilisés.

Dans le cadre des travaux pratiques, il pourra s'agir de :

- Installer et configurer un serveur DNS
- Installer un utilitaire client DNS (nslookup, dig, host, ...)
- Installer et configurer un serveur SMTP
- Installer et configurer un serveur IMAP et/ou POP
- Installer et configurer un client IMAP et/ou POP

#### Apprentissages critiques ciblés :

- AC21.03 | Déployer des postes clients et des solutions virtualisées adaptées à une situation donnée
- AC21.04 | Déployer des services réseaux avancés

#### Mots clés :

DNS – Messagerie électronique – Mail – SMTP – POP – IMAP

#### Volume horaire :

Volume horaire défini nationalement : 15 heures dont 11 heures de TP

Le besoin de faire de la résolution de nom est un besoin essentiel. 😊

*Besoin d'accomplissement de soi*

*Besoins d'estime* (confiance et respect de soi, reconnaissance et appréciation des autres)

*Besoins d'appartenance et d'amour*  
(affection des autres)

*Besoins de sécurité*  
(environnement stable et prévisible, sans anxiété ni crise)

*Besoins physiologiques*  
(faim, soif, sexualité, respiration, sommeil, élimination)

**DNS**

# Ça n'a l'air de rien la résolution de nom ...

Mais si le DNS s'arrête ou si le DNS est corrompu c'est Internet qui s'arrête !!!

C'est donc une **cible privilégiée** et sa résilience aux attaques est essentielle.

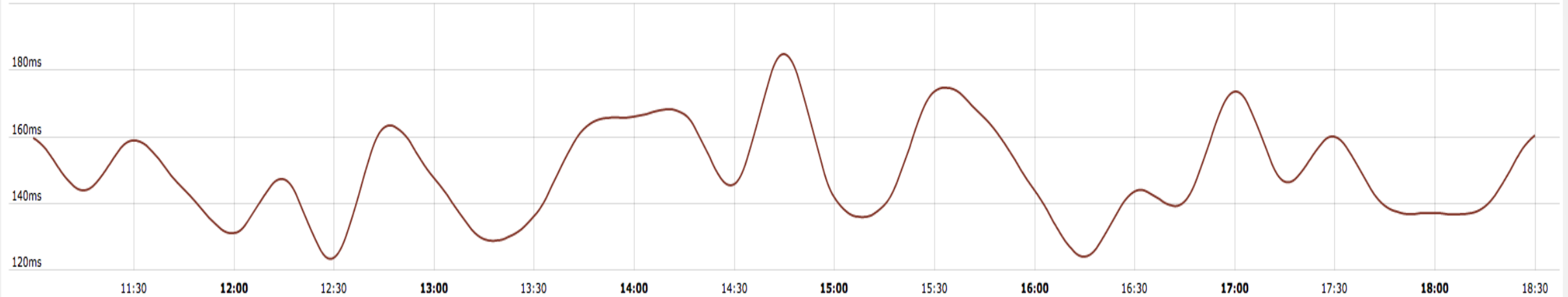
Le **temps de résolution** DNS est essentiel pour la performance des affichages des pages d'un serveur web.

Pour donner une idée des trafics générés, le nombre de résolution servies par un domaine comme ac-montpellier.fr est de l'ordre de 3 Millions de requêtes par jour.

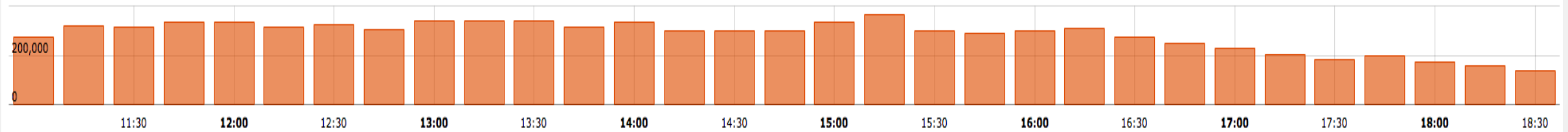
# Activité DNS

Info Début : 2014-09-02 11:00 Niveau d'agrégat : 15 minutes  
Fin : 2014-09-02 18:45

● DNS RT



● Packets



● DNS RT (avg) 149.4 ms

● Packets (sum) 8778144

# Analyser l'activité du DNS est toujours intéressant

	Heure de début	Heure de Fin	Nom demandé	Paquets ▼	Volume	DNS RT
	2014-09-02 11:15:06	2014-09-02 17:33:24	update.services.openoffice.org	695 134	93.1 Mio	7ms
	2014-09-02 11:01:06	2014-09-02 18:26:34	domain1.com	126 355	9.9 Mio	23ms
	2014-09-02 11:00:00	2014-09-02 18:44:59	www.google.com	116 642	15.9 Mio	27ms
	2014-09-02 11:00:00	2014-09-02 18:44:59	bp-eole.ac-dijon.fr	80 334	9.0 Mio	36ms
	2014-09-02 11:00:00	2014-09-02 18:44:59	mur3.ac-montpellier.fr	44 512	5.7 Mio	22ms
	2014-09-02 11:00:06	2014-09-02 17:58:59	2-01-2a40-0015.cdx.cedexis.net	44 161	4.3 Mio	41ms
	2014-09-02 11:00:00	2014-09-02 18:44:50	a151.g.akamai.net	38 731	6.7 Mio	22ms
	2014-09-02 11:00:11	2014-09-02 18:43:16	2-01-2a40-0009.cdx.cedexis.net	27 898	2.8 Mio	56ms
	2014-09-02 11:00:02	2014-09-02 18:44:58	ac-montpellier.fr	26 686	4.2 Mio	22ms
	2014-09-02 11:00:02	2014-09-02 18:44:52	a1961.g.akamai.net	15 428	2.4 Mio	22ms
	2014-09-02 11:00:02	2014-09-02 18:44:54	nossllsearch.google.com	13 937	1.9 Mio	29ms
	2014-09-02 11:00:12	2014-09-02 18:44:52	2-01-2967-0010.cdx.cedexis.net	13 341	2.5 Mio	40ms
	2014-09-02 11:00:02	2014-09-02 18:44:51	a26.ms.akamai.net	13 083	2.1 Mio	18ms
	2014-09-02 11:00:00	2014-09-02 18:44:53	star.c10r.facebook.com	12 868	1.5 Mio	29ms
	2014-09-02 11:00:01	2014-09-02 18:44:43	syndication.twitter.com	12 074	1.9 Mio	37ms
	2014-09-02 11:00:02	2014-09-02 18:44:59	e8218.ce.akamaiedge.net	11 986	1.6 Mio	28ms
	2014-09-02 11:00:00	2014-09-02 18:44:53	renater.ac-montpellier.fr	11 927	1.4 Mio	19ms
	2014-09-02 11:00:04	2014-09-02 18:44:55	www.bluecoat.com	11 837	1.6 Mio	27ms
	2014-09-02 11:00:00	2014-09-02 18:44:40	www.ac-montpellier.fr	11 689	1.5 Mio	21ms
	2014-09-02 11:00:04	2014-09-02 18:44:42	e6033.g.akamaiedge.net	11 615	1.8 Mio	30ms
	2014-09-02 11:00:04	2014-09-02 18:44:02	ocsp.verisign.net	11 590	1.9 Mio	46ms
	2014-09-02 11:00:00	2014-09-02 18:44:57	safebrowsing.cache.l.google.com	11 080	2.4 Mio	21ms
	2014-09-02 11:00:00	2014-09-02 18:44:57	e526.d.akamaiedge.net	10 974	1.5 Mio	18ms
	2014-09-02 11:00:12	2014-09-02 18:43:30	a1284.g.akamai.net	10 796	1.7 Mio	15ms
	2014-09-02 11:00:11	2014-09-02 18:44:32	ib.anycast.adnxs.com	10 563	1.6 Mio	33ms
	2014-09-02 11:00:05	2014-09-02 18:44:57	www.google.fr	10 228	1.3 Mio	23ms
	2014-09-02 11:00:07	2014-09-02 18:44:58	renater12.ac-montpellier.fr	10 149	1.3 Mio	18ms
	2014-09-02 11:00:10	2014-09-02 18:44:58	renater11.ac-montpellier.fr	10 116	1.3 Mio	18ms
	2014-09-02 11:00:08	2014-09-02 18:44:49	clients.l.google.com	9 469	2.0 Mio	18ms
	2014-09-02 11:00:19	2014-09-02 18:44:50	d2uzsrnmmf6tds.cloudfront.net	9 159	2.1 Mio	32ms



# Le côté politique des DNS ...

Contrôler le DNS c'est contrôler le trafic des usagers... et ça peut être très tentant de le faire.

Pour des sociétés commerciales, il est intéressant d'analyser le trafic DNS. Rediriger des flux en fonction des requêtes vers des serveurs amis l'est aussi.

C'est une grande tentation d'agir contre la neutralité du NET.

# C'est quoi un DNS ?

*« Une technologie d'infrastructure : indispensable mais invisible. Une base de données répartie et décentralisée, Qui associe des données à des noms de domaine. »*

*Source : Stéphane Bortmeyer la cantine septembre 2013*



# Le DNS suit un principe hiérarchique

*« Chaque niveau hiérarchique est soit autoritaire pour une ressource, soit délègue cette autorité à un niveau inférieur.*

*Outre la racine de la hiérarchie qui se présente par un point « . », il existe les domaines de plus haut niveau, aussi connus sous l'acronyme TLDs (Top Level Domains). Les plus populaires sont .COM, .ORG, .NET, .FR, etc.*

*On distingue généralement parmi les TLDs ceux attribués en fonction des pays<sup>1</sup> comme .FR, les ccTLDs (country code TLDs), et ceux plus génériques, les gTLDs, comme .COM, .NET ou .ORG.*

*Le domaine arpa est en charge de la résolution des zones inverses*

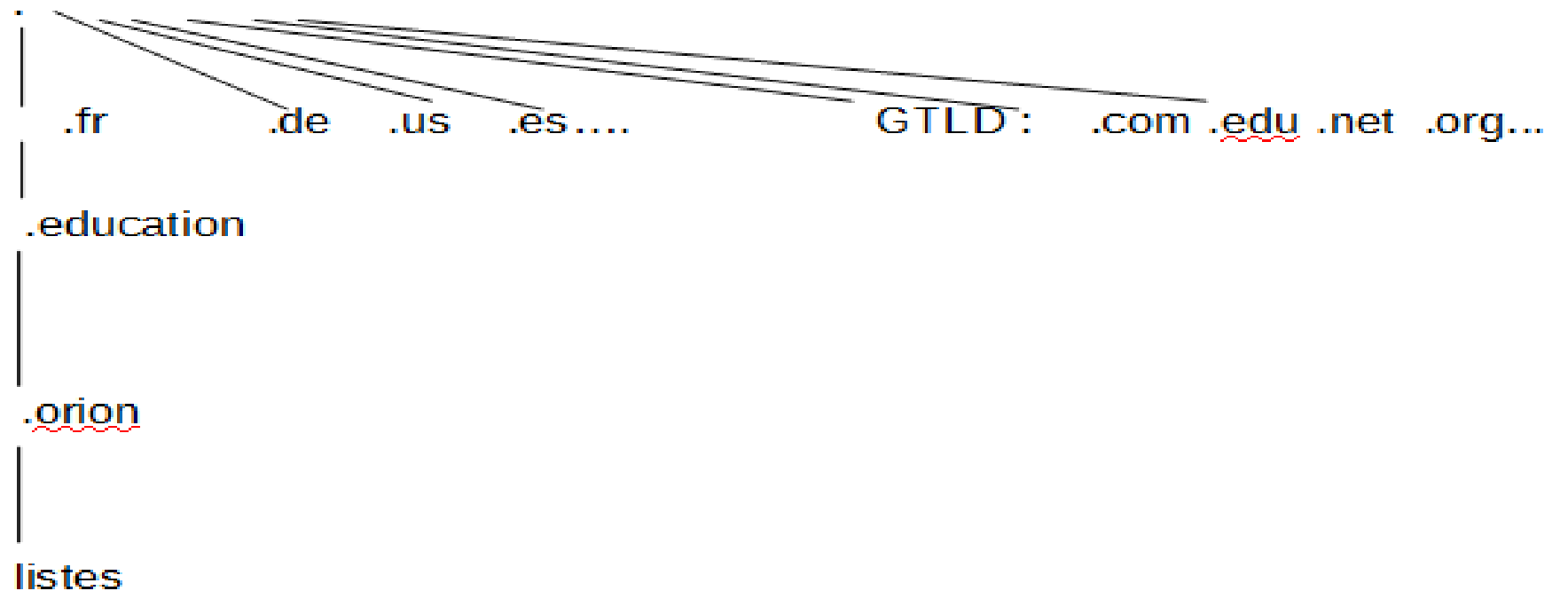
*Les serveurs responsables de la racine de la hiérarchie sont appelés serveurs racines (il en existe 13 à la date de rédaction de ce document). »*

*Source CERTA*

# Les Tops Levels Domains

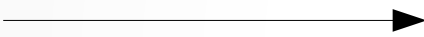
Racine

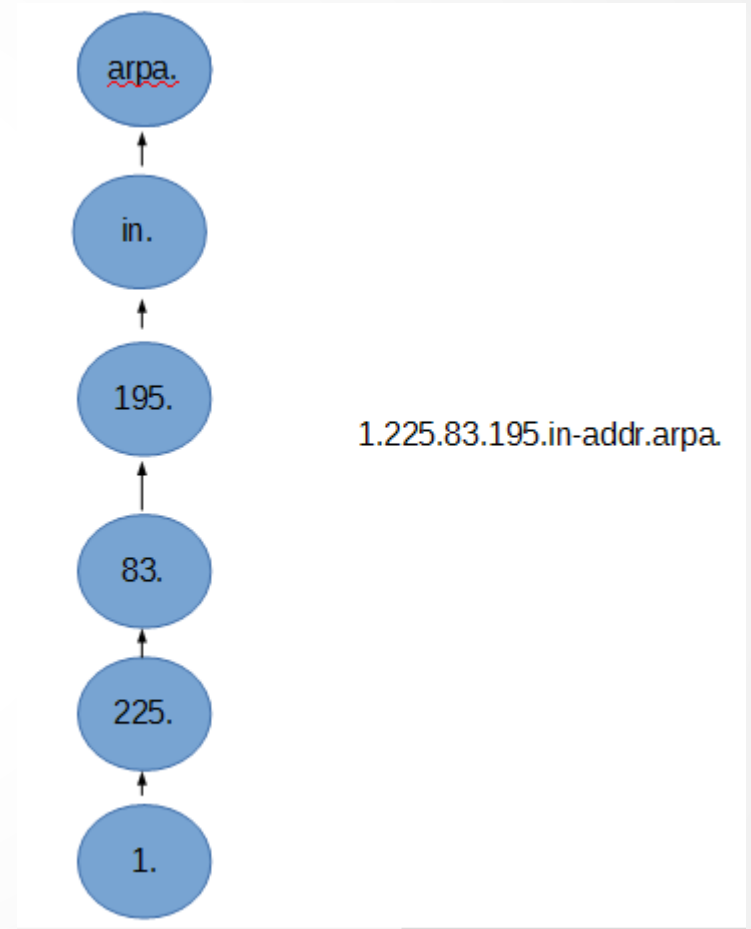
CCTLD :



FQDN=listes.orion.education.fr

# Respect du principe hiérarchique avec l'arborescence inverse in.addr.arpa

- La résolution inverse suit le même principe hiérarchique.
- Une adresse ip 195.83.225.13 sera vu comme 



# Les serveurs Racines

- Les serveurs « racines » sont connus des resolvers Ils sont treize ( en fait bien plus) :

1218	IN	NS	a.root-servers.net.
1218	IN	NS	b.root-servers.net.
1218	IN	NS	c.root-servers.net.
1218	IN	NS	d.root-servers.net.
1218	IN	NS	e.root-servers.net.
1218	IN	NS	f.root-servers.net.
1218	IN	NS	g.root-servers.net.
1218	IN	NS	h.root-servers.net.
1218	IN	NS	i.root-servers.net.
1218	IN	NS	j.root-servers.net.
1218	IN	NS	k.root-servers.net.
1218	IN	NS	l.root-servers.net.
1218	IN	NS	m.root-servers.net.

- Ils connaissent au moins les serveurs de noms et les adresses ip des serveurs des TLD.
- Ils utilisent l'anycast qui offre la possibilité qu' une même adresse IP existe dans plusieurs endroits. C'est le serveur DNS le mieux servi qui va répondre à la requête du client. Le DNS se prête bien à l'Anycast car les requêtes sont généralement en UDP.
- Un serveur DNS choisit son « root server ».

# Les noms de domaines

- Ce sont des noms uniques , faits pour être mémorisés par l'utilisateur.

ex `www.ac-montpellier.fr`

- Ces noms peuvent avoir de 1 à n niveaux de profondeur.

ex `www.orion.education.fr`

- Un nom de domaine peut être public ou privé. Ex : `ac-montpellier.fr` est visible sur Internet. `in.ac-montpellier.fr` ne l'est pas.

# En substance un serveur dns résout un nom en une adresse IP

Un DNS permet entre autre de résoudre une adresse IP en **FQDN** ( **Fully Qualified Domain Name**)

Exemple l'adresse IP v4 de *sync.ac-montpellier.fr* est *195.83.225.141*

Et vice versa ( on parle de **reverse DNS**)

Mais le DNS ne fait pas que ça , il permet de localiser un serveur d'authentification Kerberos , de stocker des clefs publiques cryptographiques, de stocker les adresses IP de spammeurs , de stocker des enregistrements textes...

Une IP peut changer mais un nom de domaine reste lui **stable**. Sans le DNS ce n'est pas possible.

Les noms de domaines dans des alphabets différents du notre sont possibles grâce au support d'Unicode.

# Deux manières de « requêter » un DNS

- **Requête récursive.**

En faisant une requête récursive, un client DNS attend une réponse complète. C'est le serveur qui reçoit la requête qui est en charge de rendre un résultat ( soit parce qu'il a la réponse dans son cache, soit en faisant une requête récursive ou itérative). Un DNS récursif ne doit pas être ouvert sur Internet.

- **Requête itérative.**

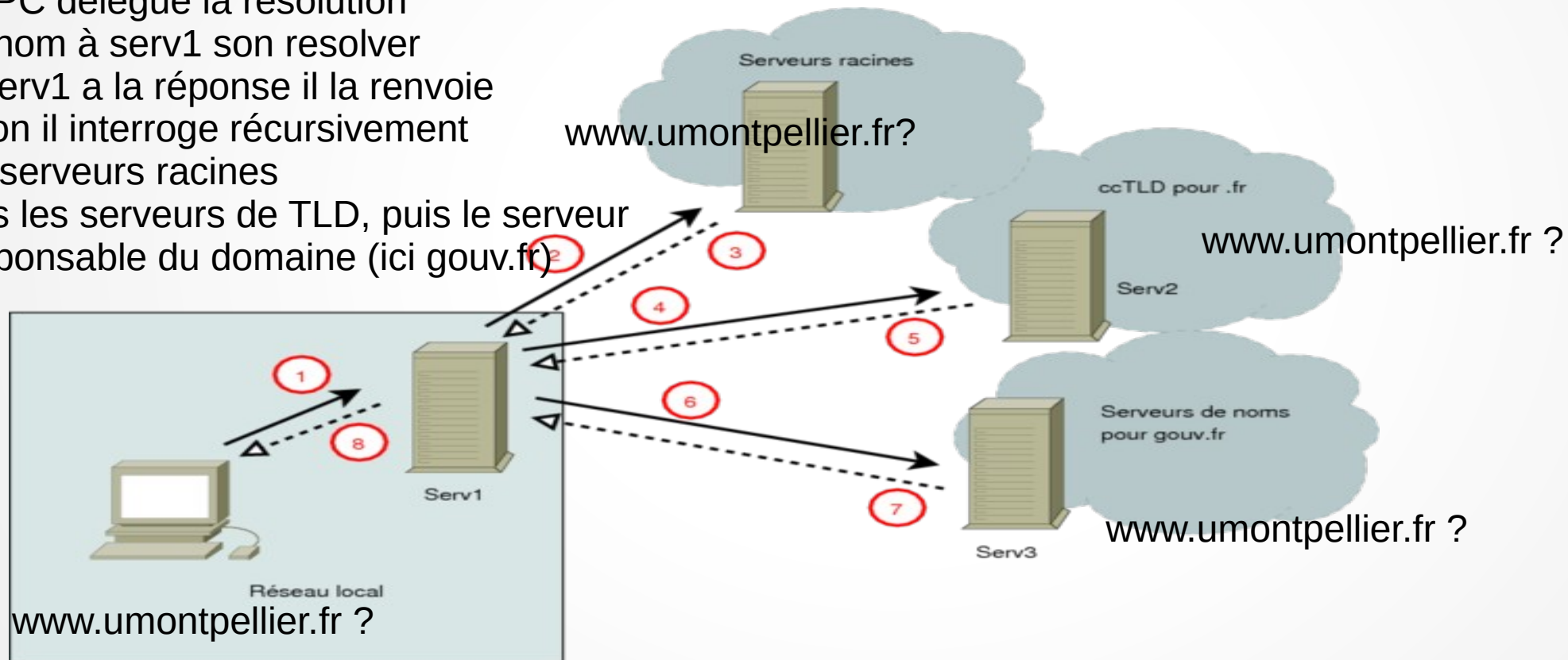
Le client DNS fait une demande mais, a en charge de piloter la résolution. Le client envoie une requête au serveur DNS ; ce dernier renvoie soit la réponse complète (s'il est autoritaire pour la zone concernée) soit une réponse partielle (adresse d'un autre serveur de noms qui va permettre au client d'avancer dans le processus de résolution).

NB Il faut noter que le resolver envoie une demande avec un FQDN complet (ex `www.umontpellier.fr`) au serveur racine et pas la demande `.fr`. Le root server renvoie un referral qui est le NameServer de FR.



# Schéma d'une résolution depuis un poste client

Le PC délègue la résolution de nom à serv1 son resolver  
si serv1 a la réponse il la renvoie  
sinon il interroge récursivement les serveurs racines  
puis les serveurs de TLD, puis le serveur responsable du domaine (ici gouv.fr)



# Resolver/Serveur dns/Cache

Il faut distinguer :

- Le **résolveur** DNS va être en charge d'effectuer une requête DNS. C'est un client, qui va agir comme mandataire pour vous et chercher une réponse à votre demande de résolution de nom.
- Le **serveur DNS** lui va être en charge de répondre à la requête d'un resolver. Il peut être **autoritaire**, c'est à dire qu'il connaît la réponse à votre demande de résolution de nom.

Le serveur ou un resolver dns peuvent servir de **cache** dont la durée est réglée par les DNS autoritaires (\$TTL).

# Les types d'enregistrements DNS

- Un DNS contient des enregistrements, appelés RR (Resource Records), concernant les noms de domaines

Exemple :

Nom de domaine (FQDN)	TTL	Type	RDATA
www.ac-montpellier.fr	86400	A	195.83.225.87

# Quelques types d'enregistrements DNS

- **A** = www.ac-montpellier.fr
- **CNAME** = alias 3w.ac-montpellier.fr
- **NS** = name servers ( IP des serveurs responsables de la zone de nom)
- **PTR** = Pour l'adresse 195.83.225.1, nous obtenons le nom de domaine 1.225.83.195.in-addr.arpa.
- **MX** = Serveurs vers lesquels le courrier doit être envoyé
- **HINFO** = permet de donner des éléments sur le serveur (peu usité)
- **SOA** = Start Of Authority spécifie le serveur de noms qui est la meilleure source d'information pour les données d'une zone, l'adresse électronique d'un contact technique et des paramètres d'expiration du cache.

# D'autres notions essentielles

- Un serveur est dit **authoritative** ou **autoritaire** pour une Zone DNS, s' il dispose de l'ensemble des informations de la Zone.
- Un serveur est dit **primaire** ou **maître** pour une Zone, s' il gère les informations de la zone. Il n'y a qu'un primaire par Zone, il est responsable de la validité des informations de la zone.
- Un serveur est dit **secondaire** ou **esclave** pour une Zone, s' il possède une copie des informations du domaine DNS. La copie provient en général du primaire mais peut aussi provenir d'un autre secondaire. Un secondaire est aussi autoritaire sur une Zone.

# Transferts de ZONES

- A chaque mise à jour d'un serveur primaire un **numéro de série** est incrémenté par la personne qui a modifié le fichier et une notification est envoyée vers le DNS esclave qui vient chercher sa mise à jour ( attention aux firewalls )
- Le transfert peut être **complet** ( AXFR ) ou **différentiel** ( IXFR )

# GLUE RECORD

- Un « **glue record** » est juste l'adresse IP d'un serveur de nom. Ils sont essentiels quand le serveur de noms appartient au domaine pour lequel il fait office de DNS.
- Exemple : un des serveurs de noms de ac-montpellier.fr est **renater.ac-montpellier.fr**. Si on pose la question à un DNS "trouve `www.ac-montpellier.fr` " le serveur de l'AFNIC va renvoyer `renater.ac-montpellier.fr` comme serveur de nom de la Zone. Le resolver va alors demander l'adresse IP de **renater.ac-montpellier.fr** qui va lui renvoyer aussi `renater.ac-montpellier.fr` puisqu'il ne le connaît pas
- Le seul moyen de **casser cette boucle** est que le serveur de l'AFNIC connaisse l'adresse IP de `renater.ac-montpellier.fr`. Il pourra ainsi répondre au resolver d'aller chercher la résolution de `ac-montpellier.fr` sur cette IP. On **duplique** donc l'info NS sur le CcTLD et sur les serveurs de la zone `ac-montpellier.fr`.



# Couches réseaux DNS

- Les DNS échangent des messages en **UDP** en règle générale. EDNS0 est une extension qui permet de franchir sans soucis la barrière des 512 Octets comme taille maximale des paquets.
- Néanmoins le DNS peut échanger en **TCP**. C'est en particulier vrai lors de l'échange de zones entre DNS maître et esclave.
- Les clients comme Firefox ou des resolver comme systemd peuvent utiliser aussi https avec **DoH** (DNS over https). Ce qui permet un chiffrement du DNS de bout en bout et de maintenir privée vos requêtes. On peut aussi utiliser DoT (DNS over TLS) sur le port 853

# Les risques de sécurité liés au DNS

- Le port 53 est un port parfois accessible sur Internet. Il est aisé de se servir de ce port pour faire passer un **tunnel**.
- Le cache d'un DNS peut être **empoisonné**. Une fois une mauvaise valeur mise en cache ce sont toutes les réponses liées à cet enregistrement qui sont corrompues.
- **Déni de service** : Le DNS utilise UDP et peut servir pour des attaques par amplification. Il convient donc de ne répondre aux requêtes provenant de l'Internet que sur le domaine dont on est responsable sous peine de participer à un ddos.
- La vie privée dépend fortement des requêtes DNS qui sont très révélatrices de votre activité. Les DNS gratuits de certains GAFAM ne sont pas innocents... DoH est une réponse mais en entreprise mais ce protocole laisse une liberté qui peut être problématique.
- La **haute disponibilité** des DNS dépend de l'architecture sous-jacente : Il faut donc **deux DNS à minima** par zone autoritaire et en règle générale sur des lieux physiques différents (2 A.S. B.G.P. différentes en fait).

# Sécuriser le DNS

- Assurer la meilleure **redondance** possible (HA).
- Veiller à utiliser des **versions à jour** des logiciels DNS(Patch Management).
- Assurer une **surveillance** régulière de ses serveurs et de leurs configurations (Supervision).
- Utiliser **DNSSEC**.
- Définir un « Plan de continuité d'activité »

Source : **AFNIC**

# Les principaux DNS opensource

- BIND
- UNBOUND
- DNSMAQ...

# Menaces sur le DNS

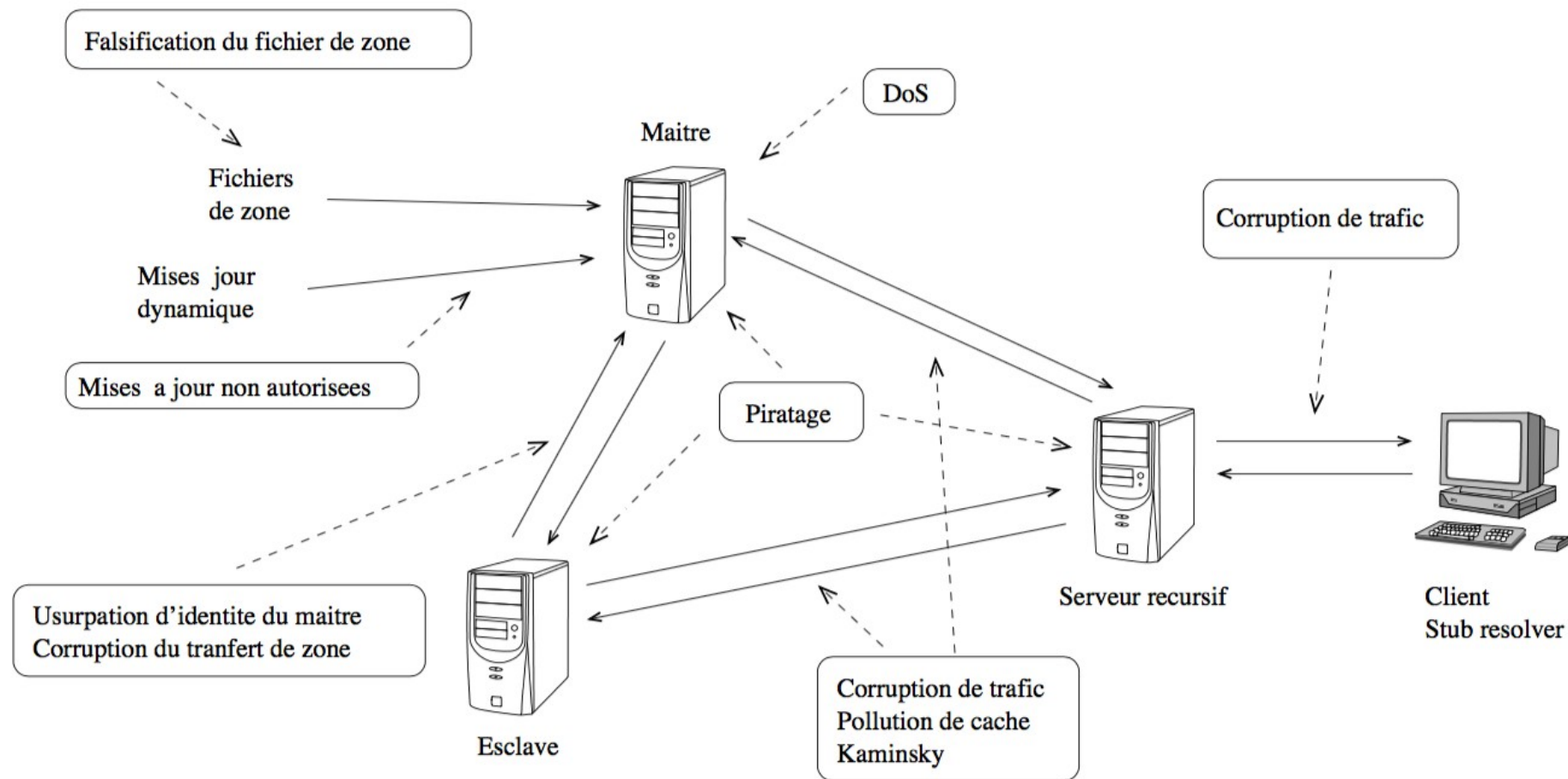


Figure 2. Les différents types d'attaques sur le DNS. Source : Stéphane Bortzmeyer de l'AFNIC.

# Cryptographie Symétrique pour le DNS

- TSIG est un mécanisme qui permet de sécuriser les transactions entre serveurs DNS par l'utilisation de signatures électroniques. On va l'utiliser lors de la transmission des zones d'un maître à un esclave par exemple.
- TSIG utilise une technique de cryptographie à clé symétrique, c'est-à-dire que les parties impliquées dans la communication utilisent un secret partagé qui est valable aussi bien pour le codage que pour le décodage.
- Avec TSIG, tous les messages DNS sont signés ; ceci concerne aussi bien les requêtes que les réponses à ces dernières.
- Il permet de vérifier qui envoie et reçoit les messages mais pas l'intégrité des contenus. Chaque signature n'est utilisée qu'une fois et la synchronisation du temps via NTP est requise.

# DNSSEC

- L'introduction de la cryptographie à clefs publiques dans le DNS permet de sécuriser les informations contenues dans le DNS. Il protège donc les données pas le canal de transmission.
- Un RRSet est un groupe de ressources ayant les mêmes valeurs pour les champs NAME, TYPE et CLASS mais avec un RDATA différent.
- Un serveur autoritaire DNS va signer un hash d'un élément contenu dans le DNS ( ex Ressource Record) à l'aide de sa clef privée. Le RRset et le hash sont envoyés au Resolver qui vérifiera à l'aide de la clef publique que le hash envoyé est le bon.



# Enregistrement DNSKEY et RRSIG

- L'enregistrement DNSKEY est utilisé pour transmettre une clé publique entre le résolveur et le serveur de nom.
- L'enregistrement RRSIG (Ressource Record Signature )contient la signature de l'enregistrement envoyé par le serveur d'autorité. Il s'agit de la signature obtenue en signant le hash du RRset avec la clé privée du serveur d'autorité.
- NSEC (Next SECure), NSEC3 et NSEC3PARAM sont des ressources qui permettent d'être sûr qu'un enregistrement n'existe pas.

# Chaine de confiance Trust Anchor

- DS (Delegation Signer) est une ressource qui permet d'authentifier les clés DNSKEY, elle contient dans la partie RDATA le hash de la clef publique de la zone. Elle sera transmise à la zone parent et signée par sa clef privée établissant ainsi le lien de confiance entre la zone mère et la zone fille. Chaque zone enfant fera de même avec sa zone parent.
- On se sert donc de la zone parent afin de valider la zone enfant. Mais on ne peut valider la racine du DNS car elle n'a pas de parent. On récupère donc la clef publique de la KSK de racine. C'est ce qu'on appelle une « **trust anchor** » qu'il faut récupérer manuellement quand on veut travailler avec DNSSEC.