WIKIPÉDIA

Domain Name System

Le **Domain Name System**, généralement abrégé **DNS**, qu'on peut traduire en « système de noms de domaine », est le service informatique distribué utilisé pour traduire les noms de domaine Internet en adresse IP ou autres enregistrements. En fournissant dès les premières années d'Internet, autour de 1985, un service distribué de résolution de noms, le DNS a été un composant essentiel du développement du réseau.

À la demande de la DARPA (Defense Advanced Research Projects Agency, « Agence pour les projets de recherche avancée de défense ») américaine, Jon Postel et Paul Mockapetris ont conçu le « *Domain Name System* » en 1983 et en ont rédigé la première implémentation.

Domain Name System

Informations

Fonction	Traduction de nom de domaine en adresse IP
Sigle	DNS
Port	53
RFC	1983: RFC 882[1] - RFC 883[2] 1987: RFC 1034[3] - RFC 1035[4] 1994: RFC 1591[5] 2011: RFC 6195[6] 2013: RFC 6895[7] 2018: RFC 8375[8] - RFC 8467[9] - RFC 8483[10] - RFC 8484[11] 2019: RFC 8499[12]

Sommaire

Rôle du DNS

Les équipements (*hôtes*) connectés à un réseau IP, comme Internet, possèdent une adresse IP qui les identifie sur le réseau. Ces adresses sont numériques afin de faciliter leur traitement par les machines. En IPv4, elles sont représentées sous la forme « xxx.xxx.xxx.xxx », où « xxx » est un nombre entre 0 et 255 (en représentation décimale). En IPv6, les adresses sont

représentées sous la forme « xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx », où « xxxx » représente une valeur hexadécimale de 0000 à FFFF.

Pour faciliter l'accès aux hôtes sur un réseau IP, un mécanisme a été mis en place pour associer un nom à une adresse IP. Ce nom, plus simple à retenir, est appelé « nom de domaine ». *Résoudre un nom de domaine* consiste à trouver l'adresse IP qui lui est associée.

En plus des adresses IP, des informations complémentaires peuvent être associées aux noms de domaines comme des enregistrements dans le contexte de la lutte contre le spam (SPF), RRSIG pour la sécurité des informations du DNS (DNSSEC) ou NAPTR pour associer des numéros de téléphone à des adresses e-mail (ENUM).

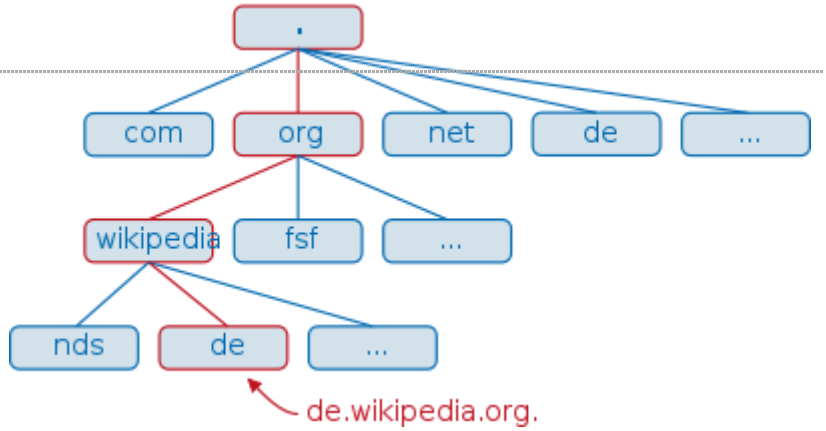
Histoire

Avant le DNS, la résolution d'un nom sur Internet devait se faire grâce à un fichier texte appelé *HOSTS.TXT* (RFC 608[13]) maintenu par le NIC du Stanford Research Institute (SRI) et copié sur chaque ordinateur du réseau par transfert de fichier. En 1982, ce système centralisé montre ses limites et plusieurs propositions de remplacement voient le jour, parmi lesquelles le système distribué *Grapevine* de Xerox et IEN 116[14]. Le premier (*Grapevine*) est jugé trop compliqué tandis que le second (IEN 116) est insuffisant[15]. C'est finalement l'équipe dirigée par Elizabeth Feinler au NIC qui définira le Domain Name System afin de gérer la croissance de l'internet en déléguant la gestion des noms de domaine à des serveurs de noms distribués. Paul Mockapetris publie le design du système dans les RFC 882[1] et RFC 883[2] en 1983. La norme correspondante est publiée dans les RFC 1034[3] et RFC 1035[4] en 1987. En 1987, le fichier HOSTS.TXT contenait 5 500 entrées, tandis que 20 000 hôtes étaient définis dans le DNS.

Un système hiérarchique et distribué

Hiérarchie du DNS

Le système des noms de domaine consiste en une hiérarchie dont le sommet est appelé la *racine*. On représente cette dernière par un point. Dans un domaine, on peut créer un ou plusieurs sous-domaines ainsi qu'une *délégation* pour ceux-ci, c'est-à-dire une indication que les informations relatives à ce sous-domaine sont enregistrées sur un autre serveur. Ces sous-domaines peuvent à leur tour déléguer des sous-domaines vers d'autres serveurs.



Hiérarchie du DNS.

Tous les sous-domaines ne sont pas nécessairement délégués. Les délégations créent des *zones*, c'est-à-dire des ensembles de domaines et leurs sous-domaines non délégués qui sont configurés sur un serveur déterminé. Les zones sont souvent confondues avec les domaines.

Les domaines se trouvant immédiatement sous la racine sont appelés domaine de premier niveau (TLD : Top Level Domain). Les noms de domaines ne correspondant pas à une extension de pays sont appelés des domaines génériques (gTLD), par exemple .org ou .com. S'ils correspondent à des codes de pays (fr, be, ch…), ce sont des domaines de premier niveau national, aussi appelés ccTLD de l'anglais *country code TLD*.

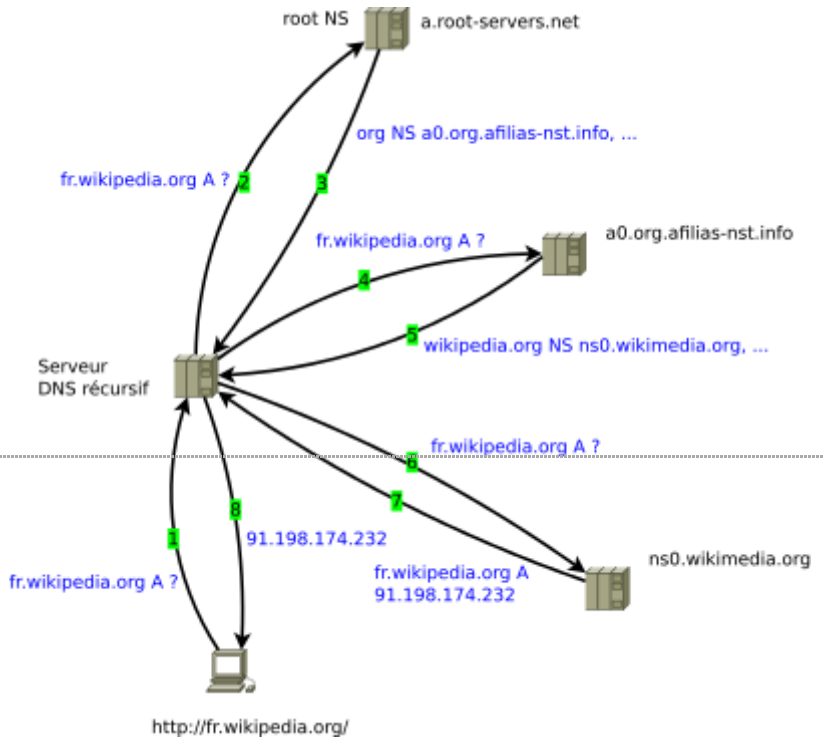
On représente un nom de domaine en indiquant les domaines successifs séparés par un point, les noms de domaines supérieurs se trouvant à droite. Par exemple, le domaine *org.* est un TLD, sous-domaine de la racine. Le domaine *wikipedia.org.* est un sous-domaine de *.org*. Cette délégation est accomplie en indiquant la liste des serveurs DNS associée au sous-domaine dans le domaine de niveau supérieur.

Les noms de domaines sont donc résolus en parcourant la hiérarchie depuis le sommet et en suivant les délégations successives, c'est-à-dire en parcourant le nom de domaine de droite à gauche.

Pour qu'il fonctionne normalement, un nom de domaine doit avoir fait l'objet d'une délégation correcte dans le domaine de niveau supérieur.

Résolution du nom par un hôte

Les hôtes n'ont qu'une connaissance limitée du système des noms de domaine. Quand ils doivent résoudre un nom, ils s'adressent à un ou plusieurs serveurs de noms dits *récursifs*, c'est-à-dire qu'ils vont parcourir la hiérarchie DNS et faire suivre la requête à un ou plusieurs autres serveurs de noms pour fournir une réponse. Les adresses IP de ces serveurs récursifs sont souvent obtenues via DHCP ou encore configurés *en dur* sur la machine hôte. Les fournisseurs d'accès à Internet mettent à disposition de leurs clients ces serveurs récursifs. Il existe également des serveurs récursifs publics comme ceux de Yandex.DNS, Google Public DNS ou OpenNIC.



Résolution itérative d'un nom dans le DNS par un serveur DNS (étapes 2 à 7) et réponse (étape 8) suite à l'interrogation récursive (étape 1) effectuée par un client (resolver) DNS. (remarque: Le serveur DNS récursif est dit récursif car il accepte ce type de requêtes mais il effectue des requêtes itératives)

Quand un serveur DNS récursif doit trouver l'adresse IP de *fr.wikipedia.org*, un processus itératif démarre pour consulter la hiérarchie DNS. Ce serveur demande aux serveurs DNS appelés *serveurs racine* quels serveurs peuvent lui répondre pour la zone *org*. Parmi ceux-ci, le serveur va en choisir un pour savoir quels serveurs sont capables de lui répondre pour la zone *wikipedia.org*. C'est un de ces derniers qui pourra lui donner l'adresse IP de *fr.wikipedia.org*. S'il se trouve qu'un serveur ne répond pas, un autre serveur de la liste sera consulté.

Pour optimiser les requêtes ultérieures, les serveurs DNS récursifs font aussi office de *DNS cache* : ils gardent en mémoire (*cache*) la réponse d'une résolution de nom afin de ne pas effectuer ce processus à nouveau ultérieurement. Cette information est conservée pendant une période nommée *Time to live* et associée à chaque nom de domaine.

Un nom de domaine peut utiliser plusieurs serveurs DNS. Généralement, les noms de domaines en utilisent au moins deux : un primaire et un secondaire. Il peut y avoir plusieurs serveurs secondaires.

L'ensemble des serveurs primaires et secondaires font autorité pour un domaine, c'est-à-dire que la réponse ne fait pas appel à un autre serveur ou à un cache. Les serveurs récursifs fournissent des réponses qui ne sont pas nécessairement à jour, à cause du cache mis en place. On parle alors de réponse ne faisant pas autorité (*non-authoritative answer*).

Cette architecture garantit au réseau Internet une certaine continuité dans la résolution des noms. Quand un serveur DNS tombe en panne, le bon fonctionnement de la résolution de nom n'est pas remis en cause dans la mesure où des serveurs secondaires sont disponibles.

Résolution inverse

Pour trouver le nom de domaine associé à une adresse IP, on utilise un principe semblable. Dans un nom de domaine, la partie la plus générale est à droite : org dans fr.wikipedia.org, le mécanisme de résolution parcourt donc le nom de domaine de droite à gauche. Dans une adresse IP V4, c'est le contraire : 213 est la partie la plus générale de 213.228.0.42. Pour conserver une logique cohérente, on inverse l'ordre des quatre termes de l'adresse et on la concatène au pseudo domaine *in-addr.arpa*. Ainsi, par exemple, pour trouver le nom de domaine de l'adresse IP 91.198.174.2, on résout 2.174.198.91.in-addr.arpa.

La déclaration inverse est importante sur les adresses IP publiques Internet puisque l'absence d'une résolution inverse est considérée comme une erreur opérationnelle (RFC 1912[16]) qui peut entraîner le refus d'accès à un service. Par exemple, un serveur de messagerie électronique se présentant en envoi avec une adresse IP n'ayant pas de résolution inverse (PTR) a de grandes chances de se voir refuser, par l'hôte distant, la transmission du courrier (message de refus de type : *IP lookup failed*).

De plus, cette résolution inverse est importante dans le cadre de la réalisation de diagnostics réseaux car c'est elle qui permet de rendre les résultats de la commande traceroute humainement exploitables. Les dénominations des noms d'hôtes inverses sont souvent des composites de sous-domaines de localisation (ville, région, pays) et de domaines explicites indiquant le fournisseur d'accès Internet traversé comme francetelecom.net (XXXX.nctou202.Toulouse.francetelecom.net) et opentransit.net (XXXX.Aubervilliers.opentransit.net) pour France Télécom, ou encore proxad.net (XXXX.intf.routers.proxad.net) pour Free.

Une adresse IP peut être associée à différents noms de domaine via l'enregistrement de plusieurs entrées PTR dans le sous-domaine *.arpa* consacré à cette adresse (in-addr.arpa. pour IPv4 et ip6.arpa. pour IPv6). L'utilisation d'enregistrements PTR multiples pour une même adresse IP est éventuellement présente dans le cadre de l'hébergement virtuel de multiples domaines web derrière la même adresse IP mais n'est pas recommandée dans la mesure où le nombre des champs PTR à renvoyer peut faire dépasser à la réponse la taille des paquets UDP de réponse et entraîner l'utilisation du protocole TCP (plus coûteux en ressources) pour envoyer la réponse à la requête DNS[17].

Résolution inverse CIDR

Les délégations des zones inverses se font sur une frontière d'octet, ce qui fonctionne quand les blocs d'adresses sont distribués de façon classful mais pose des problèmes quand les blocs assignés sont de taille quelconque.

Par exemple, si deux clients A et B disposent chacun des blocs 192.168.0.0/25 et 192.168.0.128/25, il n'est pas possible de déléguer 0.168.192.in-addr.arpa. au premier pour qu'il puisse définir les PTR correspondant à ses hôtes, car cela empêcherait le second de faire de même.

La RFC 2317[18] a défini une approche pour traiter ce problème, elle consiste à faire usage de domaines intermédiaires et de CNAME.

```
$ORIGIN 0.168.192.in-addr.arpa.
0/25 NS ns.clientA.fr.
128/25 NS ns.clientB.fr.

0 CNAME 0.0/25.0.168.192.in-addr.arpa.
1 CNAME 1.0/25.0.168.192.in-addr.arpa.
...
127 CNAME 127.0/25.0.168.192.in-addr.arpa.
128 CNAME 128.128/25.0.168.192.in-addr.arpa.
...
255 CNAME 255.128/25.0.168.192.in-addr.arpa.
```

Le client A définit la zone 0/25.0.168.192.in-addr.arpa. :

```
$ORIGIN 0/25.0.168.192.in-addr.arpa.
1 PTR hote1.clientA.fr.
...
127 PTR hote127.clientA.fr.
```

Le client B fait de même pour 128/25.0.168.192.in-addr.arpa. et les adresses 128 à 255.

La résolution inverse de 192.168.0.1 aboutira aux requêtes suivantes :

```
1.0.168.192.in-addr.arpa. CNAME 1.0/25.0.168.192.in-addr.arpa.
1.0/25.0.168.192.in-addr.arpa. PTR hote1.clientA.fr.
```

Ce qui assure le fonctionnement de la résolution inverse, moyennant un niveau d'indirection supplémentaire.

Serveurs DNS racine

Les serveurs racine sont gérés par douze organisations différentes : deux sont européennes, une japonaise et les neuf autres sont américaines. Sept de ces serveurs sont en réalité distribués dans le monde grâce à la technique anycast et neuf disposent d'une adresse IPv6[19]. Grâce à anycast, plus de 200 serveurs répartis dans 50 pays du monde assurent ce service[20]. Il existe 13 autorités de nom appelées de a à m.root-servers.net. Le serveur *k* reçoit par exemple de l'ordre de 70 000 à 100 000 requêtes par seconde en avril 2019[21].

Le DNS ne fournit pas de mécanisme pour découvrir la liste des serveurs racine, chacun des serveurs doit donc connaître cette liste au démarrage grâce à un encodage explicite. Cette liste est ensuite mise à jour en consultant l'un des serveurs indiqués. La mise à jour de cette liste est peu fréquente de façon que les serveurs anciens continuent à fonctionner.

Fully Qualified Domain Name

On entend par Fully qualified domain name (FQDN), ou *Nom de domaine pleinement qualifié* un nom de domaine écrit de façon absolue, y compris tous les domaines jusqu'au domaine de premier niveau (TLD), il est ponctué par un point final, par exemple fr.wikipedia.org. .

La norme prévoit qu'un élément d'un nom de domaine (appelé *label*) ne peut dépasser 63 caractères, un FQDN ne pouvant dépasser 253 caractères.

Nom de domaine internationalisé

Dans leur définition initiale, les noms de domaines sont constitués des caractères de A à Z (sans casse : les lettres capitales ne sont pas différenciées), de chiffres et du trait d'union.

La RFC 3490[22] définit un format appelé Punycode qui permet l'encodage d'un jeu de caractère plus étendu.

Les techniques du DNS *Round-Robin* pour la distribution de la charge

Lorsqu'un service génère un trafic important, celui-ci peut faire appel à la technique du *DNS Round-Robin* (en français tourniquet DNS), une des techniques de répartition de charge qui consiste à associer plusieurs adresses IP à un FQDN. Les différentes versions de Wikipedia, comme *fr.wikipedia.org* par exemple, sont associées à plusieurs adresses IP : 207.142.131.235, 207.142.131.236, 207.142.131.245, 207.142.131.246, 207.142.131.247 et 207.142.131.248. L'ordre dans lequel ces adresses sont renvoyées sera modifié d'une requête à la suivante. Une rotation circulaire entre ces différentes adresses permet ainsi de répartir la charge générée par ce trafic important entre les différentes machines ayant ces adresses IP. Il faut cependant nuancer cette répartition car elle n'a lieu qu'à la résolution du nom d'hôte et reste par la suite en cache sur les différents *resolvers* (client DNS).

Principaux enregistrements DNS

Le type d'enregistrement de ressource (RR pour Resource Record) est codé sur 16 bits[23], l'IANA conserve le registre des codes assignés[24]. Les principaux enregistrements définis sont les suivants :

- **A record** ou **address record** (également appelé *enregistrement d'hôte*) qui fait correspondre un nom d'hôte ou un nom de domaine ou un sous-domaine 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** qui définit les serveurs de courriel pour ce domaine ;
- **PTR record** ou **pointer record** qui associe une adresse IP à un enregistrement de nom de domaine, aussi dit « *reverse* » puisqu'il fait exactement le contraire du A record ;
- **NS record** ou **name server record** qui définit les serveurs DNS de ce domaine ;
- **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, standardisé dans la RFC 2782[25] ;
- **NAPTR record** ou **Name Authority Pointer record** qui donne accès à des règles de réécriture de l'information, permettant des correspondances assez lâches entre un nom de domaine et une ressource. Il est spécifié dans la RFC 3403[26] ;
- **TXT record** permet à un administrateur d'insérer un texte quelconque dans un enregistrement DNS (par exemple, cet enregistrement est utilisé pour implémenter la spécification *Sender Policy Framework*) ;
- d'autres types d'enregistrements sont utilisés occasionnellement, ils servent simplement à donner des informations (par exemple, un enregistrement de type **LOC** indique l'emplacement physique d'un hôte, c'est-à-dire sa latitude et sa longitude). Certaines personnes disent que cela aurait un intérêt majeur mais n'est que très rarement utilisé sur le monde Internet.

NS record

L'enregistrement NS crée une délégation d'un sous-domaine vers une liste de serveurs.

Dans la zone *org*, les enregistrements NS suivants créent le sous-domaine *wikipedia* et délèguent celui-ci vers les serveurs indiqués.

L'ordre des serveurs est quelconque. Tous les serveurs indiqués doivent faire autorité pour le domaine.

```
wikipedia NS ns1.wikimedia.org.
wikipedia NS ns2.wikimedia.org.
wikipedia NS ns0.wikimedia.org.
```

PTR record

À l'inverse d'une entrée de type A ou AAAA, une entrée PTR indique à quel nom d'hôte correspond une adresse IPv4 ou IPv6. Si elle est spécifiée, elle doit contenir l'enregistrement inverse d'une entrée DNS A ou AAAA.

Par exemple (pour une adresse IPv4) cet enregistrement PTR est :

```
232.174.198.91.in-addr.arpa. IN PTR text.esams.wikimedia.org.
```

correspond à cette entrée A :

```
text.esams.wikimedia.org. IN A 91.198.174.232
```

Dans le cas d'une adresse IPv6, les entrées de type PTR sont enregistrées dans la zone ip6.arpa. (pendant de la zone in-addr.arpa. des adresses IPv4).

La règle permettant de retrouver l'entrée correspondant à une adresse IPv6 est similaire à celle pour les adresses IPv4 (renversement de l'adresse et recherche dans un sous-domaine dédié de la zone arpa.), mais diffère au niveau du nombre de bits de l'adresse utilisés pour rédiger le nom du domaine où rechercher le champ PTR : là où pour IPv4 le découpage de l'adresse se fait par octet, pour IPv6 c'est un découpage par quartet qui est utilisé.

Par exemple à l'adresse IPv6 :

```
2001:610:240:22::c100:68b
```

correspond le nom de domaine :

```
b.8.6.0.0.0.1.c.0.0.0.0.0.0.0.0.2.2.0.0.0.4.2.0.0.1.6.0.1.0.0.2.ip6.arpa. PTR   www.ipv6.ripe.net.
```

MX record

Une entrée DNS MX indique les serveurs SMTP à contacter pour envoyer un courriel à un utilisateur d'un domaine donné. Par exemple :

```
wikimedia.org. IN MX 10 mchenry.wikimedia.org.
wikimedia.org. IN MX 50 lists.wikimedia.org.
```

On voit que les courriels envoyés à une adresse en @wikimedia.org sont envoyés au serveur mchenry.wikimedia.org. ou lists.wikimedia.org. Le nombre précédant le serveur représente la priorité. Le serveur avec la priorité numérique la plus petite est employé en priorité. Ici, c'est donc mchenry.wikimedia.org. qui doit être utilisé en premier, avec une valeur de 10.

Les serveurs indiqués doivent avoir été configurés pour accepter de relayer les courriers pour le nom de domaine indiqué. Une erreur courante consiste à indiquer des serveurs quelconques comme serveurs secondaires, ce qui aboutit au rejet des courriers quand le serveur primaire devient inaccessible. Il n'est pas indispensable de disposer de serveurs secondaires, les serveurs émetteurs conservant les messages pendant un temps déterminé (typiquement, plusieurs jours) jusqu'à ce que le serveur primaire soit à nouveau disponible.

Les entrées MX sont généralisées par les entrées SRV qui permettent de faire la même chose mais pour tous les services, pas seulement SMTP (le courriel). L'avantage des entrées SRV par rapport aux entrées MX est aussi qu'elles permettent de choisir un port arbitraire pour chaque service ainsi que de faire de la répartition de charge plus efficacement. L'inconvénient c'est qu'il existe encore peu de programmes clients qui gèrent les entrées SRV. Cependant, depuis 2009, avec l'augmentation de l'utilisation du protocole SIP sur les services de VoIP, les enregistrements SRV deviennent plus fréquents dans les zones DNS.

CNAME record

L'enregistrement CNAME permet de créer un *alias*.

Par exemple :

```
fr.wikipedia.org. IN CNAME text.wikimedia.org.
text.wikimedia.org. IN CNAME text.esams.wikimedia.org.
text.esams.wikimedia.org. IN A 91.198.174.232
```

Celui-ci exclut tout autre enregistrement (RFC 1034[3] section 3.6.2, RFC 1912[16] section 2.4), c'est-à-dire qu'on ne peut avoir à la fois un CNAME et un A record pour le même nom de domaine.

Par exemple, ceci est interdit :

```
fr.wikipedia.org. IN CNAME text.wikimedia.org.
fr.wikipedia.org. IN A 91.198.174.232
```

Par ailleurs, pour des raisons de performance, et pour éviter les boucles infinies du type

```
fr.wikipedia.org. IN CNAME text.wikimedia.org.
text.wikipedia.org. IN CNAME fr.wikimedia.org.
```

les spécifications (RFC 1034[3] section 3.6.2, RFC 1912[16] section 2.4) recommandent de ne pas faire pointer un CNAME sur un autre CNAME ni sur un DNAME (alias pour un nom et tous ses sous-noms).

Ainsi, le premier exemple serait préférablement enregistré de la façon suivante :

```
fr.wikipedia.org. IN CNAME text.esams.wikimedia.org.
text.wikimedia.org. IN CNAME text.esams.wikimedia.org.
text.esams.wikimedia.org. IN A 91.198.174.232
```

NAPTR record

Peu répandus à l'heure actuelle (ils sont surtout utilisés par ENUM), ils décrivent une réécriture d'une *clé* (un nom de domaine) en URI. Par exemple, dans ENUM, des enregistrements NAPTR peuvent être utilisés pour trouver l'adresse de courrier électronique d'une personne, connaissant son numéro de téléphone (qui sert de clé à ENUM).

Ses paramètres sont dans l'ordre :

1. **Order** : indique dans quel ordre évaluer les enregistrements NAPTR ; tant qu'il reste des enregistrements d'une certaine valeur de **order** à examiner, les enregistrements des valeurs suivantes de **order** n'entrent pas en considération ;
2. **Preference** : donne une indication de priorité relative entre plusieurs enregistrements NAPTR qui ont la même valeur de **order** ;
3. **Flags** : indique par exemple si l'enregistrement décrit une réécriture transitoire (dont le résultat est un nom de domaine pointant sur un autre enregistrement NAPTR) ou une réécriture finale ; la sémantique précise du paramètre **flags** dépend de l'application DDDS ('Dynamic Delegation Discovery System', RFC 3401[27]) employée (ENUM en est une parmi d'autres) ;
4. **Services** : décrit le service de réécriture ; par exemple dans ENUM, la valeur de **services** spécifie le type de l'URI résultante ; la sémantique précise de ce paramètre dépend également de l'application DDDS employée ;
5. **Regexp** : l'opération de réécriture elle-même, formalisée en une expression rationnelle ; cette expression rationnelle est à appliquer à la clé ; ne peut être fourni en même temps que **replacement** ;
6. **Replacement** : nom de domaine pointant sur un autre enregistrement NAPTR, permettant par exemple une réécriture transitoire par délégation ; ne peut être fourni en même temps que **regexp**.

L'enregistrement NAPTR est défini par la RFC 3403[26].

SOA record

Cet enregistrement permet d'indiquer le serveur de nom maître (primaire), l'adresse e-mail d'un contact technique (avec @ remplacé par un point) et des paramètres d'expiration.

Il désigne l'autorité (start of authority) ou le responsable de la zone dans la hiérarchie DNS. C'est l'acte de naissance de la zone DNS.

Ces paramètres sont dans l'ordre :

```
wikipedia.org. IN SOA ns0.wikimedia.org. hostmaster.wikimedia.org. 2010060311 43200 7200 1209600 3600
```

1. **Serial** : indique un numéro de version pour la zone (32 bits non signé). Ce nombre doit être incrémenté à chaque modification du fichier zone ; on utilise par convention une date au format « yyyymmddnn » (« yyyy » pour l'année

sur 4 chiffres, « mm » pour le mois sur 2 chiffres, « dd » pour le jour sur 2 chiffres, « nn » pour un compteur de révision si le numéro de série est modifié plusieurs fois dans un même jour. Cette convention évite tout débordement du 32 bits non signé jusqu'en l'an 4294) ;
2. **Refresh** : l'écart en secondes entre les demandes successives de mise à jour réalisées depuis le serveur secondaire ou les serveurs esclaves ;
3. **Retry** : le délai en secondes que doivent attendre le serveur secondaire ou les serveurs esclaves lorsque leur précédente requête a échoué ;
4. **Expire** : le délai en secondes au terme duquel la zone est considérée comme invalide si le secondaire ou les esclaves ne peuvent joindre le serveur primaire ;
5. **Minimum** ou **negative TTL** : utilisé pour spécifier, en secondes, la durée de vie pendant laquelle sont conservées en cache les réponses qui correspondent à des demandes d'enregistrements inexistants.

Les versions récentes de BIND (*named*) acceptent les suffixes M, H, D ou W pour indiquer un intervalle de temps en minutes, heures, jours ou semaines respectivement.

Time to live

Chaque record est associé à un *Time to live* (TTL) qui détermine combien de temps il peut être conservé dans un serveur *cache*. Ce temps est typiquement d'un jour (86400 s) mais peut être plus élevé pour des informations qui changent rarement, comme des records NS. Il est également possible d'indiquer que des informations ne doivent pas être mises en cache en spécifiant un TTL de zéro.

Certaines applications, comme des navigateurs web disposent également d'un cache DNS, mais qui ne respecte pas nécessairement le TTL du DNS. Ce cache applicatif est généralement de l'ordre de la minute, mais Internet Explorer par exemple conserve les informations jusqu'à 30 minutes[28], indépendamment du TTL configuré.

Glue records

Quand un domaine est délégué à un serveur de noms qui appartient à ce sous-domaine, il est nécessaire de fournir également l'adresse IP de ce serveur pour éviter les références circulaires. Ceci déroge au principe général selon lequel l'information d'un domaine n'est pas dupliquée ailleurs dans le DNS.

Par exemple, dans la réponse suivante au sujet des NS pour le domaine wikimedia.org :

```
wikimedia.org. IN NS ns2.wikimedia.org.
wikimedia.org. IN NS ns1.wikimedia.org.
wikimedia.org. IN NS ns0.wikimedia.org.
```

Il est nécessaire de fournir également les adresses IP des serveurs indiqués dans la réponse (glue records[29]), car ils font partie du domaine en question :

```
ns0.wikimedia.org. IN A 208.80.152.130
ns1.wikimedia.org. IN A 208.80.152.142
ns2.wikimedia.org. IN A 91.198.174.4
```

Mise à jour dynamique

Une extension du DNS nommée DNS dynamique (DDNS) permet à un client de mettre à jour une zone avec des informations qui le concernent (RFC 2136[30]). Ceci est utile quand des clients obtiennent une adresse IP par DHCP et qu'ils souhaitent que le DNS reflète le nom réel de la machine.

Considérations opérationnelles

Mise à jour du DNS

Les mises à jour se font sur le serveur primaire du domaine, les serveurs secondaires recopiant les informations du serveur primaire dans un mécanisme appelé transfert de zone. Pour déterminer si un transfert de zone doit avoir lieu, le serveur secondaire consulte le numéro de version de la zone et le compare à la version qu'il possède. Le serveur primaire détermine à quelle fréquence le numéro de version est consulté. Quand un changement est effectué, les serveurs envoient des messages de notification aux serveurs secondaires pour accélérer le processus.

Il se peut que des informations qui ne sont plus à jour soient cependant conservées dans des serveurs cache. Il faut alors attendre l'expiration de leur *Time to live* pour que ces informations cachées disparaissent et donc que la mise à jour soit pleinement effective. On peut minimiser le temps nécessaire en diminuant le TTL associé aux noms de domaines qui vont être modifiées préalablement à une opération de changement.

Cohérence du DNS

Quand la liste des serveurs de noms change, ou quand une adresse IP qui fait l'objet d'un 'Glue record' est modifiée, le gestionnaire du domaine de niveau supérieur doit effectuer la mise à jour correspondante.

Robustesse du DNS

Pour éviter les points individuels de défaillance, on évite de partager l'infrastructure entre les serveurs qui font autorité. Un serveur secondaire sera de préférence délocalisé et routé différemment que le serveur primaire.

Bien que cela soit techniquement possible, on évite de mêler sur un même serveur le rôle de DNS récursif et celui de serveur qui fait autorité.

De même, un hôte sera configuré avec plusieurs serveurs récursifs, de sorte que si le premier ne répond pas à la requête, le suivant sera employé. En général, les serveurs récursifs fournis par les FAI refusent les requêtes émanant d'adresses IP appartenant à d'autres FAI.

Il existe des services de DNS récursifs ouverts, c'est-à-dire qu'ils acceptent les requêtes de tous les clients. Il est donc possible à un utilisateur de configurer ceux-ci en lieu et place de ceux fournis par le FAI. Ceci pose cependant les problèmes suivants :

- il n'y a pas de garantie que les réponses fournies seront les mêmes qu'avec des serveurs récursifs habituels. Un tel service pourrait en effet faire référence à une autre hiérarchie depuis la racine, disposer de TLD additionnels non standard, restreindre l'accès à certains domaines, voire altérer certains records avant leur transmission au client.
- il n'y a pas de garantie de confidentialité, c'est-à-dire que ce service pourrait déterminer à quels domaines un utilisateur a accès en conservant des traces des requêtes DNS.

Sécurité du DNS

Le protocole DNS a été conçu avec un souci minimum de la sécurité. Plusieurs failles de sécurité du protocole DNS ont été identifiées depuis. Les principales failles du DNS ont été décrites dans le RFC 3833[31] publié en août 2004.

Interception des paquets

Une des failles mises en avant est la possibilité d'intercepter les paquets transmis. Les serveurs DNS communiquent au moyen de paquets uniques et non signés. Ces deux spécificités rendent l'interception très aisée. L'interception peut se concrétiser de différentes manières, notamment via une attaque de type « man in the middle », de l'écoute des données transférées et de l'envoi de réponse falsifiée (voir paragraphe ci-dessous).

Fabrication d'une réponse

Les paquets des serveurs DNS étant faiblement sécurisés, authentifiés par un numéro de requête, il est possible de fabriquer de faux paquets. Par exemple, un utilisateur qui souhaite accéder au site http://mabanque.example.com fait une demande au site DNS. Il suffit, à ce moment, qu'un pirate informatique réponde à la requête de l'utilisateur avant le serveur DNS pour que l'utilisateur se retrouve sur un site d'hameçonnage.

Corruption des données

La trahison par un serveur, ou corruption de données, est, techniquement, identique à une interception des paquets. La seule différence venant du fait que l'utilisateur envoie volontairement sa requête au serveur. Cette situation peut arriver lorsque, par exemple, l'opérateur du serveur DNS souhaite mettre en avant un partenaire commercial.

Empoisonnement du cache DNS

L'empoisonnement du cache DNS ou pollution de cache DNS (en anglais, DNS cache poisoning) est une technique permettant de leurrer les serveurs DNS afin de leur faire croire qu'ils reçoivent une requête valide tandis qu'elle est frauduleuse[32].

Déni de service

Une attaque par déni de service (ou attaque par saturation; en anglais, Denial of Service attack ou DoS attack) est une attaque sur un serveur informatique qui résulte en l'incapacité pour le serveur de répondre aux requêtes de ses clients.

DNSSEC

Pour contrer ces vulnérabilités (corruption des données, empoisonnement de cache DNS, etc), le protocole DNSSEC (RFC 4033, RFC 4034, RFC 4035) a été développé. Il utilise les principes de cryptographie asymétrique et de signature numérique pour garantir l'intégrité des données, ainsi qu'une preuve de non-existence si l'enregistrement demandé n'existe pas. La zone racine du DNS a été signée le 15 juillet 2010[33], et le déploiement de DNSSEC sur les domaines de premier niveau (TLD : Top Level Domain) continue, une liste des domaines couverts étant disponible.

Chiffrement

Depuis 2015[34], l'IETF travaille à la sécurité du canal de communication du DNS (là où DNSSEC protège les données). Cela a débouché sur la publication de plusieurs RFC permettant l'utilisation de TLS afin de chiffrer la communication entre les clients DNS et les résolveurs. Il s'agit principalement de : DNS sur TLS **(en)** (RFC 7858[35], utilisant le port 853) et DNS sur HTTPS (RFC 8484[11], requête DNS encapsulée dans une requête HTTP, et traitée par un serveur Web).

Il n'y a pas, en 2018, de possibilités de chiffrer – via TLS – les communications entre un résolveur et un serveur faisant autorité.

Exemple d'attaques majeures contre des serveurs DNS

En juillet 2008, quelques jours après la publication du rapport de la United States Computer Emergency Readiness Team concernant la faille de sécurité des serveurs DNS permettant d'empoisonner leur cache, plusieurs serveurs DNS majeurs ont subi des attaques. Une des plus importantes fut celle menée contre les serveurs de AT&T. L'attaque empoisonnant le cache des serveurs DNS de AT&T a permis au pirate informatique de rediriger toutes les requêtes de Google vers un site d'hameçonnage[36].

Détails du protocole

DNS utilise en général UDP et le port 53. La taille maximale des paquets utilisée est de 512 octets. Si une réponse dépasse cette taille, la norme prévoit que la requête doit être renvoyée sur le port TCP 53. Ce cas est cependant rare et évité, et les firewalls bloquent souvent le port TCP 53. Les transferts de zone s'effectuent par TCP sur le même numéro de port. Pour des raisons de

sécurité, les serveurs restreignent généralement la possibilité de transférer des zones.

L'extension EDNS0 (RFC 2671[37]) permet d'utiliser une taille de paquets plus élevée, sa prise en charge est recommandée pour IPv6 comme pour DNSSEC.

La norme prévoit qu'il existe une *classe* associée aux requêtes. Les classes IN (Internet), CH (Chaos) et HS (Hesiod **(en)**) sont définies, seule la classe IN étant réellement utilisée en pratique. La classe *chaos* est utilisée par BIND pour révéler le numéro de version[38].

Exemples de consultation DNS

Pour vérifier l'association entre un nom et une adresse IP, plusieurs commandes sont disponibles suivant les systèmes d'exploitation utilisés.

Par exemple sur Windows la commande nslookup est disponible via l'invite de commande :

```
> nslookup www.google.fr
Serveur : Livebox-6370
Address: 192.168.1.1

Réponse ne faisant pas autorité :
Nom : www.l.google.com
Addresses:
         209.85.229.104
         209.85.229.106
         209.85.229.103
         209.85.229.147
         209.85.229.105
         209.85.229.99
Aliases: www.google.fr
          www.google.com
```

ou encore dig sur les systèmes compatibles avec UNIX :

```
> dig www.google.com aaaa

; <<>> DiG 9.7.0-P1 <<>> www.google.com aaaa
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47055
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 4, ADDITIONAL: 0

;; QUESTION SECTION:
;www.google.com.            IN  AAAA

;; ANSWER SECTION:
www.google.com.     422901  IN  CNAME   www.l.google.com.
www.l.google.com.   77  IN  AAAA    2a00:1450:8004::67
www.l.google.com.   77  IN  AAAA    2a00:1450:8004::68
www.l.google.com.   77  IN  AAAA    2a00:1450:8004::69
www.l.google.com.   77  IN  AAAA    2a00:1450:8004::6a
www.l.google.com.   77  IN  AAAA    2a00:1450:8004::93
www.l.google.com.   77  IN  AAAA    2a00:1450:8004::63

;; AUTHORITY SECTION:
google.com.     155633  IN  NS  ns2.google.com.
google.com.     155633  IN  NS  ns1.google.com.
google.com.     155633  IN  NS  ns3.google.com.
google.com.     155633  IN  NS  ns4.google.com.

;; Query time: 0 msec
;; SERVER: ::1#53(::1)
;; WHEN: Sun May 23 16:23:49 2010
;; MSG SIZE rcvd: 292
```

Notes et références

1. **(en)** Request for Comments nº 882 (https://tools.ietf.org/html/rfc882).
2. **(en)** Request for Comments nº 883 (https://tools.ietf.org/html/rfc883).
3. **(en)** Request for Comments nº 1034 (https://tools.ietf.org/html/rfc1034).
4. **(en)** Request for Comments nº 1035 (https://tools.ietf.org/html/rfc1035).
5. **(en)** Request for Comments nº 1591 (https://tools.ietf.org/html/rfc1591).
6. **(en)** Request for Comments nº 6195 (https://tools.ietf.org/html/rfc6195).
7. **(en)** Request for Comments nº 6895 (https://tools.ietf.org/html/rfc6895).
8. **(en)** Request for Comments nº 8375 (https://tools.ietf.org/html/rfc8375).
9. **(en)** Request for Comments nº 8467 (https://tools.ietf.org/html/rfc8467).
10. **(en)** Request for Comments nº 8483 (https://tools.ietf.org/html/rfc8483).
11. **(en)** Request for Comments nº 8484 (https://tools.ietf.org/html/rfc8484).
12. **(en)** Request for Comments nº 8499 (https://tools.ietf.org/html/rfc8499).
13. **(en)** Request for Comments nº 608 (https://tools.ietf.org/html/rfc608).
14. IEN 116 (http://www.postel.org/ien/pdf/ien116.pdf) *Internet Name Server*, Jon Postel 1979
15. *Development of the Domain Name System (http://cseweb.ucsd.edu/classes/wi01/cse222/papers/mockapetris-dns-sigcomm88.pdf)*, Paul Mockapetris, Kevin Dunlap, Sigcomm 1988
16. **(en)** Request for Comments nº 1912 (https://tools.ietf.org/html/rfc1912).
17. Voir la section *4.4 Usage and deployment considerations* du draft draft-ietf-dnsop-reverse-mapping-considerations (http://tools.ietf.org/html/draft-ietf-dnsop-reverse-mapping-considerations-06)
18. **(en)** Request for Comments nº 2317 (https://tools.ietf.org/html/rfc2317).
19. **(en)** « named.root » (https://www.internic.net/domain/named.root), sur *www.internic.net*
20. « Root Server Technical Operations Assn » (https://root-servers.org/), sur *root-servers.org* (consulté le 28 avril 2019)
21. k statistics (http://k.root-servers.org/index.html#stats)
22. **(en)** Request for Comments nº 3490 (https://tools.ietf.org/html/rfc3490).
23. RFC 1035, chapitre 3.2.1
24. « Domain Name System (DNS) Parameters » (https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml), sur *www.iana.org* (consulté le 28 avril 2019)
25. **(en)** Request for Comments nº 2782 (https://tools.ietf.org/html/rfc2782).
26. **(en)** Request for Comments nº 3403 (https://tools.ietf.org/html/rfc3403).
27. **(en)** Request for Comments nº 3401 (https://tools.ietf.org/html/rfc3401).
28. « Comment Internet Explorer utilise le cache pour les entrées d'hôte DNS » (https://support.microsoft.com/fr-fr/help/263558/how-internet-explorer-uses-the-cache-for-dns-host-entries), sur *support.microsoft.com* (consulté le 28 avril 2019)
29. « Glue Records (enregistrements Glue) — Documentation Documentation Gandi » (https://docs.gandi.net/fr/noms_domaine/utilisateurs_avances/glue_records.html), sur *docs.gandi.net* (consulté le 28 avril 2019)
30. **(en)** Request for Comments nº 2136 (https://tools.ietf.org/html/rfc2136).
31. **(en)** Request for Comments nº 3833 (https://tools.ietf.org/html/rfc3833).
32. **(en)** « Multiple DNS implementations vulnerable to cache poisoning » (https://www.kb.cert.org/vuls/id/800113/), sur *www.kb.cert.org* (consulté le 28 avril 2019)
33. **(en-US)** « Root DNSSEC » (https://www.root-dnssec.org/) (consulté le 25 août 2019)
34. « Dprive Status Pages » (https://tools.ietf.org/wg/dprive/), sur *tools.ietf.org* (consulté le 28 avril 2019)
35. **(en)** Request for Comments nº 7858 (https://tools.ietf.org/html/rfc7858).
36. **(en)** « DNS Attack Writer a Victim of His Own Creation » (https://www.pcworld.com/article/149126/dns_attack_writer.html), sur *PCWorld*, 29 juillet 2008 (consulté le 28 avril 2019)
37. **(en)** Request for Comments nº 2671 (https://tools.ietf.org/html/rfc2671).
38. dig CH @k.root-servers.net version.bind txt

Voir aussi

Articles connexes

- *DNS black holing*
- Dig
- *DNS Black Listing*
- Empoisonnement du cache DNS
- Hébergement de nom de domaine

Sur les autres projets Wikimedia :

Domain Name System, sur Wikibooks

- host
- *Hosts*
- ICANN
- Manipulation de l'espace des noms de domaine (*DNS menteurs*)
- nslookup
- Serveur racine du DNS
- RadioDNS

Liens externes

- Auto-formation au DNS par l'AFNIC (http://www.afnic.fr/ext/dns/)
- DNS dans tous ses détails (http://www.frameip.com/dns/)
- DNS sur le site commentcamarche.net (http://www.commentcamarche.net/internet/dns.php3)
- Support Cours de l'UREC/CNRS sur le DNS (http://www.urec.cnrs.fr/IMG/pdf/cours.dns.pdf) **[PDF]**
- Tester la mise à jour des DNS (http://www.firasofting.com/blog/2011/tester-les-serveurs-dns/)
- **(en)** RFC6195 relatives au DNS (http://tools.ietf.org/html/rfc6195)
- **(en)** RFC1035 relatives au DNS (http://tools.ietf.org/html/rfc1035)
- **(en)** Information sur le DNS (http://www.iana.org/protocols/)
- **(en)** Bonne explication des NAPTR par Nominet (http://blog.nominet.org.uk/tech/2006/07/14/naptr-records/)