**Domain Name System**

Le **Domain Name System** (ou **DNS**, système de noms de domaine) est un service permettant de traduire un nom de domaine en informations de plusieurs types qui y sont associées, notamment en adresses IP de la machine portant ce nom. À la demande de la DARPA, Jon Postel et Paul Mockapetris ont conçu le « *Domain Name System* » en 1983 et en écrivirent la première réalisation.



La DARPA

**Rôle du DNS**

Les ordinateurs connectés à un réseau IP, comme Internet, possèdent une adresse IP. Ces adresses sont numériques afin d'être plus facilement traitées par une machine. En IPv4, elles sont représentées sous la forme *xxx.xxx.xxx.xxx*, où xxx est un nombre variant entre 0 et 255 (en système décimal). En IPv6, les IP sont de la forme xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx, où x représente un caractère au format hexadécimal. Pour faciliter l'accès aux systèmes qui disposent de ces adresses, un mécanisme a été mis en place pour permettre d'associer un nom à une adresse IP, plus simple à retenir, appelé *nom de domaine*. *Résoudre* un nom de domaine consiste à trouver l'adresse IP qui lui est associée.

Les noms de domaines peuvent être également associés à d'autres informations que des adresses IP.

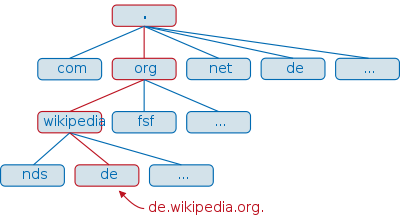


Représentation de l’internet

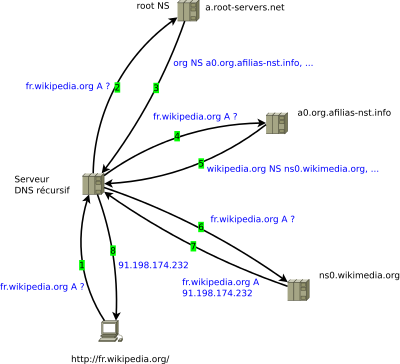
**Histoire**

Avant le DNS, la résolution d'un nom sur Internet devait se faire grâce à un fichier texte appelé *HOSTS.TXT* (RFC 608) maintenu par le NIC du Stanford Research Institute (SRI) et recopié sur chaque ordinateur 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 1161. Le premier est jugé trop compliqué tandis que le second est insuffisant2. La tâche de développer un autre système revient à Paul Mockapetris qui publie le design du système dans les RFC 882 et RFC 883 en 1983. La norme correspondante est publiée dans les RFC 1034 et RFC 1035 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**

[](http://commons.wikimedia.org/wiki/File:Dns-raum.svg)

Hiérarchie du DNS

[](http://commons.wikimedia.org/wiki/File:DNS_iterations.svg)

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)

**Hiérarchie du DNS**

Le système des noms de domaines 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.

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…), on les appelle ccTLD (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 qui 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 ouverts comme ceux de Google Public DNS ou OpenDNS.

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) qui peut entrainer 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 exploitable. 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 plusieurs différents noms de domaine via l'enregistrement de plusieurs entrées PTR dans le sous-domaine *.arpa* dédié 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 DNS3.

**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 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 hote.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 IPv64. Grâce à anycast, plus de 200 serveurs répartis dans 50 pays du monde assurent ce service5. Il existe 13 autorités de nom appelées de a à m.root-servers.net6. Le serveur *k* reçoit par exemple de l'ordre de 20 000 requêtes par seconde7.

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 à ce 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 définit un format appelé Punycode qui permet l'encodage d'un jeu de caractère plus étendu.

**Technique du *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), qui consiste à associer plusieurs adresses IP à un nom de domaine. 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 bits8, l'IANA conserve le registre des codes assignés9. Les principaux enregistrements définis sont les suivants :

* **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** 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 ;
* **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 ;
* **TXT record** permet à un administrateur d'insérer un texte quelconque dans un enregistrement DNS (par exemple, cet enregistrement était 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). Cet enregistrement aurait un intérêt majeur mais n'est malheureusement que très rarement utilisé sur le monde Internet.

**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 à 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 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 (DNS cache poisoning ou DNS cache pollution en français) est une technique permettant de leurrer les serveurs DNS afin de leur faire croire qu'ils reçoivent une requête valide tandis qu'elle est frauduleuse11.

**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, le protocole DNSSEC a été développé.



DNSSEC

**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 à Google vers un site d'hameçonnage12.

**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) 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 version13.

**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. Pour exemple sur Windows la commande nslookup est disponible via l'invite de commande :

C:\>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.

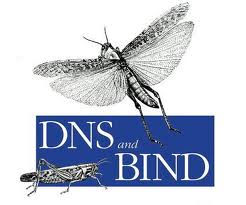
google.com. 155633 IN NS ns4.google.com.

;; Query time: 0 msec

;; SERVER: ::1#53(::1)

;; WHEN: Sun May 23 16:23:49 2010

;; MSG SIZE rcvd: 292



Dns et Bind