Network Working Group
Request for Comments: 1034
Obsoletes: RFCs 882, 883, 973

P. Mockapetris
ISI

November 1987

NOMS DE DOMAINES - CONCEPTS ET ELEMENTS DE BASE

1. STATUT DE CE MEMO

Cette RFC est une introduction au système de noms de domaines (Domain Name System) DNS, et passe sous silence de nombreux détails qui pourront être trouvés dans une RFC complémentaire, "Domain Names - Implementation and Specification" [RFC-1035]. Cette dernière suppose que le lecteur est déjà familiarisé avec les concepts énoncés dans le présent.

Un sous-ensemble des fonctions DNS et des types de données associés constitue un protocole officiel. Le protocole officiel comprend la définition des requêtes standard et des réponses qui y sont faites ainsi que la plupart des formats de classes de données Internet (ex., adresses d'hôtes).

Cependant, le système de domaines est intentionnellement extensible. Les chercheurs proposent continuellement, implémentent et expérimentent de nouveaux types de données, types de requêtes, classes, fonctions, etc. De ce fait, alors que les constituants du protocole officiel sont supposés rester stables et pouvoir être utilisés en exploitation, des comportements expérimentaux pourront être observés au delà des limites de la définition officielle. Les RFC concernées mentionnent clairement les fonctions expérimentales, ou obsolètes, et l'information qui y est reportée doit toujours être considérée avec précaution.

Il est signalé au lecteur de ne jamais considérer les valeurs données à titre d'exemple comme opérationnelles ou complètes, dans la mesure où leur utilisation n'est faite que dans un but pédagogique. La distribution de ce mémo est illimitée.

2. INTRODUCTION

Cette RFC expose les styles admis pour les noms de domaines, leur utilisation dans le cadre de la messagerie par Internet et pour la recherche d'hôtes, et décrit les protocoles et serveurs utilisés pour les services réseaux liés aux noms de domaines.

2.1. Historique des noms de domaines

La motivation essentielle et impérieuse conduisant à la mise en œuvre du système de domaines a été la croissance exponentielle d'Internet :

- Au début, la transcription de noms d'hôtes en adresses Internet s'appuyait sur une table de correspondance maintenue par le Network Information Center (NIC) sous forme d'un unique fichier (HOSTS.TXT) lequel était transmis par FTP sur tous les hôtes [RFC-952, RFC-953]. La bande passante consommée par la distribution d'une remise à jour de cette base par cette méthode est proportionnelle au carré du nombre d'hôtes sur le réseau, et même lorsque plusieurs niveaux de retransmissions FTP étaient utilisés, le trafic sortant du serveur NIC restait considérable. La croissance explosive du nombre d'hôtes a fait rapidement exploser du même coup ce mécanisme.
- La population internaute changeait dans le même temps de nature. Les hôtes en temps partagé (mainframes) constituant l'ARPANET originel ont été remplacés par une architecture distribuée de stations connectées sur des réseaux et sous-réseaux locaux. Les organismes locaux ont commencé à administrer leurs propres noms et adresses, mais devaient attendre que le NIC reporte les changements dans le fichier HOSTS.TXT pour que ceux-ci soient visibles de l'ensemble de la communauté Internet. Les organisations souhaitaient néanmoins pouvoir conserver une certaine autonomie quant à la gestion de leur infrastructure locale.
- Les applications exploitant Internet sont devenues de plus en plus sophistiquées et ont créé le besoin d'un traitement plus généralisé des noms de domaines.

Le résultat de tout ceci ont fait émerger certaines idées sur l'espace des noms et sa gestion[IEN-116, RFC-799, RFC-819, RFC-830]. Les propositions ont été diverses, mais l'une des tendances émergentes a été celle d'un espace de noms hiérarchisé, et dont le principe hiérarchique s'appuierait autant que possible sur la structure des organismes eux-mêmes, et où les noms utiliseraient le caractère "." pour marquer la frontière entre deux niveaux hiérarchiques. Un design mettant en œuvre une base de données distribuée et des ressources généralisées a été décrit dans les [RFC-882, RFC-883]. Sur la base de nombreuses expérimentations, le système théorique des domaines a évolué vers celui qui est présenté dans ce mémo.

Les termes "domaine" ou "nom de domaine" sont utilisés dans de nombreux contextes tournant autour du DNS décrit ici. Très souvent, le terme "nom de domaine" est souvent utilisé pour décrire un nom écrit sous forme de chaîne de caractères reliées par des points, sans relation expresse au DNS. Ceci est particulièrement vrai pour ce qui concerne les adresses de messagerie [Quarterman 86].

2.2. Objectifs de la conception du DNS

Les objectifs dans lesquels a été conçu le DNS ont influencé sa structure. Ces objectifs sont les suivants :

- Le but premier est la création d'un espace de noms conséquent utilisables pour référencer des ressources. Pour éviter tout problème de transcodage ou d'encodage, les noms peuvent ne mentionner ou inclure aucun des identificateurs réseau, adresses, chemins, ou information similaire communément utilisés pour l'implémentation technique.
- La taille énorme de la base de données et sa fréquence de mise à jour suggère une maintenance distribuée, avec cache local pour une performance accrue. Toute approche qui tentera d'obtenir une copie intégrale de la base de donnée sera de plus en plus coûteuse et difficile à traiter, et de ce fait devra être écartée. Les mêmes principes courent pour la constitution de l'espace des noms, et en particulier les mécanismes pour créer et supprimer des noms ; ceux-ci devront être également distribués.
- Lorsque l'on doit composer entre le coût technique d'acquisition d'une donnée, la fréquence des mises à jour, et la validité des caches, c'est toujours la source première de données qui contrôle les priorités à donner.
- Le coût important inhérant à l'implémentation d'un tel service suppose qu'il a une utilité générale, qui n'est pas restreinte à une application particulière. Les noms ainsi constitués devront pouvoir servir à identifier des hôtes, récupérer des courriers dans des boîtes aux lettres, et toute autre information non encore identifiée. Toute donnée associée à un nom sera typée, et les requêtes sur ce nom seront limitées à ce seul type.
- Nous souhaitions que l'espace de noms puisse être utilisé sur des réseaux de nature différente, et pour cela, nous avons conçu ce système de telle sorte qu'il puisse s'appuyer sur plusieurs familles de protocoles. Par exemple, les adresses d'hôtes diffèrent dans leur forme suivant la nature des systèmes, bien que tous les protocoles utilisent une notion similaire. Le DNS attribue une classe aux données de la même façon qu'il attribue un type, ce qui nous permettra de pouvoir parallèlement utiliser plusieurs formats distincts pour des données de type adresse.
- Nous souhaitions de plus que les transactions avec les serveurs de nom soient indépendantes du système de communication utilisé. Certains systèmes utiliseront le format du datagramme pour les requêtes et les réponses, et n'établiront de circuit commuté virtuel que lorsque la transaction nécessite une certaine sécurité (ex., la mise à jour des bases de données, des transactions de longue durée); d'autres systèmes demanderont à n'utiliser que des circuits commutés virtuels.
- Le système doit être compatible avec une grande variété de plates-formes. Devront pouvoir utiliser ce système tant des micro-ordinateurs que des "mainframes", les méthodes d'utilisation pouvant être différentes.

2.3. Présupposés concernant l'utilisation

L'organisation du système de domaines découle de certains présupposés quant aux besoins et aux schémas d'exploitation de la communauté utilisatrice, et est conçue de sorte à éviter un grand nombre de problèmes classiques des grandes bases de données généralistes.

Les suppositions faites sont :

- La taille totale de la base de données sera initialement proportionnelle au nombre d'hôtes utilisant le système, mais pourra rapidement devenir proportionnelle au nombre d'utilisateurs

utilisant ces machines dans la mesure où des informations telles que les boîtes aux lettres y seront intégrées.

- La fréquence de modification de la plupart des données de cette base sera assez basse (ex., les changements de boîtes aux lettres, les adresses d'hôtes), mais le système devra pouvoir traiter des sous ensembles de données nécessitant une période de remise à jour plus élevée (de l'ordre de quelques secondes à quelques minutes).
- Les limites administratives définies pour répartir la responsabilité de gestion de la base de données seront généralement associées à celles d'organisations possédant un ou plusieurs hôtes. Chaque organisation ayant la responsabilité d'un ensemble de domaines particulier devra mettre en œuvre plusieurs serveurs de domaines redondants, soit sur l'hôte même de l'organisme, ou sur d'autres hôtes dont l'organisme s'occupe ou exploite.
- Les clients du système de domaines devront pouvoir choisir le serveur qu'ils décident d'utiliser parmi un ensemble de serveurs nommés et considérés comme sûrs avant d'accepter de s'appuyer sur un serveur hors de cet ensemble.
- L'accès à l'information est plus important que la garantie de remise à jour instantanée et d'une consistance permanente de la base. De ce fait le processus de remise à jour utilise un principe de diffusion de l'information de proche en proche plutôt qu'un mécanisme dont le but serait de remettre à jour simultanément toutes les copies d'une information. Lorsque les mises à jour sont indisponibles suite à une défaillance réseau ou de l'hôte, il sera d'usage de s'en remettre à l'information "ancienne", pendant que les efforts sont faits pour remettre à jour la base. Le modèle général précise que les copies d'informations sont faites tenant compte d'un certain délai de rafraîchissement. Le distributeur mentionne le délai et le récepteur des données est responsable de l'opération de remise à jour. Dans certains cas très particuliers, des délais très courts peuvent être spécifiées, ou encore la copie peut être interdite.
- Dans tout système possédant une base de données répartie, un serveur de nom pourra recevoir des requêtes auxquelles seuls d'autres serveurs peuvent répondre. Les deux approches principales pour contourner le problème sont soit la méthode "récursive", par laquelle le serveur reporte la requête vers un autre serveur pour le compte du client, soit la méthode "itérative", par laquelle le client est enjoint de requérir sur un autre serveur. Les deux approches ont leurs avantages et inconvénients, mais la méthode itérative reste toutefois préférée dans le cas où le mode de requête est le datagramme. Le système de domaines nécessitera l'implémentation de l'approche itérative, mais fournira la méthode récursive en option.

Le système de domaines suppose que toutes les données proviennent de fichiers maîtres éparpillés dans les hôtes parties prenantes de ce système. Ces fichiers maîtres de données sont maintenus par l'administrateur local. Les fichiers maîtres sont des fichiers texte lus par un serveur de domaines local, et qui deviennent de ce fait accessibles à tous les utilisateurs du système de domaines par l'intermédiaire de la chaîne de serveurs. Le programme de l'utilisateur accède à ces différents serveurs par l'intermédiaire d'une fonction logicielle de "résolution d'adresse".

Le format standardisé des fichiers maîtres leur permet d'être échangés entre hôtes différents (via FTP, mail, ou tout autre mécanisme); cette opportunité est utile lorsque par exemple, un organisme désire s'attribuer un domaine, mais ne souhaite pas supporter l'administration d'un serveur de domaines. L'organisme pourra maintenir localement le fichier maître avec un simple éditeur de texte, puis le transférer sur un hôte déporté sur lequel sont exécutés les serveurs de domaines, puis voir avec l'administrateur système pour savoir quel serveur de domaines ira lire les fichiers ainsi chargés.

Chaque hôte gérant un serveur de noms de domaines et une fonction de résolution d'adresse est configuré par un administrateur local [RFC-1033]. Pour un serveur de noms, cette configuration définit entre autres l'identité des fichiers maîtres locaux ainsi que des instructions pour savoir quels fichiers maîtres externes doivent être chargés et à partir de quels serveurs distants. Le serveur de noms

utilise les fichiers principaux ou ses copies pour charger ces zones. Pour les programmes de résolution d'adresse, les données de configuration identifient les serveurs de noms qui seront les sources primaires d'information.

Le système de domaines définit des procédures pour accéder aux données ou pour faire référence à d'autres serveurs de noms. Le système de domaines définit aussi des procédures pour stocker les données récupérées et pour rafraîchir périodiquement les données selon les voeux de l'administrateur système.

L'administrateur renseigne :

- La définition des limites de zones.
- Les fichiers principaux de données.
- La remise à jour des fichiers de données.
- Les spécifications de cette remise à jour.

Le système de domaines fournit :

- Des formats standard pour les ressources.
- Des méthodes standard d'accès à la base de données.
- Des méthodes standard à l'attention des serveurs de noms pour rafraîchir les données à partir de serveurs de noms distants.

2.4. Eléments du DNS

Le DNS a trois composants principaux :

- L'ESPACE DE NOMS DE DOMAINES et les ENREGISTREMENTS DE RESSOURCES, qui sont les spécifications d'un espace de noms structuré en arbre et des données associées à ces noms. Conceptuellement, chaque noeud et chaque feuille de l'arbre de l'espace de noms de domaines contient un ensemble d'informations ; les requêtes sont des tentatives pour extraire un type spécifique d'information dans cet ensemble. Une requête cite le nom du domaine d'intérêt et décrit le type d'information désiré quant aux ressources concernées. Par exemple, Internet utilise certains de ses noms de domaines pour identifier des hôtes ; une requête pour des adresses de ressources renverront l'adresse Internet de l'hôte.
- Les SERVEURS DE NOM sont des programmes serveurs qui détiennent l'information sur la structure arborescente et les informations de domaines. Un serveur de nom peut stocker momentanément en "cache" des informations de structure ou de ressources sur toute partie de l'espace de noms de domaines, mais en général, un serveur de nom n'accueillera que les informations relatives à un sous ensemble de l'espace, et des pointeurs vers d'autres serveurs de noms qui, par leur association, se répartissent la définition de l'ensemble de l'espace. Les serveurs de nom connaissent la partie de l'arbre des domaines pour laquelle il détiennent une information complète ; un serveur de noms est dit être AUTORISE pour cette partie de l'espace de noms. L'information "autorisée" est organisée en unités appelées ZONES, ces zones pouvant être automatiquement distribuées aux serveurs de noms faisant partie de la "sphère de redondance" pour la zone de données considérées.
- Les processus de résolution, ou RESOLVEURS sont des programmes qui extraient l'information des serveurs de noms en réponse aux requêtes clientes. Les résolveurs doivent pouvoir accéder à au moins un serveur de noms et utiliser l'information qu'ils y trouvent pour donner directement une réponse au client, ou utiliser les références à d'autres serveurs de nom contenues dans le serveur "visible" pour les contacter à leur tour et continuer la

résolution. un résolveur sera habituellement une routine système qui peut être appelée directement par un programme utilisateur ; en général aucun protocole n'est nécessaire entre le résolveur et l'application utilisatrice.

Ces trois composants correspondent en gros aux trois "couches" ou points de vue sur le système denoms de domaines :

- Du point de vue de l'utilisateur, le système de noms de domaines est accessible via une procédure simple ou un appel système à un résolveur local. L'espace de domaines consiste en un arbre unique dont toutes les parties sont accessibles à l'utilisateur.
- Du point de vue du résolveur, le système de domaines est composé d'un nombre non connu de serveurs de noms. Chaque serveur de noms héberge une ou plusieurs pièces de l'ensemble des données constituant l'arbre des domaines, le résolveur considérant chacune de ces bases de données comme essentiellement statique.
- Du point de vue d'un serveur de noms, le système de domaines consiste en un regroupement d'ensembles de données locales séparées appelées zones. Le serveur de noms dispose d'une copie locale de certaines zones. Le serveur de noms doit rafraîchir périodiquement ses zones à partir de fichiers principaux locaux ou situés dans des serveurs de noms distants. Les serveurs de noms doivent traiter les requêtes arrivant des résolveurs de façon concurrente.

Pour une meilleure performance, les implémentations pourront coupler ces fonctions. Par exemple, un résolveur exécuté sur la même machine qu'un serveur de noms pourrait partager la base de données accueillant les zones gérées par le serveur de nom et le cache géré, lui, par le résolveur.

3. ESPACE DE NOMS DE DOMAINES et ENREGISTREMENTS DE RESSOURCES

3.1. Spécifications et terminologie des Noms de Domaine

L'espace de noms de domaines est une structure arborescente. Chaque noeud et feuille de l'arbre correspond à un ensemble de ressources (qui peut être vide). Le système de domaines ne fait aucune distinction entre l'utilisation des feuilles et de la partie information des noeuds, ce mémo n'utilisant par la suite que le terme "noeud" pour référencer les deux types d'entités.

Chaque noeud dispose d'un identifiant, d'une longueur de zéro à 63 octets. Des noeuds "frères" ne peuvent pas avoir le même identifiant, bien que le même identifiant puisse se retrouver dans deux noeuds distincts (mais sans relation de fratrie). L'identifiant nul (c-à-d., de longueur zéro) est réservé pour désigner la racine.

Le nom de domaine d'un noeud est constitué de la liste des identifiants de tous les noeuds constituant le chemin entre ce noeud et la racine de l'arbre. Par convention, les identifiants qui composent un nom de domaine seront exprimés ou lus de gauche à droite, du plus spécifique (le plus bas, le plus éloigné de la racine) au plus globalisant (le plus haut, le plus proche de la racine).

En interne, les programmes manipulant les noms de domaines peuvent les représenter comme des séquences d'étiquettes, chacune comportant une octet de longueur suivi d'une chaîne d'octets. Comme tous les noms de domaines se terminent par la racine, laquelle éant identifiée par une chaîne de longueur nulle, ces représentations internes peuvent utiliser l'octet nul pour terminer un nom de domaine (à considérer comme l'octet de longueur du dernier identifiant, qui vaut zéro).

Par convention, les noms de domaines peuvent être stockés avec une casse arbitraire, mais toute comparaison de noms de domaines pour ce qui est des fonctions actuelles sont faites indépendamment de la casse, en supposant l'usage du jeu de caractères ASCII, et le bit de poids fort à zéro dans chque octet. Ceci signifie que vous serez libre de créer un noeud d'identifiant "A" ou encore "a", mais pas les

deux en tant que frères l'un de l'autre ; ces deux domaines pourront être références par "a" ou "A" indistinctement. Cependant, lorsque vous recevez un nom de domaine ou un identifiant, il faudra en préserver la casse. La raison en est qu'il sera peut être nécessaire, dans un futur proche d'étendre les noms de domaines à une représentation binaire complète, dans le but d'accueillir de nouveaux services ; les services existants devant pouvoir rester inchangés.

Lorsqu'un utilisateur doit entrer un nom de domaine, la longueur de chaque identifiant est omise et les identifiants devront être séparés par des points ("."). Un nom de domaine complet atteignant toujours la racine, la forme écrite exacte de tout domaine entièrement qualifié se termine par un point. Nous utiliserons cette propriété pour distinguer les cas :

- d'une chaîne de caractères représentant un nom de domaine complet (souvent appelé "absolu" ou "entièrement qualifié"). Par exemple, "poneria.ISI.EDU."
- d'une chaîne de caractères représentant les premiers identifiants d'un nom de domaine incomplet, et devant être complété par l'application locale avec un complément absolu (expression appelée "relative"). Par exemple, "poneria", à utiliser relativement au domaine ISI.EDU.

Les noms relatifs sont exprimés soit par rapport à une référence absolue connue, ou par rapport à une liste de domaines constituant ainsi une liste de recherche. Les noms relatifs existent souvent au niveau de l'interface utilisateur, où leur interprétation varie d'une implémentation à l'autre, ainsi que dans les fichiers principaux, dans lesquels sont exprimées des domaines relativement à un nom de domaine de référence. L'interprétation la plus courant utilise la racine "." soit comme l'origine simple soit comme l'un des membres d'une liste de recherche, et un nom à multiple identifiants est souvent exprimé sans le point final par "paresse".

Pour simplifier les implémentations, le nombre total d'octets composant un nom de domaine entièrement qualifié (c'est à dire la somme de tous les identifiants plus la mention des longueurs d'identifiants) est limité à 255.

Un domaine est identifié par un nom de domaine, et consiste de la portion de l'espace de domaine située au niveau et en dessous du noeud spécifié par le domaine. Un domaine est le sous-domaine d'un autre domaine s'il est contenu ans ce dernier. Cette relation peut être vérifiée en regardant si le nom du sous-domaine se termine par le nom du domaine le contenant. Par exemple, A.B.C.D est un sous-domaine de B.C.D, C.D, D, et "".

3.2. Conseils pour l'administration

En matière de politique d'administration, les spécifications techniques du DNS n'imposent aucune structure d'arbre particulière ni de règles pour le choix des identifiants ; son but est d'être le plus général possible, et il doit pouvoir servir à toutes sortes d'applications. En particulier, le système était conçu de sorte que l'espace de noms n'ait pas nécessairement à suivre les limites physiques de la constitution du réseau, des serveurs de noms, etc. La motivation de ceci ne signifie pas que le système de noms refuse toute notion de sémantique, mais plutôt que le choix d'une sémantique implicite puisse rester ouvert pour pouvoir s'adapter aux problèmes particuliers lorsqu'ils se présenteraient, et qu'il reste acceptable que certaines sous parties de l'arbre puisse user de sémantiques différentes. Par exemple, le domaine IN-ADDR.ARPA est organisé et distribué en réseaux et adresses d'hôtes car son rôle est la traduction d'adresses complètes d'hôtes en noms ; Les domaines NetBIOS [RFC-1001, RFC- 1002] sont des domaines plats conformément aux besoins de cette application.

Cependant, nous pouvons donner quelques règles à suivre pour des parties "normales" du domaine de noms qui utilisent le schéma réseau/hôte, ou boîtes aux lettres, etc., qui permettent de maintenir l'espace de noms uniforme, préserve sa capacité de croissance, et minimise les problèmes lorsque des logiciels sont mis à jour à partir de tables anciennes. Les premières décisions "politiques" concernant les niveaux haut de l'arbre ont été données dans la RFC-920. La politique actuelle pour les niveaux

haut sont discutées dans la [RFC-1032]. Les conversions pour l'espace MILNET sont couvertes dans la [RFC-1031].

Les domaines de plus bas niveaux qui peuvent à leur tour être subdivisés en zones multiples devront pouvoir proposer des branchements vers le haut du domaine de telle sorte que d'éventuelles décompositions puissent se faire sans changement de noms. Les identifiants de noeuds utilisant des caractères spéciaux, des chiffres en tête, etc., risquent de poser des problèmes à des programmes plus anciens basés sur des choix plus restrictifs.

3.3. Conseils techniques d'utilisation

Avant que le DNS puisse être utilisé pour accueillir les informations de nommage de quelque catégorie d'objets que ce soit, deux besoins doivent être remplis :

- Une convention pour relier les noms d'objets et les noms de domaines. Ceci décrit comment l'on accède à l'information sur un objet.
- les types RR et les formats de données pour la description des objets.

Ces règles peuvent être du plus simple au plus complexe. Très souvent, le concepteur doit prendre en compte les formats existants et doit planifier une compatibilité ascendante pour les utilisations actuelles. Plusieurs conversions et niveaux de conversion peuvent devenir nécessaires.

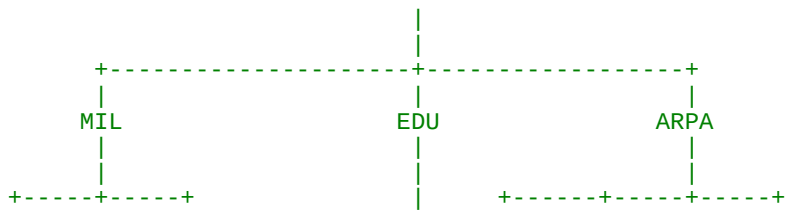
Pour les hôtes, la conversion dépends de la syntaxe actuelle pour les noms d'hôtes, elle-même un sous ensemble de la représentation textuelle courante pour les noms de domaine, ainsi que des formats RR décrivant les adresses des hôtes. Dans la mesure ou nous souhaitons pouvoir disposer d'un transcodage inverse fiable depuis les adresses d'hôtes vers les noms d'hôtes, un codage particulier pour les adresses du domaine IN-ADDR.ARPA est défini.

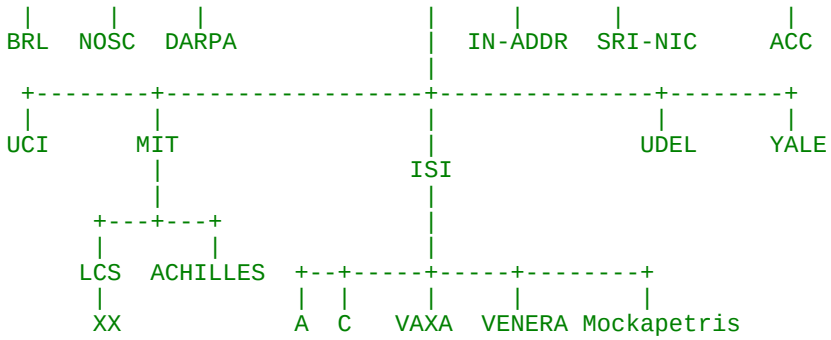
Pour les boîtes aux lettres, le transcodage est légèrement plus complexe. L'adresse Mail habituelle <partie-locale>@<domaine-de-mail> est transcodé en nom de domaine par la conversion de la <partie-locale> en un identifiant simple (sans tenir compte des points qu'elle contient), et en convertissant le <domaine-mail> en un nom de domaine conforme à sa représentation textuelle générique (des points séparant les identifiants), l'opération finale consistant à concaténer les deux parties pour former un nom de domaine unique. Ainsi, la boîte aux lettres HOSTMASTER@SRI-NIC.ARPA est représenté par le nom de domaine HOSTMASTER.SRI-NIC.ARPA. L'appréciation des raisons conduisant à ce design doit aussi prendre en compte le schema des échanges de courrier [RFC- 974].

L'utilisateur type n'est pas concerné par l'établissement de ces règles, mais doit néanmoins comprendre qu'elles résultent de nombreux compromis entre des contraintes de compatibilité ascendante pour d'anciennes applications toujours en service, des interactions entre les différentes définitions d'objets, et l'incontournable urgence qu'il y a à développer de nouvelles fonctionnalités lorsque de nouvelles règles sont établies. La manière dont le DNS est exploité pour prendre en compte tel ou tel objet est souvent plus importante que les restrictions inhérentes au DNS.

3.4. Exemple d'espace de noms

La figure suivante montre une partie de l'espace de noms de domaines actuel, et sert de base d'exemple à de nombreuses reprises dans cette RFC. Notez que cet arbre est un tout petit sous ensemble de l'étendue réelle de l'espace des noms de domaines.





Dans cet exemple, le domaine racine a trois sous-domaines immédiats : MIL, EDU, et ARPA. Le domaine LCS.MIT.EDU a un sous domaine immédiat appelé XX.LCS.MIT.EDU. Toutes les feuilles sont également des domaines.

3.5. Syntaxe préférentielle pour les noms

Les spécifications du DNS tentent d'être aussi génériques que possible pour ce qui concerne les règles de construction des noms de domaines. L'idée de base est que le nom existant pour un objet puisse être utilisé pour exprimer un nom de domaine avec le minimum de transformation. Cependant, au moment d'assigner un nom de domaine à un objet, l'utilisateur prudent choisira un nom qui d'une part respecte les règles de construction du système de domaines et d'autre part respecte les règles inhérentes à l'objet à nommer lui-même, que ces règles aient été publiées ou existent de façon implicite par le fait qu'elles sont utilisées par certains programmes.

Par exemple, pour nommer un domaine de courrier, l'utilisateur devra respecter les règles de ce mémo ainsi que celles établies par la RFC-822. Pour nommer un hôte, les anciennes règles instaurées pour les fichiers HOSTS.TXT d'alors devront être suivies. Ceci permet d'éviter des problèmes lorsque d'anciens programmes sont transformés pour prendre en compte les noms de domaine.

La syntaxe suivante diminuera notablement le risque de problèmes avec toute application utilisant les noms de doamine (ex., mail, TELNET).

```
<domaine> ::= <sous-domaine> | " "
<sous-domaine> ::= <identifiant> | <sous-domaine> "." <identifiant>
<identifiant> ::= <lettre> [ [ <ch-ldh> ] <let-dig> ]
<ch-ldh> ::= <let-dig-hyp> | <let-dig-hyp> <ch-ldh>
<let-dig-hyp> ::= <let-dig> | "-"
<let-dig> ::= <lettre> | <digit>

<lettre> ::= un des 52 caractères alphabetiques de "A" à "Z" (majuscules) ou de
"a" à "z" (minuscules)

<digit> ::= un des dix digits de "0" à "9"
```

Notez que, bien que tant les majuscules que les minuscules soient autorisées dans des noms de domaines, aucune importance n'est accordée à la casse. C'est-à-dire, deux noms lexicographiquement identiques, mais écrit dans une casse différente seront considérés comme identiques.

Les identifiants doivent suivre les règles définies pour les noms d'hôtes ARPANET. Ils doivent commencer par une lettre, terminer par une lettre ou un digit, et n'avoir à l'intérieur que des lettres, des digits, et éventuellement le caractère Hyphénation (-). On notera de plus quelques restrictions à propos de la longueur. Un identifiant doit avoir au plus 63 caractères.

Par exemple, les chaînes suivantes identifient des hôtes d'Internet :

```
A.ISI.EDU  XX.LCS.MIT.EDU  SRI-NIC.ARPA
```

3.6. Enregistrements

Un nom de domaine identifie un noeud. A chaque noeud est attribué un ensemble d'informations sur des ressource, lequel peut être vide. L'ensemble d'informations de ressources associé à un nom particulier est composé de quatre enregistrements de ressources séparés (RR). L'ordre des RR dans un ensemble n'est pas significatif, et ne doit pas nécessairement être préservé par les serveurs de noms, les résolveurs, ou tout autre partie du DNS.

Lorsque nous parlons d'un RR spécifique, nous supposons qu'il contient les éléments suivants :

owner le nom de domaine où le RR est trouvé.

Type une valeur encodée sur16 bits spécifiant le type de ressource décrit par cet enregistrement. Les types se réfèrent à une définition abstraite des ressources.

Ce mémo définit les types suivants :

A	une adresse d'hôte
CNAME	le nom canonique d'un alias
HINFO	le CPU et le système d'exploitation (OS) s'un hôte
MX	un schéma d'échange de courrier pour ce domaine. Voir [RFC-974] pour plus de détails.
NS	le serveur de noms "autorisé" pour le domaine
PTR	un pointeur vers une autre partie de l'espace de noms de domaines
SOA	le début d'une sphère d'autorité

class une valeur encodée sur 16 bits identifiant une famille de protocoles ou une instance d'un protocole.

Ce mémo définit les classes suivantes :

IN	le système Internet
CH	le système Chaotique

TTL la durée de vie du RR. Cette valeur est représentée sous forme d'un entier sur 32 bits et est exprimée en secondes, et est principalement utilisée par les résolveurs lorsqu'ils mémorisent temporairement des RR. Le champ TTL définit combien de temps un RR peut être gardé localement avant de devoir être considéré comme obsolète.

RDATA le type et parfois les données dépendantes de la classe décrivant la ressource :

A	Pour la classe IN, une adresse IP sur 32 bits. Pour la classe CH, un nom de domaine suivi d'une adresse octale Chaotique sur 16 bits.
CNAME	un nom de domaine.
MX	une valeur de préférence sur 16 bits (la plus basse possible) suivie d'un nom d'hôte souhaîtant servir d'échangeur de courrier pour le domaine de l'*owner*.
NS	un nom d'hôte.
PTR	un nom de domaine.
SOA	plusieurs champs.

Le nom du propriétaire (owner) est souvent implicite, plutôt que formant une partie intégrante du RR. Par exemple, de nombreux serveurs de noms représentent l'espace de nom en interne sous forme d'arbre ou de tableaux associatifs, et pointent les RR à partir des noeuds. Le restant des données des RR, soit l'en-tête fixe (type, classe, TTL) valable pour tous les RR, et la partie variable (RDATA) adaptée au type de ressource décrite, étant habituellement stockée à l'extérieur de la représentation de la structure de l'espace.

La signification du champ TTL est la durée limite pendant laquelle un RR peut être conservé dans un cache local. Cette limite ne s'applique pas aux données "autorisées" stockées dans les zones ; celles-ci disposent aussi d'une temporisation, mais définie par la politique de rafraîchissement de la zone elle-même. La TTL est définie par l'administrateur pour toute la zone contenant cet enregistrement. Alors qu'une valeur faible de la TTL peut être utilisée pour diminuer la durée de cache, et qu'une valeur de zéro empêche tout stockage local, l'analyse réelle des performances d'Internet suggère que cette valeur soit de l'ordre de quelques jours pour un hôte type. Lorsque l'on

peut anticiper sur une modification, la TTL pourra être réduite juste avant d'effectuer la modification pour optimiser la consistance de l'information au moment du changement, puis être rétablie à sa valeur d'origine après un certain délai.

Les données dans la section RDATA d'un RR est stockée comme une combinaison de chaînes binaires et de noms de domaines. Les noms de domaines seront souvent utilisés à titre de "pointeurs" sur d'autres structures de données du DNS.

3.6.1. Expression des RR sous forme textuelle

Les RR sont représentés sous forme binaire dans les paquets du protocole DNS, et sont habituellement représentés sous une forme fortement compactée lorsque stockés par un serveur de noms ou un résolveur. Dans ce document, nous adopterons une notation similaire à celle utilisée dans les fichiers principaux de façon a exprimer le contenu des RR. Dans ce format, la plupart des RR peuvent être exprimés sur une seule ligne, bien qu'une extension sur plusieurs lignes soit possible par l'utilisation de parenthèses.

Au début de la ligne est mentionné le propriétaire du RR. Si une ligne commence par un esapce, alors on suppose que le propriétaire de l'enregistrement est le même que celui du RR précédent. Des lignes vides sont aussi souvent insérées pour augmenter la lisibilité.

Après le propriétaire, nous exprimerons la TTL, le type, et la classe du RR. Classe et type utilisent les mnémoniques définis ci-avant, la TTL étant exprimée sous forme d'entier apparaissant avant le champ type. De façon à éviter des ambiguïtés d'interprétation lors de l'analyse des lignes, les mnémoniques du type et de la classe sont disjoints, la TTL est un entier, et le mnémonique de type apparaît toujours en dernier. La classe IN et les valeurs de la TTL seront souvent omises dans les exemples par souci de clarté.

Les données de ressource ou section RDATA de l'enregistrement sont exprimées selon la connaissance que nous avons de la représentation classique de ces données.

Par exemple, nous exprimerons un RR transporté par un message sous la forme suivante :

```
ISI.EDU.         MX      10 VENERA.ISI.EDU.
                 MX      10 VAXA.ISI.EDU.
VENERA.ISI.EDU.  A       128.9.0.32
                 A       10.1.0.52
VAXA.ISI.EDU.    A       10.2.0.27
                 A       128.9.0.33
```

Le RR MX a une section RDATA consistant en un entier sur 16 bits suivi par un nom de domaine. L'adresse du RR utilise la représentation standard d'une adresse IP sur 32 bits.

Cet exemple montre donc six RR, répartis en groupes de deux RR pour chacun des trois noms de domaines.

De la même manière nous pourrions voir :

```
 XX.LCS.MIT.EDU. IN      A       10.0.0.44
                 CH      A       MIT.EDU. 2420
```

représentant deux adresses pour l'hôte XX.LCS.MIT.EDU, chacune d'une classe différente.

3.6.2. Alias et expression canonique des noms

Dans des systèmes existants, les hôtes et autres ressources peuvent avoir plusieurs noms différents pour les désigner. Par exemple, les noms C.ISI.EDU et USC-ISIC.ARPA pointent sur le même hôte. De la même manière, dans le cas de boîtes aux lettres, de nombreuses organisations font pointer sur la même boîte aux lettres de multiples noms; par exemple Mockapetris@C.ISI.EDU, Mockapetris@B.ISI.EDU, et PVM@ISI.EDU pointent toutes la même boîte aux lettres (bien que l mécanisme derrière tout ça soit légèrement plus compliqué que cela).

La plupart de ces systèmes manipulent une notion selon laquelle l'un de ces noms est dit primaire (ou canonique), et tous les autres sont des alias.

Le système de domaines dispose d'une telle fonctionnalité par l'utilisation du nom canonique (CNAME) RR. Un RR CNAME identifie son propriétaire comme étant un alias, et spécifie le nom canonique correspondant dans la section RDATA du RR. Si un RR CNAME est présent dans un noeud, aucune autre donnée ne peut y être inscrite; cette restriction garantit que les données relatives à un nom canonique et son alias seront toujours identiques. Cette règle assure de plus qu'un enregistrement CNAME peut petre utilisé sans nécessité de consulter un serveur "autorisé" pour obtenir d'autres types de RR.

Les RR CNAME provoquent des actions particulières dans un processus DNS. Lorsqu'un serveur de noms échoue lorsqu'il cherche un enregistrement particulier dans l'ensemble des informations associées au nom de domaine, il vérifie si la ressource n'est pas un enregistrement CNAME avec la bonne classe. Si c'est le cas, le serveur de noms répond à la requête en retournant l'enregistrement CNAME dans la réponse et relance une résolution sur le nom de domaine mentionné dans la section RDATA de l'enregistrement CNAME (donc sur le domaine canonique). La seule exception admise pour cette règle est lorsque la requête initiale demande déjà une information de type CNAME, auquel cas la redirection n'est pas effectuée.

Par exemple, supposez qu'un serveur de noms traite une requête sur le domaine USC- ISIC.ARPA, pour un type d'information A, et dispose des enregistrements suivants :

```
USC-ISIC.ARPA   IN      CNAME   C.ISI.EDU
C.ISI.EDU       IN      A       10.0.0.52
```

Les deux RR seraient retournés pour une requête sur le type A, tandis qu'une requête sur le type CNAME ou * se verrait répondre par l'enregistrement CNAME uniquement. Les noms de domaine dans les RR pointnat sur un autre nom de domaine devra toujours pointer sur un nom canonique et jamais sur un alias. Ceci évitera une multiplication inutile des étapes d'indirection. Par exemple, l'adresse à utiliser dans les RR pour pointer sur l'hôte ci-dessus serait :

```
52.0.0.10.IN-ADDR.ARPA  IN      PTR     C.ISI.EDU
```

plutôt qu'un pointeur sur USC-ISIC.ARPA. Bien sûr, selon le principe de robustesse, les logiciels de domaines ne devraient pas échouer face à des chaînes ou des boucles de CNAME; Les enchaînement de CNAME devront être suivis et les bouclages de CNAME signalées par une erreur.

3.7. Requêtes

Les requêtes sont des messages qui sont envoyées à un serveur de noms pour obtenir une réponse. Dans Internet, les requêtes sont transportées par des datagrammes UDP ou sur via des connexions TCP. La réponse du serveur de nom peut soit répondre à la question posée par la requête, rediriger le requérant vers un autre serveur de noms, ou signaler une condition d'erreur.

En général, l'utilisateur ne pourra émettre lui-même directement des requêtes, mais passera par un résolveur qui émettra une ou plusieurs requêtes vers des serveurs de noms et sera en mesure de traiter les conditions d'erreur ou les redirections qui pourraient en résulter. Bien sûr, les questions possibles qui peuvent être posées dans une requête doivent correspondre au type de service pour lequel le résolveur est conçu.

Les requêtes et réponses DNS sont transportées dans un message de format standardisé. Le format de message définit une en-tête contenant un certain nombre de champs fixes toujours présents, et quatre sections pour transporter les paramètres et les RR.

Le champ d'en-tête le plus important est un champ de 4 bits appelé "code opération", permettant de distinguer les différentes requêtes. Parmi les 16 valeurs possibles, une (requête standard) fait partie du

protocole officiel, deux autres (requête inverse et requête d'état) sont optionnelles, une autre (fin de traitement) est obsolète, les autres restant non assignées à l'heure actuelle.

Les quatre sections sont :

Question	contient le nom de la requête et ses autres paramètres.
Réponse	contient les RR qui répondent directement à la requête.
Autorisation	contient les RR décrivant d'autres serveurs "autorisés". Peut aussi contenir un RR SOA contenant les données d'autorisation dans la section réponse.
Additionnel	contient les RR qui peuvent aider à exploiter les RR contenus dans les autres sections.

Notez que le contenu, (mais pas le format) de ces sections peut varier suivant le code opération de l'en-tête.

3.7.1. Requête Standard

Une requête standard contient un nom de domaine cible (QNAME), un type de requête (QTYPE), et une classe de requête (QCLASS) et requiert les RR correspondants. Ce type de requête représente la très grande majorité des requêtes qui peuvent arriver à un DNS et nous les appellerons "requête" sans autre mention qualificative. Les requêtes plus particulières seront explicitement qualifiées. Les champs QTYPE et QCLASS font chacun 16 bits de long, et sont un surensemble des types et classes définies. Le champ QTYPE peut contenir :

<tout type>	demande la correspondance sur ce type. (ex., A, PTR).
AXFR	QTYPE spécial pour transfert de zone.
MAILB	demande la correspondance pour toutes les RR de boîtes aux lettres RRs (ex. MB et MG).
*	demande la correspondance sur tous les types de RR.

Le champ QCLASS peut contenir :

<toute class>	demande la correspondance sur cette classe uniquement (e., IN, CH).
*	demande la correspondance sur toutes les classes de RR.

A partir du nom de domaine cible, du QTYPE, et QCLASS, le serveur de noms recherche les RR correspondants. En plus des enregistrements trouvés, le serveur de noms de domaines peut renvoyer les RR pointant vers un serveur de noms détenant l'information demandée ainsi que tout RR qui peut aider à l'interprétation des RR renvoyés. Par exemple, un serveur de noms de domaines ne disposant pas de l'information demandée peut connaître un autre serveur de nom qui la détient ; un serveur de noms qui renvoie des RR pertinents peut aussi y adjoindre d'autres RR qui relient le nom de domaine vers une adresse.

Par exemple, un agent de courrier tentant d'envoyer un message dans la boîte aux lettres Mockapetris@ISI.EDU peut demander à son résolveur des informations sur le serveur de courrier de ISI.EDU, constituant pour cela la requête QNAME=ISI.EDU, QTYPE=MX, QCLASS=IN. La section réponse renvoyée serait dans ce cas :

```
ISI.EDU.        MX      10 VENERA.ISI.EDU.
                MX      10 VAXA.ISI.EDU.
```

et la section additionnelle :

```
VAXA.ISI.EDU.   A       10.2.0.27
                A       128.9.0.33
VENERA.ISI.EDU. A       10.1.0.52
                A       128.9.0.32
```

Comme le serveur suppose que si le requérant désire des informations sur un échangeur de courrier, il désirera obtenir les adresses de ces échangeurs peu après.

Notez que l'écriture QCLASS=* nécessite une interprétation particulière notamment concernant les "autorisations". Dans la mesure où un serveur de nom ne connaît pas nécessairement toutes les classes disponivles dans le système de domaines, il ne peut jamais savoir si il est "autorisé" pour toutes les classes. Par voie de conséquence, aucune réponse à une requête QCLASS=* peut se qualifier "d'autorisée".

3.7.2. Rétro-requêtes (Optionnel)

Les serveurs de noms peuvent également supporter des requêtes inverses ou "rétro-requêtes" transcodant une ressource particulière en un nom de domaine ou les noms de domaines disposant de cette ressource. par exemple, alors qu'une requête standard peut transcoder un nom de domaine en un RR SOA, la rétro-requête correspondante transcodera ce RR SOA vers le nom de domaine.

L'implémentation de ce service est optionnel dans un serveur de noms de domaines, bien que tous les serveurs de noms doivent au moins être capable d'identifier une requête inverse et renvoyer le message d'erreur "not-implemented". Le système de domaines ne peut pas garantir le résultat ou l'unicité de la réponse à une rétro-requête du fait de son organisation basée sur une hiérarchie de noms de domaines, et non sur une adresse d'hôte ou tout autre type de ressource. Les rétro-requêtes sont principalement utiles pour des fonctions de déboggage et la maintenance des bases de données.

Les réponses à rétro-requêtes ne devront pas renvoyer la TTL, et ne doit pas indiquer les cas où le RR identifié fait partie d'un ensemble corrélé (par exemple, une adresse d'un hôte disposant de plusieurs adresses). De ce fait, les RR renvoyés par des rétro-requêtes ne doivent jamais être stockés.

Les rétro-requêtes NE sont PAS une méthode acceptable pour transcoder des adresses d'hôtes en noms d'hôtes ; utiliser plutôt le domaine IN-ADDR.ARPA.

Une discussion détaillée sur les rétro-requêtes peut être consultée dans la [RFC-1035].

3.8. Requêtes d'état (Expérimental)

A définir.

3.9. Requêtes de Fin de Traitement (Obsolète)

Le service optionnel de mention de fin de traitement défini dans les RFC 882 et 883 a été supprimé. Un service de ce type complètement rénové sera peut être rétabli dans le futur, ou bien les codes opération réservés par cet ancienne fonctionnalité seront réaffectés.

4. SERVEURS DE NOMS DE DOMAINE

4.1. Introduction

Les serveurs de noms sont dépositaires de l'information qui constitue la base de données des domaines. La base de données est divisée en sections appelées zones, lesquelles sont distribuées sur l'ensemble des serveurs de noms. Comme les serveurs de noms peuvent implémenter des fonctions optionnelles et des sources de données différentes, la tâche essentielle d'un serveur de nom reste de répondre à des requêtes sur ses données propres. Par conception, les serveurs de noms peuvent répondre à ces requêtes d'une manière simple ; la réponse peut toujours être générée à partir des seules données locales, et contiendra soit la réponse à la question posée, soit une référence à un autre serveur de noms "plus suceptible" d'avoir l'information demandée.

Une zone donnée pourra être accessible à partir de plusieurs serveurs de noms afin d'assurer son accessibilité même en cas de défaillance du serveur ou des liaisons. Par décision "administrative", nous imposons qu'une zone soit accessible au moins par deux serveurs, et souhaitons que la redondance soit en réalité plus importante que cela.

Un serveur de noms donné support typiquement une ou plusieurs zones, ceci ne le qualifiant d'autorisé que pour une petite partie de l'arbre des domaines. Il pourra de plus détenir une certaine quantité de données "non-autorisées" dans son cache, spécifiant certaines autres parties de l'arbre. Le serveur de nom renseigne sa réponse de telle sorte que le requérant sache si les informations proviennent de la partie "autorisée" ou nom du serveur de noms.

4.2. Comment la base de données est divisée en zones

La base de données de domaines est divisée selon deux méthodes : en classes, et par "découpage" de l'espace des noms de domaines.

La partition en classes est assez simple. La base de données est organisée, déléguée, et maintenue séparément pour chacune des classes. Comme par convention, l'espace de noms est identique quel que soit la classe, la séparation par classe peut conduire à voir l'espace de domaines comme un tableau d'arbres de noms parallèles. Notez que les données attachées aux noeuds des arbres seront différentes dans chaque arbre. Les raisons les plus courantes poussant à créer une nouvelle classe est soit la nécessité de gérer un nouveau format de données pour des types existants, soit la nécessité de gérer différemment une partie des informations.

Dans une classe, des "coupes" dans l'espace de noms peuvent être faites entre deux noeuds adjacents quelconques. Un fois toutes les coupes définies, chaque groupe de noeuds interconnectés devient une zone indépendante. La zone est alors définie comme étant la "sphère d'autorisation" pour tout nom à l'intérieur de la zone. Notez que les "coupes" dans l'espace de noms peuvent être à des endroits différents de l'arbre suivant la classe, les serveurs de noms associés peuvent être différents, etc.

Ces règles signifient que chaque zone doit avoir au moins un noeud, et donc un nom de domaine, pour lequel il est "autorisé", et que tous les noeuds d'une zone particulière sont connectés. Du fait de la structure d'arbre, chaque zone contient un noeud "de plus haut niveau" qui est plus proche de la racine que tous les autres noeuds de cette zone. Le nom de ce noeud est souvent utilisé pour identifier la zone elle-même.

Selon ce concept, il est possible, bien que pas forcément utile, de découper l'espace de noms de telle façon que chaque nom de domaine se retrouve dans une zone séparée ou au contraire que tous les noeuds se retrouvent dans une zone unique. En fait, la base de données est découpée selon la volonté d'une organisation particulière d'accepter de gérer le sous-arbre inférieur au point de coupure. Une fois que cette organisation contrôle sa propre zone, elle pourra modifier les données dans cette zone de façon unilatérale, créer des nouveaux sous-arbres à l'intérieur de la zone, supprimer des noeuds existants, ou déléguer la gestion de sous-zone à d'autres organisations plus locales.

Si l'organisation est elle même structurée, elle souhaitera certainement créer des sous partitions dont elle déléguera à son tour la gestion. Dans certains cas, de telles divisions ne sont faites que pour rendre plus maniable la maintenance de la base de données.

4.2.1. Considérations techniques

Les données décrivant une zone se divisent en quatre parties majeures :

- Les données autorisées pour tous les noeuds dans la zone.
- Des données définissant le noeud de plus haut niveau de la zone (peuvent être considérées comme faisant part des données autorisées).
- Des données décrivant les sous zones déléguées, c'est-à-dire, les points de coupure dans les étages inférieurs de la zone.
- Les données permettant l'accès aux serveurs de noms traitant les sous-zones déléguées (appelées souvent "glue data").

Toutes ces données sont exprimées sous forme de RR, et donc une zone peut être entièrement décrite comme un ensemble de RR. Des zones entières peuvent être transférées de serveur à serveur

par le transfert des RR, soit par une suite de messages, ou en transférant le fichier principal, qui en est une représentation textuelle, par FTP.
Les données autorisées pour une zone sont en fait tous les RR attachés à tous les noeuds depuis la tête de la zone jusqu'aux feuilles les plus basses, ou le noeud immédiatement au-dessus d'un point de coupe d'une sous-zone.

Bien que faisant logiquement partie des données autorisées, les RR qui décrivent la tête de la zone sont fondamentaux pour la gestion des zones. Ces RR sont de deux types : des RR de serveurs de noms qui listent, à raison de un par RR, tous les serveurs autorisés pour cette zone, et un RR SOA unique qui donne les paramètres de gestion de cette zone.

Les RR qui décrivent les coupures vers le bas de la zone sont des RR NS qui pointent sur les serveurs de noms auxquels ont été déléguées les sous-zones. Dans la mesure où les coupures se font entre deux noeuds, ces RR NE font PAS partie des données autorisées de la zone, et devront donner exactement les mêmes informations que le RR donnant les informations sur le noeud de tête de la sous-zone, dans le serveur délégué. Comme les serveurs de noms sont toujours associés par rapport aux limites de zones, les RR NS ne peuvent être trouvés que dans des noeuds représentant la tête d'une zone. Dans les données qui caractérisent la zone, les RR NS seront trouvés dans le noeud qui caractérise la tête de la zone courante (constituant un enregistrement autorisé*) et au niveau des noeuds pointant sur la tête d'une sous-zone (lequel noeud n'est pas considéré comme une information "autorisée"), mais jamais dans un noeud intermédiaire.

L'un des objectifs de cette structure en zones est que toute zone puisse disposer localement de toutes les données nécessaires pour communiquer avec les serveurs de noms de chacune de ses sous-zones. En d'autres termes, que les zones mères disposent de toute l'information requise pour accéder aux serveurs des zones filles. Les RR NS qui nomment ces serveurs pour les sous-zones ne suffisent pas toujours à cet tâche dans la mesure où, s'ils définissent le nom du serveur, n'en donnent pas l'adresse. En particulier, si le nom du serveur de noms est lui-même dans la sous-zone, nous pourrions être confronté à la situation embarrassante où le RR NS nous dit que pour connaître l'adresse du serveur de noms de la sous-zone, nous devons contacter un serveur portant l'adresse que nous souhaitons justement apprendre. Pour contourner ce problème, une zone contient des RR qualifiés de "glue" qui ne font pas partie de l'ensemble des données dites "autorisées", et représentent des adresses de serveurs sous forme de RR. Ces RR ne sont nécessaires que si le nom du serveur de nom se situe justement dans la portion d'espace "en dessous" de la coupure, et ne sont utilisés que comme constituant partiel d'une réponse référentielle.

4.2.2. Considérations en termes d'administration

Lorsque certaines organisations veulent récupérer le contrôle de leur propre domaine, la première étape est d'identifier la bonne zone mère, et obtenir des "légataires" de la zone mère un accord de délégation de gestion. Comme aucune contrainte technique particulière ne vient déterminer un point précis de l'arbre où la coupure peut être effectuée, certains regroupements "arbitraires" ont été exposés dans la [RFC-1032] concernant l'organisation des niveaux supérieurs, les zones médianes restant libres de créer leurs propres règles. Par exemple, une université pourrait choisir de n'utiliser qu'une seule zone, tandis qu'une autre pourrait choisir d'organiser son parc en sous-zones, dédiées chacune à un département ou une école. La [RFC-1033] liste les logiciels de DNS disponibles et expose les procédures administratives.

Une fois le nom de la sous-zone choisi, les futurs "légataires" devront prouver leur capacité à supporter la redondance des serveurs de domaines. Notez qu'il n'y a pas d'obligation à ce que le serveur pour une zone soit implanté sur une machine disposant d'un nom dans cette zone. Dans de nombreux cas, une zone sera bien plus largement accessible à Internet si les serveurs qui la gèrent sont dispersés, plutôt que concentrés sur le site physique contrôlé par le "légataire" de la la zone. Par exemple, l'un des serveurs de noms pour le sous-domaine United Kingdom, ou UK, est situé aux Etats-Unis. Ceci permet aux hôtes américains d'obtenir les informations sur les serveurs anglais sans être contraint par les limites de bande passante transatlantique.

Dans la dernière étape, les RR NS et les RR "glue" pointant sur le serveur délégué, et nécessaires pour rendre opérationnel le transfert de gestion, devront être ajoutés dans la zone mère. Les administrateurs de chaque zones devront s'assurer que les RR NS et "glue" situés de chaque côté de la coupure sont identiques et le restent.

4.3. Serveur de noms - spécifications internes

4.3.1. Requêtes et réponses

La principale activité d'un serveur de noms est de répondre aux requêtes standard. La requête et sa réponse sont toutes deux véhiculées par un message de format standardisé décrit dans la [RFC-1035]. La requête contient des champs QTYPE, QCLASS, et QNAME, qui décrivent le(s) type(s) et la ou les classes de l'information souhaitée, et quel nom de domaine cette information concerne.

La manière dont le serveur de noms répond peut être différente selon que le serveur utilise le mode récursif ou non :

- Le mode le plus simple de fonctionnement du point de vue serveur est le mode non-récursif, dans la mesure où le serveur peut répondre à la requête uniquement sur la base de ses informations locales : la réponse contient une erreur, la réponse demandée, ou donne la référence d'un autre serveur plus "susceptible" de disposer de l'information demandée. Tous les serveurs de noms se doivent d'implémenter le mode non-récursif.
- Le mode le plus simple de fonctionnement du point de vue du client est le mode récursif, dans la mesure où dans ce mode, le serveur prend à son tour le rôle de résolveur et ne peut retourner qu'un message d'erreur ou une réponse valide, mais jamais de référence. Ce service est optionnel dans un serveur de noms, ce dernier pouvant de plus choisir de restreindre les possibilités de clients gérant le mode récursif.
- Le service récursif est utile dans plusieurs situations :
- une implémentation simplifiée d'un résolveur qui ne sait exploiter d'autres réponses qu'une réponse directe à la question.
- une requête qui doit passer à travers d'autres protocoles ou autres "frontières" et doit pouvoir être envoyée à un serveur jouant le rôle d'intermédiaire.
- un réseau dans lequel intervient une politique de cache commun plutôt qu'un cache individuel par client.

Le service non-récursif est approprié si le résolveur est capable de façon autonome de poursuivre sa recherche et est capable d'exploiter l'information supplémentaire qui lui est envoyée pour l'aider à résoudre son problème.

L'utilisation du mode récursif est limité aux cas qui résultent d'un accord négocié entr ele client et le serveur. Cet accord est négocié par l'utilisation de deux bits particuliers des messages de requête et de réponse :

- Le bit RA (Récursion admissible), est marqué ou non par le serveur dans toutes les réponses. Ce bit est marqué si le serveur accepte à priori de fournir le service récursif au client, que ce dernier l'ait demandé ou non. Autrement dit, le bit RA signale la disponibilité du service plutôt que son utilisation.
- Les requêtes disposent d'un bit RD (pour "récursion désirée"). Ce bit indique que le requérant désire utiliser le service récursif pour cette requête. Les clients peuvent demander le service récursif à n'importe quel serveur de noms, bien que ce service ne puisse leur être fourni que par les serveurs qui auront déjà marqué leur bit RA, ou des serveurs qui auront donné leur accord pour ce service par une négociation propriétaire ou tout autre moyen hors du champ du protocole DNS.

Le mode récursif est mis en œuvre lorsqu'une requête arrive avec un bit RD marqué sur un serveur annonçant disposer de ce service ; le client peut vérifier si le mode récursif a été utilisé en constatant que les deux bits RA et RD ont été marqués dans la réponse. Notez que le serveur de noms ne doit pas utiliser le service récursif s'il n'a pas été explicitement demandé par un bit RD, car cela interfère avec la maintenance des serveurs de noms et de leurs bases de données. Lorsque le service récursif est demandé et est disponible, la réponse récursive à une requête doit être l'une des suivantes :

- La réponse à la requête, éventuellement préfacée par un ou plusieurs RR CNAME qui indiquent les alias trouvés pendant la recherche de la réponse.
- Une erreur de nom indiquant que le nom demandé n'existe pas. Celle-ci peut inclure des RR CNAME qui indiquent que la requête originale pointait l'alias d'un nom qui n'existe pas.
- Une indication d'erreur temporaire.

Si le service récursif n'est pas requis, ou n'est pas disponible, la réponse non-récursive devra être l'une des suivantes :

- Une réponse d'erreur "autorisée" indiquant que le nom n'existe pas.
- Une indication temporaire d'erreur.
- Une combinaison :
  - des RR qui répondent à la question, avec indication si les données sont extraites d'une zone ou d'un cache.
  - d'une référence à un serveur de noms qui gère une zone plus "proche" du nom demandé que le serveur qui a été contacté.
- Les RR que le serveur de nom pense être utile au requérant pour continuer sa recherche.

4.3.2. Algorithme

L'algorithme actuel utilisé par le serveur de noms dépends de l'OS local et des structures de données physiques utilisées pour stocker les RR. L'algorithme suivant suppose que les RR sont organisés en structures d'arbre multiples, un pour chaque zone, et une particulière pour le cache :

1. Marquer ou effacer la valeur de "récursion admissible" (RA) dans la réponse suivant que le serveur de noms souhaite proposer le service récursif ou non. Si le service récursif est disponible et requis par le marquage du bit RD dans la requête, aller au pas 5, autrement continuez au pas 2.

2. Chercher, dans les zones disponibles, celle qui est l'ancêtre le plus proche de QNAME. Si une telle zone existe, aller au pas 3, sinon sautez au pas 4.

3. Commencer la recherche dans la zone, identifiant par identifiant. Le processus de recherche peut s'achever de plusieurs manières :

a. Si le nom QNAME est trouvé entièrement, le noeud a été trouvé.

Si l'information présente dans le noeud est un CNAME, et QTYPE ne correspond pas à CNAME, copier le RR CNAME dans la section "réponse" du message retourné, changer QNAME en le nom canonique dans le RR CNAME, et retournez au pas 1.

Autrement, copier tous les RR qui correspondent au QTYPE dans la section "réponse" et sauter au pas 6.

b. Si la recherche nous amène hors des données "autorisées", il s'agit d'une référence. Ceci arrive lorsque nous rencontrons un noeud contenant des RR NS indiquant que nous arrivons à un point de coupe vers une sous-zone.

Copier les RR NS de la sous-zone dans la section "autorisée" de la réponse. Mettre toute adresse disponibles pouvant aider à atteindre la sous-zone dans la section additionnelle, en utilisant des RR "glue" si ces adresses ne peuvent être extraites que d'une partie non-autorisée ou du cache. Sauter au pas 4.

c. Si pour l'un des identifiants du nom, aucune correspondance n'est possible (c-à-d., l'identifiant correspondant n'existe pas), voir si un identifiant "*" existe.

Si l'identifiant "*" n'existe pas, vérifier si le nom que nous cherchons est le QNAME original de la requête et non un nom donné par une ou plusieurs redirections dans des CNAME. Si le nom est l'original, préparer une réponse d'erreur "autorisée" et sortir. Sinon on sort de l'algorithme sans autre action.

Si l'identifiant "*" existe, chercher les RR de ce noeud correspondant au QTYPE de la requête. S'il en existe, les copier dans la section "réponse", mais en requalifiant le propriétaire (owner) des RR comme étant QNAME, et non celui du RR contenant l'identifiant "*". Sauter en 6.

4. Commencer la recherche dans le cache. Si QNAME y est trouvé, copier dans la section "réponse" tous les RR qui y sont rattachés et dont le QTYPE est conforme. S'il n'existe pas de délégation sur ces données pour ce qui concerne "l'autorisation", chercher dans le cache la meilleure information d'autorisation, et la placer dans la section "autorisation". Aller en 6.

5. Utiliser le résolveur local ou une copie de son algorithme (voir la section résolveur de ce mémo) pour répondre à la requête. Enregistrer les résultats, y compris tout CNAME intermédiaires, dans la section réponse.

6. Sur la base des données locales uniquement, tenter d'ajouter tous les autres RR qui pourraient être utiles dans la section additionnelle de la réponse et sortir.

4.3.3. Métaenregistrements

Dans l'algorithme précédant, un traitement spécial a été fait sur les RR dont le nom de propriétaire (owner) commence par l'identifiant "*". De tels RR sont appelés "métaenregistrements" (NdT : nous avons choisi ce terme au même titre que l'on utilise sous UNIX des métacaractères dans les expressions régulières pour remplacer une chaîne de caractères quelconque). Les métaenregistrements peuvent être compris comme des instructions servant à synthétiser des RR. Lorsque les conditions appropriées sont remplies, le serveur de nom crée des RR dont le nom de propriétaire est celui présent dans la requête et le contenu récupéré dans le métaenregistrement.

Cette faculté est le plus souvent utilisée pour créer une zone servant à rediriger des mail depuis Internet vers d'autres systèmes de messagerie. L'idée générale est que tout nom dans cette zone présenté au serveur dans une requête doit être supposé exister, doté de certaines propriétés, sauf si une évidence explicite conduit à l'affirmation du contraire. Notez que l'utilisation du terme zone dans ce cas, plutôt que domaine, est intentionnel ; un tel usage "par défaut" ne se propage pas au delà d'une limite de zone, bien qu'une sous-zone puisse elle aussi présenter ce fonctionnement en mettant en place un usage par défaut similaire.

Le contenu des métaenregistrements suit les règles et formats généraux des RR. Un métaenregistrement dans une zone a un propriétaire (owner) contrôlant pour quel nom requis l'enregistrement sera choisi. Le format pour le propriétaire du métaenregistrement est de la forme "*.<unDomaine>", où <unDomaine> représente n'importe quel nom de domaine. <unDomaine> ne doit pas contenir d'autre identifiant "*", et doit pointer dans les données autorisées de la zone. La globalisation obtenue par l'usage des métaenregistrements s'applique potentiellement aux descendants de <unDomaine>, mais pas à <unDomaine> lui-même. Une autre aspect de ceci est que l'identifiant "*" représente toujours un identifiant complet ou une suite d'identifiants (domaines relatifs), et jamais une partie d'identifiant.

Les métaenregistrement ne peuvent s'appliquer :

- Lorsque la requête pointe sur une autre zone. C'est-à-dire que le principe de délégation de gestion désactive l'usage des défauts.
- Lorsque le nom requis ou un nom entre l'identifiant "*" et le nom demandé existe. Par exemple, si un métaenregistrement était spécifié avec un propriétaire "*.X", et la zone contenait en outre des RR pour le

domaine B.X, les défauts s'appliqueraient à une requête sur Z.X (à supposer qu'aucune information explicite n'existe pour Z.X dans cette zone), mais pas à B.X, A.B.X, ni X lui-même.

Un identifiant "*" apparaissant dans une requête n'a pas d'effet particulier, mais peut être utilisé pour tester les défauts d'une zone autorisée ; une telle requête est la seule manière d'obtenir des réponses contenant des RR portant un nom de propriétaire incluant un "*". Le résultat de telles requêtes ne devra pas être conservé en cache.

Notez que le contenu des métaenregistrements n'est pas modifié lorsque ceux-ci sont lus pour synthétiser des RR.

Pour illustrer l'usage des métaenregistrements, supposez qu'une grande société disposant d'un réseau étendu, non basé sur TCP/IP, désire créer une passerelle de messagerie. Si la compagnie s'appelle X.COM, et la passerelle vers TCP/IP, A.X.COM, les RR suivants devraient être rentrés dans la zone COM :

```
X.COM           MX      10      A.X.COM

*.X.COM         MX      10      A.X.COM

A.X.COM         A       1.2.3.4
A.X.COM         MX      10      A.X.COM

*.A.X.COM       MX      10      A.X.COM
```

Avec une telle programmation, toute requête de type MX pour tout domaine terminant par X.COM retournera un RR MX pointant sur A.X.COM. Pour obtenir cet effet, deux métaenregistrements sont nécessaires. En effet, l'effet du métaenregistrement *.X.COM est inhibé pour tous les sous-domaines de A.X.COM du fait de l'existence d'un enregistrement explicite pour A.X.COM. Notez de plus que les enregistrements MX pour X.COM et A.X.COM eux-mêmes sont requis, et qu'aucun RR de la zone ci-dessus ne doit apparaître avec un propriétaire XX.COM.

4.3.4. Mise en cache de réponses négatives (Optionnel)

Le DNS définit un service optionnel qui permet aux serveurs de nom de distribuer, et aux résolveurs de stocker en cache, les réponses négatives dotées d'une durée de vie. Par exemple, un serveur de noms peut distribuer une durée de vie associée à une indication d'erreur de nom de domaine. Un résolveur recevant ce message à la possibilité de déduire que le nom demandé n'existe pas pendant ladite durée de vie sans être obligé de consulter des données autorisées. De même, un résolveur peut émettre une requête avec un QTYPE qui accepte plusieurs types à la fois, et conserver dans le cache l'information concernant l'absence de certains types.

Cette fonctionnalité peut être particulièrement importante dans un système qui propose des raccourcis recherche en utilisant des listes car ces raccourcis, qui nécessitent généralement un suffixe en fin de liste, peuvent générer de multiples erreurs de noms lorsqu’elles sont utilisées.

La méthode qu’utilise un serveur de noms sera d’ajouter un RR SOA dans la section additionnelle d’une réponse lorsque celle-ci est "autorisée". Le SOA doit être celui fourni par la zone source des données autorisées incluses dans la section réponse, ou un message d'erreur de nom si applicable. Le champ MINIMUM de l'enregistrement SOA contrôle le temps pendant lequel la réponse négative peut être gardée dans le cache.

Notez que dans certaines circonstances, la section réponse peut spécifier plusieurs noms de propriétaires. Dans ce cas, le mécanisme de SOA ne devra être employé que pour les données correspondant au QNAME, ces données étant les seules autorisées dans cette section.

Les serveurs de noms et les résolveurs ne devront jamais tenter d'ajouter des SOA dans la section additionnelle d'une réponse non-autorisée, ni tenter d'exploiter des résultats autres que ceux déclarés comme "autorisés". Il y a à cela plusieurs raisons, dont : l'information dans le cache ne suffit pas en général à déterminer des RR et leurs nom de zone associée, les RR SOA pourront être maintenues en

cache suite à une requête explicite dirigée vers des SOA, et les serveurs de noms n'inscrivent pas nécessairement les SOA dans la section autorisée (authority).

Ce fonctionnement reste optionnel, bien que l'on puisse prévoir qu'une version plus précise puisse rentrer dans le cadre du protocole officiel dans le futur. Les serveurs de noms ne sont pas tenus d'ajouter les RR SOA dans toutes leurs réponses autorisées, et les résolveurs ne sont pas plus tenus de conserver les réponses négatives en cache. Les deux comportements restent cependant recommandés. Tous les résolveurs et serveurs de noms récursifs sont tenu, au moins, de savoir ignorer des RR SOA lorsqu'ils se présentent dans une réponse.

Certaines expérimentations ont également été proposées pour utiliser cette fonctionnalité. L'idée qui les conduisait était que, si les données en cache sont connues comme venant d'une zone identifiée, et si une copie autorisée du SOA de la zone est obtenue, et enfin si le SERIAL de la zone n'a pas changé depuis que les données ont été mises dans le cache, alors la durée de vie des données du cache peut être réduite au MINIMUM de la zone considérée, si ce minimum est inférieur à la valeur cachée. Cet usage est décrit pour une recherche future, est n'est pas recommandé à présent.

4.3.5. Maintenance et transfert de zones

Une partie du travail d'un administrateur de zone est de maintenir l'état des zones dans chacun des serveurs de noms autorisés pour celles-ci. Lorsque des modifications, inévitables, sont reportées, elles doivent l'être sur tous les serveurs de noms associés à la zone. Bien que la distribution des données puisse se faire par FTP ou toute autre procédure de transfert conséquente, la méthode préférée sera la transfert de données de zone par le protocole DNS lui-même.

Le modèle général d'un transfert de zone ou rafraîchissement automatiques consiste à dire que l'in des serveurs de noms est le serveur maître (ou encore primaire) pour cette zone. Les modifications sont coordonnées depuis le serveur maître, en général en éditant le fichier primaire pour la zone. A la fin de l'édition, l'administrateur ordonne au serveur maître de charger la zone modifiée. Les autres serveurs secondaires (ou esclaves) pour cette zone doivent vérifier périodiquement (l'intervalle de périodicité doit être réglable) si des changements sont intervenus dans la définition primaire de la zone et demanderont des copies remises à jour une fois les modifications effectuées.

Pour détecter ces modifications, il suffit aux serveurs secondaires de vérifier le champ SERIAL de l'enregistrement SOA pour cette zone. En plus de tous les autres changements apportés dans la zone, le champ SERIAL du SOA de la zone est toujours modifié dès que le moindre changemement intervient. Cette modification peut se limiter à l'incrémentation d'un compteur, ou peut être basée sur l'écriture de la date et de l'heure de dernière écriture du fichier maître, etc. Le but est de donner un moyen de pouvoir savoir laquelle des deux copies du fichier de zone est le plus récent, par la simple comparaison d'un numéro de série. L'incrémentation et la comparaison des numéros de série s'appuie sur un espace séquentiel arithmétique, et il existe de ce fait une limite théorique sur la fréquence de rafraîchissement d'une zone. Celle-ci peut s'exprimer en disant que les anciennes copies doivent être totalement éliminées du système avant que le numéro de série ne "tourne" sur plus de la moitié de sa plage de 32 bits. En pratique, le seul point délicat est de vérifier que la comparaison fonctionne correctement lorsque l'on se trouve de part et d'autre du point où le compteur de série repart à 0.

La périodicité de la vérification de version par les serveurs secondaires est définie par des paramètres marqués dans le RR SOA attribué à la zone, qui définissent l'intervalle minimum entre deux vérifications. Ces paramètres sont appelés REFRESH, RETRY, et EXPIRE. Lorsqu'une zone est enregistrée dans un serveur secondaire, celui-ci attendra REFRESH secondes avant de vérifier sur le primaire si un nouveau numéro de série a été donné pour la zone. Si cette vérification ne peut être effectuée, des nouvelles tentatives seront faites toutes les RETRY secondes. La vérification est une requête simple visant le RR SOA de la zone sur le serveur principal. Si le numéro de série dans la copie hébergée par le secondaire est égal au numéro de série indiqué dans le RR retourné par la requête, alors aucune remise à jour n'est nécessaire, et la temporisation REFRESH doit être réactivée. Si le serveur secondaire n'arrive pas à faire la vérification après que l'intervalle EXPIRE se soit écoulé, il doit alors admettre que sa copie de la zone est obsolète et doit la supprimer.

Lorsque la vérification conclue en une différence de numéros de série, le serveur secondaire devra demander un transfert de données de zone via une requête AXFR pour cette zone. La requête AXFR peut retourner une erreur, telle que "refusé", mais donne suite en temps normal à la réception d'une séquence de messages de réponse. Les premier et dernier messages doivent contenir les données du noeud autorisé de plus haut niveau dans la zone. Les messages intermédiaires transportent tous les autres RR enregistrés pour la zone, les RR non autorisés y compris. Le flux de messages permettent au serveur secondaire de reconstituer une copie conforme de la zone. Comme la précision est essentielle, on utilisera TCP ou tout autre protocole fiabilisé pour les requêtes AXFR.

Tout serveur secondaire est tenu d'effectuer cette remise à jour auprès du principal, mais doit pouvoir optionnellement permettre à un autre serveur secondaire pour la même zone d'effectuer sa remise à jour à partir de cette copie secondaire. Cette stratégie permet d'améliorer globalement la pertinence des données, en proposant une solution en cas d'arrêt du serveur principal, ou de problèmes de réseau, ou tout simplement lorsqu'un serveur secondaire dispose d'un "meilleur" accès sur un autre serveur secondaire plutôt que sur le principal.

5. RESOLVEURS

5.1. Introduction

Les "résolveurs" sont des programmes qui interfacent les applications utilisateur aux serveurs de noms de domaines. Dans le cas le plus simple, un résolveur reçoit une requête provenant d'une application (ex., applications de courrier électronique, TELNET, FTP) sous la forme d'un appel d'une fonction ed bibliothèque, d'un appel système etc., et renvoie une information sous une forme compatible avec la représentation locale de données du système.

Le résolveur est situé sur la même machine que l'application recourant à ses services, mais devra par contre consulter des serveurs de noms de domaines sur d'autres hôtes. Comme un résolveur peut avoir besoin de contacter plusieurs serveurs de noms, ou à l'extrême inverse obtenir les informations directement à partir de son cache local, le temps de réponse d'un résolveur peut varier selon de grandes proportions, depuis quelques millisecondes à plusieurs secondes.

L'une des raisons les plus importantes qui justifient l'existence des résolveurs est d'éliminer le temps d'acheminement de l'information depuis le réseau, et de décharger simultanément les serveurs de noms, en répondant à partir des données cachées en local. Il en résulte qu'un cache partagé entre plusieurs processus, utilisateurs, machines, etc., sera incomparablement plus efficace qu'une cache non partagé.

5.2. Interface résolveur-client

5.2.1. Fonctions typiques

L'interface client du résolveur est largement dépendante des conventions de l'hôte local, mais on peut exprimer trois fonctions typiques qui rentrent dans le cadre de cette interface :

1. translation depuis les noms d'hôtes vers les adresses d'hôtes.

Cette fonction est souvent définie à l'image d'anciens mécanismes utilisant les fichiers HOSTS.TXT. A partir d'une certaine chaîne de caractères donnée, l'appelant désire obtenir une ou plusieurs adresses IP sur 32 bits. Dans le contexte du DNS, cette demande se traduit par une requête sur des RR de type A. Comme le DNS ne préserve pas l'ordre des RR, cette fonction peut choisir de trier et répondre par toutes les adresses obtenues, ou seulement la "meilleure" adresse si le client est plus enclin à n'accepter qu'une adresse unique en retour. Notez qu'il est recommander d'utiliser un retour multiple, mais un retour d'adresse unique peut dans certains cas être la seule solution pour émuler le fonctionnement d'anciens services basés sur des fichiers HOSTS.TXT.

2. Translation depuis une adresse vers un nom d'hôte

Cette focntionnalité suivra souvent dans sa formulation, la forme de fonctions plus anciennes. A partir d'une adresse IP 32 bits, le requérant désire une chaîne de caractères donnant le nom de la machine. Le sens des octets de l'adresse IP sera alors inversé. Les quatre octets ainsi transformés étant alors utilisé comme composante de nom, on les suffixera de la chaîne "IN-ADDR.ARPA". Une requête de type PTR est alors envoyée pour obtenir le RR donnant le nom primaire de l'hôte en question. Par exemple, une requête à propos de la machine d'adresse IP 1.2.3.4 recherchera des RR PTR pour le nom de domaine "4.3.2.1.IN-ADDR.ARPA".

3. Fonction de recherche générale

Cette fonction récupère des informations arbitraire à partir du DNS, et n'a pas de fonctionnalité antérieure correspondante. Le requérant fournit un QNAME, QTYPE, et une QCLASS, et désire obtenir tous les RR correspondants. Cette fonction utilisera de préférence la représentation DNS pour toutes les données des RR plutôt que celle de l'hôte local, et retournera le contenu de ces RR (ex., TTL) plutôt qu'une forme traitée selon les conventions locales.

Lorsque le résolveur exécute la fonction, on pourra s'attendre à l'une des réponses suivantes :

- Un ou plusieurs RR donnant les données demandées. Dans ce cas, le résolveur fournit la réponse dans le format approprié.
- Une erreur de nom (NE). Ceci arrive lorsque le nom présenté n'existe pas. Par exemple, un utilisateur peut très bien avoir mal orthographié un nom d'hôte.
- Une erreur "donnée non trouvée". Ceci intervient lorsque le nom demandé existe, mais pas les données du type spécifié pour le nom demandé. Par exemple, une requête pour une adresse d'hôte appliquée à un nom de boîte aux lettres retournerait cette erreur, car le nom existe bel et bien, mais pas le RR donnant une adresse d'hôte.

Il est important de noter que les fonctions de translation entre des noms d'hôte et des adresses peuvent retourner la combinaison d'une "erreur de nom" et d'une erreur "donnée non trouvée" dans un seul retour d'erreur, alors que la fonction de recherche générale ne le pourra pas. Une raison pour ceci est que les applications peuvent dans un premier temps demander un certain type d'information à propos d'un hôte puis dans une seconde requête un autre type d'information à propos du même hôte ; si les deux erreurs sont combinées, alors les requêtes inutiles ralentiraient l'application.

5.2.2. Alias

Lorsqu'il essaie de résoudre une requête particulière, le résolveur peut s'apercevoir que le nom demandé est un alias. Par exemple, le résolveur s'aperçoit que le résultat d'une translation est de ce type lorsque le RR renvoyé est un CNAME. Si possible, ce cas de détection d'un alias devra être signalée au client.

Dans la plupart des cas, un résolveur relancera la résolution sur le nouveau nom translaté donné dans le RR CNAME. Cependant, lors de l'appel à la fonction de recherche généralisée, le résolveur ne suivra pas la chaîne d'alias lorsque le RR CNAME est reçu en réponse à une requête sur ce même type. Ceci permet justement d'émettre des requêtes pour savoir si un alias existe. Autrement dit, si le type de requête demandé est CNAME, l'utilisateur s'intéresse effectivement au RR CNAME en tant que tel, et non l'objet final qu'il représente.

Un certain nombre de conditions spécifiques peuvent apparaître lorsque l'on traite avec des alias. On cherchera à éviter des longues chaînes de redirection par alias, dans la mesure ou cela porte atteinte à l'efficacité du système. Par contre, cette situation ne sera pas signalé comme un cas d'erreur. Inversément, des bouclages d'une chaîne de redirection par alias ainsi que des alias pointant sur des noms inexistants devront être signalés comme erreur, laquelle sera reportée au client.

5.2.3. Fautes temporaires

Concrètement, tous les résolveurs pourront se retrouver occasionnellement dans l'incapacité d'achever une résolution. Cette situation peut apparaître lorsqu'un résolveur perd le contact avec une partie du réseau à cause d'une panne réseau ou d'un routeur, ou, de façon moins courante dans le cas d'une indisponibilité ou défection simultanée de tous les serveurs correspondant à une zone.

Il est essentiel que cette classe d'erreur ne soit pas reportée au client au même titre qu'une erreur de nom ou que d'une erreur sur l'existence des données. Non seulement ce type d'erreur irrite notablement l'utilisateur humain, mais peut aussi causer de sérieux problème lorsqu'une messagerie utilise le DNS.

Il serait possible dans de tels cas d'erreurs de résoudre cette situation momentanée en bloquant la requête indéfiniment, dans l'attente de la levée de la condition de faute. Mais il s'agit d'une mauvaise idée, surtout lorsque le client est un processus serveur qui pourrait exploiter ce temps d'attente à des tâches bien plus utiles. La solution préconisée est de toujours admettre qu'une erreur "temporaire" puisse être un des résultats possibles de l'appel à une fonction de résolution, même si cela peut rendre l'émulation de certaines fonctions basées sur le principe du HOSTS.TXT plus ardue.

5.3. Résolveurs - spécifications internes

Toutes les implémentations de résolveur utilisent des algorithmes sensiblement différents, et utilisent la plupart de leur code pour traiter les erreurs de toutes natures plutôt que les occurrences "normales". Cette section expose une stratégie recommandée de base pour mener les opérations de résolution, les détails pouvant être trouvés dans la [RFC-1035].

5.3.1. Noyau de résolution minimal

Une possibilité pour l'implémentation de la fonction résolveur est de déporter la fonction de résolution au dehors de la machine locale vers un serveur de noms supportant le mode récursif. Ceci constitue une méthode simple pour offrir le service de de noms de domaines à un PC un peu faible en ressources, et pour centraliser le cache pour l'ensemble des machines et utilisateurs d'un réseau local ou d'une organisation.

Tout ce dont le noyau de résolution minimal à besoin est une liste d'adresses de serveurs de noms susceptible de mener la résolution récursive à notre place. Ce type de résolveur aura certainement besoin de ces informations stockées dans un fichier de configuration, car il n'aura probablement pas la sophistication suffisante pour localiser celle-ci dans la base de données des domaines. De plus, l'utilisateur aura besoin de vérifier que les serveur mentionnés peuvent effectivement fournir le service récursif ; un serveur de noms quel qu'il soit est tout à fait libre de refuser le service récursif à une partie ou la totalité de ses clients. L'utilisateur devra contacter son administrateur local pour savoir quels sont les serveurs qu'il pourra utiliser.

Ce type de service comport un certain nombre de limites et de défauts. Comme les requêtes peuvent être traitées dans un temps totalement arbitraire, le noyau aura souvent des difficultés à optimiser les temporisations de retransmission de paquets UDP signalant des paquets perdus ou des serveurs défaillants ; le serveur de noms pourra facilement être débordé par un noyau un peu trop zélé, surtout s'il interprète toutes les retransmissions comme des nouvelles requêtes. L'utilisation de TCP pourrait être une réponse à ce problème, mais l'implantation de TCP reviendrait à consommer pratiquement autant de ressources sur l'hôte qu'un résolveur complet.

5.3.2. Ressources

En plus de ses ressources propres, le résolveur peut aussi disposer d'accès partagés à des données de zones maintenues par un serveur de noms local. Ceci donne au résolveur l'avantage d'un accès plus rapide aux données, mais impose que le résolveur surveille qu'aucune donnée mise en cache ne vienne masquer une donnée pour cette zone. Dans cette partie, nous appellerons "information locale" la

réunion des informations contenues dans le cache et celles de la zone partagée, en sous-entendant que les données "autorisées" seront toujours utilisées de préférence à toute donnée du cache lorsque les deux sont présentes simultanément en local.

Les algorithmes de résolution qui suivent supposent que toutes les fonctions ont été exprimées sous forme d'une fonction de recherche généralisée, et utilisent les structures de données suivantes pour représenter la progression de la requête au niveau du résolveur :

- SNAME : le nom de domaine sur lequel porte la recherche.
- STYPE : le QTYPE de la requête.
- SCLASS : la QCLASS de la requête.
- SLIST : une structure qui décrit les serveurs de noms et la zone concernée par la requête en cours de traitement par le résolveur. Cette structure garde la trace des serveurs de noms que le résolveur estime actuellement être les plus susceptibles de détenir l'information désirée ; cette trace est remise à jour si l'information arrivant au résolveur est de nature à remettre en cause cette estimation. Cette structure contiendra un champ équivalent à un nom de zone, des champs pour représenter les serveurs de noms identifiés pour cette zone, les adresses de ces derniers, et des informations de type "historique" permettant d'estimer quel est le serveur suivant le mieux placé pour obtenir les informations recherchées. L'équivalent du nom de zone est un décompte du nombre d'identifiants (considérés à partir de la racine) que SNAME a en commun avec la zone objet de la recherche courante ; ceci donne une mesure de la "distance" qui sépare encore le résolveur de SNAME.
- SBELT : une structure "ceinture de sécurité" d'une forme identique à SLIST, initialisée à partir d'un fichier de configuration, et qui liste les serveurs qui devraient être utilisés lorsque le résolveur ne dispose plus ou pas assez d'informations locales pour guider la sélection de serveurs de noms. Le décompte de correspondance sera fixé à -1 pour indiquer qu'aucun des identifiants ne correspond.
- CACHE : une structure qui enregistre les résultats de précédentes réponses. Les résolveurs sont responsables de la purge des RR dont la durée de vie a expiré (dans le cache). La plupart des implémentations convertissent l'intervalle spécifié dans les RR reçus en une sorte de date "absolue" de péremption lorsque le RR est enregistré dans le cache. Plutôt que décrémenter les durées de vie individuellement, le résolveur n'a plus qu'à ignorer ou supprimer les RR "périmés" lorsqu'il les rencontre au cours d'une recherche, ou encore supprimer les RR dont la date de péremption est antérieure à la date courante lors d'opérations temporisées de récupération de mémoire (par destruction des vieux RR).

5.3.3. Algorithme

L'algorithme de haut niveau contient quatre étapes :

1. Vérifier si la réponse est disponible en local, si c'est le cas on la renvoie au client.

2. Déterminer les "meilleurs" serveurs à contacter.

3. Leur envoyer des requêtes jusqu'à ce qu'un d'eux réponde.

4. Analyser la réponse, soit :

a. Si la réponse répond à la question posée, ou stipule une erreur de nom, retouner l'information au client et mettre cette information en cache.

b. Si la réponse indique une redirection de "plus grande proximité", mettre en cache les références de ce nouveau serveur, et retourner à l'étape 2.

c. Si la réponse est un CNAME tout en étant pas la réponse attendue, cacher le CNAME, remplacer le SNAME par le nom canonique lu dans le RR CNAME et retourner à l'étape 1.

d. Si la réponse rend compte d'une défaillance du serveur ou autre cas bizarre, supprimer le serveur de la SLIST et retourner à l'étape 3.

L'étape 1 cherche la donnée désirée dans le cache. Si la donnée y est trouvée, on suppose qu'elle est suffisamment fiable pour une exploitation "normale". Certains résolveurs disposent d'une option configurable par l'utilisateur qui force à ignorer les données du cache et à consulter directement un serveur "autorisé". Ce fonctionnement n'est pas recommandé par défaut. Si le résolveur dispose d'un accès direct à la définition de zone d'un serveur de nom local, il cherchera à établir si les données demandées n'y sont pas présente sous une forme "autorisée". Si oui, on utilisera de préférence les données "autorisées", plutôt que les données trouvées dans le cache.

L'étape 2 recherche auprès de quel serveur de nom l'on va requérir la donnée recherchée. La stratégie usuelle est de chercher d'abord des RR de serveurs de noms disponibles en local, en partant du domaine SNAME, puis le père de SNAME, son grand-père, et ainsi de suite vers la racine. Ainsi, si SNAME était Mockapetris.ISI.EDU, cette étape rechercherait des RR NS pour le domaine Mockapetris.ISI.EDU, puis ISI.EDU, puis EDU, et enfin . (la racine). Ces RR NS donnent la liste des noms des hôtes de la zone coïncidant avec ou englobant SNAME. On copie ces noms dans SLIST, puis on détermine leur adresse à l'aide des données locales. Il peut se produire que certaines adresses ne puissent pas être identifiées en local. Le résolveur peut alors adopter plusieurs attitudes ; la plus sage est de "forker" le résolveur pour créer des processus "fils" dont la tâche sera de récupérer ces adresses, tandis que le père continuera la recherche sur les serveurs dont l'adresse est d'ores et déjà disponible. De façon évidente, les choix et options de design sont assez complexes et largement fonction des possibilités de l'hôte local. Les concepteurs de résolveurs devront considéré les priorités dans l'ordre suivant :

1. Brider la quantité de traitement (nombre de paquets émis, nombre de processus parallèles démarrés) de sorte qu'une requête ne puisse partir en boucle infinie ou initier une réaction en chaîne de création de processus et/ou d'émission de requêtes MEME DANS LE CAS DE DONNEES MAL CONFIGUREES.

2. Donner une réponse à chaque fois que possible.

3. Eviter la consommation superflue de ressources de transmission.

4. Donner les réponses aussi vite que possible.

Si la recherche de RR NS échoue, alors le résolveur initialisera la SLIST à partir de la structure SBELT. L'idée de base est que, lorsque le résolveur ne sait pas par où commencer pour rechercher une information, il utilisera une liste prédéfinie dans un fichier de configuration de serveurs supposés pouvoir être utiles. Bien qu'il puisse exister certaines situations particulières, on intégrera usuellement dans cette liste deux serveurs sur la racine et deux serveurs dans la zone où réside l'hôte. Le chiffre de deux (minimum) est donné pour assurer la redondance des données. Les serveurs de la racine pourront permettre un accès éventuel à d'autres parties (en fait, toutes) de l'espace de domaines. Les deux serveurs locaux permettront au résolveur de pouvoir être capable de continuer la résolution de noms locaux, même si le réseau local se retrouve isolé d'Internet suite à la défaillance d'une liaison ou d'un routeur.

En plus de fournir les adresses et noms des serveurs, la structure de données SLIST peut être triée de façon à donner une indication sur le "meilleur" serveur à utiliser, et s'assurer que toutes les adresses et noms de serveurs seront contactés chacun son tour. Le tri peut être une simple fonction de préséance des adresses locale sur les autres, ou peut être une fonction complexe de statistiques sur les événements passés, telle qu'une analyse des temps de réponse moyens et des taux de réussite.

L'étape 3 émet des requêtes jusqu'à ce qu'une réponse soit donnée. La stratégie consiste à utiliser toutes les adresses à tour de rôle, avec une temporisation donnée entre chaque émission. En pratique, il est important d'utiliser toutes les adresses d'un hôte "multiport", sachant qu'une politique de retransmission trop soutenue ralentit la réponse lorsque de multiples résolveurs sont impliqués dans une recherche sur le même serveur de nom (et même parfois pour un seul résolveur). La structure SLIST contient typiquement des données pour contrôler les temporisations et garder trace des précédentes transmissions.

L'étape 4 s'occupe de l'analyse des réponses. Le résolveur devra faire preuve d'une grande paranoïa dans l'analyse de ces réponses. Il devra en outre que la réponse correspond bien à la requête émise grâce à la lecture du champ ID dans la réponse. La réponse idéale proviendrait d'un serveur "autorisé" donnant les informations (en général l'adresse) attendues ou encore une erreur de nom. L'information est alors transmise au client et recopiée dans le cache en prévision d'une requête identique future si sa durée de vie (TTL) est supérieure à 0.

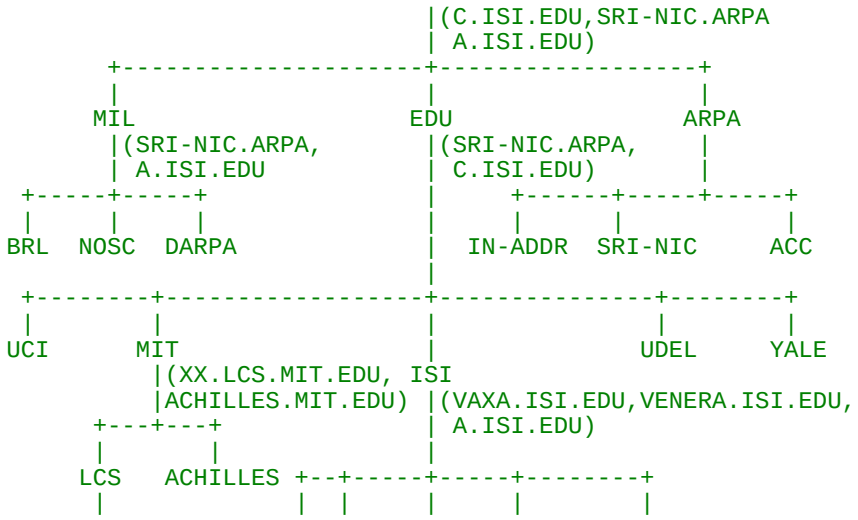
Si la réponse indique une redirection à effectuer, le résolveur vérifiera d'abord si le serveur mentionné est plus "proche" de la réponse que les serveurs dont on dispose dans SLIST. Ceci sera fait en comparant le décompte d'identifiants correspondants dans SLIST avec celui obtenu par calcul sur le SNAME dans le RR NS du message de redirection. Si tel n'est pas le cas, la réponse est considérée comme dénuée d'intérêt et sera ignorée. Si la redirection est utilisable, le RR NS de redirection et toute RR d'adresse obtenue pour les serveurs délégués devront être copiés dans le cache. Les serveurs de noms sont ajoutés à la SLIST, et la recherche recommence.

Si la réponse contient un CNAME, la recherche doit être réitérée avec le nom canonique CNAME sauf si la réponse donnée contient déjà l'information pour ce même nom canonique, où si la requête originale portait effectivement sur un CNAME lui-même.

Plus de détails et des astuces d'implémentation peuvent être trouvés dans la [RFC-1035].

6. UN SCENARIO

Dans notre espace de domaine "test", supposez que nous souhaitions obtenir un contrôle administratif autonome pour la racine, et les zones MIL, EDU, MIT.EDU et ISI.EDU. Nous définirions des serveurs de noms comme suit :



Dans cet exemple, le serveur de noms "autorisé" est noté entre parenthèses au point de l'arbre de domaines à patir duquel il prend le contrôle administratif de ce dernier.

Bien que les serveurs de noms pour la racine soient sur C.ISI.EDU, SRI-NIC.ARPA, et A.ISI.EDU. Le domaine MIL est servi par SRI-NIC.ARPA et A.ISI.EDU. Le domaine EDU est servi par SRI-

NIC.ARPA. et C.ISI.EDU. Notez que les serveurs peuvent supporter des zones contiguës ou disjointes. Dans notre scénario, C.ISI.EDU gère les zones contiguës de la racine et du domaine EDU. A.ISI.EDU gère les zones contiguës de la racine et du domaine MIL, mais gère une troisième zone non-contiguë aux deux autres, la zone ISI.EDU.

6.1. Serveur de nom C.ISI.EDU

C.ISI.EDU est un serveur de nom pour les domaines racine, MIL, et EDU pour la classe IN, et maintiendra des zones pour ces domaines. Les données de zone pour le domaine racine seraient :

```
.       IN      SOA     SRI-NIC.ARPA. HOSTMASTER.SRI-NIC.ARPA. (
                        870611          ;serial
                        1800            ;refresh every 30 min
                        300             ;retry every 5 min
                        604800          ;expire after a week
                        86400)          ;minimum of a day
                NS      A.ISI.EDU.
                NS      C.ISI.EDU.
                NS      SRI-NIC.ARPA.

MIL.    86400   NS      SRI-NIC.ARPA.
        86400   NS      A.ISI.EDU.

EDU.    86400   NS      SRI-NIC.ARPA.
        86400   NS      C.ISI.EDU.

SRI-NIC.ARPA.   A       26.0.0.73
                A       10.0.0.51
                MX      0 SRI-NIC.ARPA.
                HINFO   DEC-2060 TOPS20

ACC.ARPA.       A       26.6.0.65
                HINFO   PDP-11/70 UNIX
                MX      10 ACC.ARPA.

USC-ISIC.ARPA.  CNAME   C.ISI.EDU.

73.0.0.26.IN-ADDR.ARPA.  PTR    SRI-NIC.ARPA.
65.0.6.26.IN-ADDR.ARPA.  PTR    ACC.ARPA.
51.0.0.10.IN-ADDR.ARPA.  PTR    SRI-NIC.ARPA.
52.0.0.10.IN-ADDR.ARPA.  PTR    C.ISI.EDU.
103.0.3.26.IN-ADDR.ARPA. PTR    A.ISI.EDU.

A.ISI.EDU. 86400 A      26.3.0.103
C.ISI.EDU. 86400 A      10.0.0.52
```

Ces données sont représentées comme elles seraient écrites dans un fichier maître. La plupart des RR sont des entrées à une ligne ; la seule exception ci-dessus est le RR SOA pour cette zone, qui utilise une "(" pour commencer une définition multi-ligne et une ")" pour indiquer la fin du RR multi-ligne. Comme la classe de tous les RR d'une zone doit être identique, seul le premier RR d'une zone mentionnera la classe. Lorsqu'un serveur de noms charge une zone, il force la durée de vie de tout RR "autorisé" à la valeur indiquée dans le champ MINIMUM du SOA, ici 86400 secondes, soit un jour. Le RR NS indiquant la délégation administrative pour les domaines MIL et EDU, combinés aux RR de "glue" indiquant les adresses des hôtes de ces serveurs, ne font pas partie des données dites "autorisées" de la zone, et de ce fait mentionnent des durées de vies explicites.

Quatre RR sont rattachés au noeud racine : le SOA qui décrit la zone racine et les 3 RR NS qui listent les serveurs de noms pour la racine. Les données dans le RR SOA décrivent comment est gérée la zone. Les données de zone sont maintenues sur l'hôte SRI-NIC.ARPA, et le responsable de cette zone est joignable à l'adresse HOSTMASTER@SRI-NIC.ARPA. Une des données clef de ce SOA est la durée de vie minimum marquée 86400, qui signifie que toutes les données autorisées de cette zone sont valides pendant au moins cette durée, mais que des valeurs plus grandes peuvent être explicitement spécifiées.

Les RR NS pour les domaines MIL et EDU marquent la frontière entre ce qui est du ressort de la zone racine et ce qui est du ressort des sous-domaines MIL et EDU. Notez que dans cet exemple, un zone de niveau inférieure se retrouve gérée par un serveur supportant la zone racine.

Le fichier maître de la zone EDU devra être constitué relativement au point EDU. Les données de zone pour le domaine EDU pourraient être :

```
EDU.  IN SOA SRI-NIC.ARPA. HOSTMASTER.SRI-NIC.ARPA. (
                        870729 ;serial
                        1800 ;refresh every 30 minutes
                        300 ;retry every 5 minutes
                        604800 ;expire after a week
                        86400 ;minimum of a day
                        )
                NS SRI-NIC.ARPA.
                NS C.ISI.EDU.

UCI 172800 NS ICS.UCI
                172800 NS ROME.UCI
ICS.UCI 172800 A 192.5.19.1
ROME.UCI 172800 A 192.5.19.31
ISI 172800 NS VAXA.ISI
                172800 NS A.ISI
                172800 NS VENERA.ISI.EDU.
VAXA.ISI 172800 A 10.2.0.27
                172800 A 128.9.0.33
VENERA.ISI.EDU. 172800 A 10.1.0.52
                172800 A 128.9.0.32
A.ISI 172800 A 26.3.0.103

UDEL.EDU.  172800 NS LOUIE.UDEL.EDU.
                172800 NS UMN-REI-UC.ARPA.
LOUIE.UDEL.EDU. 172800 A 10.0.0.96
                172800 A 192.5.39.3

YALE.EDU.  172800 NS YALE.ARPA.
YALE.EDU.  172800 NS YALE-BULLDOG.ARPA.

MIT.EDU.  43200 NS XX.LCS.MIT.EDU.
                  43200 NS ACHILLES.MIT.EDU.
XX.LCS.MIT.EDU.  43200 A 10.0.0.44
ACHILLES.MIT.EDU. 43200 A 18.72.0.8
```

Notez ici l'utilisation de noms relatifs. Le nom du propriétaire (owner) pour ISI.EDU. est noté en mode relatif, ainsi que le contenu de deux RR relatifs aux serveurs de noms. Les noms de domaines exprimés en relatif et en absolu peuvent être librement mélangés dans un fichier maître.

6.2. Exemple de requête standard

Les requêtes et réponses suivantes illustrent le comportement d'un serveur de noms. Sauf mention contraire, les requêtes ne demandent pas le mode récursif (RD) dans leur en-tête. Notez que les réponses à des requêtes non-récursives dépendent du serveur de noms contacté, et non de l'identité du requérant.

6.2.1. QNAME=SRI-NIC.ARPA, QTYPE=A

La requête aurait l'aspect suivant :

```
             +---------------------------------------------------+
En-tête      | OPCODE=SQUERY                                     |
             +---------------------------------------------------+
Question     | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=A           |
             +---------------------------------------------------+
Réponse      | <vide>                                            |
             +---------------------------------------------------+
Autorisation | <vide>                                            |
```

```
              +-------------------------------------------------+
Additionnel   | <vide>                                          |
              +-------------------------------------------------+
```

La réponse de C.ISI.EDU serait :

```
              +-------------------------------------------------+
En-tête       | OPCODE=SQUERY, RESPONSE, AA                     |
              +-------------------------------------------------+
Question      | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=A         |
              +-------------------------------------------------+
Réponse       | SRI-NIC.ARPA. 86400 IN A 26.0.0.73              |
              |               86400 IN A 10.0.0.51              |
              +-------------------------------------------------+
Autorisation  | <vide>                                          |
              +-------------------------------------------------+
Additionnel   | <vide>                                          |
              +-------------------------------------------------+
```

L'en-tête de la réponse ressemble à celle de la requête, excepté que le bit RESPONSE est marqué, indiquant que ce message est bel est bien une réponse, et non une requête, le bit Réponse Autorisée (AUTHORITATIVE ANSWER = AA) est marqué, indiquant que les adresses données dans les RR de la section réponse sont "autorisées". La section question de la réponse est une recopie de la section question de la requête.

Si la même requête avait été envoyée à d'autres serveurs "non autorisés" pour le domaine SRI-NIC.ARPA, la réponse aurait été :

```
              +-------------------------------------------------+
En-tête       | OPCODE=SQUERY,RESPONSE                          |
              +-------------------------------------------------+
Question      | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=A         |
              +-------------------------------------------------+
Réponse       | SRI-NIC.ARPA. 1777 IN A 10.0.0.51               |
              |               1777 IN A 26.0.0.73               |
              +-------------------------------------------------+
Autorisation  | <vide>                                          |
              +-------------------------------------------------+
Additionnel   | <vide>                                          |
              +-------------------------------------------------+
```

Cette réponse est différente de la précédente à deux titres : l'en-tête ne présente pas le bit AA marqué, et les durées de vie sont différentes. La déduction peut être faite que ces données ne proviennent pas d'une zone, mais plutôt d'un cache. La différence de durée de vie est alors due à l'intervention de la durée passée de stationnement dans le cache. Les différences d'ordre dans lequel les RR de la section Réponse sont présentés n'est pas significative.

6.2.2. QNAME=SRI-NIC.ARPA, QTYPE=*

Une requête similaire à la précédente, mais utilisant un QTYPE "*", recevrait la réponse suivante de la part de C.ISI.EDU:

```
              +-------------------------------------------------+
En-tête       | OPCODE=SQUERY, RESPONSE, AA                     |
              +-------------------------------------------------+
Question      | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=*         |
              +-------------------------------------------------+
Réponse       | SRI-NIC.ARPA. 86400 IN  A     26.0.0.73         |
              |                         A     10.0.0.51         |
              |                         MX    0 SRI-NIC.ARPA.   |
              |                         HINFO DEC-2060 TOPS20   |
              +-------------------------------------------------+
Autorisation  | <vide>                                          |
              +-------------------------------------------------+
```

```
Additionnel  | <vide>                                              |
             +-----------------------------------------------------+
```

Si une requête identique était émise vers des serveurs "non autorisés" pour le domaine SRI-NIC.ARPA, la réponse aurait été :

```
             +-----------------------------------------------------+
En-tête      | OPCODE=SQUERY, RESPONSE                             |
             +-----------------------------------------------------+
Question     | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=*             |
             +-----------------------------------------------------+
Réponse      | SRI-NIC.ARPA. 12345 IN     A       26.0.0.73        |
             |                            A       10.0.0.51        |
             +-----------------------------------------------------+
Autorisation | <vide>                                              |
             +-----------------------------------------------------+
Additionnel  | <vide>                                              |
             +-----------------------------------------------------+
```

et

```
             +-----------------------------------------------------+
En-tête      | OPCODE=SQUERY, RESPONSE                             |
             +-----------------------------------------------------+
Question     | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=*             |
             +-----------------------------------------------------+
Réponse      | SRI-NIC.ARPA. 1290 IN HINFO  DEC-2060 TOPS20        |
             +-----------------------------------------------------+
Autorisation | <vide>                                              |
             +-----------------------------------------------------+
Additionnel  | <vide>                                              |
             +-----------------------------------------------------+
```

Aucune de ces deux réponse ne présente le bit AA marqué, et donc aucune des deux ne s'est basée sur des données "autorisées". Les différences de contenu et de durées de vie suggèrent que les deux serveurs ont enregistré les données dans leur cache à une date différente, et que le premier serveur à obtenu ces données suite à une requête QTYPE=A, le second les ayant obtenues suite à une requête HINFO.

6.2.3. QNAME=SRI-NIC.ARPA, QTYPE=MX

Ce type de requête pourrait émaner d'un agent de courrier électronique essayant de trouver le chemin pour le destinataire de courrier HOSTMASTER@SRI-NIC.ARPA. La réponse de C.ISI.EDU serait :

```
             +-----------------------------------------------------+
En-tête      | OPCODE=SQUERY, RESPONSE, AA                         |
             +-----------------------------------------------------+
Question     | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=MX            |
             +-----------------------------------------------------+
Réponse      | SRI-NIC.ARPA. 86400 IN     MX      0 SRI-NIC.ARPA.  |
             +-----------------------------------------------------+
Autorisation | <vide>                                              |
             +-----------------------------------------------------+
Additionnel  | SRI-NIC.ARPA. 86400 IN     A       26.0.0.73        |
             |                            A       10.0.0.51        |
             +-----------------------------------------------------+
```

Cette réponse contient un RR MX dans la section Réponse du message. La section additionnelle contient les RR d'adresse car le serveur de nom à C.ISI.EDU devine que le requérant aura besoin de ces adresses pour exploiter correctement les information transmises dans le RR MX.

6.2.4. QNAME=SRI-NIC.ARPA, QTYPE=NS

C.ISI.EDU répondrait à la requête par :

```
             +-------------------------------------------------+
En-tête      | OPCODE=SQUERY, RESPONSE, AA                     |
             +-------------------------------------------------+
Question     | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=NS        |
             +-------------------------------------------------+
Réponse      | <vide>                                          |
             +-------------------------------------------------+
Autorisation | <vide>                                          |
             +-------------------------------------------------+
Additionnel  | <vide>                                          |
             +-------------------------------------------------+
```

La seule différence entre la requête et la réponse est le marquage des bits AA et RESPONSE dans l'en-tête. L'interprétation de cette réponse est que le serveur est bien "autorisé" pour le nom en question, lequel existe, mais pour lequel aucun enregistrement de type NS n'est disponible.

6.2.5. QNAME=SIR-NIC.ARPA, QTYPE=A

Lorsqu'un utilisateur orthographie mal un nom de domaine, nous pourrons voir ce type de requête. C.ISI.EDU répondrait dans ce cas :

```
             +-------------------------------------------------+
En-tête      | OPCODE=SQUERY, RESPONSE, AA, RCODE=NE           |
             +-------------------------------------------------+
Question     | QNAME=SIR-NIC.ARPA., QCLASS=IN, QTYPE=A         |
             +-------------------------------------------------+
Réponse      | <vide>                                          |
             +-------------------------------------------------+
Autorisation | . SOA SRI-NIC.ARPA. HOSTMASTER.SRI-NIC.ARPA.    |
             |       870611 1800 300 604800 86400              |
             +-------------------------------------------------+
Additionnel  | <vide>                                          |
             +-------------------------------------------------+
```

Cette réponse indique que le nom demandé n'existe pas. Cette condition est exprimée par le code de réponse (RCODE) dans l'en-tête.

Le RR SOA dans la section Autorisation est l'information optionnelle à enregistrer en cas de cache des réponses négatives, qui permet au résolveur utilisant cette réponse de supposer que le nom recherché n'existera pas pendant au moins SOA MINIMUM (86400) secondes.

6.2.6. QNAME=BRL.MIL, QTYPE=A

Si cette requête était envoyée à C.ISI.EDU, la réponse serait :

```
             +-------------------------------------------------+
En-tête      | OPCODE=SQUERY, RESPONSE                         |
             +-------------------------------------------------+
Question     | QNAME=BRL.MIL, QCLASS=IN, QTYPE=A               |
             +-------------------------------------------------+
Réponse      | <vide>                                          |
             +-------------------------------------------------+
Autorisation | MIL.            86400 IN NS       SRI-NIC.ARPA. |
             |                 86400    NS       A.ISI.EDU.    |
             +-------------------------------------------------+
Additionnel  | A.ISI.EDU.              A         26.3.0.103    |
             | SRI-NIC.ARPA.           A         26.0.0.73     |
             |                         A         10.0.0.51     |
             +-------------------------------------------------+
```

Cette réponse affiche une section Réponse vide, et n'est pas "autorisée". Il s'agit donc d'une redirection. Le serveur de noms à C.ISI.EDU, réalisant qu'il n'est pas "autorisé" pour le domaine MIL, redirige le requérant vers les serveurs à A.ISI.EDU et SRI-NIC.ARPA, qu'il connaît être autorisés pour le domaine MIL.

6.2.7. QNAME=USC-ISIC.ARPA, QTYPE=A

La réponse à cette requête venant de A.ISI.EDU serait :

```
              +-------------------------------------------------+
En-tête       | OPCODE=SQUERY, RESPONSE, AA                     |
              +-------------------------------------------------+
Question      | QNAME=USC-ISIC.ARPA., QCLASS=IN, QTYPE=A        |
              +-------------------------------------------------+
Réponse       | USC-ISIC.ARPA. 86400 IN CNAME      C.ISI.EDU.   |
              | C.ISI.EDU.     86400 IN A          10.0.0.52    |
              +-------------------------------------------------+
Autorisation  | <vide>                                          |
              +-------------------------------------------------+
Additionnel   | <vide>                                          |
              +-------------------------------------------------+
```

Notez que le bit AA dans l'en-tête garantit que les données correspondantes au QNAME sont autorisées, mais ne permet pas d'en déduire l'état d'autorisation pour les données concernant C.ISI.EDU. Cette réponse complète est possible car A.ISI.EDU se trouve être autorisé à la fois pour le domaine ARPA ou se trouve USC-ISIC.ARPA et pour le domaine ISI.EDU où l'on trouve C.ISI.EDU.

Si la même requête était envoyée à C.ISI.EDU, la réponse serait la même que celle ci-dessus, dans le cas où il disposerait de ces adresses en cache, mais pourrait aussi prendre la forme :

```
              +-------------------------------------------------+
En-tête       | OPCODE=SQUERY, RESPONSE, AA                     |
              +-------------------------------------------------+
Question      | QNAME=USC-ISIC.ARPA., QCLASS=IN, QTYPE=A        |
              +-------------------------------------------------+
Réponse       | USC-ISIC.ARPA.   86400 IN CNAME   C.ISI.EDU.    |
              +-------------------------------------------------+
Autorisation  | ISI.EDU.        172800 IN NS      VAXA.ISI.EDU. |
              |                         NS      A.ISI.EDU.    |
              |                         NS      VENERA.ISI.EDU. |
              +-------------------------------------------------+
Additionnel   | VAXA.ISI.EDU.   172800    A       10.2.0.27     |
              |                 172800    A       128.9.0.33    |
              | VENERA.ISI.EDU. 172800    A       10.1.0.52     |
              |                 172800    A       128.9.0.32    |
              | A.ISI.EDU.      172800    A       26.3.0.103    |
              +-------------------------------------------------+
```

Cette réponse contient des données autorisées pour l'alias USC-ISIC.ARPA, plus des données de redirection vers les serveurs de noms pour ISI.EDU. Cette sorte de réponse ne serait en général pas celle donnée si la requête visait le nom d'hôte du serveur de noms lui-même, mais serait courante pour d'autres alias.

6.2.8. QNAME=USC-ISIC.ARPA, QTYPE=CNAME

Si cette requête est émise vers A.ISI.EDU ou C.ISI.EDU, la réponse serait :

```
              +-------------------------------------------------+
En-tête       | OPCODE=SQUERY, RESPONSE, AA                     |
              +-------------------------------------------------+
Question      | QNAME=USC-ISIC.ARPA., QCLASS=IN, QTYPE=A        |
              +-------------------------------------------------+
Réponse       | USC-ISIC.ARPA. 86400 IN CNAME      C.ISI.EDU.   |
              +-------------------------------------------------+
Autorisation  | <vide>                                          |
```

```
              +-------------------------------------------------+
Additionnel   | <vide>                                          |
              +-------------------------------------------------+
```

Parce que le QTYPE=CNAME, le RR CNAME lui-même répond à la requête, et le serveur de nom ne tentera pas de rechercher quoi que ce soit d'autre concernant C.ISI.EDU. (Sauf éventuellement pour renseigner une section additionnelle).

6.3. Exemple de résolution

Les exemples suivants illustrent les opérations qu'un résolveur devra effectuer pour son client. Nous supposons que le résolveur part avec un cache vide, ce qui pourrait être le cas après un redémarrage système. Nous supposerons de plus que le système n'est pas l'un des hôtes marqué dans les données de zone et que l'hôte est situé quelque part sur le réseau d'adresse 26, et de plus, que sa structure "ceinture de sécurité" (SBELT) est constituée comme suit :

```
Match count = -1
SRI-NIC.ARPA.   26.0.0.73       10.0.0.51
A.ISI.EDU.      26.3.0.103
```

Ces informations spécifient les serveurs à essayer, leurs adresses, et un décompte de concordance à -1, qui indique que les serveurs ne sont pas "proches" de la cible. Notez que cette valeur -1 n'exprime pas une mesure précise de la "distance", mais juste une valeur conventionnelle sur laquelle se basera l'algorithme au départ.

Les exemples suivants illustrent l'utilisation de la méthode du cache et sa construction, et donc chaque étape suppose que la requête de l'étape précédente a été résolue.

6.3.1. Résolution de MX pour ISI.EDU.

Supposez que la première demande au résolveur provienne de l'agent de courrier local, qui doit émettre un message à PVM@ISI.EDU. L'agent de courrier demande alors les RR MX pour le domaine ISI.EDU.

Le résolveur va d'abord chercher dans son cache des RR MX RR pour ISI.EDU, mais le cache est vide et cela ne résout rien. Le résolveur va en conclure qu'il doit contacter un serveur de nom distant et doit dans un premier temps déterminer quel est le meilleur serveur à contacter pour cette requête. Cette phase recherche dans le cache des RR NS pour le domaine ISI.EDU ou l'un de ses pères, EDU, et la racine. Le cache étant vide, ces recherches échoueront de la même manière. En dernier ressort, le résolveur va utiliser les informations de la structure SBELT, en les copiant dans la structure SLIST.

Arrivé à ce point, le résolveur devra choisir l'une des trois adresses disponibles et essayer une requête. Du fait que le résolveur est sur le réseau d'adresse 26, il choisira l'une des deux adresses 26.0.0.73 ou 26.3.0.103 en premier. Il émettra donc une requête de la forme :

```
              +-------------------------------------------------+
En-tête       | OPCODE=SQUERY                                   |
              +-------------------------------------------------+
Question      | QNAME=ISI.EDU., QCLASS=IN, QTYPE=MX             |
              +-------------------------------------------------+
Réponse       | <vide>                                          |
              +-------------------------------------------------+
Autorisation  | <empty>                                         |
              +-------------------------------------------------+
Additionnel   | <vide>                                          |
              +-------------------------------------------------+
```

Le résolveur attendra pour cette requête soit une réponse, soit la durée d'une temporisation. Si la temporisation déclenche, il essayera le serveur suivant, ou une adresse différente pour le même

serveur, et enfin en réessayant les adresses déjà tentées. Il pourra donc recevoir une réponse du serveur à SRI-NIC.ARPA:

```
              +-------------------------------------------------+
En-tête       | OPCODE=SQUERY, RESPONSE                         |
              +-------------------------------------------------+
Question      | QNAME=ISI.EDU., QCLASS=IN, QTYPE=MX             |
              +-------------------------------------------------+
Réponse       | <vide>                                          |
              +-------------------------------------------------+
Autorisation  | ISI.EDU.        172800 IN NS       VAXA.ISI.EDU.  |
              |                           NS       A.ISI.EDU.     |
              |                           NS       VENERA.ISI.EDU.|
              +-------------------------------------------------+
Additionnel   | VAXA.ISI.EDU.   172800    A        10.2.0.27      |
              |                 172800    A        128.9.0.33     |
              | VENERA.ISI.EDU. 172800    A        10.1.0.52      |
              |                 172800    A        128.9.0.32     |
              | A.ISI.EDU.      172800    A        26.3.0.103     |
              +-------------------------------------------------+
```

Le résolveur s'apercevra alors que la réponse propose une redirection vers un serveur plus "proche" d'ISI.EDU que ceux qu'il référence dans sa structure SLIST (du fait que trois identifiants correspondent). Le résolveur enregistrera cette information dans son cache et ajoutera ces références à sa SLIST :

```
Match count = 3
A.ISI.EDU.      26.3.0.103
VAXA.ISI.EDU.   10.2.0.27       128.9.0.33
VENERA.ISI.EDU. 10.1.0.52       128.9.0.32
```

A.ISI.EDU apparaît dans cette liste ainsi que le précédent, bien qu'il ne s'agisse ici que d'une coïncidence. Le résolveur recommence la transmission de requêtes et attend des nouvelles réponses. Le cas échéant, il obtient une réponse :

```
              +-------------------------------------------------+
En-tête       | OPCODE=SQUERY, RESPONSE, AA                     |
              +-------------------------------------------------+
Question      | QNAME=ISI.EDU., QCLASS=IN, QTYPE=MX             |
              +-------------------------------------------------+
Réponse       | ISI.EDU.                MX 10 VENERA.ISI.EDU.   |
              |                         MX 20 VAXA.ISI.EDU.     |
              +-------------------------------------------------+
Autorisation  | <vide>                                          |
              +-------------------------------------------------+
Additionnel   | VAXA.ISI.EDU.   172800  A  10.2.0.27            |
              |                 172800  A  128.9.0.33           |
              | VENERA.ISI.EDU. 172800  A  10.1.0.52            |
              |                 172800  A  128.9.0.32           |
              +-------------------------------------------------+
```

Le résolveur ajoutera cette information à son cache, et renvoie le RR MX obtenu au client.

6.3.2. Obtenir le nom d'hôte à partir de l'adresse 26.6.0.65

Le résolveur va traduire ceci en une requête sur des RR PTR pour le domaine 65.0.6.26.IN-ADDR.ARPA. Cette information n'est pas dans le cache, et donc le résolveur doit chercher un serveur distant auquel la demander. Aucun serveur du cache ne va correspondre, et il faudra exploiter les données de SBELT une fois encore. (Notez que les serveurs pour le domaine ISI.EDU sont bien mentionnés dans le cache, mais ISI.EDU n'est pas un "ancêtre" du domaine 65.0.6.26.IN-ADDR.ARPA, et donc SBELT sera réutilisé).

Comme l'information recherchée fait partie des données autorisées des deux serveurs mentionnés dans SBELT, l'un d'eux répondra certainement :

```
              +-------------------------------------------------+
En-tête       | OPCODE=SQUERY, RESPONSE, AA                     |
              +-------------------------------------------------+
Question      | QNAME=65.0.6.26.IN-ADDR.ARPA.,QCLASS=IN,QTYPE=PTR |
              +-------------------------------------------------+
Réponse       | 65.0.6.26.IN-ADDR.ARPA.    PTR     ACC.ARPA.    |
              +-------------------------------------------------+
Autorisation  | <vide>                                          |
              +-------------------------------------------------+
Additionnel   | <vide>                                          |
              +-------------------------------------------------+
```

6.3.3. Obtenir l'adresse de l'hôte du domaine poneria.ISI.EDU

Cette requête se traduira par un requête sur le type A pour le domaine poneria.ISI.EDU. Le résolveur ne tentera pas d'exploiter une quelconque donnée cachée pour ce nom, mais trouvera dans le cache les RR NS pour le serveur relatif à la zone ISI.EDU lorsqu'il y recherche des adresses de serveurs de noms distants. Avec ces données, il construira une SLIST de la forme :

```
Match count = 3

A.ISI.EDU.       26.3.0.103
VAXA.ISI.EDU.    10.2.0.27       128.9.0.33
VENERA.ISI.EDU.  10.1.0.52
```

A.ISI.EDU est choisi en premier, en suivant le raisonnement que le résolveur utilise une méthode hiérarchique préférentielle , et du fait qu'A.ISI.EDU est sur le même réseau.

Un de ces serveurs doit pouvoir répondre à la requête.

7. REFERENCES et BIBLIOGRAPHIE

[Dyer 87] Dyer, S., and F. Hsu, "Hesiod", Project Athena Technical Plan - Name Service, April 1987, version 1.9. Describes the fundamentals of the Hesiod name service.

[IEN-116] J. Postel, "Internet Name Server", IEN-116, USC/Information Sciences Institute, August 1979. A name service obsoleted by the Domain Name System, but still in use.

[Quarterman 86] Quarterman, J., and J. Hoskins, "Notable Computer Networks",Communications of the ACM, October 1986, volume 29, number 10.

[RFC-742] K. Harrenstien, "NAME/FINGER", RFC-742, Network Information Center, SRI International, December 1977.

[RFC-768] J. Postel, "User Datagram Protocol", RFC-768, USC/Information Sciences Institute, August 1980.

[RFC-793] J. Postel, "Transmission Control Protocol", RFC-793, USC/Information Sciences Institute, September 1981.

[RFC-799] D. Mills, "Internet Name Domains", RFC-799, COMSAT, September 1981. Suggests introduction of a hierarchy in place of a flat name space for the Internet.

[RFC-805] J. Postel, "Computer Mail Meeting Notes", RFC-805, USC/Information Sciences Institute, February 1982.

[RFC-810] E. Feinler, K. Harrenstien, Z. Su, and V. White, "DOD Internet Host Table Specification", RFC-810, Network Information Center, SRI International, March 1982. Obsolete. See RFC-952.

[RFC-811] K. Harrenstien, V. White, and E. Feinler, "Hostnames Server", RFC-811, Network Information Center, SRI International, March 1982. Obsolete. See RFC-953.

[RFC-812] K. Harrenstien, and V. White, "NICNAME/WHOIS", RFC-812, Network Information Center, SRI International, March 1982.

[RFC-819] Z. Su, and J. Postel, "The Domain Naming Convention for Internet User Applications", RFC-819, Network Information Center, SRI International, August 1982. Early thoughts on the design of the domain system. Current implementation is completely different.

[RFC-821] J. Postel, "Simple Mail Transfer Protocol", RFC-821, USC/Information Sciences Institute, August 1980.
[RFC-830] Z. Su, "A Distributed System for Internet Name Service", RFC-830, Network Information Center, SRI International, October 1982. Early thoughts on the design of the domain system. Current implementation is completely different.
[RFC-882] P. Mockapetris, "Domain names - Concepts and Facilities," RFC-882, USC/Information Sciences Institute, November 1983. Superceeded by this memo.
[RFC-883] P. Mockapetris, "Domain names - Implementation and Specification," RFC-883, USC/Information Sciences Institute, November 1983. Superceeded by this memo.
[RFC-920] J. Postel and J. Reynolds, "Domain Requirements", RFC-920, USC/Information Sciences Institute October 1984. Explains the naming scheme for top level domains.
[RFC-952] K. Harrenstien, M. Stahl, E. Feinler, "DoD Internet Host Table Specification", RFC-952, SRI, October 1985. Specifies the format of HOSTS.TXT, the host/address table replaced by the DNS.
[RFC-953] K. Harrenstien, M. Stahl, E. Feinler, "HOSTNAME Server", RFC-953, SRI, October 1985. This RFC contains the official specification of the hostname server protocol, which is obsoleted by the DNS. This TCP based protocol accesses information stored in the RFC-952 format, and is used to obtain copies of the host table.
[RFC-973] P. Mockapetris, "Domain System Changes and Observations", RFC-973, USC/Information Sciences Institute, January 1986. Describes changes to RFC-882 and RFC-883 and reasons for them. Now obsolete.
[RFC-974] C. Partridge, "Mail routing and the domain system", RFC-974, CSNET CIC BBN Labs, January 1986. Describes the transition from HOSTS.TXT based mail addressing to the more powerful MX system used with the domain system.
[RFC-1001] NetBIOS Working Group, "Protocol standard for a NetBIOS service on a TCP/UDP transport: Concepts and Methods", RFC-1001, March 1987. This RFC and RFC-1002 are a preliminary design for NETBIOS on top of TCP/IP which proposes to base NetBIOS name service on top of the DNS.
[RFC-1002] NetBIOS Working Group, "Protocol standard for a NetBIOS service on a TCP/UDP transport: Detailed Specifications", RFC-1002, March 1987.
[RFC-1010] J. Reynolds and J. Postel, "Assigned Numbers", RFC-1010, USC/Information Sciences Institute, May 1987 Contains socket numbers and mnemonics for host names, operating systems, etc.
[RFC-1031] W. Lazear, "MILNET Name Domain Transition", RFC-1031, November 1987. Describes a plan for converting the MILNET to the DNS.
[RFC-1032] M. K. Stahl, "Establishing a Domain - Guidelines for Administrators", RFC-1032, November 1987. Describes the registration policies used by the NIC to administer the top level domains and delegate subzones.
[RFC-1033] M. K. Lottor, "Domain Administrators Operations Guide", RFC-1033, November 1987. A cookbook for domain administrators.
[Solomon 82] M. Solomon, L. Landweber, and D. Neuhengen, "The CSNET Name Server", Computer Networks, vol 6, nr 3, July 1982. Describes a name service for CSNET which is independent from the DNS and DNS use in the CSNET.