

Groupe de travail Réseau
Request for Comments : 1035
STD 13
RFC rendues obsolètes : RFC 882, 883, 973

P. Mockapetris
ISI
novembre 1987
Traduction Claude Brière de L'Isle

Noms de domaines – mise en œuvre et spécification

1. Statut du présent mémoire

La présente RFC décrit les détails du système et protocole des domaines, et suppose que les concepts exposés dans la RFC "Serveur de noms de domaines - Concepts et éléments de base" [RFC1034] sont familiers au lecteur.

Le système des domaines est un mélange de types de fonctions et de données qui sont un protocole officiel et de types de fonctions et de données qui sont encore expérimentales. Comme le système des domaines est intentionnellement extensible, de nouveaux types de données et des comportements expérimentaux devraient toujours être envisagés dans les parties du système qui vont au delà du protocole officiel. Les parties du protocole officiel comportent des interrogations standard, des réponses et les formats de données RR de classe Internet (par exemple, les adresses d'hôtes). Comme dans le précédent ensemble de RFC, plusieurs définitions ont changé, de sorte que certaines définitions antérieures sont obsolètes.

Les dispositifs expérimentaux ou obsolètes sont clairement indiqués dans la présente RFC, et de telles informations devraient être utilisées avec précaution.

Le lecteur est particulièrement invité à ne pas se fier aux valeurs qui apparaissent dans les exemples, qui ne sont ni actuelles ni complètes, car leur objet est principalement pédagogique. La distribution du présent mémoire n'est soumise à aucune restriction.

Table des matières

1. Statut du présent mémoire.....	1
2. Introduction.....	2
2.1 Généralités.....	2
2.2 Configurations communes.....	2
2.3 Conventions.....	5
3. Espace de nom de domaine et définitions de RR.....	7
3.1 Définitions d'espace de nom.....	7
3.2 Définitions de RR.....	7
3.3 RR standard.....	8
3.4 RR spécifiques de l'Internet.....	13
3.5 Domaine IN-ADDR.ARPA.....	13
3.6 Définition de nouveaux types, classes, et espaces de nom particuliers.....	15
4. Messages.....	15
4.1 Format.....	15
4.2 Transport.....	19
5. Fichiers maîtres.....	20
5.1 Format.....	20
5.2 Utilisation des fichiers maîtres pour définir des zones.....	21
5.3 Exemple de fichier maître.....	21
6. Mise en œuvre du serveur de noms.....	22
6.1 Architecture.....	22
6.2 Processus d'interrogation standard.....	23
6.3 Rafraîchissement de zone et processus de rechargement.....	24
6.4 Interrogations inverses (facultatif).....	24
6.5 Achèvement des interrogations et des réponses.....	25
7. Mise en œuvre du résolveur.....	25
7.1 Transformation d'une demande d'utilisateur en une interrogation.....	26
7.2 Envoi des interrogations.....	26
7.3 Traitement des réponses.....	27
7.4 Utilisation de l'antémémoire.....	28
8. Prise en charge de la messagerie.....	28
8.1 Liens d'échange de messagerie.....	29
8.2 Lien de messagerie (expérimental).....	29

9. Références et bibliographie.....30

2. Introduction

2.1 Généralités

Le but des noms de domaine est de fournir un mécanisme pour désigner des ressources d'une façon telle que les noms soient utilisables dans des hôtes, réseaux, familles de protocoles, internets, et organisations administratives différentes. Du point de vue de l'utilisateur, les noms de domaine sont utiles comme arguments envers un agent local, appelé un résolveur, qui restitue les informations associées au nom de domaine. Et donc un utilisateur peut demander l'adresse d'un hôte ou les informations de messagerie associées à un nom de domaine particulier. Pour permettre à l'utilisateur de demander un type d'informations particulier, un type d'interrogation approprié est transmis au résolveur avec le nom de domaine. Pour l'usager, l'arbre des domaines est un seul espace d'information ; le résolveur est chargé de cacher à l'usager la distribution de données entre les serveurs de noms.

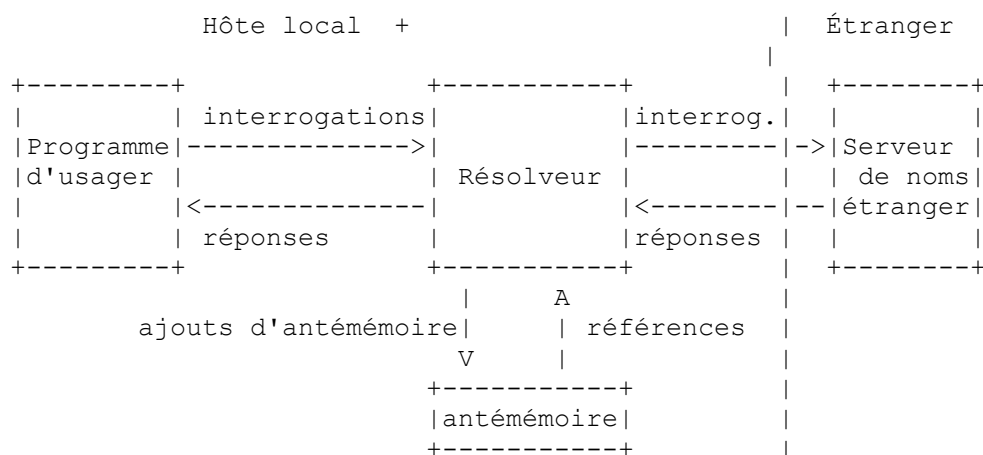
Du point de vue du résolveur, la base de données qui constitue l'espace de domaine est distribuée entre divers serveurs de noms. Différentes parties de l'espace de domaine sont mémorisées dans différents serveurs de noms, bien qu'un élément de données particulier soit mémorisé de façon redondante dans deux serveurs de noms, ou plus. Le résolveur commence par connaître au moins un serveur de noms. Quand le résolveur traite l'interrogation d'un usager, il demande les informations à un serveur de noms connu ; en retour, le résolveur reçoit les informations désirées ou une référence à un autre serveur de noms. En utilisant ces références, les résolveurs apprennent les identités et le contenu des autres serveurs de noms. Les résolveurs sont chargés de s'occuper de la distribution de l'espace des domaines et de traiter des effets des défaillances des serveurs de noms en consultant les bases de données redondantes dans les autres serveurs.

Les serveurs de noms gèrent deux sortes de données. Les premières d'entre elles sont détenues dans des ensembles appelés des zones ; chaque zone est la base de données complète pour un sous-arbre "élagué" particulier de l'espace de domaine. Ces données sont appelées d'autorité. Un serveur de nom vérifie périodiquement que ses zones sont à jour, et si elles ne le sont pas, il obtient une nouvelle copie des zones mises à jour des fichiers maîtres mémorisés localement ou dans un autre serveur de noms. Les données du second type sont des données d'antémémoire qui sont acquises par un résolveur local. Ces données peuvent être incomplètes, mais elles améliorent les performances du processus de restitution lorsque on accède de façon répétée à des données non locales. Les données en antémémoire sont finalement éliminées par un mécanisme de temporisation

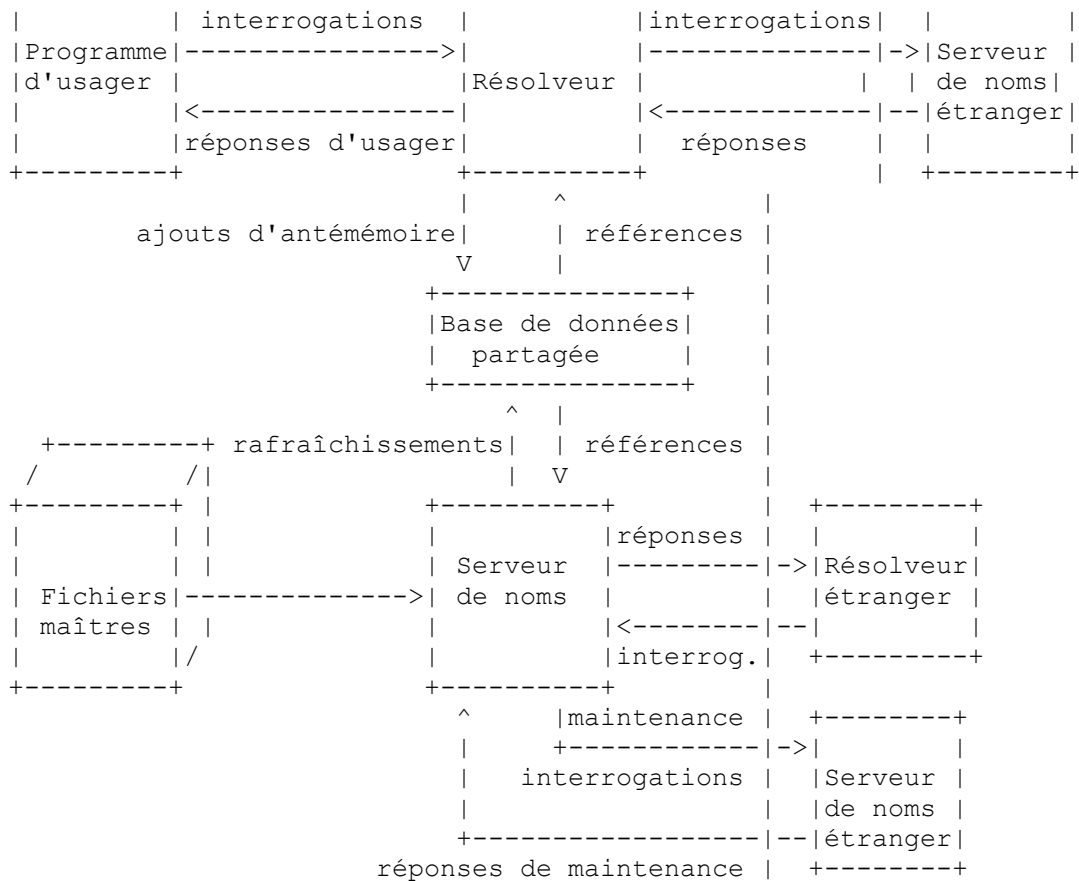
Cette structure fonctionnelle isole les problèmes de l'interface d'utilisateur, de la récupération des défaillances, et de la distribution dans les résolveurs et isole les problèmes de mise à jour et de rafraîchissement de base de données dans les serveurs de noms.

2.2 Configurations communes

Un hôte peut participer de nombreuses façons au système de noms de domaines, selon qu'il a des programmes qui restituent les informations provenant du système des domaines, des serveurs de noms qui répondent aux interrogations provenant des autres hôtes, ou de diverses combinaisons des deux fonctions. La configuration la plus simple, et peut-être la plus normale, est indiquée ci-dessous :

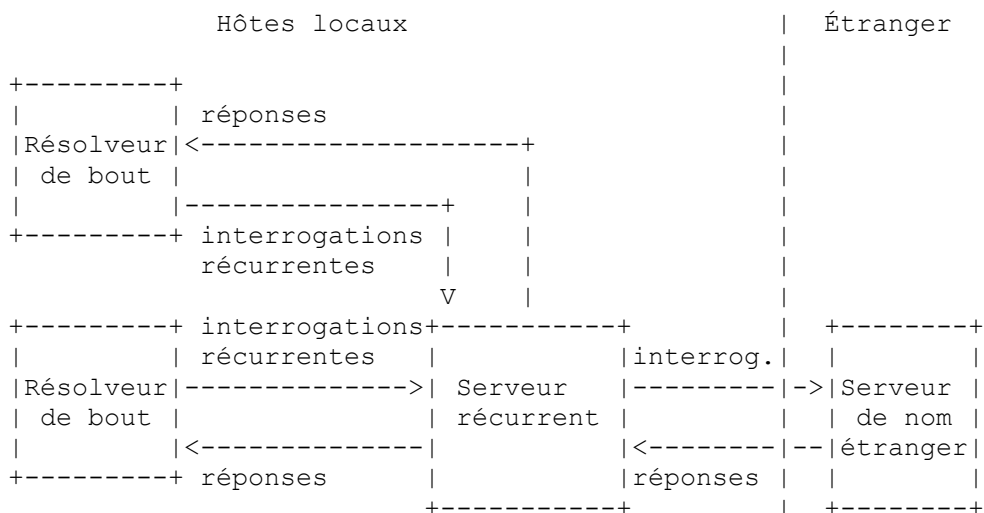


Les programmes d'usager interagissent avec l'espace de noms de domaines à travers les résolveurs ; le format des



La base de données partagée détient les données de l'espace de domaine pour le serveur de noms local et le résolveur. Le contenu de la base de données partagée sera normalement un mélange de données d'autorité entretenues par les opérations périodiques de rafraîchissement du serveur de noms et de données de l'antémémoire provenant des demandes antérieures du résolveur. La structure des données de domaine et la nécessité de la synchronisation entre les serveurs de noms et les résolveurs impliquent les caractéristiques générales de cette base de données, mais le format réel est une décision de la mise en œuvre locale.

Les flux d'informations peuvent aussi être taillés sur mesure de sorte qu'un groupe d'hôtes agissent ensemble pour optimiser les activités. Cela est fait parfois pour décharger les hôtes de plus faible capacité de sorte qu'ils n'aient pas à mettre en œuvre un résolveur complet. Cela peut être approprié pour les micro ordinateurs ou les hôtes qui veulent minimiser la quantité de nouveau code réseau nécessaire. Ce schéma peut aussi permettre à un groupe d'hôtes de partager un plus petit nombre d'antémémoires plutôt que d'entretenir un grand nombre d'antémémoires distinctes, sachant que des antémémoires centralisées auront un plus fort rendement. Dans les deux cas, les résolveurs sont remplacés par des sous résolveurs qui agissent comme frontaux des résolveurs localisés dans un serveur récursif dans un ou plusieurs serveurs de noms connus pour effectuer ce service :



```

|Antémémoire|      |
| centrale   |      |
+-----+      |

```

Dans tous les cas, noter que les composants de domaines sont toujours dupliqués pour la fiabilité chaque fois que possible.

2.3 Conventions

Le système des domaines possède plusieurs conventions sur des questions de bas niveau, mais néanmoins fondamentales. Bien que les mises en œuvre soient libres de violer ces conventions AU SEIN DE LEUR PROPRE SYSTEME, elles doivent observer ces conventions dans TOUS les comportements observés par les autres hôtes.

2.3.1 Syntaxe de nom préférée

Les spécifications du DNS essayent d'être aussi générales que possible dans les règles de construction des noms de domaines. L'idée de base est que le nom de tout objet existant peut être exprimé comme un nom de domaine avec des changements minimes.

Cependant, lorsqu'on alloue un nom de domaine à un objet, l'utilisateur prudent va choisir un nom qui satisfait à la fois aux règles du système des domaines et à toutes les règles existantes pour l'objet, que ces règles soient publiques ou impliquées par les programmes existants.

Par exemple, pour nommer un domaine de messagerie, l'usager devrait respecter à la fois les règles du présent mémoire et celles de la [RFC0822]. Lors de la création d'un nouveau nom d'hôte, les vieilles règles pour HOSTS.TXT devraient être suivies. Cela évite des problèmes lorsque un vieux logiciel est converti pour utiliser les noms de domaines.

La syntaxe suivante donnera moins de problèmes avec de nombreuses applications qui utilisent les noms de domaines (par exemple, la messagerie électronique, TELNET).

```

<domaine> ::= <sous-domaine> | " "
<sous-domaine> ::= <étiquette> | <sous-domaine> "." <étiquette>
<étiquette> ::= <lettre> [ [ <ldh-str> ] <let-dig> ]
<ldh-str> ::= <let-dig-hyp> | <let-dig-hyp> <ldh-str>
<let-dig-hyp> ::= <let-dig> | "-"
<let-dig> ::= <lettre> | <chiffre>
<lettre> ::= n'importe lequel des 52 caractères alphabétiques de A à Z en majuscules et de a à z en minuscules
<chiffre> ::= n'importe lequel des dix chiffres de 0 à 9

```

Noter que bien que les lettres majuscules et minuscules soient admises dans les noms de domaines, aucune signification n'est attachée à la casse. C'est à dire que deux noms qui s'épellent de la même façon mais d'une casse différente sont à traiter comme identiques.

Les étiquettes doivent suivre les règles des noms d'hôtes ARPANET. Elles doivent commencer par une lettre, se terminer par une lettre ou un chiffre, et avoir seulement des lettres, des chiffres et traits d'union comme caractères intérieurs. Il y a aussi quelques restrictions sur la longueur. Les étiquettes ne doivent pas dépasser 63 caractères.

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

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

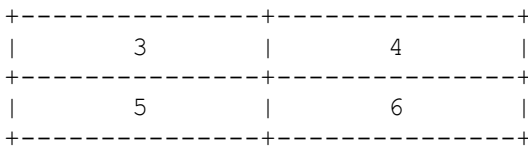
2.3.2 Ordre de transmission des données

L'ordre de transmission de l'en-tête et des données décrites dans le présent document est résolu au niveau de l'octet. Chaque fois qu'un diagramme montre un groupe d'octets, l'ordre de transmission de ces octets est l'ordre normal dans lequel ils se lisent en français. Par exemple, dans le diagramme suivant, les octets sont transmis dans l'ordre dans lequel ils sont numérotés.

```

      0                               1
0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
+-----+-----+-----+-----+
|           1           |           2           |

```



Chaque fois qu'un octet représente une quantité numérique, le bit le plus à gauche du diagramme est le bit d'ordre le plus élevé ou de plus fort poids. C'est à dire que le bit étiqueté 0 est le bit de plus fort poids. Par exemple, le diagramme suivant représente la valeur 170 (décimal).

0	1	2	3	4	5	6	7
1	0	1	0	1	0	1	0

De même, chaque fois qu'un champ multi-octet représente une quantité numérique, le bit le plus à gauche de tout le champ est le bit de plus fort poids. Lorsqu'une quantité multi-octet est transmise, l'octet de plus fort poids est transmis en premier.

2.3.3 Casse des caractères

Pour toutes les parties du DNS qui sont dans le protocole officiel, toutes les comparaisons entre chaînes de caractères (par exemple, étiquettes, noms de domaines, etc.) sont faites de manière insensible à la casse. À présent, cette règle est en vigueur sans exception dans tous le système des domaines. Cependant, des ajouts futurs en dehors de l'usage courant peuvent avoir besoin d'utiliser les pleines capacités des octets binaires dans les noms, de sorte que devraient être évitées les tentatives de mémorisation des noms de domaines en ASCII à 7 bits ou l'utilisation d'octets spéciaux pour terminer les étiquettes, etc..

Lorsque des données entrent dans le système des domaines, leur casse d'origine devrait être préservée chaque fois que possible. Dans certaines circonstances, ceci ne peut être tenu. Par exemple, si deux enregistrements de ressource (RR, *Resouce Record*) sont mémorisés dans une base de données, l'une à x.y et l'autre à X.Y, elles sont en réalité mémorisées au même endroit dans la base de données, et donc une seule écriture serait préservée. La règle de base est que la casse ne peut être éliminée que lorsque des données sont utilisées pour définir une structure dans une base de données, et deux noms sont identiques lorsqu'ils sont comparés d'une façon insensible à la casse.

La perte de données sensibles à la casse doit être minimisée. Et donc bien que les données pour x.y et X.Y puissent toutes deux être mémorisées dans une seule localisation x.y ou X.Y, les données pour a.x et B.X ne devraient jamais être mémorisées sous A.x, A.X, b.x, ou b.X. En général, cela préserve la casse de la première étiquette d'un nom de domaine, mais force à la normalisation des étiquettes de nœud intérieur.

Les administrateurs de systèmes qui entrent des données dans la base de données de domaine devraient veiller à représenter les données qu'ils fournissent au système de domaines d'une façon cohérente quant à la casse si leur système est sensible à la casse. Le système de distribution des données dans le système des domaines veillera à assurer que les représentations cohérentes sont préservées.

2.3.4 Limites de taille

Divers objets et paramètres du DNS ont des limites de taille. Leur liste figure ci-dessous. Certaines peuvent être changées facilement, d'autres sont plus fondamentales.

étiquettes	63 octets ou moins
noms	255 octets ou moins
TTL	valeurs positive d'un nombre de 32 bits signé.
messages UDP	512 octets ou moins

3. Espace de nom de domaine et définitions de RR

3.1 Définitions d'espace de nom

Les noms de domaine dans les messages sont exprimés en termes de séquences d'étiquettes. Chaque étiquette est représentée comme un champ d'une longueur d'un octet suivi par ce nombre d'octets. Comme chaque nom de domaine se termine par l'étiquette nulle de la racine, un nom de domaine se termine par un octet de longueur zéro. Les deux bits de plus fort poids de chaque octet de longueur doivent être zéro, et les six bits restants du champ Longueur limitent l'étiquette à 63 octets ou moins.

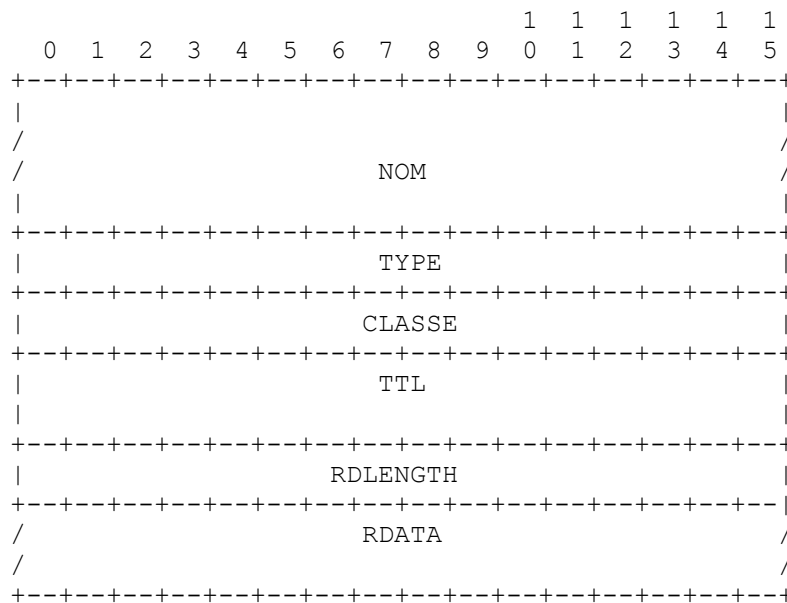
Pour simplifier les mises en œuvre, la longueur totale d'un nom de domaine (c'est-à-dire les octets de l'étiquette et les octets de longueur d'étiquette) est restreinte à 255 octets ou moins.

Bien que les étiquettes puissent contenir toutes valeurs de 8 bits en octets qui constituent l'étiquette, il est fortement recommandé que les étiquettes suivent la syntaxe préférée décrite plus loin dans ce mémoire, qui est compatible avec les conventions existantes pour la dénomination des hôtes. Les serveurs de noms et les résolveurs doivent comparer les étiquettes d'une façon insensible à la casse (c'est-à-dire, A = a) en supposant ASCII avec la parité zéro. Les codes non alphabétiques doivent correspondre exactement.

3.2 Définitions de RR

3.2.1 Format

Tous les RR ont le même format de niveau supérieur indiqué ci-dessous :



où :

- Nom un nom de détenteur, c'est à dire, le nom du nœud auquel appartient cet enregistrement de ressource.
- Type deux octets contenant un des codes de Type RR.
- Classe deux octets contenant un des codes de Classe RR.
- TTL un entier non signé de 32 bits qui spécifie l'intervalle de temps pendant lequel l'enregistrement de ressource peut rester en antémémoire avant que la source des informations doive être consultée à nouveau. Les valeurs zéro sont interprétées comme signifiant que le RR peut seulement être utilisé pour la transaction en cours, et ne devrait pas être mis en antémémoire. Par exemple, les enregistrements SOA sont toujours distribués avec un TTL de zéro pour interdire la mise en antémémoire. Les valeurs zéro peuvent aussi être utilisées pour des données extrêmement volatiles.
- RDLENGTH entier signé de 16 bits qui spécifie la longueur en octets du champ RDATA.
- RDATA chaîne d'octets de longueur variable qui décrit la ressource. Le format de ces informations varie selon le Type et la Classe de l'enregistrement de ressource.

3.2.2 Valeurs de Type

Les champs Type sont utilisés dans les enregistrements de ressource. Noter que ces types sont un sous-ensemble de QTYPE.

Type	valeur	signification
A	1	adresse d'hôte
NS	2	serveur de nom d'autorité
MD	3	destination de messagerie (obsolète - utiliser MX)
MF	4	transmetteur de messagerie (obsolète - utiliser MX)

CNAME	5	nom canonique pour un alias
SOA	6	marque le début d'une zone d'autorité
MB	7	nom de domaine de boîte aux lettres (expérimental)
MG	8	membre d'un groupe de messagerie (expérimental)
MR	9	nom de domaine de renommage de messagerie (expérimental)
NULL	10	RR nul (expérimental)
WKS	11	description de service bien connu
PTR	12	pointeur de nom de domaine
HINFO	13	information d'hôte
MINFO	14	information de messagerie ou de liste de messagerie
MX	15	échange de messagerie
TXT	16	chaînes de texte

3.2.3 Valeurs de QTYPE

Le champ QTYPE apparaît dans la partie question d'une interrogation. Les QTYPE sont un sur ensemble de TYPE, et donc tous les TYPE sont des QTYPE valides. De plus, on définit les QTYPE suivants :

AXFR	252	demande de transfert d'une zone entière
MAILB	253	demande d'enregistrements se rapportant à une boîte aux lettres (MB, MG ou MR)
MAILA	254	demande de RR d'agent de messagerie (obsolète - voir MX)
*	255	demande de tous les enregistrements

3.2.4 Valeurs de CLASSE

Les champs CLASSE apparaissent dans les enregistrement de ressource. Les mnémoniques et valeurs de CLASSE suivants sont définis :

IN	1	l'Internet
CS	2	classe CSNET (obsolète – utilisée seulement comme exemple dans des RFC obsolètes)
CH	3	classe CHAOS
HS	4	Hesiod [Dyer 87]

3.2.5 Valeurs de QCLASS

Les champs QCLASS apparaissent dans la section question d'une interrogation. Les valeurs de QCLASS sont un sur ensemble des valeurs de CLASSE ; chaque CLASSE est une QCLASS valide. En plus des valeurs CLASSE, la QCLASS suivante est définie :

*	255	toute classe
---	-----	--------------

3.3 RR standard

Les définitions de RR suivantes devraient survenir, au moins potentiellement, dans toutes les classes. En particulier, NS, SOA, CNAME, et PTR seront utilisés dans toutes les classes, et ont le même format dans toutes les classes. Comme leur format RDATA est connu, tous les noms de domaine dans la section RDATA de ces RR peuvent être compressés.

<nom-de-domaine> est un nom de domaine représenté par une série d'étiquettes, et terminé par une étiquette de longueur zéro. <chaîne-de-caractères> est un seul octet de longueur suivi par ce nombre de caractères. <chaîne-de-caractères> est traité comme une information binaire, et peut aller jusqu'à 256 caractères (incluant l'octet de longueur).

3.3.1 Format de CNAME RDATA

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
/                               CNAME                               /
/                               /
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

où :

CNAME est un <nom-de-domaine> qui spécifie le nom canonique ou principal pour le détenteur. Le nom du détenteur

est un alias.

Les RR CNAME ne causent pas de traitement supplémentaire de la section, mais les serveurs de noms peuvent choisir dans certains cas de redémarrer l'interrogation au nom canonique. Voir des précisions à la description de la logique de serveur de nom dans la [RFC1034].

3.3.2 Format de HINFO RDATA

```
+-----+
/                               CPU                               /
+-----+
/                               OS                               /
+-----+
```

où :

CPU est une <chaîne-de-caractères> qui spécifie le type de CPU.

OS est une <chaîne-de-caractères> qui spécifie le type de système d'exploitation.

Les valeurs standard pour CPU et OS figurent dans la [RFC1010].

Les enregistrements HINFO sont utilisés pour acquérir des informations générales sur un hôte. La principale utilisation est faite par des protocoles tels que FTP qui peuvent utiliser des procédures particulières lors de dialogues entre machines ou systèmes d'exploitation de même type.

3.3.3 Format de MB RDATA (expérimental)

```
+-----+
/                               MADNAME                          /
/                               /
+-----+
```

où :

MADNAME est un <nom-de-domaine> qui spécifie un hôte qui a la boîte aux lettres spécifiée.

Les enregistrements MB causent une traitement de section additionnelle qui recherche les RR de type A correspondants à MADNAME.

3.3.4 Format de MD RDATA (obsolète)

```
+-----+
/                               MADNAME                          /
/                               /
+-----+
```

où :

MADNAME est un <nom-de-domaine> qui spécifie un hôte qui a un agent de messagerie pour le domaine qui devrait être capable de délivrer de la messagerie pour le domaine.

Les enregistrements MD causent un traitement supplémentaire de la section qui recherche un enregistrement de type A correspondant à MADNAME.

MD est obsolète. Voir la définition de MX et la [RFC0974] pour des précisions sur le nouveau schéma. La politique recommandée pour traiter les RR MD trouvés dans un fichier maître est de les rejeter, ou de les convertir en RR MX avec une préférence de 0.

3.3.5 Format de MF RDATA (obsolète)

```
+-----+
/                               MADNAME                          /
/                               /
+-----+
```

où :

MADNAME est un <nom-de-domaine> qui spécifie un hôte qui a un agent de messagerie pour le domaine qui va accepter la messagerie à retransmettre au domaine.

Les enregistrements MF causent un traitement supplémentaire de la section qui recherche un enregistrement de type A correspondant à MADNAME.

MF est obsolète. Voir la définition de MX et la [RFC0974] pour des précisions sur le nouveau schéma. La politique recommandée pour traiter les RR MF trouvés dans un fichier maître est de les rejeter, ou de les convertir en RR MX avec une préférence de 10.

3.3.6 Format de MG RDATA (expérimental)

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                               MADNAME                               /
/                               /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  
```

où :
 MGMNAM est un <nom-de-domaine> qui spécifie une boîte aux lettres qui est membre du groupe de messagerie spécifié par le nom de domaine.

Les enregistrements MG ne causent pas de traitement supplémentaire de la section.

3.3.7 Format de MINFO RDATA (expérimental)

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                               RMAILBX                             /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                               EMAILBX                             /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  
```

où
 RMAILBX est un <nom-de-domaine> qui spécifie une boîte aux lettres qui est responsable de la liste de diffusion ou de la boîte aux lettres. Si ce nom de domaine désigne la racine, le détenteur du RR MINFO est responsable pour lui-même. Noter que de nombreuses listes de diffusion existantes utilisent une demande X de boîte aux lettres pour le champ RMAILBX de la liste de diffusion X, par exemple, Msgroup-request pour Msgroup. Ce champ procure un mécanisme plus général.

EMAILBX est un <nom-de-domaine> qui spécifie une boîte aux lettres qui va recevoir des messages d'erreur qui se rapportent à la liste de diffusion ou à la boîte aux lettres spécifiée par le détenteur du RR MINFO (comme pour le champ ERRORS-TO: qui a été proposé). Si ce nom de domaine désigne la racine, les erreurs devraient être retournées à l'expéditeur du message.

Les enregistrements MINFO ne causent pas de traitement supplémentaire de la section. Bien que ces enregistrements puissent être associés à une simple boîte aux lettres, ils sont normalement utilisés avec une liste de diffusion.

3.3.8 Format de MR RDATA (expérimental)

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                               NEWNAME                             /
/                               /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  
```

où :
 NEWNAME est un <nom-de-domaine> qui spécifie une boîte aux lettres qui est la redénomination appropriée de la boîte aux lettres spécifiée.

Les enregistrements MR ne causent pas de traitement supplémentaire de la section. La principale utilisation de MR est une entrée de transmission pour un utilisateur qui est passé à une boîte aux lettres différente.

3.3.9 Format de MX RDATA

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               PREFERENCE                           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  
```

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                               EXCHANGE                               /
/                               /                                     /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  
```

où :

PREFERENCE Un entier de 16 bits qui spécifie la préférence donnée à ce RR sur les autres chez le même propriétaire. Les valeurs les plus faibles sont les préférées.

EXCHANGE Un <nom-de-domaine> qui spécifie un hôte qui veut agir comme commutateur de messagerie pour le nom du propriétaire.

Les enregistrements MX causent un traitement de section additionnelle de type A pour l'hôte spécifié par EXCHANGE. L'utilisation des RR MX est expliquée en détails dans la [RFC0974].

3.3.10 Format de NULL RDATA (expérimental)

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                               <n'importe quoi>                               /
/                               /                                     /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  
```

Le champ RDATA peut contenir n'importe quoi du moment que cela fait au plus 65 535 octets. Les enregistrements NULL ne causent pas de traitement de section additionnelle. Les RR NULL ne sont pas permis dans les fichiers maîtres. Ils sont utilisés comme bouche trou dans certaines extensions expérimentales du DNS.

3.3.11 Format de NS RDATA

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                               NSDNAME                               /
/                               /                                     /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  
```

où :

NSDNAME <nom-de-domaine> qui spécifie un hôte qui devrait être une autorité pour la classe et le domaine spécifiés.

Les enregistrements NS causent à la fois le traitement de section additionnelle usuel pour localiser un enregistrement de type A, et, quand ils sont utilisés dans une référence, une recherche particulière des informations de "glu" dans la zone dans laquelle ils résident.

Le RR NS établit qu'on devrait s'attendre à ce que l'hôte désigné ait une zone commençant au nom du propriétaire de la classe spécifiée. Noter que la classe peut ne pas indiquer la famille de protocole qui devrait être utilisée pour communiquer avec l'hôte, bien que ce soit normalement un conseil fort. Par exemple, des hôtes qui sont des serveurs de noms pour des informations de classe Internet (IN) ou Hesiod (HS) sont normalement interrogés en utilisant des protocoles de classe IN.

3.3.12 Format de PTR RDATA

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                               PTRDNAME                               /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  
```

où :

PTRDNAME <nom-de-domaine> qui pointe sur une localisation quelconque dans l'espace des noms de domaines.

Les enregistrements PTR ne causent pas de traitement de section additionnelle. Ces RR sont utilisés dans des domaines particuliers pour pointer sur une autre localisation dans l'espace des domaines. Ces enregistrements sont des données simples, et n'impliquent aucun traitement particulier similaire à celui effectué par CNAME, qui identifie les alias. Voir par exemple la description du domaine IN-ADDR.ARPA.

3.3.13 Format de SOA RDATA

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                               MNAME                               /
/                               /                                     /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  
```


Les noms de domaines dans le domaine IN-ADDR.ARPA sont définis comme ayant jusqu'à quatre étiquettes en plus du suffixe IN-ADDR.ARPA. Chaque étiquette représente un octet d'une adresse Internet, et est exprimé par une chaîne de caractères pour une valeur décimale dans la gamme 0 à 255 (les zéros en-tête étant omis sauf dans le cas d'un octet de zéros qui est représenté par un seul zéro).

Les adresses d'hôtes sont représentées par des noms de domaines qui ont toutes leurs quatre étiquettes spécifiées. Et donc les données pour l'adresse Internet 10.2.0.52 sont localisées au nom de domaine 52.0.2.10.IN-ADDR.ARPA. L'inverse, bien que difficile à lire, permet de déléguer les zones, ce qui fait exactement un réseau d'espace d'adresse. Par exemple, 10.IN-ADDR.ARPA peut être une zone contenant des données pour ARPANET, alors que 26.IN-ADDR.ARPA peut être une zone séparée pour MILNET. Les nœuds d'adresse sont utilisés pour contenir des pointeurs sur les noms des hôtes primaires dans l'espace de domaine normal.

Les numéros de réseau correspondent à certains nœuds non terminaux à des profondeurs diverses dans le domaine IN-ADDR.ARPA, car les numéros de réseau Internet font 1, 2, ou 3 octets. Les nœuds réseau sont utilisés pour contenir des pointeurs sur les noms d'hôte primaire des passerelles attachées à ce réseau. Comme une passerelle est, par définition, sur plus d'un réseau, il y a normalement deux nœuds de réseau qui pointent sur elle. Les passerelles auront aussi des pointeurs de niveau hôte sur leurs adresses pleinement qualifiées.

Les pointeurs de passerelle aux nœuds de réseau et les pointeurs d'hôte normaux aux nœuds d'adresse pleinement qualifiée utilisent tous deux le RR PTR pour repointer sur les noms de domaine primaire des hôtes correspondants.

Par exemple, le domaine IN-ADDR.ARPA va contenir des informations sur la passerelle ISI entre les réseaux 10 et 26, une passerelle MIT du réseau 10 au réseau 18 du MIT, et les hôtes A.ISI.EDU et MULTICS.MIT.EDU. En supposant que la passerelle ISI a les adresses 10.2.0.22 et 26.0.0.103, et un nom MILNET-GW.ISI.EDU, et que la passerelle du MIT a les adresses 10.0.0.77 et 18.10.0.4 et un nom GW.LCS.MIT.EDU, la base de données du domaine contiendrait :

10.IN-ADDR.ARPA.	PTR MILNET-GW.ISI.EDU.
10.IN-ADDR.ARPA.	PTR GW.LCS.MIT.EDU.
18.IN-ADDR.ARPA.	PTR GW.LCS.MIT.EDU.
26.IN-ADDR.ARPA.	PTR MILNET-GW.ISI.EDU.
22.0.2.10.IN-ADDR.ARPA.	PTR MILNET-GW.ISI.EDU.
103.0.0.26.IN-ADDR.ARPA.	PTR MILNET-GW.ISI.EDU.
77.0.0.10.IN-ADDR.ARPA.	PTR GW.LCS.MIT.EDU.
4.0.10.18.IN-ADDR.ARPA.	PTR GW.LCS.MIT.EDU.
103.0.3.26.IN-ADDR.ARPA.	PTR A.ISI.EDU.
6.0.0.10.IN-ADDR.ARPA.	PTR MULTICS.MIT.EDU.

Et donc, un programme qui veut localiser une passerelle sur le réseau 10 va générer une interrogation de la forme QTYPE=PTR, QCLASS=IN, QNAME=10.IN-ADDR.ARPA. Il va recevoir deux RR en réponse :

10.IN-ADDR.ARPA.	PTR MILNET-GW.ISI.EDU.
10.IN-ADDR.ARPA.	PTR GW.LCS.MIT.EDU.

Le programme pourra alors générer les interrogations QTYPE=A, QCLASS=IN pour MILNET-GW.ISI.EDU. et GW.LCS.MIT.EDU. pour découvrir les adresses Internet de ces passerelles.

Un résolveur qui veut trouver le nom d'hôte correspondant à l'adresse d'hôte Internet 10.0.0.6 effectuera une interrogation de la forme QTYPE=PTR, QCLASS=IN, QNAME=6.0.0.10.IN-ADDR.ARPA, et recevra :

6.0.0.10.IN-ADDR.ARPA.	PTR MULTICS.MIT.EDU.
------------------------	----------------------

L'utilisation de ces services devra faire l'objet d'une attention particulière sur plusieurs points :

- Comme le domaine particulier IN-ADDR.ARPA et le domaine normal pour un hôte ou passerelle particulier seront dans des zones différentes, il existe une possibilité que les données ne soient pas cohérentes.
- Les passerelles auront souvent deux noms dans des domaines séparés, dont un seul peut être le principal.
- Les systèmes qui utilisent la base de données des domaines pour initialiser leurs tableaux d'acheminement doivent commencer avec suffisamment d'informations de passerelles pour garantir qu'ils peuvent accéder au serveur de noms approprié.
- Les données de passerelles ne reflètent que l'existence d'une passerelle d'une manière équivalente au fichier HOSTS.TXT actuel. Elles ne remplacent pas les informations de disponibilité dynamiques de GGP ou EGP.

3.6 Définition de nouveaux types, classes, et espaces de nom particuliers

Les types et classes précédemment définis sont ceux utilisés à la date du présent mémoire. On peut s'attendre à de nouvelles définitions. La présente section fait des recommandations aux concepteurs qui envisagent des ajouts aux facilités existantes. La liste de diffusion NAMEDROPPERS@SRI-NIC.ARPA est le forum où se tient la discussion générale sur les questions de conception.

En général, un nouveau type est approprié lorsque de nouvelles informations sont à ajouter à la base de données à propos d'un objet existant, ou qu'on a besoin de nouveaux formats de données pour un objet totalement nouveau. Les concepteurs devraient essayer de définir des types et des formats RDATA généralement applicables à toutes les classes, et qui évitent les duplications d'information. Les nouvelles classes sont appropriées lorsque le DNS doit être utilisé pour un nouveau protocole, etc., qui exige des nouveaux formats de données spécifiques d'une classe, ou lorsque on désire une copie de l'espace de noms existant, mais qu'un domaine de gestion séparé est nécessaire.

Les nouveaux types et classes ont besoin de mnémoniques pour les fichiers maîtres ; le format des fichiers maîtres exige que les mnémoniques pour le type et la classe soient disjoints.

Les valeurs de TYPE et de CLASS doivent être un sous-ensemble approprié de, respectivement, QTYPE et QCLASS.

Le système actuel utilise plusieurs RR pour représenter des valeurs multiples d'un type plutôt que de mémoriser plusieurs valeurs dans la section RDATA d'un seul RR. Ceci est moins efficace pour la plupart des applications, mais cela rend les RR plus courts. L'hypothèse de RR multiples est incorporée dans certains travaux expérimentaux sur les méthodes de mise à jour dynamique.

Le système actuel essaye de minimiser la duplication des données dans la base de données afin d'assurer la cohérence. Et donc, afin de trouver l'adresse de l'hôte pour un échange de messages, on transpose le nom de domaine en nom d'hôte, puis le nom d'hôte en adresse, plutôt que de transposer directement en adresse d'hôte. Cette approche est préférée parce qu'elle évite l'introduction d'incohérences.

Pour définir un nouveau type de données, on ne devrait pas utiliser plusieurs types de RR pour créer un ordre des entrées ou exprimer des formats différents pour des liens équivalents. Ces informations devraient plutôt être portées dans le corps du RR et on devrait utiliser un seul type. Cette politique évite les problèmes avec la mise en antémémoire de plusieurs types et la définition des QTYPE pour correspondre à plusieurs types.

Par exemple, la forme d'origine d'un lien d'échange de messagerie utilisait deux types RR, un pour représenter un échange "plus proche" (MD) et un pour représenter un échange "moins proche" (MF). La difficulté est que la présence d'un type RR dans une antémémoire n'apporte aucune information sur les autres parce que l'interrogation qui a acquis les informations de l'antémémoire pourrait avoir utilisé un QTYPE de MF, MD, ou MAILA (qui correspondent avec les deux). La nouvelle conception du service utilise un seul type (MX) avec une valeur de "préférence" dans la section RDATA qui peut ordonner des RR différents. Cependant, si des RR MX sont trouvés dans l'antémémoire, ils devraient être tous là.

4. Messages

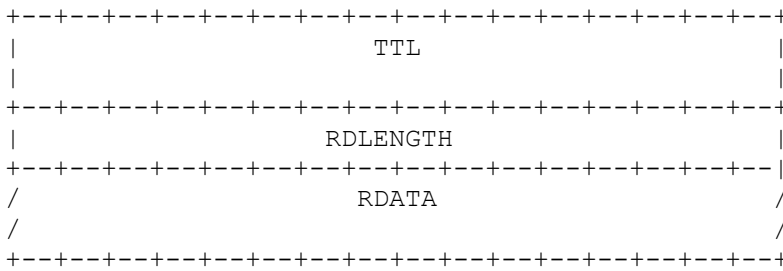
4.1 Format

Toutes les communications à l'intérieur d'un protocole de domaine sont portées dans un seul format appelé message. Le format de niveau supérieur d'un message est divisé en cinq sections (dont certaines sont vides dans certains cas) indiquées ci-dessous :

En-tête	
Question	la question posée au serveur de noms
Réponse	les RR qui répondent à la question
Autorité	les RR qui pointent vers une autorité
Additionnelle	les RR qui détiennent des informations supplémentaires

La section En-tête est toujours présente. L'en-tête inclut des champs qui spécifient lesquelles des sections restantes sont présentes, et spécifie aussi si le message est une interrogation ou une réponse, une interrogation standard ou quelque autre opcode, etc.

Le nom des sections après l'en-tête est déduit de leur utilisation dans les interrogations standard. La section Question contient des champs qui décrivent une question posée à un serveur de noms. Ces champs ont un type d'interrogation (QTYPE), une classe d'interrogation (QCLASS), et un nom de domaine d'interrogation (QNAME). Les trois dernières



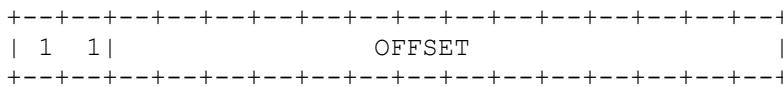
où :

- NAME nom de domaine auquel appartient cet enregistrement de ressource.
- TYPE deux octets contenant un des codes de type de RR. Ce champ spécifie la signification des données du champ RDATA.
- CLASS deux octets qui spécifient la classe des données dans le champ RDATA.
- TTL entier non signé de 32 bits qui spécifie l'intervalle de temps (en secondes) pendant lequel l'enregistrement de ressource peut rester en antémémoire avant d'être éliminé. Les valeurs zéro sont interprétées comme signifiant que le RR ne peut être utilisé que pour la transaction en cours, et ne devrait pas être mis en antémémoire.
- RDLENGTH entier non signé de 16 bits qui spécifie la longueur en octets du champ RDATA.
- RDATA chaîne d'octets de longueur variable qui décrit la ressource. Le format de ces informations varie selon le TYPE et la CLASSE de l'enregistrement de ressource. Par exemple, si le TYPE est A et la CLASSE est IN, le champ RDATA est une adresse Internet ARPA de 4 octets.

4.1.4 Compression de message

Afin de réduire la taille des messages, le système des domaines utilise un schéma de compression qui élimine la répétition des noms de domaine dans un message. Dans ce schéma, un nom de domaine entier ou une liste d'étiquettes à la fin d'un nom de domaine est remplacé par un pointeur sur une occurrence antérieure du même nom.

Le pointeur prend la forme d'une séquence de deux octets :



Les deux premiers bits sont des uns. Cela permet de distinguer un pointeur d'une étiquette, car l'étiquette doit commencer par deux bits à zéro, les étiquettes étant limitées à 63 octets ou moins. (Les combinaisons 10 et 01 sont réservées pour une utilisation future.) Le champ OFFSET spécifie un décalage par rapport au début du message (c'est-à-dire, le premier octet du champ ID dans l'en-tête de domaine). Un décalage de zéro spécifie le premier octet du champ ID, etc.

Le schéma de compression permet de représenter un nom de domaine dans un message soit comme :

- une séquence d'étiquettes se terminant par un octet de zéros
- un pointeur
- une séquence d'étiquettes se terminant par un pointeur.

Les pointeurs ne peuvent être utilisés que pour des occurrences d'un nom de domaine où le format n'est pas spécifique de la classe. Si ce n'est pas le cas, un serveur de noms ou un résolveur aurait besoin de connaître le format de tous les RR qu'il traite. Pour l'instant, ce n'est pas le cas, mais cela pourrait arriver dans des formats futurs de RDATA.

Si un nom de domaine est contenu dans une partie de message soumise à un champ de longueur (tel que la section RDATA d'un RR) et si la compression est utilisée, la longueur du nom compressé est utilisée dans le calcul de longueur, plutôt que la longueur du nom développé.

Les programmes sont libres d'éviter d'utiliser des pointeurs dans les messages qu'ils génèrent, bien que cela réduise la capacité des datagrammes, et puisse causer la troncature. Cependant il est exigé de tous les programmes qu'ils comprennent les messages d'arrivée qui contiennent des pointeurs.

Par exemple, un datagramme peut avoir besoin d'utiliser les noms de domaines F.ISI.ARPA, FOO.F.ISI.ARPA, ARPA, et la racine. En ignorant les autres champs du message, ces noms de domaine pourraient être représentés par :

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
20 |           1           |           F           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
22 |           3           |           I           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
24 |           S           |           I           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
26 |           4           |           A           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
28 |           R           |           P           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
30 |           A           |           0           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
40 |           3           |           F           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
42 |           O           |           O           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
44 | 1  1 |           20           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
64 | 1  1 |           26           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
92 |           0           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Le nom de domaine pour F.ISI.ARPA est indiqué au décalage 20. Le nom de domaine FOO.F.ISI.ARPA est indiqué au décalage 40 ; cette définition utilise un pointeur pour enchaîner une étiquette pour FOO sur le F.ISI.ARPA précédemment défini. Le nom de domaine ARPA est défini au décalage 64 en utilisant un pointeur sur le composant ARPA du nom F.ISI.ARPA à 20 ; noter que ce pointeur s'appuie sur ARPA en étant la dernière étiquette dans la chaîne à 20. Le nom de domaine racine est défini par un seul octet de zéros à 92 ; le nom de domaine racine n'a pas d'étiquette.

4.2 Transport

Le DNS suppose que les messages seront transmis comme des datagrammes ou dans un flux d'octets porté par un circuit virtuel. Bien que les circuits virtuels puissent être utilisés pour toute activité du DNS, les datagrammes sont préférés pour les interrogations du fait de leur plus faible redondance et de meilleures performances. Les activités de rafraîchissement de zone doivent utiliser des circuits virtuels à cause du besoin d'un transfert fiable.

L'Internet prend en charge l'accès au serveur de noms en utilisant TCP [RFC0793] sur l'accès serveur 53 (en décimal) ainsi que l'accès datagramme en utilisant UDP [RFC0768] sur l'accès UDP 53 (en décimal).

4.2.1 Utilisation de UDP

Les messages envoyés avec UDP utilisent l'accès serveur 53 (en décimal).

Les messages portés par UDP sont limités à 512 octets (en ne comptant pas les en-têtes IP ou UDP). Les messages plus longs sont tronqués et le bit TC est mis à un dans l'en-tête.

UDP n'est pas acceptable pour les transferts de zone, mais il est la méthode recommandée pour les interrogations standard dans l'Internet. Les interrogations qui utilisent UDP peuvent être perdues, et donc une stratégie de retransmission est nécessaire. Les interrogations ou leurs réponses peuvent être réordonnées par le réseau, ou traitées dans les serveurs de noms, aussi les résolveurs ne devraient pas dépendre d'eux pour un retour en ordre.

La politique de retransmission UDP optimale va varier selon les performances de l'Internet et les besoins du client, mais on recommande ce qui suit :

- Le client devrait essayer les autres serveurs et adresses de serveur avant de répéter une interrogation sur une adresse spécifique d'un serveur.
- L'intervalle de retransmission devrait être fondé sur des statistiques antérieures si possible. Des retransmissions trop agressives peuvent facilement ralentir les réponses pour l'ensemble de la communauté. Selon la façon dont le client est plus ou moins bien connecté à ses serveurs présumés, l'intervalle minimum de retransmission devrait être de 2 à 5 s.

On trouvera plus de suggestions sur le choix du serveur et la politique de retransmission dans la section sur le résolveur du présent mémoire.

4.2.2 Utilisation de TCP

Les messages envoyés sur les connexions TCP utilisent l'accès serveur 53 (en décimal). Le message est préfixé par un champ de longueur de deux octets qui donne la longueur du message, excluant le champ de longueur de deux octets. Ce champ de longueur permet au traitement de niveau inférieur d'assembler un message complet avant de commencer à l'analyser.

Plusieurs politiques de gestion de connexion sont recommandées :

- Le serveur ne devrait pas bloquer les autres activités en attendant les données TCP.
- Le serveur devrait accepter plusieurs connexions.
- Le serveur devrait supposer que le client va initier la clôture de la connexion, et devrait différer la clôture de son extrémité de la connexion jusqu'à ce que toutes les demandes client en cours aient été satisfaites.
- Si le serveur a besoin de fermer une connexion dormante pour des ressources réclamées, il devrait attendre jusqu'à ce que la connexion ait été inactive pendant une période de l'ordre de deux minutes. En particulier, le serveur devrait permettre de faire la séquence de demande SOA et AXFR (qui commence une opération de rafraîchissement) sur une seule connexion. Comme le serveur serait de toutes façons incapable de répondre aux interrogations, une clôture ou remise à zéro unilatérale peut être utilisée à la place d'une clôture en douceur.

5. Fichiers maîtres

Les fichiers maîtres sont des fichiers de texte qui contiennent des RR en forme textuelle. Comme le contenu d'une zone peut être exprimé sous la forme d'une liste de RR, un fichier maître est le plus souvent utilisé pour définir une zone, bien qu'il puisse être utilisé pour faire la liste du contenu d'une antémémoire. Et donc, la présente section expose d'abord le format des RR dans un fichier maître, puis des considérations particulières au cas où un fichier maître est utilisé pour créer une zone dans un serveur de nom.

5.1 Format

Le format de ces fichiers est une séquence d'entrées. Les entrées sont principalement orientées ligne, bien que des parenthèses puissent être utilisées pour continuer une liste d'éléments à travers une limite de ligne, et que le texte puisse contenir un CRLF en son sein. Toute combinaison de tabulations et d'espaces agit comme délimiteur entre les éléments séparés qui constituent une entrée. La fin de toute ligne du fichier maître peut être constituée d'un commentaire. Le commentaire débute par un ";" (point-virgule).

Les entrées suivantes sont définies :

```
<blanc>[<commentaire>]  
$ORIGIN <nom-de-domaine> [<commentaire>]  
$INCLUDE <nom-de-fichier> [<nom-de-domaine>] [<commentaire>]  
<nom-de-domaine><rr> [<commentaire>]  
<blanc><rr> [<commentaire>]
```

Les lignes blanches, avec ou sans commentaires, sont admises partout dans le fichier.

Deux entrées de commande sont définies : \$ORIGIN et \$INCLUDE. \$ORIGIN est suivi par un nom de domaine, et rétablit l'origine actuelle pour les noms de domaines relatifs au nom déclaré. \$INCLUDE insère le fichier nommé dans le fichier en cours, et peut en option spécifier un nom de domaine qui établit l'origine de nom de domaine relative pour le fichier inclus. \$INCLUDE peut aussi avoir un commentaire. Noter qu'une entrée \$INCLUDE ne change jamais l'origine relative du fichier parent, sans considération des changements à l'origine relative faits au sein du fichier inclus.

Les deux dernières formes représentent des RR. Si une entrée pour un RR commence par un blanc, le RR est alors supposé

être possédé par le dernier possesseur déclaré. Si une entrée de RR commence par un <nom-de-domaine>, le nom du possesseur est rétabli.

Le contenu <rr> prend une des formes suivantes :

```
[<TTL>] [<classe>] <type> <RDATA>
[<classe>] [<TTL>] <type> <RDATA>
```

Le RR commence par les champs TTL et Classe facultatifs, suivis par les champs Type et RDATA appropriés au type et à la classe. Classe et type utilisent les mnémoniques standard, TTL est un entier décimal. Les valeurs omises de classe et de TTL sont par défaut aux dernières valeurs explicitement déclarées. Comme les mnémoniques de type et de classe sont disjoints, l'analyse est unique. (Noter que cet ordre est différent de l'ordre utilisé dans les exemples et de l'ordre utilisé dans les RR réels ; l'ordre donné permet plus facilement l'analyse et les valeurs par défaut.)

Les <nom-de-domaine> constituent une large part des données dans le fichier maître. Les étiquettes dans le nom de domaine sont exprimées par des chaînes de caractères et sont séparées par des points. Les conventions de citation permettent de mémoriser des caractères arbitraires dans les noms de domaines. Les noms de domaines qui se terminent par un point sont appelés absolus, et sont considérés comme complets. Les noms de domaines qui ne se terminent pas par un point sont appelés relatifs ; le nom de domaine réel est l'enchaînement de la partie relative avec une origine spécifiée dans un \$ORIGIN, \$INCLUDE, ou bien un argument de la routine de chargement du fichier maître. Un nom relatif est une erreur quand une origine n'est pas disponible.

Une <chaîne-de-caractères> est exprimée de l'une des deux façons suivantes : comme un ensemble de caractères contigus sans espaces intérieures, ou comme une chaîne commençant par un " et se terminant par un ". À l'intérieur d'une chaîne délimitée par des guillemets (") peut se trouver tout caractère, sauf le guillemet (") lui-même, qui doit être cité en utilisant le caractère \ (barre oblique inverse).

Parce que ces fichiers sont des fichiers de texte, plusieurs codages spéciaux sont nécessaires pour permettre de charger des données arbitraires. En particulier :

- @ un @ (*arobase*) autonome note l'origine actuelle.
- \X où X est tout caractère autre qu'un chiffre (0-9), est utilisé pour noter ce caractère de sorte que sa signification spéciale ne s'applique pas. Par exemple, "\" peut être utilisé pour placer un caractère point dans une étiquette.
- \DDD où chaque D est un chiffre, est l'octet correspondant au nombre décimal décrit par DDD. L'octet résultant est supposé être du texte et l'analyse n'y vérifiera pas la présence d'une signification particulière.
- () Les parenthèses sont utilisées pour grouper des données qui dépassent la limite d'une ligne. En effet, les terminaisons de ligne ne sont pas reconnues lorsqu'elles sont entre parenthèses.
- ; Le point-virgule est utilisé pour débiter un commentaire ; le reste de la ligne est ignoré.

5.2 Utilisation des fichiers maîtres pour définir des zones

Lorsqu'un fichier maître est utilisé pour charger une zone, l'opération devrait être supprimée si des erreurs sont rencontrées dans le fichier maître. La raison en est qu'une seule erreur peut avoir de graves conséquences. Par exemple, supposons que les RR qui définissent une délégation aient des erreurs de syntaxe ; le serveur va alors retourner des noms d'autorité erronés pour tous les noms de la sous zone (excepté dans le cas où la sous zone est aussi présente sur le serveur).

Plusieurs autres vérifications de validité devraient être effectuées en sus de celle de la correction syntaxique du fichier :

1. Tous les RR du fichier devraient avoir la même classe.
2. Exactement un RR SOA devrait être présent au sommet de la zone.
3. Si des délégations sont présentes et si des informations de "glu" sont requises, elles devraient être présentes.
4. Les informations présentes en dehors des nœuds d'autorité dans la zone devraient être des informations de "glu", plutôt que le résultat d'une erreur d'origine ou similaire.

5.3 Exemple de fichier maître

Ci-après est un exemple de fichier qui pourrait être utilisé pour définir la zone ISI.EDU qui est chargée avec une origine de ISI.EDU :

```
@      IN      SOA      VENERA      Action\domains (
                20          ; SERIAL
                7200       ; REFRESH
                600        ; RETRY
                3600000    ; EXPIRE
                60)       ; MINIMUM
```

```
      NS  A.ISI.EDU.
      NS  VENERA
      NS  VAXA
      MX  10      VENERA
      MX  20      VAXA

A      A      26.3.0.103

VENERA  A      10.1.0.52
        A      128.9.0.32

VAXA    A      10.2.0.27
        A      128.9.0.33
```

\$INCLUDE <SUBSYS>ISI-MAILBOXES.TXT

Où le fichier <SUBSYS>ISI-MAILBOXES.TXT est :

```
MOE      MB      A.ISI.EDU.
LARRY    MB      A.ISI.EDU.
CURLEY   MB      A.ISI.EDU.
STOOGES  MG      MOE
          MG      LARRY
          MG      CURLEY
```

Noter l'utilisation du caractère \ dans le RR SOA pour spécifier la boîte aux lettres du responsable "Action.domains@ISI.EDU".

6. Mise en œuvre du serveur de noms

6.1 Architecture

La structure optimale pour le serveur de noms va dépendre du système d'exploitation de l'hôte et de l'intégration ou non du serveur de noms dans le fonctionnement d'un résolveur, soit par la prise en charge du service récurrent, soit par le partage de sa base de données avec un résolveur. La présente section expose les considérations de mise en œuvre d'un serveur de noms qui partage une base de données avec un résolveur, mais la plupart de ces problèmes sont présents dans tous les serveurs de noms.

6.1.1 Contrôle

Un serveur de noms doit employer plusieurs activités concurrentes, qu'elles soient mises en œuvre comme des tâches séparées dans le système d'exploitation de l'hôte ou multiplexées au sein d'un programme d'un seul serveur de noms. Il n'est simplement pas acceptable qu'un serveur de noms bloque le service des demandes UDP pendant qu'il attend les données TCP pour les activités de rafraîchissement ou d'interrogation. De même, un serveur de noms ne devrait pas tenter de fournir le service récurrent sans traiter de telles demandes en parallèle, bien qu'il puisse choisir de mettre en série les demandes provenant d'un seul client, ou de considérer des demandes identiques provenant du même client comme des répliques. Un serveur de noms ne devrait pas retarder substantiellement les demandes pendant qu'il recharge une zone d'après les fichiers maîtres ou pendant qu'il incorpore une zone qui vient d'être rafraîchie dans sa base de données.

6.1.2 Base de données

Alors que les mises en œuvre de serveur de noms ont toute liberté pour utiliser les structures de données internes de leur choix, la structure suggérée consiste en trois parties majeures :

- Un "catalogue" des structures de données qui fait la liste des zones disponibles de ce serveur, et un "pointeur" sur la structure des données de la zone. Le principal objet de cette structure est de trouver la plus proche zone ancêtre, s'il en est, pour les interrogations standard entrantes.
- Des structures de données séparées pour chacune des zones détenues par le serveur de noms.
- Une structure de données pour les données en antémémoire. (Ou peut-être des antémémoires séparées pour les différentes classes.)

Toutes ces structures de données peuvent être mises en œuvre dans un format de structure arborescente identique, avec des données différentes chaînées aux nœuds dans les différentes parties : dans le catalogue les données sont des pointeurs sur les zones, tandis que dans les structures de données de zone et d'antémémoire, les données seront des RR. En concevant le cadre d'arborescence, le concepteur devrait intégrer que le processus d'interrogation doit traverser l'arbre en utilisant des comparaisons d'étiquettes insensibles à la casse ; et dans les données réelles, quelques nœuds ont un coefficient d'embranchements extrêmement élevé (de 100 à 1000 ou plus) mais la grande majorité a un coefficient d'embranchements très faible (0 ou 1).

Une façon de résoudre le problème de la casse est de mémoriser en deux morceaux les étiquettes de chaque nœud : une représentation en casse normalisée des étiquettes dans laquelle tous les caractères ASCII sont dans une seule casse, avec un gabarit binaire qui note quels caractères sont en fait dans une casse différente. La diversité du coefficient d'embranchements peut être maîtrisée à l'aide d'une liste avec un lien simple sur un nœud jusqu'à ce que le coefficient d'embranchements dépasse un certain seuil, et de passer à une structure de hachage après le franchissement du seuil. Dans tous les cas, les structures de hachage qui sont utilisées pour mémoriser les sections d'arborescence doivent garantir que ces fonctions et procédures de hachage préservent les conventions de casse du DNS.

L'utilisation de structures séparées pour les différentes parties de la base de données est motivée par plusieurs facteurs :

- La structure de catalogue peut être une structure presque statique qui n'a besoin d'être changée que lorsque l'administrateur du système change les zones prises en charge par le serveur. Cette structure peut aussi être utilisée pour mémoriser des paramètres utilisés pour commander les activités de rafraîchissement.
- Les structures de données individuelles pour les zones permettent de remplacer une zone avec un simple changement de pointeur dans le catalogue. Les opérations de rafraîchissement de zone peuvent construire une nouvelle structure et, quand elle est achevée, l'introduire dans la base de données via un simple remplacement de pointeur. Il est très important que lors du rafraîchissement d'une zone, les interrogations n'utilisent pas simultanément les anciennes et les nouvelles données.
- Avec les procédures de recherche appropriées, les données d'autorisation dans les zones seront toujours "cachées", et vont donc toujours prendre la préséance sur les données en antémémoire.
- Les erreurs de définitions de zone qui causent des chevauchements de zones, etc., peuvent causer des réponses erronées aux interrogations, mais les problèmes de détermination sont simplifiés, et le contenu d'une "mauvaise" zone ne peut pas en corrompre une autre.
- Comme l'antémémoire est très fréquemment mise à jour, elle est très vulnérable à la corruption durant les réamorçages de système. Elle peut aussi être pleine de données de RR arrivés à expiration. Dans l'un et l'autre cas, elle peut facilement être éliminée sans perturber les données de zone.

Un aspect majeur de la conception de la base de données est de choisir une structure qui permet au serveur de noms de faire face aux défaillances de l'hôte du serveur de noms. Les informations d'état que le serveur de noms devrait sauvegarder lors des défaillances système incluent la structure du catalogue (y compris l'état de rafraîchissement pour chaque zone) et les données de zone elles-mêmes.

6.1.3 Heure

Les données de TTL pour les RR et les données de temporisation pour les activités de rafraîchissement dépendent des temporisateurs à 32 bits en unités de secondes. À l'intérieur de la base de données, les temporisateurs de rafraîchissement et les TTL pour les données en antémémoire font un "décompte" conceptuel, alors que les données dans la zone restent avec des TTL constants.

Une stratégie de mise en œuvre recommandée est de mémoriser l'heure de deux façons : comme un incrément relatif et comme un temps absolu. Une façon de le faire est d'utiliser les nombres positifs de 32 bits pour un type et les nombres négatifs pour l'autre. Les RR dans les zones utilisent les temps relatifs ; les temporisateurs de rafraîchissement et les données d'antémémoire utilisent des temps absolus. Les nombres absolus sont pris par rapport à une origine connue et convertis en valeurs relatives lorsqu'ils sont placés dans la réponse à une interrogation. Quand un TTL absolu est négatif après conversion en relatif, les données sont arrivées à expiration et devraient être ignorées.

6.2 Processus d'interrogation standard

Le principal algorithme pour le traitement d'interrogation standard est présenté dans la [RFC1034].

Lors du traitement des interrogations avec QCLASS=*, ou quelque autre QCLASS qui correspond à plusieurs classes, la réponse ne devrait jamais être d'autorité sauf si le serveur peut garantir que la réponse couvre toutes les classes.

Lors de la composition d'une réponse, les RR qui sont à insérer dans la section additionnelle, mais dupliquent les RR dans la réponse ou les sections d'autorité, peuvent être omis de la section additionnelle.

Lorsqu'une réponse est si longue qu'il est nécessaire de la tronquer, la troncature devrait commencer à la fin de la réponse et se poursuivre dans le datagramme. Et donc, si il y a des données pour la section d'autorité, la section de réponse est à coup sûr unique.

La valeur MINIMUM dans le SOA devrait être utilisée pour établir un plancher du TTL des données distribuées à partir d'une zone. Cette fonction plancher devrait être faite lorsque les données sont copiées dans une réponse. Cela permettra que des protocoles futurs de mise à jour dynamique changent le champ MINIMUM du SOA sans sémantique ambiguë.

6.3 **Rafraîchissement de zone et processus de rechargement**

En dépit des meilleurs efforts d'un serveur, il peut se trouver incapable de charger les données de zone à partir d'un fichier maître du fait d'erreurs de syntaxe, etc., ou être incapable de rafraîchir une zone dans les délais de son paramètre d'expiration. Dans ce cas, le serveur de noms devrait répondre aux interrogations comme s'il n'était pas supposé posséder la zone.

Si un fichier maître envoie une zone via AXFR, et si une nouvelle version est créée durant le transfert, le fichier maître devrait continuer si possible l'envoi de l'ancienne version. Dans tous les cas, il ne devrait jamais envoyer une partie d'une version et une partie de l'autre. Si il ne lui est pas possible d'achever, le fichier maître devrait redémarrer la connexion sur laquelle le transfert de zone a lieu.

6.4 **Interrogations inverses (facultatif)**

Les interrogations inverses sont une partie facultative du DNS. Les serveurs de noms ne sont pas obligés de prendre en charge une forme quelconque d'interrogations inverses. Si un serveur de noms reçoit une interrogation inverse qu'il ne prend pas en charge, il retourne une réponse d'erreur avec l'erreur "Non mis en œuvre" mise dans l'en-tête. Bien que la prise en charge de l'interrogation inverse soit facultative, tous les serveurs de noms doivent être au moins capables de retourner la réponse d'erreur.

6.4.1 **Contenu des questions et réponses inverses**

Les interrogations inverses renversent la transposition effectuée par les opérations d'interrogation standard ; alors qu'une interrogation standard transpose un nom de domaine en ressource, une interrogation inverse transpose une ressource en nom de domaine. Par exemple, une interrogation standard peut lier un nom de domaine à une adresse d'hôte ; l'interrogation inverse correspondante lie l'adresse d'hôte à un nom de domaine.

Les interrogations inverses prennent la forme d'un seul RR dans la section Réponse du message, avec une section Question vide. Le possesseur du nom du RR d'interrogation et son TTL ne sont pas significatifs. La réponse porte les questions dans la section Question qui identifie tous les noms qui possèdent le RR d'interrogation QUE CONNAIT LE SERVEUR DE NOMS. Comme aucun serveur de noms ne connaît tout l'espace des noms de domaines, il ne peut jamais être garanti que la réponse sera complète. Et donc, les interrogations inverses sont principalement utiles pour la gestion de base de données et les activités de débogage. Les interrogations inverses NE SONT PAS une méthode acceptable pour la transposition des adresses d'hôte en nom d'hôte ; il faut à la place utiliser le domaine IN-ADDR.ARPA.

Lorsque c'est possible, les serveurs de noms devraient fournir des comparaisons insensibles à la casse pour les interrogations inverses. Et donc, une interrogation inverse qui demande un RR MX de "Venera.isi.edu" devrait obtenir la même réponse qu'une interrogation pour "VENERA.ISI.EDU" ; une interrogation pour HINFO RR "IBM-PC UNIX" devrait produire le même résultat qu'une interrogation inverse pour "IBM-pc unix". Cependant, ceci ne peut pas être garanti parce que les serveurs de noms peuvent posséder des RR qui contiennent des chaînes de caractères mais le serveur de noms ne sait pas que les données sont des caractères.

Lorsque un serveur de noms traite une interrogation inverse, il retourne :

1. zéro, un, ou plusieurs noms de domaine pour la ressource spécifiée, comme des QNAME dans la section Question,
2. un code d'erreur indiquant que le serveur de noms ne prend pas en charge la transposition inverse du type de ressource spécifiée.

Lorsque la réponse à une interrogation inverse contient un ou plusieurs QNAME, le nom du possesseur et le TTL du RR dans la section Réponse qui définit l'interrogation inverse sont modifiés pour correspondre exactement à un RR trouvé au premier QNAME.

Les RR retournés dans les interrogations inverses ne peuvent pas être mis en antémémoire en utilisant le même mécanisme qu'utilisé pour les réponses aux interrogations standard. Une raison pour cela est qu'un nom peut avoir plusieurs RR du

même type, et seulement l'un d'eux va apparaître. Par exemple, une interrogation inverse pour une seule adresse d'un hôte à rattachements multiples peut donner l'impression qu'il n'existait qu'une adresse.

6.4.2 Exemple d'interrogation et réponse inverse

La structure globale d'une interrogation inverse pour restituer le nom de domaine qui correspond à l'adresse Internet 10.1.0.52 est donnée ci-dessous :

En-tête	OPCODE=IQUERY, ID=997
Question	<vide>
Réponse	<tout-nom> A IN 10.1.0.52
Autorité	<vide>
Additionnelle	<vide>

Cette interrogation pose une question dont la réponse est l'adresse de style Internet 10.1.0.52. Comme le nom du possesseur n'est pas connu, tout nom de domaine peut être utilisé comme bouche-trou (et être ignoré). Un seul octet de zéro, signifiant la racine, est normalement utilisé parce qu'il minimise la longueur du message. Le TTL du RR n'est pas significatif. La réponse à cette interrogation pourrait être :

En-tête	OPCODE=IQUERY, ID=997, QR=1
Question	QTYPE=A, QCLASS=IN, QNAME=VENERA.ISI.EDU
Réponse	VENERA.ISI.EDU A IN 10.1.0.52
Autorité	<vide>
Additionnelle	<vide>

(corrigé conformément à l'errata du 9/02/2003)

Noter que le QTYPE dans une réponse à une interrogation inverse est le même que le champ TYPE dans la section Réponse de l'interrogation inverse. Les réponses aux interrogations inverses peuvent contenir plusieurs questions lorsque l'inverse n'est pas unique. Si la section Question de la réponse n'est pas vide, le RR dans la section Réponse est alors modifié pour correspondre à une copie exacte d'un RR au premier QNAME.

6.4.3 Processus de l'interrogation inverse

Les serveurs de noms qui prennent en charge l'interrogation inverse peuvent accepter ces opérations à travers des recherches exhaustives dans leurs bases de données, mais cela devient impraticable lorsque la taille de la base de données augmente. Une autre approche est d'inverser la base de données conformément à la clé de recherche.

Pour les serveurs de noms qui prennent en charge plusieurs zones et une grande quantité de données, l'approche recommandée est d'effectuer des inversions séparées pour chaque zone. Lorsqu'une zone particulière est changée durant un rafraîchissement, seules ses inversions ont besoin d'être effectuées.

La prise en charge de transferts de ce type d'inversion pourra être incluse dans de futures versions du système de domaines, mais n'est pas acceptée dans la présente version.

6.5 Achèvement des interrogations et des réponses

Les services d'achèvement facultatifs décrits dans la RFC0882 et la RFC0883 ont été supprimés. Une nouvelle conception de ces services pourrait devenir disponible à l'avenir.

7. Mise en œuvre du résolveur

Les niveaux supérieurs de l'algorithme de résolveur recommandé sont exposés dans la [RFC1034]. La présente section expose les détails de mise en œuvre en supposant la structure de base de données suggérée dans la section de mise en œuvre de serveur de noms du présent mémoire.

7.1 Transformation d'une demande d'utilisateur en une interrogation

La première étape du travail d'un résolveur est de transformer la demande du client, établie dans un format convenable pour le système d'exploitation local, en une spécification de recherche des RR à un nom spécifique qui correspond à un QTYPE et une QCLASS spécifiques. Lorsque c'est possible, le QTYPE et la QCLASS devraient correspondre à un seul type et une

seule classe, parce que cela rend l'utilisation des données en antémémoire beaucoup plus simple. La raison en est que la présence de données d'un seul type dans une antémémoire ne confirme pas l'existence ou la non existence de données d'autres types, et donc la seule façon d'être sûr est de consulter une source d'autorité. Si QCLASS=* est utilisé, les ressources d'autorité ne seront pas disponibles.

Comme un résolveur doit être capable de multiplexer plusieurs demandes si il veut effectuer efficacement sa fonction, chaque demande en cours est habituellement représentée dans un bloc d'informations d'état. Ce bloc d'état va normalement contenir :

- Un horodatage qui indique l'heure de début de la demande.
L'horodatage est utilisé pour décider si les RR qui sont dans la base de données peuvent être utilisés ou sont périmés. Cet horodatage utilise le format de temps absolu précédemment exposé pour la mémorisation de RR dans les zones et les antémémoires. Noter que quand un TTL de RR indique une heure relative, le RR doit être à l'heure, car il fait partie d'une zone. Lorsque le RR a une heure absolue, il fait partie d'une antémémoire, et le TTL du RR est comparé à l'horodatage pour le début de la demande.

Noter que l'utilisation de l'horodatage est supérieure à l'utilisation de l'heure en cours, car cela permet d'entrer les RR avec des TTL de zéro dans l'antémémoire de la façon habituelle, mais d'utiliser encore la demande en cours, même après des intervalles de nombreuses secondes du fait de la charge du système, de fins de temporisation de retransmission d'interrogations, etc.

- Certaines sortes de paramètres pour limiter la quantité de travail à effectuer pour cette demande.
La quantité de travail que va effectuer un résolveur en réponse à une demande d'un client doit être limitée pour se garder contre les erreurs de la base de données, comme des références CNAME circulaires, et des problèmes de fonctionnement, tels qu'une partition de réseau qui empêche le résolveur d'accéder aux serveurs de noms dont il a besoin. Alors que les limites locales sur le nombre de fois qu'un résolveur va retransmettre une interrogation particulière à une adresse de serveur de noms particulier sont essentielles, le résolveur devrait avoir un compteur global par demande pour limiter le travail sur une seule demande. Le compteur devrait être réglé à une valeur initiale et être décrémenté chaque fois que le résolveur effectue une action (fin de temporisation de retransmission, retransmission, etc.) Si le compteur passe à zéro, la demande est terminée avec une erreur temporaire.

Noter que si la structure du résolveur permet de commencer une demande en parallèle avec d'autres, comme lorsque le besoin d'accéder à un serveur de noms pour une demande cause une résolution parallèle pour les adresses du serveur de noms, la demande en cause devrait être commencée avec un compteur plus faible. Cela empêche les références circulaires dans la base de données de commencer une réaction en chaîne d'activité de résolveur.

- La structure de données SLIST exposée dans la [RFC1034].
Cette structure garde trace de l'état d'une demande si elle doit attendre des réponses de serveurs de noms étrangers.

7.2 Envoi des interrogations

Comme décrit dans la [RFC1034], la tâche de base du résolveur est de formuler une interrogation qui va répondre à la demande du client et diriger cette interrogation sur les serveurs de noms qui peuvent fournir les informations. Le résolveur aura normalement seulement de très fortes indications sur les serveurs à interroger, sous la forme de RR NS, et peut avoir à réviser l'interrogation, en réponse aux CNAME, ou à réviser l'ensemble des serveurs de noms que le résolveur interroge, en réponse aux réponses de délégation qui pointent le résolveur sur les serveurs de noms plus proches des informations désirées. En plus des informations demandées par le client, le résolveur peut avoir à appeler ses propres services pour déterminer l'adresse des serveurs de noms qu'il souhaite contacter.

Dans tous les cas, le modèle utilisé dans le présent mémoire suppose que le résolveur partage son attention entre plusieurs demandes, certaines provenant du client, et certaines générées en interne. Chaque demande est représentée par des informations d'état, et le comportement désiré est que le résolveur transmette les interrogations aux serveurs de noms d'une façon qui maximise la probabilité que la demande reçoive une réponse, minimise le temps que prend la demande, et évite des transmissions excessives. L'algorithme clé utilise les informations d'état de la demande pour choisir l'adresse du prochain serveur de noms à interroger, et calcule aussi une temporisation qui va causer l'action suivante si une réponse n'arrive pas. La prochaine action sera habituellement une transmission à quelque autre serveur, mais peut être une erreur temporaire envoyée au client.

Le résolveur commence toujours par une liste des serveurs de noms à interroger (SLIST). Cette liste sera celle de tous les RR NS qui correspondent à la plus proche zone ancêtre dont le résolveur a connaissance. Pour éviter les problèmes de démarrage, le résolveur devrait avoir un ensemble de serveurs par défaut qu'il interrogera s'il devait n'avoir aucun RR NS en cours approprié. Le résolveur ajoute alors à la SLIST toutes les adresses connues de serveurs de noms, et peut

commencer des demandes parallèles pour acquérir les adresses des serveurs quand le résolveur a le nom, mais pas d'adresse, pour les serveurs de noms.

Pour achever l'initialisation de SLIST, le résolveur joint toutes les informations d'historique qu'il possède quelles qu'elles soient à chaque adresse de la SLIST. Cela va habituellement consister en une sorte de moyenne pondérée sur le temps de réponse de l'adresse, et la moyenne des scores de l'adresse (c'est-à-dire, combien de fois l'adresse a répondu à toutes les demandes). Noter que ces informations devraient rester adresse par adresse, plutôt que par serveur de noms, parce que le temps de réponse et le score moyen d'un serveur particulier peut varier considérablement d'une adresse à l'autre. Noter aussi que ces informations sont réellement spécifiques d'une paire adresse de résolveur / adresse de serveur, de sorte qu'un résolveur avec plusieurs adresses peut souhaiter garder des historiques distincts pour chacune de ses adresses. Une partie de cette étape doit tenir compte des adresses qui n'ont pas un tel historique ; dans ce cas, un délai d'aller-retour attendu de 5 à 10 secondes devrait être le pire cas, avec des estimations plus faibles pour le même réseau local, etc.

Noter que chaque fois qu'une délégation est suivie, l'algorithme de résolveur réinitialise SLIST.

Les informations établissent un classement partiel des adresses de serveur de noms disponibles. Chaque fois qu'une adresse est choisie et que l'état devrait être altéré pour empêcher sa sélection à nouveau jusqu'à ce que toutes les autres adresses aient été essayées. La fin de temporisation pour chaque transmission devrait être 50 à 100 % supérieure à la valeur moyenne prédite pour permettre la variance des réponses.

Quelques points délicats :

- Le résolveur peut rencontrer une situation où aucune adresse n'est disponible pour aucun des serveurs de noms désignés dans la SLIST, et où les serveurs de la liste sont précisément ceux qui auraient normalement été utilisés pour rechercher leurs propres adresses. Cette situation survient normalement lorsque les RR d'adresse glu ont un plus petit TTL que les RR NS qui marquent la délégation, ou quand le résolveur met en antémémoire le résultat d'une recherche NS. Le résolveur devrait détecter cette condition et redémarrer la recherche à la prochaine zone ancêtre, ou autrement, à la racine.
- Si un résolveur obtient une erreur de serveur ou une autre réponse bizarre de la part d'un serveur de noms, il devrait le retirer de la SLIST, et peut souhaiter programmer une transmission immédiate à la prochaine adresse de candidat serveur.

7.3 Traitement des réponses

La première étape du traitement des datagrammes de réponse entrants est d'analyser la réponse. Cette procédure devrait inclure :

- De vérifier la cohérence des en-têtes. Éliminer les datagrammes qui sont des interrogations quand on attend des réponses.
- D'analyser les sections du message, et s'assurer que tous les RR sont formatés correctement.
- Étape facultative, vérifier les TTL des données entrantes à la recherche de RR avec des TTL excessivement longs. Si un RR a un TTL excessivement long, disons supérieur à une semaine, éliminer la réponse toute entière, ou limiter tous les TTL à une semaine dans la réponse.

L'étape suivante est de confronter la réponse à une demande de résolveur en cours. La stratégie recommandée est de faire une confrontation préliminaire à l'aide du champ ID dans l'en-tête de domaine, puis de vérifier que la section Question correspond aux informations actuellement désirées. Cela exige que l'algorithme de transmission consacre plusieurs bits du champ ID de domaine à une sorte d'identifiant de requête. Cette étape comporte plusieurs points délicats :

- Certains serveurs de noms envoient leurs réponses à partir d'adresses différentes de celle utilisée pour recevoir l'interrogation. C'est à dire qu'un résolveur ne peut pas s'appuyer sur l'idée qu'une réponse viendra de la même adresse que celle où il a envoyé l'interrogation correspondante. Ce défaut des serveurs de noms est typique des systèmes UNIX.
- Si le résolveur retransmet une demande particulière à un serveur de noms, il devrait être capable d'utiliser une réponse à partir de n'importe laquelle des transmissions. Cependant, si il utilise la réponse pour échantillonner le délai d'aller-retour de l'accès au serveur de noms, il doit être capable de déterminer quelle transmission correspond à la réponse (et de conserver les temps de transmission pour chaque message sortant) ou seulement de calculer les délais d'aller-retour sur la base des transmissions initiales.

- De temps en temps, un serveur de noms n'aura pas de copie en cours d'une zone alors qu'il devrait l'avoir conformément à certains RR NS. Le résolveur devrait simplement retirer le serveur de noms de la SLIST en cours, et continuer.

7.4 Utilisation de l'antémémoire

En général, on s'attend à ce qu'un résolveur mette en antémémoire toutes les données qu'il reçoit en réponse car elles peuvent lui être utiles pour répondre à de futures demandes de clients. Cependant, il y a plusieurs types de données qui ne devraient pas être mises en antémémoire :

- Lorsque plusieurs RR du même type sont disponibles pour un nom de propriétaire particulier, le résolveur devrait tous les mettre en antémémoire ou aucun. Lorsque une réponse est tronquée, et qu'un résolveur ne sait pas si il a un ensemble complet, il ne devrait pas mettre en antémémoire un ensemble de RR qui pourrait n'être que partiel.
- Les données en antémémoire ne devraient jamais être utilisées de préférence à des données d'autorité, aussi si la mise en antémémoire devait causer un tel événement, il ne faut pas mettre les données en antémémoire.
- Les résultats d'une interrogation inverse ne devraient pas être mis en antémémoire.
- Les résultats des interrogations standard où le QNAME contient des étiquettes "*" si les données pourraient être utilisées pour construire des enregistrements génériques. La raison en est que l'antémémoire ne contient pas nécessairement les informations des RR existants ou de frontière de zone qui sont nécessaires pour contenir l'application de RR génériques.
- Les données de RR dans des réponses de fiabilité douteuse. Lorsque un résolveur reçoit des réponses non sollicitées ou des données de RR autres que celles qu'il a demandé, il devrait les éliminer sans les mettre en antémémoire. L'implication de base est que toutes les vérifications de bonne santé sur un paquet devraient être effectuées avant toute mise en antémémoire.

Dans une veine similaire, lorsqu'un résolveur a un ensemble de RR pour un certain nom dans une réponse, et qu'il veut mettre les RR en antémémoire, il devrait vérifier les RR déjà existants dans l'antémémoire. Selon les circonstances, il peut préférer les données de la réponse ou de l'antémémoire, mais les deux ne devraient jamais être combinées. Si les données de la réponse sont des données d'autorité dans la section réponse, elles seront toujours préférées.

8. Prise en charge de la messagerie

Le système des domaines définit une norme pour la transposition des boîtes aux lettres en noms de domaines, et deux méthodes pour utiliser les informations de boîte aux lettres pour déduire les informations d'acheminement de la messagerie. La première méthode est appelée lien d'échange de messagerie et l'autre méthode lien de boîte aux lettres. La norme de codage de boîte aux lettres et le lien d'échange de messagerie font partie du protocole officiel du DNS, et sont les méthodes recommandées pour l'acheminement de la messagerie dans l'Internet. Le lien de boîte aux lettres est un dispositif expérimental qui est encore en cours de développement et sujet à variations.

La norme de codage de boîte aux lettres suppose un nom de boîte aux lettres de la forme "<partie-locale>@<domaine-de-messagerie>". Alors que la syntaxe permise dans chacune de ces sections varie de façon substantielle entre les diverses messageries internet, la syntaxe préférée pour l'Internet ARPA est donnée dans la [RFC0822].

Le DNS code la <partie-locale> comme une seule étiquette, et code le <domaine-de-messagerie> comme un nom de domaine. La seule étiquette provenant de la <partie-locale> est préfacée au nom de domaine provenant du <domaine-de-messagerie> pour former le nom de domaine correspondant à la boîte aux lettres. Et donc la boîte aux lettres HOSTMASTER@SRI-NIC.ARPA est transposée en le nom de domaine HOSTMASTER.SRI-NIC.ARPA. Si la <partie-locale> contient des points ou autres caractères spéciaux, sa représentation dans un fichier maître exigera l'utilisation d'un marquage avec une barre oblique inverse pour s'assurer que le nom de domaine est correctement codé. Par exemple, la boîte aux lettres Action.domains@ISI.EDU serait représentée par Action\domains.ISI.EDU.

8.1 Liens d'échange de messagerie

Le lien d'échange de messagerie utilise la partie <domaine-de-messagerie> d'une spécification de boîte aux lettres pour déterminer où devraient être envoyés les messages. La <partie-locale> n'est même pas consultée. La [RFC0974] spécifie cette méthode en détail, et devrait être consultée avant de tenter d'utiliser la prise en charge de l'échange de messagerie.

Un des avantages de cette méthode est qu'elle découple la dénomination de la destination de la messagerie des hôtes utilisés pour prendre en charge le service de messagerie, au prix d'une autre couche d'indirection dans la fonction de recherche. Cependant, cette couche additionnelle devrait éliminer le besoin des codages "%", "!", etc. compliqués dans la <partie-locale>.

L'essence de la méthode est que le <domaine-de-messagerie> est utilisé comme nom de domaine pour localiser les RR de type MX qui font la liste des hôtes qui acceptent les messages pour le <domaine-de-messagerie>, avec des valeurs de préférence qui rangent les hôtes selon un ordre spécifié par les administrateurs du <domaine-de-messagerie>.

Dans le présent mémoire, le <domaine-de-messagerie> ISI.EDU est utilisé dans les exemples, conjointement avec les hôtes VENERA.ISI.EDU et VAXA.ISI.EDU comme échanges de messages pour ISI.EDU. Si un expéditeur de messagerie a un message pour Mockapetris@ISI.EDU, il l'acheminerait en cherchant des RR MX pour ISI.EDU. Les RR MX au nom ISI.EDU VENERA.ISI.EDU et VAXA.ISI.EDU, et les interrogations de type A peuvent trouver les adresses d'hôte.

8.2 Lien de messagerie (expérimental)

Dans le lien de boîte aux lettres, l'expéditeur de message utilise toute la spécification de destination de messagerie pour construire un nom de domaine. Le nom de domaine codé pour la boîte aux lettres est utilisé comme champ de QNAME dans une interrogation QTYPE=MAILB.

Plusieurs résultats sont possibles pour cette interrogation :

1. L'interrogation peut retourner une erreur de nom indiquant que la boîte aux lettres n'existe pas comme nom de domaine. À long terme, cela indiquerait que la boîte aux lettres spécifiée n'existe pas. Cependant, tant que l'utilisation du lien de boîte aux lettres est universel, cette condition d'erreur devrait être interprétée comme signifiant que l'organisation identifiée par la partie globale ne prend pas en charge le lien de boîte aux lettres. La procédure appropriée est de revenir au lien d'échange à ce point.
2. L'interrogation peut retourner un RR Mail Rename (MR, *changement de dénomination de boîte aux lettres*). Le RR MR porte une nouvelle spécification de boîte aux lettres dans son champ RDATA. L'expéditeur de messages devrait remplacer l'ancienne boîte aux lettres par la nouvelle et réessayer l'opération.
3. L'interrogation peut retourner un RR MB. Le RR MB porte un nom de domaine pour un hôte dans son champ RDATA. L'expéditeur de messages devrait délivrer le message à cet hôte via tout protocole applicable, par exemple, par SMTP.
4. L'interrogation peut retourner un ou plusieurs RR Mail Group (MG, *groupe de boîte aux lettres*). Cette condition signifie que la boîte aux lettres est en réalité une liste de diffusion ou un groupe de boîtes aux lettres, plutôt qu'une seule boîte aux lettres. Chaque RR MG a un champ RDATA qui identifie une boîte aux lettres qui est membre du groupe. l'expéditeur de messages devrait délivrer une copie du message à chaque membre.
5. L'interrogation peut retourner un RR MB aussi bien qu'un ou plusieurs RR MG. Cette condition signifie que la boîte aux lettres est en réalité une liste de diffusion. L'expéditeur de messages peut délivrer le message à l'hôte spécifié par le RR MB, qui va à son tour le délivrer à tous les membres, ou il peut utiliser les RR MG pour faire la diffusion lui-même.

Dans chacun de ces cas, la réponse peut inclure un RR Mail Information (MINFO, *informations de messagerie*). Ce RR est normalement associé à un groupe de boîte aux lettres, mais est légal avec un MB. Le RR MINFO identifie deux boîtes aux lettres. Une d'elles identifie un responsable du nom original de la boîte aux lettres. Cette boîte aux lettres devrait être utilisée pour des demandes à ajouter à un groupe de boîtes aux lettres, etc. Le second nom de boîte aux lettres dans le RR MINFO identifie une boîte aux lettres qui devrait recevoir des messages d'erreur pour les défaillances de la messagerie. Ceci est particulièrement approprié pour les listes de diffusion lorsque des erreurs de noms des membres devraient être rapportées à une personne autre que celle qui envoie un message à la liste.

De nouveaux champs pourront être ajoutés à ce RR à l'avenir.

9. Références et bibliographie

- [Dyer 87] Dyer, S., and F. Hsu, "Hesiod, Project Athena Technical Plan - Name Service", avril 1987, version 1.9. Décrit les fondements du service de noms Hesiod.
- [IEN-116] J. Postel, "Internet Name Server", IEN-116, USC/Information Sciences Institute, août 1979. Service de noms rendu obsolète par le Système de noms de domaines, mais toujours utilisé.
- [Quarterman 86] Quarterman, J., and J. Hoskins, "Notable Computer Networks", Communications of the ACM, octobre 1986, volume 29, numéro 10.
- [RFC0742] K. Harrenstien, "NAME/FINGER", décembre 1977. (*Obsolète, voir la RFC1288*)
- [RFC0768] J. Postel, "Protocole de [datagramme d'utilisateur](#) (UDP)", (STD 6), 28 août 1980.
- [RFC0793] J. Postel (éd.), "Protocole de [commande de transmission](#) – Spécification du protocole du programme Internet DARPA", STD 7, septembre 1981.
- [RFC0799] D. Mills, "Domaines de noms Internet", septembre 1981.
- [RFC0805] J. Postel, "Notes de réunion de la messagerie informatique", février 1982.
- [RFC0810] E. Feinler et autres, "Spécification du tableau des hôtes Internet du DOD", mars 1982. *Obsolète, voir RFC0952.*
- [RFC0811] K. Harrenstien, V. White et E. Feinler, "Serveur des noms d'hôtes", mars 1982. *Obsolète, voir la RFC0953.*
- [RFC0812] K. Harrenstien et V. White, "Surnoms/Qui est qui", mars 1982.
- [RFC0819] Z. Su et J. Postel, "[Convention de nommage](#) des domaines pour les applications d'utilisateurs de l'Internet", août 1982. Premières réflexions sur la conception du système des domaines. La mise en œuvre actuelle est complètement différente.
- [RFC0821] J. Postel, "Protocole simple de [transfert de messagerie](#)", STD 10, août 1982.
- [RFC0830] Z. Su, "Système distribué pour le [service de noms Internet](#)", octobre 1982. Premières réflexions sur la conception du système des domaines. La mise en œuvre actuelle est complètement différente.
- [RFC0882] P. Mockapetris, "Noms de domaines - Concepts et facilités", novembre 1983. Remplacé par le présent mémoire.
- [RFC0883] P. Mockapetris, "Noms de domaines – Spécification et mise en œuvre", (*obsolète, voir RFC1034/1035*), novembre 1983. Remplacée par le présent mémoire.
- [RFC0920] J. Postel et J. Reynolds, "Exigences pour les domaines", octobre 1984. Explique le schéma de nommage pour les domaines de niveau supérieur.
- [RFC0952] K. Harrenstien, M. Stahl, E. Feinler, "Spécification du tableau des hôtes de l'Internet du DOD", octobre 1985. Spécifie le format de HOSTS.TXT, le tableau des hôtes/adresses remplacé par le DNS.
- [RFC0953] K. Harrenstien, M. Stahl, E. Feinler, "Serveur HOSTNAME", octobre 1985. (*Historique*). Cette RFC contient la spécification officielle du protocole de serveur de nom d'hôte, qui est rendue obsolète par le DNS. Ce protocole fondé sur TCP donne accès à des informations mémorisées dans le format de la RFC 952, et est utilisé pour obtenir des copies du tableau des hôtes.
- [RFC0953] K. Harrenstien, M. Stahl, E. Feinler, "Serveur HOSTNAME", octobre 1985. (*Historique*). Décrit les changements aux RFC 882 et 883 leurs raisons. Maintenant obsolète.
- [RFC0974] C. Partridge, "L'acheminement de la messagerie et le système des domaines", janvier 1986. (*obsolète, voir la RFC2821*) Décrit la transition de l'adressage de messagerie fondé sur HOSTS.TXT au plus puissant système MX utilisé avec le système des domaines.
- [RFC1001] "Protocole standard pour un [service NetBIOS sur un transport TCP/UDP](#) : concepts et méthodes", STD 19,

mars 1987. Cette RFC et la RFC 1002 sont un dessin préliminaire de NETBIOS par dessus TCP/IP qui propose de placer le service de noms NetBIOS par dessus le DNS. (STD 19)

- [RFC1002] "Protocole standard pour un service NetBIOS sur un transport TCP/UDP : [Spécifications détaillées](#)", STD 19, mars 1987.
- [RFC1010] J. Reynolds et J. Postel, "Numéros alloués", mai 1987. (*Historique, voir www.iana.org*) Contient les numéros des prises et les mnémoniques pour les noms des hôtes, des systèmes d'exploitation, etc.
- [RFC1031] W. Lazear, "Transition du domaine des noms MILNET", novembre 1987. Décrit un plan pour convertir le MILNET en DNS.
- [RFC1032] M. K. Stahl, "[Établissement d'un domaine](#) - lignes directrices pour les administrateurs", novembre 1987. Décrit les politiques d'enregistrement utilisées par le NIC pour administrer les domaines de niveau supérieur et déléguer les sous zones.
- [RFC1033] M. K. Lottor, "[Guide de fonctionnement](#) de l'administrateur de domaine", novembre 1987. Livre de recettes pour les administrateurs de domaine.
- [Solomon 82] M. Solomon, L. Landweber, et D. Neuhengen, "Le serveur de noms CSNET", Computer Networks, vol 6, n°3, juillet 1982. Décrit un service de noms pour CSNET indépendant du DNS et l'utilisation du DNS dans CSNET.
- (La présente RFC est mise à jour par les [RFC1101](#), [RFC1183](#), [RFC1348](#), [RFC1876](#), [RFC1982](#), [RFC1995](#), [RFC1996](#), [RFC2065](#), [RFC2136](#), [RFC2181](#), [RFC2137](#), [RFC2308](#), [RFC2535](#), [RFC2673](#), [RFC2845](#), [RFC3425](#), [RFC3658](#), [RFC4033](#), [RFC4034](#), [RFC4035](#), [RFC4343](#), [RFC5936](#), [RFC5966](#), [RFC6604](#), [RFC7766](#))