

域名 - 实现和规范

1. 本备忘录的状态

本 RFC 描述了域名系统的细节和协议, 并假定读者熟悉配套 RFC 《域名 - 概念和功能》[RFC-1034] 中讨论的概念。

域名系统是一系列官方协议的函数和数据类型, 以及一系列仍在实验中的函数和数据类型。由于域名系统有意设计为可扩展的, 因此官方协议之外的系统部分应始终预期有新的数据类型和实验性行为。官方协议部分包括标准查询、响应和互联网类 RR 数据格式(例如, 主机地址)。由于之前的 RFC 集, 一些定义已经更改, 因此一些之前的定义已过时。

实验性或过时的功能在这些 RFC 中明确标记, 并且应谨慎使用此类信息。

特别提醒读者不要依赖示例中出现的值来判断其是否当前或完整, 因为其主要目的是教学性的。本备忘录的分发不受限制。

目录

1. 本备忘录的状态	1
2. 引言	3
2.1. 概述	3
2.2. Common configurations	4
2.3. Conventions	7
2.3.1. Preferred name syntax	7
2.3.2. Data Transmission Order	8
2.3.3. Character Case	9
2.3.4. Size limits	10
3. 域名空间和 RR 定义	10
3.1. 命名空间定义	10
3.2. RR 定义	11
3.2.1. 格式	11
3.2.2. TYPE values	12
3.2.3. QTYPE values	12
3.2.4. CLASS values	13

3.2.5. QCLASS values	13
3.3. Standard RRs	13
3.3.1. CNAME RDATA format	14
3.3.2. HINFO RDATA format	14
3.3.3. MB RDATA format (EXPERIMENTAL)	14
3.3.4. MD RDATA format (Obsolete)	15
3.3.5. MF RDATA format (Obsolete)	15
3.3.6. MG RDATA format (EXPERIMENTAL)	16
3.3.7. MINFO RDATA format (EXPERIMENTAL)	16
3.3.8. MR RDATA format (EXPERIMENTAL)	17
3.3.9. MX RDATA format	17
3.3.10. NULL RDATA format (EXPERIMENTAL)	17
3.3.11. NS RDATA format	18
3.3.12. PTR RDATA format	18
3.3.13. SOA RDATA format	19
3.3.14. TXT RDATA format	20
3.4. ARPA Internet specific RRs	20
3.4.1. A RDATA format	20
3.4.2. WKS RDATA format	21
3.5. IN-ADDR.ARPA domain	22
3.6. 定义新的类型、类和特殊命名空间	24
4. 消息	25
4.1. 格式	25
4.1.1. Header section format	26
4.1.2. Question section format	28
4.1.3. Resource record format	29
4.1.4. Message compression	30
4.2. 传输	32
4.2.1. UDP 使用	32
4.2.2. TCP 使用	32
5. 主文件	33
5.1. 格式	33
5.2. 使用主文件定义区域	35
5.3. 主文件示例	36
6. 名称服务器实现	37
6.1. 架构	37
6.1.1. 控制	37
6.1.2. 数据库	37
6.1.3. 时间	39
6.2. 标准查询处理	39
6.3. 区域刷新和重新加载处理	39
6.4. 反向查询 (可选)	39
6.4.1. 反向查询和响应的内容	40
6.4.2. 反向查询和响应示例	40
6.4.3. 反向查询处理	41
	42

6.5. 完成查询和响应 .42	
7. 解析器实现	43
7.1. 转换使用查询	43
7.2. 发送查询 .44	43
7.3. 处理响应 .46	43
7.4. 使用缓存 .	47
8. 邮件支持	47
8.2. 邮箱绑定（实验性） .48	47
9. 参考文献和书目 .50	
索引 .54	

2. 引言

2.1. 概述

域名的目的是提供一种命名资源的机制，以便这些名称可以在不同的主机、网络、协议族、互联网和管理组织中使用。

从用户的角度来看，域名作为本地代理（称为解析器）的参数非常有用，该代理检索与域名关联的信息。因此，用户可能会请求与特定域名关联的主机地址或邮件信息。为了使用户能够请求特定类型的信息，需要将适当的查询类型与域名一起传递给解析器。对用户而言，域名树是一个单一的信息空间；解析器负责向用户隐藏数据在名称服务器之间的分布。

从解析器的角度来看，构成域空间的数据库分布在各种名称服务器中。域空间的不同部分存储在不同的名称服务器中，尽管特定的数据项将冗余地存储在两个或多个名称服务器中。解析器最初至少知道一个名称服务器。当解析器处理用户查询时，它会向一个已知的名称服务器请求信息；作为回报，解析器要么接收所需信息，要么收到另一个名称服务器的引用。使用这些引用，解析器学习其他名称服务器的身份和内容。解析器负责处理域空间的分布，并通过咨询其他服务器中的冗余数据库来处理名称服务器故障的影响。

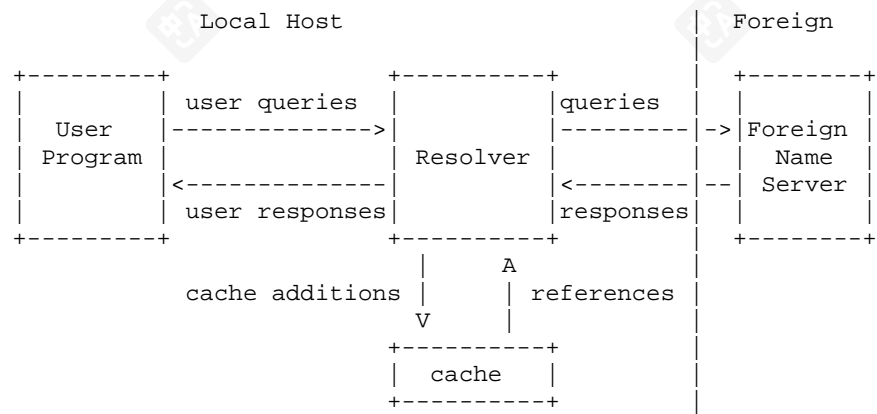
Name servers manage two kinds of data. The first kind of data held in sets called zones; each zone is the complete database for a particular "pruned" subtree of the domain space. This data is called authoritative. A name server periodically checks to make sure that its zones are up to date, and if not, obtains a new copy of updated zones

从本地存储的主文件或另一个名称服务器的主文件中获取。第二种数据是本地解析器获取的缓存数据。这些数据可能不完整，但在非本地数据反复访问时可以提高检索过程的性能。缓存数据最终会被超时机制丢弃。

这种功能结构将用户界面、故障恢复和分发的问题隔离在解析器中，将数据库更新和刷新问题隔离在名称服务器中。

2.2. 常见配置

主机可以通过多种方式参与域名系统，具体取决于主机是否运行从域名系统检索信息的程序、回答其他主机查询的名称服务器，或这两种功能的各种组合。最简单、也许最典型的配置如下：

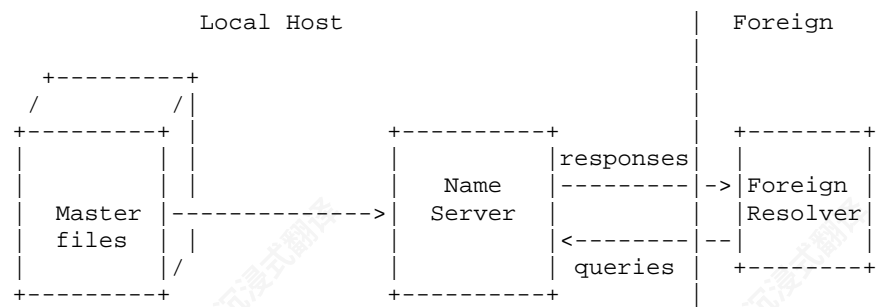


用户程序通过解析器与域名空间交互；用户查询和用户响应的格式特定于主机及其操作系统。用户查询通常是操作系统调用，解析器及其缓存将是主机操作系统的一部分。功能较弱的主机可以选择将解析器实现为子程序，以便链接到每个需要其服务的程序中。解析器通过向外部名称服务器和本地缓存发起查询来响应用户查询。

请注意，解析器可能需要向多个不同的外部名称服务器发起多个查询来回答特定的用户查询，因此用户查询的解析可能涉及多个网络访问和任意的时间量。向外部名称服务器的查询及其相应的响应具有标准格式，该格式已描述。

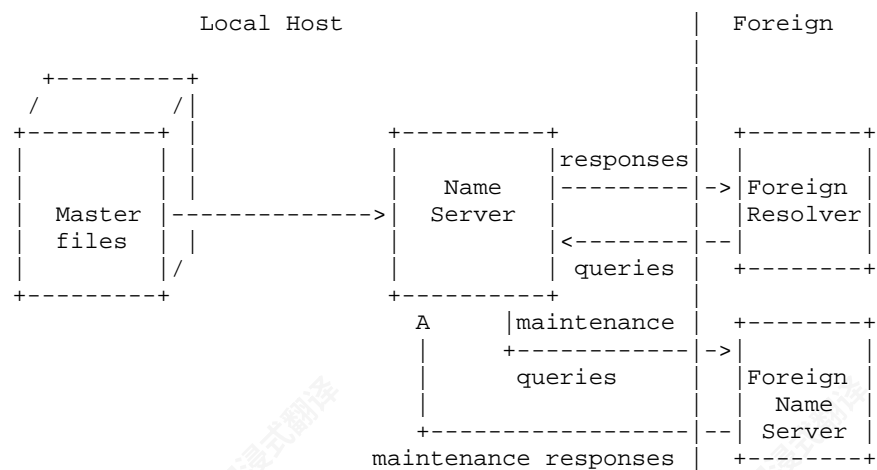
在本备忘录中，并且可能是数据报。

根据其功能，名称服务器可以是一个专用机器上的独立程序，或者是一个大型分时主机上的一个或多个进程。一个简单的配置可能是：



在这里，主名称服务器通过读取其本地文件系统的主文件来获取一个或多个区域的信息，并应答来自外部解析器关于这些区域的查询。

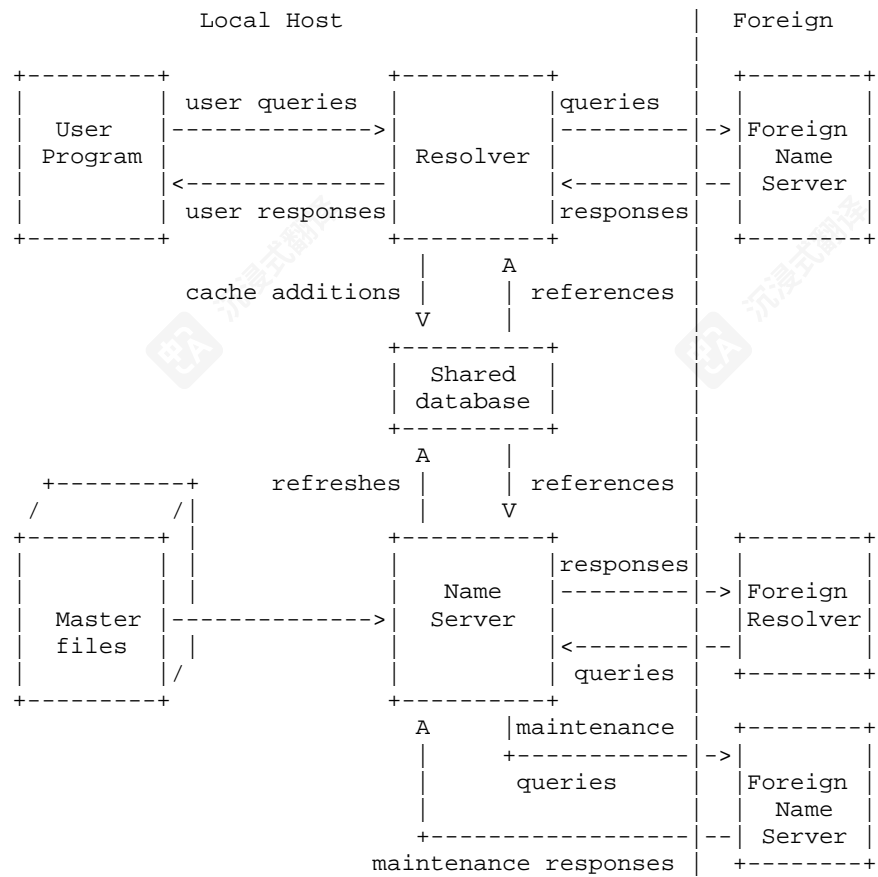
DNS 要求所有区域都由多个名称服务器冗余支持。指定辅助服务器可以使用 DNS 的区域传输协议获取区域并从主要服务器检查更新。此配置如下所示：



在此配置中，名称服务器会定期向外部名称服务器建立虚拟电路，以获取区域副本或检查现有副本是否未更改。发送的消息用于

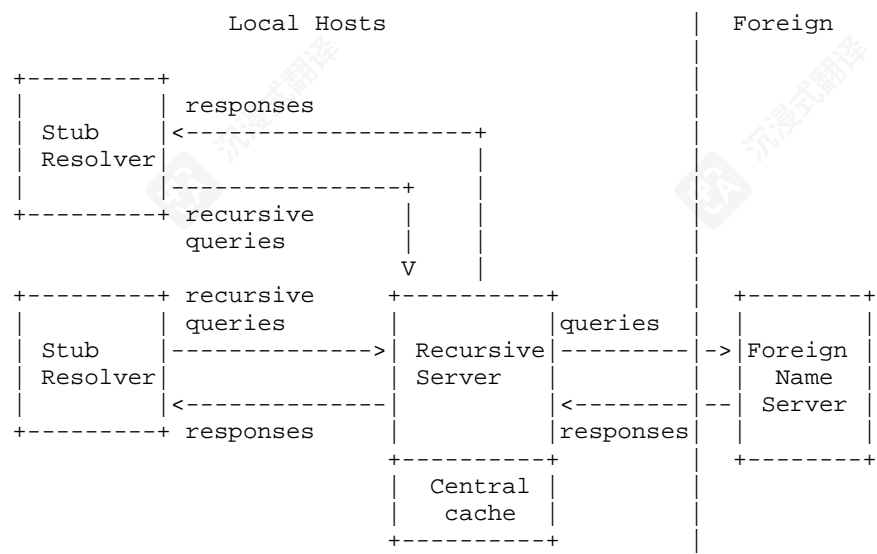
这些维护活动与查询和响应具有相同的形式，但消息序列略有不同。

The information flow in a host that supports all aspects of the domain name system is shown below:



共享数据库包含本地名称服务器和解析器的域空间数据。共享数据库的内容通常是名称服务器定期刷新操作维护的权威数据以及来自先前解析器请求的缓存数据。域数据的结构以及名称服务器和解析器之间同步的必要性意味着此数据库的一般特性，但实际格式由本地实现者决定。

信息流也可以定制，以便一组主机协同工作以优化活动。有时这是为了将较不强大的主机卸载，以便它们不必实现完整的解析器。这对于个人电脑或希望最小化所需新网络代码的主机来说是合适的。此方案还可以允许一组主机共享少量缓存，而不是维护大量单独的缓存，前提是集中式缓存具有更高的命中率。在任何情况下，解析器都被替换为 Stub 解析器，它们作为递归服务器中位于一个或多个已知执行该服务的名称服务器中的解析器的前端：



在任何情况下，请注意，只要有可能，域名组件总是为了可靠性而复制。

2.3. 规范

域名系统有一些关于低级但根本性问题的约定。虽然实现者可以在自己的系统内部违反这些约定，但他必须在所有从其他主机观察到的行为中遵守这些约定。

2.3.1. 推荐的名称语法

DNS 规范在构造域名规则方面尽可能通用。其思想是，任何现有对象的名称都可以通过最小的更改表示为域名。

然而，在为对象分配域名时，谨慎的用户会选择一个满足域名系统规则且符合对象现有规则（无论这些规则是否已发布或由现有程序隐含）的名称。

例如，在命名邮件域名时，用户应同时满足本备忘录的规则和 RFC-822 中的规则。在创建新主机名时，应遵循 HOSTS.TXT 的旧规则。这避免了在旧软件转换为使用域名时出现的问题。

以下语法在许多情况下会导致较少的问题。

使用域名的应用程序（例如，邮件、TELNET）。

< 域名 > ::= < 子域名 > | ""

< 子域名 > ::= < 标签 > | < 子域名 > "." < 标签 >

< label > ::= < letter > [[< ldh-str >] < let-dig >]

< ldh-str > ::= < let-dig-hyp > | < let-dig-hyp > < ldh-str >

< let-dig-hyp > ::= < let-dig > | "-"

< let-dig > ::= < 字母 > | < 数字 >

< 字母 > ::= 52 个字母字符中的任意一个，大写为 A 到 Z，小写为 a 到 z

< 数字 > ::= 10 个数字中的任意一个，0 到 9

注意，虽然域名中允许大小写字母，但大小写不区分。也就是说，拼写相同但大小写不同的两个名称应被视为相同。

标签必须遵循 ARPANET 主机名的规则。它们必须以字母开头，以字母或数字结尾，中间只能包含字母、数字和连字符。还有一些关于长度的限制。标签必须不超过 63 个字符。

例如，以下字符串标识互联网中的主机：

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

2.3.2. 数据传输顺序

本文件中描述的头部和数据传输顺序在八位字节级别上得到解决。每当一个图示显示一个

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

```

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

```

2.3.3. 字符大小写

当数据进入域名系统时，应尽可能保留其原始大小写。在某些情况下，这无法做到。例如，如果两个 RRs 存储在数据库中，一个存储在 x.y，另一个存储在 X.Y，它们实际上存储在数据库中的同一位置，因此只有一种大小写会被保留。基本规则是，只有当数据用于在数据库中定义结构时，大小写才可以被丢弃，并且当以不区分大小写的方式比较时，两个名称是相同的。

大小写敏感数据丢失必须最小化。因此，虽然 x.y 和 X.Y 的数据可以存储在同一个位置 x.y 或 X.Y 下，但 a.x 和 B.X 的数据永远不会存储在 A.x、A.X、b.x 或 b.X 下。通常，这保留了域名首标签的大小写，但强制内部节点标签的标准化。

将数据输入域名数据库的系统管理员应注意，如果他们的系统是大小写敏感的，应以大小写一致的方式向域名系统提供数据。域名系统中的数据分发系统将确保一致表示得以保留。

2.3.4. 大小限制

DNS 中的各种对象和参数都有大小限制。它们列在下面。有些可以轻易更改，而另一些则更为根本。

标签	63 个字节或更少
名称	255 个字节或更少
TTL	positive values of a signed 32 bit number.
UDP 消息	512 字节或更少

3. 域名空间和 RR 定义

3.1. 命名空间定义

消息中的域名以标签序列表示。每个标签表示为一个长度字段，后跟相应数量的字节。由于每个域名都以根的空标签结尾，域名以长度为零的字节终止。每个长度字节的最高两位必须为零，长度字段的其余六位将标签限制为 63 字节或更少。

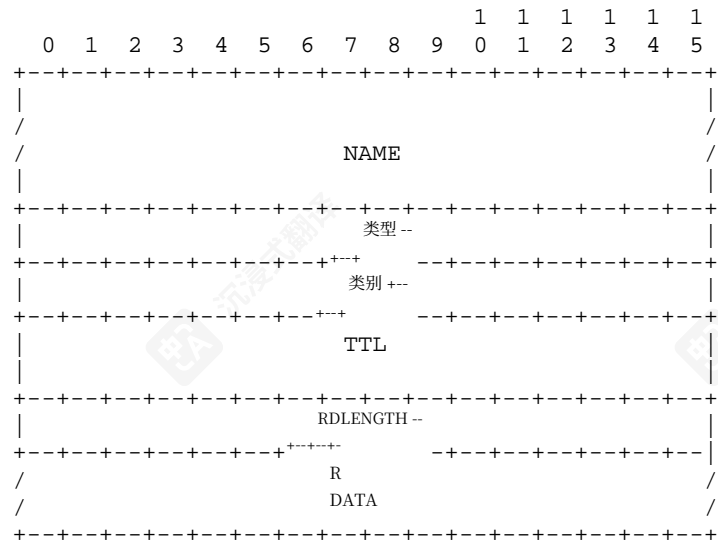
为简化实现，域名的总长度（即标签字节和标签长度字节）被限制为 255 字节或更少。

尽管标签可以包含构成标签的任何八位字节中的 8 位值，但强烈建议标签遵循备忘录中描述的推荐语法，该语法与现有的主机命名约定兼容。名称服务器和解析器必须以不区分大小写的方式比较标签（即 A=a），假设 ASCII 无奇偶校验。非字母代码必须完全匹配。

3.2. RR 定义

3.2.1. 格式

所有 RR 都具有以下所示相同的一级格式:



其中:

- NAME

an owner name, i.e., the name of the node to which this 资源记录适用。
- TYPE

包含一个 RR 类型代码的两个八位字节。
- 类别

包含一个 RR 类别代码的两个八位字节。
- TTL

一个 32 位有符号整数, 用于指定时间间隔
资源记录在被缓存之前, 信息源
应再次被查询。零
值被解释为仅 RR 可以
用于进行中的事务, 不应
缓存。例如, SOA 记录始终带有零 TTL 以禁止缓存。
零值可以
也可用于极不稳定的数据。
- RDLENGTH

一个无符号 16 位整数, 用于指定长度, 单位为
RDATA 字段的字节。

RDATA 一个可变长度的字节串，用于描述资源。此信息的格式各不相同资源。此信息的格式各不相同
根据资源记录的 TYPE 和 CLASS。

3.2.2. TYPE 值

类型字段用于资源记录。注意这些类型是 QTYPEs 的子集。

TYPE	值和含义
A	1 主机地址
NS	2 权威域名服务器
MD	3 一个邮件目的地（过时的 - 使用 MX）
MF	4 一个邮件转发器（过时的 - 使用 MX）
CNAME	5 别名的规范名称
SOA	6 标记权威区域的开始
MB	7 一个邮箱域名（实验性）
MG	8 一个邮件组成员（实验性）
MR	9 一个邮件重命名域名（实验性）
NULL	10 一个空 RR（实验性）
WKS	11 一个知名服务描述
PTR	12 一个域名指针
HINFO	13 主机信息
MINFO	14 mailbox or mail list information
MX	15 邮件交换
TXT	16 文本字符串

3.2.3. QTYPE values

QTYPE 字段出现在查询的问题部分。QTYPES 是 TYPEs 的超集，因此所有 TYPEs 都是有效的 QTYPEs。此外，定义了以下 QTYPEs：

AXFR	252 A request for a transfer of an entire zone
MAILB	253 A request for mailbox-related records (MB, MG or MR)
MAILA	254 请求邮件代理 RR (已过时 - 请参阅 MX)
*	255 请求所有记录

3.2.4. 类值

类字段出现在资源记录中。以下定义了类助记符和值：

IN	1 互联网	
CS	2 CSNET 类 (过时) 一些过时的 RFC 文件)	- 仅用于示例中
CH	3 the CHAOS 类别	
HS	4 Hesiod [Dyer 87]	

3.2.5. QCLASS 值

QCLASS 类字段出现在查询的问题部分中。QCLASS 值是 CLASS 值的超集；每个 CLASS 都是有效的 QCLASS。除了 CLASS 值外，还定义了以下 QCLASSes：

*	255 任何类别
---	----------

3.3. 标准 RR

以下 RR 定义预期将在所有类别中至少潜在地出现。特别是，NS、SOA、CNAME 和 PTR 将在所有类别中使用，并且在所有类别中具有相同的格式。由于它们的 RDATA 格式已知，这些 RR 的 RDATA 部分中的所有域名都可能被压缩。

< 域名 > 表示为一系列标签，并以零长度的标签终止。< 字符串 > 是一个长度八位组，后跟该数量的字符。< 字符串 > 被视为二进制信息，长度最多可达 256 个字符（包括长度八位组）。

3.3.1. CNAME RDATA 格式

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                                         CNA                               /
/                                         ME                               /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

其中:

CNAME A <域名> , 它指定了规范名称或主要名称
 用于所有者的名称。所有者名称是一个别名。

CNAME 记录不会导致附加部分处理, 但名称服务器可以选择在某些情况下在规范名称处重新开始查询。有关详细信息, 请参阅 [RFC-1034] 中关于名称服务器逻辑的描述。

3.3.2. HINFO RDATA 格式

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                                         CPU --                               /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                                         操                               /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
                                         作系统
                                         +---

```

其中:

CPU A <character-string> which specifies the CPU type.

OS A <character-string> which specifies the operating
 system type.

CPU 和操作系统的标准值可以在 [RFC-1010] 中找到。

HINFO 记录用于获取主机的通用信息。主要用途是用于 FTP 等协议, 这些协议在机器之间或相同类型的操作系统之间通信时可以使用特殊程序。

3.3.3. MB RDATA 格式 (实验性)

```

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

```

在哪里:

MADNAME A <domain-name> which specifies a host which has the
 specified mailbox.

MB 记录导致附加节处理，该处理会查找与 MADNAME 对应的 A 类型 RRs。

3.3.4. MD RDATA 格式 (过时的)

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

在:

MADNAME 一个 < 域名 > , 它指定了一个拥有该域邮件代理的主机。
该域的邮件代理应该能够为该域传递邮件。

MD 记录会导致附加节处理，该处理会查找与 MADNAME 对应的 A 类型记录。

MD 已过时。有关新方案详情，请参阅 MX 的定义和 [RFC-974]。处理主文件中发现的 MD 记录的推荐策略是拒绝它们，或将其转换为优先级为 0 的 MX 记录。

3.3.5. MF RDATA 格式 (过时的)

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

其中:

MADNAME A < 域名 > , 它指定了一个拥有邮件代理的主机，该代理将接受邮件并转发到该域名。
该邮件代理将接受邮件并转发到该域名。

MF 记录导致附加节处理，该处理会查找与 MADNAME 对应的 A 类型记录。

MF 已过时。有关新方案详情，请参阅 MX 的定义和 [RFC-974]。处理主文件中发现的 MD 记录的推荐策略是拒绝它们，或将其转换为优先级为 10 的 MX 记录。

3.3.6. MG RDATA 格式 (实验性)

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                                     MGMNAME                               /
/                                     /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

其中:

MGMNAME A <domain-name> which specifies a mailbox which is a member of the mail group specified by the domain name.

MG 记录不会导致附加节处理。

3.3.7. MINFO RDATA 格式 (实验性)

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                                     RMAILBX +---+                               /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                                     E                               /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                                     MAILBX +---+---+                               /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

where:

RMAILBX 一个 < 域名 > , 它指定了一个邮箱, 该邮箱负责邮件列表或邮箱。如果它负责自己。注意, 许多现有的邮件列表使用邮箱 X-request 作为邮件列表 X 的 RMAILBX 字段的值, 例如, Msgroup-request 用于 Msgroup。这个字段提供了一种更通用的机制。

EMAILBX 一个 < 域名 > , 它指定了一个要接收与邮件列表相关的错误消息由 MINFO 记录的所有者指定的邮箱 (类似) 到已建议的 ERRORS-TO: 字段。此域名指代根, 错误应为返回消息的发送者。 If

MINFO 记录不会导致附加节处理。尽管这些记录可以与一个简单邮箱关联, 但它们通常用于邮件列表。

3.3.8. MR RDATA 格式 (实验性)

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                               NE                               /
/                               WNAME                           /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

where:

NEWNAME 一个 < 域名 > , 它指定了一个邮箱
 指定邮箱的正确重命名。

MR 记录不会导致附加节处理。MR 的主要用途是为已迁移到不同邮箱的用户作为转发条目。

3.3.9. MX RDATA 格式

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               优先级 +---+---+---+---+---+---+
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                               交换                               /
/                               /                               /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

其中:

优先级 A 16 位整数, 用于指定此记录在相同所有者下的优先级。较小的值优先。

EXCHANGE A < 域名 > , 用于指定一个愿意作为所有者名称邮件交换的主机。

MX 记录导致对 EXCHANGE 指定的主机进行类型 A 附加节处理。MX 记录的使用在 [RFC-974] 中有详细说明。

3.3.10. NULL RDATA 格式 (实验性)

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                               <anything>                       /
/                               /                               /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

RDATA 字段中可以包含任何内容, 只要其长度不超过 65535 个八位字节。

NULL 记录不会导致附加节处理。NULL RRs 不允许在主文件中使用。NULL 用于 DNS 的一些实验性扩展中的占位符。

3.3.11. NS RDATA 格式

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

其中:

NSDNAME A <domain-name> which specifies a host which should be authoritative for the specified class and domain.

NS 记录会导致常规的附加节处理以定位 A 记录，并在用于引用时，对它们所在的区域进行特殊的粘合信息搜索。

NS RR 指出，命名主机应预期具有一个从指定类所有者名称开始的区域。注意，类别可能不会指示与主机通信时应使用的协议族，尽管它通常是强烈的提示。例如，作为互联网（IN）或 Hesiod（HS）类别信息名称服务器的主机通常使用 IN 类协议进行查询。

3.3.12. PTR RDATA 格式

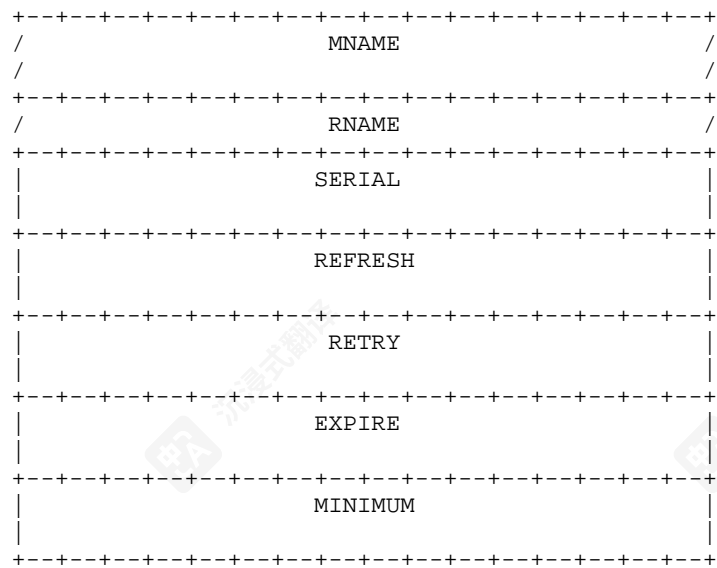
```
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                               P                               /
/                               /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
                               PTRDNAME +---+---+---+---+---+---+
      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

其中:

PTRDNAME A < 域名 > ，它指向域名空间中的某个位置。

PTR 记录不会导致附加节处理。这些 RR 用于特殊域，以指向域名空间中的其他位置。这些记录是简单数据，并不像 CNAME 那样标识别名所进行的特殊处理。有关示例，请参阅 IN-ADDR.ARPA 域的描述。

3.3.13. SOA RDATA 格式



where:

MNAME	该名称服务器的 < 域名 > 是 此区域原始或主要数据源。
RNAME	一个 < 域名 > ， 用于指定此区域负责人的 邮箱。
SERIAL	The unsigned 32 bit version number of the original copy 该区域。区域传输保留此值。This 值会回绕，应使用序列空间 算术。
刷新	区域之前应有一个 32 位时间间隔 刷新。
重试	一个 32 位时间间隔，在失败的刷新之前应该经过 应该重试。
过期	A 32 bit time value that specifies the upper limit on 区域在失效之前可以经过的时间间隔 更长的权威的。

最小值 The unsigned 32 bit minimum TTL field that should be exported with any RR from this zone.

SOA 记录不会导致附加节处理。

所有时间都以秒为单位。

这些字段中的大多数仅与名称服务器维护操作相关。然而，最小值在所有从区域检索 RR 的查询操作中使用。每当 RR 被发送到查询的响应中时，TTL 字段被设置为 RR 的 TTL 字段和相应 SOA 中的最小值中的较大值。因此，最小值是区域中所有 RR 的 TTL 字段的下限。请注意，这种使用最小值的情况应在 RR 复制到响应中时发生，而不是在从主文件或通过区域传输加载区域时发生。这种规定的目的是允许未来的动态更新功能以已知的语义更改 SOA RR。

3.3.14. TXT RDATA 格式

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/                               TXT-DATA                               /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

其中：

TXT-DATA One or more <character-string>s.

TXT RR 用于存储描述性文本。文本的语义取决于它被发现的域。

3.4. 互联网特定 RR

3.4.1. A RDATA 格式

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               ADDRESS                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

where:

地址 A 32 bit Internet address.

拥有多个互联网地址的主机将拥有多个 A 记录。

A 记录不会导致附加节处理。主文件中 A 行的 RDATA 部分是一个互联网地址，由四个用点分隔的十进制数组成，且不包含嵌入空格（例如，“10.2.0.52”或“192.0.5.6”）。

3.4.2. WKS RDATA 格式

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     |
|               ADDRESS               |
|                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               PROTOCOL               |
|                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     |
|               <BIT MAP>              |
|                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

where:

ADDRESS An 32 bit Internet address

PROTOCOL An 8 bit IP protocol number

<BIT MAP> A variable length bit map. The bit map must be a multiple of 8 bits long.

WKS 记录用于描述特定协议在特定互联网地址上支持的服务。PROTOCOL 字段指定一个 IP 协议号，位图对指定协议的每个端口有一个位。第一个位对应端口 0，第二个位对应端口 1，以此类推。如果位图不包含感兴趣的协议的位，则该位被视为零。端口和协议的适当值和助记符在 [RFC-1010] 中指定。

例如，如果 PROTOCOL=TCP（6），第 26 位对应 TCP 端口 25（SMTP）。如果该位被设置，SMTP 服务器应该在 TCP 端口 25 上监听；如果为零，则指定地址不支持 SMTP 服务。

WKS RRs 的目的是为 TCP 和 UDP 的服务器提供可用性信息。如果一个服务器支持 TCP 和 UDP，或者有多个互联网地址，则使用多个 WKS RRs。

WKS 记录不会导致附加节处理。

在主文件中，端口和协议都使用助记符或十进制数表示。

3.5. IN-ADDR.ARPA 域

互联网使用一个特殊域来支持网关定位和互联网地址到主机映射。其他类别可能也在其他域中采用类似策略。此域的目的是提供一种保证的方法来执行主机地址到主机名映射，并促进查询以定位互联网中特定网络上的所有网关。

请注意，这两种服务都与反向查询可以执行的功能相似；区别在于这部分域名空间是按地址结构化的，因此可以保证无需对整个域名空间进行穷举搜索即可找到适当的数据。

该域从 IN-ADDR.ARPA 开始，并具有以下子结构
互联网地址结构。

IN-ADDR.ARPA 域中的域名被定义为除了 IN-ADDR.ARPA 后缀外最多包含四个标签。每个标签代表互联网地址的一个八位字节，并以表示 0-255 范围内十进制值的字符串形式表达（除零八位字节外，其余情况省略前导零，零八位字节用单个零表示）。

主机地址由包含所有四个标签的域名表示。因此，互联网地址 10.2.0.52 的数据位于域名 52.0.2.10.IN-ADDR.ARPA。这种反转虽然阅读不便，但允许将区域委托给正好是一个地址空间的网络。例如，10.IN-ADDR.ARPA 可以是一个包含 ARPANET 数据的区域，而 26.IN-ADDR.ARPA 可以是 MILNET 的独立区域。地址节点用于在普通域空间中存储指向主要主机名的指针。

网络编号对应于 IN-ADDR.ARPA 域中不同深度的某些非终端节点，因为互联网网络编号是 1、2 或 3 个字节。网络节点用于存储指向连接到该网络的网关的主要主机名的指针。由于网关根据定义位于一个以上的网络上，因此它通常有两个或多个指向它的网络节点。网关在其完全限定地址上还会有主机级别的指针。

网络节点中的网关指针和完全地址节点中的正常主机指针都使用 PTR 资源记录指向相应主机的主要域名。

例如，IN-ADDR.ARPA 域将包含有关网 10 和 26 之间的 ISI 网关以及从网 10 到 MIT 的 MIT 网关的信息。

网络 18, 以及主机 A.ISI.EDU 和 MULTICS.MIT.EDU。假设 ISI 网关有地址 10.2.0.22 和 26.0.0.103, 名称为 MILNET-GW.ISI.EDU, 而 MIT 网关有地址 10.0.0.77 和 18.10.0.4, 名称为 GW.LCS.MIT.EDU, 则域名数据库将包含:

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.

因此, 一个想要在网络上定位网关的程序将发起一个形式为 QTYPE=PTR、QCLASS=IN、QNAME=10.IN-ADDR.ARPA 的查询。它将收到两个 RR 响应:

10.IN-ADDR.ARPA.	PTR 军事网络 -GW.ISI.EDU.
10.IN-ADDR.ARPA.	PTR GW.LCS.MIT.EDU.

该程序随后可以针对 MILNET- GW.ISI.EDU. 和 GW.LCS.MIT.EDU. 发起 QTYPE=A、QCLASS=IN 查询, 以发现这些网关的互联网地址。

一个想要找到与互联网主机地址 10.0.0.6 对应的主机名的解析器会发起 QTYPE=PTR、QCLASS=IN、QNAME=6.0.0.10.IN-ADDR.ARPA 形式的查询, 并会收到:

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

在使用这些服务时, 有几个注意事项:

- 由于 IN-ADDR.ARPA 特殊域和特定主机或网关的普通域位于不同的区域, 因此存在数据可能不一致的可能性。
- 网关通常在不同的域中有两个名称, 其中只有一个可以是主要的。
- 使用域名数据库初始化其路由表的系统必须以足够的网关信息开始, 以确保它们可以访问适当的名称服务器。
- 网关数据仅以与当前 HOSTS.TXT 文件等效的方式反映网关的存在。它不会取代来自 GGP 或 EGP 的动态可用性信息。

3.6. 定义新的类型、类和特殊命名空间

先前定义的类型和类是截至本备忘录日期正在使用的类型和类。应预期新的定义。本节为考虑向现有设施添加内容的那些设计人员提供一些建议。邮件列表

NAMEDROPPERS@SRI-NIC.ARPA 是讨论设计问题的通用论坛。

通常情况下，当需要向数据库中现有对象添加新信息，或我们需要为某个全新对象提供新数据格式时，应定义新类型。设计人员应尝试定义适用于所有类的一般类型及其 RDATA 格式，并避免信息重复。当 DNS 用于新协议等需要特定类数据格式的情况，或当需要现有名称空间的副本但必须使用单独的管理域时，应定义新类。

新类型和类别需要主文件的助记符；主文件的格式要求类型和类别的助记符是互斥的。

类型和类别的值必须分别是 QTYPEs 和 QCLASSes 的真子集。

当前系统使用多个 RR 来表示一个类型的多个值，而不是将多个值存储在单个 RR 的 RDATA 部分中。这对于大多数应用程序效率较低，但确实使 RR 更短。多个 RR 的假设包含在一些关于动态更新方法的实验性工作中。

当前系统试图最小化数据库中的数据重复，以确保一致性。因此，为了找到邮件交换的主机地址，您将邮件域名映射到主机名，然后将主机名映射到地址，而不是直接映射到主机地址。这种方法更受青睐，因为它避免了不一致的机会。

在定义一种新的数据类型时，不应使用多个 RR 类型来创建条目之间的顺序或表示等效绑定的不同格式，相反，这些信息应在 RR 主体中携带，并使用单个类型。这种政策避免了缓存多个类型和定义 QTYPEs 以匹配多个类型的问题。

例如，邮件交换绑定的原始形式使用了两种 RR 类型，一种表示“更近”的交换（MD），另一种表示“较近”的交换（MF）。困难在于，缓存中存在一种 RR 类型并不能提供任何关于另一种类型的信息，因为获取缓存信息的查询可能使用了 MF、MD 或 MAILA（这两种类型都匹配）的 QTYPE。重新设计

服务使用单一类型（MX）并在 RDATA 部分中包含一个 " 优先级 " 值，该值可以排序不同的 RRs。然而，如果缓存中找到任何 MX 记录，则所有记录都应存在。

4. 消息

4.1. 格式

域名的所有通信都使用一种称为消息的单一格式进行传输。消息的顶层格式分为 5 个部分（在某些情况下某些部分为空），如下所示：

-----+ 头部 ----- +-----+-----+ 问题 向名称服务器的查询 +-----+-----+ 应答 RRs 应答问题 +-----+-----+ 权威 RRs 指向一个权威 +-----+-----+ 附加 包含附加信息的资源记录 +-----+-----+

头部始终存在。头部包含指定其余部分是否存在以及消息是查询还是响应、是标准查询还是其他操作码等字段的字段。

头部之后的各部分的名称是根据其在标准查询中的用途得出的。问题部分包含描述向名称服务器提出问题的字段。这些字段是查询类型（QTYPE）、查询类别（QCLASS）和查询域名（QNAME）。最后三个部分具有相同的格式：一个可能为空的、连接起来的资源记录（RR）列表。答案部分包含回答问题的 RR；权威部分包含指向权威域名服务器的 RR；附加记录部分包含与查询相关但严格来说不是问题答案的 RR。

4.1.1. 头部格式

头部包含以下字段:



where:

- ID

A 16 位标识符分配给程序生成任何类型的查询。此标识符被复制相应的应答，并且可以被请求者使用以匹配未处理的查询的应答。
- QR

A one bit field that specifies whether this message is a query (0), or a response (1).
- OPCODE

A four bit field that specifies kind of query in this 消息。此值由查询发起者设置并复制到响应中。这些值是:

0

标准查询 (QUERY)

1

反向查询 (IQUERY)

2

服务器状态请求 (STATUS)

3-15

reserved for future use
- AA

权威应答 - 这部分在响应中有效, 并指定响应的名称服务器是该域名的问题部分中的权威。

注意, 由于别名, 答案部分的内容可能包含多个所有者名称。AA 位

corresponds to the name which matches the query name, or
答案部分中的第一个所有者名称。

TC TrunCation - specifies that this message was truncated
due to length greater than that permitted on the
传输通道。

RD 递归请求 - 该位可以在查询中设置并
被复制到响应中。 如果 RD 被设置，它将指示
名称服务器递归地执行查询。
递归查询支持是可选的。

RA 递归可用 - 这在响应中被设置或清除，并指示递归查询支持是否
可用。
名称服务器中可用。

Z 保留用于未来使用。 Must be zero in all queries
和响应。

RC Response code - this 4 bit field is set as part of
ODE 响应。 这些值具有以下
解释：

- 0 无错误条件
- 1 格式错误 - 名称服务器是
无法解释查询。
- 2 服务器故障 - 名称服务器是
由于服务器故障，无法处理此查询
名称服务器问题。
- 3 名称错误 - 仅适用于
来自权威名称
服务器的响应，此代码表示查询中引用的
域名不存在。
。
- 4 未实现 - 名称服务器不
支持请求的查询类型。
- 5 拒绝 - 名称服务器拒绝
因政策原因执行指定的操作。例如，一个名称

服务器可能不希望向特定的请求者提供
信息，
或者名称服务器可能不希望执行
特定的操作（例如，区域

传输)用于特定数据。

6-15

保留用于未来使用。

QDCOUNT

一个无符号 16 位整数，用于指定问题部分的条目数。

一个无符号 16 位整数, 用于指定问题部分的一个无符号 16 位整数, 用于指定答案部分中的资源记录数。

应答计数

一个无符号 16 位整数，用于指定答案部分中的资源记录数。

名称服务器计
数

an unsigned 16 bit integer specifying the number of name
服务器资源记录在授权记录中
节。

ARCOUNT

一个无符号 16 位整数，用于指定附加记录部分中的资源记录。

4.1.2. Question section format

问题部分用于在大多数查询中携带“问题”，即定义所询问内容的参数。该部分包含

QDCOUNT (通常为 1) 条条目, 每种格式如下:

```

      0   1   2   3   4   5   6   7   8   9    1   1   1   1   1   1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                         |
|                                     Q         /
|                               NAME          /
|                                         /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                   QTYPE+--+ |
+---+---+---+---+---+---+---++-----+---+---+---+---+---+
|                                   QCLASS+--+ |
+---+---+---+---+---+---+---++-----+---+---+---+---+---+

```

where:

QNAME

一个域名表示为一系列标签的序列，其中每个标签由一个长度八位组后跟相应数量的八位组组成。域名以根的空标签的零长度八位组终止。注意该字段可能是奇数个八位组；不使用填充。

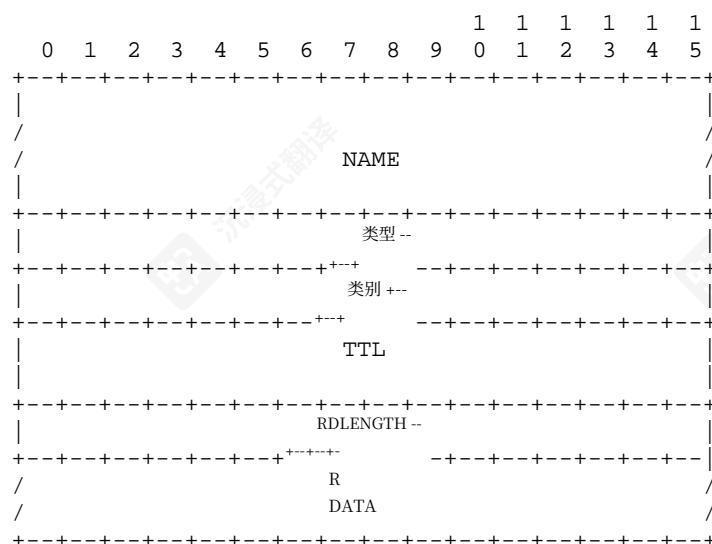
QTYPE

一个两八位组代码，用于指定查询的类型。该字段的值包括所有适用于 TYPE 字段的有效代码，以及一些更通用的代码，这些代码可以匹配多种类型的 RR。

QCLASS a two octet code that specifies the class of the query.
For example, the QCLASS field is IN for the Internet.

4.1.3. 资源记录格式

应答、权威和附加区域均采用相同的格式：一个资源记录的变长序列，其中记录的数量在头部相应的计数字段中指定。每个资源记录具有以下格式：



where:

NAME 一个与此资源记录相关的域名。

TYPE two octets containing one of the RR type codes. This field specifies the meaning of the data in the RDATA 字段。

类别	two octets which specify the class of the data in the RDATA 字段。
----	---

TTL 一个 32 位无符号整数，用于指定时间资源记录允许的间隔（以秒为单位）缓存后应丢弃。零值是
根据上下文，RR 只能用于当前进行中的事务，不应被缓存。

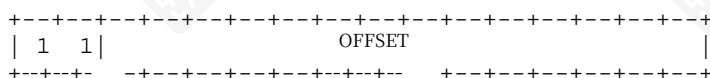
RDLENGTH	an unsigned 16 bit integer that specifies the length in RDATA 字段的字节。
RDATA	一个由八位字节组成的可变长度字符串，用于描述资源。此信息的格式各不相同 根据资源 r 的 TYPE 和类别 例如，如果 TYPE 是 A 且 CLASS 是 IN， RDATA 字段是一个 4 个八位字节的 ARPA 互联网地址。

ecord.

4.1.4. 消息压缩

为了减小消息的大小，域名系统利用了一种压缩方案，该方案消除了消息中域名名称的重复。在该方案中，一个完整的域名或域名末尾的一组标签被替换为指向该名称先前出现位置的指针。

指针的形式为一个两字节序列：



前两位比特为 1。这允许指针与标签区分开来，因为标签必须以两位零比特开头，因为标签限制在 63 个字节或更少。（10 和 01 组合保留用于未来使用。）偏移量字段指定从消息开始处（即域名头中 ID 字段的第一个字节）的偏移量。零偏移量指定 ID 字段的第一个字节，等等。

压缩方案允许消息中的域名表示为以下两种形式：

- 一系列以零字节结尾的标签
- 指针 - 一系列以指针结尾的标签

指针只能用于格式不是类别特定的域名出现。如果情况不是这样，名称服务器或解析器将需要知道它处理的 RR 的格式。目前还没有这种情况，但它们可能在未来 RDATA 格式中出现。

如果一个域名包含在受长度字段约束的消息部分中（例如 RR 的 RDATA 部分），并且存在压缩

使用时，长度计算使用压缩名称的长度，而不是展开名称的长度。

程序可以自由地避免在它们生成的消息中使用指针，尽管这将减少数据报的容量，并可能导致截断。但是所有程序都必须能够理解包含指针到达的消息。

例如，一个数据报可能需要使用域名 F.ISI.ARPA、FOO.F.ISI.ARPA、ARPA 和根。忽略消息的其他字段，这些域名可以表示为：

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

F.ISI.ARPA 的域名显示在偏移量 20 处。FOO.F.ISI.ARPA 的域名显示在偏移量 40 处；此定义使用指针将 FOO 的标签连接到先前定义的 F.ISI.ARPA。ARPA 的域名在偏移量 64 处使用指针定义，该指针指向 F.ISI.ARPA 名称中的 ARPA 组件（位于 20 处）；请注意，此指针依赖于 ARPA 是 20 处字符串中的最后一个标签。根域名是

由位于 92 处的单个零八位字节定义；根域名没有标签。

4.2. 传输

DNS 假定消息将作为数据报或通过虚拟电路承载的字节流进行传输。虽然虚拟电路可用于任何 DNS 活动，但由于开销较低和性能更好，数据报更适用于查询。由于需要可靠传输，区域刷新活动必须使用虚拟电路。

互联网支持使用 TCP [RFC-793] 在服务器端口 53（十进制）上进行名称服务器访问，以及使用 UDP [RFC-768] 在 UDP 端口 53（十进制）上进行数据报访问。

4.2.1. UDP usage

使用 UDP 用户服务器端口 53（十进制）发送的消息

通过 UDP 承载的消息限制为 512 字节（不计 IP 或 UDP 头部）。较长的消息将被截断，并在头部设置 TC 位。

UDP 不适用于区域传输，但建议用于互联网中的标准查询。使用 UDP 发送的消息可能会丢失，因此需要重传策略。查询或其响应可能会被网络或名称服务器处理而重新排序，因此解析器不应依赖它们按顺序返回。

最佳 UDP 重传策略会因互联网的性能和客户端的需求而异，但建议如下：

- 客户端在重复向特定服务器地址发送查询之前，应尝试其他服务器和服务器地址。
- 重传间隔应尽可能基于先前的统计数据。过于激进的重传可能会轻易地减慢整个社区（社区）的响应。取决于
 - 客户端与其预期服务器的连接情况，
 - 最小重传间隔应为 2-5 秒。

有关服务器选择和重传策略的更多建议，请参阅本备忘录的解析器部分。

4.2.2. TCP 使用

通过 TCP 连接发送的消息使用服务器端口 53（十进制）。消息以一个两字节的长度字段开头，该字段给出消息

长度，不包括两个字节长度字段。此长度字段允许低级处理在开始解析之前组装一条完整消息。

建议使用以下连接管理策略：

- 服务器不应阻塞其他等待 TCP 数据的活动。
- 服务器应支持多个连接。
- 服务器应假定客户端将发起连接关闭，并且应延迟关闭其连接端，直到所有未处理的客户端请求都已得到满足。
- 如果服务器需要关闭一个空闲连接以回收资源，它应等待该连接空闲一段时间，大约为两分钟。特别是，服务器应允许在单个连接上发起 SOA 和 AXFR 请求序列（该序列开始刷新操作）。由于无论如何服务器都无法应答查询，因此可以使用单方面关闭或重置来代替优雅关闭。

5. 主文件

主文件是包含以文本形式表示的 RRs 的文本文件。由于区域的内容可以表示为 RRs 列表的形式，因此主文件通常用于定义区域，尽管它也可以用于列出缓存的内容。因此，本节首先讨论主文件中 RRs 的格式，然后讨论当主文件用于在某些名称服务器中创建区域时的特殊注意事项。

5.1. Format

这些文件的格式是一系列条目。条目主要是按行组织的，但可以使用括号来延续跨行边界的条目列表，并且文本字面量可以包含文本中的 CRLF。任何制表符和空格的组合都作为组成条目的不同项之间的分隔符。主文件中的任何行的末尾可以以注释结束。注释以 “;” 开头（分号）。

以下条目被定义：

< 空行 > [< 注释 >]

\$ORIGIN < 域名 > [< 注释 >]

\$INCLUDE <file-name> [<domain-name>] [<comment>]

< 域名 > <RR> [< 注释 >]

< 空行 > <RR> [< 注释 >]

文件中任何位置允许空行，可以带有或没有注释。

定义了两个控制条目：\$ORIGIN 和 \$INCLUDE。\$ORIGIN 后跟一个域名，将当前相对域名的原点重置为所声明的名称。\$INCLUDE 将命名文件插入当前文件，并可以选择指定一个域名，该域名设置包含文件的相对域名原点。\$INCLUDE 还可以有注释。请注意，\$INCLUDE 条目永远不会改变父文件的相对原点，无论在包含文件中相对原点发生何种变化。

最后两种形式表示 RRs。如果 RR 的条目以空行开头，则假定该 RR 属于最后声明的所有者。如果 RR 条目以 < 域名 > 开头，则所有者名称将被重置。

<rr> 内容采用以下形式之一：

[<TTL>] [< 类别 >] < 类型 > <RDATA>

[< 类别 >] [<TTL>] < 类型 > <RDATA>

记录以可选的 TTL 和类别字段开头，然后是适当于类型和类别的类型和 RDATA 字段。类别和类型使用标准助记符，TTL 是一个十进制整数。省略的类别和 TTL 值默认为最后明确声明的值。由于类型和类别助记符是互斥的，解析是唯一的。（注意，这个顺序与示例中使用的顺序以及实际记录中使用的顺序不同；给定的顺序允许更简单的解析和默认处理。）

< 域名 > 在主文件中占很大比例。域名中的标签表示为字符串，并由点分隔。引号约定允许任意字符存储在域名中。以点结尾的域名称为绝对域名，并被视为完整。不以点结尾的域名称为相对域名；实际域名是相对部分与 \$ORIGIN、\$INCLUDE 或作为主文件加载例程参数指定的原点的连接。当没有可用原点时，相对名称是错误的。

< 字符串 > 以一种或两种方式表示：作为一组连续的字符，不包含内部空格，或作为以 " 开头并以 " 结尾的字符串。在 " 分隔的字符串中，任何字符都可以出现，除了 " 本身，它必须使用 \（反斜杠）进行转义。

因为这些文件是文本文件，所以需要几种特殊的编码来允许任意数据被加载。特别是：

	根的。
@	一个独立的 @ 用于表示当前 ORIGIN。
\X	where X is any character other than a digit (0-9), is used to quote that character so that its special meaning does not apply. For example, "\" can be used to place 标签中的一个点字符。
\DDD	其中每个 D 是一个数字，对应于由 DDD 描述的十进制数。结果八位字节被假定为文本，并且不检查其特殊含义。
()	括号用于分组跨越行边界的 DATA。实际上，行终止符在括号内不被识别。 括号内识别。
;	分号用于开始注释；该行的其余部分被忽略。 该行被忽略。

5.2. 使用主文件定义区域

当使用主文件加载区域时，如果在主文件中遇到任何错误，应抑制该操作。这是因为在单个错误可能会产生广泛的影响。例如，假设定义委托的 RR 存在语法错误；那么服务器将为子区域中的所有名称返回权威名称错误（除非子区域也存在于服务器上）。

除确保文件语法正确外，还应在以下方面进行其他有效性检查：

1. 文件中的所有 RR 应具有相同的类别。
2. 区域顶部应存在且仅存在一个 SOA RR。
3. 如果存在委托且需要粘合信息，则粘合信息应存在。

4. 区域中权威节点之外的节点所包含的信息应为粘合信息，而不是源或类似错误的结果。

5.3. 主文件示例

以下是一个示例文件，可用于定义 ISI.EDU 区域，并使用 ISI.EDU 作为源进行加载：

```
@      IN SOA      维纳斯      Action\domains (
                                20      ; 序列号
                                7200    ; 刷新
                                600     ; 重试
                                3600000; 过期
                                60)    ; 最小值

      NS      A.ISI.EDU.
      NS      维纳斯
      NS      VAXA
      MX      10      维纳斯
      MX      20      VAXA

A      A      26.3.0.103

维纳斯 A      10.1.0.52
      A      128.9.0.32

VAXA   A      10.2.0.27
      A      128.9.0.33
```

\$INCLUDE <SUBSYS>ISI- 邮箱 .TXT

文件 <SUBSYS>ISI-MAILBOXES.TXT 的位置是：

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

Note the use of the \ character in the SOA RR to specify the responsible person mailbox "Action.domains@E.ISI.EDU".

6. 名称服务器实现

6.1. 架构

名称服务器的最佳结构将取决于主机操作系统以及名称服务器是否与解析器操作集成，无论是通过支持递归服务，还是通过与解析器共享其数据库。本节讨论与解析器共享数据库的名称服务器的实现考虑，但其中大部分问题在任何名称服务器中都存在。

6.1.1. 控制

名称服务器必须采用多个并发活动，无论它们是在主机的操作系统（OS）中作为单独的任务实现，还是在单个名称服务器程序（name server program）内部进行多路复用。名称服务器在等待刷新（refreshing）或查询活动（query activities）的 TCP 数据（TCP data）时，简单地阻止 UDP 请求（UDP requests）的服务是不可接受的。类似地，名称服务器不应在不并行处理此类请求的情况下尝试提供递归服务（recursive service），尽管它可以选择将来自单个客户端（client）的请求进行序列化（serialize），或将来自同一客户端的相同请求（identical requests）视为重复（duplicates）。名称服务器在从主文件（master files）重新加载（reload）区域（zone）或将其合并（incorporates）到其数据库（database）中时，不应显著延迟（substantially delay）请求。

6.1.2. 数据库

虽然名称服务器实现（name server implementations）可以自由使用它们选择的任何内部数据结构（internal data structures），但建议的结构由三个主要部分组成：

- 一个“目录”（catalog）数据结构，该结构列出了此服务器可用的区域（zones），并指向（pointer）区域数据结构（zone data structure）。此结构的主要目的是为到达的标准查询（arriving standard queries）找到最近的祖先区域（ancestor zone），如果有的话。
- 为每个由名称服务器持有的区域分别使用数据结构名称服务器。
- 缓存数据的结构。（或者，为不同类别使用不同的缓存）

所有这些数据结构都可以实现为相同的树结构格式，但在不同的部分，节点上链接的数据不同：在目录中，数据是指向区域的指针，而在区域和缓存数据结构中，数据将是 RR。在设计树框架时，设计者应认识到查询处理需要使用不区分大小写的标签比较来遍历树；并且

在真实数据中，少数节点具有非常高的分支因子（100-1000 或更高），但绝大多数具有非常低的分支因子（0-1）。

一种解决大小写问题的方式是将每个节点的标签存储为两部分：标签的标准化大小写表示（所有 ASCII 字符为单一大小写），以及一个位掩码，用于指示哪些字符实际上具有不同的大小写。分支因子的多样性可以使用简单的链表来处理节点，直到分支因子超过某个阈值，然后过渡到哈希结构。在任何情况下，用于存储树段的哈希结构都必须确保哈希函数和程序保留 DNS 的大小写约定。

对数据库的不同部分使用单独结构的动机有以下几个因素：

- 目录结构可以是一个几乎静态的结构，仅在系统管理员更改服务器支持的区域时才需要更改。这种结构也可以用于存储用于控制刷新活动的参数。
- 区域的单独数据结构允许通过更改目录中的指针来简单地替换区域。区域刷新操作可以构建一个新的结构，并在完成后通过简单的指针替换将其拼接到数据库中。当区域刷新时，查询不应同时使用旧数据和新数据，这一点非常重要。
- 通过适当的搜索程序，区域中的权威数据将始终“隐藏”，并因此优先于缓存数据。
- 导致区域重叠等问题的区域定义中的错误可能会导致对查询的错误响应，但问题确定被简化了，一个“坏”区域的内容无法损坏另一个。
- 由于缓存最频繁更新，因此在系统重新启动期间它最容易受到损坏。它也可以充满过期的 RR 数据。在任一情况下，它都可以很容易地被丢弃，而不会干扰区域数据。

数据库设计的一个主要方面是选择一种结构，使名称服务器能够处理名称服务器主机的崩溃。名称服务器应在系统崩溃时保存的状态信息

包括目录结构（包括每个区域的刷新状态）以及区域数据本身。

6.1.3. 时间

RR 的 TTL 数据以及刷新活动的计时数据均取决于以秒为单位的 32 位计时器。在数据库中，缓存数据的刷新计时器和 TTL 会概念性地 " 倒计时 "，而区域中的数据则保持恒定的 TTL。

推荐的实现策略是采用两种方式存储时间：作为相对增量，以及作为绝对时间。一种实现方式是，对于一种类型使用正 32 位数字，对于另一种类型使用负数。区域中的 RRs 使用相对时间；刷新计时器和缓存数据使用绝对时间。绝对数字是相对于某个已知原点获取的，并在放入查询响应时转换为相对值。当绝对 TTL 在转换为相对值后为负数时，则数据已过期，应被忽略。

6.2. Standard query processing

标准查询处理的主要算法在 [RFC-1034] 中提出。

在处理具有 QCLASS=* 或某些其他匹配多个类的 QCLASS 的查询时，响应不应具有权威性，除非服务器可以保证响应涵盖所有类。

在构建响应时，应插入附加部分的 RRs，但答案或权威部分中的重复 RRs 可以省略不插入附加部分。

当响应过长需要截断时，截断应从响应的末尾开始，向数据报的前方处理。因此，如果权威部分有任何数据，则答案部分将保证是唯一的。

SOA 中的 MINIMUM 值应用于设置从区域分发的数据的 TTL 的下限。这个地板函数应在数据复制到响应时执行。这将允许未来的动态更新协议在不产生歧义语义的情况下更改 SOA MINIMUM 字段。

6.3. 区域刷新和重新加载处理

尽管服务器尽了最大努力，但由于语法错误等原因，它可能无法从主文件加载区域数据，或者无法在它的过期参数内刷新区域。在这种情况下，名称服务器

should answer queries as if it were not supposed to possess the zone.

如果一个主服务器通过 AXFR 传输区域，并且在传输过程中创建了一个新版本，主服务器应该尽可能继续传输旧版本。在任何情况下，它都不应发送一个版本的一部分和另一个版本的一部分。如果无法完成，主服务器应该重置正在进行区域传输的连接。

6.4. 反向查询（可选）

反向查询是 DNS 的可选部分。名称服务器不需要支持任何形式的反向查询。如果一个名称服务器收到它不支持的反向查询，它将返回一个错误响应，并在头部设置“未实现”错误。虽然反向查询支持是可选的，但所有名称服务器都必须至少能够返回错误响应。

6.4.1. 反向查询和响应的内容 反向查询会逆转标准查询操作所执行的映射；而标准查询将域名映射到资源，反向查询则将资源映射到域名。例如，标准查询可能将域名绑定到主机地址；相应的反向查询则将主机地址绑定到域名。

反向查询以消息答案部分中的一个 RR 的形式出现，其问题部分为空。查询 RR 的所有者名称及其 TTL 并不重要。响应会在问题部分携带问题，以标识所有拥有该查询 RR 的名称，这些名称是名称服务器所知的。由于没有名称服务器知道所有的域名空间，因此响应永远不能被假定为完整。因此，反向查询主要用于数据库管理和调试活动。反向查询不是将主机地址映射到主机名的可接受方法；请使用 IN-ADDR.ARPA 域名。

在可能的情况下，名称服务器应提供反向查询的不区分大小写的比较。因此，一个查询 MX 记录 "Venera.isi.edu" 的反向查询应与查询 "VENERA.ISI.EDU" 获得相同的响应；一个查询 HINFO 记录 "IBM-PC UNIX" 的反向查询应与查询 "IBM-pc unix" 产生相同的结果。然而，这不能得到保证，因为名称服务器可能拥有包含字符串的记录，但名称服务器不知道这些数据是字符串。

当名称服务器处理一个反向查询时，它要么返回：

1. 零个、一个或多个指定资源的域名作为 QNAMEs 在问题部分

7. 解析器实现

推荐的解析器算法的顶级在 [RFC-1034] 中讨论。本节讨论实现细节，假设数据库结构如本备忘录的名称服务器实现部分所述。

7.1. 将用户请求转换为查询

解析器首先需要将客户端请求，以适合本地操作系统的格式，转换为针对特定名称的特定 QTYPE 和 QCLASS 的 RR 搜索规范。在可能的情况下，QTYPE 和 QCLASS 应对应于单一类型和单一类别，因为这样可以使缓存数据的使用更加简单。原因在于，缓存中存在某一类型的数据并不能确认其他类型数据的存不存在，因此唯一确定的方法是咨询权威来源。如果使用 QCLASS=*, 则无法获得权威应答。

由于解析器必须能够多路复用多个请求才能高效地执行其功能，每个挂起请求通常表示在某个状态信息块中。这个状态块通常包含：

- 一个时间戳，指示请求开始的时间。时间戳用于判断数据库中的 RR 是否可用或已过期。这个时间戳使用之前讨论的用于区域和缓存中 RR 存储的绝对时间格式。注意，当 RR 的 TTL 指示相对时间时，RR 必须是及时的，因为它属于一个区域。当 RR 具有绝对时间时，它属于一个缓存，RR 的 TTL 与请求开始的时间戳进行比较。

请注意，使用时间戳比使用当前时间更优越，因为它允许具有零 TTL 的 RRs 以通常的方式进入缓存，但在系统负载、查询重传超时等情况下，即使经过许多秒，当前请求仍然可以使用它们。

- 一些用于限制为此请求执行的工作量的参数。

解析器对客户端请求执行的工作量必须有限制，以防止数据库中的错误，例如循环 CNAME 引用，以及操作问题，例如网络分区，这会阻止 ...

解析器无法访问其所需的名称服务器。虽然限制解析器将特定查询重传给特定名称服务器地址的次数是必要的，但解析器应有一个全局的每请求计数器来限制单个请求的工作。计数器应设置为某个初始值，每当解析器执行任何操作（重传超时、重传等）时递减。如果计数器变为零，请求应使用临时错误终止。

请注意，如果解析器结构允许一个请求并行启动其他请求，例如当需要访问一个名称服务器以处理一个请求时导致对名称服务器地址的并行解析，则生成的请求应使用较低的计数器启动。这可以防止数据库中的循环引用引发解析器活动的连锁反应。

- 在 [RFC-1 中讨论的 SLIST 数据结构 034]。

这种结构用于跟踪请求的状态，如果请求必须等待来自外部名称服务器的应答。

7.2. 发送查询

如 [RFC-1034]，中所述，解析器的基本任务是制定一个查询，以应答客户端请求并将该查询发送到能够提供信息的名称服务器。解析器通常只有关于要询问哪些服务器的非常强的提示，这些提示以 NS 资源记录的形式存在，并且可能需要根据 CNAMEs 修订查询，或者根据委派响应修订解析器正在询问的名称服务器集，这些委派响应将解析器指向更接近所需信息的名称服务器。除了客户端请求的信息外，解析器可能还需要调用自己的服务来确定其希望联系的名称服务器的地址。

在任何情况下，本备忘录中使用的模型假设解析器在多个请求之间切换注意力，其中一些来自客户端，另一些是内部生成的。每个请求都由一些状态信息表示，期望的行为是解析器以最大程度地提高请求被应答的可能性、最小化请求所需的时间并避免过度传输的方式将查询发送到名称服务器。关键算法使用请求的状态信息来选择下一个要查询的名称服务器地址，并计算一个超时时间，如果响应未到达，则该超时时间将触发下一个动作。下一个动作通常是向另一个服务器发送传输，但可能是一个临时错误。

客户端。

解析器始终从一个服务器名称列表开始查询（SLIST）。此列表将是解析器所知的最近祖先区域的全部 NS RRs。为了避免启动问题，如果解析器没有当前适当的 NS RRs，它应该有一组默认服务器，并请求这些服务器。然后，解析器将所有已知的名称服务器地址添加到 SLIST 中，并且在解析器有名称但没有名称服务器地址时，可能会开始并行请求以获取服务器的地址。

为了完成 SLIST 的初始化，解析器将其拥有的任何历史信息附加到 SLIST 中的每个地址。这通常包括地址响应时间的某种加权平均值，以及地址的击球平均数（即地址对请求的响应频率）。请注意，这些信息应按地址而不是按名称服务器进行维护，因为特定服务器的响应时间和击球平均数可能因地址而异。还应注意，这些信息实际上是针对解析器地址 / 服务器地址对的，因此具有多个地址的解析器可能希望为每个地址维护单独的历史记录。此步骤的一部分必须处理没有此类历史记录地址；在这种情况下，最坏的情况是 5-10 秒的预期往返时间，对于同一本地网络等，应提供较低估计。

请注意，每当发生委托时，解析器算法会重新初始化 SLIST。

这些信息建立了一个可用名称服务器地址的部分排序。每次选择一个地址时，状态应该被更改以防止再次选择该地址，直到所有其他地址都已尝试过。每个传输的超时时间应该比预测的平均值高 50-100%，以允许响应的方差。

一些细节：

- 解析器可能会遇到一种情况，即对于 SLIST 中列出的任何名称服务器都没有可用的地址，并且列表中的服务器正是通常用于查找其自身地址的那些服务器。这种情况通常发生在粘合地址 RR 的 TTL 小于标记委托的 NS RR 时，或者当解析器缓存了 NS 搜索的结果时。解析器应检测此条件，并在下一个祖先区域重新开始搜索，或者 alternatively 在根区域。

- 如果解析器从一个名称服务器收到服务器错误或其他古怪响应，它应该将其从 SLIST 中移除，并可能希望立即向下一个候选服务器地址发送传输。

7.3. 处理响应

处理到达的响应数据报的第一步是解析响应。此过程应包括：

- 检查头部是否合理。当预期响应时，丢弃是查询的数据报。
- 解析消息的各个部分，并确保所有 RR 格式正确。
- 作为可选步骤，检查到达数据的 TTLs，查找具有过长 TTL 的 RR。如果 RR 的 TTL 过长，例如超过 1 周，则可以丢弃整个响应，或将响应中的所有 TTL 限制为 1 周。

下一步是将响应与当前的解析器请求进行匹配。推荐策略是使用域名头中的 ID 字段进行初步匹配，然后验证问题部分是否与当前所需的信息对应。这要求传输算法在域名 ID 字段中分配几位作为某种请求标识符。此步骤有几个要点：

- 一些名称服务器从与接收查询不同的地址发送响应。也就是说，解析器不能依赖响应会从其发送相应查询的同一地址发送。此名称服务器错误通常在 UNIX 系统中遇到。
- 如果解析器将特定的请求重传给名称服务器，它应该能够使用任何传输的响应。但是，如果它正在使用响应来采样访问名称服务器的往返时间，它必须能够确定哪个传输与响应匹配（并保留每个出站消息的传输时间），或者仅根据初始传输计算往返时间。
- 名称服务器偶尔可能没有它根据某些 NS 资源记录应该拥有的区域的当前副本。解析器只需从当前 SLIST 中删除名称服务器，然后继续。

7.4. 使用缓存

通常，我们期望解析器缓存其从响应中接收的所有数据，因为这些数据可能在回答未来的客户端请求时有用。然而，有几种类型的数据不应被缓存：

- 当针对特定所有者名称有多个相同类型的 RR 时，解析器应该全部缓存它们或一个都不缓存。当响应被截断时，如果解析器不知道它是否有一套完整的 RR，它不应缓存可能是不完整的 RR 集。
- 缓存数据不应优先于权威数据使用，因此如果缓存会导致这种情况发生，则不应缓存该数据。
- 反向查询的结果不应被缓存。
- 对于标准查询，如果 QNAME 包含 "*" 标签且数据可能用于构建通配符，则不应缓存其结果。原因是缓存中不一定包含现有的 RR 或区域边界信息，而这些信息对于限制通配符 RR 的应用是必要的。
- 可疑可靠性的响应中的 RR 数据。当解析器收到非请求响应或请求之外的 RR 数据时，应丢弃它而不要缓存。基本含义是，在缓存任何数据包之前，应对该数据包进行所有有效性检查。

类似地，当解析器在响应中为某个名称有一组 RR，并希望缓存这些 RR 时，它应检查其缓存中是否已存在 RR。根据具体情况，可能优先考虑响应中的数据或缓存中的数据，但两者不应结合。如果响应中的数据来自答案部分的权威数据，则始终优先考虑。

8. 邮件支持

域名系统定义了将邮箱映射到域名的标准，以及两种使用邮箱信息来推导邮件路由信息的方法。第一种方法称为邮件交换绑定，另一种方法是邮箱绑定。邮箱编码标准和邮件交换绑定是 DNS 官方协议的一部分，是互联网邮件路由的推荐方法。邮箱绑定是一个实验性功能，仍在开发中，可能会发生变化。

邮箱编码标准假定邮箱名称的形式为 "< 本地部分 >@< 邮件域 >"。虽然这些部分中允许的语法在各个邮件互联网之间差异很大，但 ARPA 互联网的推荐语法在 [RFC-822] 中给出。

DNS 将 < 本地部分 > 编码为单个标签，并将 < 邮件域 > 编码为域名。来自 < 本地部分 > 的单个标签会添加到来自 < 邮件域 > 的域名前，以形成与邮箱对应的域名。因此，邮箱 HOSTMASTER@SRI-NIC.ARPA 被映射到域名 HOSTMASTER.SRI-NIC.ARPA。如果 < 本地部分 > 包含点或其他特殊字符，其在主文件中的表示将需要使用反斜杠引号以确保域名被正确编码。例如，邮箱 Action.domains@ISI.EDU 将表示为 Action\.domains.ISI.EDU。

8.1. 邮件交换绑定

邮件交换绑定使用邮箱规范中的 < 邮件域 > 部分来确定邮件应发送到何处。< 本地部分 > 甚至不会被查询。[RFC-974] 详细规定了此方法，在尝试使用邮件交换支持之前应参考。

此方法的一个优点是它将邮件目的地命名与支持邮件服务的主机解耦，代价是查找函数中的另一层间接层。然而，增加的层应该消除了对 < 本地部分 > 中对 "%"、"!", 等编码的需求。

该方法的核心在于，< 邮件域 > 被用作域名来定位 MX 记录，这些记录列出了愿意为 < 邮件域 > 接收邮件的主机，以及优先级值，这些值根据管理员为 < 邮件域 > 指定的顺序对主机进行排序。

在本备忘录中，< 邮件域 > ISI.EDU 用于示例，并使用 VENERA.ISI.EDU 和 VAXA.ISI.EDU 作为 ISI.EDU 的邮件交换主机。如果发件人有给 Mockapetris@ISI.EDU 的消息，它会通过查询 ISI.EDU 的 MX 记录来路由该消息。ISI.EDU 的 MX 记录命名为 VENERA.ISI.EDU 和 VAXA.ISI.EDU，类型 A 查询可以找到主机地址。

8.2. 邮箱绑定（实验性）

在邮箱绑定中，邮件发送者使用整个邮件目的地规范来构造一个域名。邮箱的编码域名用作 QTYPE=MAILB 查询中的 QNAME 字段。

对此查询可能存在几种结果：

1. 查询可以返回名称错误，指示邮箱不存在作为域名。

从长远来看，这将表明指定的邮箱不存在。然而，在邮箱绑定得到普遍使用之前，此错误条件应解释为：由全局部分标识的组织不支持邮箱绑定。适当的程序是在这一点上恢复到交换绑定。

2. 查询可以返回邮件重命名 (MR) 资源记录。

MR 资源记录在其 RDATA 字段中包含新的邮箱规范。邮件发送者应将旧邮箱替换为新邮箱并重试该操作。

3. 查询可以返回一个 MB 资源记录。

MB 资源记录在其 RDATA 字段中包含一个主机的域名。邮件发送者应通过适用的任何协议将该消息投递到该主机，例如 b,SMTP。

4. 查询可以返回一个或多个邮件组 (MG) 资源记录。

这意味着该邮箱实际上是一个邮件列表或邮件组，而不是单个邮箱。每个 MG 资源记录都有一个 RDATA 字段，用于标识该组中的一个成员邮箱。邮件发送者应将消息副本投递给每个成员。

5. 查询可以返回一个 MB 资源记录以及一个或多个 MG 资源记录。

这个条件表示邮箱实际上是邮件列表。发件人可以将消息投递到由 MB 记录指定的主机，该主机将转而投递给所有成员，或者发件人可以使用 MG 记录自行进行扩展。

在任何这些情况下，响应可能包括邮件信息 (MINFO) 记录。此记录通常与邮件组相关联，但也可以与 MB 相关联。MINFO 记录标识了两个邮箱。其中一个标识了原始邮箱名称的负责人。此邮箱应用于向邮件组添加请求等。MINFO 记录中的第二个邮箱名称标识了应接收邮件失败错误消息的邮箱。这对于邮件列表特别适用，当成员名称错误时应将错误报告给除向列表发送消息的人之外的其他人。

未来可能向此 RR 添加新字段。

9. 参考文献和书目

- [Dyer 87] S. Dyer, F. Hsu, 《Hesiod》, Project Athena
技术计划 - 名称服务, 1987 年 4 月, version 1.9.
描述了 Hesiod 名称服务的基本原理。
- [IEN-116] J. Postel, "互联网名称服务器", IEN-116,
南加州大学 / 信息科学研究所, 8 月 1 日 979.
一个被域名系统取代的名称服务, 但仍在使用。
- [Quarterman 86] J. Quarterman, 和 J. Hoskins, "值得注意的计算机网络",
ACM 通信, 1986 年 10 月, 第 29 卷, 第 10.
- [RFC-742] K. Harrenstien, "NAME/FINGER", RFC-742, 网络
信息中心, SRI 国际, 1977 年 12 月。
- [RFC-768] J. Postel, "用户数据报协议", RFC-768,
南加州大学 / 信息科学研究所, 198 年 8 月 0.
- [RFC-793] J. Postel, "传输控制协议", RFC-793,
南加州大学 / 信息科学研究所, 1981 年 9 月。
- [RFC-799] D. Mills, "互联网域名", RFC-799, COMSAT,
1981 年 9 月。

Suggests introduction of a hierarchy in place of a flat
互联网的命名空间。
- [RFC-805] J. Postel, "计算机邮件会议笔记", RFC-805,
南加州大学 / 信息科学研究所, 1982 年 2 月。
- [RFC-810] E. Feinler、K. Harrenstien、Z. Su 和 V. White, "DOD
互联网主机表规范", RFC-810, 网络
SRI 国际信息中心, 1982 年 3 月。
过时的。参见 RFC-952。
- [RFC-811] K. Harrenstien、V. White 和 E. Feinler, "主机名"
服务器", RFC-811, 网络信息中心, SRI
国际, 1982 年 3 月。

RFC 1035 Domain Implementation and Specification November 1987

Obsolete. See RFC-953.

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

[RFC-819] Z. Su, and J. Postel, "The Domain Naming Convention for Internet User Applications", RFC-819, Network Information Center, SRI International, August 1982.

Early thoughts on the design of the domain system.
Current implementation is completely different.

[RFC-821] J. Postel, "Simple Mail Transfer Protocol", RFC-821, USC/Information Sciences Institute, August 1980.

[RFC-830] Z. Su, "A Distributed System for Internet Name Service", RFC-830, Network Information Center, SRI International, October 1982.

Early thoughts on the design of the domain system.
Current implementation is completely different.

[RFC-882] P. Mockapetris, "Domain names - Concepts and Facilities," RFC-882, USC/Information Sciences Institute, November 1983.

Superceded by this memo.

[RFC-883] P. Mockapetris, "Domain names - Implementation and Specification," RFC-883, USC/Information Sciences Institute, November 1983.

Superceded by this memo.

[RFC-920] J. Postel and J. Reynolds, "Domain Requirements", RFC-920, USC/Information Sciences Institute, October 1984.

Explains the naming scheme for top level domains.

[RFC-952] K. Harrenstien, M. Stahl, E. Feinler, "DoD Internet Host Table Specification", RFC-952, SRI, October 1985.

Specifies the format of HOSTS.TXT, the host/address table replaced by the DNS.

RFC 1035	域名实现与规范	1987 年 11 月
[RFC-953]	<p>K. Harrenstien, M. 施塔尔, E. Feinler, "HOSTNAME 服务器", RFC 953, SRI, 1985 年 10 月。</p> <p>本 RFC 包含 主机名服务器协议的官方规范, 该协议已被 基于 TCP 的协议访问存储在 RFC-952 格式中的信息, 并用于获取 主机表的副本。</p>	DNS.
[RFC-973]	<p>P. Mockapetris, " 域名系统变更与 " 观察", RFC 973, USC/ 信息科学 1986 年 1 月学院。</p> <p>描述 RFC-882 和 RFC-883 的变更及其原因 它们。</p>	
[RFC-974]	<p>C. Partridge, "Mail routing and the domain system", RFC-974, CSNET CIC BBN 实验室, 1986 年 1 月。</p> <p>Describes the transition from HOSTS.TXT based mail 寻址到更强大的 MX 系统, 该系统与 域名系统一起使用。</p>	
[RFC-1001]	<p>NetBIOS Working Group, "Protocol standard for a NetBIOS service on a TCP/UDP transport: Concepts and Methods", RFC-1001, 1987 年 3 月。</p> <p>本 RFC 和 RFC-1002 是 TCP/IP 上 NETBIOS 的初步设计, 提议基于 Net 在 DNS 之上的名称服务。</p>	BIOS
[RFC-1002]	<p>NetBIOS 工作组, "NetBIOS 服务在 TCP/UDP 传输上的协议标准: 详细规范 " 在 TCP/UDP 传输上的协议标准: 详细 规范, RFC-1002, 1987 年 3 月。</p>	
[RFC-1010]	<p>J. 雷诺兹和 J. Postel, " 分配编号 ", RFC-1010, 南加州大学 / 信息科学研究所, 1987 年 5 月。</p> <p>包含主机名的套接字编号和助记符, 操作系统等。</p>	
[RFC-1031]	<p>W. Lazear, "MILNET Name Domain Transition", RFC-1031, 1987 年 11 月。</p> <p>描述了将 MILNET 转换为 DNS 的计划。</p>	
[RFC-1032]	<p>M. Stahl, " 建立域 - 管理员指南 " , RFC-1032, 1987 年 11 月。</p>	

RFC 1035	域名实现与规范	1987 年 11 月
	Describes the registration policies used by the NIC to 管理顶级域名和委托子区域。	
[RFC-1033]	M. Lottor, " 域名管理员操作指南 ", RFC-1033, 1987 年 11 月。	
	域名管理员的烹饪书。	
[Solomon 82]	M. Solomon、L. Landweber 和 D. Neuhengen, "CSNET Name Server", 计算机网络, vol 6, nr 3, July 1982。	
	描述了 CSNET 的独立名称服务 从 DNS 和 CSNET 中的 DNS 使用情况。	

索引

* 13

; 33, 35

<character-string> 35

<domain-name> 34

@ 35

\ 35

A 12

字节序 8

CH 13

Character case 9

CLASS 11

CNAME 12

Completion 42

CS 13

Hesiod 13

HINFO 12

HS 13

IN 13

IN-ADDR.ARPA domain 22

Inverse queries 40

Mailbox names 47

MB 12

MD 12

MF 12

MG 12

MINFO 12

MINIMUM 20

MR 12

MX 12

NS 12

NULL 12

Port numbers 32

Primary server 5

PTR 12, 18

QCLASS 13
QTYPE 12
RDATA 12
RDLENGTH 11
辅助服务器 5
SOA 12
Stub 解析器 7

TCP 32
TXT 12 类
型 11
UDP 32
WKS 12