网络工作组
征求意见:1034
弃用:RFC 882、883、973

P. Mockapetris ISI 1987 年 11 月

域名 - 概念和设施

1. 本备忘录的状态

此 RFC 是对域名系统 (DNS) 的介绍,并省略了许多细节,这些细节可以在配套的 RFC“域名 - 实现和规范”[RFC-1035] 中找到。 RFC 假定读者熟悉本备忘录中讨论的概念。

DNS 功能和数据类型的子集构成了官方协议。官方协议包括标准查询及其响应和大多数 Internet 类数据格式（例如,主机地址）。

但是,域系统是有意扩展的。研究人员不断提出、实施和试验新的数据类型、查询类型、类、函数等。因此,虽然官方协议的组件预计将保持基本不变并作为生产服务运行,但实验行为应始终符合预期官方协议之外的扩展。这些 RFC 中明确标记了实验性或过时的功能,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是最新的或完整的,因为它们的目的主要是教学。本备忘录的分发是无限的。

2. 简介

此 RFC 介绍了域样式名称、它们在 Internet 邮件和主机地址支持中的用途,以及用于实现域名功能的协议和服务器。

2.1.域名的历史

域系统发展的动力是
互联网:

- 主机名到地址的映射由网络信息中心 (NIC) 在单个文件 (HOSTS.TXT) 中维护,该文件由所有主机通过 FTP 传输 [RFC-952、RFC-953]。总网

该方案分发新版本时消耗的带宽与网络中主机数量的平方成正比,即使使用多级FTP,网卡主机的出站
FTP负载也相当可观。

主机数量的爆炸式增长对未来来说并不是好兆头。

- 网络人口的特征也在发生变化。这
组成原始 ARPANET 的分时主机正在被本地工作站网络所取代。本地组织正在管理他们自己的名称和地址,但必须
等待 NIC 更改 HOSTS.TXT 才能使更改在整个 Internet 上可见。组织还希望在名称空间上有一些本地结构。

- 互联网上的应用越来越多
复杂并创造了对通用名称服务的需求。

结果是关于名称空间及其管理的几个想法 [IEN-116、RFC-799、RFC-819、RFC-830]。提案各不相同,但一个共同点是分层命名
空间的想法,分层大致对应于组织结构,名称使用“.”。作为标记层级之间边界的字符。 [RFC-882、RFC-883] 中描述了使用
分布式数据库和通用资源的设计。根据多次实施的经验,该系统演变为本文中描述的方案

备忘录。

术语“域”或“域名”在此处描述的 DNS 之外的许多上下文中使用。通常,术语域名用于指代结构由点表示的名称,但与 DNS 无关。

这在邮件寻址中尤其如此 [Quarterman 86]。

2.2. DNS 设计目标

DNS 的设计目标影响其结构。他们是:

- 主要目标是用于引用资源的一致名称空间。为了避免特殊编码引起的问题,不应要求名称包含网络标识符、地址、路由或
类似信息作为名称的一部分。

- 数据库的庞大规模和更新频率
建议一定要分布式维护,用本地缓存来提高性能。接近那个

尝试收集整个数据库的一致副本将变得越来越昂贵和困难,因此应该避免。同样的原则适用于名称空间的结构,特别是创建和删除名称的机制;这些也应该分发。

- 如果在获取数据的成本、更新速度和缓存的准确性之间进行权衡,数据源应控制权衡。

- 实施此类设施的成本决定了它通常是有用的,而不限于单个应用程序。

我们应该能够使用名称来检索主机地址、邮箱数据和其他尚未确定的信息。与名称关联的所有数据都标有类型,查询可以限制为单一类型。

- 因为我们希望名称空间在不同的地方有用
网络和应用程序,我们提供了使用具有不同协议系列或管理的相同名称空间的能力。例如,主机地址格式因协议而异,尽管所有协议都有地址的概念。

DNS 用类别和类型标记所有数据,因此我们可以允许并行使用地址类型数据的不同格式。

- 我们希望名称服务器事务独立于
承载它们的通信系统。一些系统可能希望使用数据报来进行查询和响应,并且只为需要可靠性的事务(例如,数据库更新、长事务)建立虚电路;其他系统将专门使用虚电路。

- 该系统应该适用于广泛的主机
能力。个人计算机和大型分时主机都应该能够使用该系统,尽管方式可能不同。

2.3.关于使用的假设

域系统的组织源于对其用户社区的需求和使用模式的一些假设,旨在避免通用数据库系统中发现的许多复杂问题。

假设是:

- 整个数据库的大小最初是成比例的

与使用该系统的主机数量相关,但随着邮箱和其他信息被添加到域系统中,最终会增长到与这些主机上的用户数量成正比。

- 系统中的大部分数据变化非常缓慢（例如,邮箱绑定、主机地址）,但系统应该能够处理变化更快的子集（以秒或分钟为单位）。

- 用于分配的行政边界

数据库的责任通常对应于拥有一台或多台主机的组织。负责一组特定域的每个组织都将提供冗余名称服务器,或者在该组织自己的主机上,或者在该组织安排的其他主机上

使用。

- 域系统的客户端应该能够识别

在接受对这个“受信任”集之外的名称服务器的推荐之前,他们更愿意使用受信任的名称服务器。

- 信息访问比瞬时更新或一致性保证更为重要。因此,更新过程允许更新通过域系统的用户渗透出去,而不是保证所有副本都同时更新。当由于网络或主机故障导致更新不可用时,通常的做法是相信旧信息,同时继续努力更新它。通用模型是副本分布有刷新超时。分发者设置超时值,分发的接收者负责执行刷新。在特殊情况下,可以指定非常短的间隔,或者所有者可以禁止复制。

- 在任何具有分布式数据库的系统中,特定名称服务器可能会收到只能由其他服务器回答的查询。处理这个问题的两种一般方法是“递归”,其中第一台服务器在另一台服务器上为客户端寻求查询,以及“迭代”,其中服务器将客户端指向另一台服务器并让客户端继续执行询问。两种方法各有优缺点,但迭代方法更适合访问数据报样式。域系统需要迭代方法的实现,但允许递归方法作为一种选择。

域系统假定所有数据都源自分散在使用域系统的主机中的主文件。这些主文件由本地系统管理员更新。主文件是由本地名称服务器读取的文本文件,因此可以通过名称服务器提供给域系统的用户。用户程序通过称为解析器的标准程序访问名称服务器。

主文件的标准格式允许它们在主机之间交换（通过 FTP、邮件或其他一些机制）;当组织需要域但不想支持名称服务器时,此功能很有用。组织可以使用文本编辑器在本地维护主文件,将它们传输到运行名称服务器的外部主机,然后与名称服务器的系统管理员安排以加载文件。

每个主机的名称服务器和解析器都由本地系统管理员配置 [RFC-1033]。对于名称服务器,此配置数据包括本地主文件的标识以及要从外部服务器加载哪些非本地主文件的说明。名称服务器使用主文件或副本来加载其区域。对于解析器,配置数据标识了应该作为主要信息来源的名称服务器。

域系统定义了访问数据和引用其他名称服务器的过程。域系统还定义了缓存检索数据和定期刷新系统管理员定义的数据的过程。

系统管理员提供:

- 区域边界的定义。
- 数据的主文件。
- 更新主文件。
- 所需刷新策略的声明。

域系统提供:

- 资源数据的标准格式。
- 查询数据库的标准方法。
- 名称服务器从中刷新本地数据的标准方法
  外国名称服务器。

2.4. DNS 的要素

DNS 包含三个主要组件：

- 域名空间和资源记录，它们是
  树结构名称空间和与名称相关的数据的规范。从概念上讲，域名空间树的每个节点和叶子都命名了一组信息，查询操作是
  尝试从特定集合中提取特定类型的信息。查询命名感兴趣的域名并描述所需资源信息的类型。例如，互联网使用它的一些域名来识
  别主机；对地址资源的查询返回 Internet 主机地址。

- 名称服务器是保存有关域树结构和设置信息的服务器程序。名称服务器可以缓存关于域树任何部分的结构或设置信息，但一般而言，
  特定名称服务器具有关于域空间子集的完整信息，以及指向其他名称服务器的指针，可用于从中获取信息域树的任何部分。名称服务
  器知道它们拥有完整信息的域树部分；名称服务器被认为是名称空间这些部分的权威。权威信息被组织成称为区域的单元，这些区
  域可以自动分配给为区域中的数据提供冗余服务的名称服务器。

- RESOLVERS 是从名称中提取信息的程序
  服务器响应客户端请求。解析器必须能够访问至少一个名称服务器并使用该名称服务器的信息直接回答查询，或使用对其他
  名称服务器的引用来继续查询。解析器通常是用户程序可以直接访问的系统例程；因此解析器和用户程序之间不需要协议。

这三个组件大致对应于域系统的三个层或视图：

- 从用户的角度来看，通过简单的过程或操作系统调用本地解析器来访问域系统。

  域空间由一棵树组成，用户可以从树的任何部分请求信息。

- 从解析器的角度来看，域系统是
  由未知数量的名称服务器组成。每个名字

服务器拥有整个域树数据的一个或多个部分,但解析器将这些数据库中的每一个视为本质上是静态的。

- 从名称服务器的角度来看,域系统由称为区域的独立本地信息集组成。名称服务器具有某些区域的本地副本。名称服务器必须定期从本地文件或外部名称服务器中的主副本刷新其区域。名称服务器必须同时处理来自解析器的查询。

为了性能,实现可能会耦合这些功能。例如,与名称服务器位于同一台机器上的解析器可能共享一个数据库,该数据库由名称服务器管理的区域和解析器管理的缓存组成。

3. 域名空间和资源记录

3.1.名称空间规范和术语

域名空间是一个树状结构。树上的每个节点和叶子对应一个资源集(可能为空)。域系统在内部节点和叶子的使用之间没有区别,本备忘录使用术语“节点”来指代两者。

每个节点都有一个标签,长度为 0 到 63 个八位字节。兄弟节点可能没有相同的标签,尽管相同的标签可以用于非兄弟节点。保留一个标签,即用于根的空(即,零长度)标签。

节点的域名是从节点到树根的路径上的标签列表。按照惯例,组成域名的标签从左到右打印或阅读,从最具体(最低,离根最远)到最不具体(最高,最接近根)。

在内部,处理域名的程序应该将它们表示为标签序列,其中每个标签都是一个长度八位字节,后跟一个八位字节字符串。因为所有域名都在根部结束,根部的标签为空字符串,所以这些内部表示可以使用长度为零的字节来终止域名。

按照惯例,域名可以任意大小写存储,但是所有现有域函数的域名比较都是以不区分大小写的方式完成的,假设是 ASCII 字符集和高阶零位。这意味着您可以自由创建一个带有标签“A”的节点或一个带有标签“a”的节点,但不能同时作为兄弟;你可以参考使用“a”或“A”。当您收到域名或

标签,你应该保留它的大小写。这种选择的理由是,我们可能有一天需要为新服务添加完整的二进制域名;现有服务不会改变。

当用户需要键入域名时,省略每个标签的长度,并用点（“.”）分隔标签。由于完整的域名以根标签结尾,这导致打印形式以点结尾。我们使用此属性来区分:

- 代表完整域名的字符串（通常称为“绝对”）。例如,“poneria.ISI.EDU”。

- 表示不完整域名起始标签的字符串,应由本地软件使用本地域（通常称为“相对”）的知识来完成。例如,ISI.EDU 域中使用的“poneria”。

相对名称要么相对于众所周知的来源,要么相对于用作搜索列表的域列表。相对名称主要出现在用户界面上,它们的解释因实现而异,在主文件中,它们与单一来源域名相关。最常见的解释是使用词根“.”。作为单一来源或作为搜索列表的成员之一,因此多标签相对名称通常是省略尾随点以节省键入的名称。

为了简化实现,表示域名的八位字节总数（即所有标签八位字节和标签长度的总和)限制为 255。

域由域名标识,并且由位于或低于指定域的域名的那部分域名空间组成。如果一个域包含在另一个域中,则该域是另一个域的子域。可以通过查看子域的名称是否以包含域的名称结尾来测试这种关系。

例如,ABCD 是 BCD、CD、D 和 的子域。

3.2.行政使用指南

作为一项政策,DNS 技术规范不强制要求特定的树结构或选择标签的规则;它的目标是尽可能通用,以便可以用来构建任意应用程序。特别是,系统的设计使得名称空间不必按照网络边界、名称服务器等来组织。这样做的理由不是名称空间不应该有隐含的语义,而是隐含语义的选择应该保持开放,以用于解决以下问题

手,并且树的不同部分可以有不同的隐含语义。例如,IN-ADDR.ARPA 域是按网络和主机地址组织和分布的,因为它的作用是将网络或主机号转换为名称； NetBIOS 域 [RFC-1001、RFC 1002] 是扁平的,因为它适用于该应用程序。

然而,有一些准则适用于用于主机、邮箱等的名称空间的“正常”部分,这将使名称空间更加统一,提供增长,并最大限度地减少软件从旧版本转换时出现的问题主机表。关于树顶层的政治决定起源于 RFC-920。

[RFC-1032] 中讨论了顶层的当前策略。 MILNET 转换问题包含在 [RFC-1031] 中。

最终将被分成多个区域的较低域应该在域的顶部提供分支,以便最终的分解可以在不重命名的情况下完成。使用特殊字符、前导数字等的节点标签可能会破坏依赖于更多限制性选择的旧软件。

3.3.使用技术指南

在 DNS 可用于保存某种对象的命名信息之前,必须满足两个需求：

- 对象名称和域名之间的映射约定。这描述了如何访问有关对象的信息。

- 用于描述对象的 RR 类型和数据格式。

这些规则可以非常简单或相当复杂。通常,设计人员必须考虑现有格式并为现有用途的向上兼容性制定计划。可能需要多个映射或映射级别。

对于主机,映射取决于主机名的现有语法,它是域名通常文本表示的子集,以及用于描述主机地址的 RR 格式等。因为我们需要从地址到主机名的可靠逆映射,还定义了地址到 IN-ADDR.ARPA 域的特殊映射。

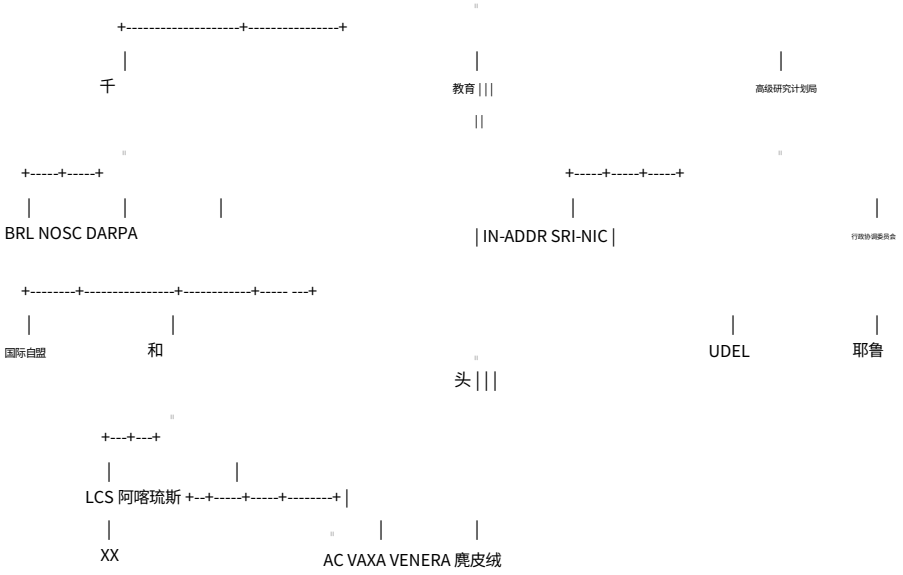
对于邮箱,映射稍微复杂一些。通常的邮件地址 <local-part>@<mail-domain> 通过将 <local-part> 转换为单个标签（不管它包含的点)映射到域名,将 <mail-domain> 转换为域名使用域名的常用文本格式（点表示标签中断）,并将两者连接起来形成一个域名。于是邮箱

HOSTMASTER@SRI-NIC.ARPA 由 HOSTMASTER.SRI-NIC.ARPA 表示为域名。理解这种设计背后的原因还必须考虑邮件交换方案 [RFC 974]。

典型的用户并不关心定义这些规则，但应该明白，它们通常是在与旧用法向上兼容的愿望、不同对象定义之间的交互以及在定义规则时不可避免地添加新功能的愿望之间进行多次折衷的结果. DNS 用于支持某些对象的方式通常比 DNS 固有的限制更为重要。

3.4.名称空间示例

下图展示了当前域名空间的一部分，在这个RFC的很多例子中都用到了。请注意，树是实际名称空间的一个非常小的子集。



在此示例中，根域具有三个直接子域：MIL、EDU 和 ARPA。 LCS.MIT.EDU 域有一个名为 XX.LCS.MIT.EDU 的直接子域。所有的叶子也是域。

3.5.首选名称语法

DNS 规范试图在规则中尽可能通用

用于构建域名。这个想法是任何现有对象的名称都可以表示为域名,只需进行最少的更改。

但是,在为对象分配域名时,谨慎的用户会选择一个既满足域系统规则又满足对象任何现有规则的名称,无论这些规则是已发布的还是已存在的程序暗示的。

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

以下语法将减少许多使用域名的应用程序(例如,邮件、TELNET)出现的问题。

<域> ::= <子域> | “”

<子域> ::= <标签> | <子域> “。” <标签>

<标签> ::= <字母> [[<ldh-str>] <let-dig>]

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

<让你炒作> ::= <让你> | “ ”

<let-dig> ::= <字母> | <数字>

<letter> ::= 52 个字母字符中的任何一个,大写字母 A 到 Z 和小写字母 a 到 z

<digit> ::= 0 到 9 的十个数字中的任何一个

请注意,虽然域名中允许使用大写和小写字母,但大小写无关紧要。也就是说,拼写相同但大小写不同的两个名称将被视为相同。

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,并且只有字母、数字和连字符作为内部字符。长度也有一些限制。标签不得超过 63 个字符。

例如,以下字符串标识 Internet 中的主机:

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

3.6.资源记录

域名标识一个节点。每个节点都有一组资源

信息,可能为空。与特定名称关联的资源信息集由单独的资源记录 (RR) 组成。一组 RR 的顺序并不重要,不需要由名称服务器、解析器或 DNS 的其他部分保留。

当我们谈论特定的 RR 时,我们假设它具有以下内容:

所有者 这是找到 RR 的域名。

类型 这是一个编码的 16 位值,指定此资源记录中的资源类型。类型指的是抽象资源。

本备忘录使用以下类型:

A 主机地址

别名 标识别名的规范名称

资讯 标识主机使用的 CPU 和操作系统

MX 标识域的邮件交换。有关详细信息,请参阅 [RFC-974。

NS
域的权威名称服务器

PTR
指向域名空间另一部分的指针

SOA
标识权限区域的开始]

班级 这是一个编码的 16 位值,用于标识协议族或协议实例。

本备忘录使用以下类:

在 因特网系统

CH 混沌系统

TTL 这是RR的生存时间。该字段是一个以秒为单位的 32 位整数,主要由解析器在缓存 RR 时使用。 TTL 描述了 RR 在应该被丢弃之前可以缓存多长时间。

数据 这是描述资源的类型,有时是类依赖数据:

A	对于 IN 类,一个 32 位 IP 地址 对于 CH 类,域名后跟一个 16 位八进制混沌地址。
别名	一个域名。
MX	一个 16 位首选项值（越低越好）,后跟愿意充当所有者域的邮件交换的主机名。
NS	一个主机名。
PTR	一个域名。
面向服务架构	几个领域。

所有者名称通常是隐含的,而不是构成 RR 的组成部分。例如,许多名称服务器在内部为名称空间形成树或散列结构,并将 RR 链接到节点。剩余的 RR 部分是对所有 RR 一致的固定标头（类型、类、TTL)和适合所描述资源需要的可变部分（RDATA）。

TTL字段的含义是一个RR在缓存中可以保存多长时间的时间限制。此限制不适用于区域中的权威数据;它也超时了,但是通过区域的刷新策略。 TTL 由管理员为数据来源区域分配。虽然短 TTL 可用于最小化缓存,零 TTL 禁止缓存,但 Internet 性能的现实表明这些时间对于典型的主机来说应该是天的数量级。如果可以预料到变化,则可以在变化之前降低 TTL 以最大限度地减少变化期间的不一致,然后在变化之后增加回原来的值。

RR 的 RDATA 部分中的数据以二进制字符串和域名的组合形式携带。域名经常用作指向 DNS 中其他数据的“指针”。

3.6.1. RR的文本表达

RR 在 DNS 协议的数据包中以二进制形式表示,存储在名称服务器或解析器中时通常以高度编码的形式表示。在这份备忘录中,我们采用了一种类似于使用的风格

在主文件中以显示 RR 的内容。在这种格式中,大多数 RR 都显示在一行中,尽管可以使用括号续行。

该行的开头给出了 RR 的所有者。如果一行以空白开头,则所有者被假定为与前一个 RR 的所有者相同。通常包含空行以提高可读性。

在所有者之后,我们列出了 RR 的 TTL、类型和类别。 Class和type使用上面定义的助记符,TTL是type字段前的一个整数。为了避免解析时出现歧义,类型和类助记符是不相交的,TTL 是整数,类型助记符总是在最后。为清楚起见,示例中经常省略 IN 类和 TTL 值。

RR 的资源数据或 RDATA 部分是使用数据典型表示的知识给出的。

例如,我们可以将消息中携带的 RR 显示为:

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

MX RRs 有一个 RDATA 部分,它由一个 16 位数字和一个域名组成。地址 RR 使用标准 IP 地址格式来包含 32 位互联网地址。

此示例显示六个 RR,三个域名中的每一个都有两个 RR。

同样我们可能会看到:

```
XX.LCS.MIT.EDU。在               A       10.0.0.44 麻省理
                         CH      A       工学院。 2420
```

此示例显示了 XX.LCS.MIT.EDU 的两个地址,每个地址属于不同的类别。

3.6.2.别名和规范名称

在现有系统中,主机和其他资源通常有多个名称来标识同一资源。例如,名称 C.ISI.EDU 和 USC-ISIC.ARPA 都标识同一台主机。同样,在邮箱的情况下,许多组织提供了许多实际上去同一个邮箱的名称;例如 Mockapetris@C.ISI.EDU, Mockapetris@B.ISI.EDU,

和 PVM@ISI.EDU 都转到同一个邮箱（尽管这背后的机制有些复杂）。

这些系统中的大多数都有一个概念,即一组等效名称中的一个是规范名称或主要名称,而所有其他名称都是别名。

域系统使用规范名称 (CNAME) RR 提供此类功能。 CNAME RR 将其所有者名称标识为别名,并在 RR 的 RDATA 部分指定相应的规范名称。 如果 CNAME RR 出现在节点上,则不应出现其他数据;这可确保规范名称及其别名的数据不能不同。此规则还确保可以使用缓存的 CNAME，而无需检查其他 RR 类型的权威服务器。

CNAME RR 在 DNS 软件中引起特殊操作。当名称服务器无法在与域名关联的资源集中找到所需的 RR 时,它会检查资源集是否包含具有匹配类的 CNAME 记录。如果是,名称服务器在响应中包含 CNAME 记录,并在 CNAME 记录的数据字段中指定的域名处重新启动查询。此规则的一个例外是匹配 CNAME 类型的查询不会重新启动。

例如,假设名称服务器正在处理 USC ISIC.ARPA 的查询,要求 A 类信息,并具有以下资源记录：

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

C.ISI.EDU          在               A          10.0.0.52
```

这两个 RR 都将在对类型 A 查询的响应中返回,而类型 CNAME 或
查询应该只返回 CNAME。

RR 中指向另一个名称的域名应始终指向主名称而不是别名。这避免了访问信息时的额外间接访问。例如,上述主机的名称为 RR 的地址应该是：

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

而不是指向 USC-ISIC.ARPA。当然,根据健壮性原则,域软件在出现 CNAME 链或循环时不应失败;应遵循 CNAME 链并将 CNAME 循环标记为错误。

3.7.查询

查询是可以发送到名称服务器以激发

回复。在 Internet 中,查询通过 UDP 数据报或 TCP 连接进行。名称服务器的响应要么回答查询中提出的问题,要么将请求者引向另一组名称服务器,要么发出某种错误情况的信号。

通常,用户不直接生成查询,而是向解析器发出请求,解析器依次向名称服务器发送一个或多个查询,并处理可能导致的错误情况和引用。当然,查询中可能提出的问题确实决定了解析器可以提供的服务类型。

DNS 查询和响应以标准消息格式传送。消息格式有一个标题,其中包含许多始终存在的固定字段,以及四个携带查询参数和 RR 的部分。

标头中最重要的字段是称为操作码的四位字段,用于分隔不同的查询。在可能的 16 个值中,一个(标准查询)是官方协议的一部分,两个(反向查询和状态查询)是选项,一个(完成)已过时,其余未分配。

这四个部分是:

问题	携带查询名称和其他查询参数。
回答	携带直接回答查询的 RR。
权威	携带描述其他权威服务器的 RR。 可以选择性地在回答部分携带SOA RR作为权威数据。
额外的	携带可能有助于在其他部分中使用 RR 的 RR。

请注意,这些部分的内容而非格式随标头操作码而变化。

3.7.1.标准查询

标准查询指定目标域名 (QNAME)、查询类型 (QTYPE) 和查询类别 (QCLASS),并要求匹配的 RR。此类查询占 DNS 查询的绝大部分,除非另有说明,否则我们使用术语“查询”来表示标准查询。 QTYPE 和 QCLASS 字段都是 16 位长,并且是已定义类型和类的超集。

QTYPE 字段可能包含：

<任何类型>	只匹配那种类型。（例如,A、PTR）。
飞行器	特区转移QTYPE。
邮箱	匹配所有与邮箱相关的 RR（例如 MB 和 MG）。
*	匹配所有 RR 类型。

QCLASS 字段可能包含：

<任何班级>	仅匹配该类别（例如,IN、CH）。
*	匹配所有 RR 类。

使用查询域名、QTYPE 和 QCLASS,名称服务器查找匹配的 RR。除了相关记录之外,名称服务器还可以返回指向具有所需信息的名称服务器的 RR,或者返回预期对解释相关 RR 有用的 RR。例如,没有请求信息的名称服务器可能知道有请求信息的名称服务器;在相关 RR 中返回域名的名称服务器也可能返回将该域名绑定到地址的 RR。

例如,发送邮件到 Mockapetris@ISI.EDU 的邮件程序可能会要求解析器提供有关 ISI.EDU 的邮件信息,从而导致查询 QNAME=ISI.EDU、QTYPE=MX、QCLASS=IN。响应的答案部分将是：

ISI.EDU。	MX	10 VENERA.ISI.EDU。
	MX	10 VAXA.ISI.EDU。

而附加部分可能是：

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

因为服务器假设如果请求者想要邮件交换信息,它可能很快就会想要邮件交换的地址。

请注意,QCLASS=* 构造需要有关权限的特殊解释。由于特定名称服务器可能不知道域系统中所有可用的类,因此它永远无法知道它是否对所有类都是权威的。因此,对 QCLASS=* 查询的响应可以

永远不要权威。

3.7.2.反向查询（可选）

名称服务器还可以支持将特定资源映射到域名或具有该资源的域名的反向查询。例如,虽然标准查询可能将域名映射到 SOA RR,但相应的反向查询可能会将 SOA RR 映射回域

姓名。

此服务的实现在名称服务器中是可选的,但所有名称服务器必须至少能够理解反向查询消息并返回未实现的错误响应。

域系统不能保证反向查询的完整性或唯一性,因为域系统是按域名组织的,而不是按主机地址或任何其他资源类型组织的。反向查询主要用于调试和数据库维护活动。

反向查询可能不会返回正确的 TTL,并且不会指示识别的 RR 是一组中的一个的情况（例如,一个地址对应具有多个地址的主机）。因此,永远不要缓存反向查询中返回的 RR。

反向查询不是将主机地址映射到主机名的可接受方法;请改用 IN-ADDR.ARPA 域。

[RFC-1035] 中包含对反向查询的详细讨论。

3.8.状态查询（实验）

被定义为。

3.9.完成查询（过时）

RFC 882 和 883 中描述的可选完成服务已被删除。重新设计的服务将来可能会可用,或者操作码可能会被回收用于其他用途。

4. 名称服务器

4.1.介绍

名称服务器是构成域数据库的信息存储库。数据库被分成称为区域的部分,这些部分分布在名称服务器之间。虽然名称服务器可以具有多种可选功能和数据源,但名称服务器的基本任务是使用其区域中的数据回答查询。通过设计,

名称服务器可以以简单的方式回答查询;响应总是可以仅使用本地数据生成,并且包含问题的答案或对“更接近”所需信息的其他名称服务器的引用。

一个给定的区域将可从多个名称服务器获得,以确保其在主机或通信链路出现故障时的可用性。根据管理法令,我们要求每个区域至少在两台服务器上可用,并且许多区域的冗余度比这更高。

给定的名称服务器通常会支持一个或多个区域,但这只为其提供了有关域树一小部分的权威信息。它也可能有一些缓存的关于树的其他部分的非权威数据。名称服务器标记其对查询的响应,以便请求者可以判断响应是否来自权威数据。

4.2.数据库如何划分区域

域数据库以两种方式分区:按类分区,以及按在节点之间的名称空间中进行的“切割”。

类划分很简单。任何类的数据库都与所有其他类分开组织、委托和维护。由于按照惯例,所有类的名称空间都是相同的,因此可以将单独的类视为并行名称空间树的数组。请注意,对于这些不同的并行类,附加到节点的数据将有所不同。创建新类的最常见原因是现有类型需要新的数据格式,或者需要现有名称空间的单独管理版本。

在一个类中,可以在任何两个相邻节点之间对名称空间进行“切割”。完成所有切割后,每组连接的名称空间都是一个单独的区域。据说该区域对连接区域中的所有名称具有权威性。请注意,对于不同的类,名称空间中的“切割”可能在不同的地方,名称服务器可能不同,等等。

这些规则意味着每个区域至少有一个节点,因此具有权威的域名,并且特定区域中的所有节点都是连接的。给定树结构,每个区域都有一个最高节点,该节点比该区域中的任何其他节点更接近根。此节点的名称通常用于标识区域。

尽管不是特别有用,但可以对名称空间进行分区,使每个域名都在一个单独的区域中,或者所有节点都在一个区域中。相反,数据库在特定组织想要接管控制的位置进行分区

一个子树。一旦组织控制了自己的区域,它就可以单方面更改区域中的数据、增加连接到区域的新树部分、删除现有节点或在其区域下委派新的子区域。

如果组织有子结构,它可能希望进行进一步的内部分区以实现名称空间控制的嵌套委派。

在某些情况下,进行这样的划分纯粹是为了方便数据库维护。

4.2.1.技术考虑

描述区域的数据有四个主要部分:

- 区域内所有节点的权威数据。

- 定义区域顶部节点的数据(可以认为
作为权威数据的一部分)。

- 描述委托子区域的数据,即围绕区域底部进行切割。

- 允许访问子区域名称服务器的数据
(有时称为“胶水”数据)。

所有这些数据都以 RR 的形式表示,因此一个区域可以完全用一组 RR 来描述。整个区域可以通过传输 RR 在名称服务器之间传输,可以在一系列消息中携带,也可以通过 FTP 传输作为文本表示的主文件。

区域的权威数据只是附加到所有节点的所有 RR,从区域的顶部节点向下到叶节点或区域底部边缘周围切割上方的节点。

虽然逻辑上是权威数据的一部分,但描述区域顶级节点的资源记录对于区域的管理尤为重要。这些 RR 有两种类型:列出区域的所有服务器的名称服务器 RR,每个 RR 一个,以及描述区域管理参数的单个 SOA RR。

描述围绕区域底部的切割的 RR 是为子区域命名服务器的 NS RR。由于切割是在节点之间进行的,因此这些 RR 不是区域权威数据的一部分,并且应该与子区域顶部节点中的相应 RR 完全相同。由于名称服务器总是与区域边界相关联,因此 NS RR 仅在某些区域的顶级节点处找到。在构成一个区域的数据中,NS RR 位于该区域的顶部节点

区域（并且是权威的)和围绕区域底部的切口（它们不是权威的），但从不在两者之间。

区域结构的目标之一是任何区域都拥有与任何子区域的名称服务器建立通信所需的所有数据。也就是说，父区域具有访问其子区域的服务器所需的所有信息。为子区域命名服务器的 NS RR 通常不足以完成此任务，因为它们命名服务器，但不提供它们的地址。特别是，如果名称服务器的名称本身在子区域中，我们可能会遇到 NS RR 告诉我们为了了解名称服务器的地址，我们应该使用我们希望的地址联系服务器的情况学习。要解决此问题，区域包含“胶水”RR，它们不是权威数据的一部分，并且是服务器的地址 RR。

这些 RR 仅在名称服务器的名称“低于”切割时才是必需的，并且仅用作推荐响应的一部分。

4.2.2.行政考虑

当一些组织想要控制自己的域时，第一步是确定合适的父区域，并让父区域的所有者同意控制委派。虽然没有特定的技术限制来处理可以在树中的哪个位置完成此操作，但 [RFC-1032] 中讨论了一些处理顶级组织的管理分组，中级区域可以自由创建自己的规则。例如，一所大学可能选择使用单个区域，而另一所大学可能选择按专用于各个部门或学校的子区域组织。 [RFC-1033] 列出了可用的 DNS 软件并讨论了管理程序。

一旦为新的子区域选择了正确的名称，新的所有者应该被要求证明冗余名称服务器支持。请注意，不要求区域的服务器位于在该域中具有名称的主机中。在许多情况下，如果一个区域的服务器分布广泛，而不是位于由管理该区域的同一组织控制的物理设施内，那么整个区域将更容易访问 Internet。例如，在当前的 DNS 中，英国或 UK 域的名称服务器之一位于美国。这允许美国主机在不使用有限的跨大西洋带宽的情况下获取英国数据。

作为最后的安装步骤，应该将使委托生效所必需的委托 NS RR 和粘合 RR 添加到父区域。两个区域的管理员应确保标记切割两侧的 NS 和粘合 RR 一致并保持一致。

4.3.名称服务器内部结构

4.3.1.查询和响应

名称服务器的主要活动是回答标准查询。
查询及其响应都以 [RFC-1035] 中描述的标准消息格式携带。该查询包含 QTYPE、QCLASS 和 QNAME,它们描述了所需信息的类型和类别以及感兴趣的名称。

名称服务器回答查询的方式取决于它是否在递归模式下运行:

- 服务器最简单的模式是非递归的,因为它
  可以仅使用本地信息回答查询:响应包含错误、答案或对“更接近”答案的其他服务器的引用。所有名称服务器都必须实现非递归查询。

- 客户端最简单的模式是递归的,因为在这种模式下名称服务器充当解析器的角色并返回错误或答案,但从不返回引用。

  该服务在名称服务器中是可选的,名称服务器也可以选择限制可以使用递归模式的客户端。

递归服务在以下几种情况下很有用:

- 一个相对简单的请求者,除了直接回答问题外,没有其他能力。

- 需要跨越协议或其他边界并可以发送到充当中介的服务器的请求。

- 我们希望集中缓存而不是网络
  为每个客户端都有一个单独的缓存。

如果请求者能够进行转介并且对有助于未来请求的信息感兴趣,则非递归服务是合适的。

递归模式的使用仅限于客户端和名称服务器都同意使用的情况。该协议是通过在查询和响应消息中使用两个位来协商的:

- 递归可用或 RA 位由名称服务器在所有响应中设置或清除。如果名称服务器愿意为客户端提供递归服务,则该位为真,无论客户端是否请求递归服务。

  也就是说,RA 表示可用性而不是使用。

- 查询包含一个称为所需递归或 RD 的位。这
  位指定指定请求者是否希望为该查询提供递归服务。客户端可以从任何名称服务器请求递归服务,尽管它们应该依赖于仅从先前
  发送过 RA 的服务器或同意通过私有协议或 DNS 协议之外的其他方式提供服务的服务器接收服务。

当带有 RD 集的查询到达愿意提供递归服务的服务器时,就会出现递归模式;客户端可以通过检查回复中是否设置了 RA 和 RD 来验证是否使用了递归模式。请注意,除非通过 RD 请求,否则名称服务器不应执行递归服务,因为这会干扰名称服务器及其数据库的故障排除。

如果递归服务被请求并且可用,则对查询的递归响应将是以下之一:

- 查询的答案,可能由一个或多个 CNAME RR 开头,这些 CNAME RR 指定在回答过程中遇到的别名。

- 名称错误,表示该名称不存在。这可能包括 CNAME RR,指示原始查询名称是不存在的名称的别名。

- 临时错误指示。

如果没有请求递归服务或递归服务不可用,则非递归响应将是以下之一:

- 权威名称错误,表明该名称不
  存在。

- 临时错误指示。

- 一些组合:

  回答问题的 RR,以及数据是来自区域还是缓存的指示。

  对名称服务器的引用,这些服务器具有比发送回复的服务器更接近名称祖先的区域。

- 名称服务器认为对
  请求者。

4.3.2.算法

名称服务器使用的实际算法将取决于本地操作系统和用于存储 RR 的数据结构。下面的算法假设 RR 被组织成几个树结构,一个用于每个区域,另一个用于缓存:

1.根据名称服务器是否愿意提供递归服务,设置或清除响应中递归可用的值。如果递归服务可用并通过查询中的 RD 位请求,则转到步骤 5,否则转到步骤 2。

2. 在可用区域中搜索与 QNAME 最近的祖先区域。如果找到这样的区域,则转到第 3 步,否则转到第 4 步。

3. 开始在区域中逐个标签向下匹配。匹配过程可以通过几种方式终止:

A。如果整个 QNAME 都匹配,我们就找到了节点。

如果节点上的数据是CNAME,并且QTYPE不匹配CNAME,则将CNAME RR复制到响应的答案部分,将QNAME更改为CNAME RR中的规范名称,然后返回步骤1。

否则,将所有匹配 QTYPE 的 RR 复制到答案部分,然后转到步骤 6。

b.如果匹配将使我们脱离权威数据,我们就有了推荐。当我们遇到一个带有 NS RRs 标记沿着 a 底部的切割的节点时,就会发生这种情况区。

将子区域的 NS RR 复制到回复的权限部分。将任何可用的地址放入附加部分,如果地址无法从权威数据或缓存中获得,则使用胶水 RR。转到步骤 4。

C。如果在某个标签上,匹配是不可能的(即对应的标签不存在),查看“*”标签是否存在。

如果“*”标签不存在,则检查我们要查找的名称是否为查询中的原始QNAME

或我们因 CNAME 而遵循的名称。如果名称是原始名称,则在响应中设置权威名称错误并退出。否则就退出。

如果“*”标签确实存在,则将该节点的 RR 与 QTYPE 进行匹配。如果有任何匹配,将它们复制到答案部分,但将 RR 的所有者设置为 QNAME,而不是带有“*”标签的节点。转到步骤 6。

4.在缓存中开始向下匹配。如果在
缓存,将所有附加到它的匹配 QTYPE 的 RR 复制到答案部分。如果没有来自权威数据的委托,从缓存中寻找最好的,放在权威部分。转到步骤 6。

5. 使用本地解析器或其算法的副本(参见
本备忘录的解析器部分)来回答查询。将结果(包括任何中间 CNAME)存储在响应的答案部分。

6. 仅使用本地数据,尝试添加可能对查询的附加部分有用的其他 RR。出口。

4.3.3.通配符

在之前的算法中,对于所有者名称以标签“*”开头的 RR 进行了特殊处理。这样的 RR 称为通配符。

可以将通配符 RR 视为合成 RR 的指令。
当满足适当的条件时,名称服务器创建 RR,其所有者名称等于查询名称和从通配符 RR 中获取的内容。

此功能最常用于创建一个区域,该区域将用于将邮件从 Internet 转发到其他邮件系统。一般的想法是,在查询中呈现给服务器的该区域中的任何名称都将被假定存在,并具有某些属性,除非存在相反的明确证据。请注意,这里使用术语区域而不是域是有意的;这样的默认值不会跨区域边界传播,尽管子区域可以选择通过设置类似的默认值来实现这种外观。

通配符 RR 的内容遵循 RR 的通常规则和格式。区域中的通配符有一个所有者名称,用于控制它们将匹配的查询名称。通配符 RR 的所有者名称的格式为“*.<anydomain>”,其中 <anydomain> 是任何域名。 <anydomain> 不应包含其他 * 标签,并且应在区域的权威数据中。通配符可能适用于 <anydomain> 的后代,但不适用于 <anydomain> 本身。其他方式

看看这个是“*”标签总是匹配至少一个完整的标签,有时甚至更多,但总是匹配整个标签。

通配符 RR 不适用:

- 当查询在另一个区域时。即委托取消
  通配符默认值。

- 当查询名称或通配符域与查询名称之间的名称已知存在时。例如,如果通配符 RR 的所有者名称为“*.X”,并且该区域还包含附加到 BX 的 RR,则
  通配符将适用于名称 ZX 的查询(假设没有关于 ZX 的明确信息),但不是到 BX.ABX 或 X。

出现在查询名称中的 * 标签没有特殊作用,但可用于测试权威区域中的通配符;这样的查询是获得包含所有者名称带有 * 的 RR 的响应的唯一方法。不应缓存此类查询的结果。

请注意,通配符 RR 的内容在用于合成 RR 时不会被修改。

为了说明通配符 RR 的使用,假设一家拥有大型非 IP/TCP 网络的大公司想要创建一个邮件网关。如果公司名为 X.COM,并且支持 IP/TCP 的网关机器名为 AXCOM,则可能会在 COM 区域中输入以下 RR:

X.COM	MX	10	雅康
*.X.COM	MX	10	雅康
雅康	A	1.2.3.4 10	
雅康	MX		雅康
*.AXCOM	MX	10	雅康

这将导致对以 X.COM 结尾的任何域名的任何 MX 查询返回指向 AXCOM 的 MX RR。需要两个通配符 RR,因为通配符在 *.X.COM 的作用在 AXCOM 子树中被 AXCOM 的显式数据抑制。另请注意,X.COM 和 AXCOM 的显式 MX 数据是必需的,并且上述 RR 均不匹配 XX.COM 的查询名称。

4.3.4.否定响应缓存(可选)

DNS 提供了一项可选服务,该服务允许名称服务器使用 TTL 分发负面结果,并允许解析器缓存负面结果。为了

例如,名称服务器可以分发 TTL 以及名称错误指示,并且允许接收此类信息的解析器假设名称在 TTL 期间不存在,而无需咨询权威数据。同样,解析器可以使用匹配多种类型的 QTYPE 进行查询,并缓存某些类型不存在的事实。

此功能在实现使用搜索列表的命名速记的系统中尤为重要,因为流行的速记恰好需要在搜索列表末尾添加后缀,无论何时使用它都会产生多个名称错误。

该方法是当响应是权威的时,名称服务器可以将 SOA RR 添加到响应的附加部分。 SOA 必须是作为答案部分中权威数据来源的区域的 SOA,或者名称错误(如果适用)。 SOA 的 MINIMUM 字段控制否定结果可能被缓存的时间长度。

请注意,在某些情况下,答案部分可能包含多个所有者姓名。在这种情况下,SOA机制应该只用于匹配QNAME的数据,这是本节中唯一的权威数据。

名称服务器和解析器永远不应尝试将 SOA 添加到非权威响应的附加部分,或尝试推断未在权威响应中直接说明的结果。

这有几个原因,包括:缓存的信息通常不足以匹配 RR 及其区域名称,SOA RR 可能由于直接 SOA 查询而被缓存,并且名称服务器不需要在权限部分输出 SOA .

此功能是可选的,尽管经过改进的版本有望在未来成为标准协议的一部分。名称服务器不需要在所有权威响应中添加 SOA RR,解析器也不需要缓存否定结果。两者都推荐。

当 SOA RR 出现在响应中时,所有解析器和递归名称服务器都需要至少能够忽略它。

还提出了一些将使用此功能的实验。
这个想法是,如果已知缓存数据来自特定区域,并且如果获得了该区域 SOA 的权威副本,并且如果该区域的 SERIAL 自数据被缓存以来没有更改,则缓存数据的 TTL 可以如果较小,则重置为区域最小值。

提及此用法仅用于规划目的,目前不推荐使用。

4.3.5.区域维护和传输

区域管理员的部分工作是在对该区域具有权威性的所有名称服务器上维护区域。当进行不可避免的更改时,必须将它们分发到所有名称服务器。虽然可以使用 FTP 或其他一些临时程序来完成此分发,但首选方法是 DNS 协议的区域传输部分。

自动区域传输或刷新的一般模型是名称服务器之一是区域的主服务器或主要服务器。更改在主服务器上进行协调,通常是通过编辑区域的主文件来进行。编辑后,管理员向主服务器发出信号以加载新区域。该区域的其他非主服务器或辅助服务器定期检查更改(以可选择的时间间隔)并在进行更改时获取新的区域副本。

要检测更改,辅助节点只需检查区域的 SOA 的 SERIAL 字段。除了所做的任何其他更改之外,只要对该区域进行任何更改,该区域的 SOA 中的 SERIAL 字段总是会被推进。推进可以是简单的增量,或者可以基于主文件的写入日期和时间等。目的是可以通过比较序列号来确定一个区域的两个副本中哪个副本更新。序列号推进和比较使用序列空间算法,因此理论上可以限制区域的更新速度,基本上旧副本必须在序列号覆盖其 32 位范围的一半之前消失。在实践中,唯一需要关注的是比较操作正确地处理了最正数和最负数 32 位数字之间的边界周围的比较。

辅助服务器的定期轮询由区域的 SOA RR 中的参数控制,这些参数设置了可接受的最小轮询间隔。这些参数称为 REFRESH、RETRY 和 EXPIRE。每当一个新区域加载到次要区域时,次要区域会等待 REFRESH 秒,然后再与主要区域检查新的序列号。

如果无法完成此检查,则每 RETRY 秒开始新的检查。该检查是对区域的 SOA RR 的主要查询的简单查询。如果次要区域副本中的序列字段等于主要返回的序列,则没有发生任何更改,重新开始 REFRESH 间隔等待。如果辅助发现无法对 EXPIRE 间隔执行串行检查,则它必须假定其区域副本已过时并丢弃它。

当轮询显示区域已更改时,辅助服务器必须通过区域的 AXFR 请求来请求区域传输。 AXFR 可能会导致错误,例如被拒绝,但通常由一系列响应消息来回答。第一条和最后一条消息必须包含

区域顶级权威节点的数据。中间消息携带来自该区域的所有其他 RR,包括权威和非权威 RR。消息流允许辅助节点构建区域的副本。由于准确性至关重要,因此 AXFR 请求必须使用 TCP 或其他一些可靠的协议。

每个辅助服务器都需要对主服务器执行以下操作,但也可以选择对其他辅助服务器执行这些操作。当由于主机停机或网络问题导致主服务器不可用时,或者当辅助服务器对“中间”辅助服务器的网络访问比对主服务器具有更好的网络访问时,此策略可以改进传输过程。

5. 解析器

5.1.介绍

解析器是将用户程序连接到域名服务器的程序。在最简单的情况下,解析器以子程序调用、系统调用等形式接收来自用户程序(例如,邮件程序、TELNET、FTP)的请求,并以与本地主机兼容的形式返回所需信息数据格式。

解析器与请求解析器服务的程序位于同一台机器上,但它可能需要咨询其他主机上的名称服务器。因为解析器可能需要咨询多个名称服务器,或者可能在本地缓存中有请求的信息,所以解析器完成所需的时间可能会有很大差异,从几毫秒到几秒不等。

解析器的一个非常重要的目标是通过从其先前结果的缓存中回答它们来消除大多数请求的网络延迟和名称服务器负载。因此,由多个进程、用户、机器等共享的缓存比非共享缓存更有效。

5.2.客户端解析器接口

5.2.1.典型功能

解析器的客户端接口受本地主机约定的影响,但典型的解析器-客户端接口具有三个功能:

1. 主机名到主机地址的转换。

   这个函数通常被定义为模仿以前的 HOSTS.TXT

基于功能。给定一个字符串,调用者想要一个或多个 32 位 IP 地址。在 DNS 下,它转换为对类型 A RR 的请求。由于 DNS 不保留 RR 的顺序,如果服务仅向客户端返回一个选择,则此函数可以选择对返回的地址进行排序或选择“最佳”地址。请注意,建议返回多个地址,但单个地址可能是模拟先前 HOSTS.TXT 服务的唯一方法。

2.主机地址到主机名的翻译

这个函数通常会沿用前面函数的形式。给定一个 32 位 IP 地址,调用者需要一个字符串。 IP 地址的八位字节被反转,用作名称组件,并以“IN-ADDR.ARPA”为后缀。类型 PTR 查询用于获取具有主机主要名称的 RR。例如,请求对应于 IP 地址 1.2.3.4 的主机名查找域名“4.3.2.1.IN-ADDR.ARPA”的 PTR RR。

3.通用查找功能

该功能从 DNS 中检索任意信息,并且在以前的系统中没有对应的功能。调用者提供 QNAME、QTYPE 和 QCLASS,并需要所有匹配的 RR。该函数通常对所有 RR 数据而不是本地主机的数据使用 DNS 格式,并返回所有 RR 内容（例如,TTL)而不是使用本地引用约定的处理形式。

当解析器执行指定的功能时,它通常会将以下结果之一传递回客户端：

- 一个或多个 RR 提供请求的数据。

  在这种情况下,解析器以适当的格式返回答案。

- 名称错误（NE）。

  当引用的名称不存在时会发生这种情况。例如,用户可能输入了错误的主机名。

- 未找到数据错误。

  当引用的名称存在但适当类型的数据不存在时,就会发生这种情况。例如,主机地址

应用于邮箱名称的函数将返回此错误,因为该名称存在,但地址 RR 不存在。

重要的是要注意,在主机名和地址之间进行转换的函数可能会将“名称错误”和“未找到数据”错误条件组合成单一类型的错误返回,但一般函数不应该这样。这样做的一个原因是,应用程序可能会首先询问有关名称的一种类型的信息,然后再向同一名称请求其他类型的信息;如果这两个错误结合在一起,那么无用的查询可能会降低应用程序的速度。

5.2.2.别名

在尝试解析特定请求时,解析器可能会发现所讨论的名称是别名。例如,当解析器找到 CNAME RR 时,解析器可能会发现为主机名到地址转换提供的名称是别名。如果可能,别名条件应该从解析器发回信号给客户端。

在大多数情况下,解析器在遇到 CNAME 时只是简单地以新名称重新启动查询。但是,在执行一般功能时,当 CNAME RR 与查询类型匹配时,解析器不应追求别名。这允许询问是否存在别名的查询。

例如,如果查询类型是 CNAME,则用户对 CNAME RR 本身感兴趣,而不是它指向的名称的 RR。

别名可能会出现几种特殊情况。由于效率低下,应避免使用多级别名,但不应将其视为错误。别名循环和指向不存在名称的别名应该被捕获,并将错误情况传回给客户端。

5.2.3.临时故障

在一个不完美的世界中,所有解析器有时都无法解决特定请求。这种情况可能是由于链接故障或网关问题导致解析器与网络的其余部分分离,或者由于特定域的所有服务器同时发生故障或不可用而导致的。

重要的是,不应将这种情况作为名称或数据不存在错误来通知应用程序。这种行为对人类来说很烦人,并且在邮件系统使用 DNS 时会造成严重破坏。

虽然在某些情况下可以通过无限期地阻止请求来处理这样的临时问题,但这通常不是一个好的选择,特别是当客户端是一个服务器进程可以继续

其他任务。推荐的解决方案是始终将临时故障作为解析器功能的可能结果之一,即使这可能会使现有 HOSTS.TXT 功能的模拟更加困难。

5.3.解析器内部

每个解析器实现都使用略有不同的算法,并且通常会花费比典型事件更多的逻辑来处理各种错误。本节概述了解析器操作的推荐基本策略,但将细节留给 [RFC-1035]。

5.3.1.存根解析器

实现解析器的一种选择是将解析功能移出本地机器并移入支持递归查询的名称服务器。这可以提供一种在缺乏执行解析器功能的资源的 PC 中提供域服务的简单方法,或者可以集中整个本地网络或组织的缓存。

剩下的存根所需要的只是一个将执行递归请求的名称服务器地址列表。这种类型的解析器可能需要配置文件中的信息,因为它可能缺乏在域数据库中定位它的复杂性。

用户还需要验证列出的服务器是否会执行递归服务;名称服务器可以自由拒绝为任何或所有客户端执行递归服务。用户应咨询本地系统管理员以找到愿意执行该服务的名称服务器。

这种类型的服务有一些缺点。由于递归请求可能需要任意时间来执行,存根可能难以优化重传间隔以处理丢失的 UDP 数据包和死服务器;如果名称服务器将重新传输解释为新请求,则名称服务器很容易因过于狂热的存根而过载。使用 TCP 可能是一个答案,但 TCP 可能会给主机的能力带来负担,这与真正的解析器的能力相似。

5.3.2.资源

除了自己的资源外,解析器还可以共享访问由本地名称服务器维护的区域。这为解析器提供了更快速访问的优势,但解析器必须小心,不要让缓存的信息覆盖区域数据。在本次讨论中,术语“本地信息”是指缓存和此类共享区域的结合,理解为

当两者都存在时,总是优先使用权威数据而不是缓存数据。

以下解析器算法假定所有函数都已转换为通用查找函数,并使用以下数据结构来表示解析器中正在进行的请求的状态:

折断	我们正在搜索的域名。
支持	搜索请求的 QTYPE。
类	搜索请求的 QCLASS。
单列表	描述名称服务器和解析器当前尝试查询的区域的结构。 该结构跟踪解析器当前对哪些名称服务器持有所需信息的最佳猜测;当到达的信息改变猜测时,它会更新。该结构包括区域名称的等效项、区域的已知名称服务器、名称服务器的已知地址以及可用于建议下一个可能是最佳服务器的历史信息。等效区域名称是从根向下的标签数量的匹配计数,SNAME 与被查询的区域共有;这用于衡量解析器与 SNAME 的“接近”程度。
皮带	与 SLIST 形式相同的“安全带”结构,它从配置文件初始化,并列出当解析器没有任何本地信息来指导名称服务器选择时应该使用的服务器。匹配计数将为 -1,表示没有已知匹配的标签。
缓存	存储先前响应结果的结构。由于解析器负责丢弃 TTL 已过期的旧 RR,因此当 RR 存储在缓存中时,大多数实现会将到达的 RR 中指定的间隔转换为某种绝对时间。解析器不会单独对 TTL 进行计数,而是在搜索过程中遇到旧 RR 时忽略或丢弃它们,或者在定期扫描期间丢弃它们以回收旧 RR 消耗的内存。

5.3.3.算法

顶层算法有四个步骤：

1.查看本地信息中是否有答案,如果有则返回给客户端。

2. 找到最好的服务器来询问。

3. 向他们发送查询,直到有人回复为止。

4. 分析响应,或者：

A。如果响应回答了问题或包含名称错误,则缓存数据并将其返回给客户端。

b.如果响应包含对其他服务器的更好委托,则缓存委托信息,然后转到步骤 2。

C。如果响应显示 CNAME 而不是
自己回答,缓存 CNAME,将 SNAME 更改为 CNAME RR 中的规范名称,然后转到步骤 1。

d.如果响应显示服务器故障或其他
奇怪的内容,从 SLIST 中删除服务器并返回步骤 3。

第 1 步在缓存中搜索所需数据。如果数据在缓存中,则假定它足以正常使用。一些解析器在用户界面上有一个选项,这将强制解析器忽略缓存的数据并咨询权威服务器。不建议将其作为默认值。如果解析器可以直接访问名称服务器的区域,它应该检查所需数据是否以权威形式存在,如果是,则优先使用权威数据而不是缓存数据。

第 2 步查找名称服务器以请求所需数据。一般策略是查找本地可用的名称服务器 RR,从 SNAME 开始,然后是 SNAME 的父域名、祖父域名,依此类推直至根。因此,如果 SNAME 是 Mockapetris.ISI.EDU,则此步骤将查找 Mockapetris.ISI.EDU 的 NS RR,然后是 ISI.EDU,然后是 EDU,然后是 。 （根）。

这些 NS RR 列出了 SNAME 或 SNAME 以上区域的主机名。将名称复制到 SLIST 中。使用本地数据设置他们的地址。可能是地址不可用的情况。解析器在这里有很多选择;最好的办法是启动并行解析器进程

的地址,同时继续使用可用的地址。显然,设计选择和选项很复杂,并且取决于本地主机的能力。为解析器设计者推荐的优先级是:

1.绑定工作量(发送的数据包,并行进程
已启动),这样即使有人错误地配置了某些数据,请求也不会进入无限循环或开始请求或查询与其他实现的连锁反应。

2. 尽可能得到答复。

3. 避免不必要的传输。

4. 尽快得到答案。

如果搜索 NS RR 失败,则解析器从安全带 SBELT 初始化 SLIST。基本思想是,当解析器不知道要询问哪些服务器时,它应该使用来自配置文件的信息,该配置文件列出了几台预计会有帮助的服务器。

虽然有特殊情况,但通常的选择是两台根服务器和两台主机域服务器。每个两个的原因是为了冗余。根服务器将提供对所有域空间的最终访问。如果本地网络由于网关或链接故障而与 Internet 隔离,这两个本地服务器将允许解析器继续解析本地名称。

除了服务器的名称和地址之外,还可以对 SLIST 数据结构进行排序以首先使用最好的服务器,并确保以循环方式使用所有服务器的所有地址。排序可以是一个简单的功能,即优先选择本地网络上的地址而不是其他地址,或者可能涉及过去事件的统计数据,例如以前的响应时间和平均成功率。

第 3 步发出查询,直到收到响应。该策略是循环所有服务器的所有地址,并在每次传输之间设置超时。在实践中,使用多宿主主机的所有地址很重要,过于激进的重传策略实际上会在多个解析器争用同一名称服务器甚至偶尔为单个解析器使用时减慢响应速度。 SLIST 通常包含数据值来控制超时并跟踪以前的传输。

第 4 步涉及分析响应。解析器在解析响应时应该高度偏执。它还应检查响应是否与它使用响应中的 ID 字段发送的查询相匹配。

理想的答案是来自对查询具有权威性的服务器,它要么提供所需的数据,要么提供名称错误。如果 TTL 大于零,数据将传回给用户并输入缓存以备将来使用。

如果响应显示委托,解析器应检查委托是否比 SLIST 中的服务器“更接近”答案。这可以通过将 SLIST 中的匹配计数与从 SNAME 和委托中的 NS RR 计算的匹配计数进行比较来完成。如果不是,则回复是虚假的,应忽略。如果授权有效,NS 授权 RR 和服务器的任何地址 RR 都应该被缓存。

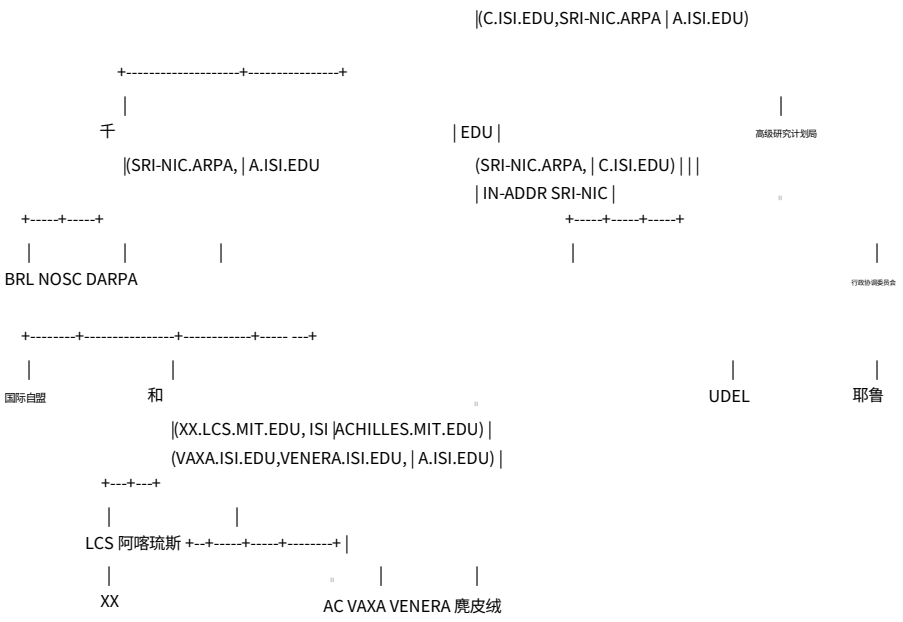
将名称服务器输入到 SLIST 中,然后重新开始搜索。

如果响应包含 CNAME,搜索将在 CNAME 处重新开始,除非响应包含规范名称的数据或者 CNAME 本身就是答案。

细节和实现提示可以在 [RFC-1035] 中找到。

6. 场景

在我们的示例域空间中,假设我们需要对根、MIL、EDU、MIT.EDU 和 ISI.EDU 区域进行单独的管理控制。我们可能会按如下方式分配名称服务器:



在此示例中,权威名称服务器显示在域树中承担控制权的点的括号中。

因此,根名称服务器位于 C.ISI.EDU、SRI-NIC.ARPA 和 A.ISI.EDU 上。 MIL 域由 SRI-NIC.ARPA 和 A.ISI.EDU 提供服务。 EDU 域由 SRI-NIC.ARPA 提供服务。和 C.ISI.EDU。请注意,服务器可能具有连续或不相交的区域。在这种情况下,C.ISI.EDU 在根域和 EDU 域中具有连续的区域。 A.ISI.EDU 在根域和 MIL 域中有连续区域,但在 ISI.EDU 中也有非连续区域。

6.1. C.ISI.EDU 名称服务器

C.ISI.EDU 是 IN 类的根域、MIL 域和 EDU 域的名称服务器,并且将具有这些域的区域。根域的区域数据可能是:

```
.                   在      面向服务架构    SRI-NIC.ARPA。 HOSTMASTER.SRI-NIC.ARPA。 ( 870611 ;序列号 1800 ;每 30 分钟刷新
                                            一次 300 ;每 5 分钟重试一次 604800 ;一周后到期 86400) ;最少一天 A.ISI.EDU。

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

千。                86400 国民服役          SRI-NIC.ARPA。
                    86400 国民服役          A.ISI.EDU。

教育。              86400 国民服役          SRI-NIC.ARPA。
                    86400 国民服役          C.ISI.EDU。

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

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

USC-ISIC.ARPA。 CNAME C.ISI.EDU。

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

```
103.0.3.26.IN-ADDR.ARPA。 PTR A.ISI.EDU。

A.ISI.EDU。 86400 A C.ISI.EDU。 26.3.0.103
86400 一个 10.0.0.52
```

此数据的表示方式与主文件中的方式相同。大多数 RR 是单行条目;这里唯一的例外是 SOA RR,它使用“(”开始多行 RR,使用 “)”显示多行 RR 的结尾。

由于一个zone 中所有RR 的类必须相同,因此只有一个zone 中的第一个RR 需要指定类。当名称服务器加载区域时,它会强制所有权威 RR 的 TTL 至少为 SOA 的 MINIMUM 字段,此处为 86400 秒或一天。标记 MIL 和 EDU 域授权的 NS RR 以及服务器主机地址的粘合 RR 不是区域中权威数据的一部分,因此具有明确的 TTL。

四个 RR 附加到根节点:描述根区域的 SOA 和列出根名称服务器的 3 个 NS RR。 SOA RR 中的数据描述了区域的管理。区域数据在主机 SRI-NIC.ARPA 上维护,区域的负责方是 HOSTMASTER@SRI-NIC.ARPA。 SOA 中的一个关键项是 86400 秒最小 TTL,这意味着该区域中的所有权威数据至少具有该 TTL,尽管可能会明确指定更高的值。

MIL 和 EDU 域的 NS RR 标记了根区域与 MIL 和 EDU 区域之间的边界。请注意,在此示例中,较低区域恰好受到也支持根区域的名称服务器的支持。

EDU 区域的主文件可以相对于原始 EDU 进行说明。 EDU 域的区域数据可能是:

```
教育。在 SOA SRI-NIC.ARPA 中。 HOSTMASTER.SRI-NIC.ARPA。 （
                                870729 ;序列号 1800 ;每
                                30 分钟刷新一次 300 ;每 5 分钟重试一次 604800 ;一周后到
                                期 86400 ;最少一天）

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

UCI 172800 NS ICS.UCI 172800
                    NS ROME.UCI ICS.UCI
172800 A 192.5.19.1 ROME.UCI 172800 A 192.5.19.31
```

```
ISI 172800 NS VAXA.ISI 172800 NS
                                    A.ISI 172800 NS
                                    VENERA.ISI.EDU。
VAXA.ISI 172800 A 10.2.0.27
                                    172800 A 128.9.0.33
VENERA.ISI.EDU。 172800 A 10.1.0.52 172800 A 128.9.0.32
                                    A.CHAPTER 172800 A 26.3.0.103


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


耶鲁大学。 172800 NS YALE.ARPA。
耶鲁大学。 172800 NS YALE-BULLDOG.ARPA。

麻省理工学院。 43200 NS XX.LCS.MIT.EDU。
                                    43200 NS ACHILLES.MIT.EDU。
XX.LCS.MIT.EDU。 43200 A 10.0.0.44 ACHILLES.MIT.EDU。
43200 A 18.72.0.8
```

注意这里使用的相对名称。 ISI.EDU 的所有者名称。使用相对名称表示，名称服务器 RR 内容中的两个也是如此。

相对和绝对域名可以在主控中自由混合

6.2.示例标准查询

以下查询和响应说明了名称服务器的行为。
除非另有说明，否则查询在标头中没有所需的递归 (RD)。请注意，非递归查询的答案确实取决于被询问的服务器，但不取决于请求者的身份。

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

查询看起来像：

```
                +---------------------------------------------- --+
标头 |操作码=查询 |
                +---------------------------------------------- --+
问题 | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=A |
                +---------------------------------------------- --+
回答 | <空> |
                +---------------------------------------------- --+
权限 | <空> |
                +---------------------------------------------- --+
附加 | <空> |
                +---------------------------------------------- --+
```

C.ISI.EDU 的响应将是：

```
                +---------------------------------------------- --+
标头 |操作码=查询、响应、AA |
                +---------------------------------------------- --+
问题 | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=A |
                +---------------------------------------------- --+
回答 | SRI-NIC.ARPA。 86400 在一个 26.0.0.73 | 86400 在 10.0.0.51

                +---------------------------------------------- --+
权限 | <空> |
                +---------------------------------------------- --+
附加 | <空> |
                +---------------------------------------------- --+
```

响应的头看起来像查询的头，只是设置了 RESPONSE 位，表示此消息是响应，而不是查询，并且设置了权威答案 (AA) 位，表示地址 RRs 在答题部分均来自权威资料。响应的问题部分与查询的问题部分匹配。

如果将相同的查询发送到对 SRI-NIC.ARPA 没有权威的其他服务器,则响应可能是:

```
                +---------------------------------------------- --+
标头 |操作码=查询,响应 |
                +---------------------------------------------- --+
问题 | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=A |
                +---------------------------------------------- --+
回答 | SRI-NIC.ARPA。 1777 在 10.0.0.51 | 1777 在 26.0.0.73

                +---------------------------------------------- --+
权限 | <空> |
                +---------------------------------------------- --+
附加 | <空> |
                +---------------------------------------------- --+
```

此响应在两个方面与前一个不同:标头没有设置 AA,并且 TTL 不同。推断是数据不是来自区域,而是来自缓存。权威TTL和这里的TTL的区别是由于缓存中数据的老化。答案部分中 RR 的排序差异并不显着。

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

与前一个类似的查询,但使用 * 的 QTYPE,将从 C.ISI.EDU 收到以下响应:

```
                +---------------------------------------------- --+
标头 |操作码=查询、响应、AA |
                +---------------------------------------------- --+
问题 | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=* |
                +---------------------------------------------- --+
回答 | SRI-NIC.ARPA。 86400 在一个 | | |
                                                    26.0.0.73 10.0.0.51
                                              A     0 SRI-NIC.ARPA。
                                              MX
                                              信息 DEC-2060 TOPS20
                +---------------------------------------------- --+
权限| <空> |
                +---------------------------------------------- --+
附加 | <空> |
                +---------------------------------------------- --+
```

如果将类似的查询定向到两个对 SRI-NIC.ARPA 没有权威的名称服务器,则响应可能是:

```
                +---------------------------------------------- --+
标头 |操作码=查询,响应 |
                +---------------------------------------------- --+
问题 | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=* |
                +---------------------------------------------- --+
回答 | SRI-NIC.ARPA。 12345 英寸 | A 26.0.0.73 10.0.0.51
                                                          A
                +---------------------------------------------- --+
权限| <空> |
                +---------------------------------------------- --+
附加 | <空> |
                +---------------------------------------------- --+
```

和

```
                +---------------------------------------------- --+
标头 |操作码=查询,响应 |
                +---------------------------------------------- --+
问题 | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=* |
                +---------------------------------------------- --+
回答 | SRI-NIC.ARPA。 1290 IN HINFO DEC-2060 TOPS20 |
                +---------------------------------------------- --+
权限| <空> |
                +---------------------------------------------- --+
附加 | <空> |
                +---------------------------------------------- --+
```

这两个答案都没有AA集,所以这两个答案都来自权威数据。不同的内容和不同的 TTL 表明两台服务器在不同的时间缓存数据,第一台服务器缓存了对 QTYPE=A 查询的响应,第二台缓存了对 HINFO 查询的响应。

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

这种类型的查询可能是由于邮件程序试图查找邮件目的地 HOSTMASTER@SRI-NIC.ARPA 的路由信息而导致的。

C.ISI.EDU 的响应将是：

```
                +---------------------------------------------- --+
标头 |操作码=查询、响应、AA |
                +---------------------------------------------- --+
问题 | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=MX |
                +---------------------------------------------- --+
回答 | SRI-NIC.ARPA。 86400 英寸 MX 0 SRI-NIC.ARPA.|
                +---------------------------------------------- --+
权限| <空> |
                +---------------------------------------------- --+
附加 | SRI-NIC.ARPA。 86400 英寸 A 26.0.0.73 10.0.0.51
                | A
                +---------------------------------------------- --+
```

此响应在响应的答案部分包含 MX RR。
附加部分包含地址 RR，因为 C.ISI.EDU 的名称服务器猜测请求者将需要地址以便正确使用 MX 携带的信息。

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

C.ISI.EDU 将回复此查询：

```
                +---------------------------------------------- --+
标头 |操作码=查询、响应、AA |
                +---------------------------------------------- --+
问题 | QNAME=SRI-NIC.ARPA., QCLASS=IN, QTYPE=NS |
                +---------------------------------------------- --+
回答 | <空> |
                +---------------------------------------------- --+
权限 | <空> |
                +---------------------------------------------- --+
附加 | <空> |
                +---------------------------------------------- --+
```

响应和查询之间的唯一区别是标头中的 AA 和 RESPONSE 位。对该响应的解释是服务器对该名称具有权威性，并且该名称存在，但那里不存在 NS 类型的 RR。

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

如果用户输入错误的主机名，我们可能会看到这种类型的查询。

C.ISI.EDU 会这样回答：

```
                +---------------------------------------------- --+
标头            | OPCODE=SQUERY, RESPONSE, AA, RCODE=NE           |
                +---------------------------------------------- --+
问题 | QNAME=SIR-NIC.ARPA., QCLASS=IN, QTYPE=A                    |
                +---------------------------------------------- --+
回答            | <空>                                            |
                +---------------------------------------------- --+
权限 | |              . SOA SRI-NIC.ARPA。 HOSTMASTER.SRI-NIC.ARPA。
                          870611 1800 300 604800 86400
                +---------------------------------------------- --+
附加 | <空>                                                       |
                +---------------------------------------------- --+
```

此响应指出该名称不存在。此条件在标头的响应代码 (RCODE) 部分中发出信号。

授权部分中的 SOA RR 是可选的负缓存信息，它允许使用此响应的解析器假定该名称在 SOA MINIMUM (86400) 秒内不存在。

6.2.6. QNAME=BRL.MIL, QTYPE=A

如果将此查询发送到 C.ISI.EDU，回复将是：

```
                +---------------------------------------------- --+
标头            |操作码=查询，响应                                |
                +---------------------------------------------- --+
问题 | QNAME=BRL.MIL,QCLASS=IN,QTYPE=A                            |
                +---------------------------------------------- --+
回答            | <空>                                            |
                +---------------------------------------------- --+
权限 |军用。 |                  86400 在 NS 86400      SRI-NIC.ARPA。 | | A.ISI.EDU。
                                          NS
                +---------------------------------------------- --+
                                          A       26.3.0.103 26.0.0.73
附加 | A.ISI.EDU。                        A       10.0.0.51
                | SRI-NIC.ARPA。 |         A
                +---------------------------------------------- --+
```

此回复有一个空的回答部分，但不具有权威性，所以它是一个推荐。 C.ISI.EDU 上的名称服务器意识到它对 MIL 域没有权威，已将请求者推荐给 A.ISI.EDU 和 SRI-NIC.ARPA 上的服务器，它知道它们对 MIL 域是权威的。

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

A.ISI.EDU 对此查询的响应将是：

```
                +---------------------------------------------- --+
标头 |操作码=查询、响应、AA |
                +---------------------------------------------- --+
问题 | QNAME=USC-ISIC.ARPA., QCLASS=IN, QTYPE=A |
                +---------------------------------------------- --+
回答 | USC-ISIC.ARPA。 CNAME 中的 86400 | C.ISI.EDU。 C.ISI.EDU。
                                                  86400 在一个 10.0.0.52
                +---------------------------------------------- --+
权限 | <空> |
                +---------------------------------------------- --+
附加 | <空> |
                +---------------------------------------------- --+
```

请注意，标头中的 AA 位保证匹配 QNAME 的数据是权威的，但并没有说明 C.ISI.EDU 的数据是否是权威的。这个完整的回复是可能的，因为 A.ISI.EDU 恰好对发现 USC-ISIC.ARPA 的 ARPA 域和发现 C.ISI.EDU 数据的 ISI.EDU 域都是权威的。

如果相同的查询被发送到 C.ISI.EDU，如果它在缓存中有自己的地址，它的响应可能与上面显示的相同，但也可能是：

```
                +---------------------------------------------- --+
标头 |操作码=查询、响应、AA |
                +---------------------------------------------- --+
问题 | QNAME=USC-ISIC.ARPA., QCLASS=IN, QTYPE=A |
                +---------------------------------------------- --+
回答 | USC-ISIC.ARPA。 CNAME C.ISI.EDU 中的 86400。 |
                +---------------------------------------------- --+
权限 | ISI.EDU。 172800 印第安纳州 NS VAXA.ISI.EDU。 | | A.ISI.EDU。
                                                  NS VENERA.ISI.EDU。 |
                +---------------------------------------------- --+
附加 | VAXA.ISI.EDU。 172800 | 172800 | VENERA.ISI.EDU。 172800 | A 10.2.0.27
                 172800 | A.ISI.EDU。 172800 A 128.9.0.33 10.1.0.52
                                                  A 128.9.0.32 26.3.0.103
                                                  A
                                                  A
                +---------------------------------------------- --+
```

此回复包含对别名 USC-ISIC.ARPA 的权威回复，以及对 ISI.EDU 名称服务器的引用。这种回复不太可能，因为查询是针对被询问的名称服务器的主机名，但对于其他别名来说很常见。

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

如果将此查询发送到 A.ISI.EDU 或 C.ISI.EDU，回复将是：

```
                +---------------------------------------------- --+
标头 |操作码=查询、响应、AA |
                +---------------------------------------------- --+
问题 | QNAME=USC-ISIC.ARPA., QCLASS=IN, QTYPE=A |
                +---------------------------------------------- --+
回答 | USC-ISIC.ARPA。 CNAME 中的 86400 C.ISI.EDU。 |
                +---------------------------------------------- --+
权限 | <空> |
                +---------------------------------------------- --+
附加 | <空> |
                +---------------------------------------------- --+
```

因为 QTYPE=CNAME，CNAME RR 本身会回答查询，并且名称服务器不会尝试为 C.ISI.EDU 查找任何内容。 （附加部分可能除外。）

6.3.分辨率示例

以下示例说明解析器必须为其客户端执行的操作。我们假设解析器在没有

缓存,就像系统启动后的情况一样。我们进一步假设系统不是数据中的主机之一,并且该主机位于网络 26 上的某个位置,并且其安全带（SBELT)数据结构具有以下信息：

匹配计数 = -1 SRI-NIC.ARPA。
26.0.0.73 26.3.0.103 10.0.0.51
A.ISI.EDU。

此信息指定要尝试的服务器、它们的地址和匹配计数 -1,这表示服务器不是很接近目标。请注意，-1 不应该是一个准确的接近度度量,它只是一个值,以便算法的后期阶段可以工作。

以下示例说明了缓存的使用,因此每个示例都假定先前的请求已完成。

6.3.1.为 ISI.EDU 解析 MX。

假设对解析器的第一个请求来自本地邮件程序,它有 PVM@ISI.EDU 的邮件。然后,邮件程序可能会要求域名 ISI.EDU 的类型 MX RR。

解析器会在 ISI.EDU 的缓存中查找 MX RR,但空缓存不会有帮助。解析器会认识到它需要查询外部服务器并尝试确定要查询的最佳服务器。此搜索将为域 ISI.EDU、EDU 和根查找 NS RR。这些缓存搜索也会失败。作为最后的手段,解析器将使用来自 SBELT 的信息,将其复制到其 SLIST 结构中。

此时,解析器需要从三个可用地址中选择一个进行尝试。鉴于解析器位于网络 26 上,它应该选择 26.0.0.73 或 26.3.0.103 作为其首选。然后它会发出以下形式的查询：

```
                +---------------------------------------------- --+
标头 |操作码=查询 |
                +---------------------------------------------- --+
问题 | QNAME=ISI.EDU., QCLASS=IN, QTYPE=MX |
                +---------------------------------------------- --+
回答 | <空> |
                +---------------------------------------------- --+
权限| <空> |
                +---------------------------------------------- --+
附加 | <空> |
                +---------------------------------------------- --+
```

然后,解析器将等待对其查询的响应或超时。
如果发生超时,它将尝试不同的服务器,然后是相同服务器的不同地址,最后重试已经尝试过的地址。

它最终可能会收到来自 SRI-NIC.ARPA 的回复:

```
                +---------------------------------------------- --+
标头 |操作码=查询,响应 |
                +---------------------------------------------- --+
问题 | QNAME=ISI.EDU., QCLASS=IN, QTYPE=MX |
                +---------------------------------------------- --+
回答 | <空> |
                +---------------------------------------------- --+
权限| ISI.EDU。 172800 印第安纳州 VAXA.ISI.EDU。 | | A.ISI.EDU。
                                                NS
                                                NS VENERA.ISI.EDU.|
                +---------------------------------------------- --+
附加 | VAXA.ISI.EDU。 172800 | 172800 | VENERA.ISI.EDU。 172800 | A 10.2.0.27
                  172800 | A.ISI.EDU。 172800 A 128.9.0.33 10.1.0.52
                                                A 128.9.0.32 26.3.0.103
                                                A
                                                A
                +---------------------------------------------- --+
```

解析器会注意到响应中的信息比现有的 SLIST 更接近 ISI.EDU(因为它匹配三个标签)。然后,解析器将缓存此响应中的信息并使用它来设置新的 SLIST:

```
匹配计数 = 3
                                26.3.0.103 A.ISI.EDU。
VAXA.ISI.EDU。 10.2.0.27 维内拉.ISI.EDU。 10.1.0.52 128.9.0.33 128.9.0.32
```

A.ISI.EDU 出现在这个列表以及上一个列表中,但这纯属巧合。解析器将再次开始传输并等待响应。最终它会得到一个答案:

```
                    +---------------------------------------------- --+
标头                |操作码=查询、响应、AA                                |
                    +---------------------------------------------- --+
问题 | QNAME=ISI.EDU., QCLASS=IN, QTYPE=MX                             |
                    +---------------------------------------------- --+
回答                | ISI.EDU。 |                  MX 10 VENERA.ISI.EDU。
                                                   MX 20 VAXA.ISI.EDU。
                    +---------------------------------------------- --+
权限| <空>                                                             |
                    +---------------------------------------------- --+
附加 | VAXA.ISI.EDU。 172800 一个 10.2.0.27 | 172800 一个 128.9.0.33 | VENERA.ISI.EDU。 172800 A
                      10.1.0.52 | 172800 128.9.0.32


                    +---------------------------------------------- --+
```

解析器会将此信息添加到其缓存中,并将 MX RR 返回给其客户端。

6.3.2.获取地址 26.6.0.65 的主机名

解析器会将其转换为对 65.0.6.26.IN-ADDR.ARPA 的 PTR RR 请求。此信息不在缓存中,因此解析器会寻找外部服务器进行询问。没有服务器会匹配,
所以它会再次使用 SBELT。 （请注意,ISI.EDU 域的服务器在缓存中,但 ISI.EDU 不是 65.0.6.26.IN-ADDR.ARPA 的祖先,因此使用 SBELT。）

由于这个请求在两个服务器的权威数据之内
SBELT,最终会返回：

```
                +---------------------------------------------- --+
标头            |操作码=查询、响应、AA                                |
                +---------------------------------------------- --+
问题 | QNAME=65.0.6.26.IN-ADDR.ARPA.,QCLASS=IN,QTYPE=PTR |
                +---------------------------------------------- --+
回答            | 65.0.6.26.IN-ADDR.ARPA。          PTR     ACC.ARPA. |
                +---------------------------------------------- --+
权限| <空>                                                          |
                +---------------------------------------------- --+
附加 | <空>                                                          |
                +---------------------------------------------- --+
```

6.3.3.获取 poneria.ISI.EDU 的主机地址

该请求将转化为对 poneria.ISI.EDU 的 A 类请求。
解析器不会为该名称找到任何缓存数据,但会在查找要请求的外部服务器时在 ISI.EDU 的缓存中找到 NS RR。使用这些数据,它将构建一个如下形式的 SLIST:

匹配计数 = 3

A.ISI.EDU。 26.3.0.103
VAXA.ISI.EDU。 10.2.0.27 维内拉.ISI.EDU。 128.9.0.33
10.1.0.52

A.ISI.EDU 列在第一位,前提是解析器按优先顺序排列其选择,并且 A.ISI.EDU 在同一网络上。

这些服务器之一将回答查询。

7. 参考文献和书目

[代尔 87] Dyer, S. 和 F. Hsu,“Hesiod”,雅典娜项目技术计划 - 名称服务,1987 年 4 月,版本 1.9。

描述 Hesiod 名称服务的基础知识。

[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,1979 年 8 月。

被域名系统淘汰但仍在使用的名称服务。

[Quarterman 86] Quarterman, J. 和 J. Hoskins，“著名的计算机网络”，ACM 通讯，1986 年 10 月，第 29 卷，第 10 期。

[RFC-742] K. Harrenstien，“NAME/FINGER”，RFC-742，网络信息中心，SRI International，1977 年 12 月。

[RFC-768] J. Postel，“用户数据报协议”，RFC-768，USC/信息科学研究所，1980 年 8 月。

[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，“Internet 名称域”，RFC-799，COMSAT，1981 年 9 月。

建议引入层次结构来代替 Internet 的平面名称空间。

[RFC-805] J. Postel，“计算机邮件会议记录”，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 Internet 主机表规范”，RFC-810，网络信息中心，SRI International，1982 年 3 月。

过时的。请参阅 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 International，1982 年 3 月。

过时的。请参阅 RFC-953。

[RFC-812] K. Harrenstien 和 V. White，“NICNAME/WHOIS”，RFC-812，网络信息中心，SRI International，1982 年 3 月。

[RFC-819] Z. Su 和 J. Postel，“互联网用户应用程序的域命名约定”，RFC-819，网络信息中心，SRI International，1982 年 8 月。

关于域系统设计的早期思考。
当前的实现完全不同。

[RFC-821] J. Postel，“简单邮件传输协议”，RFC-821，南加州大学/信息科学研究所，1980 年 8 月。

[RFC-830] Z. Su，“Internet 名称服务的分布式系统”，RFC-830，SRI International 网络信息中心，1982 年 10 月。

关于域系统设计的早期思考。
当前的实现完全不同。

[RFC-882] P. Mockapetris，“域名 - 概念和设施”，RFC-882，南加州大学/信息科学研究所，
1983 年 11 月。

由本备忘录取代。

[RFC-883] P. Mockapetris，“域名 - 实施和规范”，RFC-883，南加州大学/信息科学研究所，1983 年 11 月。

由本备忘录取代。

[RFC-920] J. Postel 和 J. Reynolds，“域要求”，RFC-920，南加州大学/信息科学研究所，1984 年 10 月。

解释顶级域的命名方案。

[RFC-952] K. Harrenstien、M. Stahl、E. Feinler，“DoD Internet 主机表规范”，RFC-952，SRI，1985 年 10 月。

指定 HOSTS.TXT 的格式，即由 DNS 替换的主机/地址表。

[RFC-953] K. Harrenstien、M. Stahl、E. Feinler，“主机名服务器”，RFC-953，SRI，1985 年 10 月。

此 RFC 包含已被 DNS 废弃的主机名服务器协议的官方规范。

这种基于 TCP 的协议访问以 RFC-952 格式存储的信息，并用于获取主机表的副本。

[RFC-973] P. Mockapetris，“域系统变化和观察”，RFC-973，南加州大学/信息科学研究所，
1986 年 1 月。

描述对 RFC-882 和 RFC-883 的更改及其原因。现在过时了。

[RFC-974] C. Partridge，“邮件路由和域系统”，RFC-974，CSNET CIC BBN 实验室，1986 年 1 月。

描述从基于 HOSTS.TXT 的邮件寻址到与域系统一起使用的更强大的 MX 系统的转换。

[RFC-1001] NetBIOS 工作组，“TCP/UDP 传输上 NetBIOS 服务的协议标准：概念和方法”，RFC-1001，1987 年 3 月。

此 RFC 和 RFC-1002 是 TCP/IP 之上的 NETBIOS 的初步设计，建议在 DNS 之上建立 NetBIOS 名称服务。

[RFC-1002] NetBIOS 工作组，“TCP/UDP 传输上 NetBIOS 服务的协议标准：详细规范”，RFC-1002，1987 年 3 月。

[RFC-1010] J. Reynolds 和 J. Postel，“Assigned Numbers”，RFC-1010，南加州大学/信息科学研究所，1987 年 5 月

包含主机名、操作系统等的套接字号和助记符。

[RFC-1031] W. Lazear，“MILNET 名称域转换”，RFC-1031，1987 年 11 月。

描述了将 MILNET 转换为 DNS 的计划。

[RFC-1032] MK Stahl，“建立域 - 管理员指南”，RFC-1032，1987 年 11 月。

描述 NIC 用于管理顶级域和委托子区域的注册策略。

[RFC-1033] MK Lottor，“域管理员操作指南”，RFC-1033，1987 年 11 月。

域管理员的食谱。

[所罗门 82] M. Solomon、L. Landweber 和 D. Neuhengen，“CSNET 名称服务器”，计算机网络，第 6 卷，第 3 期，1982 年 7 月。

描述了 CSNET 的名称服务，它独立于 CSNET 中的 DNS 和 DNS 使用。

指数