IMAP 服务器把每个报文与一个文件夹联系起来；当报文第一次到达服务器时，它与收件人的 INBOX 文件夹相关联。收件人则能够把邮件移到一个新的、用户创建的文件夹中，阅读邮件，删除邮件等。IMAP 协议为用户提供了创建文件夹以及将邮件从一个文件夹移动到另一个文件夹的命令。IMAP 还为用户提供了在远程文件夹中查询邮件的命令，按指定条件去查询匹配的邮件。值得注意的是，与 POP3 不同，IMAP 服务器维护了 IMAP 会话的用户状态信息，例如，文件夹的名字以及哪些报文与哪些文件夹相关联。

IMAP 的另一个重要特性是它具有允许用户代理获取报文组件的命令。例如，一个用户代理可以只读取一个报文的报文首部，或只是一个多部分 MIME 报文的一部分。当用户代理和其邮件服务器之间使用低带宽连接的时候，这个特性非常有用（如一个低速调制解调器链路）。使用这种低带宽连接时，用户可能并不想取回他邮箱中的所有邮件，尤其要避免可能包含如音频或视频片断的大邮件。

3. 基于 Web 的电子邮件

今天越来越多的用户使用他们的 Web 浏览器收发电子邮件。20 世纪 90 年代中期 Hotmail 引入了基于 Web 的接入。今天，谷歌、雅虎以及几乎所有重要的大学或者公司也提供了基于 Web 的电子邮件。使用这种服务，用户代理就是普通的浏览器，用户和他远程邮箱之间的通信则通过 HTTP 进行。当一个收件人（如 Bob），想从他的邮箱中访问一个报文时，该电子邮件报文从 Bob 的邮件服务器发送到他的浏览器，使用的是 HTTP 而不是 POP3 或者 IMAP 协议。当发件人（如 Alice）要发送一封电子邮件报文时，该电子邮件报文从 Alice 的浏览器发送到她的邮件服务器，使用的是 HTTP 而不是 SMTP。然而，Alice 的邮件服务器在与其他的邮件服务器之间发送和接收邮件时，仍然使用的是 SMTP。

2.5 DNS：因特网的目录服务

人类能以很多方式来标识。例如，我们能够通过出生证书上的名字来标识；能够通过社会保险号码来标识；也能够通过驾驶执照上的号码来标识。尽管这些标识办法都可以用来识别一个人，但是在特定环境下，某种识别方法可能比另一种方法更为适合。例如，IRS（美国的一个声名狼藉的税务征收机构）的计算机更喜欢使用定长的社会保险号码而不是出生证书上的姓名。另一方面，普通人乐于使用更好记的出生证书上的姓名而不是社会保险号码。（毫无疑问，你能想象人们之间以这种方式说话吗？如“你好，我叫 132-67-9875。请找一下我的丈夫 178-87-1146”。）

因特网上的主机和人类一样，可以使用多种方式进行标识。主机的一种标识方法是用它的**主机名**（hostname），如 cnn. com、www. yahoo. com、gaia. cs. umass. edu 以及 cis. poly. edu 等，这些名字便于记忆也乐于被人们接受。然而，主机名几乎没有提供（即使有也很少）关于主机在因特网中位置的信息。（一个名为 www. eurecom. fr 的主机以国家码 . fr 结束，告诉我们该主机很可能在法国，仅此而已。）况且，因为主机名可能由不定长的字母数字组成，路由器难以处理。由于这些原因，主机也可以使用所谓 **IP 地址**（IP address）进行标识。

我们将在第 4 章更为详细地讨论 IP 地址，但现在简略地介绍一下还是有必要的。一个 IP 地址由 4 个字节组成，并有着严格的层次结构。例如 121. 7. 106. 83 这样一个 IP 地址，其中的每个字节都被句点分隔开来，表示了 0 ~ 255 的十进制数字。我们说 IP 地址具

有层次结构，是因为当我们从左至右扫描它时，我们会得到越来越具体的关于主机位于因特网何处的信息（即在众多网络的哪个网络里）。类似地，当我们从下向上查看邮政地址时，我们能够获得该地址位于何处的越来越具体的信息。

2.5.1 DNS 提供的服务

我们刚刚看到了识别主机有两种方式，通过主机名或者 IP 地址。人们喜欢便于记忆的主机名标识方式，而路由器则喜欢定长的、有着层次结构的 IP 地址。为了折衷这些不同的偏好，我们需要一种能进行主机名到 IP 地址转换的目录服务。这就是**域名系统**（Domain Name System，DNS）的主要任务。DNS 是：①一个由分层的 **DNS 服务器**（DNS server）实现的分布式数据库；②一个使得主机能够查询分布式数据库的应用层协议。DNS 服务器通常是运行 BIND（Berkeley Internet Name Domain）软件［BIND 2012］的 UNIX 机器。DNS 协议运行在 UDP 之上，使用 53 号端口。

实 践 原 则

DNS：通过客户 - 服务器模式提供的重要网络功能

与 HTTP、FTP 和 SMTP 协议一样，DNS 协议是应用层协议，其原因在于：①使用客户 - 服务器模式运行在通信的端系统之间；②在通信的端系统之间通过下面的端到端运输协议来传送 DNS 报文。然而，在其他意义上，DNS 的作用非常不同于 Web 应用、文件传输应用以及电子邮件应用。与这些应用程序不同之处在于，DNS 不是一个直接和用户打交道的应用。相反，DNS 是为因特网上的用户应用程序以及其他软件提供一种核心功能，即将主机名转换为其背后的 IP 地址。我们在 1.2 节就提到，因特网体系结构的复杂性大多数位于网络的“边缘”。DNS 通过采用了位于网络边缘的客户和服务器，实现了关键的名字到地址转换功能，它还是这种设计原理的另一个范例。

DNS 通常是由其他应用层协议所使用的，包括 HTTP、SMTP 和 FTP，将用户提供的主机名解析为 IP 地址。举一个例子，考虑当某个用户主机上的一个浏览器（即一个 HTTP 客户）请求 URL www.someschool.edu/index.html 页面时会发生什么现象。为了使用户的主机能够将一个 HTTP 请求报文发送到 Web 服务器 www.someschool.edu，该用户主机必须获得 www.someschool.edu 的 IP 地址。其做法如下。

- 同一台用户主机上运行着 DNS 应用的客户端。
- 浏览器从上述 URL 中抽取出主机名 www.someschool.edu，并将这台主机名传给 DNS 应用的客户端。
- DNS 客户向 DNS 服务器发送一个包含主机名的请求。
- DNS 客户最终会收到一份回答报文，其中含有对应于该主机名的 IP 地址。
- 一旦浏览器接收到来自 DNS 的该 IP 地址，它能够向位于该 IP 地址 80 端口的 HTTP 服务器进程发起一个 TCP 连接。

从这个例子中，我们可以看到 DNS 给使用它的因特网应用带来了额外的时延，有时还相当可观。幸运的是，如我们下面讨论的那样，想获得的 IP 地址通常就缓存在一个“附近的”DNS 服务器中，这有助于减少 DNS 的网络流量和 DNS 的平均时延。

除了进行主机名到 IP 地址的转换外，DNS 还提供了一些重要的服务：

- **主机别名**（host aliasing）。有着复杂主机名的主机能拥有一个或者多个别名。例如，一台名为 relay1. west-coast. enterprise. com 的主机，可能还有两个别名为 enterprise. com 和 www. enterprise. com。在这种情况下，relay1. west-coast. enterprise. com 也称为**规范主机名**（canonical hostname）。主机别名（当存在时）比主机规范名更加容易记忆。应用程序可以调用 DNS 来获得主机别名对应的规范主机名以及主机的 IP 地址。
- **邮件服务器别名**（mail server aliasing）。显而易见，人们也非常希望电子邮件地址好记忆。例如，如果 Bob 在 Hotmail 上有一个账户，Bob 的邮件地址就像 bob@ hotmail. com 这样简单。然而，Hotmail 邮件服务器的主机名可能更为复杂，不像 hotmail. com 那样简单好记（例如，规范主机名可能像 relay1. west-coast. hotmail. com 那样）。电子邮件应用程序可以调用 DNS，对提供的邮件服务器别名进行解析，以获得该主机的规范主机名及其 IP 地址。事实上，MX 记录（参见后面）允许一个公司的邮件服务器和 Web 服务器使用相同（别名化的）的主机名；例如，一个公司的 Web 服务器和邮件服务器都能叫做 enterprise. com。
- **负载分配**（load distribution）。DNS 也用于在冗余的服务器（如冗余的 Web 服务器等）之间进行负载分配。繁忙的站点（如 cnn. com）被冗余分布在多台服务器上，每台服务器均运行在不同的端系统上，每个都有着不同的 IP 地址。由于这些冗余的 Web 服务器，一个 IP 地址集合因此与同一个规范主机名相联系。DNS 数据库中存储着这些 IP 地址集合。当客户对映射到某地址集合的名字发出一个 DNS 请求时，该服务器用 IP 地址的整个集合进行响应，但在每个回答中循环这些地址次序。因为客户通常总是向 IP 地址排在最前面的服务器发送 HTTP 请求报文，所以 DNS 就在所有这些冗余的 Web 服务器之间循环分配了负载。DNS 的循环同样可以用于邮件服务器，因此，多个邮件服务器可以具有相同的别名。一些内容分发公司如 Akamai 也以更加复杂的方式使用 DNS［Dilley 2002］，以提供 Web 内容分发（参见第 7 章）。

DNS 由 RFC 1034 和 RFC 1035 定义，并且在几个附加的 RFC 中进行了更新。DNS 是一个复杂的系统，我们在这里只是就其运行的主要方面进行学习。感兴趣的读者可以参考这些 RFC 文档和 Albitz 和 Liu 写的书［Albitz 1993］；亦可参阅文章［Mockapetris 1998］和［Mockapetris 2005］，其中［Mockapetris 1998］是回顾性的文章，它提供了 DNS 组成和工作原理的精细的描述。

2.5.2 DNS 工作机理概述

下面给出一个 DNS 工作过程的总体概括，我们的讨论将集中在主机名到 IP 地址转换服务方面。

假设运行在用户主机上的某些应用程序（如 Web 浏览器或邮件阅读器）需要将主机名转换为 IP 地址。这些应用程序将调用 DNS 的客户端，并指明需要被转换的主机名（在很多基于 UNIX 的机器上，应用程序为了执行这种转换需要调用函数 gethostbyname()）。用户主机上的 DNS 接收到后，向网络中发送一个 DNS 查询报文。所有的 DNS 请求和回答报文使用 UDP 数据报经端口 53 发送。经过若干毫秒到若干秒的时延后，用户主机上的

DNS 接收到一个提供所希望映射的 DNS 回答报文。这个映射结果则被传递到调用 DNS 的应用程序。因此，从用户主机上调用应用程序的角度看，DNS 是一个提供简单、直接的转换服务的黑盒子。但事实上，实现这个服务的黑盒子非常复杂，它由分布于全球的大量 DNS 服务器以及定义了 DNS 服务器与查询主机通信方式的应用层协议组成。

DNS 的一种简单设计是在因特网上只使用一个 DNS 服务器，该服务器包含所有的映射。在这种集中式设计中，客户直接将所有查询直接发往单一的 DNS 服务器，同时该 DNS 服务器直接对所有的查询客户做出响应。尽管这种设计的简单性非常具有吸引力，但它不适用于当今的因特网，因为因特网有着数量巨大（并持续增长）的主机。这种集中式设计的问题包括：

- **单点故障**（a single point of failure）。如果该 DNS 服务器崩溃，整个因特网随之瘫痪！
- **通信容量**（traffic volume）。单个 DNS 服务器不得不处理所有的 DNS 查询（用于为上亿台主机产生的所有 HTTP 请求报文和电子邮件报文服务）。
- **远距离的集中式数据库**（distant centralized database）。单个 DNS 服务器不可能“邻近”所有查询客户。如果我们将单台 DNS 服务器放在纽约市，那么所有来自澳大利亚的查询必须传播到地球的另一边，中间也许还要经过低速和拥塞的链路。这将导致严重的时延。
- **维护**（maintenance）。单个 DNS 服务器将不得不为所有的因特网主机保留记录。这不仅将使这个中央数据库非常庞大，而且它还不得不为解决每个新添加的主机而频繁更新。

总的来说，在单一 DNS 服务器上运行集中式数据库完全没有可扩展能力。因此，DNS 采用了分布式的设计方案。事实上，DNS 是一个在因特网上实现分布式数据库的精彩范例。

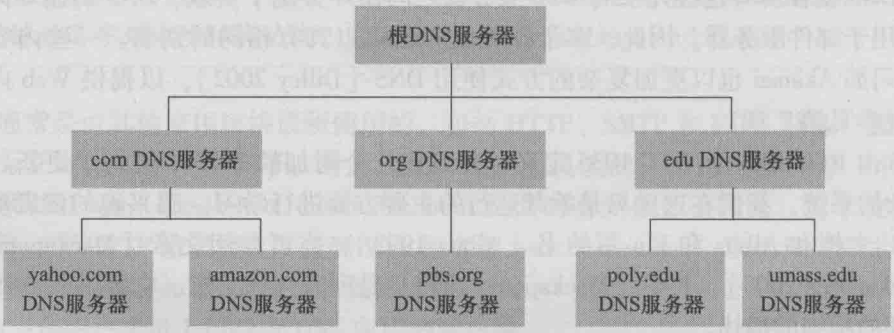

图 2-19 DNS 服务器的部分层次结构

1. 分布式、层次数据库

为了处理扩展性问题，DNS 使用了大量的 DNS 服务器，它们以层次方式组织，并且分布在全世界范围内。没有一台 DNS 服务器拥有因特网上所有主机的映射。相反，该映射分布在所有的 DNS 服务器上。大致说来，有 3 种类型的 DNS 服务器：根 DNS 服务器、顶级域（Top-Level Domain，TLD）DNS 服务器和权威 DNS 服务器。这些服务器以图 2-19 中所示的层次结构组织起来。为了理解这 3 种类型的 DNS 服务器交互的方式，假定一个 DNS 客户要决定主机名 www. amazon. com 的 IP 地址。粗略说来，将发生下列事件。客户首

先与根服务器之一联系，它将返回顶级域名 com 的 TLD 服务器的 IP 地址。该客户则与这些 TLD 服务器之一联系，它将为 amazon. com 返回权威服务器的 IP 地址。最后，该客户与 amazon. com 权威服务器之一联系，它为主机名 www. amazon. com 返回其 IP 地址。我们将很快更为详细地考察 DNS 查找过程。不过我们先仔细看一下这 3 种类型的 DNS 服务器。

- **根 DNS 服务器**。在因特网上有 13 个根 DNS 服务器（标号为 A 到 M），它们中的大部分位于北美洲。图 2-20 中显示的是一张 2012 年的根 DNS 服务器分布图；通过［Root-servers 2012］可查看当前可用的根 DNS 服务器列表。尽管我们将这 13 个根 DNS 服务器中的每个都视为单个的服务器，但每台“服务器”实际上是一个冗余服务器的网络，以提供安全性和可靠性。到了 2011 年秋季，共有 247 个根服务器。
- **顶级域（DNS）服务器**。这些服务器负责顶级域名如 com、org、net、edu 和 gov，以及所有国家的顶级域名如 uk、fr、ca 和 jp。Verisign Global Registry Services 公司维护 com 顶级域的 TLD 服务器；Educause 公司维护 edu 顶级域的 TLD 服务器。所有顶级域的列表参见［IANA TLD 2012］。
- **权威 DNS 服务器**。在因特网上具有公共可访问主机（如 Web 服务器和邮件服务器）的每个组织机构必须提供公共可访问的 DNS 记录，这些记录将这些主机的名字映射为 IP 地址。一个组织机构的权威 DNS 服务器收藏了这些 DNS 记录。一个组织机构能够选择实现它自己的权威 DNS 服务器以保存这些记录；另一种方法是，该组织能够支付费用，让这些记录存储在某个服务提供商的一个权威 DNS 服务器中。多数大学和大公司实现和维护它们自己基本和辅助（备份）的权威 DNS 服务器。

图 2-20 2012 年的 DNS 根服务器（名称、组织和位置）

根、TLD 和权威 DNS 服务器都处在该 DNS 服务器的层次结构中，如图 2-19 中所示。还有另一类重要的 DNS，称为**本地 DNS 服务器**（local DNS server）。一个本地 DNS 服务器严格说来并不属于该服务器的层次结构，但它对 DNS 层次结构是重要的。每个 ISP（如一个大学、一个系、一个公司或一个居民区的 ISP）都有一台本地 DNS 服务器（也叫默认名字服务器）。当主机与某个 ISP 连接时，该 ISP 提供一台主机的 IP 地址，该主机具有一台

或多台其本地 DNS 服务器的 IP 地址（通常通过 DHCP，将在第 4 章中讨论）。通过访问 Windows 或 UNIX 的网络状态窗口，能够容易地确定你本地 DNS 服务器的 IP 地址。主机的本地 DNS 服务器通常“邻近”本主机。对某机构 ISP 而言，本地 DNS 服务器可能就与主机在同一个局域网中；对于某居民区 ISP 来说，本地 DNS 服务器通常与主机相隔不超过几台路由器。当主机发出 DNS 请求时，该请求被发往本地 DNS 服务器，它起着代理的作用，并将该请求转发到 DNS 服务器层次结构中，我们下面将更为详细地讨论。

我们来看一个简单的例子，假设主机 cis. poly. edu 想知道主机 gaia. cs. umass. edu 的 IP 地址。同时假设理工大学（Polytechnic）的本地 DNS 服务器为 dns. poly. edu，并且 gaia. cs. umass. edu 的权威 DNS 服务器为 dns. umass. edu。如图 2- 21 所示，主机 cis. poly. edu 首先向它的本地 DNS 服务器 dns. poly. edu 发送一个 DNS 查询报文。该查询报文含有被转换的主机名 gaia. cs. umass. edu。本地 DNS 服务器将该报文转发到根 DNS 服务器。该根 DNS 服务器注意到其 edu 前缀并向本地 DNS 服务器返回负责 edu 的 TLD 的 IP 地址列表。该本地 DNS 服务器则再次向这些 TLD 服务器之一发送查询报文。该 TLD 服务器注意到 umass. edu 前缀，并用权威 DNS 服务器的 IP 地址进行响应，该权威 DNS 服务器是负责马萨诸塞大学的 dns. umass. edu。最后，本地 DNS 服务器直接向 dns. umass. edu 重发查询报文，dns. umass. edu 用 gaia. cs. umass. edu 的 IP 地址进行响应。注意到在本例中，为了获得一台主机名的映射，共发送了 8 份 DNS 报文：4 份查询报文和 4 份回答报文！我们将很快看到利用 DNS 缓存减少这种查询流量的方法。

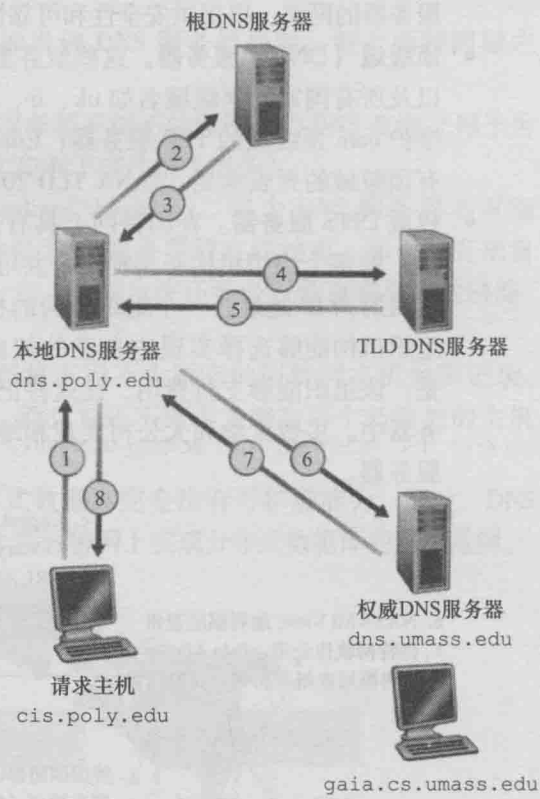

图 2-21 各种 DNS 服务器的交互

我们前面的例子假设了 TLD 服务器知道用于主机的权威 DNS 服务器的 IP 地址。一般而言，这种假设并不总是正确的。相反，TLD 服务器只是知道中间的某个 DNS 服务器，该中间 DNS 服务器依次才能知道用于该主机的权威 DNS 服务器。例如，再次假设马萨诸塞大学有一台用于本大学的 DNS 服务器，它称为 dns. umass. edu。同时假设该大学的每个系都有自己的 DNS 服务器，每个系的 DNS 服务器是本系所有主机的权威服务器。在这种情况下，当中间 DNS 服务器 dns. umass. edu 收到了对某主机的请求时，该主机名是以 cs. umass. edu 结尾，它向 dns. poly. edu 返回 dns. cs. umass. edu 的 IP 地址，后者是所有以 cs. umass. edu 结尾的主机的权威服务器。本地 DNS 服务器 dns. poly. edu 则向权威 DNS 服务器发送查询，该权威 DNS 服务器将请求的映射发送给本地 DNS 服务器，该本地服务器依次向请求主机返回该映射。在这个例子中，共发送了 10 份 DNS 报文！

图 2-21 所示的例子利用了**递归查询**（recursive query）和**迭代查询**（iterative query）。从

cis. poly. edu 到 dns. poly. edu 发出的查询是递归查询，因为该查询请求 dns. poly. edu 以自己的名义获得该映射。而后继的 3 个查询是迭代查询，因为所有的回答都是直接返回给 dns. poly. edu。从理论上讲，任何 DNS 查询既可以是迭代的也能是递归的。例如，图 2-22 显示了一条 DNS 查询链，其中的所有查询都是递归的。实践中，查询通常遵循图 2-21 中的模式。从请求主机到本地 DNS 服务器的查询是递归的，其余的查询是迭代的。

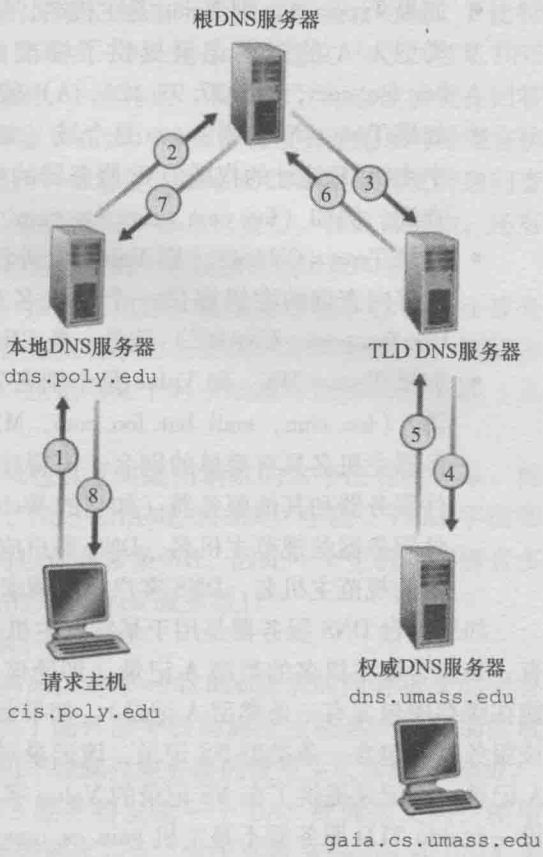

图 2-22 DNS 中的递归查询

2. DNS 缓存

至此我们的讨论还没有涉及 DNS 系统的一个非常重要特色：**DNS 缓存**（DNS caching）。实际上，为了改善时延性能并减少在因特网上到处传输的 DNS 报文数量，DNS 广泛使用了缓存技术。DNS 缓存的原理非常简单。在一个请求链中，当某 DNS 服务器接收一个 DNS 回答（例如，包含主机名到 IP 地址的映射）时，它能将该回答中的信息缓存在本地存储器中。例如，在图 2-21 中，每当本地 DNS 服务器 dns. poly. edu 从某个 DNS 服务器接收到一个回答，它能够缓存包含在该回答中的任何信息。如果在 DNS 服务器中缓存了一台主机名/IP 地址对，另一个对相同主机名的查询到达该 DNS 服务器时，该 DNS 服务器就能够提供所要求的 IP 地址，即使它不是该主机名的权威服务器。由于主机和主机名与 IP 地址间的映射并不是永久的，DNS 服务器在一段时间后（通常设置为两天）将丢弃缓存的信息。

举一个例子，假定主机 apricot. poly. edu 向 dns. poly. edu 查询主机名 cnn. com 的 IP 地址。此后，假定过了几个小时，Polytechnic 理工大学的另外一台主机如 kiwi. poly. edu 也向 dns. poly. edu 查询相同的主机名。因为有了缓存，该本地 DNS 服务器可以立即返回 cnn. com 的 IP 地址，而不必查询任何其他 DNS 服务器。本地 DNS 服务器也能够缓存 TLD 服务器的 IP 地址，因而允许本地 DNS 绕过查询链中的根 DNS 服务器（这经常发生）。

2.5.3 DNS 记录和报文

共同实现 DNS 分布式数据库的所有 DNS 服务器存储了**资源记录**（Resource Record，RR），RR 提供了主机名到 IP 地址的映射。每个 DNS 回答报文包含了一条或多条资源记录。在本小节以及后续小节中，我们概要地介绍 DNS 资源记录和报文；更详细的信息可以在［Albitz 1993］或有关 DNS 的 RFC 文档［RFC 1034；RFC 1035］中找到。

资源记录是一个包含了下列字段的 4 元组：

(Name，Value，Type，TTL)

TTL 是该记录的生存时间，它决定了资源记录应当从缓存中删除的时间。在下面给出的记录例子中，我们忽略掉 TTL 字段。Name 和 Value 的值取决于 Type：

- 如果 Type = A，则 Name 是主机名，Value 是该主机名对应的 IP 地址。因此，一条类型为 A 的资源记录提供了标准的主机名到 IP 地址的映射。例如（relay1. bar. foo. com，145. 37. 93. 126，A）就是一条类型 A 记录。
- 如果 Type = NS，则 Name 是个域（如 foo. com），而 Value 是个知道如何获得该域中主机 IP 地址的权威 DNS 服务器的主机名。这个记录用于沿着查询链来路由 DNS 查询。例如（foo. com，dns. foo. com，NS）就是一条类型为 NS 的记录。
- 如果 Type = CNAME，则 Value 是别名为 Name 的主机对应的规范主机名。该记录能够向查询的主机提供一个主机名对应的规范主机名，例如（foo. com，relay1. bar. foo. com，CNAME）就是一条 CNAME 类型的记录。
- 如果 Type = MX，则 Value 是个别名为 Name 的邮件服务器的规范主机名。举例来说，（foo. com，mail. bar. foo. com，MX）就是一条 MX 记录。MX 记录允许邮件服务器主机名具有简单的别名。值得注意的是，通过使用 MX 记录，一个公司的邮件服务器和其他服务器（如它的 Web 服务器）可以使用相同的别名。为了获得邮件服务器的规范主机名，DNS 客户应当请求一条 MX 记录；而为了获得其他服务器的规范主机名，DNS 客户应当请求 CNAME 记录。

如果一台 DNS 服务器是用于某特定主机名的权威 DNS 服务器，那么该 DNS 服务器会有一条包含该主机名的类型 A 记录（即使该 DNS 服务器不是其权威 DNS 服务器，它也可能在缓存中包含有一条类型 A 记录）。如果服务器不是用于某主机名的权威服务器，那么该服务器将包含一条类型 NS 记录，该记录对应于包含主机名的域；它还将包括一条类型 A 记录，该记录提供了在 NS 记录的 Value 字段中的 DNS 服务器的 IP 地址。举例来说，假设一台 edu TLD 服务器不是主机 gaia. cs. umass. edu 的权威 DNS 服务器，则该服务器将包含一条包括主机 cs. umass. edu 的域记录，如（umass. edu，dns. umass. edu，NS）；该 edu TLD 服务器还将包含一条类型 A 记录，如（dns. umass. edu，128. 119. 40. 111，A），该记录将名字 dns. umass. edu 映射为一个 IP 地址。

1. DNS 报文

在本节前面，我们提到了 DNS 查询和回答报文。DNS 只有这两种报文，并且，查询和回答报文有着相同的格式，如图 2-23 所示。DNS 报文中各字段的语义如下：

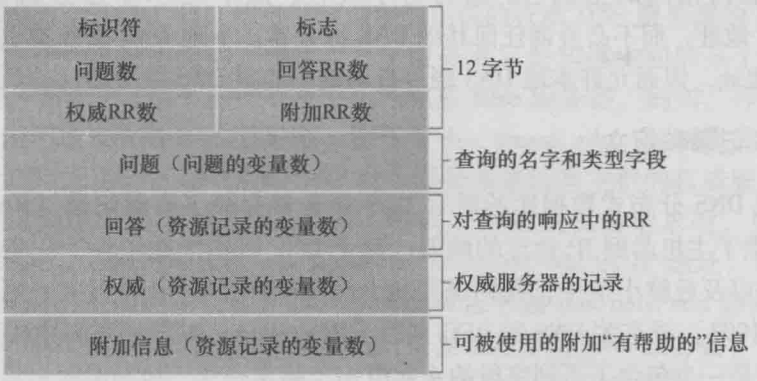

图 2-23 DNS 报文格式

- 前12个字节是首部区域，其中有几个字段。第一个字段（标识符）是一个16比特的数，用于标识该查询。这个标识符会被复制到对查询的回答报文中，以便让客户用它来匹配发送的请求和接收到的回答。标志字段中含有若干标志。1比特的“查询/回答”标志位指出报文是查询报文（0）还是回答报文（1）。当某DNS服务器是所请求名字的权威DNS服务器时，1比特的“权威的”标志位被置在回答报文中。如果客户（主机或者DNS服务器）在该DNS服务器没有某记录时希望它执行递归查询，将设置1比特的“希望递归”标志位。如果该DNS服务器支持递归查询，在它的回答报文中会对1比特的“递归可用”标志位置位。在该首部中，还有4个有关数量的字段，这些字段指出了在首部后的4类数据区域出现的数量。
- 问题区域包含着正在进行的查询信息。该区域包括：①名字字段，指出正在被查询的主机名字；②类型字段，它指出有关该名字的正被询问的问题类型，例如主机地址是与一个名字相关联（类型A）还是与某个名字的邮件服务器相关联（类型MX）。
- 在来自DNS服务器的回答中，回答区域包含了对最初请求的名字的资源记录。前面讲过每个资源记录中有Type（如A、NS、CNAME和MX）字段、Value字段和TTL字段。在回答报文的回答区域中可以包含多条RR，因此一个主机名能够有多个IP地址（例如，就像本节前面讨论的冗余Web服务器）。
- 权威区域包含了其他权威服务器的记录。
- 附加区域包含了其他有帮助的记录。例如，对于一个MX请求的回答报文的回答区域包含了一条资源记录，该记录提供了邮件服务器的规范主机名。该附加区域包含一个类型A记录，该记录提供了用于该邮件服务器的规范主机名的IP地址。

你愿意从正在工作的主机直接向某些DNS服务器发送一个DNS查询报文吗？使用**nslookup程序**（nslookup program）能够容易地做到这一点，对于多数Windows和UNIX平台，nslookup程序是可用的。例如，从一台Windows主机打开命令提示符界面，直接键入“nslookup”即可调用该nslookup程序。在调用nslookup后，你能够向任何DNS服务器（根、TLD或权威）发送DNS查询。在接收到来自DNS服务器的回答后，nslookup将显示包括在该回答中的记录（以人可读的格式）。从你自己的主机运行nslookup还有一种方法，即访问允许你远程应用nslookup的许多Web站点之一（在一个搜索引擎中键入“nslookup”就能够得到这些站点中的一个）。本章最后的DNS Wireshark实验将使你更为详细地研究DNS。

2. 在DNS数据库中插入记录

上面的讨论只是关注如何从DNS数据库中取数据。你可能想知道这些数据最初是怎么进入数据库中的。我们在一个特定的例子中看看这是如何完成的。假定你刚刚创建一个称为网络乌托邦（Network Utopia）的令人兴奋的新创业公司。你必定要做的第一件事是在注册登记机构注册域名networkutopia.com。**注册登记机构**（registrar）是一个商业实体，它验证该域名的唯一性，将该域名输入DNS数据库（如下面所讨论的那样），对提供的服务收取少量费用。1999年前，唯一的注册登记机构是Nework Solution，它独家经营对于com、net和org域名的注册。但是现在有许多注册登记机构竞争客户，因特网名字和地址分配机构（Internet Corporation for Assigned Names and Numbers，ICANN）向各种注册登记机构授权。在http://www.internic.net上可以找到授权的注册登记机构的列表。

当你向某些注册登记机构注册域名 networkutopia. com 时，需要向该机构提供你的基本和辅助权威 DNS 服务器的名字和 IP 地址。假定该名字和 IP 地址是 dns1. networkutopia. com 和 dns2. networkutopia. com 及 212. 212. 212. 1 和 212. 212. 212. 2。对这两个权威 DNS 服务器的每一个，该注册登记机构确保将一个类型 NS 和一个类型 A 的记录输入 TLD com 服务器。特别是对于用于 networkutopia. com 的基本权威服务器，该注册登记机构将下列两条资源记录插入该 DNS 系统中：

```
(networkutopia.com, dns1.networkutopia.com, NS)

(dns1.networkutopia.com, 212.212.212.1, A)
```

你还必须确保用于 Web 服务器 www. networkutopia. com 的类型 A 资源记录和用于邮件服务器 mail. networkutopia. com 的类型 MX 资源记录被输入你的权威 DNS 服务器中。（直到最近，每个 DNS 服务器中的内容都是静态配置的，例如来自系统管理员创建的配置文件。最近，在 DNS 协议中添加了一个更新（UPDATE）选项，允许通过 DNS 报文对数据库中的内容进行动态添加或者删除。[RFC 2136] 和 [RFC 3007] 定义了 DNS 动态更新。）

一旦完成所有这些步骤，人们将能够访问你的 Web 站点，并向你公司的雇员发送电子邮件。我们通过验证该说法的正确性来总结 DNS 的讨论。这种验证也有助于充实我们已经学到的 DNS 知识。假定在澳大利亚的 Alice 要观看 www. networkutopia. com 的 Web 页面。如前面所讨论，她的主机将首先向其本地 DNS 服务器发送请求。该本地服务器接着则联系一个 TLD com 服务器。（如果 TLD com 服务器的地址没有被缓存，该本地 DNS 服务器也将必须与根 DNS 服务器相联系。）该 TLD 服务器包含前面列出的类型 NS 和类型 A 资源记录，因为注册登记机构将这些资源记录插入所有的 TLD com 服务器。该 TLD com 服务器向 Alice 的本地 DNS 服务器发送一个回答，该回答包含了这两条资源记录。该本地 DNS 服务器则向 212. 212. 212. 1 发送一个 DNS 查询，请求对应于 www. networkutopia. com 的类型 A 记录。该记录提供了所希望的 Web 服务器的 IP 地址，如 212. 212. 71. 4，本地 DNS 服务器将该地址回传给 Alice 的主机。Alice 的浏览器此时能够向主机 212. 212. 71. 4 发起一个 TCP 连接，并在该连接上发送一个 HTTP 请求。当一个人在网上冲浪时，有比满足眼球更多的事情在进行！

关注安全性

DNS 脆弱性

我们已经看到 DNS 是因特网基础设施的一个至关重要的组件，对于包括 Web、电子邮件等的许多重要的服务，没有它都不能正常工作。因此，我们自然要问，DNS 能够被怎样攻击呢？DNS 是一个易受攻击的目标吗？它是将会被淘汰的服务吗？大多数因特网应用会随同它一起无法工作吗？

想到的第一种针对 DNS 服务的攻击是分布式拒绝服务（DDoS）带宽洪泛攻击（参见 1. 6 节）。例如，某攻击者能够试图向每个 DNS 根服务器发送大量的分组，使得大多数合法 DNS 请求得不到回答。这种对 DNS 根服务器的 DDoS 大规模攻击实际发生在 2002 年 10 月 21 日。在这次攻击中，该攻击者利用了一个僵尸网络向 13 个 DNS 根服务器中的每个都发送了大批的 ICMP ping 报文。（第 4 章中讨论了 ICMP 报文。此时，知道 ICMP

分组是特殊类型的 IP 数据报就可以了。）幸运的是，这种大规模攻击所带来的损害很小，对用户的因特网体验几乎没有或根本没有影响。攻击者的确成功地将大量的分组指向了根服务器，但许多 DNS 根服务器受到了分组过滤器的保护，配置的分组过滤器阻挡了所有指向根服务器的 ICMP ping 报文。这些被保护的服务器因此未受伤害并且与平常一样发挥着作用。此外，大多数本地 DNS 服务器缓存了顶级域名服务器的 IP 地址，使得这些请求过程通常绕过了 DNS 根服务器。

对 DNS 的潜在更为有效的 DDoS 攻击将是向顶级域名服务器（例如向所有处理.com 域的顶级域名服务器）发送大量的 DNS 请求。过滤指向 DNS 服务器的 DNS 请求将更为困难，并且顶级域名服务器不像根服务器那样容易绕过。但是这种攻击的严重性通过本地 DNS 服务器中的缓存技术可将部分地被缓解。

DNS 能够潜在地以其他方式被攻击。在中间人攻击中，攻击者截获来自主机的请求并返回伪造的回答。在 DNS 毒害攻击中，攻击者向一台 DNS 服务器发送伪造的回答，诱使服务器在它的缓存中接收伪造的记录。这些攻击中的任一种，都能够将满怀信任的 Web 用户重定向到攻击者的 Web 站点。然而，这些攻击难以实现，因为它们要求截获分组或扼制住服务器［Skoudis 2006］。

另一种重要的 DNS 攻击本质上并不是一种对 DNS 服务的攻击，而是充分利用 DNS 基础设施来对目标主机发起 DDoS 攻击（例如，你所在大学的邮件服务器）。在这种攻击中，攻击者向许多权威 DNS 服务器发送 DNS 请求，每个请求具有目标主机的假冒源地址。这些 DNS 服务器则直接向目标主机发送它们的回答。如果这些请求能够精心制作成下述方式的话，即响应比请求（字节数）大得多（所谓放大），则攻击者不必自行产生大量的流量就有可能淹没目标主机。这种利用 DNS 的反射攻击至今为止只取得了有限的成功［Mirkovic 2005］。

总而言之，DNS 自身已经显示了对抗攻击的令人惊讶的健壮性。至今为止，还没有一个攻击已经成功地妨碍了 DNS 服务。已经有了成功的反射攻击；然而，通过适当地配置 DNS 服务器，能够处理（和正在处理）这些攻击。

2.6 P2P 应用

在目前为止本章中描述的应用（包括 Web、电子邮件和 DNS）都采用了客户 - 服务器体系结构，极大地依赖于总是打开的基础设施服务器。2.1.1 节讲过，使用 P2P 体系结构，对总是打开的基础设施服务器有最小的（或者没有）依赖。与之相反，成对间歇连接的主机（称为对等方）彼此直接通信。这些对等方并不为服务提供商所拥有，而是受用户控制的桌面计算机和膝上计算机。

在本节中我们将研究两种不同的特别适合于 P2P 设计的应用。第一种应用是文件分发，其中应用程序从单个源向大量的对等方分发一个文件。文件分发是开始研究 P2P 的良好起点，因为它清晰地揭示了 P2P 体系结构的自扩展性。作为文件分发的一个特定的例子，我们将描述流行的 BitTorrent 协议。我们将研究的第二种 P2P 应用是分布在大型对等方社区中的数据库。对于这个应用，我们将探讨**分布式散列表**（Distributed Hash Table，DHT）的概念。