第2章 TCP/IP网络

大家在把自己的机器连到一个 TCP/IP网络上时，肯定会碰到许多问题，比如如何处理 IP地址和主机名，有时甚至还会碰到路由问题。本章将为大家提供相关背景，帮助大家尽快了解安装时需要什么，下一章将为大家介绍几个处理这些问题的工具。

2.1 网络接口

先撇开联网环境中的设备可能不同这一情况。 TCP/IP定义了一个抽象接口，所有对设备的访问都将通过这一接口。该接口为所有类型的硬件设备提供了同一个操作集，基本上用于包的收发。

针对准备用于联网的各个外设，内核中都必须有相应的接口。比如，以太网接口即称为eth0和eth1，SLIP接口称为sl0、sll，以此类推。在打算针对内核对特定的物理设备命名时，出于配置上的需要，可使用这些接口名。除此以外，它们没有别的用途。

如果要在TCP/IP联网中使用接口，必须为它（它们）分配 IP地址，藉此向网络中的其他机器标识自己。这个地址不同于上面所提的接口名；打个比喻说，如果把接口比作门，那么，其IP地址就是钉在这扇门上的门牌号。

当然，还需要设立别的设备参数；其中之一便是特定的硬件能够处理的数据报的最大字节数，也叫作“最大传输单元”或MTU。其他属性，我们将在后续章节中讨论。

2.2 IP地址

正如前面提到的那样， IP联网协议能够识别的地址是一个 32位的编号。在网络环境中，为每台机器分配的编号必须是独一无二的。如果你运行的是局域网，而且没有与其他网络进行TCP/IP通信传输，就可以按自己的喜好，为各台机器分配编号。但是，如果运行的是面向因特网的网站，其编号必须由ICANN（www.icann.org）分配。

为便于理解，IP地址由四个8位编号（叫做8位元）组成。比如，quark.physics.groucho.edu有一个IP地址是0x954C0C04，该地址表示为149.76.12.4。这一格式就是人们常说的点分四段式。

采用点分四段式的另一个原因是IP地址由两大部分组成：网络部分（即排在前面的8位元）和主机部分（其余的）。根据网络的大小，主机编号可大可小。

为适应不同的需求，这里为大家列出了几类IP编址，它们定义IP地址的不同划分方式：

■ 范围在1.0.0.0到127.0.0.0之间的A类地址。这类地址的网络部分在第一个8位元内。A类编址提供了一个24位的主机编号，也就是说，网络中的主机可以多达160万台。

■ 范围在128.0.0.0和191.255.0.0之间的B类地址。网络部分在前两个 8位元中。可接纳16 320个网络，而每个网络可接纳 65 024台主机

■ 范围在192.0.0.0和223.255.255.0之间的C类地址，其网络部分在前三个8位元内。可接纳近200万个网络， 254台主机。

■ D、E和F类地址的范围在224.0.0.0和254.0.0.0之间。这几类地址尚处于实验阶段，或者是为将来的应用而保留的，不指定任何网络。

回头看看前一章的示例，我们会发现 149.76.12.4（Quark的IP地址）指的是 B类网络149.76.0.0中的12.4主机。

需要注意的是，有两个地址是保留地址，它们是：0.0.0.0和127.0.0.0。前者称为默认路由，后者称为回送（loopback）地址。默认路由和IP协议对数据报的路由方式有关，它主要用于简化IP路由选择信息，详情参见下一小节。回送地址 127.0.0.0是为本地主机的IP通信保留的。通常，127.0.0.1这个地址是分配给本地主机上的特殊接口的，也就是所谓的“回送接口”（loopback interface）。回送接口就像一个封闭式的回路。任何自 TCP或UDP传给该接口的IP包都将被返回始发地，就像它们刚从某个网络返回一样。有了回送接口，就可以在单机主机上开发和测试联网效果，利用网络软件了。这类情况并不罕见，比如，许多 UUCP站点根本就没有IP连接功能，但人们仍然想在这些站点上运行 INN新闻系统。

2.3 地址解析

现在，大家已经了解了 IP地址的构成情况，可能还想进一步了解它们是如何用于以太网，为不同主机定址的。毕竟，以太网协议按 6位8位元编号（和IP地址完全没有关系）来识别主机。

是的，我们需要一种机制，能把 IP地址映射为以太网地址。这就是所谓的“地址解析协议”（简称ARP）。事实上，ARP的使用并不局限于以太网，它还用于其他类型的网络，比如“火腿无线电网”。ARP的基本思路源于此：必须在150个人当中寻找一个名为某某的人，大多数人都会采用类似的办法——四处转，高声呼叫他的名字，因为对方听到叫声，肯定会答应。

ARP想找出与具体IP地址对应的以太网地址时，将利用“广播”这一以太网特性，这样就能将一个数据报同步传递到网络中的各个角落。 ARP发送的广播数据报中，包含一个 IP地址查询。每个收到广播数据报的主机都会将其中的 IP地址和它自己的进行比较，如果相同，就向发出查询的主机返回一个 ARP应答。这样一来，发出查询的主机就可以从收到的应答中，获悉对方的以太网地址。

当然，有些人可能觉得奇怪：全世界的以太网举不胜举，主机是怎样在如此庞大的体系中找到自己的目标的呢？要解决这一困惑，还需要了解路由选择，有了路由选择，才能够在网络中找到一台主机的物理位置。这是我们下一小节的主题。

关于ARP，我们还有话说。一旦主机知道了自己想要的以太网地址，它就会把该地址保存在它自己的ARP缓冲区内，这样在下一次查询该地址时，它会直接向目标主机发送数据报。但是，要想永久性地保留该地址信息是很不明智的；比如说，远程主机的以太网卡可能因为技术上的原因已发生替换，所以你保留的 ARP条目将毫无意义。因此，要执行另一次 IP地址查询，就应该时不时地丢弃ARP缓冲区内的条目。

有时，还有必要找出与具体以太网地址相关联的 IP地址。比如说，想从网络上的一台服务器启动无盘客户机这样的事，在局域网内是家常便饭。但是，无盘客户机事实上根本就没有任何关于它自己的信息——除开它自己的以太网地址！所以，它能做的只能是广播一条消息，请求根服务器将它的 IP地址告诉它。这牵涉到另一个协议，名为“逆向地址解析协议”（简称RARP）。它和BOOTP协议一起，为网络上的无盘客户机的启动运行定义一个进程。

2.4 IP路由

2.4.1 IP网络

大家可能有这种经历；在写信给某人时，通常会在信封上写明收信人的确切地址，指定国家、地区、省份，城市以及详细到几栋几号。然后，再把信投入邮箱，邮政服务将把信送达目的地：先送到指定国家，再由那个国家的邮政服务将其分发到相应的省或地区。邮政服务的层次结构非常清楚：不管在哪里寄信，本地的邮政主管都知道信件投递路径，并将进行信件转发，并不注重信件在目标国内的投递方式（注意，邮件投递系统只注重完成任务，而不管如何去完成任务）。

IP网络其实与邮件投递系统类似。整个因特网由无数个称为独立系统的网络构成。每个这样的系统都会在各自的成员主机之间执行路由，所以投递数据报的任务实际上就是找出通往目标主机所在网络的路径。也就是说，只要数据报被传到特定网络上的任何一台主机，怎样到达最终的目标主机，则由这个特定网络自行负责。

2.4.2 子网

我们还可以从被分为网络部分和主机部分的 IP地址中，看出IP网络的结构。默认情况下，目标网络（即目标主机所处的网络）是从 IP地址的网络部分衍生的。所以， IP网络编号相同的主机应该处于同一个网络内，反之亦然。但一个独立系统中，往往包含不止一个 IP网络。

由于一个网络可能由上百个小型网络集合而成，各小型网中还有诸如以太网的更小的物理网络单元，所以，在这种网络内部提供类似于前面的方案是非常有意义的。这样，我们就可以把IP网络分成若干个子网。

子网主要负责从自己所处的 IP网络，把数据报投到特定范围内的 IP地址。A、B或C类地址的识别，是通过 IP地址的网络部分来完成的。但是现在的网络编号被扩展到包含主机部分的位数。这些被视作子网编号的位数就是所谓的子网掩码（或网络掩码）所赋予的。它同样是一个32位的编号，用于为IP地址的网络编号指定位掩码。

Groucho Marx大学(GMU)校园网就是一个很好的范例。它采用的是 B类地址，其网络编号是149.76.00，所以其网络掩码是255.255.0.0。

从其内部结构来说，GMU大学校园网由若干个小型网组成，比如各个系的局域网。所以，这个IP网络被分为254个子网，IP地址在149.76.1.0到149.76.254.0之间。举个例子来说，理论物理系分配的IP地址是149.76.12.0。校园主干网的地址是 149.76.1.0。这些子网共享同一个 IP编号，其中的第三个8位元是用来区分它们的。所以，它们的子网掩码将是 255.255.255.0。

值得注意的是子网只是一个网络内部分区。子网是由网络拥有者（或管理员）一手炮制的。通常情况下，子网主要用于反映现有的地址边界，用于各子网间的物理上（两个以太网之间）、管理上（两个系之间）或地理上的边界。但是，这类结构不仅会影响整个网络的内部行为，而且子网只能本地识别，其地址仍然被看成是标准的 IP地址。

2.4.3 网关

子网不仅能带来结构上的好处，还时常用来反映硬件边界。具体物理网络上的主机，比

如以太网，是非常受限的：它能够直接与之交谈的主机只能是本网络内的。要对其他的主机进行访问，只有通过所谓的“网关”来进行。网关是同时连接两个或两个以上物理网络的主机，被配置为执行网络间的包交换。

对IP网络来说，要想轻松识别主机是否在本地网络，不同的物理网络只能属于不同的 IP网络。比如，网络编号149.76.4.0是为数学系局域网上的主机保留的。在向 Quark发送一个数据报时，Erdos主机上的网络软件立即就能知道该数据报来自 149.76.12.4这个IP地址，而且其目标主机处于另一个物理网络上。因此，这个数据报只能通过一个网关（默认设置是 Sophus）抵达目的地。

Sophus本身连接了两个子网：数学系的局域网和校园主干网。它分别通过两个不同的接口（eth0和fddi0）访问这两个子网。现在，我们为这个网关分配什么样的 IP地址呢？应该根据149.76.1.0子网进行分配，还是根据149.76.4.0子网进行分配？

答案是：两者都要。在提到数学系局域网上的主机时，就应该用 149.76.4.1这个IP地址；在提及校园主干网上的主机时，就应该用149.76.1.4这个地址。

所以，这个网关就有两个IP地址。这两个地址——还有其相应的网络掩码——都绑在接口上（通过这个接口访问子网）。因此，接口和Sophus地址之间的对应关系就会像表 2-1列出的那样。

表2-1 接口和Sophus地址之间的对应关系

接 口	地 址	网络掩码
eth0	149.76.4.1	255.255.255.0
fddi0	149.76.1.4	255.255.255.0
l0	127.0.0.1	255.0.0.0

最后一条说明的是回送接口l0，我们在前面已经介绍过它。注意，随两个子网上的主机同时出现的还有两个地址。

一般说来，把地址附于主机和把地址附于其接口之间有些细微差别，但我们可以忽略它们之间的差别。对处于同一个网络的主机来说，比如 Erdos，一般能够很确切地指出主机的 IP地址是多少。比如说，这是一个以太网接口，它的 IP地址是这样的。但是在提及网关时，其间的区别则是不容忽视的，必须指明地址是附于主机上，还是附于其接口上。

2.5 路由表

接下来的重头戏是：把数据报投到远程网络时， IP协议是如何选择网关的。

在此之前，我们曾见过 Erdos网关的工作流程：它收到发向 Quark的数据报后，便对其目标地址进行检查，并发现其目标主机不在本地网络内。所以，它把该数据报发给默认网关 Sophus，现在就看网关怎样运作了。 Sophus识别出Quark不在它直接连接的任何一个网络内，所以，它就开始寻找另一个网关来接替它的工作。于是它选中了 Niels，这是通往物理系局域网的网关。如此一来， Sophus就需要更多的信息，把目标网络和一个适当的网关关联起用。

它采用的路由选择信息 IP实际上就是一个表，把网络和准备抵达的网关链接起来。一般说来，我们必须提供一个catch-all条目(默认路由)；它是一个与0.0.0.0网络关联在一起的网关。发向未知网络的所有包都会通过这个默认路由得以发送。针对 Sophus网关，它的路由表就像表2-2一样：

表2-2 Sophus的路由信息表

网 络	网 关	接 口
149.76.1.0	…	fddi0
149.76.2.0	149.76.1.2	fddi0
149.76.3.0	149.76.1.3	fddi0
149.76.4.0	…	eth0
149.76.5.0	149.76.1.5	fddi0
…	…	…
0.0.0.0	149.76.1.2	fddi0

对直接与Sophus连接的网络来说，通向它的路由不需要网关，所以显示的网关条目是“ - ”。

路由表的建立方式较多。对小型局域网来说，在启动时间（参见第 3章），利用route命令，手工构建路由表并把它们投入 IP网络，通常是最有效的。对于大型的网络，需要根据路由daemon，在运行时构建路由表并适时进行调整；路由信息运行于网络内的中心主机上，并在其成员网络间实行路由信息交换，从而计算最佳路由。

根据网络的不同大小，可能会用上不同的路由协议。对独立系统（比如 Groucho Marx校园网）内的路由选择来说，将采用内部路由协议。最优秀的是 RIP协议，即路由信息协议。它是由BSD路由daemon实施的。针对独立系统间的路由，则必须使用诸如 EGP（外部网关协议）或BGP（边界网关协议）之类的外部路由协议；这类协议（包括 RIP）已经被用于康奈尔大学的通道daemon中。

度量值

基于RIP的动态路由信息将根据网关“ 跳 ”（hop）数来选定抵达目标主机或网络的最佳路由。也就是说，在抵达目标主机或网络之前，数据报必须经过的网关越少，其 RIP级别就越高。一个网关即为一个跳。超长路由将被视为不可用路由而被丢弃。

要想利用RIP来管理本地网络中的路由信息，必须在所有主机上运行 gated（通道）。在启动时，通道将对所有激活的接口进行检查。如果通道发现激活接口不止一个（不包括回送接口在内），就会假设主机正在进行不同网络间的包交换，从而主动地进行路由信息的交换和广播。如若不然，它就会被动地接收 RIP更新信息，更新本地路由表中的信息。

在广播本地路由表中的信息时，通道将对路由长度进行计算，也就是说用与路由表中的条目关联在一起的度量值来衡量路由长度。这个度量值是系统管理员在配置路由时设置的，它应该能反映出利用该路由产生的花费。因为，对主机直接连接的子网来说，通向它的路由之度量值应该始终为 0，而对通过两个网关的路由，其度量值必须是 2。但需要注意的是，在没有使用RIP或通道时，大可不必理睬度量值。

2.6 Internet控制消息协议

IP还有一个“ 伴侣 ”协议。它就是Internet控制消息协议（ICMP），它是内核联网程序用以与错误消息和其他主机进行通信的协议。比方说，假设我们又回到 Erdos，并打算登录到Quark的12345端口，但这个端口上没有监听进程。所以，发向这个端口的第一个 TCP包就会抵达Quark，网络层将认出这个包并立即向Erdos返回一条ICMP消息，指出“ 不能抵达指定端口 ”。

ICMP能够识别的消息相当多，而且大多数都能对错误情况进行处理。然而，其中有一

条非常有意思的消息，叫作“重定向”消息。它是在有更短路由的情况下，发现另一个主机正把它用作一个网关时，由路由选择模块生成的。例如，在启动之后， Sophus的路由表可能会不完整，其中包含通向数学系局域网和 FDDI主干网的路由，以及通向 Groucho计算中心的网关（gcc1）的默认路由。因此，任何一个发向 Quark的包都会被发送到gcc1，而不是物理系局域网的Niels网关。在收到这类包后， gcc1将注意到这一路由非常糟糕，所以在把包转发到Niels时，向Sophus返回一条ICMP重定向消息，并将最佳路由告诉它。

现在看来，手动配置路由似乎比必须设立路由简单的多。但要注意，单纯依赖于动态路由方案以及RIP和ICMP重定向消息，始终不是上策。在验证某些路由信息是否真正可靠时，ICMP重定向消息和RIP能够提供的选择很少，甚至没有。这样某些恶意的、一无是处的包将扰乱你的整个网络通信，甚至可能导致网络瘫痪。鉴于此，联网程序有几个版本，对影响网络路由的重定向消息进行了处理，令其只能对主机路由进行重定向。

2.7 域名系统

2.7.1 主机名解析

如上所述，TCP/IP连网协议中的地址利用的是 32位的编号。但是，有几个人能够一口就答上来，某某主机的 IP地址是什么呢？因此，主机一般都有一个“普通”的（比如“高斯”或“异人”）主机名。然后，由特定的应用程序负责找出和这个主机名对应的 IP地址。这个过程就叫作“主机名解析”。

对想找出与具体主机名对应的 IP地址的应用程序来说，不必为主机和 IP地址的查找提供它自己的例程。相反地，它依靠大量的库函数进行透明操作，其中有： gethostbyname(3)和gethostbyaddr(3)。按照惯例，这两个函数和大量的相关进程都被归在一个独立的库内，这个库叫作解析器库；Linux中，它们是属于标准 libc的。所以说白了，这个函数集合就是“解析器”。

现在，在像以太网之类的小型网络上，甚至在一个以太网聚簇上，要维护主机名和 IP地址的对应表不是件难事。维护信息通常保存在一个名为 /etc/hosts的文件中。在增添或删除主机名或重新分配IP地址时，只须根据所有的主机，更新主机名即可。显而易见，这对由若干台机器组成的网络来说，是颇为头疼的。

要解决这一问题，方法之一是NIS（网络信息系统），这是Sun公司开发的。简单地说，叫作YP或黄页。NIS把主机文件（和其他信息）保存在一个主要主机的数据库内，客户机如有需要，就可从该数据库内提取。但是，这个方法仍然只适用于中等大小的网络，比如局域网，因为它只对整个主机数据库进行集中维护，并把它分发到各个服务器。

在因特网上，地址信息最初也是保存在一个独立的HOSTS.TXT数据库内。这个文件的维护在“网络信息中心”（NIC）进行，而且必须由所有参与站点进行下载和安装。随着网络的扩大，这一方案的问题就越来越突出。除了定期安装HOSTS.TXT涉及的管理开销外，各个服务器的下载量也越来越大。更为严重的是，所有主机名都必须利用NIC进行注册，以保证主机名不重复。

这就是1984年新的主机名解析方案出台的原因。它就是域名系统（ DNS）。DNS是保罗·莫克皮特里斯设计的，完满解决了上面提到的两大问题。

2.7.2 输入DNS

DNS采用域分层结构来管理主机名。一个域是若干个站点的集合，因为这些站点可以组成一个网络（例如，校园网内的所有机器，或 BITNET上的所有主机），而且因为它们都属于某个特定的组织（比如美国政府），或仅仅因为它们的地理位置相当接近。例如，所有的大学可以归为一个.edu域，而各个大学或学院又分为一个单独的子域。如果为 Groucho Marx大学指定的是groucho.edu域，那么，为数学系局域网分配的域就是 maths.groucho.edu。数学系局域网内的各台主机分到的域就是再在前面的域名前加上主机名，比如 Erdos的就是erdos.maths.groucho.edu。这就是所谓的“完整资格域名”（FQDN），它能够唯一地标识全球各地的主机。

这个域分层结构的根是一个单独的点（.），名副其实地称作根域（rootdomain），所有的域都包含在这个根域内。为指明一个主机名的确是一个完整资格域，而不是相对于某一（隐式）本地域的域名，有时会把它表示成一个追尾点。它表示该名的最后一个组件是根域。

根据其在域名分层结构中的位置，域可以称作顶级域、二级域或三级域。多级子分区不是没有，但极为罕见。下面是一些大家经常能见到的顶级域：

- .edu （主要在美国）教育部门，比如大学
- .com 商业组织，公司
- .org 非商业组织（这个域中通常是些UUCP网络）
- .net 网络上的网关和其他管理性主机
- .mil 美国军方
- .gov 美国政府部门
- .uucp 正式场合中，对以前用做UUCP名的所有站点名来说，如果没有为它们指定域，就会被统统归入这个域内。

从技术上来讲，前4个域属于Internet的美国部分。但这个域中同样包含有非美国站点。特别是.net域中，此类非美国站点比比皆是。但是，.mil和.gov是美国专用域。

除美国外的其他国家都可以使用得名于 ISO-3166中定义的两个字母，来作为其顶级域。比如芬兰，它使用的是 .f域；法国使用的是 .fr域；德国使用的是 .de域；澳大利亚使用的则是.au域。在这个顶级域下，各个国家可按照自己的方式组织主机名。比如澳大利亚，它有类似于国际顶级域的二级域，名为com.au和edu.au。其他的国家，比如德国，则不采用这种结构，而是采用一个稍长的名字直接指向正在运行特殊域的组织（或企业）。例如，ftp.informatik和uni-erlangen.de此类的主机名是很不常见的，但在德国，类似的主机名却非常普遍。

当然，这些国家域并不能代表这个域下面的主机真正就位于该国；它只是表示这台主机在这个国家的NIC（国家Internet中心）已经注了册。比如，一个瑞士厂商可能在澳大利亚有一个分部，仍然可以用.se顶级域来注册其下属所有主机。

现在，对域名分层结构内的域名空间进行组织，就能很好地解决域名唯一性的问题。利用DNS，主机名在其自己所处的域内，必须是独一无二的，这样才能保证它的名字有别于全球各地的主机。此外，完整资格域名也非常容易记，正因为有了它们，便可将一个大型的域分成若干个子域。

但DNS还有别的好处：它允许你选择子域内的授权者作为它的管理员。比如， Groucho计

算中心的维护人员可能为各系创建一个子域；我们已见识过数学系和物理系子域。在发现物理系局域网太大，太混乱，以至于很难自外部进行管理时（毕竟，学物理的人是出了名的“放荡不羁”），计算中心的维护人员会把physics.groucho.edu域的控制权移交给该网络的管理员。然后，该网络的管理员就可能自由地使用自己喜欢的主机名，并在不与网络外部发生冲突的前提下，按照自己的方式为主机分配IP地址。

最后，域名空间分为若干个区，每个区都包含在一个域内。注意，区和域之间的细微差别：groucho.edu域内涵括Groucho Marx大学的所有主机，而groucho.edu区内只包括计算中心直接管理的主机，比如数学系局域网内的主机。而物理系的主机则属于另一个不同的区，名为physics.groucho.edu。

2.7.3 利用DNS进行名字查找

乍一看，域和区几乎令域名解析复杂不堪。毕竟，如果没有集中式授权机构对主机名分配的控制，小小应用程序怎能识别出这些纷繁复杂的域名？

现在该来谈谈DNS的妙用了。如果想找到Erdos主机的IP地址，DNS就会说，去问管理这台主机的人，他们会告诉你的。

事实上，DNS是一个巨大的分布式数据库。它是通过一个所谓的“域名服务器”来实现的，这些域名服务器将提供具体域或域集合相关的信息。对每个区而言，至少有两个域名服务器中保存有那个区中的所有主机之验证信息。要想获得Erdos的IP地址，只须和groucho.edu区的域名服务器进行沟通，然后，它就会返回你所需要的数据。

大家可能会这样想：“说时容易，做时难。我怎样才能抵达Groucho Marx大学的域名服务器呢？”如果你的计算机没有装地址解析先知，DNS有。在你的应用程序想查找关于Erdos的相关信息时，它会与一个本地域名服务器取得联系，该服务器将为此实施所谓的“交互式”查询。向根域的一个域名服务器发出查询之后，该服务器便要求得到erdos.maths.groucho.edu的IP地址。根域名服务器认出这个名字不属于自己的辖区，而是属于.edu域下面的一个辖区。所以，它就会要求你与.edu区域名服务器取得联系，并将列有所有.edu域名服务器及其地址的清单封装起来。然后，你的本地域名服务器将开始对清单中的域名服务器一一进行查找，比如a.isi.edu域名服务器。它采用的方式类似于根域名服务器的方式，它知道groucho.edu处的人们在运行它们自己的区，并把你引向他们的服务器。最后，本地域名服务器再向其中一个服务器提出查找Erdos，Erdos所在的域名服务器认出它是属于自己这一区之后，便返回其对应的IP地址。

现在看来，查找一个小小的IP地址似乎会产生很大的网络开销，但事实上，和目前尚未解决的HOSTS.TXT传输比较起来，简直不足一提。但仍有必要对这一查找进行改进。

为了缩短查找时间，域名服务器将把获得的信息保存在自己的本地缓存内。所以，下一次你的网络中，有人想查找groucho.edu域内的主机之IP地址时，你的域名服务器就不必费心了，直接联系groucho.edu域名服务器即可。如果域名服务器不保存获得的信息，DNS和其他的方法一样糟糕，因为每次查找都涉及到根域名服务器。

当然，域名服务器不会永久性地保存这一信息，隔段时间后，它就会丢弃这一信息。何时丢弃由生存时间（也就是TTL）决定。DNS数据库中的每项资料都有一个由区管理员分配

的一个TTL。

2.7.4 域名服务器

容纳某一区内主机信息的域名服务器叫做这个区的验证服务器，有时也被称为主域名服务器。任何对区内主机的查询最终都会涉及到主域名服务器。

为了保证整个区的连贯性，其主域名服务器必须能够得到同步更新。这是通过令其中一个主域名服务器成为首要服务器来实现的。这个首要服务器定期令传输区数据的其他域名服务器作为其从属服务器，从而从数据文件中载入该区信息。

要有若干个域名服务器的原因之一是：分担工作负荷。另一个原因是备用。如果一个域名服务器良性“失效”，比如系统崩溃或丢失网络链接时，所有的查询都会转向其他的服务器。当然，这一方案并不能有效地避免服务器故障（导致所有的 DNS请求做出错误的应答）和服务器程序本身出现软件错误。

当然，我们也可以设想一些域内不存在作为验证域名服务器的情况（但一个域名服务器至少能为本地主机和127.0.0.1逆向查找提供域名服务）。不管怎么说，这类服务器都是相当有用的，因为它仍然能够针对运行于本地网络的应用程序实施 DNS查询。因此，它们通常也被称为“caching-only”（只用于缓存）服务器。

2.7.5 DNS数据库

由上可知，DNS不仅能够处理主机的IP地址，还能够交换关于域名服务器的信息。事实上，DNS数据库内可能有整整一打不同类型的条目。

DNS数据库内，一条单一的信息叫作一条资源记录，或简称 RR。每条记录都有一个与之关联的类型（描述了该记录代表的数据类别）和一个类（指定该记录适用的网络类型）。后者可根据不同的编址方案（比如 IP地址，是IN类；Hesiod网络则是MIT）需求进行调节。标准的资源记录类型是A记录，它把一个完整资格域名和一个IP地址关联在一起。

当然，一个主机可以有若干个主机名。但是，必须将其中一个主机名标识为正式名或规范名，而其他的则是可以代表前者的别名。其间的区别是：规范主机名是 A记录所关联的主机名，而其他的则只有一条CANME类型的记录，该记录指向规范主机名。

关于所有的记录类型，将留在下一章深入讨论。这里只是简要介绍一下。

除了A和CNAME类记录外，我们还看到文件顶部有一条特殊的记录。这是 SOA资源记录，表示Start of Authority，其中容纳服务器所属的区之普通信息。比如，它包含所有记录的默认TTL值。

注意，不是以点“.”结尾的示范文件中，所有主机名都应该被解释为与 groucho.edu域有关。SOA记录中的特殊名“@”代表这个域本身的域名。

由上可知，groucho.edu域的域名服务器多少应该知道物理区，使之能够将主机查询引向这些主机各自的域名服务器。这通常是通过两条记录来完成的：指定该服务器之 FQDN的NS记录和把地址与主机名关联在一起的 A记录。由于这两条记录把域名和空间集中保存在一起，所有它们通常也被称作“glue records”。对真正保存子区内主机相关信息的父区来说，它们是唯一的记录实例。指向physics.groucho.edu域的域名服务器的glue记录如清单2-1所示。

清单2-1 物理系采用的程序，摘自named.hosts文件

```
;
; Authoritative Information on physics.groucho.edu
@                       IN      SOA             (
                        niels.physics.groucho.edu.
                        hostmaster.niels.physics.groucho.edu.
                        1034                    ; serial no
                        360000                  ; refresh
                        3600                    ; retry
                        3600000                 ; expire
                        3600                    ; default ttl
                       )
;
; Name servers
        IN      NS      niels
        IN      NS      gauss.maths.groucho.edu.
gauss.maths.groucho.edu. IN A           149.76.4.23
;
; Theoretical Physics (subnet 12)
niels                   IN      A       149.76.12.1
                        IN      A       149.76.1.12
nameserver              IN      CNAME   niels
otto                    IN      A       149.76.12.2
quark                   IN      A       149.76.12.4
down                    IN      A       149.76.12.5
strange                 IN      A       149.76.12.6
...
; Collider Lab. (subnet 14)
boson                   IN      A       149.76.14.1
muon                    IN      A       149.76.14.7
bogon                   IN      A       149.76.14.12
...
```

2.7.6 逆向查找

除了查找属于一个主机的 IP地址外，我们有时还希望查找和 IP地址对应的规范主机名。这就是所谓的“逆向映射”，有几个网络服务用它来验证客户机的身份。在使用一个独立的主机文件时，逆向查找只在该文件中查找问题中 IP地址所对应的主机名。如果用 DNS，当然还要对问题外的名字空间进行彻底查找。 DNS创建了一个in-addr.arpa特殊域，其中包含所有主机的IP地址，这些地址采用的格式是点分四段式。例如， 149.76.12.4这个IP地址对应的主机名是4.12.76.149.in-addr.arpa。把这些域名和其规范主机名链接起来的资源记录类型是 PTR。具体程序参见清单2-2。

清单2-2 GMU采用的程序，摘自named.hosts文件

```
;
; Zone data for the groucho.edu zone.
@                       IN      SOA             (
                         vax12.gcc.groucho.edu.
                         hostmaster.vax12.gcc.groucho.edu.
```

```
                        233             ; serial no
                        360000          ; refresh
                        3600            ; retry
                        3600000         ; expire
                        3600            ; default ttl
                        )
....
;
; Glue records for the physics.groucho.edu zone
physics                 IN      NS      niels.physics.groucho.edu.
                        IN      NS      gauss.maths.groucho.edu.
niels.physics           IN      A       149.76.12.1
gauss.maths             IN      A       149.76.4.23
...
```

创建一个特区通常意味着其管理员能够拥有分配 IP地址的绝对支配权。由于他们手中有一个或若干个IP网络或子网，所以DNS区和IP网络之间可能会存在一对多的映射关系。比如，物理条由149.76.8.0、149.76.12.0和149.76.14.0三个子网组成。

结果是，in-addr.arpa域内的新区必须随物理区一起创建，而且供该系的网络管理员专用，它们是：8.76.149.in-addr.arpa、12.76.149.in-addr.arpa和14.76.149.in-addr.arpa。否则的话，在Collider实验室安装一台新主机时，就会要求新区与其父域取得联系，以便将新地址输入它们的in-addr.arpa区文件。

子网12所用的区数据库参见清单2-3。

清单2-3 子网12所用的数据库，摘自in-addr.arpa文件

```
;
; the 12.76.149.in-addr.arpa domain.
@                   IN      SOA     {
                        niels.physics.groucho.edu.
                        hostmaster.niels.physics.groucho.edu.
                        233 360000 3600 3600000 3600
                        }
2                   IN      PTR         otto.physics.groucho.edu.
4                   IN      PTR         quark.physics.groucho.edu.
5                   IN      PTR         down.physics.groucho.edu.
6                   IN      PTR         strange.physics.groucho.edu.
```

这些新区之父区所用的记录，则参见清单2-4。

清单2-4 网络149.76所用的记录，摘自named.rev文件

```
;
; the 76.149.in-addr.arpa domain.
@                     IN        SOA           {
                       vax12.gcc.groucho.edu.
                       hostmaster.vax12.gcc.groucho.edu.
                       233 360000 3600 3600000 3600
                      }
...
; subnet 4: Mathematics Dept.
```

```
1.4                IN      PTR       sophus.maths.groucho.edu.
17.4               IN      PTR       erdos.maths.groucho.edu.
23.4               IN      PTR       gauss.maths.groucho.edu.
...
; subnet 12: Physics Dept, separate zone
12                 IN      NS        niels.physics.groucho.edu.
                   IN      NS        gauss.maths.groucho.edu.
niels.physics.groucho.edu. IN  A 149.76.12.1
gauss.maths.groucho.edu. IN  A   149.76.4.23
```

上面的记录产生的重要结果是验证区只能被视作 IP网络的超级子集，而且更严格地说，这类网络的网络掩码必须根据字节边界来定。 Groucho Marx大学的所有子网的网络掩码都是255.255.255.0，因此，应该为每个字网创建一个 in-addr.arpa区。但是，如果各子网的网络掩码是255.255.255.128，我们就可以为子网 149.76.12.128创建验证区，因为无法告知 DNS“ 12.76.149.in-addr.arpa域已经被分为两个特区，所以主机名分别是 1到127，和128到255 ”。