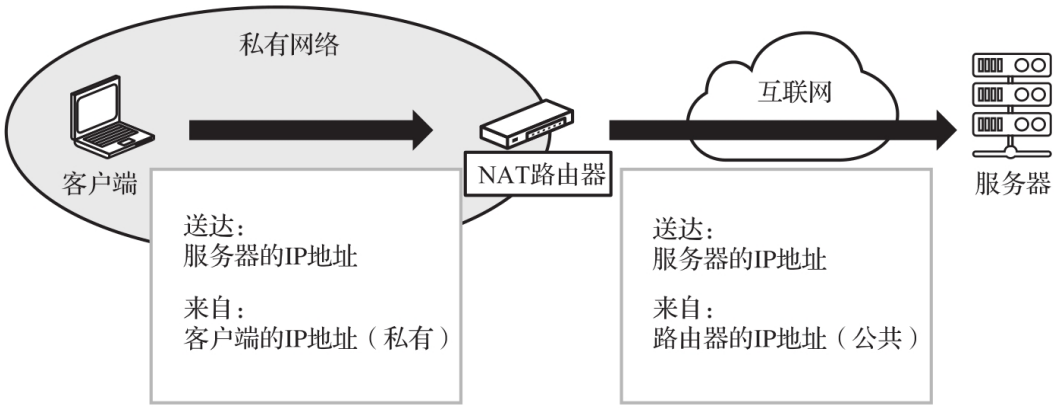

图11-16 NAT路由器把私有IP地址替换为它自己的公网IP地址

当响应返回路由器时，它把目标IP地址设置为发起请求的主机的私有地址。如此一来，所有来自这个家庭网络的流量似乎都来自同一个公网IP地址，即使这个私有网络上实际有多个设备也一样。NAT还附带一个安全方面的好处：私有网络上的设备不会直接暴露在公网上，所以互联网上的恶意用户不能直接发起对私有设备的连接。大多数卖给消费者用于家庭网络的路由器都是NAT路由器，一般还有内置无线接入点功能。

私有IP地址不仅对家庭网络是有价值的，还对那些不想将其计算机暴露于公网的企业也是有价值的。许多公司网络使用的是代理服务器，而不是NAT路由器。在允许私有网络上的设备访问互联网这一点上，代理服务器类似于NAT路由器，但不同的是，代理服务器一般工作于应用层，而不是网络层。通常，代理还提供其他的功能，比如用户身份验证、流量日志记录以及内容过滤。

注意

请参阅设计34查看设备所分配的IP地址是公网IP地址还是私有IP地址。

11.4.3 域名系统

我们已经看到互联网上的主机是用IP地址来识别的。但是，互联网的大多数用户很少（如果有的话）直接面对IP地址。尽管IP地址适用于计算

机，但它们对用户很不友好。没有人想要去记住一组用句点分割的四个数字。幸运的是，有域名系统（DNS）让我们更轻松。DNS是一种互联网服务，它把名称映射为IP地址。这使得我们能用类似于www.example.com（而不是IP地址）这样的名称来指代主机。

计算机的完整DNS名称为完全限定域名（Fully Qualified Domain Name，FQDN）。像travel.example.com这样的名称就是一个FQDN。这个名称由一个简短的本地主机名（travel）和一个域后缀（example.com）组成。术语"主机名"（hostname）常常可以交换用于表示计算机的简称或FQDN。本节将使用主机名来表示计算机的FQDN。域（比如example.com）表示由组织管理的一组网络资源。example.com和travel.example.com都是域名。前者表示网络域，后者表示这个域上的一个特定主机。

软件需要能查询DNS以把主机名转换成IP地址——这被称为主机名解析。要启用这个功能，主机需要配置DNS服务器的IP地址列表。这个表一般由DHCP提供，通常由互联网服务提供商维护的DNS服务器或运行在本地网络的DNS服务器组成。当客户端想要通过名称连接到服务器时，它向DNS服务器请求对应该名称的IP地址。如果可以，服务器将用被请求IP地址进行响应，如图11-17所示。

图11-17 简化的DNS查询。example.com的IP地址不准确

客户端有了服务器的IP地址后，它继续用这个IP地址与服务器通信，如前所述。我听说DNS被描述成互联网的电话簿，不过这个类比对某些读者来说可能有些不太合适，因为电话簿没有以前那么普遍了！

你可能会设想IP地址与名称之间是一对一的关系。实际并非如此。一个名称可以映射到多个IP地址。在这种情况下，不同的客户端向DNS查询某个名称，收到的响应可能是不同的IP地址。这对于给定服务负载需要被分布到多个服务器的情况非常有用。它可以在物理上实现，例如，在欧洲的客户端得到的IP地址不同于在亚洲的客户端得到的IP地址，这能让每个区域的客户端连接到在物理上靠近自己的服务器IP地址。

反之亦然：多个名称也可以映射到同一个IP地址。在这种情况下，对不同名称的查询会返回同一个IP地址。当服务器托管同一类型服务的多个实例时，这是很有用的，其中的每个实例都用名称识别。这在网络托管中很常见，当一个服务器托管多个网站时，每个网站都用自己的DNS名称来识别。

DNS中的每个条目都被称为一条记录（record）。记录有不同的类型：最基础的是A记录，它只是把主机名映射到IP地址。其他例子还有CNAME（canonical name，规范名称）记录，它把一个主机名映射到另一个主机名；以及用于电子邮件服务的MX（mail exchanger，邮件交换器）记录。

没有一个组织愿意承担管理现存的大量DNS记录的任务。幸而这不是必需的，DNS的实现方式允许责任分担。像www.example.com这样的DNS名实际表示的是一个记录层次结构，不同的DNS服务器负责维护这个层次结构中不同级别的记录。应用于www.example.com的DNS层次结构如图11-18所示。

这个层次结构树的顶端是根域。根域不会像www.example.com那样得到文本表示的DNS名，但它是DNS层次结构的重要组成部分。根域包含了所有TLD（Top-Level Domain，顶级域）的记录，比如.com、.org、.edu、.net等。截至2020年，世界范围内有13个根名服务器，每个根名服务器都负责了解所有顶级域服务器的详细信息。假设你想查找以.com结尾的域内的一条记录。根服务器可以指向TLD服务器，这个服务器了解.com下的域。

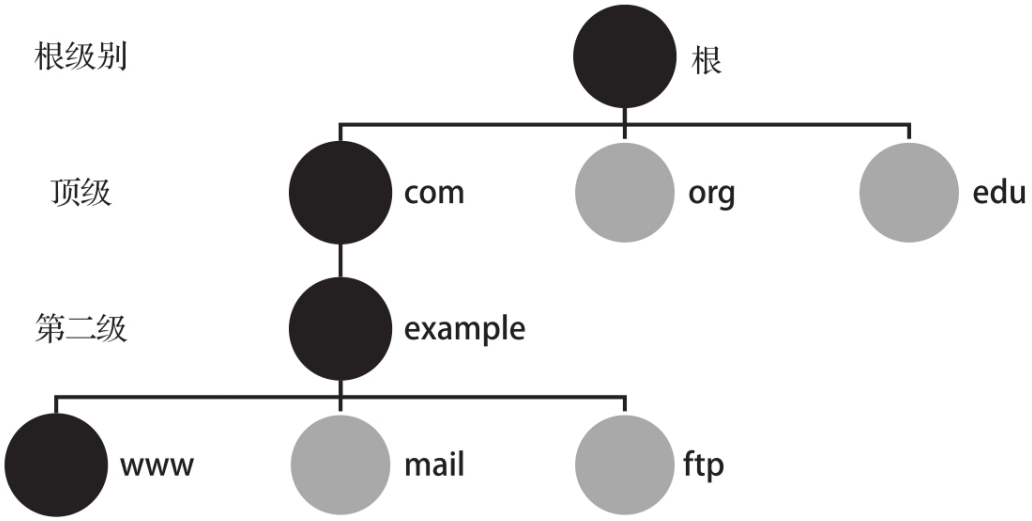

图11-18 DNS层次结构示例，突出显示www.example.com

顶级DNS服务器负责了解其下所有的二级域。.com的顶级DNS服务器可以指向example.com的二级DNS服务器。二级域的DNS服务器维护属于二级域的主机与三级域的记录。这意味着example.com的DNS服务器要负责维护类似于www.example.com和mail.example.com之类的主机记录。这种模式不断重复，并允许嵌套域。一个域在顶级域下注册后，该域的拥有者就可以在这个域的下面创建任意数量的记录。

如前所述，当计算机需要查找FQDN的IP地址时，它向其配置的DNS服务器发送一个请求。DNS服务器收到请求后要做什么呢？如果服务器最近查找过被请求的记录，那么它的缓存中可能保存有这个记录的副本，服务器可以立刻把IP地址返回给客户端。如果DNS服务器在缓存中没有响应，它可能根据需要查询其他的DNS服务器以获得答案。这涉及从根开始，沿着服务器层次结构向下来查找被请求的记录。当服务器得到这条记录后，它可以缓存该记录，这样将来在查询该记录时，服务器就可以立即予以响应。最终，缓存的记录会被删除，以确保服务器总是能提供合理的最新数据。

注意

请参阅设计35查看DNS的信息。

11.5 网络即计算

让我们花点时间来考虑这样一个问题：互联网是如何适应本书已经介绍过的更广泛的计算图景的。网络看上去像是一个无关紧要的话题，但实际上它与一般的计算并无太大差别。互联网由硬件和软件组成，它们协同工作，允许设备之间进行通信。在互联网上传输的数据可以归结为0和1，它们用各种形式表示，比如导线上的电压。从计算机的角度来看，网络接口（比如Wi-Fi或者以太网适配器）就是另一个I/O设备。操作系统通过设备驱动程序与这样的适配器进行交互，且操作系统中包括了能让应用程序在互联网上轻松通信的软件库。像路由器和交换机这样的网络设备也是计算机，只不过是高度专业化的设备。互联网和一般的网络是本地计算机的延伸，允许在单个设备边界之外进行数据传送和处理。

11.6 总结

本章介绍了互联网，一组全球连接的计算机网络，它们都使用一套通用协议。通过本章，你了解了互联网协议套件的四个协议层——链路层、网络层、传输层和应用层。你看到了数据如何穿过互联网，设备如何在不同的层进行交互。你了解了DHCP如何提供网络配置数据，NAT如何允许私有网络上的设备连接到互联网，DNS如何提供能替代IP地址的友好名称。第12章将介绍万维网，由HTTP通过互联网提供的一组资源。

设计29：查看链路层

前提条件：运行Raspberry Pi操作系统的Raspberry Pi。

在本设计中，你将使用Raspberry Pi来查看本地网络的链路层。我们首先使用下面的命令列出以太网适配器的MAC地址：

```
$ ifconfig eth0 | grep ether
```

输出应如下所示：

```
ether b8:27:eb:12:34:56  txqueuelen 1000  (Ethernet)
```

本例中，MAC地址是b8:27:eb:12:34:56。这是48位数的十六进制表示。请记住，每个十六进制字符表示4个位，所以12个字符×4位=48位。

MAC地址的前24位表示硬件的供应商/制造商。这个数字被称为组织唯一标识符（Organizationally Unique Identifier，OUI），由电气和电子工程师协会（Institute of Electrical and Electronics Engineers，IEEE）管理。本例中的OUI是B827EB，它被分配给Raspberry Pi基金会。

Raspberry Pi的Wi-Fi适配器也有自己的MAC地址。它可以用下面的方式查看：

```
$ ifconfig wlan0 | grep ether
        ether b8:27:eb:78:9a:bc  txqueuelen 1000  (Ethernet)
```

在我的系统中，Wi-Fi适配器的OUI（MAC地址的前24位）与以太网适配器的OUI相同。这是因为两个适配器都是Raspberry Pi内置硬件，都使用了Raspberry Pi基金会的OUI。

从Raspberry Pi上，你还可以看到本地网络上其他设备的MAC地址。为此，你可以使用名为arp-scan的工具，该工具尝试连接本地网络上的每个计算机，并检索其MAC地址。

首先，安装该工具：

```
$ sudo apt-get install arp-scan
```

然后，运行以下命令（命令的结尾是小写字母l，不是数字1）：

```
$ sudo arp-scan -l
```

你应该得到IP地址和MAC地址的列表，以及一个试图把MAC前缀与制造商进行匹配的列。我在自己的本地网络上得到了10个结果，其中的一些我没有立即识别出来。你可能会看到返回了一些重复的结果，第三列中用DUP指示这些结果。返回的列表通常不包含运行扫描的计算机的地址。

你可能会在第三列中看到一些结果显示为（Unknown）。这表示arp-scan工具不能把OUI编号与已知制造商匹配上，可能是因为这个工具使用了未更新的OUI列表。你可以尝试从IEEE下载当前OUI编号列表，然后再次运行扫描来解决这个问题，如下所示：

```
$ get-oui
$ sudo arp-scan -l
```

当看到家庭网络上有多个我无法立即识别的设备时，我马上有一种冲动想要弄清楚它们是什么！作为额外挑战，请你识别出arp-scan返回的每个设备。如果是在你无法控制的网络（比如咖啡馆或图书馆的网络）上运行这个工具，这可能是不现实的，但是，如果你在家里，这就是可以做到的。你可能需要登录网络上的每个设备，挖掘其设置，找到它的IP地址或MAC地址，看看它是否与arp-scan返回的条目匹配。提示：在Linux或Mac上使用ifconfig实用程序，在Windows上使用ipconfig工具。在移动设备上，查看用户界面中的网络设置。

设计30：查看网络层

前提条件：运行Raspberry Pi操作系统的Raspberry Pi。

在本设计中，你将使用Raspberry Pi来查看网络层。我们首先使用下面的命令列出设备上的所有网络接口及其关联的IP地址：

```
$ ifconfig
```

一般你会看到3个接口：eth0、lo和wlan0。lo接口是特殊情况：它是回送接口（loopback interface）。它用于在Pi上运行的进程，这些进程希望用TCP/IP相互通信，但实际上又不会在网络上发送任何流量。也就是说，流量停留在设备上。回送接口的IP地址为127.0.0.1。这是个特殊地址，它不能被路由，不能用作本地子网上的地址，因为任何向该地址传送消息的尝试都会导致该消息直接回到发送计算机。换句话说，每个计算机都把127.0.0.1看作自己的IP地址。

正如我们在设计29中看到的，eth0是有线以太网接口，wlan0是无线Wi-Fi接口。如果你通过这两个接口中的一个或两个连接到网络，你就会在ifconfig输出的inet文本旁边看到一个IP地址。你可能还会在inet6的旁边看到一个IPv6地址。下面是ifconfig的wlan0输出示例：

```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.138  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::8923:91b2:13e0:ed2a  prefixlen 64  scopeid 0x20<link>
```

在这个输出中，你可以看到被分配的IP地址是192.168.1.138。netmask的值（子网掩码）是255.255.255.0，广播地址（broadcast）是192.168.1.255。

ifconfig命令为我们提供了Raspberry Pi上各种网络接口的信息，但它没有告诉我们路由是如何配置的。让我们用ip route命令来看一下。我在这里给出了示例输出，你的结果可能与之不同。

```
$ ip route
default via 192.168.1.1 dev wlan0 src 192.168.1.138 metric 303
192.168.1.0/24 dev wlan0 proto kernel scope link src 192.168.1.138 metric 303
```

该命令的输出解释起来可能有点难度。简单来说，第一行给出了默认路由。当没有应用特定路由时，这是应该发送数据包的地方。在这个特定的例子中，每个没有匹配特定路由规则的包都应该发送到192.168.1.1。这意味着192.168.1.1是本地路由器的IP地址，也被称为默认网关。

下一行是一个路由条目，它告诉你任何发送到IP地址范围在192.168.1.0/24之内的包，都应该通过设备wlan0发送。这是本地子网上的Wi-Fi适配器。换句话说，这个路由规则确保与本地子网上的IP地址是直接通信的，无须通过路由器。

总而言之，这个输出告诉你，发送到与192.168.1.0/24匹配的IP地址的任意包都应该通过wlan0接口直接发送到目标地址。其他所有流量都使用默认路由，它会把流量发送到路由器192.168.1.1。最终结果是，本地子网流量被直接发送到目标设备，而到其他子网设备（可能在互联网上）的流量则被发送到默认网关。

设计31：查看端口使用情况

前提条件：运行Raspberry Pi操作系统的Raspberry Pi。

在本设计中，你将看到Raspberry Pi使用了哪些网络端口。然后，你将查看其他计算机的端口。我们首先使用下面的命令显示Raspberry Pi上监听和已建立的TCP套接字：

```
$ netstat -nat
```

让我们分析一下命令中使用的-nat选项。n选项表示应该使用数字输出来显示端口号。a选项表示显示所有的连接（监听的和已建立的），t表示把输出限制为TCP。在我的设备上，我看到如下列表：

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0     36 192.168.1.138:22         192.168.1.125:52654    ESTABLISHED
tcp        0      0 192.168.1.138:22         192.168.1.125:51778    ESTABLISHED
tcp6       0      0 :::22                   :::*                    LISTEN
```

这里，你看到4个套接字，它们都与SSH有关。之所以说它们与SSH有关，是因为所有的套接字都使用了端口22。我启用了Raspberry Pi上的SSH，以允许远程终端连接。第一行和最后一行显示，Pi正在端口22上监听使用TCP和IPv6上的TCP新传入的SSH连接。中间两行显示，这个设备有两个已建立的SSH连接，这两个连接都是从我的笔记本计算机（IP为192.168.1.125）到Pi（IP为192.168.1.138）。请注意，这两个已建立的连接是如何连接到Pi上同一个服务器端口（22）的，而我笔记本计算机上的客户端端口则不同（52654和51778），因为它们是临时端口。

再次运行该命令，这次加上p选项并在命令的前面增加前缀sudo：

```
$ sudo netstat -natp
```

这会为你提供相同的列表，只不过增加了套接字所属的进程的ID（PID）和程序名。任何发送到套接字的流量都会被定向到PID，进程处理

流量并按需予以响应。在我的计算机上，我看到使用这个端口的程序是sshd——SSH的守护程序。

现在你已经查看了Raspberry Pi上正在使用哪些端口，让我们来看看远程计算机上的端口。为此，你需要使用名为nmap的工具，首先必须在Raspberry Pi上安装这个工具：

```
$ sudo apt-get install nmap
```

工具安装好之后，选择想扫描的目标主机。它可以是网络上的设备（比如路由器或笔记本计算机），也可以是互联网上的主机。请注意，对于服务器的管理员来说，重复扫描你无法控制的主机可能会显得可疑，因此，我强烈建议你只扫描自己的设备。

至于我，我决定扫描自己的默认网关，它恰好是192.168.1.1。下面的nmap命令扫描特定IP地址上的开放TCP端口。在你的Raspberry Pi上尝试使用这个命令，把IP地址替换成你想扫描的设备地址。如果你想扫描自己的路由器，请参阅设计30回忆一下如何得到默认网关的IP地址。

```
$ nmap -sT 192.168.1.1
```

扫描结果的部分列表给出了如下端口：

```
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
```

这告诉我设备不仅充当了路由器，还充当了DNS服务器（端口53）和Web服务器（端口80）。家用路由器提供这些服务很正常。

设计32：跟踪到达互联网上一个主机的路由

前提条件：运行Raspberry Pi操作系统的Raspberry Pi。

在本设计中，你将查看从Raspberry Pi发送数据包到互联网上主机的路由。首先，你需要在互联网上选择一个主机，它可以是像www.example.com这样的网站，也可以是你恰好知道的互联网主机的IP地址或FQDN。确定主机后，输入如下命令（用你想查看的主机名或IP地址替换www.example.com）：

```
$ traceroute www.example.com
```

traceroute工具尝试显示数据包在互联网上传送时遇到的路由器。输出应逐行读取。每一行按顺序编号，显示数据包传送线路上某一个步骤遇到的路由器的名称或IP地址。如果短时间内没有响应，输出会显示星号（*）并移动到下一个路由器。你还可能在一行上看到多个IP地址，这表示有多个可能的路由。

设计33：查看IP地址

前提条件：运行Raspberry Pi操作系统的Raspberry Pi。

在本设计中，你将查看从DHCP服务器获得的、与Raspberry Pi的IP地址关联的租用信息。当然，这要假设Raspberry Pi被配置为使用DHCP（默认设置）而不是静态IP地址。为此要查看系统日志：

```
$ cat /var/log/syslog | grep leased
```

预期会看到如下输出：

```
Jan 24 19:17:09 pi dhcpcd[341]: eth0: leased 192.168.1.104 for 604800 seconds
```

这里，你可以看到IP地址192.168.1.104是从DHCP租用的，用于网络接口eth0，即Raspberry Pi的以太网接口。你的输出可能显示不同的IP地址和不同的接口，可能是wlan0。

默认情况下，syslog文件会定期清理，它的内容会被移动到备份文件中。因此，你可能在syslog文件中看不到DHCP条目。你可以释放当前的IP

地址，请求一个新的IP地址，然后按下面的步骤再次查找租用条目：

```
$ sudo dhclient -r wlan0
$ sudo dhclient wlan0
$ cat /var/log/syslog | grep leased
```

如果你想对以太网而不是Wi-Fi接口做上述工作，请把wlan0替换为eth0。

设计34：查看设备IP是公有的还是私有的

前提条件：运行Raspberry Pi操作系统的Raspberry Pi。

在本设计中，你将看到Raspberry Pi的IP地址到底是公有的，还是私有的。如果设备有私有IP地址，你还会发现用于与互联网通信的公网IP地址。和前面一样，你可以使用如下实用程序来查看设备被分配的IP地址：

```
$ ifconfig
```

在查找设备所分配的IP地址时，你可能会看到关于127.0.0.1的条目，你可以忽略这一条，因为它用于回送（参见设计30）。如前所述，任何匹配10.*x*.*x*.*x*、172.16.*x*.*x*或192.168.*x*.*x*模式的地址都是私有IP地址。现在，即使你拥有一个这样的私有IP地址，当你访问互联网上的资源时，你还是要间接使用公网IP地址。当你连接到网页或其他互联网服务时，这个地址就是它们所看到的地址。如果你是在家庭网络上，公网IP地址很可能会分配给你的路由器。如果你是在企业网络上，那么公网IP地址可能会分配给该企业网络边缘的代理设备。不论是哪种情况，从本地网络到互联网的所有网络流量都来自这个公网IP地址。

在连接到互联网设备时，为了查找你的设备使用的公网IP地址，一种选择是登录到你的路由器或代理服务器，检查它的网络配置。如果你知道如何向路由器或代理服务器查询这个信息，请放心这样做。不过，由于每个网络设备的模型多少有些差异，这里就不详细介绍这些步骤了。

更通用的选择是查询能返回当前公网IP地址的在线服务。这是可能的，因为设备连接的每个互联网服务器都知道你的IP地址，只要找到一个服务愿意告诉你它所见的IP地址即可。如果你在设备上运行一个Web浏览器，那么最简单的方法可能是向Google查询"我的IP地址"这样的内容。它一般会返回你需要的信息。

如果你正在使用终端（比如Raspberry Pi），则可以使用curl实用程序向返回当前IP地址的网站发出HTTP请求。下面是一些服务示例，这些服务在撰写本书时能用来实现刚才所说的功能：

```
$ curl http://ipinfo.io/ip
$ curl http://checkip.amazonaws.com/


$ curl http://ipv4.icanhazip.com/
$ curl http://ifconfig.me/ip
```

上面这些服务都可以把你的公网IP地址返回到终端窗口。将这个地址与之前你从ifconfig得到的地址进行比较。如果它们相同，那么你的设备是直接分配的公网IP地址。如果它们不同，那么你的设备可能被分配了私有IP地址，你是通过NAT路由器或代理服务器连接到互联网的。

设计35：在DNS中查找信息

前提条件：运行Raspberry Pi操作系统的Raspberry Pi。

在本设计中，你将使用Raspberry Pi来查询DNS记录。我们先查找一下网站的IP地址。你将使用host实用程序完成这项工作。下面的命令返回IP地址，www.example.com是我感兴趣的网站的主机名。请随意替换为你想查找的其他主机名。

```
$ host www.example.com
```

你应该看到输出提供了主机的IP地址。你可能还会看到IPv6地址。根据你查询的主机名，你可能会得到多条记录，因为一个DNS名可以映射到

多个IP地址。你还会了解到，你输入的名称实际上是其他名称的别名，这些名称又会映射到IP地址。

DNS还允许反向查找，即指定IP地址，返回主机名。这并非总是有效的，因为需要DNS记录的支持。如果想要试一试，只需要对IP地址使用host。在下面的命令中，用你在设计34中找到的公网IP地址或者其他想要查询的公网IP地址来替换***a.b.c.d***。同样，这仅适用于具有DNS记录以支持反向查找的IP地址。

```
$ host a.b.c.d
```

默认情况下，host实用程序使用的是你的设备被配置使用的DNS服务器。你也可以通过指定服务器的IP地址，用host来查询特定DNS服务器。互联网服务提供商为其用户提供了DNS服务，但也有许多其他免费的DNS服务可以使用。例如，在撰写本书时，Google在8.8.8.8提供了DNS服务器，Cloudflare在1.1.1.1提供了DNS服务器。如果你想使用1.1.1.1的DNS服务器查找www.example.com，可以输入下面的内容：

```
$ host www.example.com 1.1.1.1
```

这应该输出和前面一样的IP地址信息，同时还输出一些文本来说明本次查找使用的是哪个DNS服务器。

如果你对DNS查询的详细信息感兴趣，你可以在host命令中使用-v选项，它会提供详细输出：

```
$ host -v www.example.com
```

第12章

万维网

第11章描述了互联网，一组全球互连且共享一套协议的计算机网络。万维网（World Wide Web，WWW）是建立在互联网之上的一个系统，它非常受欢迎，以至于常常和互联网混淆。本章将深入探讨Web的细节。我们首先查看其关键属性和相关编程语言，然后再查看Web浏览器和Web服务器。

12.1 万维网概述

后面将We b译作“网络”。——译者注

万维网（通常简称为Web）是一组资源，使用超文本传输协议（HTTP）在互联网上传输。任何能用网络访问的东西都是网络资源，比如文档或图像。托管网络资源的计算机或软件程序被称为网络服务器（Web server），网络浏览器（Web browser）是一种通常被用于访问网络内容的应用程序。浏览器用于查看被称为网页（web page）的文档，一组相关的网页则被称为网站（website）。网络是分布式的、可寻址的，以及链接的。我们先来查看一下这些核心属性。

12.1.1 分布式网络

万维网是分布式的。没有集中的组织或系统来管理哪些内容能在网络上发布。任何连接到互联网的计算机都可以运行网络服务器，这种计算机的拥有者可以提供任何他们想要的内容。也就是说，有些组织或国家可能选择阻止用户访问网络上的某些内容，政府可以关闭含有非法内容的网站。除了这些情况之外，网络是一个开放的平台，人们可以发布任何想要发布的内容，没有一个组织来控制哪些内容可以发布。