第3章　互　联　网

本章将对互联网进行概述，并将讨论对理解安全问题很关键的几个方面。本书的后面章节还会更详细地讨论许多协议和隐含的重要安全问题。我们首先来给互联网下个定义。

互联网是一个通过网络协议[1~3]将不同设备互连在一起的设备集合。部分互联设备运行应用程序，并与用户通过接口进行交互;部分用于设备与网络的连接。图 3.1 所示是互连网的层次结构。图 3.1(a)给出的是用户眼中的互联网，一个典型的用户认为，互联网提供了让他把计算机插入，然后能和连接在互联网上的任何人进行交流的一个接入点。用户可以把互联网看成一个黑匣子。从安全的角度来讲，我们有时也可以把互联网看成攻击者出没的黑匣子，因为我们并不关心互联网是如何构建的(这是对互联网的最普通的认识)，而是关心攻击者对终端系统和网络的攻击。因为终端用户无法控制互联网，也没有能力降低对无法控制的互联网的攻击。而许多机构拥有复杂如部分互联网的网络，因此，了解互联网的组成和协议的使用将对减少对互联网的攻击有帮助。

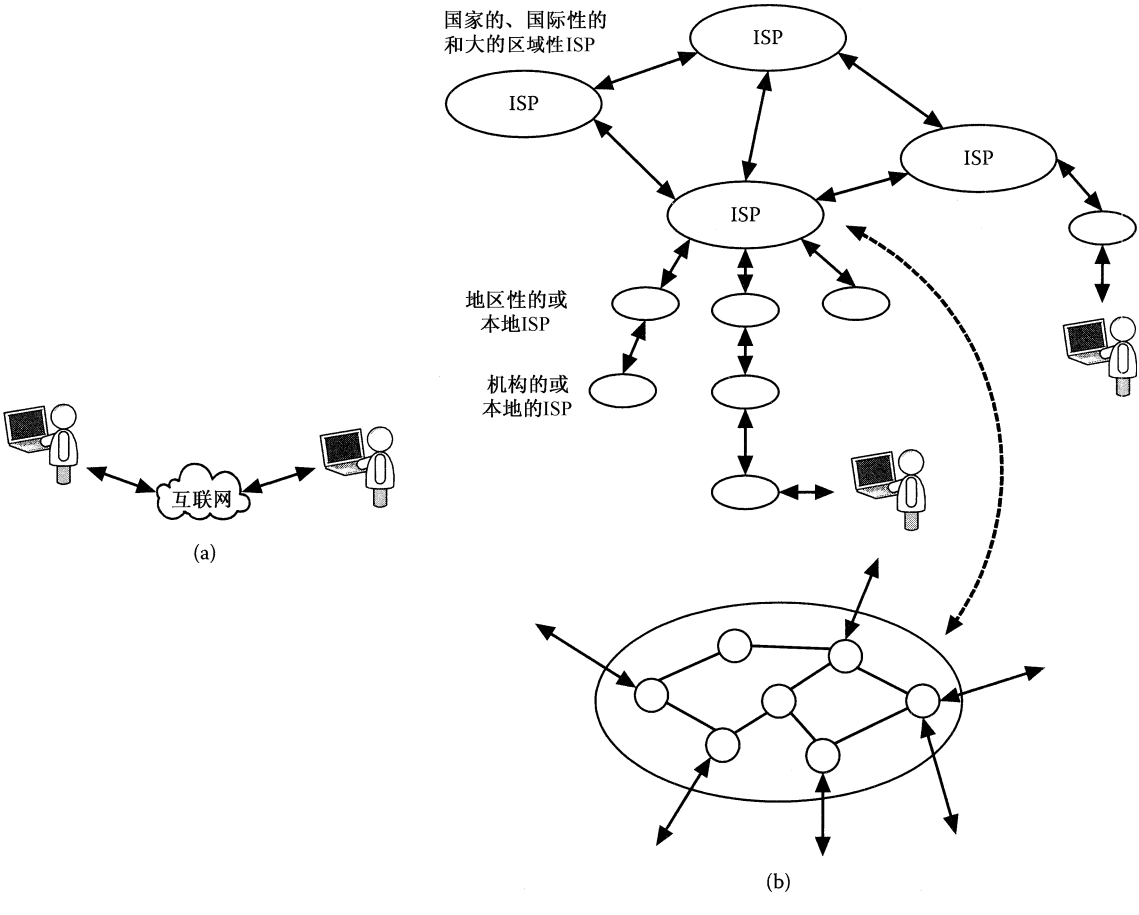

图 3.1　互联网示意图

图3.1(b)提供了互联网的另外一个视图，从图3.1(b)可以看出，互联网是由互连的互联网服务提供商(Internet Service Provider，ISP)组成的。这些ISP按照国家的、国际性的和大的区域性的ISP互连，从而形成我们称之为骨干的非正式的层次结构。这些ISP使用高速且精确的连接方法互连，并传输海量的流量。连接到骨干的是其他的ISP或大的机构，然后是较小的ISP和其他机构连接到中间的ISP，最后，终端用户和机构接入。我们从图3.1(b)还可以看出，一个ISP由一组互连的设备组成，这些设备和计算机系统及连接到互联网上的网络都会受到攻击。

从安全的角度来讲，连接到互联网上的每一台设备和协议都是脆弱的，是一个潜在的攻击源或攻击目标。因此，每一个设备或协议都要从安全的方面来评估。在详细地考察不同的协议之前，我们需要了解用在互联网中的几个关键概念，它们是安全的基本概念，那就是寻址和路由。它们是互联网的核心，是互联网中两个最关键的问题。首先我们讨论寻址，然后理解用在整个互联网中的客户-服务器模式，最后讨论互联网的路由。

3.1 寻址

在第2章中我们已经了解了网络中各层是如何使用地址识别设备、协议和应用的。互联网同样使用地址。重要的是要了解攻击者能够修改哪些地址，哪些地址是内网的，哪些地址是全球的。

如果总结一下互联网中使用的寻址，可以看出，在互联网寻址应用角度和互联网寻址低层角度之间，有一个逻辑分界，用户和应用把互联网看成相互获得数据的一种渠道。图3.2说明了从用户和应用层面看是如何寻址的。

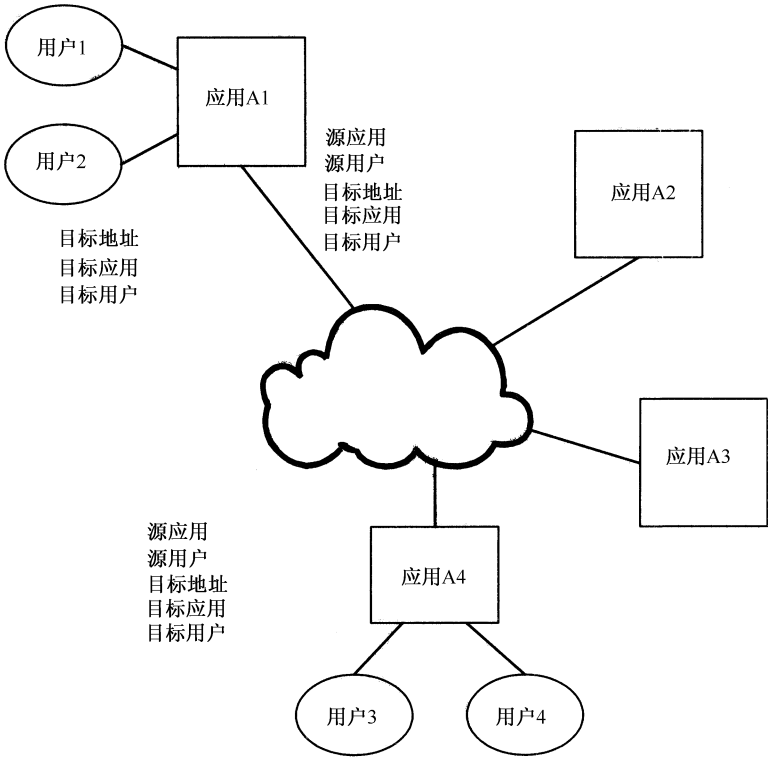

图3.2 应用寻址

由图 3.2 可以看出，从用户与应用角度看寻址，类似于一个人使用邮电系统的寻址。用户给应用提供目标计算机的地址，同时提供目标应用的地址，当然，这个动作通常是由应用设置的，用户还要提供目标用户和目标文件的地址。应用将提供地址信息，以便目标应用可以把数据发回给发送的应用。互联网的运作是，你把数据发出去，它就会到达正确的地点，用户和应用都不需要关心数据是如何到达的，途中什么样的设备来处理数据。这还是类似于邮电系统，发信的人不必关心信件是如何到达目的地的。

由图 3.2 扩展到图 3.3，图 3.3 显示了两台连接到互联网的机器，并显示了当数据传输时[4]需要的低层协议。图 3.3 也说明了如何利用不同的层地址将数据从一台设备传输到另一台设备。这里忽略在互联网上流量是如何被路由的，以及地址发现是如何处理的。计算机 C1 的用户 A 想向计算机 D1 上的用户 B 发送一条“Hello”信息。由图 3.3 可以看出，用户 A 向计算机 C1 的应用 A1 发送信息“Hello”，可以确信发送方应用不需要给目标应用提供地址，例如 URL 或邮件用户名。如果需要，那么发送方用户也需要给发送方的应用提供指定地址。用户或在某些情况下的应用给目标计算机 D1 提供地址，发送方应用使用应用端口号识别远程应用，传输控制协议(TCP)层使用应用端口号识别哪个送来的数据包和这个应用有联系。计算机 C1 上的应用需要知道计算机 D1 上的应用的端口号。随后我们可以看到，这个端口号也可以由用户提供，也可以这么说，应用同意把它们看成配置的一部分。简言之，应用和用户给 TCP 层提供了目标端口号、目标 IP 地址和用户数据(载荷)。

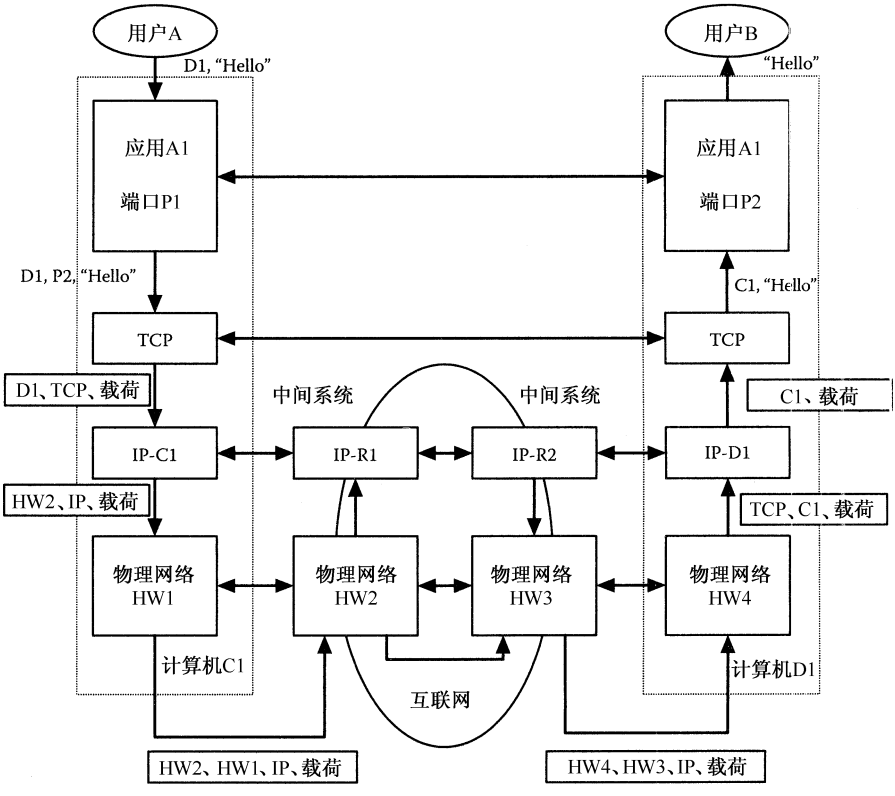

图 3.3 互联网寻址示意图

TCP 层将其包括的源端口号、目标应用和用户数据，以及 TCP 控制信息一并作为载荷的部分数据包发送到 IP 层。由应用作为识别 TCP 的地址(传输协议类型)的协议、计算机 C1 上的

源 IP 地址及计算机 D1 上的目标 IP 地址[5,6]一起追加到 IP 层头部。目标 IP 地址(由用户提供)由 TCP 层提供，源 IP 地址由 IP 层得到。IP 层将包含下一个设备的目标硬件地址及 IP 层的网络协议 ID 的数据包传输到物理网络层，物理网络层再追加它的源地址。

在网络上发送到下一台设备的数据包包含几个地址，如表 3.1 所示。这个表也说明了是谁提供了地址。

表 3.1 互联网地址

地 址		用 户	应 用	TCP	IP	网络
用户或文件	SRC	X	X			
	DST	X				
计算机地址	SRC				X	
	DST	X				
应用 ID(端口)号	SRC			X		
	DST	X	X			
传输协议					X	
IP 地址	SRC				X	
	DST	X	X			
网络层协议 ID					X	
硬件地址	SRC					X
	DST				X	

在这个例子中数据包被递交的下一台设备是路由器，路由器不关心传输层与应用层，路由器接收到数据包是因为目标硬件地址与路由器的硬件地址匹配。路由器的物理网络层将检查网络层协议 ID，看看它是什么类型的数据包。如果它是一个 IP 数据包，那么它会剥去物理网络层头部，把数据包的剩下部分传给路由器的 IP 层。路由器的 IP 层检查源和目标 IP 地址，决定将数据包发送到哪里。路由器然后把数据包传送到物理网络层，并附加一个新的源和目标硬件地址，再给 IP 数据包设置一个网络层协议 ID 标志。注意，像路由器这样的设备往往有多个物理网络接口，每个接口有它自己的物理网络协议层，数据包将继续从一个路由器传送到下一个路由器，直到数据包到达目标计算机 D1 为止。当数据包到达计算机 D1 时，源硬件地址应与最后一个路由器的地址匹配。目标硬件地址应匹配计算机 D1 的硬件地址。

当计算机 D1 收到数据包时，要检查数据包的网络层协议 ID，并将数据包传送到 IP 层，IP 层再检查目标 IP 地址，看其是否匹配计算机 D1 的 IP 地址，如果匹配，那么 IP 层将检查传输协议 ID 以决定这个数据包是否传输到 TCP 层，或传输到不同的传输协议层。TCP 层将检查目标应用端口号，以决定哪个应用应该得到数据包。最后，“Hello”将传送到在计算机 D1 上运行的应用 A1。

如果计算机 D1 上的应用希望回送一个数据包到计算机 C1 上的应用，那么它可以将其收到的数据包的源应用端口号作为计算机 C1 上的应用的识别标志，将其收到的数据包的源 IP 地址作为计算机 C1 上的识别标志。如图 3.2 所示，以与计算机 C1 发送的数据包相同的方式形成一个数据包。可以看到，两台计算机上的两个应用之间的数据包的交换有 4 个地址唯一地识别流动在每一个方向上的数据包。每两个 IP 地址和应用端口号产生一个全球唯一的识别标志，用于区分不同的数据包流。这说明了如何在同一个 Web 站点同时打开两个浏览器窗口。因为每一个浏览器窗口有一个不同的网络连接，因此它们具有不同的唯一的识别标志。

3.1.1 地址欺骗

由图 3.2 和图 3.3 我们可以看到，寻址用于互联网上正确的协议层和设备之间的直接通信。在许多情况下，地址用于标明数据从哪来和来自谁。我们常常使用这些不同的地址作为一种识别数据的发送者和接收者的方式，正如我们使用邮电地址确保一封信件到达正确的收信人，返回地址告诉我们信件来自哪里一样。与邮电系统相同，没有什么有效的方法识别这些地址的真实性。你可以在一封信上写上你想要的任何返回地址，把它放到邮箱中，这封信仍然可以到达。写一个虚假的目标地址是不起作用的，因为信件是按照信封上的目标地址送出的。在互联网上也会发生同样的事情，采用虚假的源地址的行为称为地址欺骗[7, 8]。图 3.4 给出了一个地址欺骗的例子。

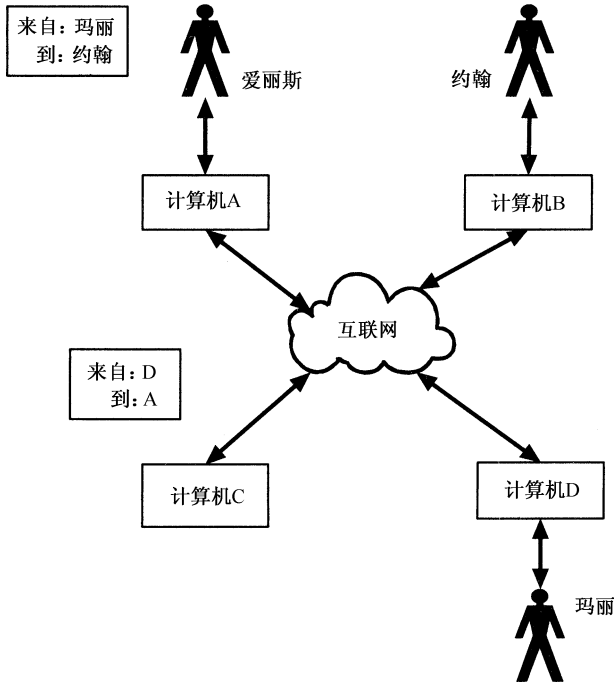

图 3.4 地址欺骗

由图 3.4 可以看出，第一个地址欺骗的例子是计算机 C 到计算机 A 的数据包发送，源地址是计算机 D，所以如果计算机 C 收到了数据包，那么它会认为数据包来自计算机 D，这样就会导致潜在的问题。如果计算机 A 信任计算机 D，就会把这个数据包看成可信任的。图 3.4 中的第二个例子是用户爱丽斯发送一条信息给用户约翰，爱丽斯把发送地址设置为用户玛丽，这样，当约翰收到信息时，他会以为这条信息来自玛丽。我们在后面章节会看到，在某些情况下，进行地址欺骗很容易；而在另外一些情况下，进行地址欺骗很困难，甚至是不可能的。

3.1.2 IP 地址

IP 地址被设计为在互联网上是全球唯一的。IP 地址由两个部分组成：网络和主机。因此，一种描述互联网的方法是把互联网看成网络的集合，而每个网络都由地址和主机组成。在 IP 协议的第 4 版中，地址空间长 32 位，用 4 个由小数点分开的数字书写而成，这样很容易使用数字，并很容易理解路由和分类。每一个数字代表 32 个比特位中的 8 个比特位。

当刚开始使用 IP 协议时，整个网络上的计算机设备数量很少。地址空间分配方法是先来先分配，地址的网络部分则按照申请机构来分配，然后再分配地址空间的主机部分。一个机构也可以把它的地址空间再划分成若干个小的网络。为了路由，产生了网络掩码，并将其作为区分网络地址和主机地址的一种方法。网络掩码的规定类似于 IP 地址，也有 4 个由小数点隔开的数字，当将其转换成 32 位的二进制值时，比特位全 1 的代表网络地址部分，例如，255.0.0.0 的前 8 个比特位全为 1，因此经过掩码运算为 A 类网络。图 3.5 所示为典型的互联网中的网络，同时给出了其网络地址与掩码。

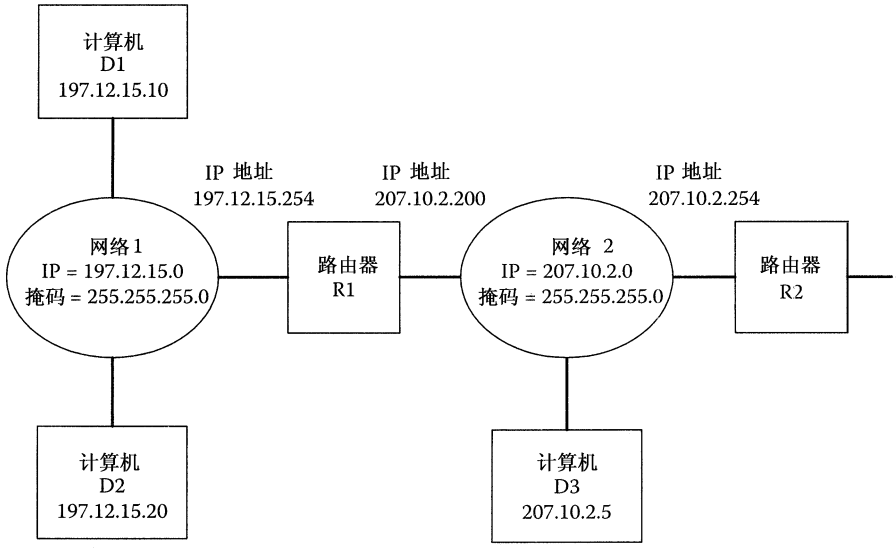

图 3.5 互联网中的网络

3.1.3 主机名与 IP 地址的匹配

大多数用户并不使用 IP 地址指定他们要连接的服务器和应用，而是使用主机名和域名。例如，当一个用户发送一封电子邮件，他或她会使用域名作为目标地址(例如，admin@ vulcan. dougj. net)。当电子邮件应用系统发送一封电子邮件到网络上时，IP 数据包头部包含有 32 位格式的目标 IP 地址。域名和 IP 地址之间的转换是通过称为域名服务(DNS)的分布式应用来完成的。这个过程使用本地 DNS 和分布式 DNS 服务器进行通信，并进行主机全名(主机名 + 域名)和它的 IP 地址之间的转换。如果我们考察一个典型的互联网上的设备名(如 Web 服务器)，就会发现，名字由几个部分组成。例如，vulcan. dougj. net 是一个主机的全名，计算机的名字是 vulcan，域名是 dougj. net，图 3.6 显示了该 DNS 模型[9]。

如图 3.6 所示，用户想给 admin@ vulcan. dougj. net 发送一条电子邮件信息，电子邮件系统先查询本地 DNS 应用，本地应用接着查询下一个 DNS 服务器。DNS 系统被部署成具备一组根 DNS 服务器的树形结构，并且它知道所有一级域名服务器的位置。一级域名服务器有它所在域内的每个主机的 IP 地址信息，并知道和它所在域内哪个 DNS 服务器通信。这种层次方法允许由一个 DNS 服务器分配基于名字与 IP 地址匹配的管理控制信息。在本书稍后，我们将讨论 DNS 协议的安全问题，现在只需要知道一台设备何时想知道已经知道名字的主机的 IP 地址，这台设备再请求它的 DNS 服务器，随后将知道答案。答案也许在缓冲区中，它也可以请求根服务器，并在根服务器中找到答案。如图 3.6 所示，请求(虚线表示)通过根服务器传播到知道答案的 DNS 服务器，然后响应再传输回去。

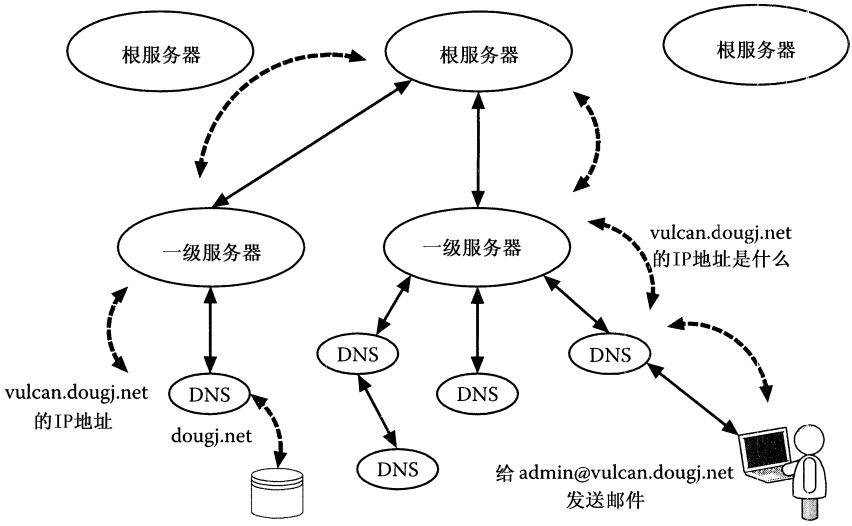

图 3.6 DNS 模型

定 义

地址欺骗

把数据包的源地址改变成不属于正在发送数据包的那台设备的值。

域名

一个机构的名字，它有一个或一个以上的网络，并有一台或一台以上的设备。域名在互联网上必须是唯一的。

域名服务（Domain Name Service，DNS）

一个分布式服务器的集合，负责把全域名转换为 IP 地址。

全域名

主机名与域名的结合，用于生成互联网中唯一的设备识别标志。

主机名

某个域内的主机的名字，主机名在指定域内必须是唯一的。

IP 地址

用于唯一地识别互联网中每台设备的地址。

网络掩码

一个 32 位的值，用于指定 IP 地址的哪一部分表示网络，哪一部分表示主机。

网络层 ID

一种识别标志，存储在物理网络层的头部，指出哪个上一层协议包含在载荷中。

子网

当一定范围的 IP 地址被划分成多个网络且要使用路由器时出现的情况。

3.2 客户-服务器模式

这是一个在互联网中十分流行的概念，指客户应用与服务器应用的交互，称为客户-服务器模式[1, 10]。客户-服务器模式是个定义而不是标准。在互联网中，一个服务器程序定义为一

个应用等待另一个应用请求连接。服务器程序常常等待默认客户端口号请求连接[1,5,6]。图3.7示意了几个客户应用通过互联网来请求与服务器程序的连接。

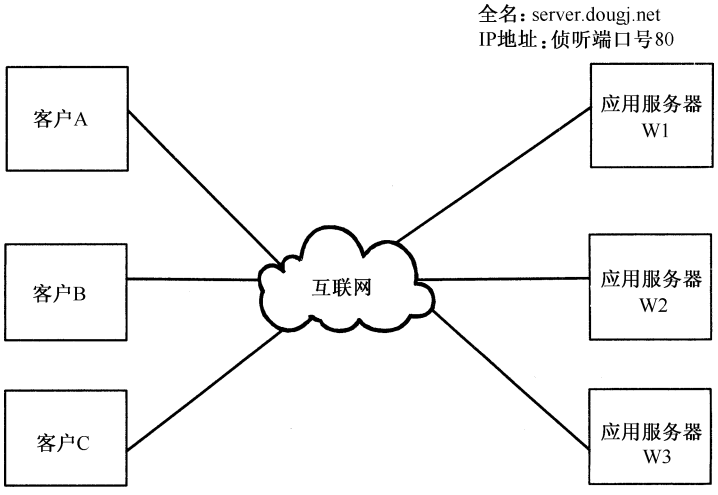

图3.7 客户-服务器模式

如图3.7所示，服务器应用驻留在具有全名及IP地址的计算机上，这个应用还被分配了应用地址(侦听端口号)。图3.7中的3个应用都在等待端口80(Web服务器使用相同的端口号)。客户应用通过指定目标地址和目标端口号请求与一个等待的服务器程序进行连接。客户也可以使用DNS系统把服务器的全域名转换为服务器程序的IP地址。

请求一个连接，就是一个服务器程序请求操作系统打开一个与TCP层的连接(一个套接字)，并侦听目标为某个端口号的连接(侦听端口号)。图3.8所示是两个客户与两个服务器程序及它们开始通信的过程。同一台主机上的每个服务器程序侦听一个不同的端口号，在建立连接时，客户必须指定目标端口号及目标IP地址。套接字是应用与操作系统之间的连接的规定名称，它是由侦听IP地址与端口号定义的。一个应用只可以侦听一个与给定的目标IP地址有联系的给定端口。如果有多个IP地址与计算机联系，应用需要指示正在侦听的目标IP地址。

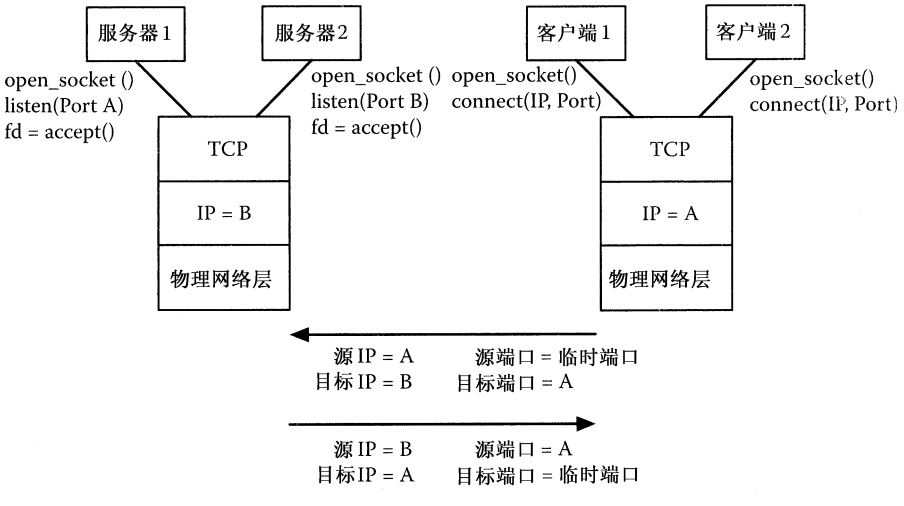

图3.8 客户-服务器模式的连接

由图3.8可以看出，服务器程序1和服务器程序2都各自打开一个套接字，并告诉TCP层它们请求侦听的是哪个端口，然后等待一个客户的连接。当客户请求与服务器程序连接时，接受回应是给客户应用返回一个打开的连接，服务器程序就使用这个打开的连接，发送和接收客户应用的数据。

为了开始一个连接，客户也需要与操作系统进行交互。客户将打开一个套接字，同时客户既可以使用源端口号，也可以让操作系统使用源端口号。当操作系统对客户使用源端口号时，称这个端口号为临时性端口。正像服务器应用一样，一个客户应用只能在某一时间使用一个给定的源端口号。客户应用指定它要连接的应用的目标IP地址和目标端口号。

客户应用通过发送第一个带目标IP地址的数据包，请求连接，这个IP地址要与服务器主机匹配，并且应用程序端口号要匹配服务器应用。表3.2列出了由客户到服务器及返回时的数据包的IP地址和端口号。这两种类型的数据包(客户到服务器和服务器到客户)的4个值(IP地址和端口号)在互联网上应该是全球唯一的。

表3.2 数据包寻址

客户到服务器的数据包	
源IP	客户IP地址
目标IP	IP服务器的IP地址
源端口	临时端口号
目标端口	熟知的服务器端口号
服务器到客户的数据包	
源IP	服务器IP地址
目标IP	客户机的IP地址
源端口	熟知的服务器端口号
目标端口	临时端口号

如果一个数据包到达目标，且没有应用在等待，那么数据包被拒绝。数据包是如何被拒绝的在本章后面讨论。

这时就有问题产生了，如果一个应用只能打开一个给定的端口，那么一个服务器应用，例如Web服务，如何支持多个连接请求(即使来自相同的客户)呢？要理解这个问题，我们需要首先考察服务器程序是如何处理多个连接请求的。当一个连接到达时，操作系统将返回一个新的连接给服务器程序，以用于服务器与客户的通信。在图3.8中该过程由accept函数体现，它返回一个新的连接识别标志，服务器程序这时产生一个新的进程来处理与那个客户的连接，父服务程器序将等待下一个来自客户的连接请求。

为了弄明白同一个服务器程序如何处理来自一个主机的同一个客户的多个连接请求，需要考察客户应用是如何处理多个连接的。在一个指定的主机上客户的每一个连接请求将由操作系统分发一个不同的临时端口号，这样，两个数据包将有不同的临时端口号，一个具有两个连接的客户应用就可以打开同一个服务器程序。一个很好的例子就是在同一个Web服务器上同一个Web浏览器可以同时打开两个窗口。如前所述，每一个客户和服务器连接都是唯一的，这是由IP地址和端口号构成的4元组进行区分的。图3.9给出一个的几个客户向两个Web服务器请求连接的例子，它的端口号是默认的80。

在图3.9中有5个连接，如表3.3所示,每个连接由不同的4元组组成。注意,目标服务器的每个数据包的4元组是如何不同的，每个返回数据包就将是如何不同的。还应注意，由客户B占用的临时端口号与客户A占用的临时端口号可以是相同的，因为源IP地址是不同的。

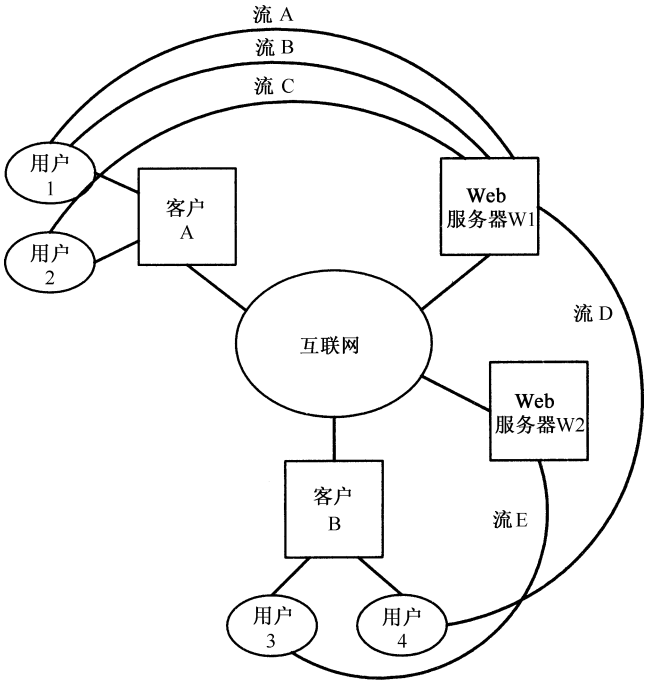

图 3.9 多客户–服务器模式

在本书的后面还可以看到，这个4元组可以由网络安全设备占用以帮助跟踪连接，且可以帮助过滤动态连接的流量。

表 3.3 流地址

流	源 IP	目标 IP	源端口	目标端口号
A	A	W1	临时端口 A1	80
B	A	W1	临时端口 A2	80
C	A	W1	临时端口 A3	80
D	B	W1	临时端口 B1	80
E	B	W2	临时端口 B2	80

定 义

客户

请求与一个等待的服务器连接的一种应用程序。

连接 4 元组

4元组用于在互联网上唯一识别每一个连接，它由源 IP 地址和目标 IP 地址、源端口号和目标端口号组成。

临时端口号

一般由操作系统给一个客户应用提供的端口号，并作为客户应用的源端口号。

侦听端口号

由一个服务器程序占用的端口号，用来等待一个客户请求连接。

服务器程序

一个等待客户应用与它建立连接的应用程序，服务器程序一般是为客户提供服务的。

套接字

处于应用层与 TCP 层之间的一种连接，它允许一个服务程序指定 IP 地址和端口号并等待，且允许一个客户应用指定目标 IP 地址和端口号。

默认端口号

与侦听端口号一样，但这个端口号是服务器程序的默认端口号，且是所有想与服务器程序交互的客户应用都知道的，如端口号 80 是 Web 流量的默认端口号。

3.3 路由

互联网的一个主要功能就是它能跨过多个网络把数据包从源路由到目标，这些网络分别由不同的机构控制。一直以来有无数的关于路由及如何高效路由[11]的研究项目和论文。本书仅把路由看作为由一系列互联的称之为路由的设备提供的简单功能。我们假定路由器有办法确定把数据包送往哪里，以便把数据包送到目标地址。由路由器确定的数据包的路由协议也有可能遭到攻击，我们将在本章后面讨论这些内容。在我们讨论互联网的路由之前，有必要先回顾一下早期的网络。

第一个网络基于与电话系统相同的概念，即任何流量通过之前，源和目标之间的路由是确定的，并且所有的流量路径是相同的。这种面向连接的网络使得数据的发送和接收变得容易，因为数据是按序到达的。这种类型网络的复杂性在于为了建立路由而面对全球所有设备的视图的需求，中间设备不需要知道网络的任何事情，只需要对全球网络管理系统给出的命令做出响应即可。

互联网使用的是无连接方法，即每一个数据包由每一个路由器分别处理，数据包由源设备发送到能处理它的下一个设备，那台设备然后检查它的本地路由表，并决定数据包要发送的下一个地点。需要注意的是，一个连接到互联网上的计算机，也需要知道流量是如何路由的，因此也需要一个路由表。这些本地路由表可以是静态的或动态的[12]。静态路由表是指在设备被配置后且直到重新修改配置前不做变化的配置状态。大多数计算机网络中都采用静态路由表，这种网络通常只有一个路由器。动态路由表是由协议根据不同因素对其进行调整的路由表。动态路由表超出了本章的讨论范围。无论路由表是动态调整的还是静态的调整，其路由的工作方式是相同的。图 3.10 很好地说明了动态路由和静态路由各自的优点。

由图 3.10 可以看出，主机 H1 并不能从动态路由中获得什么好处，因为通向网络的只有一条路径，即路由器 R1，同样，R1 也只有一条路径，动态路由表也没有必要。但是我们考察一下图 3.10 中的其他路由器，例如要穿过网络获取一个数据包就有多条路径。在这种情形下，动态路由表就有意义了。

连接到网络上的每一台设备都有一个路由表，以指明将数据包发送到的下一个可能的目标地址。这下一跳是由 IP 地址和一个接口(比如路由器，可能有两个或两个以上的接口)指定的。乍一看，如果每个可能的目标地址都占有一个条目，那么将需要一个很大的路由表。查询路由表的最好方法应该是查看数据包要到达的目标地址。目标地址是由网络地址表示的，网络地址是由地址和网络掩码组成的。图 3.11 给出了一个具有几个设备的网络和路由表的例子。

由图 3.11 可以看出，连接到网络 1 的计算机到达目标地址有两个选择:一是连接到网络 1 的所有计算机和其他地方。它的路由表有两条，第一条是与网络 1 的任何计算机相匹配的目标地址，这台计算机可以不通过路由器直接将数据包发送到网络 1 上的任何计算机。第二个

选择是不在网络1上的任何计算机。这个选择意指当所有的目标都不匹配时，默认就是路由器，在图3.11的例子中是路由器R1。

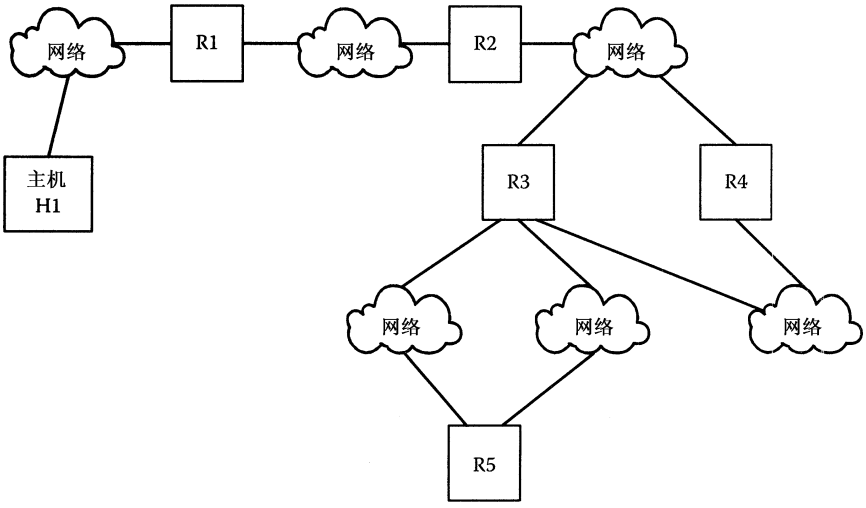

图3.10 动态与静态路由

目标	下一跳
网络1	直达
默认	路由器1

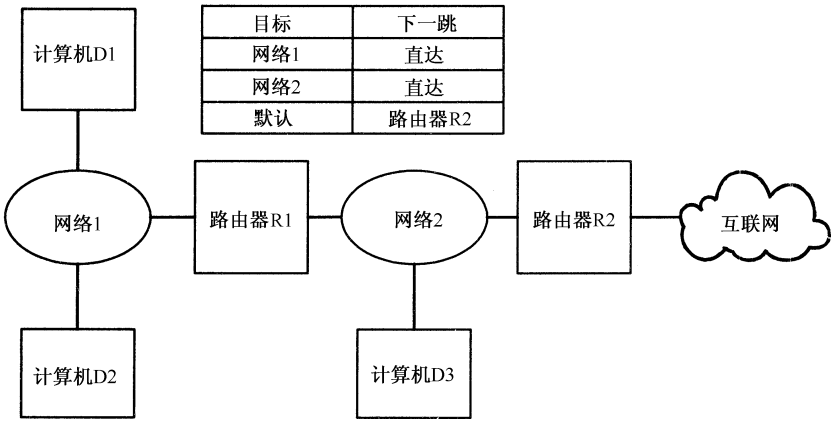

图3.11 路由示例

如果我们考察路由器R1，它可以有3个目标，网络1上的所有计算机，网络2上的所有计算机，或其他任何目标。因此，对应这3个选择的路由表也有3条。在图3.11所示的例子中，这个路由表已经被简化了。关于路由及其安全问题在本书第二部分会有更详细的讨论。

定 义

默认路由

当目标地址与路由表中的任何目标地址不匹配时所取的路径。

动态路由

一个路由表和路由表条目根据使用特殊的协议得到的额外信息而改变。

路由表

一台设备将一个数据包发往的可能的目标地址列表。目标一般是一台设备或一个路由器。

路由

把数据包从一台设备发送到通过众多路由器互连的网络的另一台设备的动作。

静态路由

一个路由表或路由表中的条目只有在系统配置或重新配置时才改变。

课后作业和实验作业

课后作业

1. 找一至两幅互联网拓扑图，对它们的准确性加以评论。
2. 在互联网上被指定的默认端口大约有多少个？
3. 这个端口数代表了互联网上的所有唯一的应用吗？
4. 如果一个客户应用使用了错误的端口号识别服务器应用会发生什么情况？
5. 如果一个服务器应用正等待一个非默认的端口号会发生什么情况？
6. 一个应用程序必须使用已经被分配的默认端口号吗？
7. 为什么服务器应用会使用一个非默认的端口号？
8. 对下列的每一条(指出哪个地址成分不用)，给出一个地址的每个组成部分(硬件、计算机[名字和 IP 地址]、应用和用户)，同时说出你如何确定你还不知道的地址组成部分的价值？

   a. E-mail 地址：admin@ dougj. net
   b. Web 地址：http://www. dougj. net
   c. Web 地址：http://129. 186. 215. 40
   d. FTP 地址：vulcan. dougj. net

9. 你能说出你可以欺骗硬件地址(改变硬件地址)的理由吗？
10. IPv4 总的地址数是多少？
11. 根 DNS 服务器的总数目是多少？
12. 两个应用之间的每一个数据包必须取相同的路径吗？给出解释。
13. 在互联网上采用无连接的方法进行路由有什么优点？

实验作业

1. 绘制一个跨互联网的至少有 5 个 Web 站点和 5 个 E-mail 服务器的列表。
2. 使用 DNS(一个名为 nslookup 或 dig 的程序)，查找实验 1 的每个站点的 IP 地址。对于 E-mail 服务器，你需要将 DNS 队列类型设置为 MX。看看运行程序的主页。
3. 使用同样的程序，查看具有像 Web 站点的 IP 地址的机器名(使用同样的 IP 地址的前 3 个 8 比特和后 1 个 8 比特)，一个攻击者是如何利用这个过程的？

4. 使用一个程序跟踪一个 UNIX 计算机或 Windows 计算机，找出你所在网络上的一台主机到实验 1 列出的服务器的路径。

   a. 使用返回的数据，绘制到这些站点的路径图。
   b. 你能确定这些站点所处的地理区域吗？
   c. 你所在的单位的网络中有多少个路由器？
   d. 你能确定你所在的单位的互联网服务提供商(ISP)的名字吗？

5. 使用 ping 程序，确定数据包到实验 1 中列出的服务器的往返所用的平均时间。

   a. 对从服务器到你的位置的广播时间加以评论。
   b. 对为何有的服务器对 ping 程序没有回答加以评论。

6. netstat-a 命令会给出你的计算机的所有连接，使用这个命令给出使用 4 元组识别的每一个客户–服务器连接。

参考文献

[1] Comer, D. E. 1995. *Internetworking with TCP/IP*. Vol. 1. *Principles, protocols and architecturel*. Englewood Cliffs, NJ : Prentice Hall.

[2] Calvert, K. I. , M. B. Doar, and E. W. Zegura. 1997. Modeling Internet topology. *IEEE Communications Magazine* 35 : 160-63.

[3] Subramanian, L. , et al. 2002. Characterizing the Internet hierarchy from multiple vantage points. In *INFOCOM 2002 : Proceedings of the TwentyFirst Annual Joint Conference of the IEEE Computer and Communications Societies*, 2. New York, NY.

[4] Kurose, J. F. , and K. W. Ross. 2003. *Computer networking : A top-down approach featuring the Internet*. Reading, MA : Addison-Wesley.

[5] Postel, J. 1981. *Assigned numbers*. RFC 790.

[6] Postel, J. 1981. *Internet protocol*. RFC 791.

[7] Heberlein, L. T. , and M. Bishop. 1996. Attack class : Address spoofing. In *Proceedings of the 19th National Information Systems Security Conference*, Baltimore, MD : 371-77.

[8] Bellovin, S. M. 1989. Security problems in the TCP/IP protocol suite. *ACM SIGCOMM Computer Communication Review* 19 : 32-48.

[9] Mockapetris, P. , and K. J. Dunlap. 1988. Development of the domain name system. *SIGCOMM Computer Communication Review* 18 : 123-33.

[10] Stevens, W. R. , and T. Narten. 1990. Unix network programming. *ACM SIGCOMM Computer Communication Review* 20 : 8-9.

[11] Huitema, C. 1995. *Routing in the Internet*. Upper Saddle River, NJ : PrenticeHall.

[12] Halabi, B. , S. Halabi, and D. McPherson. 2000. *Internet routing architectures*. Indianapolis, IN : Cisco Press.