第1章

数据包分析技术与网络基础

在计算机网络中，每天都可能发生成千上万的问题，从简单的间谍软件感染，到复杂的路由器配置错误。我们永远也不可能立即解决所有问题，而只能期盼充分地准备好相关的知识和工具，从而能够快速地响应各种类型的错误。

所有的网络问题都源于数据包层次，即使是有着最漂亮外表的应用程序，它们也可能是“金玉其外”但“败絮其中”，有着混乱的设计与糟糕的实现，又或是看起来是可信的，但背地里在搞些恶意的行为。为了更好地了解网络问题，我们需要进入到数据包层次。在这一层次，没有任何东西能够逃出我们的视线范围——这里不再有那些令人误解的菜单栏、用来吸引眼球的动画、以及无法让人信赖的员工。在数据包层次上，就不再有真正的秘密（加密通信除外），我们在数据包层次上做得越多，那我们就能够对网络有更好的控制，就能够更好更快地解决网络问题。这就是数据包分析的世界。

本书将与你一起进入到神奇的网络数据包世界里，你将学习如何解决网络通信速度慢的问题，识别出应用程序的性能瓶颈，甚至在真实世界的场景中追踪黑客。当你读完这本书后，你应该能够使用先进的数据包分析技术来解决自己网络中的实际问题，即便它们看起来是那么复杂与难以解决。

在这一章中，我们将开始学习一些网络通信方面的基础知识，这样你可以获得阅读和学习后续章节所需的基础知识。

1.1 数据包分析与数据包嗅探器

数据包分析，通常也被称为数据包嗅探或协议分析，指的是捕获和解析网络上在线传输数据的过程，通常目的是为了能更好地了解网络上正在发生的事情。数据包分析过程通常由数据包嗅探器来执行。而数据包嗅探器则是一种用来在网络媒介上捕获原始传输数据的工具。

数据包分析技术可以通过以下方法来达到目标。

- 了解网络特征。
- 查看网络上的通信主体。
- 确认谁或是哪些应用在占用网络带宽。
- 识别网络使用的高峰时间。
- 识别可能的攻击或恶意活动。
- 寻找不安全以及滥用网络资源的应用。

目前市面上有着多种类型的数据包嗅探器，包括免费的和商业的。每个软件的设计目标都会有一些差异。流行的数据包分析软件包括 tcpdump、OmniPeek 和 Wireshark（我们在这本书中将只使用此款软件）。tcpdump 是个命令行程序，而 Wireshark 和 OmniPeek 都拥有图形用户界面（GUI）。

1.1.1 评估数据包嗅探器

当你需要选择一款数据包嗅探器时，需要考虑的因素很多，包括以下内容。

支持的协议：数据包嗅探器对协议解析的支持范围各不相同，大部分通常都能解析常见的网络协议（如 IPv4 和 ICMP）、传输层协议（如 TCP 和 UDP），甚至一些应用层协议（如 DNS 和 HTTP）。然而，它们可能并不支持一些非传统

协议或新协议（如 IPv6、SMBv2、SIP 等）。在选择一款嗅探器时，需要确保它能够支持你所要用到的协议。

用户友好性：考虑数据包嗅探器的界面布局、安装的容易度，以及操作流程的通用性。你选择的嗅探器应该适合你的专业知识水平。如果你的数据包分析经验还很少的话，你可能需要避免选择那些命令行的嗅探器，比如 tcpdump。另一方面，如果你拥有丰富的经验，你可能会觉得这类命令行程序会更具有吸引力。在你逐步积累数据包分析经验时，你甚至会发现组合使用多种数据包嗅探器软件将更有助于适应特定的应用场景。

费用：关于数据包嗅探器最伟大的事情是有着很多能够与任何商业产品相媲美的免费工具。商业产品与其他替代品之间最显著的区别是它们的报告引擎，商业产品通常包括各种花哨的报告生成模块，而在免费软件中则通常缺乏，甚至没有该模块。

技术支持：即使你已经掌握了嗅探软件的基本用法，但是你还是偶尔会在遇到一些新问题时需要技术支持。在评估技术支持时，你可以寻找开发人员文档、公众论坛和邮件列表。虽然对于一些像 Wireshark 这样的免费软件可能缺乏一些开发人员文档，但使用这些应用软件的社区往往可以填补这些空白。使用者和贡献者社区会提供一些讨论区、维基、博客，来帮助你获得更多关于数据包嗅探器的使用方法。

操作系统支持：遗憾的是，并不是所有的数据包嗅探器都支持所有的操作系统平台。你需要选择一款嗅探器，能够支持所有你将要使用的操作系统。如果你是一位顾问，你可能需要在大多数操作系统平台上进行数据包捕获和分析，那么你就需要一款能够在大多数操作系统平台上运行的嗅探器。你还需要留意，你有时会在一台机器上捕获数据包，然后在另一台机器上分析它们。操作系统之间的差异，可能会迫使你在不同的设备上使用不同的嗅探器软件。

1.1.2 数据包嗅探器工作原理

数据包嗅探过程中涉及到软件和硬件之间的协作。这个过程可以分为成 3 个步骤。

第一步：**收集**，数据包嗅探器从网络线缆上收集原始二进制数据。通常情况下，通过将选定的网卡设置成混杂模式来完成抓包。在这种模式下，网卡将抓取一个网段上所有的网络通信流量，而不仅是发往它的数据包。

第二步：**转换**，将捕获的二进制数据转换成可读形式。高级的命令行数据包嗅探器就支持到这一步骤。到这步，网络上的数据包将以一种非常基础的解析方式进行显示，而将大部分的分析工作留给最终用户。

第三步，也是最后一步：**分析**，对捕获和转换后的数据进行真正的深入分析。数据包嗅探器以捕获的网络数据作为输入，识别和验证它们的协议，然后开始分析每个协议的特定属性。

1.2 网络通信原理

为了充分理解数据包分析技术，你必须准确掌握计算机是如何通过网络进行相互通信的。在本节中，我们将研究网络协议、开放系统互连模型（OSI 模型）、网络数据帧的基础知识，以及支持网络通信的硬件知识。

1.2.1 协议

现代网络是由多种运行在不同平台上的异构系统组成的。为了使它们之间能够相互通信，我们使用了一套共同的网络语言，并称之为协议。常见的网络协议包括传输控制协议（TCP）、互联网协议（IP）、地址解析协议（ARP）和动态主机配置协议（DHCP）。协议栈是一组协同工作的网络协议的逻辑组合。

理解网络协议的最佳途径之一是将它们想象成人类口头或书面语言的使用规则。每一种语言都有规则，如动词应该如何结合，人们该如何问候，甚至该如何礼貌地致谢。网络协议大多也是以同样方式进行工作的。它帮助我们定义如何路由数据包，如何发起一个连接，以及如何确认收到的数据。一个网络协议可以非常简单，也可能非常复杂，这取决于它的功能。尽管各种协议往往有着巨大的差异，但它们通常用来解决以下问题。

发起连接：是由客户端还是服务器发起连接？在真正通信之前必须要交换哪些信息？

协商连接参数：通信需要进行协议加密吗？加密密钥如何在通信双方之间进行传输？

数据格式：通信数据在数据包中如何排列？数据到达接收设备时以什么样的顺序进行处理？

错误检测与校正：当数据包花了太长的时间才到达目的地时如何处理？当

客户端暂时无法和服务器建立通信时，该如何恢复连接？

连接终止：一台主机如何告知另一台主机通信已经结束？为了礼貌地终止通信，应该传送什么样的最终信息？

1.2.2 七层 OSI 参考模型

网络协议是基于它们在行业标准 OSI 参考模型中的职能进行分层的。OSI 模型将网络通信过程分为 7 个不同层次，如图 1-1 所示。这个分层模型使得我们更容易理解网络通信。

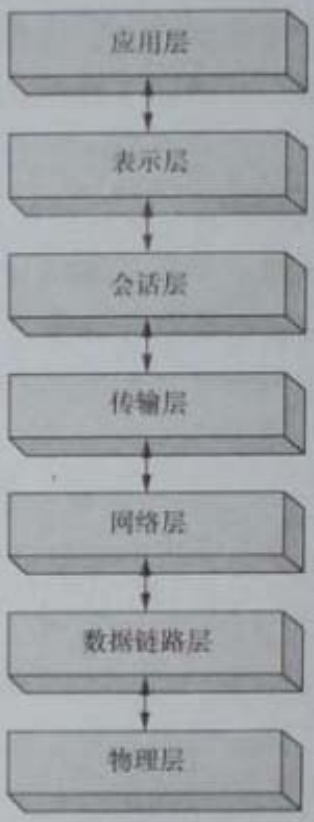

图 1-1 OSI 参考模型的七层协议视图

顶端的应用层表示用来访问网络资源的实际程序。底层则是物理层，通过它来进行实际的网络数据传播。每一层次上的网络协议共同合作，来确保通信数据在协议上层或下层中得到妥善处理。

注意 OSI 参考模型最初是在 1983 年由国际标准化组织出版，标准号为 ISO 7498。OSI 参考模型只是一个行业建议标准，协议开发并不需要严格地遵守它。OSI 参考模型也并不是现有唯一的网络模型，例如，有些人更推崇美国国防部（DoD）的网络模型，也被称为 TCP/IP 模型。

OSI 参考模型中的每层都具有特定功能，具体如下。

应用层（第 7 层）：OSI 参考模型的最上层，为用户访问网络资源提供一种手段。这通常是唯一一层能够由最终用户看到的协议，因为它提供的接口是最终用户所有网络活动的基础。

表示层（第 6 层）：这一层将接收到的数据转换成应用层可以读取的格式。在表示层完成的数据编码与解码取决于发送与接收数据的应用层协议。表示层同时进行用来保护数据的多种加密与解密操作。

会话层（第 5 层）：这一层管理两台计算机之间的对话（会话），负责在所有通信设备之间建立、管理和终止会话连接。会话层还负责以全双工或者半双工的方式来创建会话连接，在通信主机间礼貌地关闭连接，而不是粗暴地直接丢弃。

传输层（第 4 层）：传输层的主要目的是为较低层提供可靠的数据传输服务。通过流量控制、分段 / 重组、差错控制等机制，传输层确保网络数据端到端的无差错传输。因为确保可靠的数据传输极为烦琐，因此 OSI 参考模型将其作为完整的一层。传输层同时提供了面向连接和无连接的网络协议。某些防火墙和代理服务器也工作在这一层。

网络层（第 3 层）：这一层负责数据在物理网络中的路由转发，是最复杂的 OSI 层之一。它除了负责网络主机的逻辑寻址（例如通过一个 IP 地址）外，还处理数据包分片和一些情况下的错误检测。路由器工作在这一层上。

数据链路层（第 2 层）：这一层提供了通过物理网络传输数据的方法，其主要目的是提供一个寻址方案，可用于确定物理设备（例如 MAC 地址）。网桥和交换机是工作在数据链路层的物理设备。

物理层（第 1 层）：OSI 参考模型的底层是传输网络数据的物理媒介。这一层定义了所有使用的网络硬件设备的物理和电气特性，包括电压、集线器、网络适配器、中继器和线缆规范等。物理层建立和终止连接，并提供一种共享通信资源的方法，将数字信号转换成模拟信号传输，并反过来将接收的模拟信号转换回数字信号。

表 1-1 列出了 OSI 参考模型各个层次上的一些常见网络协议。

虽然 OSI 参考模型仅仅是一个建议标准，你还是应该将其牢记在心。当我们阅读本书时，你发现，对不同层网络协议进行交互才能解决你面对的网络问题。例如遇到路由器问题，你该很快确认这是个“第 3 层上的问题”，而应用软

件问题则被识别为“第 7 层上的问题”。

表 1-1　OSI 参考模型各个层次上的典型网络协议

层次	协议
应用层	HTTP、SMTP、FTP、Telnet
表示层	ASCII、MPEG、JPEG、MIDI
会话层	NetBIOS、SAP、SDP、NWLink
传输层	TCP、UDP、SPX
网络层	IP、IPX
数据链路层	Ethernet、Token Ring、FDDI、AppleTalk

注意　在讨论我们的工作时，一位同事告诉我，他曾处理过一位用户的投诉，说他不能访问网络资源，而实际原因是用户输入的密码不正确。我的同事将这个案例标成了“第 8 层的问题”，第 8 层是对用户层的一种非官方说法，通常是由那些整天工作在数据包层次上的网络工程师们使用。

那网络数据是如何流经 OSI 参考模型的各个层次呢？在网络上传输的初始数据首先在传输网络的应用层开始，沿着 OSI 参考模型的七层逐层向下，直到物理层。在物理层上，传输系统将数据发送到接收系统。接收系统从它的物理层获取传输数据，然后向上逐层处理，直到最高的应用层。

在 OSI 参考模型任意层次上由不同协议提供的服务并不是多余的。例如，如果某层上的一个网络协议提供了一种服务，那么再没有任何其他层的协议将提供与之完全相同的服务。不同层次的协议可能有目标类似的功能，但它们会以不同的方式来实现。

发送和接收计算机相同层上的网络协议是相互配合的。例如，发送系统在第 7 层的某个协议是负责对传输数据进行加密的，那么往往在接收系统的第 7 层有着相应的网络协议，负责对网络数据进行解密。

图 1-2 中连接了两个通信端图形化地说明了 OSI 参考模型。你可以看到，通信数据会从一个通信端的顶部流向底部，然后当它到达另一通信端时，将反向从底部流向顶部。

OSI 参考模型中的每一层只能和直接的上层与下层进行通信。例如，第 2 层只能从第 1 层与第 3 层发送和接收数据。

图 1-2　处于发送系统和接收系统同一层的协议

1.2.3　数据封装

OSI 参考模型不同层次上的协议在数据封装的帮助下进行通信传输。协议栈中的每层协议都负责在传输数据上增加一个协议头部或尾部，其中包含了使协议栈之间能够进行通信的额外信息。例如，当传输层从会话层接收数据时，它会在将数据传递到下一层之前，增加自己的头部信息数据。

数据封装过程将创建一个协议数据单元（PDU），其中包括正在发送的网络数据，以及所有增加的头部与尾部协议信息。随着网络数据沿着 OSI 参考模型向下流动，PDU 逐渐变化和增长，各层协议均将其头部或尾部信息添加进去，直到物理层时达到其最终形式，并发送给目标计算机。接收计算机收到 PDU 后，沿着 OSI 参考模型往上处理，逐层剥去协议头部和尾部，当 PDU 到达 OSI 参考模型的最上层时，将只剩下原始传输数据。

注意　　数据包这个术语指的是一个完整的 PDU，包括 OSI 参考模型所有层次协议的头部与尾部信息。

理解数据封装过程可能会有点困难，让我们看一个实际的例子，看数据包是如何在 OSI 参考模型中被创建、传输和接收的。作为数据包分析师，你需要

了解，我们经常会忽略掉会话层和表示层，所以它们将不会在这个例子中出现（包括本书的其余部分）。[1]

假设这样一个情形：我们试着在计算机上浏览 *http://www.google.com/*。在这个过程中，我们首先必须产生一个请求数据包，从客户端计算机传输到目标服务器上。这里我们假设 TCP/IP 通信会话已经被建立，图 1-3 所示为此案例中的数据封装处理过程。

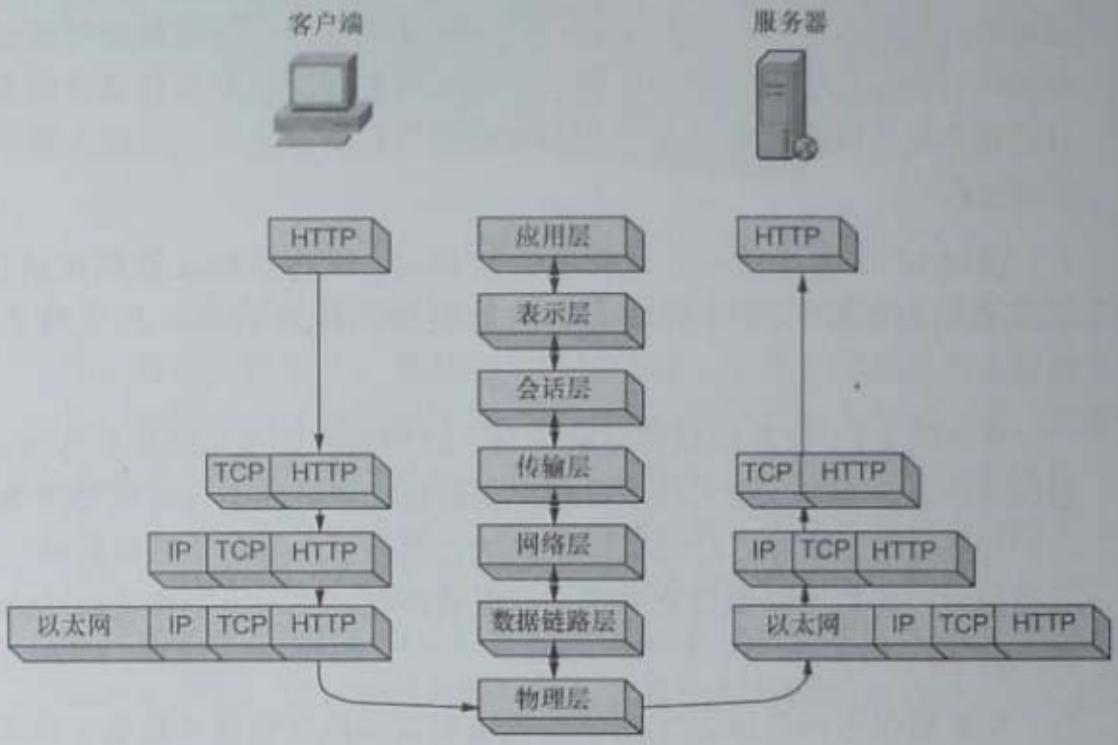

图 1-3　客户端和服务器之间数据封装过程示意图

我们从客户端计算机的应用层开始，在我们浏览一个网站时，所使用的应用层协议是 HTTP，通过此协议发出请求命令，从 google.com 下载 index.html 文件。一旦我们的应用层协议已经确定我们要完成的任务，我们现在关心的是数据包如何发送到目的地。数据包中封装的应用层数据将沿着协议栈传递给传输层。HTTP 是一个使用 TCP（或在 TCP 协议之上）的应用层协议，因此传输层中将使用 TCP 协议来确保数据包的可靠投递。所以一个包括序列号和其他数据的 TCP 协议头部将被创建，并被添加到数据包中，以确保数据包能够被正确交付。

注意　我们常说一个协议在其他协议之上，是因为 OSI 参考模型的上下层设计。例如 HTTP 等应用层协议提供了一个特定的服务，并依靠 TCP 协议来保证服务的可靠交付。正如你学习到的，DNS 协议构架于 UDP 上，而 TCP 构架在 IP 之上。

---

[1] 译者注：TCP/IP 模型中并没有会话层和表示层，因此实际的 TCP/IP 协议栈中并没有单独设计会话层和表示层网络协议。

在完成这项工作之后，TCP 协议将数据包交给 IP 协议，也就是在第 3 层上负责为数据包进行逻辑寻址的协议。IP 协议创建一个包含有逻辑寻址信息的头部，并将数据包传递给数据链路层上的以太网，然后以太网物理地址会被添加并存储在以太网帧头中。现在数据包已经完全封装好，然后传递给物理层，在这里数据包变成 0、1 信号通过网络完成传输。

封装好的数据包将穿越网络线缆，最终到达 Google 的 Web 服务器。Web 服务器开始读取数据包，从下往上，这意味着它首先读取数据链路层，从中提取到所包含的物理以太网寻址信息，确定数据包是否是发往这台服务器的。一旦处理完这些信息，第 2 层头部与尾部的信息将被剥除，并进入第 3 层的信息处理过程中。

读取 IP 寻址信息的方式和第 2 层相同，目的是确认数据包被正确转发，以及数据包未进行分片处理。这些数据也同样被剥离，并交到下一层进行处理。

现在第 4 层 TCP 协议信息被读取，以确保数据包是按序到达的。然后第 4 层报头信息被剥离，留下的只有应用层数据。这些数据会被传递到 Web 服务器应用程序。为了响应客户端发过来的这个数据包，服务器应该发回一个 TCP 确认数据包，使客户端知道它的请求已经被接收，并可以等待获取 index.html 文件内容了。

所有数据包都会以这个例子中描述的过程进行创建和处理，而无论使用的是哪种协议。

但同时，请牢记并非每个网络数据包都是从应用层协议产生的，所以你会进一步看到只包含第 2 层、第 3 层或第 4 层协议信息的数据包。

1.2.4 网络硬件

现在是时候来看看网络硬件了，至此脏活累活都已经完成，接下来的内容都很简单了。我们将专注于几种较为常见的网络硬件：集线器、交换机和路由器。

- **集线器**

  集线器一般是提供了多个 RJ-45 端口的机盒，就像图 1-4 所示的 NETGEAR 集线器。集线器从非常小的 4 端口的设备，到企业环境中安装机架设计的 48 端口机盒设备，变化很大。

因为集线器会产生很多不必要的网络流量，并仅在半双工模式下运行（不能在同一时间发送和接收数据），所以你通常不会在现代或高密度的网络中再看到它们的身影了（用交换机来代替）。然而，你应该知道集线器的工作机制，因为它们对于数据包分析技术非常重要，特别在实施我们将于第 2 章介绍的“枢纽”技术时。

图 1-4　一台典型的 4 端口以太网集线器

一台集线器无非就是工作在 OSI 参考模型物理层上的转发设备。它从一个端口接收到数据包，然后将数据包传输（中继）到设备的其他每个端口上。例如，如果一台计算机连接到一个 4 端口集线器的 1 号端口上，需要发送数据到连接在 2 号端口的计算机，那么集线器将会把数据发送给端口 1、2、3、4。连接到 3 号端口与 4 号端口上的客户端计算机通过检查以太网帧头字段中的目标媒体访问控制（MAC）地址，判断出这些数据包并不是给它们的，便丢弃这些数据包。图 1-5 所示为从计算机 A 发送数据到计算机 B 的例子，当计算机 A 发送出数据时，所有连接到集线器的计算机都将接收到数据，但只有计算机 B 会实际接受数据，而其他计算机则将丢弃它。

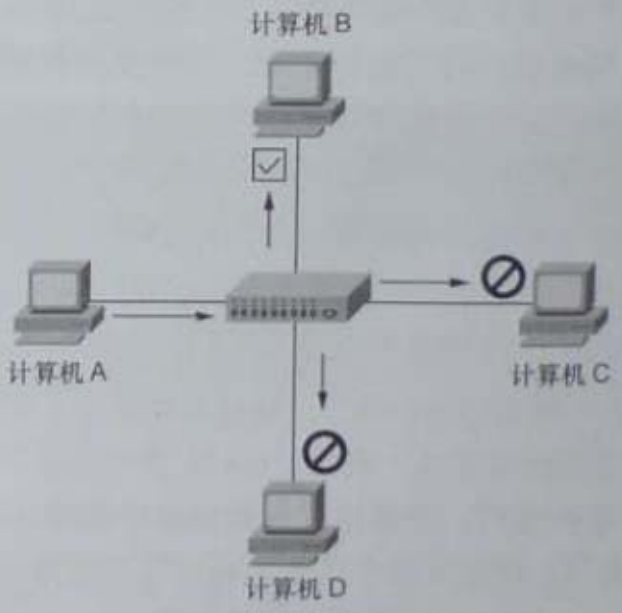

图 1-5　计算机 A 通过集线器传输数据到计算机 B 的通信流

做一个比喻，假设你发送一封主题为“所有的营销人员请注意”的电子邮件给贵公司所有雇员，而不只是那些在营销部门工作的人。市场营销部门的员工会知道这封邮件是给他们的，他们很可能会打开它，而其他员工将看到这封邮件并不是给他们的，则很可能会选择丢弃。你可以看到这会导致很多不必要的通信和时间浪费，然而这正是集线器的工作原理。在高密度的实际网络中，集线器最好的替代产品是交换机，它们是支持全双工的设备，可以同步地发送和接收数据。

- 交换机

与集线器相同，交换机也是用来中继数据包的。但与集线器不同的是，交换机并不是将数据广播到每一个端口，而是将数据发送到目的计算机所连接的端口上。如同你在图 1-6 中看到的那样，交换机的外表与集线器没什么两样。

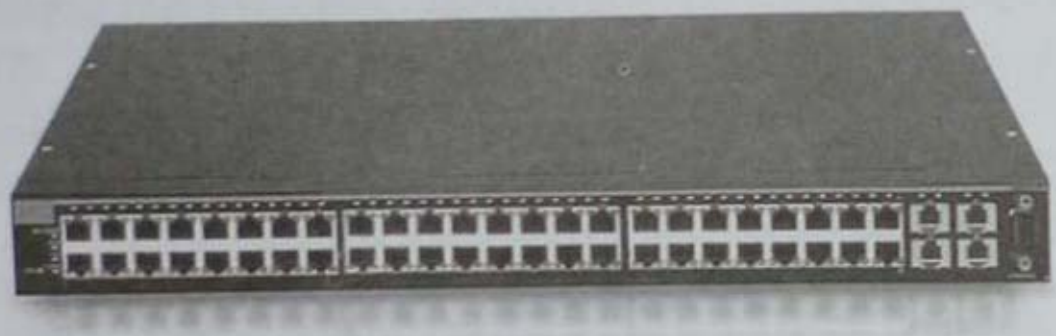

图 1-6　一个机架式 24 端口以太网交换机

市场上几个大牌公司的交换机，比如思科品牌的，能够通过专业化的供应商特定软件或 Web 接口进行远程管理。这些交换机通常被称为管理型交换机。管理型交换机提供了多种在网络管理中非常有用的功能特性，包括启用或禁用特定端口、查看端口细节参数、远程修改配置、远程重启等。

交换机在涉及处理传输数据包时，还提供一些先进的功能。为了能够直接与一些特定设备进行通信，交换机必须能够通过 MAC 地址来唯一标识设备，这意味着它们必须工作在 OSI 参考模型的数据链路层上。

交换机将每个连接设备的第 2 层地址都存储在一个 CAM（Content Addressable Memory 即内容寻址寄存器）表中，CAM 表充当着一种类似交通警察的角色。当一个数据包被传输时，交换机读取数据包中的第 2 层协议头部信息，并使用 CAM 表作为参考，决定往哪个或哪些端口发送数据包。交换机仅仅将数据包发送到特定端口上，从而大大降低了网络流量。

图 1-7 说明了流量经过交换机进行传输的过程。在这个图示中，计算机 A 发送数据到唯一的目标：计算机 B，虽然同一时间网络上可能会有很多会话，但信息将会直接通过交换机向目标接收者进行传输，而不会被传递到与交换机相连的所有计算机。

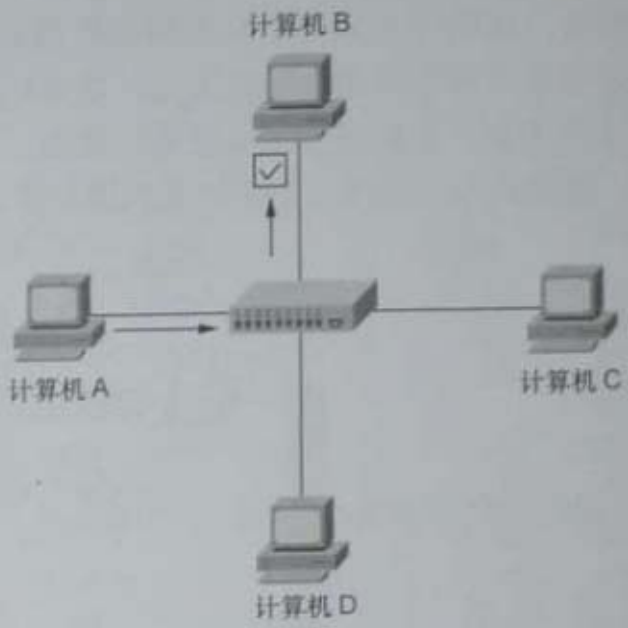

图 1-7 当计算机 A 通过交换机传输数据到计算机 B 时的通信流示意图

- 路由器

路由器是一种较交换机或集线器具有更高层次功能的先进网络设备。一个路由器可以有许多种不同的形状和外形，但大多数路由器在前面板上会有几个 LED 指示灯，在背板上会有一些网络端口，个数则取决于网络的大小。图 1-8 所示为一款路由器的示例。

图 1-8 一款低端的 Cisco 路由器，适合在一个中小型网络使用

路由器工作在 OSI 参考模型的第 3 层，它负责在两个或多个网络之间转发数据包。路由器在网络间引导数据包流向的这一过程被称为路由。几种不同类

型的路由协议定义了不同目的的数据包如何被路由到其他网络。路由器通常使用第 3 层地址（如 IP 地址）来唯一标识网络上的设备。

为了更清楚地解释路由的概念，我们以一个拥有几条街道的街区进行类比。假设有一些房子，它们都有着自己的地址，就好比网络上的计算机一样，而每条街道就好比网段，如图 1-9 所示。从你所在街道上的某个房子，你可以很容易地与同一街道中居住的邻居进行沟通交流，这类似于交换机的操作，能够允许在同一网段中的所有计算机进行相互通信。然而，与其他街道上居住的邻居进行沟通交流，就像是与不同网段中的计算机进行通信。

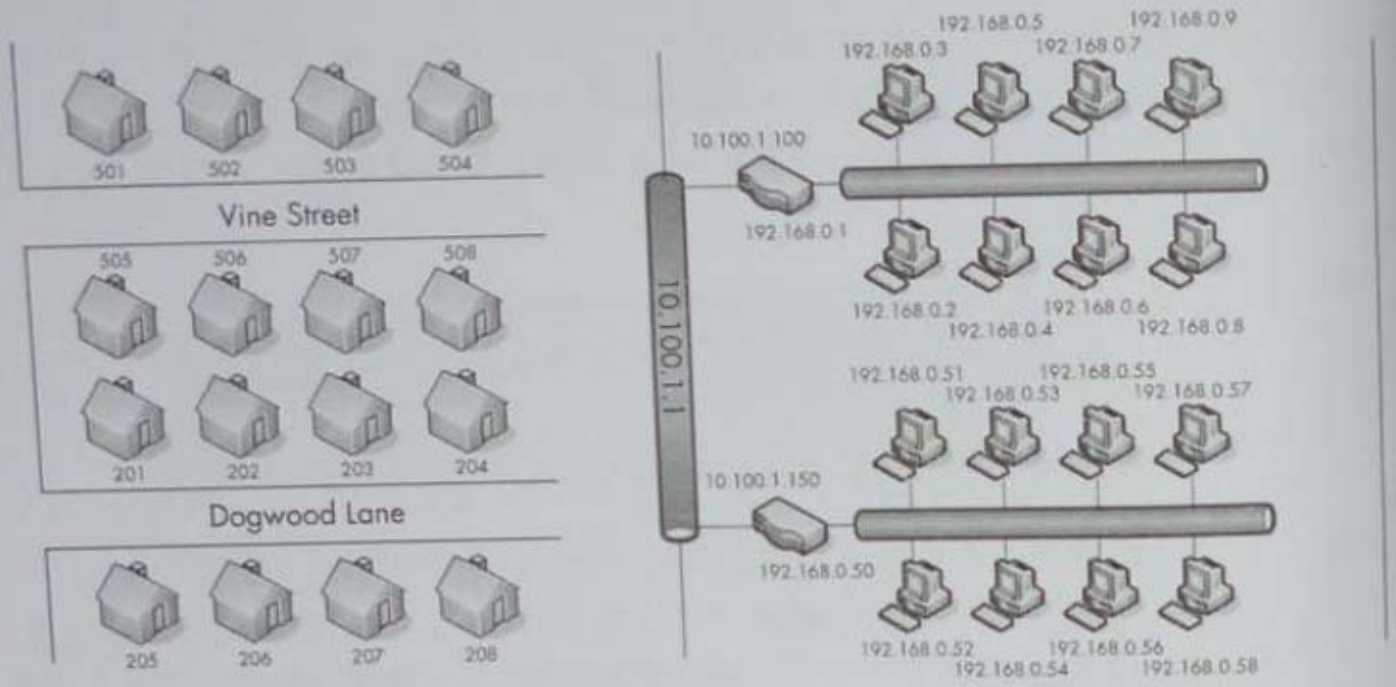

图 1-9　一个路由网络与邻街区的类比

参照图 1-9，假设你住在 Vine Street 503 号，需要到 Dogwood Lave 202 号。如果想要过去，你必须先到 Oak Steer 上，然后再到 Dogwood Lave。现在请对应到跨越网段的场景中，如果在 192.168.0.3 地址的设备需要和 192.168.0.54 地址的设备进行通信，它必须经由路由器到 10.100.1.1 网络上，然后再经过连接目的网段的路由器才可以到达目标网段上。

网络上路由器的数量与大小通常取决于网络的规模与功能。个人和家庭办公网络可能只需要一个小型路由器，放置在网络的中心。而大型企业网络则可能有几个路由器分布在不同的部门，都连接到一个大型的中央路由器或三层交换机上（具有内置功能，可以充当一台路由器的先进型交换机）。

当你开始查看越来越多的网络图时，你会更加了解网络数据流是如何流经这些不同类型的网络设备节点，图 1-10 所示为路由网络中一个非常常见的

布局形式。在这个例子中，两个单独的网络通过一个路由器进行连接。如果网络 A 上的计算机希望与网络 B 上的计算机进行通信，传输数据将必须通过路由器。

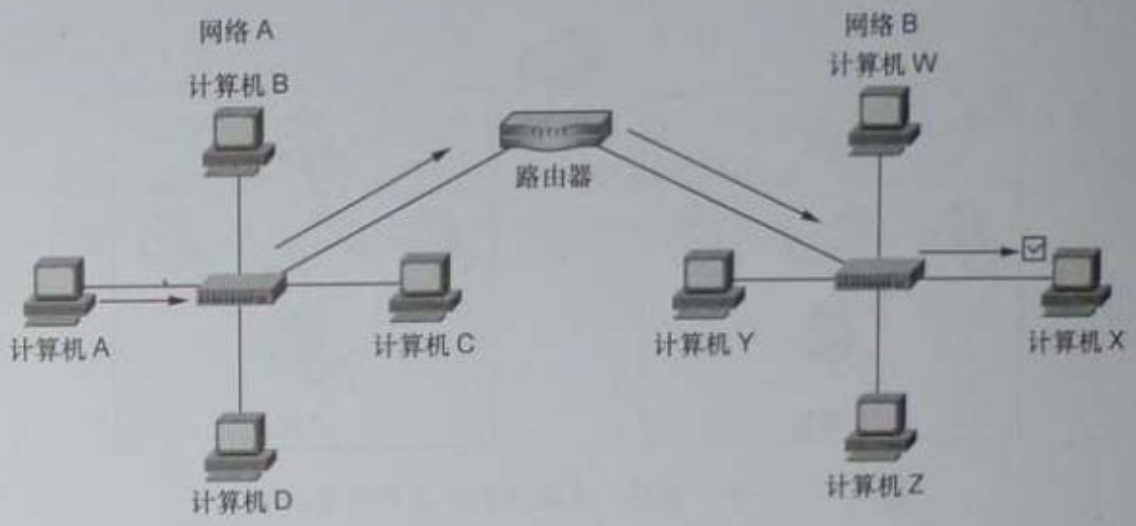

图 1-10　计算机 A 通过路由器将数据传送到计算机 X 的通信流示意图

1.3　流量分类

网络流量可以分为三大类：广播、多播和单播。每个分类都具有不同特点，决定着这一类的数据包该如何通过网络硬件进行处理。

1.3.1　广播流量

广播数据包会被发送到一个网段上的所有端口，而无论这些端口连接在集线器还是交换机上。但并非所有的广播流量都是以相同方式构建的，而是包括第 2 层广播流量和第 3 层广播流量两种主要形式。例如，在第 2 层，MAC 地址 FF:FF:FF:FF:FF:FF 是保留的广播地址，任何发送到这一地址上的流量将会被广播到整个网段。第 3 层也有一些特定的广播地址。

在一个 IP 网络范围中最大的 IP 地址是被保留作为广播地址使用的。例如，在一个配置了 192.168.0.XXX 的 IP 范围，以及子网掩码是 255.255.255.0 的地址网络中，广播 IP 地址是 192.168.0.255。

在通过多个集线器或交换机连接多种媒介的大型网络中，广播数据包将被一直从一个交换机被中继到另一交换机上，从而传输到网络连接的所有网段上。广播数据包能够到达的区域被称为“广播域”，也就是任意计算机可以不用经由

路由器即可和其他计算机进行直接传输的网段范围。图 1-11 显示了一个小型网络上存在两个广播域的例子。因为每个广播域会一直延伸直到路由器，而广播数据包只能在它特定的广播域中传输。

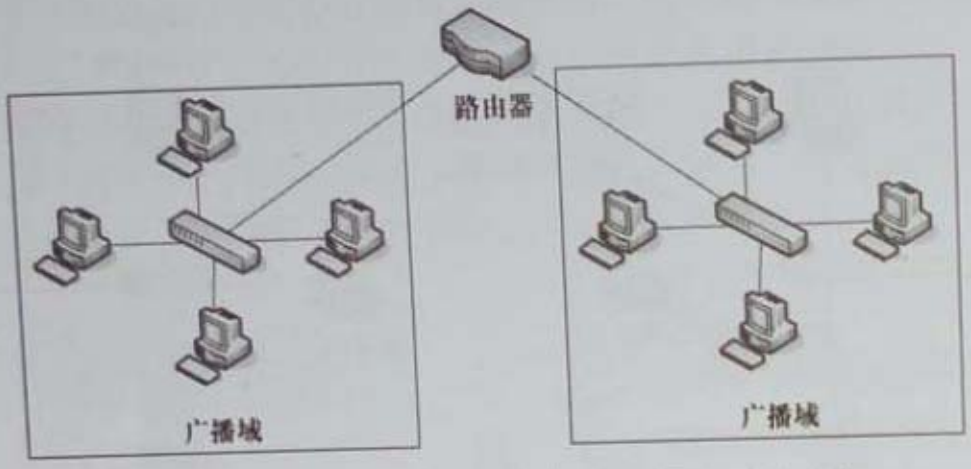

图 1-11　一个广播域一直延伸到路由器后面的网段

我们前面的类比也能很好地说明广播域是如何工作的。你可以将一个广播域想象成一条街道。如果你站在你家门口叫喊，只有街道上的人才能够听到你的声音。而如果你想与不同街道上的人说话，那么你需要找到一种与他进行直接交流的方式，而不是在你的家门口大喊大叫（广播）。

1.3.2　多播流量

多播是一种将单一来源数据包同时传输给多个目标的通信方式。多播的目的是为了简化这个过程，并使用尽可能少的网络带宽。多播流量通过避免数据包的大量复制来达到优化效果，而处置多播流量的方式则高度依赖于不同网络协议的实现细节。

实施多播的主要方法是通过一种将数据包接收者加入多播组的编址方案，这也是 IP 多播的工作原理。这种编址方案确保数据包不会被传送到未预期的目的地。事实上 IP 协议将一整段的地址都赋予了多播，如果你在网络上看到在 224.0.0.0 到 239.255.255.255 IP 范围内的地址，它很有可能就是多播流量。

1.3.3　单播流量

单播数据包会从一台计算机直接传输到另一台计算机。单播机制的具体实现方式取决于使用的协议。例如，一台设备希望与一个 Web 服务器进行通信，

这便是一个端到端的连接，所以通信过程将由客户端设备发送数据包到这台Web服务器开始。这种类型的通信就是单播流量的典型例子。

1.4 小结

本章涵盖了你学习数据包分析技术所必须掌握的基础知识。在你开始解决网络故障问题之前，你必须明白网络通信到底是怎么回事。在下一章中，我们将基于这些概念，来讨论更高级的网络通信准则。