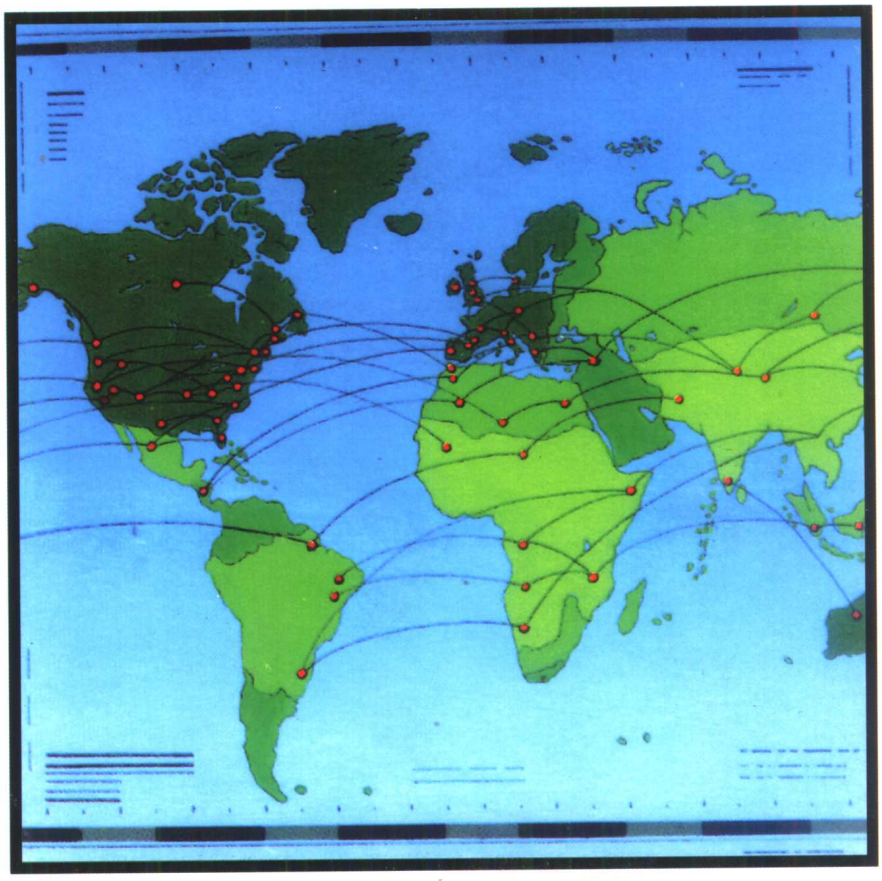
Sun公司核心技术丛书

HZ BOOKS

# Java

# 网络编程与分布式计算

## Java Network Programming and Distributed Computing



(澳) David Reilly　Michael Reilly　著
沈凤　等译

机械工业出版社
China Machine Press

Sun公司核心技术丛书

# Java网络编程与分布式计算

（澳） David Reilly
Michael Reilly 著
沈 凤 等译

机械工业出版社
China Machine Press

本书清晰地介绍了联网的基本原理，在进行网络编程时需要掌握的主要概念，以及在联网时可能遇到的问题和Java的解决方案。同时通过实例来介绍如何运用网络编程技术在Java平台上编写应用程序。

本书不仅适合于网络编程的初学者，而且还适合于有一定网络编程经验的程序员。



**本书版权登记号：图字：01-2002-453**

**图书在版编目（CIP）数据**

Java网络编程与分布式计算/（澳）赖利（Reilly, D.），（澳）赖利（Reilly, M.）著；沈风等译. -北京：机械工业出版社，2003.2
（Sun公司核心技术丛书）
书名原文：Java Network Programming and Distributed Computing
ISBN 7-111-11578-3

Ⅰ.J… Ⅱ.①赖… ②赖… ③沈… Ⅲ.Java语言－程序设计 Ⅳ.TP312

中国版本图书馆CIP数据核字（2003）第008135号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）
责任编辑：贾 梅
北京忠信诚胶印厂印刷·新华书店北京发行所发行
2003年3月第1版第1次印刷
787mm×1092mm 1/16·21.5印张
印数：0 001-4 000册
定价：38.00元

# 前　言

欢迎阅读本书。本书的目标是介绍和解释联网技术的基本概念，并讨论Java网络编程实践技巧。

本书将帮助读者提高网络编程的速度，并帮助读者运用在软件开发中所学的技巧。如果你已经拥有另一种语言的某些联网经验，而且想把你现有的技巧运用到Java中，那么你将发现本书是一本速成指南，而且是联网技术API的全面的参考书。然而，它并不要求你是联网方面的专家，因为第1～4章给出了对联网原理、Java以及Java联网API最基本要素的简明介绍。后面的章节会用Sun公司提供的作为参考的补充文档的讨论，来更详细地介绍Java API。

## 主要内容

在本书中读者将学会怎样运用网络编程技术来在Java平台上编写应用程序。Java API提供了在Internet上通信的很多方法，包括从发送数据包和数据流到实现HTTP等高层应用协议，以及分布式计算机制。

本书主要内容包括：

- Internet的工作方式、体系结构和TCP/IP协议栈。
- Java程序设计语言，包含对异常处理等最新技术的介绍。
- Java的输入/输出系统及其工作方式。
- 怎样使用UDP（User Datagram Protocol，用户数据报协议）和TCP（Transport Control Protocol，传输控制协议）编写客户和服务器。
- 使网络应用程序可以并发执行多个任务的多线程应用程序的优势。
- 怎样实现网络协议，包括实现客户/服务器实例。
- HTTP（HyperText Transfer Protocol，超文本传输协议），以及怎样通过Java访问万维网。
- 怎样为WWW编写服务器端的Java应用程序。
- 包括RMI（远程方法调用）和CORBA的分布式计算技术。
- 怎样用扩展的JavaMail API访问E-mail。

## 所需背景

要最大限度地利用本书，就要适当地熟悉Java编程。你需要能够编译并运行Java应用程序，并理解一些基本概念，像类、对象以及Java API。不过，你不必是了解本书所涵盖的I/O流和多线程等高级话题的专家。本书的所有范例使用的都是代码程序，所以不需要有GUI方面的经验。

你还必须安装Java SDK，可以从Sun公司的网站（http:// java. sun. com/j2se/）免费获得。毫无疑问，Java程序员可以使用SDK，但是请注意本书中的某些范例将需要JDK1.1，此外，更高

级的编程使用servlet、RMI和CORBA、JavaMail将需要Java 2。

还需要一些其他的软件，用于Java编程的大多数工具可以通过WWW免费下载。第2章概述了Java开发工具，但是读者也可以使用自己现有的代码编辑器。当范例重点涉及使用其他Sun公司软件时，书中会有提示。

## 相关网站

本书的Web站点包含了可下载的示例代码，以及关于Java联网技术的常见问题解答（FAQ），到联网技术资源的链接和关于本书的其他信息。可以通过下面的网址访问这个站点：

http:// www. davidreilly. com/jnpbook/。

## 联系作者

我们欢迎读者的反馈，读者可以对特定章节进行评注，或者对整本书进行评价。我们特别欢迎关于主题是否已表达清楚且足够详尽的读者来信。我们并不仅仅接受表扬，我们还很重视真诚的批评（也包括对新的联网技术论题的建议）。

与我们直接联系吧。尽管我们不能保证每个都回复，但是我们将尽力对你的询问作出反应。请通过e-mail：jnpbook@davidreilly. com发送问题和意见。

David Reilly和Michael Reilly

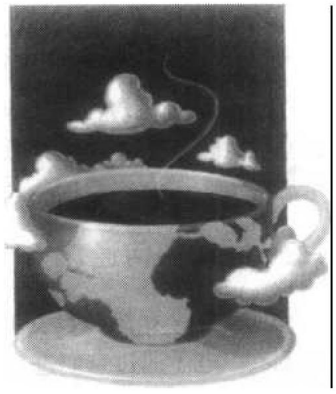
2001年9月

---

本书由沈风等组织翻译，具体参加本书翻译、录排、校对工作的人员还有：龚克、田蕴哲、牛志奇、丁天、龚舒宁、李红玲、白红利、金荣学、薛彪、叶哲、田韫、李林、张巧莉等。龚超、龚建同志对翻译稿进行了严格细致的复审。

# 目　录

# 第1章 联网原理



本章概述联网的基本概念，并讨论联网原理的基本主题。虽然关于基本的联网概念的复习教程将会是有用的，但是有联网经验的读者可以选择跳过这些预备内容的某些部分，并且在后面的章节中我们假定读者在这方面已经具备了这些理论知识。要进行网络编程，读者必须深刻理解构成TCP/IP协议族的各种协议的关系。

## 1.1 什么是网络

简而言之，网络就是共享通用通信协议和通用通信媒体（例如网络电缆、拨号连接和无线链接）的设备的集合。即使大多数人把网络看做是计算机的集合，但在此定义中，我们还是使用术语设备（device）而非计算机（computer）；当然，在大多数人的头脑里，网络的基本概念就是网络服务器和台式机的集合。

然而，把网络说成仅仅是计算机的集合那就限制了可以使用的硬件范围。例如，可以在网络中共享打印机，使得不只一台机器可以访问它们的服务。其他类型的设备也可以被连接到网络中，这些设备可以提供到信息的访问，或者提供可以远程控制的服务。实际上，存在着把非计算机设备连接到网络中的不断增长的趋势。尽管技术还在演进中，但我们还是朝着网络中心化而非计算中心化的模式转移。服务和设备可以被分布在网络中而不是被绑定在个别的机器上。同样的，用户可以在机器与机器之间移动，就像他们正坐在自己所熟悉的终端前登录系统一样。

从联网技术历史的初期开始的一个有趣并很流行的例子是连接到Internet的苏打机器，通过它世界各地的人们都可以看到某种口味的饮料还有多少。虽然只是一个微不足道的应用，但是它却被用来演示联网设备的力量。实际上，当家庭网络变得更易于使用且花费得起时，我们甚至可以看到常见的家用器具，像电话、电视以及家用立体声系统被连接到局域网中，甚至Internet中。

现在已经有网络和软件标准，如Sun的Jini可以帮助设备和硬件在网络中相互通信，并提供即时的即插即用功能。不需要复杂的管理和配置就可以把设备和服务添加到网络中或者从网络中移除（例如，当你拔掉打印机并把它转移到隔壁房间）。人们期望在接下来的几年中，用户将会在网络中心化的计算中感到舒适和熟悉，就像他们在Internet中所感受的那样。

除了提供服务的设备之外，还有一些设备是保持网络正常运作的。根据一个网络的复杂程度和它的物理结构，构成网络的要素包括网卡（network card）、路由器（router）、集线器（hub）和网关（gateway）。这些术语的定义是：

- 网卡是添加到计算机中使其可以与网络通信的硬件设备。现在用得最普遍的网卡是以太网

卡。网卡通常连接到网络电缆，后者连接到网络和媒体中，数据通过网络电缆传输。然而，还存在其他媒体，例如通过电话线的拨号连接，以及无线连接。

- 路由器是充当交换设备的机器。这些机器把数据包转到它们在网络中传输的下一“跳”。
- 集线器提供允许多台机器访问同一个网络的连接（例如，允许两台台式机访问同一个局域网络）。
- 网关把一个网络连接到另一个网络——例如，从局域网到Internet。虽然路由器和网关很相似，然而路由器不需要桥接多个网络。在某些情况下，路由器也是网关。

由于联网术语在联网文本和协议规范中被广为使用，因而理解这些术语很有用，但是程序员通常并不需要关心网络实现的细节以及它的底层结构（underlying architecture）。然而，对程序员来说，了解构成网络的不同要素却很重要。

## 1.2 网络如何通信

网络由计算机和设备间的连接组成。这些连接通常是物理连接，例如电线（wire）和电缆（cable），电子通过它们进行发送。然而，还存在很多其他媒体。例如，可以使用红外线（infrared）和无线电波（radio）作为通信媒体来无线传输数据，或者还可以使用利用光而不是电的光纤。

这些连接把数据从网络中的一点携带到另一点。这些数据以位信息的方式表示（不是“on”就是“off”，不是“0”就是“1”）。不管是通过电缆等物理媒体、还是通过空气或者使用光，基本数据都在被称为节点（node）的网络中的不同点之间被传递；节点可以代表一台计算机、其他某种类型的硬件设备（例如打印机），或者把信息前向中继到网络中的其他节点或另一个完全不同的网络的联网设备。当然，要把数据成功地传递到各个节点，这些节点必须可以被清楚地识别。

### 1.2.1 编址

网络中的每个节点通常由一个地址来表示，就像街道名称和编号、城镇或城市、以及邮政编码标识了各个家庭和办公室一样。被安装在这些设备中的网络接口卡（Network Interface Card，NIC）的制造商负责保证没有两个网卡的地址是相同的，并负责选择一个合适的编址方案。每个网卡都永久性地存储着这个地址，所以它保持固定不变——不可以通过手工指定或修改，即使在与另一个网卡的地址冲突时，某些操作系统允许这些地址被伪造。

由于NIC之间大相径庭，所以使用了很多编址方案。例如，以太网卡被指定用一个独一无二的48位编号来把它们区分开。通常，为每个网卡指定一个数字编号，为每个制造商分配一批编号。当然，这个系统必须由业界严格管制——具有相同地址的两个网卡将会使网络管理员头痛不已。物理地址与许多名称有关（某些名称是某类网卡所特有的，而其他则是通用的术语），包括：

- 硬件地址。
- 以太网地址。
- MAC（Madia Access Control，介质访问控制）地址。
- NIC地址。

这些地址被用来向适当的节点发送信息。如果两个节点共享相同的地址，那么它们将竞争

相同的信息，其中一个节点将不可避免地会失败，或者两者都接收到相同的数据。通常，机器通过不只一种地址被识别。网络服务器具有物理以太网地址以及把它与Internet上的其他主机区分开的网际协议（IP）地址，或者它可能拥有不止一个网卡。

在局域网内，机器可以使用物理地址来通信。但是，由于存在很多地址类型，所以它们并不适合于网间通信。如本章后面所讨论的那样，IP地址被用于这个目的。

### 1.2.2 使用包的数据传输

在节点间发送单独的数据位的成本是比较高的，因为每次传输必须提供必要的地址信息。作为替代方案，多数网络把数据封装成包（packet）。包由首部和数据段组成，如图1-1所示。首部包含了寻址信息（例如发送方和接收方）、确保包没有被破坏的校验和以及在网络中传输所必须的其他信息。数据段包含字节序列，由从一个节点被传送到另一个节点的实际数据组成。由于首部信息只是在传输中需要，所以应用程序只对数据段感兴趣。在理想情况下，尽可能多的数据将会被组合到包中，用以最小化首部的总开销。然而，如果需要快速地发送数据，那么当包几乎为空时也会被发送。根据包的类型和所使用的协议，包可能还会被填充以适合某个固定的字节长度。

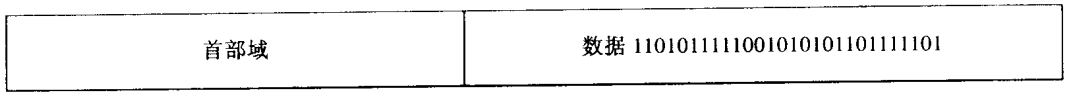



图1-1 包首部的图示

当网络上的节点准备发送包时，通常并没有到目的节点的直接连接。相反地，中间节点把包从一个位置传送到另一个位置，这个过程不断重复直到包到达它的目的地。由于网络环境（例如冲突或网络故障），包可能通过任意路由器，有时它们可能在传输中丢失或者乱序到达。这看起来像是一种混乱的通信方式，但是如同在后面的章节中所能看到的那样，这些是保证递交（delivery）和定序（sequencing）的方式。实际上，有保证的递交和排列顺序通常与特定的应用类型无关（例如视频流和音频流，在这些应用中展现当前的视频帧和音频段远比重传丢失的数据重要）。当需要这些特性时，网络软件可以跟踪应用程序丢失了的包和乱序的数据。

包传输和信息的原始位传输是低级过程，而大多数网络编程处理数据的高级传输。与其同时包含从原始字节到包然后到实际程序数据的整个传输，不如考虑把这些不同通信类型组合成单个层（layer），这样做更有帮助。

## 1.3 层间通信

层的概念被引入用来确认和表达联网原理的复杂性。最流行的网络分层方法是开放系统互联（Open System Interconnection，OSI）模型，由国际标准化组织（International Standards Organization，ISO）创建。这个模型把网络操作分成七个部分，从最基本的物理层到Web客户和E-mail服务器等软件应用程序通信的应用层。

在OSI模型之下，通信协议被包，其中七层中每一层可以用编号或描述性的名字来称呼。通常，当网络程序员说到某个特定层时（例如，层n），他们指的是OSI模型中的第几层。图1-2图

解了七层中的每一层。

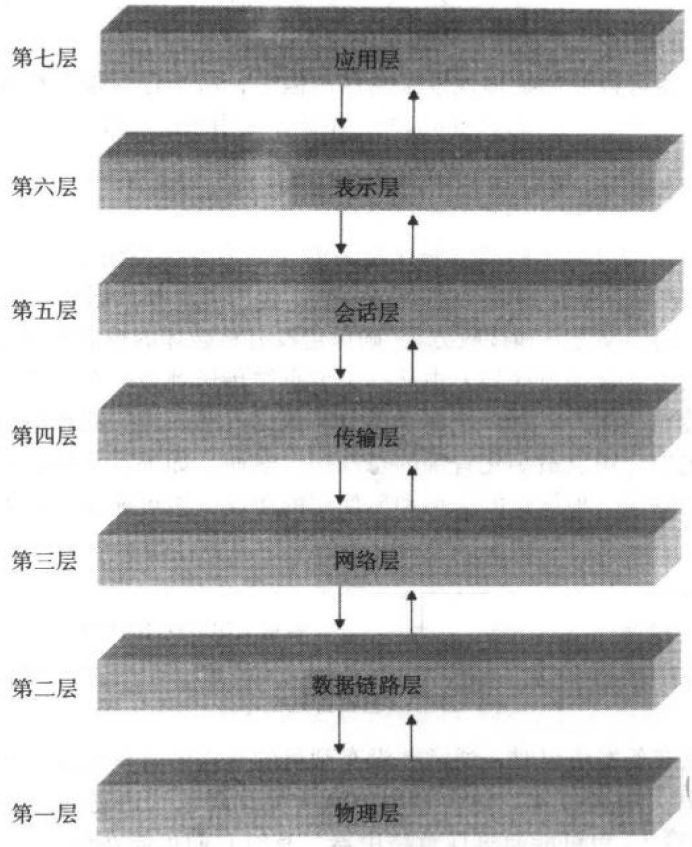



图1-2 OSI参考模型的七层

每层都负责通信任务中的某些形式，但是每个任务都是狭义地定义的，并且通常依赖于它之上的一层或多层服务。在某些系统中，一个或多个层可能不存在，而在其他系统中所有层都用到了。然而，在通常情况下，操作系统只使用了七个层次的一个子集。通常，程序员限制他们自己一次只在一个层上操作；下面层的细节因此被隐藏了。当为一个层（假定为Internet中的通信）编写软件时，我们作为程序员并不需要关心下列问题：例如初始化调制解调器连接以及把数据从通信端口发送到调制解调器。把网络分层形成了一个简单得多的系统。

### 1.3.1 第一层——物理层

物理层是网络通信的最基本层次。它管理网络节点间通信的最底层形式。在这个层次中，网络硬件，像网卡和电缆，在两个节点间传送位序列。Java程序员不在这个层次上操作——它是硬件驱动程序开发者和电子工程师的工作领域。在这个层次上，不做任何实际努力来确保无错的数据传输。存在各种导致差错的原因，例如由于外界源的干扰而产生的电压尖峰脉冲，或者使用模拟传输媒体的网络的线性噪音。

### 1.3.2 第二层——数据链路层

数据链路层负责提供更可靠的数据传输，并负责把数据组合成帧（frame）。帧与数据包相似，但它是特定于单个硬件体系类型的数据块（而数据包被用于更高的层次，并且可以从一种网络类型到另一种网络类型之间转移）。帧具有用来检测传输差错的校验和，并且通常具有警告硬件做好帧之间分割的“起始”和“结束”标记。帧序列在网络节点之间传输，如果某个帧被破坏了，那么它将会被丢弃。数据链路层确保错乱的数据帧不会被传送到更高层，从而迷惑应用程序。然而，数据链路层并不保证被破坏帧的重传，更高的层通常处理这种行为。

### 1.3.3 第三层——网络层

从在网络中发送帧的数据链路层往上移动，我们到达了网络层。网络层处理数据包而不是帧，并引入了几个重要的概念，如网络地址和路由。包在网络间被传送，在Internet情况下它的传送遍布全球。除非传播到邻接的、只有一个选择的网络，否则这些包通常会选择另一可选择的路由（路由由路由器决定）。这个层次的通信是很低级的，很少要求网络程序员为这个层次编写软件。

### 1.3.4 第四层——传输层

第四层，即传输层，与控制数据的传输方式有关。这个层次处理这些问题：如自动差错检测和纠错、流控制（限制发送的数据量以防止超载）。

### 1.3.5 第五层——会话层

会话层的目的是使得应用到应用间的数据交换，以及通信会话的建立和终止。会话管理包含了多种任务，包括建立会话、同步会话以及重建被突然中断的会话。由于面向连接的通信的额外开销会增加网络延迟和带宽消耗，所以并不是每种应用都需要这种服务。有些应用可能会选择使用通信的无连接方式作为替代。

### 1.3.6 第六层——表示层

第六层负责数据表示和数据转换。不同的机器使用不同的数据表示（在一个系统中可能用8位来表示整数，而在另一个系统中用16位表示）。某些协议可能要压缩或加密数据。任何时候,当数据类型从一种格式转换到另一种格式时，表示层都会处理这种任务。

### 1.3.7 第七层——应用层

OSI层次的最后一层是应用层，这是大量程序员编写代码的层次。应用层协议指定服务请求的语义，例如请求文件或检查E-mail。在Java中，虽然某些低层的服务可能会被调用，但几乎所有网络软件都是为应用层编写的。

## 1.4 分层的优势

把网络协议和服务划分为几个层次不仅有助于通过把它们分为更小的、更易于管理的单元

从而简化联网协议，而且有助于提供更大的灵活性。通过把网络协议划分为几个层次，协议可以为互操作性而设计。使用层n的软件可以与运行于另一台机器上的支持层n的软件通信，而不用考虑层n-1，层n-2等等的细节问题。例如，可以发布和取代更低级的层而不用修改和重新设计更高级的层，或重新编译应用软件。例如，网络层协议可以工作于以太网或令牌环网，甚至在物理层和数据链路层使用两种不同的协议和设备。在多机种网络世界中，这是个重要的特性，因为它使得网络可以互操作。

## 1.5 Internet体系结构

网络历史中最重要的革命是Internet——共享通用通信族（TCP/IP）的小型网络的全球集合——的演变。在这儿使用了*演变*而不是*创建*，是由于Internet并不是简单地在某一天突然存在并运行的。在过去的几年里，Internet已经扩展到包括现在我们所拥有的一切；它已经从一个被称为ARPANET的国防通信项目演变到世界范围内的、延伸到商业和非商业领域的网络集合。对Internet设计的贡献来自于最初的ARPANET开发者以及学术和商业研究者，他们提供了有助于形成网络现状的建议和改进措施。

Internet是建立于普通网络、传输以及应用层协议之上的开放系统，然而它却保证了连接各种计算机、设备以及操作系统的灵活性。不管个人正在运行的是PC、Unix、Macintosh或者掌上电脑，通信和翻译的复杂性都由TCP/IP协议族为用户透明地处理了。

**提示** Internet的历史是一个引人入胜的话题，然而某些读者可能会觉得枯燥无味。那些想了解更多关于Internet历史以及对参与到它的演变过程中的人感兴趣的读者可以参考大量在线资源。其中一个最好的资源来自于Internet社区，它的网址是：http://www.isoc.org/internet/history/。

### 设计Internet

我们今天所知道的Internet是多年革新和试验的产物。构成TCP/IP协议族的协议都是被仔细地设计、测试并经过多年改进的。其中一些主要目标（表述在RFC 871中[⊖]）是要达到：

- 网络间的资源共享，通过创建支持网间通信或“网络间的相互作用”的网络协议来实现。构成Internet的不同协议必须支持各种联网网关。
- 硬件和软件无关性，通过创建可以与任何CPU架构、操作系统以及网卡彼此协作的网络协议来实现。
- 可靠性和健壮性，通过创建容错的网络协议来实现，这样的协议无需考虑各种中间网络的状态，为了把数据传送到目的地，在必要时可以重新路由数据。因为Internet起源于一个国防研究项目，所以在发生灾难性网络故障情况下的健壮性就极为重要。可以绕过已损坏的网络，因此整个Internet还是依然可访问。
- 高效而简单的“好”协议，通过创建使用像通信套接字、网络端口等概念的设计原则的网络协议来实现。虽然这样的设计目标在现在看起来是及其自然的，但设计者们必须把

⊖ RFC（Request for Comment，请求评注）规范，在8.8.2节中有更详细的描述。

TCP/IP开发成适应于长期的、大量的应用，而且还要使得它使用起来尽可能简单。

与特定硬件和软件体系无关的、健壮的、容错的、以及高效且易于学习的通用协议带来了连接到Internet的计算机和网络之间互联的简易性。结果，我们就拥有了TCP/IP协议族。下面讨论这个协议族中所包含的每个协议。

1. IP协议

IP协议（Internet Protocol，网际协议）是第三层（网络层）的协议，它被用来在Internet中传输数据包。毫无疑问，它是世界上最广为使用的联网协议，并且已经扩展得极为丰富了。不管使用的是何种联网硬件，几乎都会支持IP联网。网际协议充当了不同类型网络之间的桥梁，从而形成了世界范围内的、由计算机和小型子网所构成的网络（见图1-3）。实际上，很多组织在他们的局域网内使用IP及其相关的协议，因为在网络内部和外部使用网际协议，其效果都同等地好。

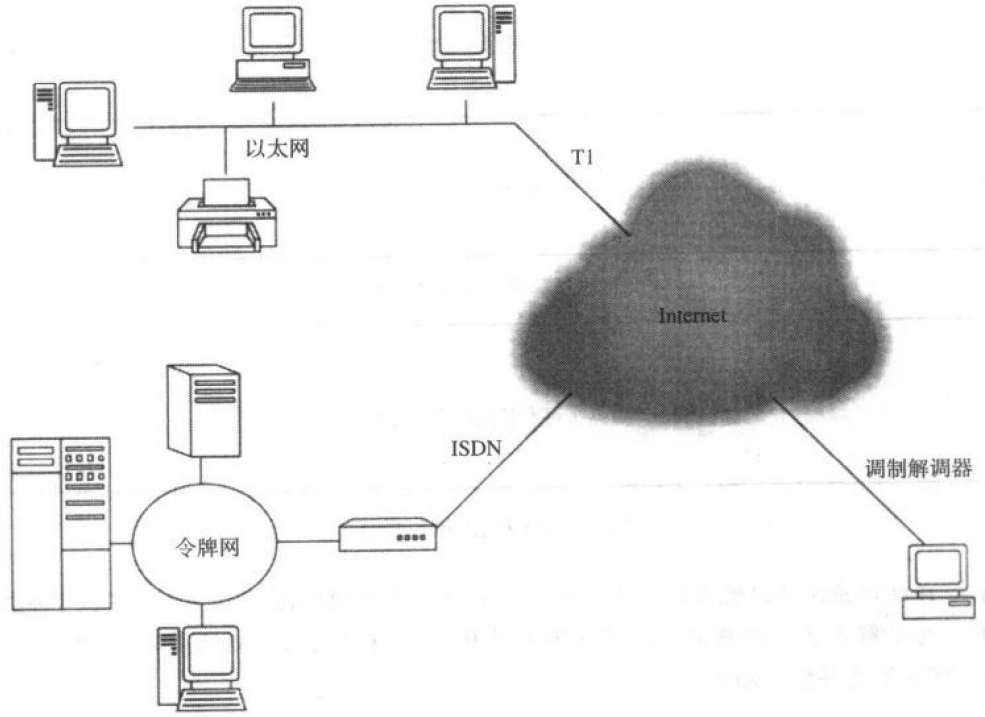



图1-3 在不同的物理网络中支持IP联网

IP协议是一个包交换网络协议。信息以IP包（通常也被称为IP数据报（datagram））的形式在两台主机（host）之间进行交换。每个数据报都被当做一个离散单元（discrete unit），与其他任何以前发送的包无关——在网络层中机器之间不存在“连接”。作为替代的是，发送一系列数据报，并且传输层中的高层协议提供连接服务。

（1）IP数据报格式

IP数据报携带了确定它的传输方式的重要信息。这些信息被存储在数据报首部，其后跟随着被发送的实际数据。在图1-4中显示了不同的首部域以及它们的大小。