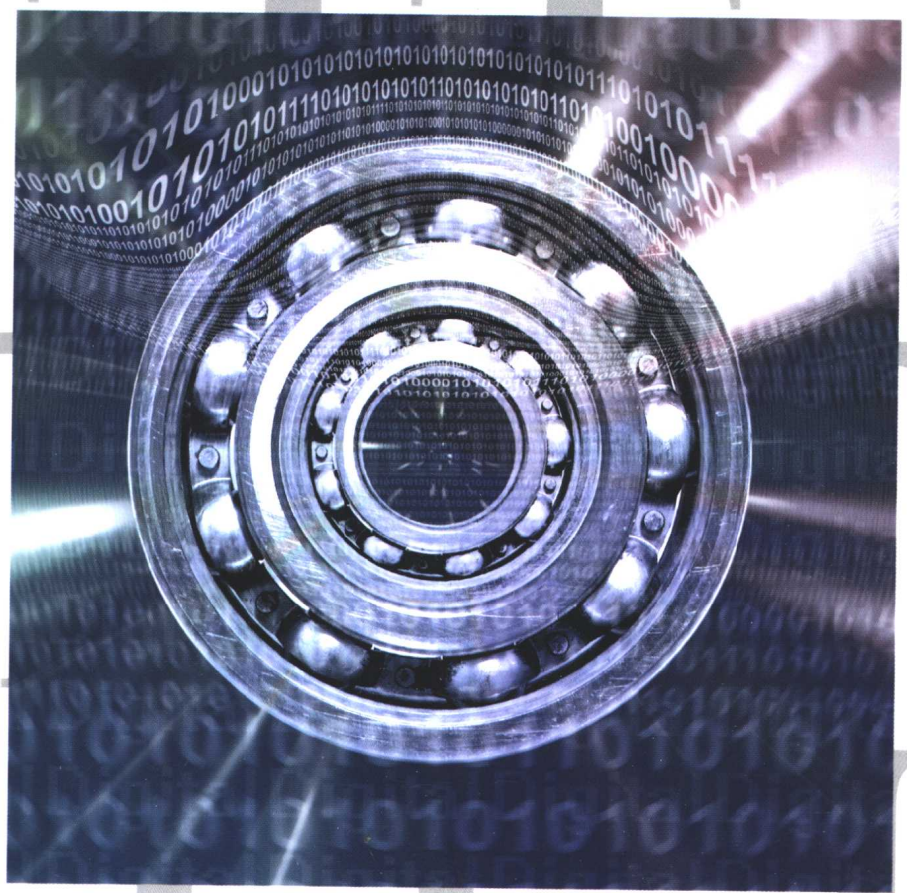
TCP/IP Application Layer Protocols for Embedded Systems

国外IT精品丛书



嵌入式系统TCP/IP应用层协议

- 使系统开发人员了解如何为添加嵌入式设备连通性而开发和部署应用层协议
- 涵盖HTTP、SMTP、SLP以及联网API
- 讨论新协议，如SOAP、XML-RPC、SIP、SCTP、OSGI、WSDL等
- 选配光碟（Win/Linux）内有本书中实现的所有协议源代码和所有相关的RFC

〔美〕 M. Tim Jones 著

路晓村 徐 宏 王泰东 等译

薛荣华 审校

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

TCP/IP Application Layer Protocols for Embedded Systems

嵌入式系统TCP/IP应用层协议

〔美〕 M. Tim Jones 著

路晓村 徐 宏 王泰东 等译

薛荣华 审校

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 提 要

本书论述如何为嵌入式设备添加连通性开发和部署应用层协议。目前，通信正在快速地成为嵌入式系统的一般需求。事实上，很难找到不包含外部通信形式的嵌入式系统。嵌入式系统现在正在低带宽无线链路上发送电表读数以免需要到现场抄表。全球定位系统（GPS）技术和嵌入式系统的无线链接还能精确测定国内任何地方卡车车队的位置、速度、油压和其他参数。创建这些应用和其他网络应用程序是本书的重点。利用TCP/IP应用层协议（如HTTP、SMTP、POP3、SNMP和SLP）的实践指南，开发人员将学会如何在他们的嵌入式系统中开发和部署这些协议。

本书适用于手持设备编程人员、无线网络编程人员及大专院校计算机和通信专业师生。

版权贸易合同登记号：01-2002-6691

图书在版编目（CIP）数据

嵌入式系统TCP/IP应用层协议/（美）琼斯（Jones, M. T.）著；路晓村等译.—北京：电子工业出版社，2003.4

书名原文：TCP/IP Application Layer Protocols for Embedded Systems

ISBN 7-5053-8619-0

Ⅰ. 嵌… Ⅱ. ①琼… ②路… Ⅲ. 计算机网络－通信协议 Ⅳ. TN915.04

中国版本图书馆CIP数据核字（2003）第023157号

责任编辑：春 丽 和 敬

印 刷：北京天竺颖华印刷厂

出版发行：电子工业出版社 http://www.phei.com.cn

北京市海淀区万寿路173信箱 邮编：100036

北京市海淀区翠微东里甲2号 邮编：100036

经 销：各地新华书店

开 本：787×1092 1/16 印张：20.5 字数：520千字

版 次：2003年4月第1版 2003年4月第1次印刷

定 价：35.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换，若书店售缺，请与本社发行部联系。联系电话：（010）68279077

译 者 序

本书论述如何为嵌入式设备添加连通性而开发和部署应用层协议。目前，通信正在快速成为嵌入式系统的一般需求。事实上，很难找到不包含外部通信形式的嵌入式系统。嵌入式系统现在正在低带宽无线链路上发送电表读数以免需要人到现场抄表。全球定位系统（GPS）技术和嵌入式系统的无线链接还能精确测定国内任何地方卡车车队的位置、速度、油压和其他参数。创建这些应用和其他网络应用程序是本书的重点。全书共有13章和5个附录。分别论述网际互联、TCP/IP与嵌入式系统、应用层协议、嵌入式SMTP客户机、嵌入式SMTP服务器、嵌入式POP3客户机、嵌入式HTTP服务器、嵌入式SNMP代理、嵌入式命令行界面、嵌入式服务位置协议（SLP）、嵌入式NNCP客户机、设计新的应用层协议和协议概论。附录包括BSD套接字API入门、BSD套接字API选项、TCP/IP协议栈、包报头及选配光碟上的内容。利用TCP/IP应用层协议（如HTTP、SMTP、POP3、SNMP和SLP）的实践指南，开发人员将学会如何在他们的嵌入式系统中开发和部署这些协议。本书的选配光碟上包含有本书中开发的所有软件的源代码。相信本书对嵌入式系统的开发人员会有指导作用。

本书由路晓村（第1～第7章）、徐宏（第8～第13章）和王泰东（全部附录）翻译，由薛荣华审校并统稿。参加本书译录校工作并给予大力协助的还有赵继红、薛姗、曹汉征、许秀英、闫慧娟、矫克民、王景中、徐小青、姚栋、徐凤麟、薛晏、刘晓玉、刘东顺、沈兰英、王建成等同志。电子工业出版社和美迪亚电子信息有限公司的编辑们为本书的出版做了大量艰苦细致的工作，译者谨向所有为本书的出版提供帮助的同志表示由衷的谢意。由于译者水平有限，译文中难免有不妥之处，欢迎读者批评指正。

致　谢

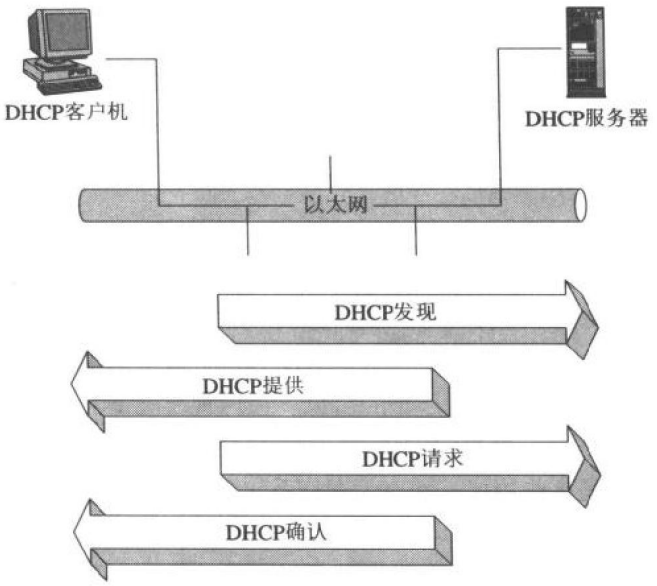
感谢我的妻子Jill和孩子们——Megan、Elise和Marc在本书写作期间给予我的支持和耐心。

还要感谢我的父母Maury和Celeta在1979年送给我的礼物——一台TRS-80，它使我开始了一个美妙的生涯。

最后，感谢Thomas Herbert和Dan Klein特别有价值和有帮助的审阅。

目　录

第1章　网际互联概述



本章介绍了使用因特网协议进行网际互联的概况，阐述了开发应用层协议所需的基本原则。本章还提供了联网原则的基础知识以及有关术语。

概述

在人们对冰箱中能够在因特网中进行通信的嵌入式系统的实用性感到惊讶的同时，已经又出现了很多依靠通信技术创造出的十分有用的产品的嵌入式系统应用实例。嵌入式系统现在可以通过低带宽的无线链路传输电度表读数，而不必到现场目视查阅。采用全球定位系统（GPS）技术和无线链路的嵌入式系统可以用来确定在国内任何地方行驶的车队的准确位置、速度、油压和其他参数。这些应用实例不但有用，而且经济上也是可行的。

本书集中讨论应用层协议，由于引入了传输控制协议/因特网协议（TCP/IP）的协议族，这是进行开发的关键所在。因特网爆炸性的增长是从两个极为重要的应用层协议的开发开始的：SMTP（简单邮件传输协议）和HTTP（超文本传输协议）。SMTP是一种通过因特网传输邮件的协议，HTTP是一种通过Web浏览器提供丰富内容传递的协议。几乎可以确信因特网的下一步发展将是新应用层协议，这些协议将满足新的且有待发现的需求。

嵌入式系统的通信

在相互连接不断加强的世界中，通信对于嵌入式系统来说很快变成是普遍性需求。很难找出没有任何形式外部通信功能的嵌入式系统。对于这个要求不必担心，因为网络技术可

以满足嵌入式领域的挑战，使得开发这种系统变得更加简单。

网络化应用程序只不过是一种程序，它能够在网络中与另一个网络化应用程序进行通信。使用TCP/IP协议族和标准的Berkeley Sockets API（应用程序接口），创建网络化应用程序变得很简单，正如在本章后面看到的客户机和服务器的实例那样。

嵌入式系统通常使用两种类型接口中的一个进行连接：以太网或串行链接。以太网连接要求提供本地网络，设备可以在上面进行连接（通常使用RJ-45连接器）。通过串行链接的连接打开了很多其他通道，因为可以利用与其他作为网关的设备实现通信，或者通过一个调制解调器（无线的或者有线的）。我们将在第2章中重新讨论这些方式及其功能。

网络结构

网络化应用程序一般都遵循所谓的客户机－服务器的模式。在这个模式下，服务器将一些服务扩展到一个或多个连接的客户机。这种模式的实例是Web服务器和客户机。Web服务器通过信息发布来提供服务。浏览器或者Web客户机负责向用户提供收集到的信息。客户机和服务器相互的通信使用的就是所谓的协议。在这种情况下，协议是HTTP。HTTP以及一般情况下的协议可以提供一种共同的语言，作为客户机和服务器进行“对话”的基础。

还有必要指出的是这些协议是独立于操作系统和平台结构的。Web服务器这里可以使用Solaris操作系统（UNIX的变形）运行在Sun SPARCstation上面，而客户机可能是一个Windows 2000桌面。虽然操作系统和物理结构不同，但是客户机和服务器仍然可以进行通信。Web服务器和客户机甚至可以使用不同的语言编写，因为双方同意在标准的协议组下面（在本例中是TCP/IP和HTTP）工作，所以通信的语义是事先有保证的。

协议分层

现代的计算机网络被称做是包交换网络。当用户向HTTP服务器请求一个大文件时，协议会将这个文件分成小片，这些片最终被定义成包。这些包不但包括准备传输的信息（被称为有效负荷），而且还有告诉网络这些信息的去向以及什么地方被积累成由更大的流（报头）构成的包集。

产生这些包的过程是通过协议栈的层（参阅图1.1）。最上层叫做应用层。HTTP协议就属于这个层。在应用层下面是传输层，这里最常见的协议有TCP和UDP（用户数据报协议）。接下来是网络层，它包含IP协议。最后是数据链路层，这里提供了对各种物理接口的接入，如以太网或串行链接。

包是在协议栈每个层面上创建起来的，封装前一层的信息并向下传递。例如，Web客户机将传递一个HTTP请求消息给传输层。

然后TCP/IP协议使用一个传输层协议数据单元（PDU）封装这个报文。该PDU通过栈继续向下传递，后续的每个层都增加一个该层的报头，形成PDU。一旦包到达数据链路层，它就在到达其目的地的路上。当到达目的地时，该过程继续进行，只不过方向相反。协议栈的每个层都分析和去除报头，并通过栈将结果PDU向上传递给预定的接收方（参见图1.2）。每个层具有自己独特的重要性；本章后面将分别详细讨论这些层。

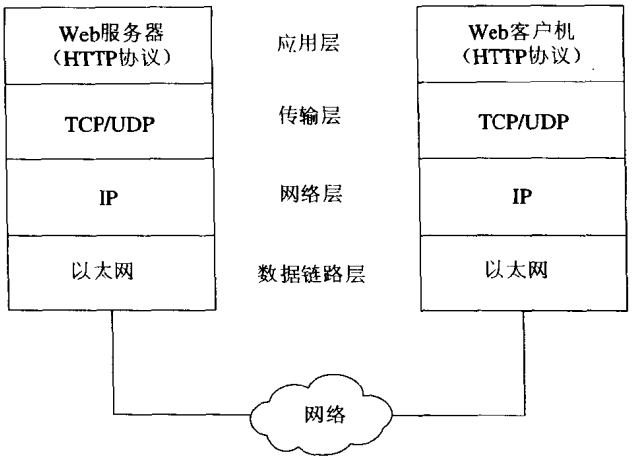


图1.1　协议栈的各个层

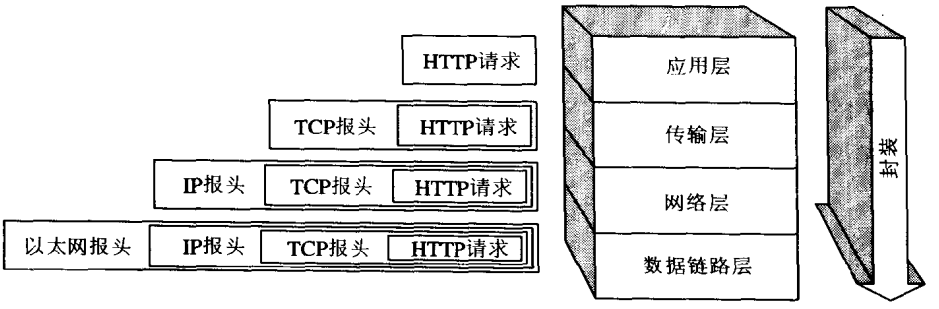


图1.2　通过协议栈的PDU构造

协议层

因为已经讨论了协议结构的某些初步知识，现在开始再深入到每个层中，更好地理解协议为什么在那里以及这些协议能做什么。

应用层

应用层与最大量协议有关，这些协议可以被应用程序最完全地识别。例如HTTP通常与Web浏览器应用程序有关，但是HTTP在很多其他环境中也被认为是应用程序。在前一节里面，已经讨论过浏览Web页的HTTP协议，但是还有很多其他协议存在（参见表1.1中的简述）。

表1.1　应用层协议实例

缩写	协议名称	说明
FTP	文件传输协议	用于在因特网中的主机之间移动文件
SMTP	简单邮件传输协议	用于通过因特网传输电子邮件
NTP	网络时间协议	允许因特网主机与因特网上的调整源同步时钟
NNTP	网络新闻传输协议	用于通过因特网传播Usernet新闻
HTTP	超文本传输协议	用于在因特网中传播丰富的媒体内容
TELNET	远程登录协议	通用的双向、面向字节的通信功能
SOAP	简单对象存取协议	以独立于平台的方式用于发布因特网上的服务

在此应该特别指出的是应用层是最一般的，可以以多种方式用于不同目的。应用层是专用和非专用的分界线。应用层协议提供了一种专门用途，可以使用基本的协议使其能够在因特网上通信。现在协议开发方面做的绝大多数工作都是发生在应用层上的。

传输层

传输层最经常与两个协议，即TCP（传输控制协议）和UDP（用户数据报协议）有关联。这些协议十分有用，但是两者之间在某些重要方面完全不同。

TCP是一种面向连接的协议，其通信是以全双工方式实施的，在双方之间有可靠的流。因为因特网的节点有时会丢失包，所以TCP考虑了这种问题，保证接收方能够按照发送方发送数据的次序接收所有数据。从这方面看，可以认为可靠了；但是在TCP中可能出现中断连接的情况。TCP只能保证是按照发送次序接收数据的。此外TCP连接的两个端点可以同时收发数据；因为是以全双工模式运行的。TCP还负责将从主机应用程序接收到的数据分割为可以在特定链路上传输的独立包。这个包大小叫做最大段长度（MSS），这是在TCP同步时由两个协议栈协商确定的。

UDP是一种无连接协议，被认为能够提供简单报文传递。UDP不能保证包的到达。包甚至可以不按照次序到达。即使存在这个缺点，在数据完整性可以很容易由应用层超时和包重传提供的情况下，UDP仍然被应用层协议经常使用。

端口

通过图1.2可以看到，当包到达时，通过协议栈向上传递，直到最后有一个应用层的PDU（协议数据单元），但是人们怎样才能够知道包实际应该到什么地方呢？传输层包括一个端口（port）的概念，它被用来作为一个具体主机的端点。这样允许具体服务在主机上为自己创建一个地址。例如在HTTP服务器上面，HTTP通常在端口80上运行。当人们发送一个电子邮件时，包在端口25上抵达。端口允许主机提供多种服务。在0到1023范围的端口被叫做众所周知端口，为已知的服务保留。

范围在1024到49151之间的端口叫做已注册端口，可以用于已知服务或者动态分配。最后其余的端口（49152到65535）叫做动态端口（短期端口）。这些端口可以分配给客户机服务。因为从用户角度看，所有通信都发生在端口上，客户机一般都创建一个动态端口，允许与服务器进行通信。

网络层

因特网协议（IP）与网络层有关联。这个协议负责创建将发送到目的主机上的包。目的和源是通过IP地址标识的（本书涉及的所有地址假设为IPv4编址）。一个IP地址是一个4个八位组（字节）结构，惟一地标识连接在因特网上的节点。IP地址历史上被分为5个类，具体确定了网络上的节点分布（参见图1.3）。

这些地址被分成类，以便组成可管理的结构。假设一个新公司需要一个可以连接到因特网上的内部网子网。如果公司中的计算机数量少于254台，使用C类地址块（参见图1.3）将可以满足需要（主机地址使用8位）。如果公司的计算机数量超过254台，但是少于6500台，于是可以使用B类地址块（主机地址使用16位）。最后是A类地址，可以让公司在网络

中拥有$2^{24}-2$台计算机。

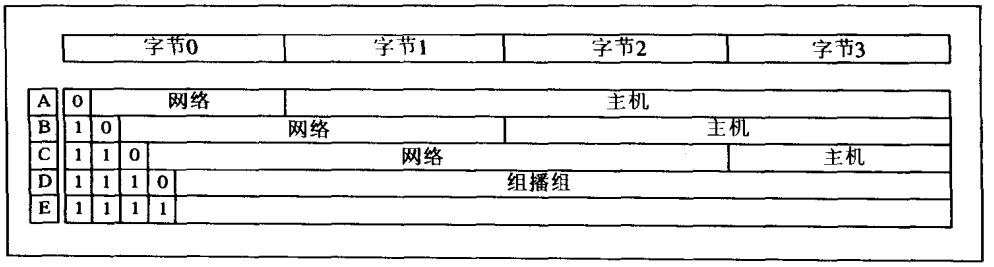


图1.3 因特网地址的类

现在来看一个地址类和产生的IP地址的实例。假定已经分配了一个C类地址块。在C类地址中，有24个位用于网络，8个位用于主机（参见图1.3）。这个类的一个实例范围是从205.15.192.1到205.15.192.254，这就是使用点分十进制方式表示的IP地址，这是十分常见的表示方法（同样的十六进制表示方法是从0xCD0FC001到0xCD0FC0FE）。因为这是一个C类地址，所以网络部分被表示为“205.15.192”，最后的字节表示网络中实际的主机。

从图1.3中可以看出对于A、B、C类来说，这些地址可以划分出网络和主机。“主机”表示主机的最大数量，“网络”表示（该主机的）网络的最大数量。A类提供比较小数量的网络，每个网络有大量的主机，而C类则提供大量的网络，每个网络具有比较小数量的主机。

请注意这个分类体系已经有历史了，并且现在IP地址被看做是无级别的。这意味着一个组织的IP地址的确定是基于一个IP地址和一个网络掩码。网络掩码确定了地址的网络部分。例如假定IP地址为192.168.1.1，网络掩码为26。这里的26表示最左面的连续的位（在本例中为0xFFFFFFC0）或者地址的主机部分为6位。

组播地址的范围为从224.0.0.255到239.255.255.255。范围从244.0.0.0到224.0.0.255的地址被保留用于路由协议和其他拓扑发现协议。一个接口可以配置为接受组播地址（除了自身以外），因此可以接收特定的组播通信量以及与其他组播应用程序通信。

在此有必要说明一下几个特殊的IP地址。地址255.255.255.255（32个“1”位）是一个广播地址。本地网络中的所有节点可以接收发自这个地址的包。地址127.0.0.1是环回地址，用于网络测试（包是通过叫做环回设备的特殊软件环回的）。

在我们继续讨论网络层之前，最后还要提一下IP地址的问题是：网络地址的三个块被用于内部用途。首先对于A类（从10.0.0.0到10.255.255.255），其次是B类（从172.16.0.0到172.31.255.255），最后是C类（从192.168.0.0到192.168.255.255）。这些地址叫做“不可路由的”，可以单独用于内部用途，不必在因特网上通过路由。在很多情况下，机构使用这些地址进行网络地址转换（NAT）或IP伪装，以便在默认网关上转换不可路由地址成为可路由地址。

网络层可以提供一些方法通过网络把包（包括预定目的IP地址）发送到目的地。因为TCP可以负责把包数据发送给主机（从接收方面）中的适当应用程序上，IP可以负责把包通过网络发送给实际主机。

网络层还包括其他一些与IP有关的协议，例如因特网控制报文协议（ICMP）。ICMP用于处理网络上接收到的错误报文，并通知给其他遇到错误的主机。ICMP的最常见应用是回应请求/应答。这些回应通常叫做测试（ping）包。

数据链路层

最后数据链路层提供了主机和网络之间的接口。数据链路层接口有很多形式，例如以太网或者调制解调器使用的串行接口等。谈到以太网络就需要保证与实际硬件（通常叫做MAC驱动）通信的设备驱动程序。以太网设备都是块设备，可以接收完整的以太网PDU，包括数据链路报头和循环冗余校验（CRC）。串行接口也使用设备驱动程序来访问串行设备，同时另一个协议层可以提供PDU封装，以便在串行链路中传输。在此大多数常见的协议有点对点协议（PPP）和串行线路接口协议（SLIP）。两种协议可以提供两台具体设备之间的点对点的链接。

数据链路层必须响应来自物理层的异步事件（进入包），同时管理来自上级层协议的发出包。为此数据链路层通常将栈的其余部分根据独立任务或者根据排队进入和发出数据的中断驱动实体进行划分。

基本的网络配置

现在已经讨论过有关TCP/IP栈的部分基本原理，下面再看一下基本网络配置的要求。

手工解决方案

通过协议栈进行网络通信的主机配置要求使用小的参数集（虽然对于更为复杂的网络，也同样涉及这些配置参数）。

配置中需要的第一个参数是设备的IP地址。IP地址部分是基于设备所连接的具体网络。因此可以或者向网络管理部门请求一个地址，或者配置一个动态地址（本章后面将进一步讨论）。

使用IP地址之后，还需要设置网络掩码，本章前面已经提到。网络掩码顾名思义就是一种掩码，可以用于分辨IP地址的网络部分。再回到IP地址的实例上来，其C类地址将24位用于网络，8位用于主机。其网络掩码将是255.255.255.0（因为使用255.255.255.0掩码其地址205.15.192.16结果将产生地址205.15.192.0，即我们的网络地址）。网络掩码通常可以使用以205.15.192.16/24形式的IP地址标识。

使用IP地址之后，得到一个名称，允许我们与网络上的其他主机对话（实际上它允许其他主机与我们对话）。但是，与其他主机谈什么？与其他节点通信的能力要求更为复杂一点。正如前面已经讨论的情况，这些节点可以在构成主机集的本地网络上面。然而这个网络也可以连接到因特网上，因特网可以提供很多其他主机与我们进行通信。所以问题是，将如何在相对于因特网的其他网络而言的本地网上实现通信？这个问题的解决方案是“路由”，对于这个例子将十分简单。

假定205.15.192.28是希望进行通信的主机地址。人们发现这个地址与其自己的IP地址（205.15.192.16）非常近似。如果使用自己的网络掩码（255.255.255.0）处理两个地址，就

会发现网络是相同的。这意味着这个主机和希望与之对话的主机在同一个本地网络上。然后路由决定通过已经配置其IP地址的接口发送包。然而如果网络不同将发生什么？这将为我们带来下面的参数——默认网关。

默认网关是一种表示所有与其本地网络不匹配包的目的地的IP地址。之所以叫做网关是因为对于其他网络来说是一个网关。所以问题就十分简单了。路由决定或者是本地（基于网络掩码的匹配）或者是全球（基于网络掩码不匹配）。不是所有的配置都如此简单，但是大多数使用单一物理接口的嵌入式系统产品可以使用这种基本结构。

配置的最后部分是针对域名系统（DNS）服务器的另一个IP地址。DNS服务器可以提供查找服务，解析完全合格的域名（如http://www.mynetwork.com）到一个IP地址。需要这么做是因为路由器基于IP地址来路由包，而不是基于名称。

手工IP地址配置的基本部分见表1.2。

表1.2 网络配置基本参数

参数	说明	实例
IP地址	设备的IPv4地址	192.168.1.1
网络掩码	地址的网络掩码	255.255.255.0
默认网关	指示目的地址与本地网络不匹配的包的主机	192.168.1.254
DNS服务器地址	用于映射一个完全合格的域名到一个IP地址的服务器的解析器地址	18.76.100.1

自动解决方案

除了静态配置每个设备（这对于大量设备来说，可能是一场管理恶梦），节点也可以动态配置。动态配置之所以普及是因为这意味着IP地址可以循环使用，并且根据需要进行分配。动态主机配置协议（DHCP）可以用来完成这项任务。一个DHCP客户机与本地网络上的一个DHCP服务器通信，通知其存在，然后接收数据进行配置。

因为当客户机没有配置好自己的网络接口时，通信不能正式进行（传统的鸡和蛋的问题），DHCP客户机将进行广播，把包发给服务器（参见图1.4）。当客户机广播一个发现包（请求配置）时，服务器会回答一个提供（建议配置）。客户机可以通过DHCP请求接受这个配置，DHCP服务器将确认已经完成的协商。这时客户机将使用提供的数据进行配置，过程完成。

对于无法跟踪在以太网中错误使用已经分配IP地址主机的任何人来说，DHCP是对于TCP/IP栈来说是一个非常好的补充。

简单的应用层协议

因为已经讨论了一些基本网络参数及其原则，所以可以使用C语言来编写一些服务器和客户机实现的代码。首先看一下服务器（参见清单1.1）。

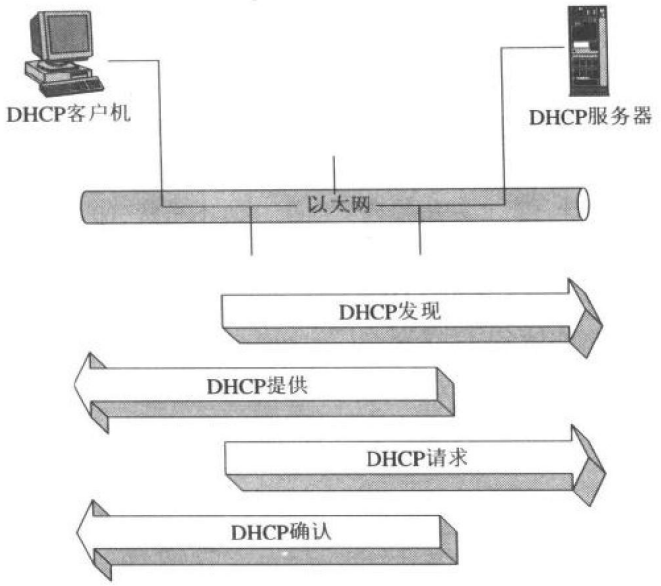


图1.4 DHCP协商

清单1.1 简单的daytime（TCP）服务器

```
int main ( void )
{
  int serverFd, connectionFd;
  struct sockaddr_in servaddr;
  char timebuffer[MAX_BUFFER+1];
  time_t currentTime;

  serverFd = socket(AF_INET, SOCK_STREAM, 0);

  memset(&servaddr, 0, sizeof(servaddr));
  servaddr.sin_family = AF_INET;
  servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
  servaddr.sin_port = htons(DAYTIME_SERVER_PORT);

  bind(serverFd, (struct sockaddr *)&servaddr, sizeof(servaddr));

  listen(serverFd, 5);

  while ( 1 ) {

    connectionFd = accept(serverFd,
                          (struct sockaddr *)NULL, NULL);

    if (connectionFd >= 0) {

      currentTime = time(NULL);
      snprintf(timebuffer,
               MAX_BUFFER, "%s\n", ctime(&currentTime));

      write(connectionFd, timebuffer, strlen(timebuffer));
```