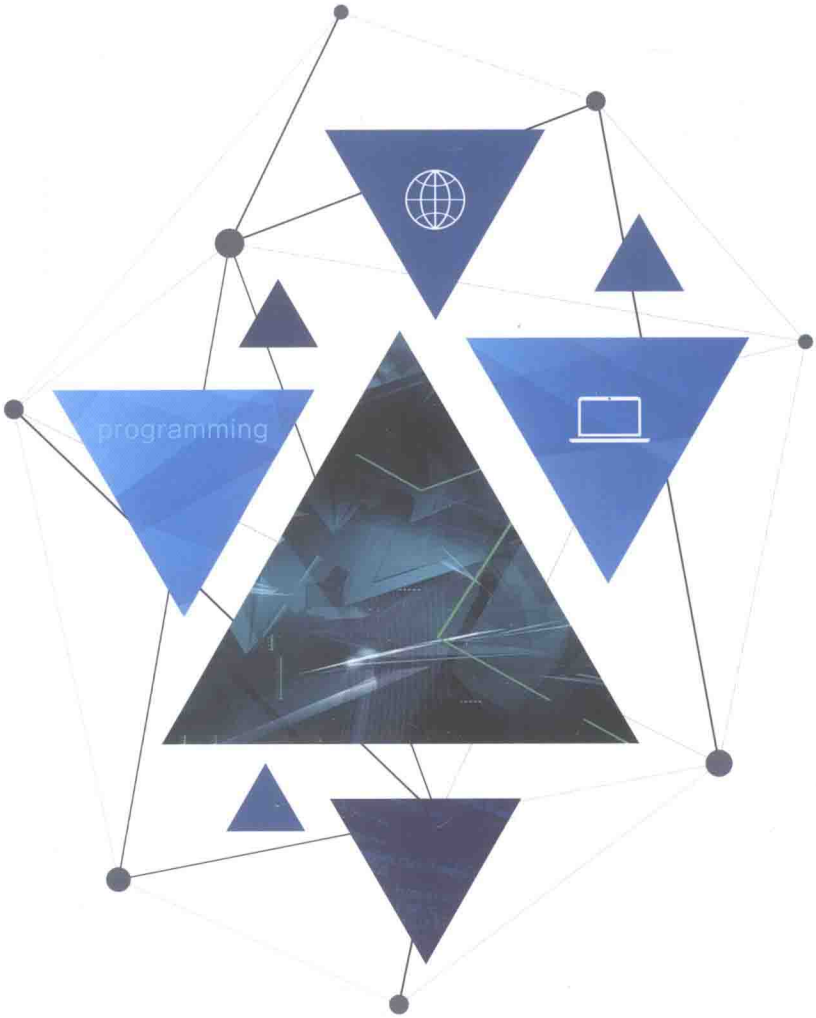
TCP/IP网络编程基础教程

主 编 王 雷



北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

TCP/IP网络编程基础教程

王 雷 主编

北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本书是一本基于 TCP/IP 协议进行计算机网络编程方面的教科书，全书主要内容包括基于套接字的 TCP/IP 网络通信原理与模型、循环服务器软件的实现原理与方法、服务器与客户进程中的并发机制、多进程并发机制的实现原理与方法、多线程并发 TCP 服务器软件的实现原理与方法、单线程并发机制的实现原理与方法、基于 POOL 和 EPOLL 的并发机制与实现方法、客户/服务器系统中的死锁问题八章内容以及 GCC 编译器简介、课程实验两个附录。全书通过原理介绍与例程剖析的形式，系统介绍了 UNIX/Linux 与 Windows 环境下如何使用 C 语言基于 TCP/IP 协议与 Socket API 进行网络编程的详细步骤与过程。

与同类教材相比，本书主要的特点为：在注重阐述 TCP/IP 网络通信原理与套接字 API 编程原理的基础上，通过对例程的深入剖析，深入浅出地介绍服务器与客户软件的编程技巧，同时，在章节的编排上更加富有衔接性。全书内容按照 TCP/IP 网络通信原理→循环服务器软件设计→并发服务器软件设计→客户/服务器系统中的死锁问题→编译环境→课程实验的顺序，通过 C 语言例程剖析，由浅入深地介绍了基于 TCP/IP 协议进行网络编程的原理与方法。通过以上连贯的章节编排，读者能够更加简洁、系统地掌握网络编程技术。

本书特别适合网络工程、计算机科学技术与通信工程等专业的本、专科学生和从事计算机网络编程的技术人员，同时也可供其他专业的学生、计算机网络技术爱好者，以及计算机应用技术相关的工程技术人员参考。

图书在版编目（CIP）数据

TCP/IP 网络编程基础教程/王雷主编. —北京：北京理工大学出版社，2017.2
ISBN 978-7-5682-3762-8

Ⅰ. ①T…　Ⅱ. ①王…　Ⅲ. ①计算机网络－通信协议－教材②计算机网络－程序设计－教材　Ⅳ. ①TN915.04②TP393.09

中国版本图书馆 CIP 数据核字（2017）第 039228 号

出版发行 / 北京理工大学出版社有限责任公司
社　　址 / 北京市海淀区中关村南大街 5 号
邮　　编 / 100081
电　　话 /（010）68914775（总编室）
（010）82562903（教材售后服务热线）
（010）68948351（其他图书服务热线）
网　　址 / http://www.bitpress.com.cn
经　　销 / 全国各地新华书店
印　　刷 / 三河市华骏印务包装有限公司
开　　本 / 787 毫米×1092 毫米　1/16
印　　张 / 18
字　　数 / 424 千字
版　　次 / 2017 年 2 月第 1 版　2017 年 2 月第 1 次印刷
定　　价 /59.00 元

责任编辑 / 李秀梅
文案编辑 / 杜春英
责任校对 / 周瑞红
责任印制 / 施胜娟

前　言

背景动机

随着 Internet 在全球范围内的迅速普及，网络对人们的学习、工作、生活以及对社会的影响越来越大。网络技术被誉为“近代最深刻的技术革命”，人们用“网络时代”“网络经济”等术语来描述网络对社会信息化与经济发展的巨大影响。目前，国内各主要高校的计算机应用技术与软件工程专业的大学生均开设了 TCP/IP 网络编程课程，但传统的教材缺乏对所给例程的深入剖析，导致初学者在采用这些教材进行学习时难以轻松掌握所学内容。为此，本书作者在多年讲授 TCP/IP 网络编程技术的基础上，在传统教材所介绍的 Socket 网络编程相关概念与技术的基础之上，进行了大幅度的内容增减与结构调整。同时，为了使不同层次的读者均能够更加方便地掌握所学内容，本书针对各章节中所给出的例程新增了全面深入的剖析，还新增了对 GCC 编译器的有关介绍与课程实验，使得全书内容更加完整。

目标读者

本书的目标读者包括计算机相关专业的专科生、本科生、研究生，计算机网络编程技术与 C 语言的爱好者，以及计算机应用技术相关的工程技术人员。

组织结构

考虑到读者在阅读本书之前对计算机网络编程技术的了解程度不尽相同，本书主要分为以下四个大的部分。

1）第 1 章为第一部分，主要介绍了 TCP/IP 网络通信原理与套接字 API 编程的基本原理与方法。

2）第 2～7 章为第二部分，其中，第 2 章主要介绍循环服务器软件的实现原理与方法，第 3 章主要介绍服务器与客户进程中的并发机制，第 4～7 章则主要介绍多种不同并发服务器的实现原理与方法。

3）第 8 章为第三部分，主要介绍了客户/服务器系统中存在的死锁问题及其解决方法。

4）附录 A 和附录 B 为第四部分，其中，附录 A 主要介绍了 GCC 编译器的安装与使用方法，附录 B 则针对全书的四个核心知识点分别给出了四个不同的课程实验项目。

编　者

湘潭大学信息工程学院

致 谢

首先，在本书的编写过程中，得到了湘潭大学信息工程学院博士生导师郑金华教授、刘任任教授、裴廷睿教授等领导和专家们的大力支持与热心帮助，在此表示衷心感谢。其次，本书的出版还得到国家自然科学基金项目（No.61640210，No.61672447）、湘潭大学教学改革研究项目（No.1129|2904101）、赛尔网络下一代互联网技术创新项目（No.NGII20160305）、湖南省重点学科建设项目、湘潭大学“产学研提质专项”资金支持项目（No.11KZ|KZ03051）、以及湖南省物联网学会华为基金项目（No.11KH|KH01116）等部分资助；本书的部分内容参考了国内外有关单位和个人的研究成果，均已经在参考文献中列出，在此一并表示感谢。

另外，由于本书的编写目的定位于 LINUX/WINDOWS 环境下的 C 语言 Socket 编程的基础知识与案例分析相结合，试图让本科生与研究生在深入了解 LINUX/WINDOWS 环境下的 C 语言 Socket 编程的相关概念与关键技术的基础上，能尝试开展 LINUX/WINDOWS 环境下的 C 语言 Socket 编程的一些初步编程工作，因此，在本书的内容编写与结构组织上具有一定的难度，加之编著者水平有限，虽然几经修改，但书中仍然会难免存在一些疏漏与不足之处，敬请读者、专家、以及同行朋友们的批评指正，在此先行表示感谢。

CONTENTS 目录

第1章 基于套接字的 TCP/IP 网络通信原理与模型

TCP/IP 协议是实现网络通信的基础，本章将在简要介绍 TCP/IP 参考模型与通信原理的基础之上，首先系统介绍套接字的基本概念以及基于套接字的 TCP/IP 网络通信实现流程，然后，再分别针对 UNIX/Linux 与 Windows 两种不同环境，详细介绍 BSD UNIX 套接字 API 中提供的主要系统函数与 Windows 套接字 API 中提供的主要扩展系统函数，并在此基础上分别针对 UNIX/Linux 与 Windows 两种不同环境，深入分析基于不同套接字类型的 TCP/IP 网络通信模型及其 C 语言实现方法。

1.1 TCP/IP 协议概述

1.1.1 TCP/IP 参考模型

TCP/IP（Transmission Control Protocol/Internet Protocol），即传输控制协议/因特网协议，是一个由多种协议组成的协议族（Protocol Family），定义了计算机通过网络互相通信及协议族各层次之间通信的规范。TCP/IP 参考模型是一个抽象的分层模型，在该模型中，属于 TCP/IP 协议族的所有网络协议都被归类到以下四个抽象的“层”之中。

1）主机-网络层（Host to Network Layer）：主机-网络层是 TCP/IP 参考模型的最低层，也称为网络接口层，它主要负责接收从互联网络层交来的 IP 数据报并将其通过低层物理网络发送出去，或者从低层物理网络上接收物理帧并从中抽出 IP 数据报交给互联网络层。其中，网络接口主要有以下两种类型：第一种是设备驱动程序，如局域网的网络接口；第二种是含自身数据链路协议的复杂子系统。在 TCP/IP 参考模型中未定义数据链路层，这主要是因为在 TCP/IP 最初的设计中已经使其可以使用各种典型的数据链路层协议。

2）互联网络层（Internet Layer）：也称为网际互联层或 IP 层，主要负责将源主机的报文分组发送到目的主机，源主机与目的主机可以在一个网络上，也可以在不同的网络上。由于 TCP/IP 参考模型中网络层协议是 IP（Internet Protocol）协议，因此互联网络层也称为 IP 层。其中，IP 协议是一种不可靠、无连接的数据报传送服务的协议，它提供的是一种“尽力

而为（Best Effort）”的服务。IP 协议的协议数据单元是 IP 分组，由于在 IP 层提供数据报服务，因此也常将 IP 分组称为 IP 数据报。

3）传输层（Transport Layer）：传输层主要负责在互联网中源主机与目的主机的对等进程实体之间提供可靠的端到端的数据传输。在 TCP/IP 参考模型的传输层中定义了以下两种协议。

TCP 协议（Transmission Control Protocol，传输控制协议）：TCP 协议是一种可靠的面向连接的传输层协议，它允许将一台主机的字节流（Byte Stream）无差错地传送到目的主机。在通信过程中，TCP 协议首先将应用层的字节流分成多个字节段（Byte Segment），然后再将一个个字节段传送到互联网络层，并最终发送到目的主机。当互联网络层将接收到的字节段传送给传输层时，传输层再将这些字节段还原成原始的字节流，并传送到应用层。TCP 协议同时要完成流量控制功能，协调收发双方的发送与接收速度，以达到正确传输的目的。

UDP 协议（User Datagram Protocol，用户数据报协议）：UDP 协议是一种不可靠的无连接的传输层协议，它主要用于不要求分组顺序到达的传输服务之中，在基于 UDP 协议的传输服务中，分组的传输顺序检查与排序将由应用层完成。UDP 协议主要面向请求/应答式的交易型应用，一次交易往往只有一来一回两次报文交换，假如为此而建立和撤销连接将导致网络开销过大，因此，在这种情况下使用 UDP 就非常有效。另外， UDP 协议也常用于那些对可靠性要求不高，但要求网络的延迟较小的场合，如话音和视频数据的传送等。

4）应用层（Application Layer）：应用层包括所有的高层协议，目前 TCP/IP 参考模型中的应用层协议主要包括以下几种：

- 网络终端协议 Telnet。
- 文件传输协议 FTP（File Transfer Protocol）。
- 简单邮件传输协议 SMTP（Simple Mail Transfer Protocol）。
- 域名系统 DNS（Domain Name System）。
- 简单网络管理协议 SNMP（Simple Network Management Protocol）。
- 超文本传输协议 HTTP（Hyper Text Transfer Protocol）。

1.1.2 TCP/IP 网络通信中的客户-服务器模型

如图 1.1 所示，在 TCP/IP 协议体系中，进程之间的相互作用采用客户/服务器（Client/Server，简称 C/S）模型，其中，客户与服务器分别表示相互通信的两个应用程序进程。在 C/S 模型中，是根据通信发起的方向来区别一个应用程序进程是客户还是服务器的。一般将发起通信的应用程序进程称为客户，而将负责等待接收客户通信请求并为客户提供服务的应用程序进程称为服务器。

1.1.3 TCP/IP 参考模型的通信原理

TCP/IP 参考模型的通信原理如图 1.2 所示，其中，一至二层为串联的，而三至四层则是端到端（End to End）的。由图 1.2 可知，网际互联层与网络接口层实现了计算机网络中处

于不同位置的主机之间的数据通信，但是数据通信不是计算机网络的最终目的，计算机网络最本质的活动是实现分布在不同地理位置的主机之间的进程间通信（InterProcess Communication，IPC），以实现各种网络服务功能。而设置传输层的主要目的就是实现上述这种分布式进程之间的通信功能。

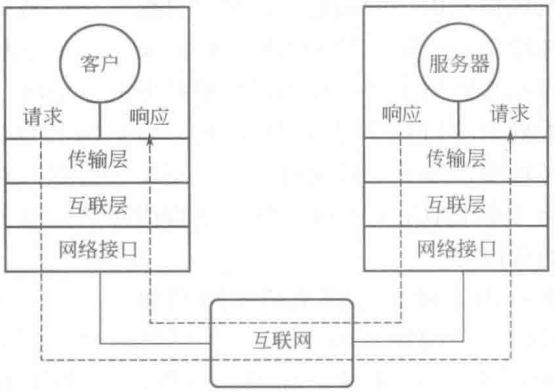


图 1.1 客户/服务器通信模型

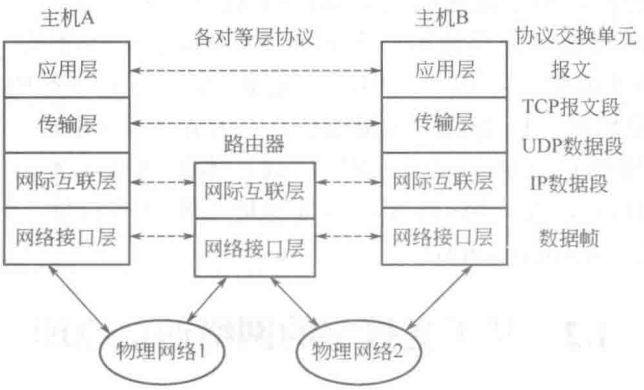


图 1.2 TCP/IP 参考模型的通信原理

在单机系统中，由于每个进程都在自己的地址范围内运行，为保证两个相互通信的进程之间既互不干扰又协调一致工作，Linux 操作系统为进程之间的通信提供了相应的设施，如管道（Pipe）、命名管道（Named Pipe）、软中断信号（Signal）和信号量（Semaphore）等，但上述这些设施都仅限于用在本机进程之间的通信。而网络环境下的分布式进程间通信要解决的是不同主机进程间的相互通信问题（显然，同机进程间通信只是其中的一个特例）。为此，传输层需要解决在网络环境下分布式进程间通信所面临的以下两个方面的问题。

1）进程的命名与寻址：按照 TCP/IP 参考模型的通信原理描述，传输层的主要目的是要实现网络环境下分布式进程之间的通信功能，显然，从这个意义上来讲，网络通信的最终地址除了主机地址之外，还需要包括可以描述进程的某种标识符。为此，TCP/IP 参考模型提出了协议端口（Protocol Port，简称端口）的概念，用于标识通信的进程。其中，端口是一种抽象的软件结构（包括一些数据结构和 I/O 缓冲区）。应用程序（即进程）在通过系统调用与某个端口建立起了连接之后，传输层传给该端口的所有数据均可被其接收，同理，其发

给传输层的数据也均可通过该端口进行输出。在 TCP/IP 协议的实现中，端口操作类似于一般的文件 I/O 操作。为此，与文件描述符类似，每个端口均拥有一个唯一的被称为端口号（Port Number）的 16 位无符号整数型标识符，范围是 0～65535，用于区别不同的端口。

端口号一般有以下两种基本的分配方式：第一种为全局分配，这是一种集中分配方式，由一个公认权威的中央机构根据用户的需要进行统一分配，并将结果公布于众，通过该方式分配的端口号也称为熟知端口号；第二种为本地分配，又称动态分配，是当进程需要访问传输层服务时，向本地操作系统提出申请，再由操作系统分配本地唯一的端口号。由于同一台机器上的不同进程所分配到的端口号不同，因此，同一台机器上的不同进程就可以用端口号来唯一标识。但在网络环境中，显然，若要标识一个完整的进程，除了端口号之外，还需使用到本地主机的 IP 地址（本地地址）来唯一标识进程所在的本地主机（这是因为不同机器上的进程可以拥有相同的端口号）。

2）多重协议的识别：由于操作系统支持的网络协议众多，不同协议的工作方式与地址格式均不相同，因此，在网络环境下，一个应用程序进程最终需要使用一个三元组<协议，本地地址，本地端口号>来唯一标识。另外，在 TCP/IP 网络环境下，一个完整的网间通信需要由两个进程完成，并且这两个进程之间只能使用相同的传输层协议才能进行通信，也就是说，不可能通信的一端使用 TCP 协议，而另一端使用 UDP 协议。因此，一个完整的网间通信需要使用一个五元组<协议，本地地址，本地端口号，远程地址，远程端口号>才能唯一标识。其中，二元组<本地地址，本地端口号>称为网间进程通信中的本地端点地址（Endpoint Address），二元组<远程地址，远程端口号>称为网间进程通信中的远程端点地址，而三元组<协议，本地地址，本地端口号>称为一个半相关（Half-Association），五元组<协议，本地地址，本地端口号，远程地址，远程端口号>则称为一个相关（Association）。

1.2 基于套接字的网络通信原理

1.2.1 套接字概述

所谓套接字（Socket），就是对网络中不同主机上的应用进程之间进行双向通信的端点的抽象。一个套接字就是网络上进程通信的一端，提供了应用层进程利用网络协议栈交换数据的机制。如图 1.3 所示，从所处的地位来讲，套接字上联应用进程，下联网络协议栈，是应用程序通过网络协议栈进行通信的接口，是应用程序与网络协议栈进行交互的接口。

套接字是实现网络通信的基石，在采用基于套接字的网络通信过程中，其通信原理如下图 1.4 所示：当主机 A 上的应用程序进程 A 需要和主机 B 上的应用程序进程 B 进行通信时，主机 A 上的应用程序进程 A 首先将一段信息写入其在本地主机 A 上的 Socket A，然后再由 Socket A 将该段信息通过 TCP/IP 网络发送到主机 B 上应用程序进程 B 所对应的 Socket B 之中，最后再由 Socket B 将该段信息传送给应用程序进程 B。

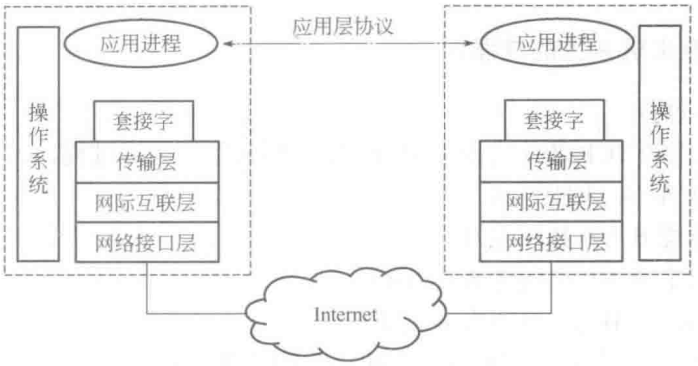


图 1.3 套接字通信模型

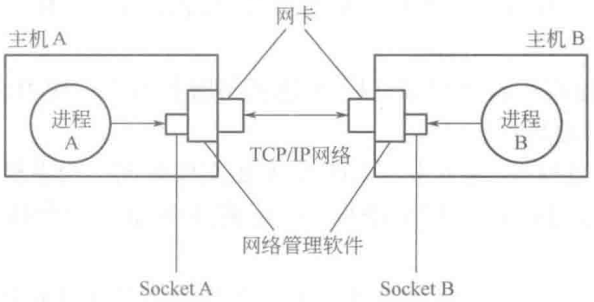


图 1.4 套接字通信原理示例

由以上描述可知，一个套接字可以看成应用程序进程进行网间通信的端点。而在网络环境下，一个应用程序进程又通常可用一个半相关<协议，本地地址，本地端口号>来进行唯一标识，因此，一个套接字显然也可以用上述半相关<协议，本地地址，本地端口号>来进行唯一标识，其中，二元组<本地地址，本地端口号>通常也称为套接字的端点地址。

显然，一个完整的 TCP/IP 网络通信连接可用通信双方（客户端和服务器端）所对应的套接字组成的套接字对（Socket Pair）来唯一标识，其中，通常将运行于客户端的套接字称为客户端套接字（Client Socket），而将运行于服务器端的套接字称为服务器端套接字（Server Socket）。

在实际网络通信中，将由客户端套接字提出连接请求，而要连接的目标就是服务器端套接字。为此，通常又将客户端套接字称为主动套接字，将服务器端套接字称为被动套接字。此外，由于客户端套接字在向服务器端套接字提出连接请求之前，首先必须知道服务器端套接字的端点地址（即服务器的 IP 地址和服务器进程的端口号），为此，服务器套接字的端点地址必须预先被客户端知道，也就意味着，服务器端套接字必须使用熟知（Well-Known）端口号。

1.2.2 基于套接字的 TCP/IP 网络通信原理

为了形象地说明基于套接字的 TCP/IP 网络通信原理，本节以图 1.4 中所示的客户端主机 A 上的进程 A 与服务器端主机 B 上的进程 B 之间的网络通信过程为例来进行阐述。

1. 针对客户端主机 A 上的进程 A

（1）TCP 通信过程

进程 A、B 之间的 TCP 通信过程类似 A、B 之间的手机通话过程，其中：

- 客户端进程 A ≌ 主叫方 A。
- 服务器端进程 B ≌ 被叫方 B。
- 客户端套接字 A ≌ 主叫方 A 的手机。
- 服务器端套接字 B ≌ 被叫方 B 的手机。
- 客户端套接字 A 的端点地址 ≌ 主叫方 A 的手机号码。
- 服务器端套接字 B 的端点地址 ≌ 被叫方 B 的手机号码。

步骤 1（手机通话过程）：若主叫方 A 想要打电话给被叫方 B，首先需要知道被叫方 B 的手机号码。

对应的 TCP 通信过程：若客户端进程 A 想要与服务器端进程 B 通信，首先需要知道服务器端套接字 B 的端点地址。

步骤 2（手机通话过程）：接下来，主叫方 A 还需要新购一台手机 A。

对应的 TCP 通信过程：客户端进程 A 还需要新建一个套接字 A，即客户端套接字 A。

步骤 3（手机通话过程）：主叫方 A 还需要为新购的手机 A 新申请一个本地手机号码。

对应的 TCP 通信过程：客户端进程 A 还需要新建的客户端套接字 A 申请一个本地端点地址。

步骤 4（手机通话过程）：主叫方 A 利用手机 A 拨打被叫方 B 的手机 B。

对应的 TCP 通信过程：客户端进程 A 利用客户端套接字 A 向服务器端套接字 B 发送 TCP 连接建立请求。

步骤 5（手机通话过程）：主叫方 A 拨通被叫方 B 的电话之后，与被叫方 B 之间进行手机通话。

对应的 TCP 通信过程：客户端进程 A 与服务器端进程 B 在建立了 TCP 连接之后，与服务器端进程 B 之间进行 TCP 通信。

步骤 6（手机通话过程）：通话结束后，主叫方 A 挂机。

对应的 TCP 通信过程：通信结束后，客户端进程 A 关闭客户端套接字 A 以释放 TCP 连接以及与套接字相关的资源。

（2）UDP 通信过程

进程 A、B 之间的 UDP 通信过程类似于 A、B 之间的短信收发过程，其中：

- 客户端进程 A ≌ 发信方 A。
- 服务器端进程 B ≌ 收信方 B。
- 客户端套接字 A ≌ 发信方 A 的手机。
- 服务器端套接字 B ≌ 收信方 B 的手机。
- 客户端套接字 A 的端点地址 ≌ 发信方 A 的手机号码。
- 服务器端套接字 B 的端点地址 ≌ 收信方 B 的手机号码。

步骤 1（短信收发过程）：若发信方 A 想要发送短信给收信方 B，首先需要知道收信方 B 的手机号码。