

**2.2 TCP 发展期**  
1980 年—1995 年 ..... 48

    拥塞崩溃问题浮出水面（1980 年）..... 49

    引入 Nagle 算法（1984 年）..... 49

    引入拥塞控制算法（1988 年）..... 51

    往互联网的迁移与万维网的诞生（1990 年）..... 52

**2.3 TCP 普及期**  
1995 年—2006 年 ..... 53

    Windows 95 发售（1995 年）..... 54

    IPv6 投入使用（1999 年）..... 54

    无线 LAN 出现（1999 年）..... 55

    各式各样的互联网应用服务（2004 年—2006 年）..... 56

**2.4 TCP 扩展期**  
21 世纪 00 年代后半期— ..... 57

    智能手机普及（2007 年）..... 57

    云计算出现（2006 年）..... 58

    移动网络的高速化（2010 年、2015 年）..... 59

    物联网的大众化（2015 年）..... 61

**2.5 小结**..... 62

第 **3** 章

**TCP 与数据传输**

实现可靠性与效率的兼顾 ..... 65

**3.1 TCP 的数据格式**  
数据包与首部的格式 ..... 66

    数据包格式 ..... 66

    TCP 报文段 ..... 66

    TCP 首部格式 ..... 68

    UDP 首部格式 ..... 72

<b>3.2</b>	<b>连接管理</b>	
	3 次握手 .....	73
	建立连接 .....	73
	断开连接 .....	74
	端口与连接 .....	75
<b>3.3</b>	<b>流量控制与窗口控制</b>	
	不宜多也不宜少, 适当的发送量与接收方缓冲区 .....	76
	流量控制 .....	76
	缓存与时延 .....	77
	窗口控制 .....	78
	复习: 流量控制、窗口控制和拥塞控制 .....	79
<b>3.4</b>	<b>拥塞控制</b>	
	预测传输量, 预测自律运行且内部宛如黑盒的网络的内部情况 .....	80
	TCP 拥塞控制的基本概念 .....	80
	慢启动 .....	81
	拥塞避免 .....	82
	快速恢复 .....	84
<b>3.5</b>	<b>重传控制</b>	
	高可靠性传输的关键——准确且高效 .....	86
	高可靠性传输所需的重传控制 .....	86
	❶ 基于重传计时器的超时控制 .....	87
	❷ 使用重复 ACK .....	91
	拥塞避免算法与重传控制综合影响下的流程及拥塞窗口大小的变化情况 .....	92
<b>3.6</b>	<b>TCP 初期的代表性拥塞控制算法</b>	
	Tahoe、Reno、NewReno 和 Vegas .....	93
	拥塞控制算法的变化 .....	93
	Tahoe .....	94
	Reno .....	96
	NewReno .....	97
	Vegas .....	99
<b>3.7</b>	<b>小结</b>	101

## 第4章

# 程序员必学的拥塞控制算法

逐渐增长的通信数据量与网络的变化..... 103

### 4.1 拥塞控制的基本理论

目的与设计，计算公式的基础知识..... 104

拥塞控制的目的..... 104

拥塞控制的基本设计..... 105

拥塞控制中的有限状态机..... 107

拥塞控制算法示例..... 109

### 4.2 拥塞控制算法

通过理论 × 模拟加深理解..... 111

本书介绍的拥塞控制算法..... 111

NewReno..... 112

Vegas..... 115

Westwood..... 117

HighSpeed..... 119

Scalable..... 121

Veno..... 123

BIC..... 125

H-TCP..... 127

Hybla..... 129

Illinois..... 130

YeAH..... 133

### 4.3 协议分析器 Wireshark 实践入门

拥塞控制算法的观察 ①..... 135

什么是 Wireshark..... 135

Wireshark 的环境搭建..... 135

使用 Wireshark 进行 TCP 首部分析..... 138

通过 Wireshark 观察拥塞控制算法..... 141

<b>4.4</b>	加深理解：网络模拟器 ns-3 入门	
	拥塞控制算法的观察 ②	149
	ns-3 的基本情况	149
	搭建 ns-3 环境	150
	基于 ns-3 的网络模拟的基础知识	151
	脚本文件 chapter4-base.cc	153
	使用 Python 运行模拟器并进行分析和可视化	155
<b>4.5</b>	小结	163

## 第 5 章

# CUBIC 算法

通过三次函数简单地解决问题	167
---------------	-----

<b>5.1</b>	网络高速化与 TCP 拥塞控制	
	长肥管道带来的变化	168
	Reno 和 NewReno	168
	快速恢复	168
	网络的高速化与长肥管道	169
	端到端之间的三大时延	170
	长肥管道下 NewReno 的新问题	171
<b>5.2</b>	基于丢包的拥塞控制	
	以丢包情况为指标的一种历史悠久的方法	173
	基于丢包的拥塞控制算法的基本情况	174
	AIMD 控制	174
	[ 实测 ] NewReno 的拥塞窗口大小的变化情况	176
	HighSpeed 与 Scalable	179
	亲和性	183
	RTT 公平性	184
<b>5.3</b>	BIC	
	以宽带、高时延环境为前提的算法	186
	BIC 是什么	186



增大拥塞窗口大小的两个阶段 .....	187
BIC 的拥塞窗口大小的变化情况 .....	188
BIC 的问题 .....	190
<b>5.4 CUBIC 的机制</b>	
使用三次函数大幅简化拥塞窗口大小控制算法 .....	190
CUBIC 的基本情况 .....	190
窗口控制算法的关键点 .....	191
CUBIC 的拥塞窗口大小的变化情况 .....	192
模拟结果中展现出来的高亲和性 .....	194
模拟结果中展现出来的 RTT 公平性 .....	194
窄带、低时延环境下的适应性 .....	195
CUBIC 的问题 .....	197
<b>5.5 使用伪代码学习 CUBIC 算法</b>	
主要的行为与处理过程 .....	198
初始化 .....	198
收到 ACK 时的行为 .....	199
丢包时的行为 .....	199
超时时的行为 .....	200
主要的函数与处理 .....	200
<b>5.6 小结</b> .....	202

## 第 6 章

# BBR 算法

检测吞吐量与 RTT 的值, 调节数据发送量 .....	205
------------------------------	-----

<b>6.1 缓冲区增大与缓冲区时延增大</b>	
存储成本下降的影响 .....	206
网络设备的缓冲区增大 .....	206
缓冲区膨胀 .....	207
基于丢包的拥塞控制与缓冲区膨胀的关系 .....	208

缓冲区增大给 CUBIC 带来的影响 .....	210
<b>6.2 基于延迟的拥塞控制</b>	
以 RTT 为指标的算法的基本情况和 Vegas 示例 .....	212
3 种拥塞控制算法和如何结合环境选择算法 .....	212
基于延迟的拥塞控制的基本设计思路 .....	213
Vegas 的拥塞窗口大小的变化情况 .....	214
过去的基于延迟控制的问题 .....	215
<b>6.3 BBR 的机制</b>	
把控数据发送量与 RTT 之间的关系，实现最大吞吐量 .....	217
BBR 的基本思路 .....	217
BBR 的拥塞窗口大小控制机制 .....	218
RTprop 的估算 .....	219
BtlBw 的估算 .....	220
<b>6.4 使用伪代码学习 BBR 算法</b>	
收到 ACK 时和发送数据时 .....	221
收到 ACK 时 .....	222
发送数据时 .....	222
<b>6.5 BBR 的流程</b>	
模拟实验中的各种流程 .....	223
只有 BBR 网络流时的表现 .....	223
当多个 BBR 网络流同时存在时 .....	225
与 CUBIC 的共存 .....	227
长肥管道下的表现 .....	229
<b>6.6 小结</b> .....	231

# 第7章

## TCP前沿的研究动向

应用程序和通信环境一旦变化,TCP也会变化 ..... 233

### 7.1 TCP 周边环境的变化

3 个视角: 通信方式、通信设备和连接目标 ..... 234

TCP 迄今为止的发展情况 ..... 234

观察通信环境变化的 3 个视角 ..... 235

通信方式的变化 ..... 236

通信设备的多样化 ..... 238

连接目标的变化 ..... 239

小结 ..... 241

### 7.2 5G ( 第 5 代移动通信 )

移动通信的大容量化、多设备支持、高可靠性与低时延 ..... 241

[ 背景 ] 5G 的应用场景与走向实用的规划 ..... 242

[ 问题 ] 如何应对严苛的需求条件 ..... 244

[TCP 相关动向 ❶] 毫米波段的处理 ..... 245

[TCP 相关动向 ❷] 多路径 TCP ..... 246

[TCP 相关动向 ❸] 高清流媒体 ..... 248

### 7.3 物联网

通过互联网控制各种各样的设备 ..... 249

[ 背景 ] 多样的设备和通信方式 ..... 249

[ 问题 ] 处理能力和通信环境上的制约 ..... 252

[TCP 相关动向] 适配物联网 ..... 253

### 7.4 数据中心

大规模化与各种需求条件并存 ..... 254

[ 背景 ] 云服务的普及和数据中心的大规模化 ..... 254

[ 问题 ] 针对缓冲区的互斥的需求条件 ..... 256

[TCP 相关动向] 面向数据中心的拥塞控制 ..... 257

<b>7.5</b>	<b>自动驾驶</b>	
	追求高可靠性与低时延、大容量的通信性能 .....	259
	[ 背景 ] 以普及自动驾驶为目的的技术 .....	259
	[ 问题 ] 高速移动时的高可靠性通信 .....	262
	[ 与 TCP 的关系 ] 关于确保可靠性 .....	263
<b>7.6</b>	<b>小结</b> .....	264



# 第 1 章

## TCP 入门

### 确保传输可靠性

互联网是全世界拥有通信功能的设备互相联通所构成的网络。所有设备并非各行其是地随意运作，而是遵循着同一套规则。这套规则称为协议，是全世界通用的标准。

通信是靠多个协议分层运作而实现的。TCP 是其中一个协议，主要承担“确保传输可靠性”的重要职责。

本章将首先概述实现网络通信的各个协议，明确传输层的职责和特征，然后介绍本书的主题之一——TCP 的基本功能。

## 1.1

### 通信与协议

#### OSI 参考模型、TCP/IP 和 RFC

所谓协议，其实是多种多样的。根据层级选择相应的协议，便能按照应用程序的要求实现通信。

本节将概述通信协议的总体情况。

#### OSI 参考模型

设备间的通信方式，其实和人与人之间的交流方式基本一致。举例来说，听不懂方言，双方便无法沟通，但如果使用普通话，双方就可以沟通（图 1.1）。只要全世界的交流语言互通，不因国家和地区而不同，那么全世界的人们就可以无障碍交流。

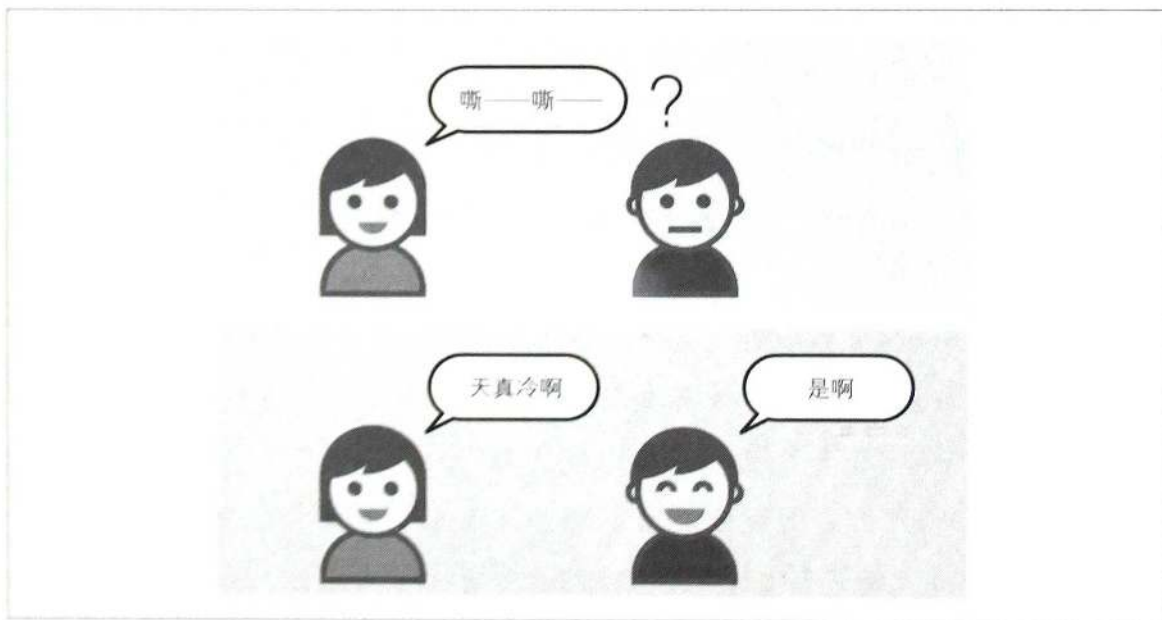


图 1.1 使用互通的语言交流

语言互通靠普通话，而通信设备互通，靠的是 OSI 参考模型。OSI 是 Open System Interconnection（开放式系统互连）的简称。OSI 参考模型由国际标准化组织（International Organization for Standardization, ISO）制定，

它支持互通的功能分层设计，能够使不同的设备具备相互通信的能力。

将协议分层后，软件开发者就只需针对具体的层所负责的功能，专注开发其专有逻辑即可。如此一来，不仅降低了实现难度，同时也将责任划分得更为具体。下面，我们先简单介绍一下各层的职能。

### ——[第7层]应用层

应用层主要定义各个应用程序中使用的通信协议。例如，实现网页浏览的 HTTP（Hypertext Transfer Protocol，超文本传输协议）、实现文件下载的 FTP（File Transfer Protocol，文件传输协议）、实现 IP 地址自动分配的 DHCP（Dynamic Host Configuration Protocol，动态主机配置协议）、实现互联网域名与 IP 地址关系对应的 DNS（Domain Name System，域名系统）、实现网络设备时间同步的 NTP（Network Time Protocol，网络时间协议）、实现电子邮件收发的 SMTP（Simple Mail Transfer Protocol，简单邮件传输协议）和 POP（Post Office Protocol，邮局协议），以及实现远程计算机操作的 Telnet 等。

举例来说，在 HTTP 协议中，客户端计算机的 Web 浏览器为了获取 Web 服务器上的 HTML（HyperText Markup Language，超文本标记语言）文件，会发出请求（GET 请求），而服务器则会返回响应内容，最后客户端完成 HTML 文件、样式表和图像数据等的下载（图 1.2）。

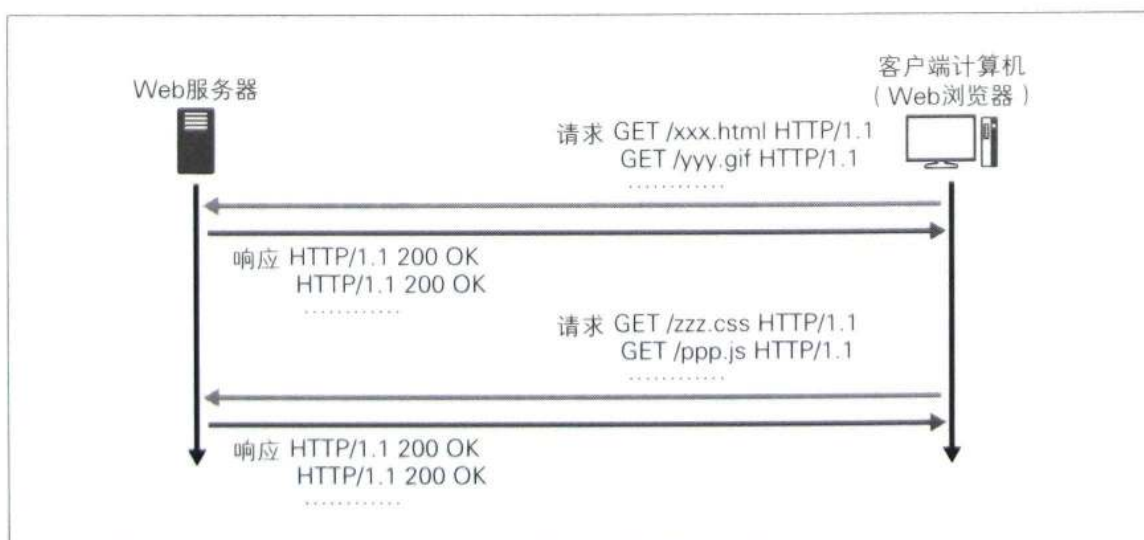


图 1.2 HTTP 通信示例



## ——[第6层]表示层

表示的英文是 presentation，它的意思是表达、表现，指的是一种向对方传递信息的方法。字符的编码方式，图像或视频的压缩方式，以及数据的加密方式种类繁多，表示层所负责的正是将应用程序中这些特定的数据格式转换为通信设备间可以互相理解的、可在网络上互通的格式。

例如，不同的应用程序使用的字符编码方式各不相同，具体有 UTF-8、UTF-16 和 GBK 等。将它们转换为可在网络上互通的数据格式，便可实现编码方式各异的应用程序间的数据通信。

其他的协议包括图像压缩格式 JPEG（Joint Photographic Experts Group，联合图像专家组）、视频压缩格式 MPEG（Moving Picture Experts Group，动态图像专家组）和音乐文件格式 MIDI（Musical Instrument Digital Interface，乐器数字接口）等。

## ——[第5层]会话层

会话的英文是 session，通常是指用于管理数据通信从开始到结束整个过程的一个基本单位。

会话层负责管理通信连接。通信连接是由各应用程序在收发数据时发出的请求（request）和响应（response）建立起来的。也就是说，会话层负责为各应用程序建立逻辑通信链路。以 HTTP 为例，在用户浏览一个 Web 页面的过程中，从发出获取 HTTP 文件的请求到 HTTP 响应的一系列数据收发过程称为一个会话。

如图 1.3 所示，Web 浏览器、电子邮件收发系统和游戏程序的数据通信是分别在不同的会话中被管理的<sup>①</sup>。

---

<sup>①</sup> 在 OSI 参考模型中，虽然会话层定义了若干个协议，但这些协议并非独立运作，而是像 HTTP 协议一样，作为应用程序的功能之一直接实现在应用程序之中。



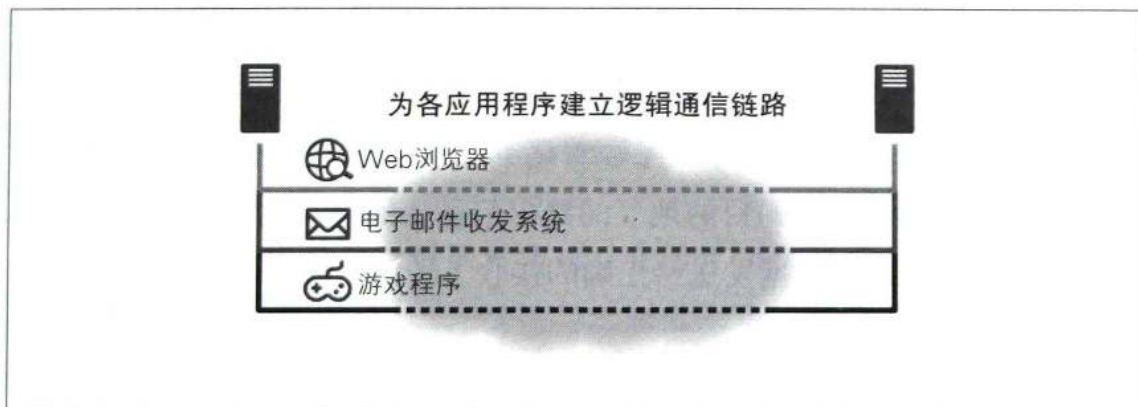


图 1.3 会话

### ——[ 第 4 层 ] 传输层

**传输层**负责建立或断开**连接**（connection），并按照应用程序的要求使用不同的方法转发数据。所谓连接，指的是在会话中，为了进行数据转发而维持的端到端的逻辑通信链路<sup>①</sup>。

传输层主要有两种协议：确保可靠性的 TCP 协议和确保实时性的 UDP 协议。

各应用程序或会话中会建立一条或多条连接。一条也好，多条也好，都是作为逻辑通信链路执行处理的，而在其下层的数据传输中，可以使用任何介质或线路。

此外，传输层会忽略应用程序转发过来的数据长度，直接将数据分割为适合下层传输介质的长度并转发。分割数据的基本单位在 TCP 中称为报文段（segment），在 UDP 中称为数据报（datagram）。从 1.2 节开始，我们将对此进行详细介绍。

### ——[ 第 3 层 ] 网络层

**网络层**负责管理地址、选择路由和把数据发送到目的地。网络层的代表协议是 IP 和 ICMP（Internet Control Message Protocol，互联网控制消息协议）。支持这两种协议的通信设备主要是路由器（router）和 3 层交换机（Layer 3 switch，也称为 L3 交换机）。

<sup>①</sup> “连接”和“会话”非常容易混淆，请务必注意两者的区别。

网络由多台路由器、交换机互相连接而成，数据经由这些设备被不断转发。网络上的所有通信设备都会被分配一个IP地址，它的功能相当于居住地址。此外，这些设备都保存着诸如“接下来将数据发送到哪台设备，数据才能到达最终的目的地”的路由信息。网络层的职责就是基于路由信息，将待发送的数据发送给正确的接收方（图1.4）。

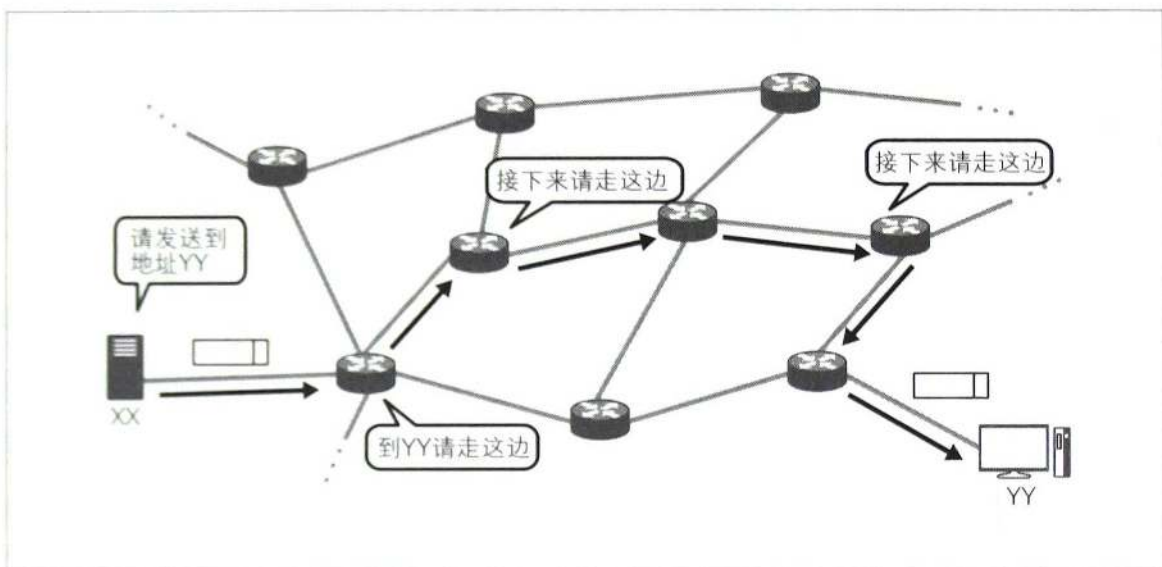


图1.4 数据分发

在发送数据时，基本单位是称为IP包（packet）的小数据块。使用IP包发送数据有很多优势。举例来说，当通信链路发生故障导致一部分IP包丢失时，只需重发这些丢失的IP包便可解决问题<sup>①</sup>，其他的IP包则可通过备用链路顺利完成传输。

但是，即使有数据在转发的过程中丢失，网络层也不会进行数据重发，因为这是上一层，即传输层的职责。

另外，为了完成数据分发，各路由器和通信设备需要提前获取网络路由信息。由于手动配置网络路由信息的工作过于烦琐，所以人们设计了应用层（第7层）协议RIP（Routing Information Protocol，路由信息协议）和BGP（Border Gateway Protocol，边界网关协议），以实现路由信息的自动配置。

<sup>①</sup> 严格来说，这里主要是通过TCP协议重发TCP报文段。

专 栏

IP 地址

IP 协议有 IPv4 和 IPv6 两个版本。两者最主要的差异是包含的地址数量不同。地址数量又称为地址空间（address space）。人们最初使用的 IPv4，其地址由 32 位二进制数表示。随着通信设备逐渐增加，可使用地址逐渐枯竭，所以出现了地址扩展到 128 位的 IPv6 协议。

IP 地址根据用途不同，有多种使用方法。我们先来看格式，IPv4 地址的格式是“xxx.xxx.xxx.xxx”，也就是由 4 个 3 位数加上用于分隔的“.”组成。这是我们常见的 IP 地址。在这种形式的 IP 地址中，高位部分称为网络地址，低位部分称为主机地址。网络地址主要用来标识网络（区域），而主机地址用来标识网络内部的具体设备。

网络地址和主机地址根据所在的网络规模不同，有 5 种划分方法（图 C1.1）。A 类地址的主机地址部分较长，通常适合大规模网络。与之相对，C 类地址主要适合小规模网络。以固定电话号码为例，网络地址和主机地址分别相当于区号和座机号。D 类地址专门用于 IP 多播，而 E 类地址则是预留给未来的。

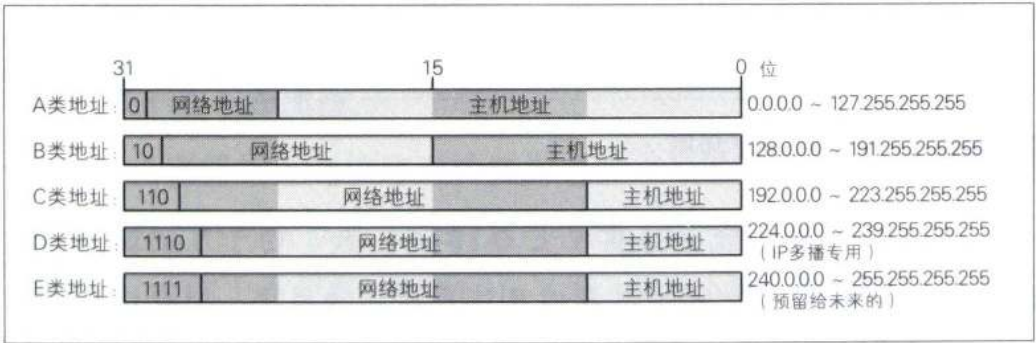


图 C1.1 IP 地址的分类

此外，每一类地址都有“公有地址”和“私有地址”之分。公有地址是分配给所有接入互联网的设备的 IP 地址，而私有地址是分配给办公室或者家庭内部的 LAN（Local Area Network，局域网）之类的，位于局部网络中的设备的 IP 地址。表 C1.1 中记述了几类