TCP是怎样工作的

TURING
图灵程序设计丛书

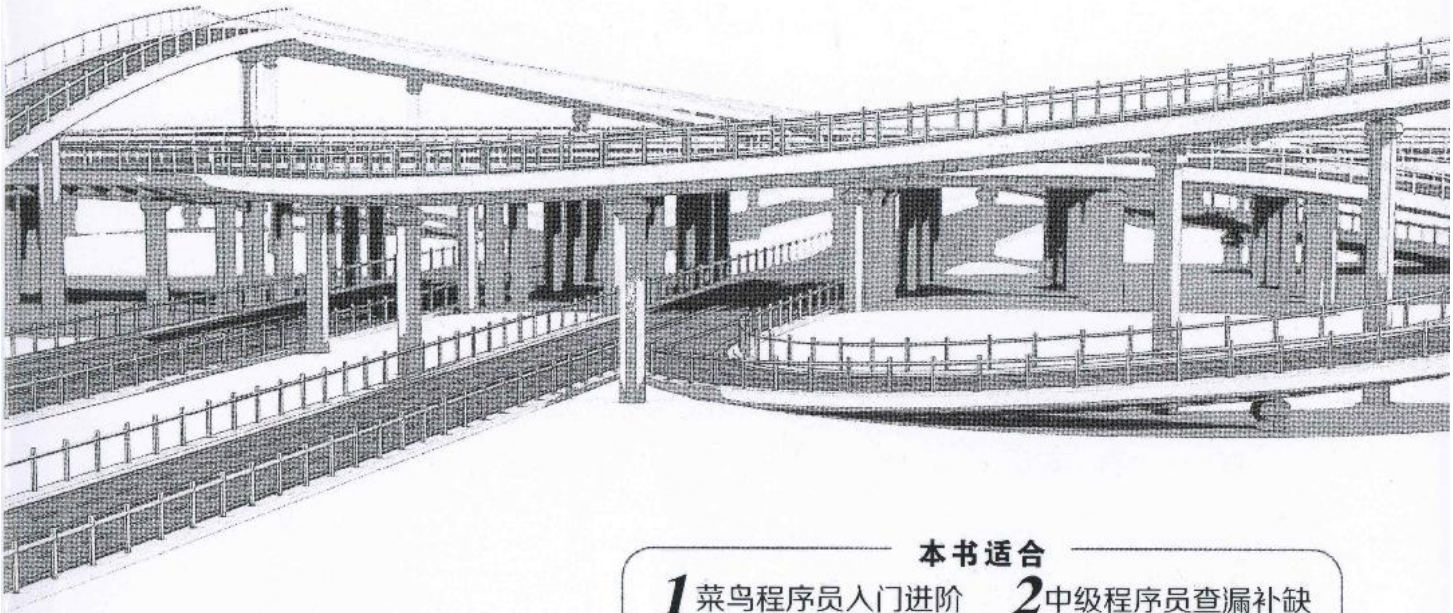
[日] 安永辽真 中山悠 丸田一辉 / 著　　尹修远 / 译

图解×模拟
直击TCP技术核心算法

How TCP Works

“TCP技术入门”通俗图解版

蹲马桶就能看懂的网络协议基础



本书适合

1 菜鸟程序员入门进阶　2 中级程序员查漏补缺
3 高手程序员 / 相关专业教师讲解网络通信关键技术

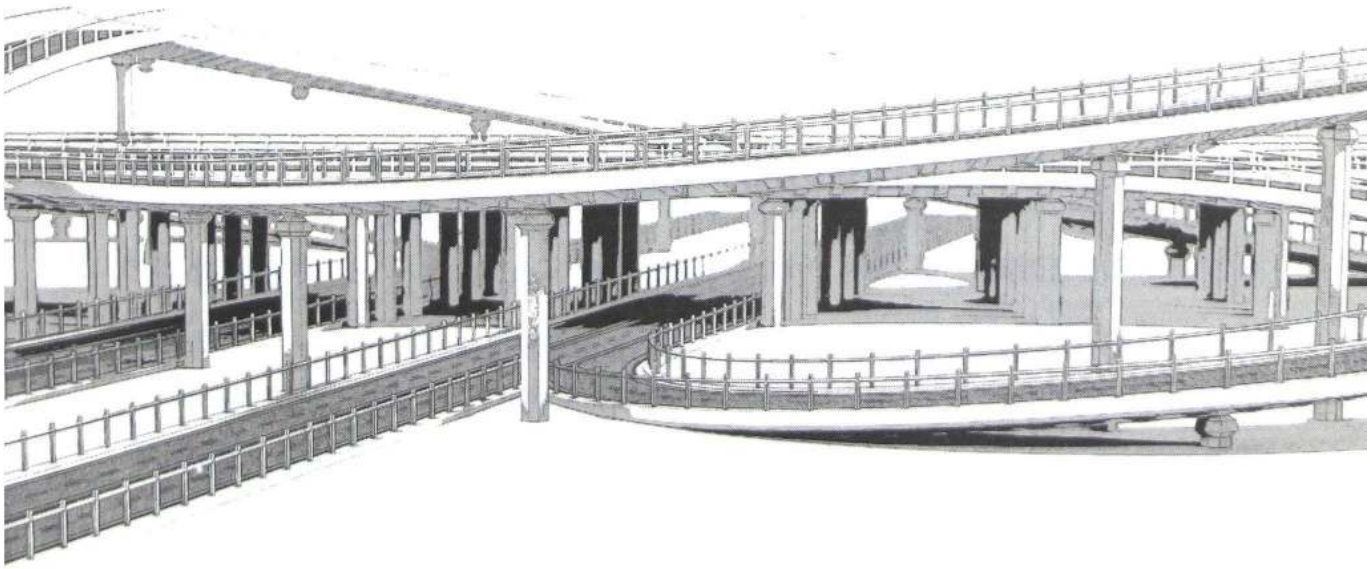
中国工信出版集团　人民邮电出版社 POSTS & TELECOM PRESS

TCP是怎样工作的

TURING
图灵程序设计丛书

[日] 安永辽真 中山悠 丸田一辉 / 著　尹修远 / 译

How TCP Works



人民邮电出版社
北　京

图书在版编目（CIP）数据

TCP是怎样工作的 /（日）安永辽真，（日）中山悠，（日）丸田一辉著；尹修远译. -- 北京：人民邮电出版社，2023.3

（图灵程序设计丛书）

ISBN 978-7-115-61074-4

Ⅰ. ①T… Ⅱ. ①安… ②中… ③丸… ④尹… Ⅲ. ①计算机网络—通信协议 Ⅳ. ①TN915.04

中国国家版本馆CIP数据核字(2023)第003933号

内 容 提 要

本书以图配文，通俗易懂地讲解了长期不会过时的TCP技术。其中，第1章至第3章讲解了TCP的基础知识，详细梳理了TCP的发展历程，并以丰富的图例展示了TCP数据传输的基本思路和过程；第4章至第6章着重介绍了TCP中极为重要的拥塞控制技术，通过图表、公式和模拟实验讲解了TCP拥塞控制的运行机制和热门算法（CUBIC、BBR等）；第7章讲解了TCP前沿的研究动向和今后的发展方向，涉及5G、物联网、数据中心、自动驾驶等内容。

本书理论与实践相结合，在详细讲解TCP原理后，还引领读者搭建模拟环境，使用Wireshark和ns-3等工具模拟TCP的运行机制，观察拥塞控制算法的执行，并辅以伪代码，帮助读者全面理解TCP技术。

本书适合网络开发和管理人员，以及对TCP基础知识及其运行机制感兴趣的人士阅读。

◆ 著　[日]安永辽真　中山悠　丸田一辉
译　尹修远
责任编辑　高宇涵
责任印制　胡　南
◆ 人民邮电出版社出版发行　北京市丰台区成寿寺路11号
邮编　100164　电子邮件　315@ptpress.com.cn
网址　https://www.ptpress.com.cn
大厂回族自治县聚鑫印刷有限责任公司印刷
◆ 开本：880×1230　1/32
印张：9　2023年3月第1版
字数：277千字　2023年3月河北第1次印刷
著作权合同登记号　图字：01-2020-3603号

定价：69.80元

读者服务热线：(010)84084456-6009　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东市监广登字20170147号

前言

作者的话 难度与趣味并存的 TCP

本书是一本入门书，聚焦于近年来日趋受到关注的 TCP（Transmission Control Protocol，传输控制协议）技术，详细解说长期不会过时的 TCP 基础知识和 TCP 前沿的研究动向。

50 年前，计算机界发生了两起革命性的事件。其一是，世界上首个采用分组交换技术的计算机网络阿帕网（Advanced Research Projects Agency Network，ARPANET）建设完成。阿帕网项目旨在解决以往采用线路交换技术的计算机网络中存在的问题，具有划时代性。其二是，AT&T 公司（美国电话电报公司）旗下的贝尔实验室研发了 UNIX 操作系统。在那之后，UNIX 操作系统不仅为阿帕网所用，更成为首个默认搭载 TCP/IP（Transmission Control Protocol/Internet Protocol，传输控制协议 / 互联网协议）的 OS（Operating System，操作系统）。

从那以后，TCP/IP 协议就一直是计算机网络的基础技术。50 年来，无数个通信协议被提出，但其中大部分已经被废弃，因此 TCP/IP 能如此长寿，属实惊人。

与 IP 协议相比，TCP 协议有两点对于初学者并不友好。第一点是“构成逻辑复杂”。TCP 为了确保传输可靠性，加入了一系列风险规避算法。第二点是“技术更新快”。2016 年谷歌提出了新的拥塞控制算法，以小见大，由此就可以看出 TCP 至今仍在不断地发展。究其原因，主要是 TCP 是应用程序端与网络端之间的桥梁。也就是说，TCP 必须随着应用程序与网络的发展，与时俱进地更新技术。

应用程序在这 50 年里飞速发展。优步（Uber）与爱彼迎（Airbnb）的出现打破了出租车行业与酒店行业的传统产业结构。在它们之后，类似的应用软件层出不穷，深入到各行各业。今后，随着 5G（5th Generation，第 5 代移动通信）、物联网（Internet of Things，IoT）和自动驾驶等新技术

的发展，想必也会有各种各样的新应用不断出现。一方面，随着应用程序的发展与变化，人们对 TCP 的要求无疑也会不断变化；另一方面，网络技术的不断发展也暴露出 TCP 中潜在的一些问题。举例来讲，不断有报告称，随着网速的提高和网络设备缓存容量的增大，TCP 的带宽利用率显著下降，同时也有许多与之对应的新解决方案被研究和提出。显而易见，今后的 TCP 也一定会随着应用程序和网络的发展而不断进化。

目前，市面上已经有很多优秀的图书着重解决前文所述的两大难点中的第一点“构成逻辑复杂”，但对于第二点“技术更新快”，也就是 TCP 前沿的研究动向，却很少有图书涉及。例如，在 TCP 拥塞控制算法方面，大部分技术书对 Reno 算法进行了介绍，却没有介绍当前的主流算法 CUBIC。换句话说，如果想要了解 TCP 前沿的发展情况，就必须阅读 RFC（Request For Comments，请求评议）文件、论文或者源代码。这对于初学者来说是一个相当高的门槛。

于是我们撰写了本书，旨在以浅显易懂的方式，从基础知识到前沿研究动向，尽可能全面地为初学者介绍 TCP。尤其是对于更新较为频繁的拥塞控制算法，本书特意采用了大篇幅来详细说明。另外，本书也提供了可供下载的模拟环境，以帮助读者进一步理解 TCP 的技术理论。读者可借助 Wireshark 和 ns-3，自行配置各种条件来研究和观察 TCP 的机制。各位读者在实际模拟的过程中，可能出现符合预期的结果，也可能出现超出预期的结果[1]。当出现超出预期的结果时，请仔细思考原因，反复实验，直到结果正确为止。如果能重复这个过程，相信读者一定会对 TCP、对计算机网络有更深入的理解。

作者代表　安永辽真

2019 年 6 月

1) 笔者在进行模拟时得到的通常是超出预期的结果，如果只模拟一次就得到了符合预期的结果，反倒会不放心。

本书的结构

如图 0.1 所示，除前言外，本书共有 7 章。

首先，第 1 章到第 3 章全局性地总览 TCP 的基础知识。第 1 章概述计算机网络的基础知识，以及 UDP（User Datagram Protocol，用户数据报协议）与 TCP 之间的差异。第 2 章介绍 TCP 诞生的背景。第 3 章讲解 TCP 协议的设计方法。

接下来，第 4 章到第 6 章深入挖掘 TCP 的核心技术——拥塞控制。第 4 章概述拥塞控制的基本思想，以及迄今为止所提出的各种拥塞控制算法。第 5 章和第 6 章重点介绍近几年来最重要的拥塞控制算法 CUBIC 和 BBR。

最后，第 7 章介绍 TCP 前沿的研究动向和今后的技术发展。

此外，每章末尾列有部分参考资料，如需了解更详细的知识，请查阅相应的参考资料。

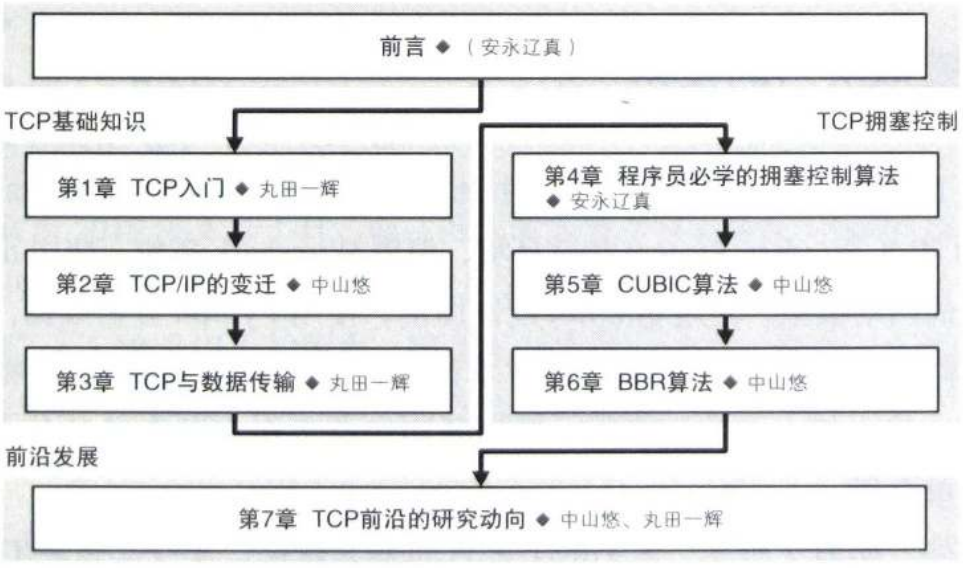


图 0.1 本书的结构及各章执笔人

本书的目标读者

本书主要面向对于 TCP 基础知识和其前沿发展情况感兴趣的读者。要想充分理解本书的内容，读者需要具备基础的高中数学知识。具体来说，只要能理解以下几个概念，那么学习本书就没有问题。

- 等号和不等号
- 集合符号（如 $\in$。$\in$ 代表左侧属于右侧）
- 逻辑符号（如 $\forall$。不过 $\forall$ 不在日本高中数学的范畴，它代表“任意”）
- 幂运算
- 对数函数
- 指数函数

此外，本书将在讲解拥塞控制算法的章节中使用伪代码，同时也会使用终端命令进行模拟，因此读者如果有编程经验，理解起来会更容易。但是，只要按照顺序认真阅读，其实伪代码也不难理解，而对于终端命令，即使不了解具体含义也无妨，所以哪怕是没有编程经验的读者也无须担心。

本书模拟所用的技术

为了进行模拟，本书使用了若干技术。具体来说，我们使用 VirtualBox、Vagrant 和 X Window System 构建环境，使用 Wireshark 抓包，使用 ns-3 进行具体的网络模拟，使用 gnuplot 绘制图形，使用 Python 分析数据，使用 shell 脚本运行终端命令。每一项技术都很有深度，值得用很长篇幅来详细介绍。但由于本书篇幅所限，模拟所用的技术在本书中都只会点到为止，简单介绍。

另外，如前文所述，本书的主要目的是方便初学者快速理解 TCP 的理论知识。尽管书中会使用 Wireshark 和 ns-3 进行网络模拟，但那只是为了加深读者对理论知识的理解，因此本书并不涉及实际的实现方法与步骤。利用了 TCP 技术的系统实现或者网络编程方面的技术，本书并不涉及，请阅读相应的参考书进行学习。

本书所需的运行环境

本书内容在以下环境中进行了验证。模拟基本上是通过虚拟机进行的，因此只需搭建出如下所示的 VirtualBox、Vagrant 和 X Server 的环境，那么在 masOS 以外的环境中也可以顺利完成模拟。

- OS：masOS Mojave 10.14.3
- 处理器：2.9 GHz Intel Core i7
- 内存：16 GB 2133 MHz LPDDR3
- VirtualBox：6.0.4r128413
- Vagrant：2.2.4

本书内容也已在如下的 VirtualBox 虚拟机环境中进行了验证。

- Ubuntu：16.04
- Wireshare：2.6.5
- ns-3：3.27
- Python：3.5.2
- GCC：5.4.0
- make：4.1

截至 2019 年 4 月 1 日，ns-3 的安装向导并没有适配 Ubuntu 18.04，因此本书使用 Ubuntu 16.04。第 5 章和第 6 章使用的 CUBIC 和 BBR 模块没有适配 ns-3.28 及以上的版本，因此本书使用 ns-3.27。此外，Python 使用 PEP 8-Style Guide for Python Code 作为代码规范。

用于模拟的环境的搭建

下面，我们将介绍如何搭建用于模拟的环境。本书使用 VirtualBox 和 Vagrant 搭建虚拟环境，并通过 X Window System 在虚拟机中运行 GUI 应用程序。

关于命令运行

本书使用终端程序运行命令，以便完成模拟。终端程序类似于Windows中的命令行提示符和macOS中的Terminal.app，这些应用程序都是通过在GUI中打开终端窗口来运行命令的。本书在终端中运行的命令主要表现为以下形式。

shell

```
$ echo 'hello world'
> hello world
```

以$开头的部分主要代表输入的命令，以>开头的部分代表标准输出。本书以使用虚拟机为主，在物理机上运行的命令以$开头，在虚拟机上运行的命令以{登录用户名}@{虚拟机名}:{当前目录名}$的形式展示，例如vagrant@ubuntu-xenial:~$。

获取源代码

本书进行模拟所用的源代码，可以通过以下网址来获取。直接下载zip文件，或是通过克隆（clone）都可以获取源代码。

URL https://github.com/ituring/tcp-book

Oracle VM VirtualBox

Oracle VM VirtualBox是一个x86虚拟机软件。宿主操作系统（运行VirtualBox的物理机OS）支持Windows、Linux、macOS和Solaris。在虚拟机上运行的客户操作系统支持Windows、Linux、OpenSolaris、OS/2和OpenBSD。Web开发工程师会使用VirtualBox作为服务器端或客户端的验证环境，网络工程师也会用它来构建验证网络的环境。本书为了统一用于模拟的环境，使用VirtualBox和Vagrant搭建虚拟环境。后文将针对Vagrant进行介绍。

下面介绍如何安装VirtualBox。在笔者执笔时（2019年4月），从VirtualBox的官方网站可以直接下载与宿主操作系统适配的安装包。运行安装程序即可完成VirtualBox的安装。

在macOS系统下，打开终端程序（例如Terminal.app），运行以下命

令，只要终端上输出了相应的版本号，就可以确认 VirtualBox 已成功安装。请注意，所用环境不同，终端上显示的版本号也可能不一样。

shell

```
$ VBoxManage -v
> 6.0.4r128413
```

Vagrant

Vagrant 是虚拟环境的自动配置工具。它基于 Ruby 开发，支持在 Debian、Windows、CentOS、Linux、macOS 和 ArchLinux 上运行。只要能共享 `Vagrantfile` 配置文件，就可以轻松地统一虚拟环境。前文所述的源代码的下载网址中提供了本书所用的 `Vagrantfile` 文件，读者使用此文件可以轻松地构建出用于模拟的环境。

在笔者执笔时（2019 年 4 月），点击 Vagrant 官方网站的 [Download] 按钮，页面会跳转到安装包的下载界面。请根据所用环境选择相应的安装包，下载完成之后运行安装程序，以便完成安装。

在 macOS 系统下，请在终端程序（Terminal.app 等）中运行以下命令。如果输出了相应的版本号，就说明 Vagrant 已经安装完毕。请注意，所用环境不同，终端上显示的版本号也可能不一样。

shell

```
$ vagrant -v
> Vagrant 2.2.4
```

X Server

本书在客户操作系统上使用 X Window System 运行 Wireshark，因此需要在宿主操作系统上搭建 X Server 环境。在笔者执笔时（2019 年 4 月），在 macOS X Serra 系统下可以通过 XQuartz 项目的官方网站获取 X11 Server（X11.app）。请注意，如果使用其他操作系统，获取方式有所不同。

为了验证 X Server 的运行情况，请启动虚拟机上的 GUI 应用程序。首先，请打开已下载的本书源代码，定位到 `wireshark/vagrant/` 目录，然后运行以下命令。

shell

```
$ vagrant up
```

这样，第 4 章所用的 Wireshark 虚拟环境就搭建完成了（可能会花费一点时间）。接下来运行下面的命令，进行 SSH 连接，启动 xeyes。

shell

```
$ vagrant ssh guest1

> Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-139-generic x86_64)
>
> * Documentation:  部分省略
> * Management:     部分省略
> * Support:        部分省略
>
> Get cloud support with Ubuntu Advantage Cloud Guest:
> 部分省略
>
> 0 packages can be updated.
> 0 updates are security updates.
>
> New release '18.04.1 LTS' available.
> Run 'do-release-upgrade' to upgrade to it.

vagrant@guest1:~$ xeyes
```

请确认是否有如图 0.2 所示的两个眼球出现。如果有，请运行以下命令，暂时退出虚拟机以停止运行。

shell

```
vagrant@guest1:~$ exit
$ vagrant halt
```

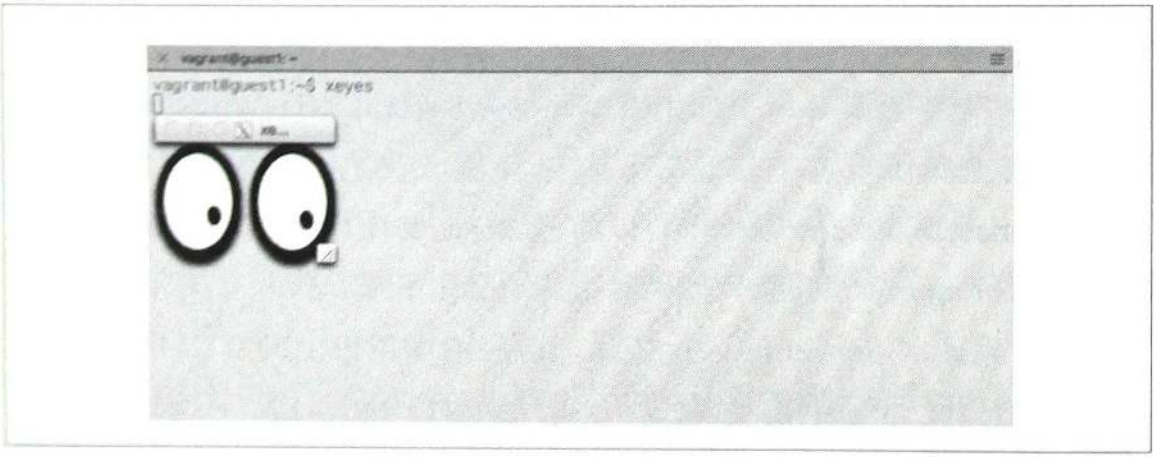


图 0.2 xeyes 的运行结果

致谢

本书能够完成写作，幸赖很多朋友的帮助。从笔者开始构想本书之时，唐仁原骏先生便提出参考意见，后面还与我们一起确认模拟环境的运行情况。大阪大学研究生院工学研究科的久野大介先生参与了本书的校对和审阅。从本书的策划组稿到写作进度管理，技术评论社的土井优子编辑给予了我们很多帮助。衷心感谢一直以来支持着笔者的家人们。真的万分感谢各位。

关于本书主页

各位读者可以通过以下网址访问本书的支持页面。

URL ituring.cn/book/2851

目录