1、网络协议和常用网络工具

T A H N K Y O U F O R W A T C H I N G

主讲老师Mark：446106311

课程安排

享学课堂-网络通信 课程表	
章节	章节名称
1	网络协议和常用网络工具
2	Java原生网络编程
3	操作系统和JDK对网络通信的实现

上课说明：

课程前置知识：Java语言基础知识、并发编程的基础知识，如线程的使用、线程池的使用等等。

1、首次出现的知识如需要进行编码，一般会进行手写，以后再出现则可能会事先准备好或者进行拷贝。

2、一个知识点如果大部分同学明白，不会重复讲解，未明白的同学请看视频、笔记、请教同学或加老师QQ。

3、以上为本课的章节安排，不是课时安排，如果一章内容在一次课内未讲完，则会顺延到后面的课程继续讲解。

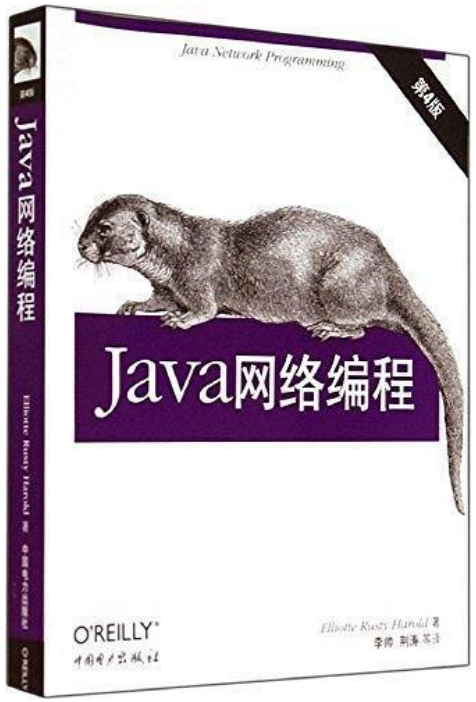
4、遵循着 **基础知识→初步应用→进阶和实战→源码和原理**学习路径

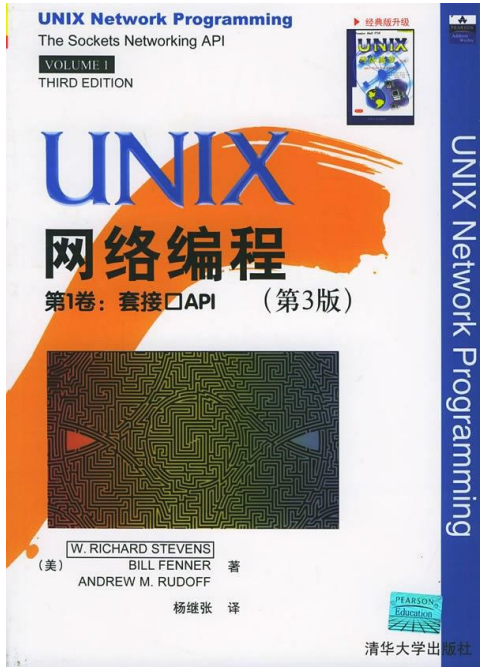
5、课程章节用不同的颜色标注了难度和应该掌握的程度：

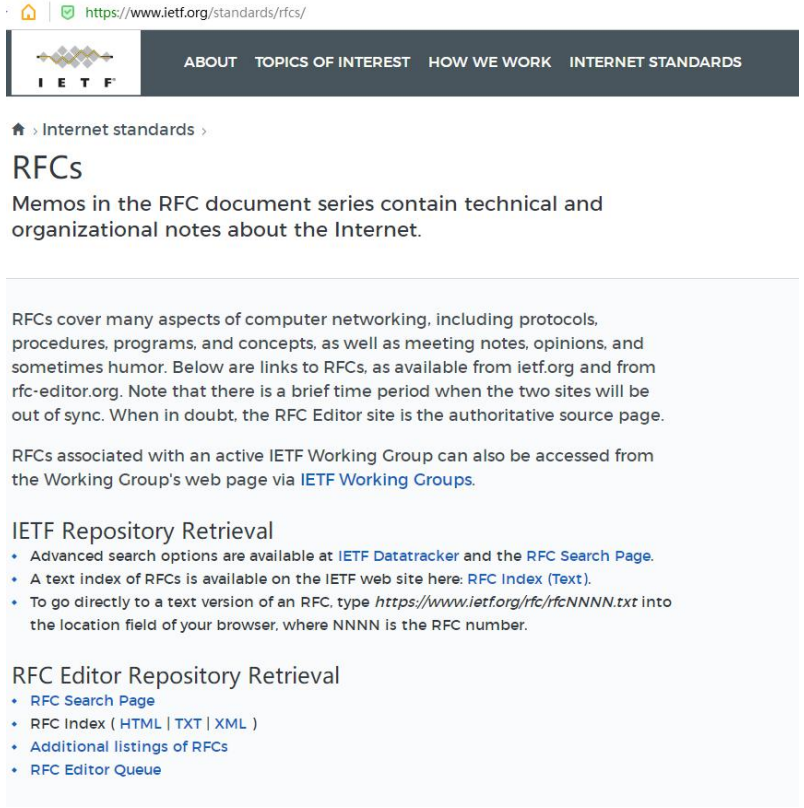
浅绿色表示作为程序员必须掌握的网络通信方面的基础知识和编程技能，必修-1，

红色供学有余力的同学研究学习，选修-2

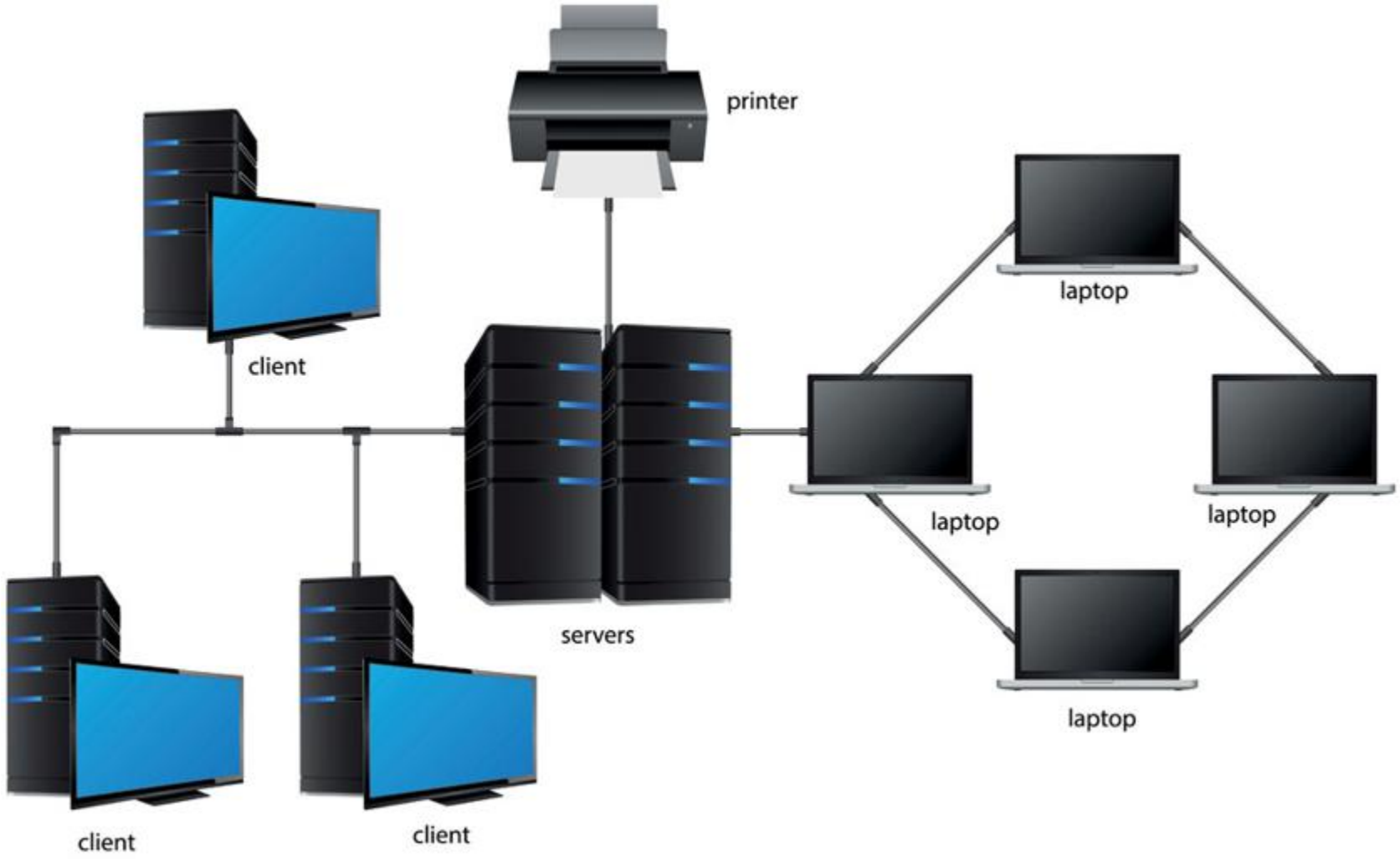
书籍推荐


Java Network Programming
第4版
Java网络编程
Java网络编程
O'REILLY


UNIX Network Programming
The Sockets Networking API
VOLUME 1
THIRD EDITION
UNIX
网络编程
第1卷：套接口API （第3版）
W. RICHARD STEVENS
BILL FENNER
ANDREW M. RUDOFF
（美） 著
杨继张 译
清华大学出版社
UNIX Network Programming


https://www.ietf.org/standards/rfcs/
ABOUT
TOPICS OF INTEREST
HOW WE WORK
INTERNET STANDARDS
Internet standards
RFCs
Memos in the RFC document series contain technical and organizational notes about the Internet.
RFCs cover many aspects of computer networking, including protocols, procedures, programs, and concepts, as well as meeting notes, opinions, and sometimes humor. Below are links to RFCs, as available from ietf.org and from rfc-editor.org. Note that there is a brief time period when the two sites will be out of sync. When in doubt, the RFC Editor site is the authoritative source page.
RFCs associated with an active IETF Working Group can also be accessed from the Working Group's web page via IETF Working Groups.
IETF Repository Retrieval
Advanced search options are available at IETF Datatracker and the RFC Search Page.
A text index of RFCs is available on the IETF web site here: RFC Index (Text).
To go directly to a text version of an RFC, type https://www.ietf.org/rfc/rfcNNNN.txt into the location field of your browser, where NNNN is the RFC number.
RFC Editor Repository Retrieval
RFC Search Page
RFC Index (HTML | TXT | XML)
Additional listings of RFCs
RFC Editor Queue

计算机网络是什么？


printer
client
laptop
laptop
laptop
servers
client
client
laptop

计算机网络是什么？

利用通信线路将地理上分散的、具有独立功能的计算机系统和通信设备按不同的形式连接起来，以功能完善的网络软件及协议实现资源共享和信息传递的系统。

主要网络有哪些?

1、局域网
2、城域网
3、广域网

计算机网络发展概史

1、诞生阶段，单个计算机为中心的远程联机系统
2、ARPANET，多个主机通过通信线路互联起来
3、开放性的标准化体系结构，OSI诞生
4、Internet互联网

计算机网络体系结构

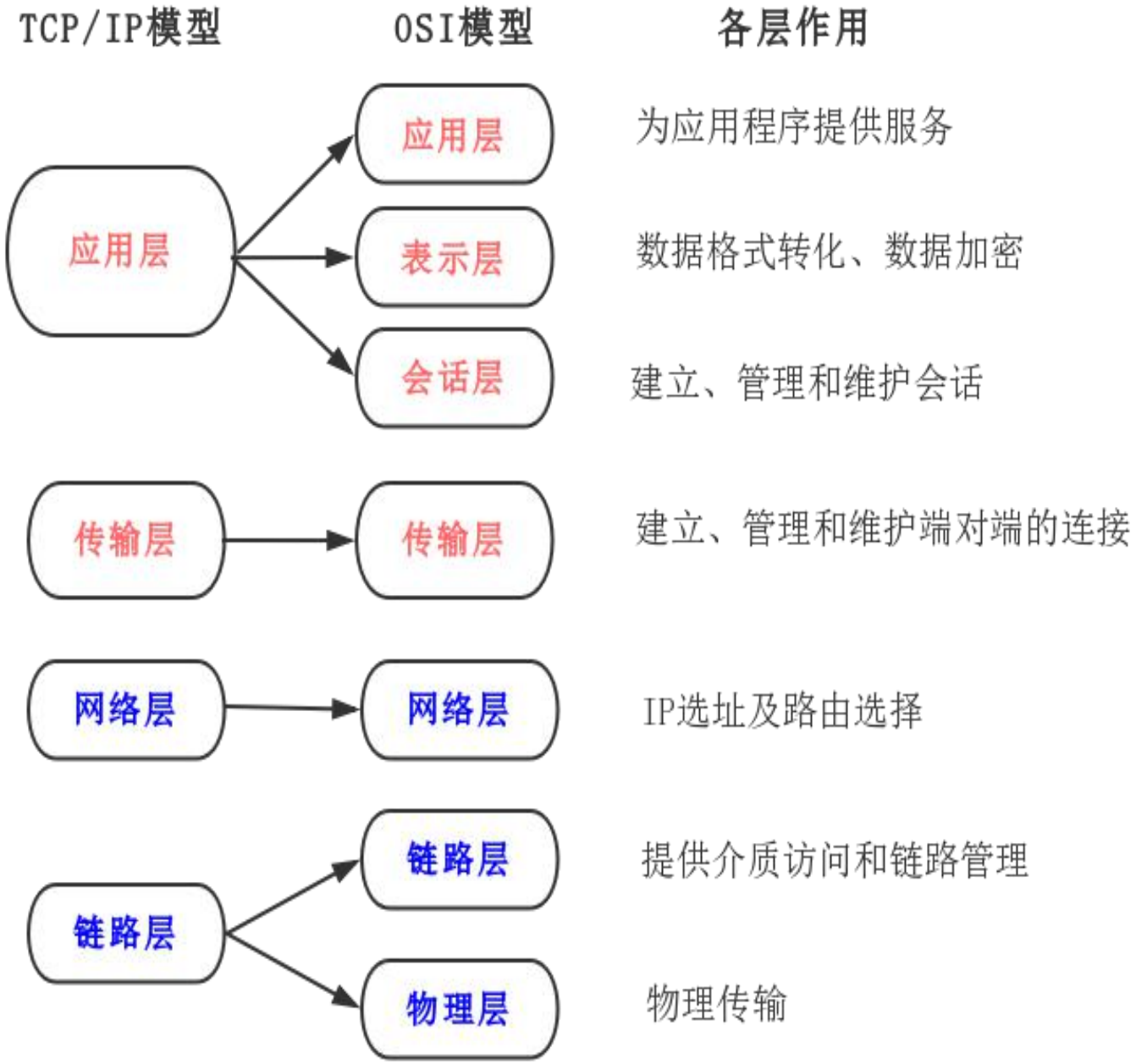
- **各层的关系**

每一个抽象层建立在低一层提供的服务上，

并且为高一层提供服务。

- **程序员重点关注TCP/IP模型，OSI七层模型了解即可**

面试点



TCP/IP协议族

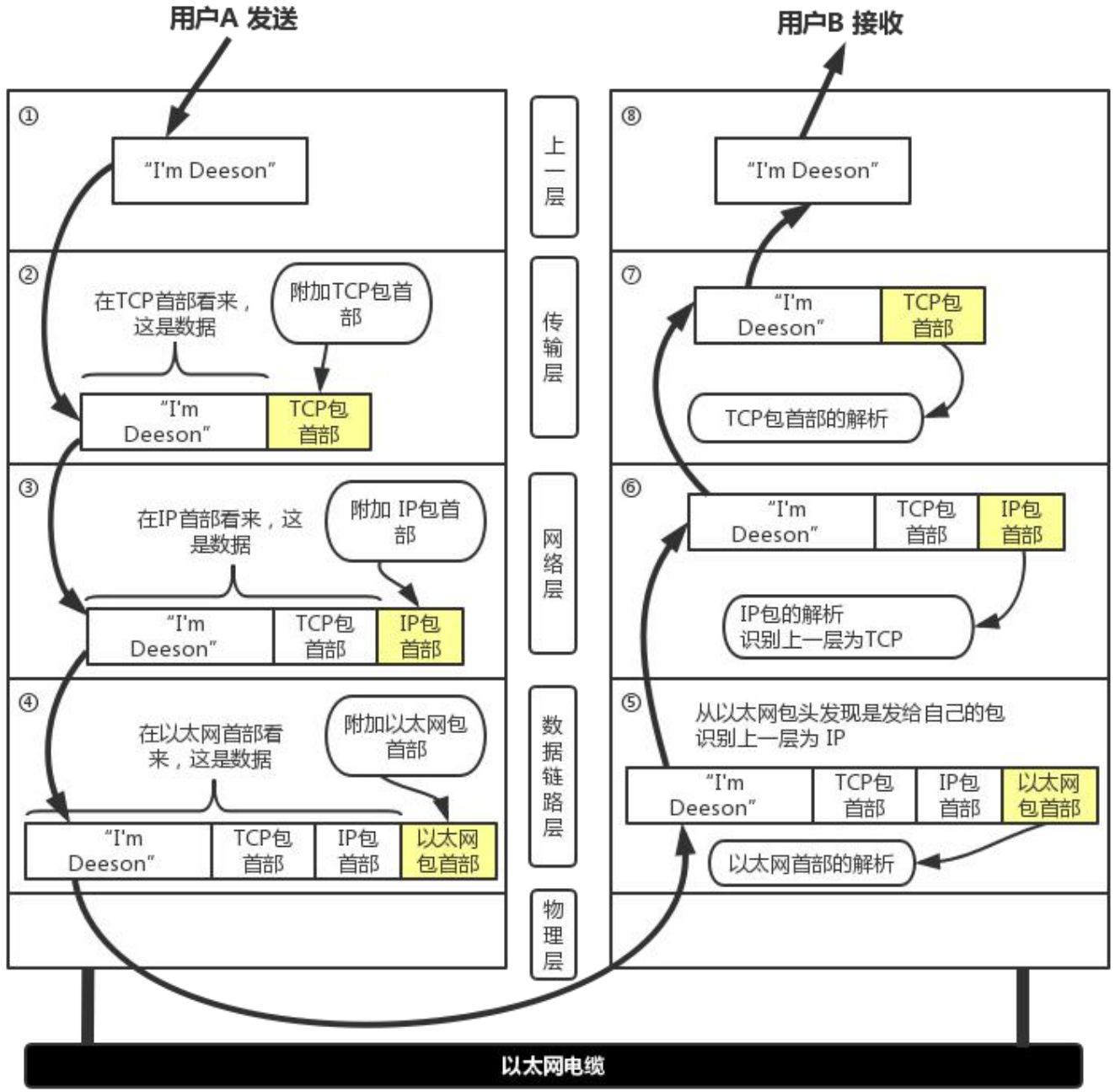
定义

Transmission Control Protocol/Internet Protocol的简写，中译名为传输控制协议/因特网互联协议，是Internet最基本的协议、Internet国际互联网络的基础，由网络层的IP协议和传输层的TCP协议组成。协议采用了4层的层级结构。然而在很多情况下，它是利用 IP 进行通信时所必须用到的协议群的统称。

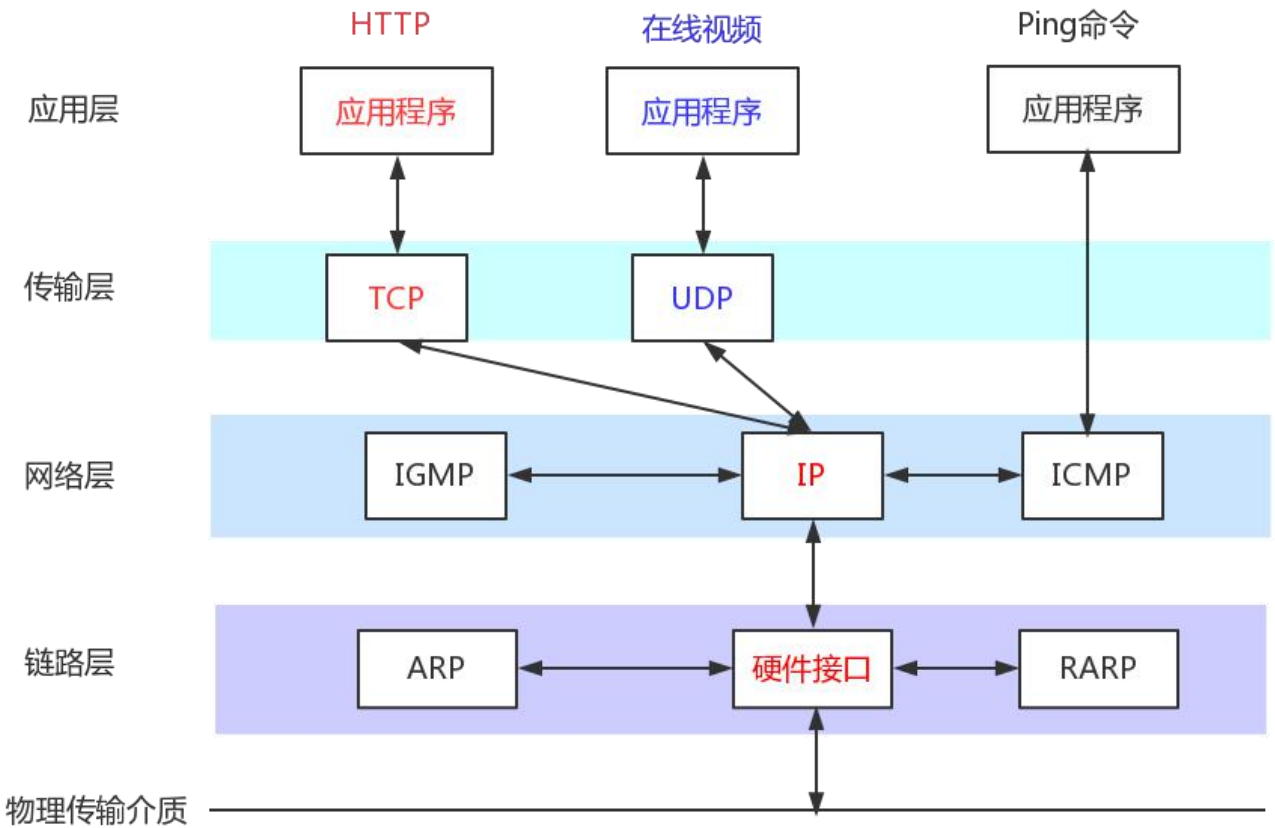
TCP/IP概念层模型	功能	TCP/IP协议族
应用层	文件传输，电子邮件，文件服务，虚拟终端	TFTP，HTTP，SNMP，FTP，SMTP，DNS，Telnet
	数据格式化，代码转换，数据加密	没有协议
	解除或建立与别的接点的联系	没有协议
传输层	提供端对端的接口	TCP，UDP
网络层	为数据包选择路由	IP，ICMP，RIP，OSPF，BGP，IGMP
链路层	传输有地址的帧以及错误检测功能	SLIP，CSLIP，PPP，ARP，RARP，MTU
	以二进制数据形式在物理媒体上传输数据	ISO2110，IEEE802，IEEE802.2

网络传输中的数据

- **包**是全能性术语；
- **帧**用于表示数据链路层中包的单位；
- **片**是 IP中数据的单位；
- **段**则表示 TCP 数据流中的信息；
- **消息**是指应用协议中数据的单位。



TCP/IP协议族



TCP 面向连接的、可靠的流协议

UDP 面向无连接的通讯协议

IP 在源地址和目的地址之间传送的数据包

ICMP 控制报文协议
IGMP internet组管理协议
ARP 地址解析协议
RARP 反向地址转化协议

移动通信中的4G、5G在哪一层？

网络通信中的地址和端口号

MAC 地址

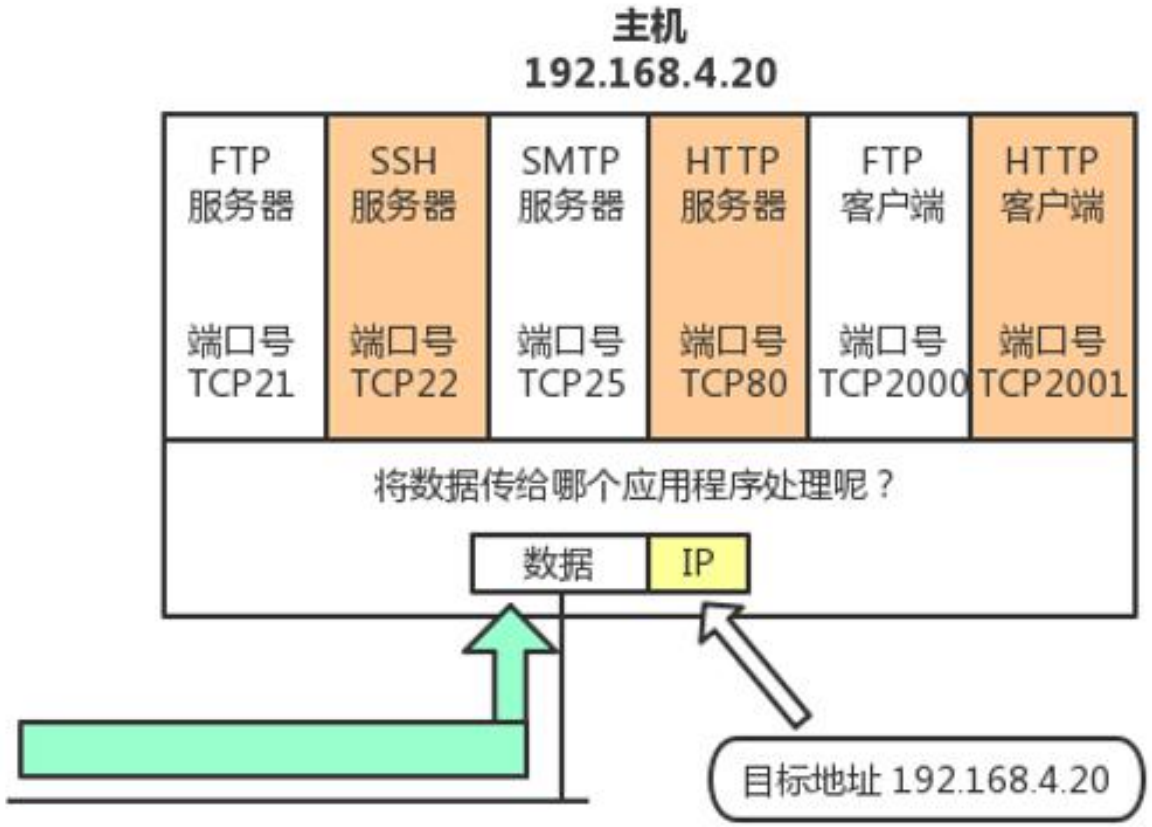
```
以太网适配器 以太网:

   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : Realtek PCIe GbE Family Controller
   物理地址. . . . . . . . . . . . . : B0-25-AA-35-[redacted]-BE
   DHCP 已启用 . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . : 是
   本地链接 IPv6 地址. . . . . . . . : fe80::c85e:d254:bf25:449a%12(首选)
   IPv4 地址 . . . . . . . . . . . . : 192.168.0.154(首选)
   子网掩码 . . . . . . . . . . . . : 255.255.255.0
   获得租约的时间 . . . . . . . . . : 2020年9月15日 10:25:02
   租约过期的时间 . . . . . . . . . : 2020年9月15日 16:08:59

无线局域网适配器 WLAN:

   媒体状态 . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : Intel(R) Wireless-AC 9462
   物理地址. . . . . . . . . . . . . : 24-41-8C-26-[redacted]-9E
   DHCP 已启用 . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . : 是
```

网络通信中的地址和端口号

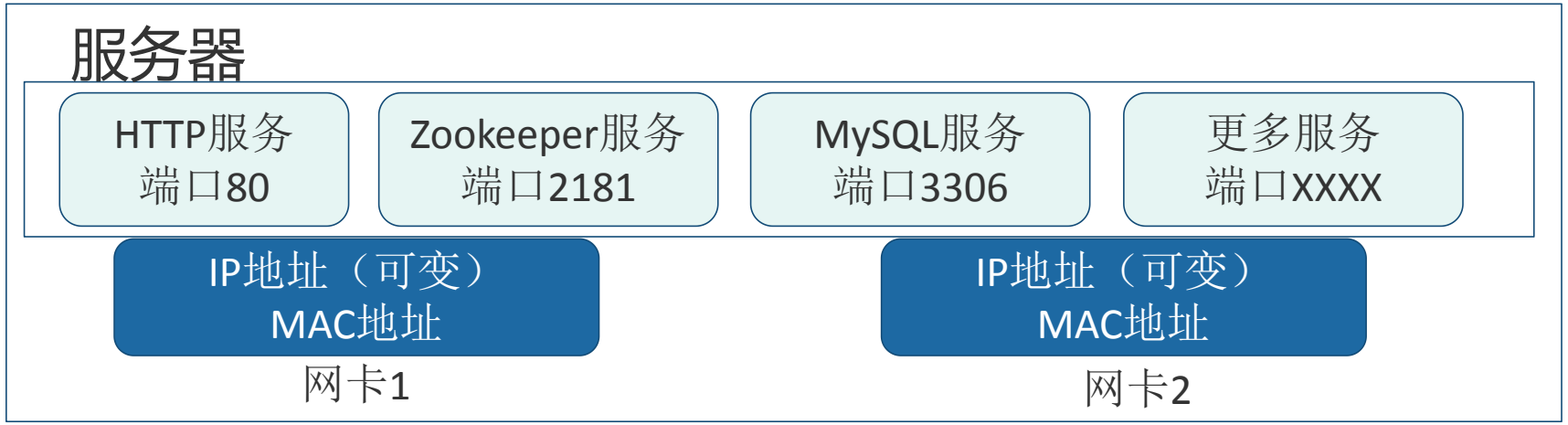
IP地址

端口号

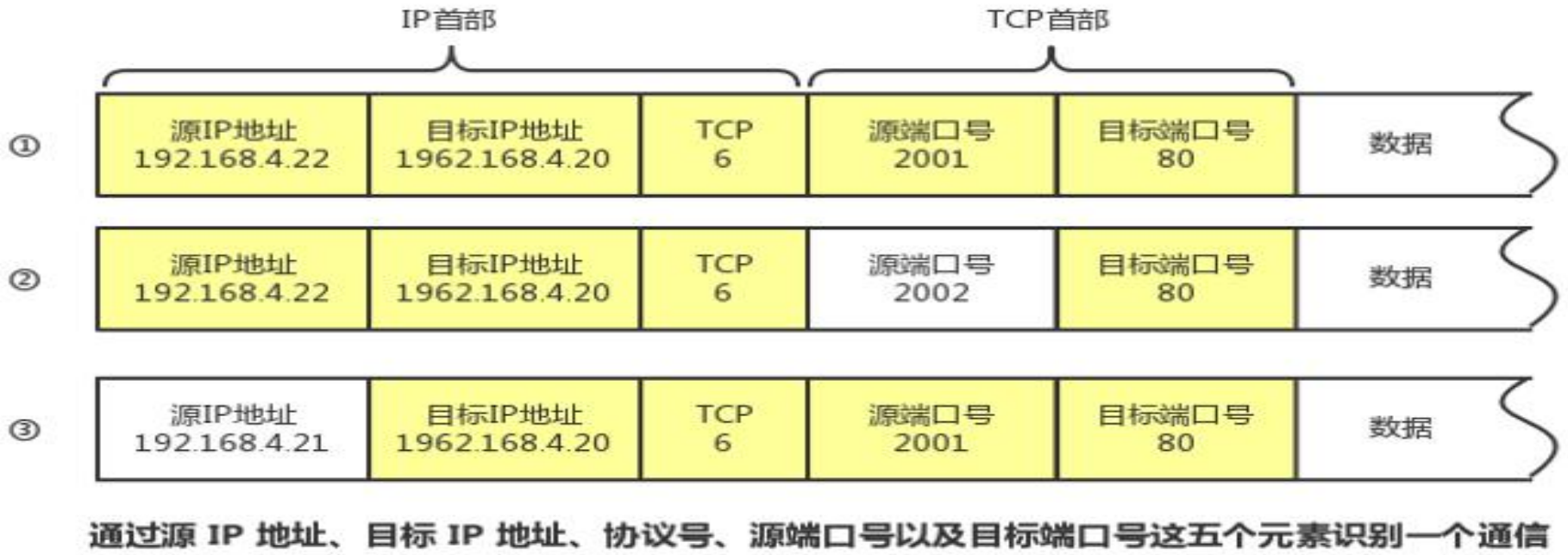
用来识别同一台计算机中进行通信的不同应用程序。因此，它也被称为**程序地址**



网络通信中的地址和端口号



- 端口号的确定



TCP概述

- 什么是TCP （ Transmission Control Protocol ）？
- TCP的基本特性
  - 面向连接
  - 可靠性
  - RTT和RTO
  - 数据排序
  - 流量控制
  - 全双工

TCP连接中的三次握手

面试点

三次握手

建立一个TCP连接时，需要客户端和服务端总共发送3个包以确认连接的建立。

- **第一次握手**

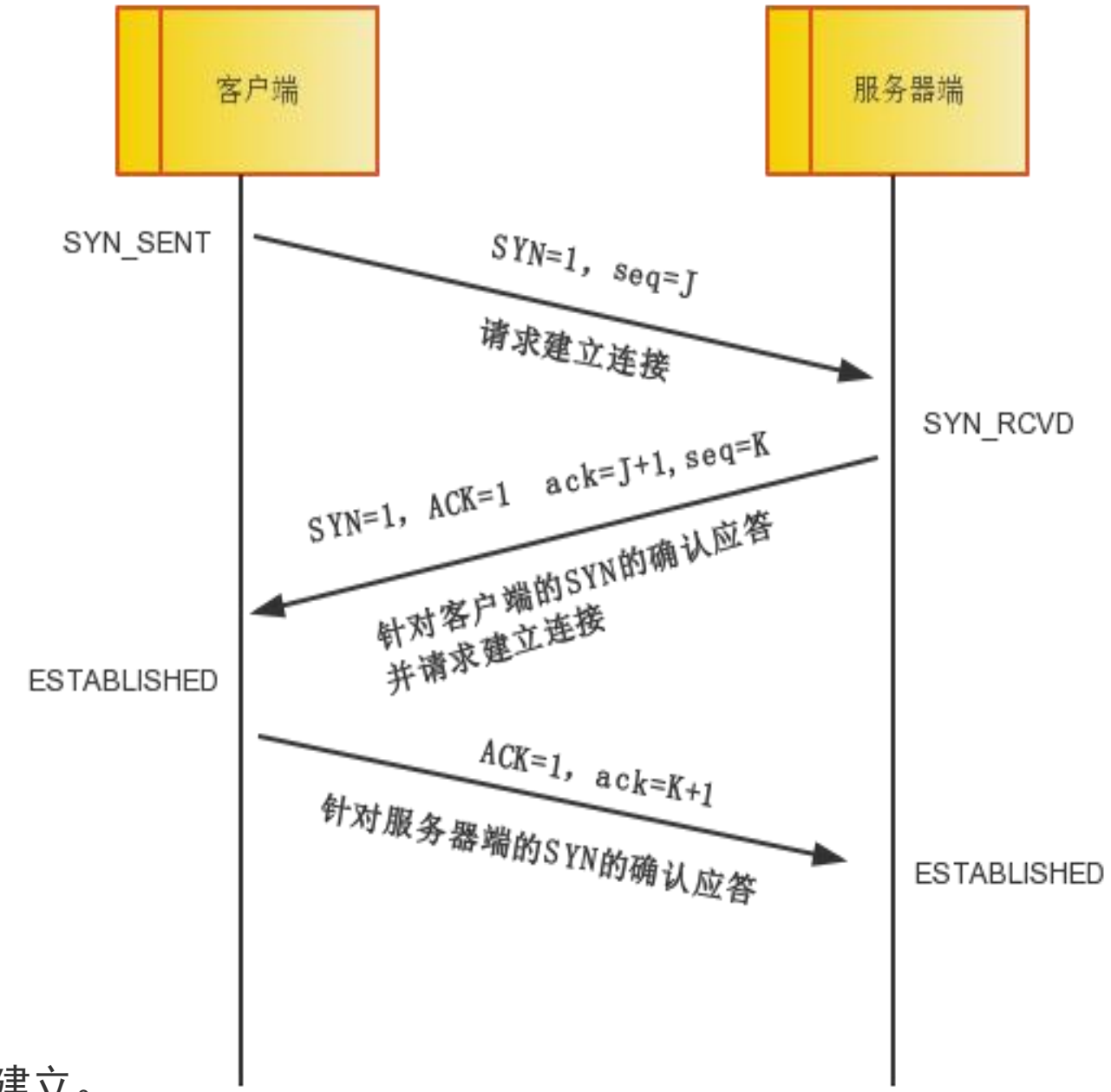
  客户端请求建立连接。

- **第二次握手**

  服务端应答客户端，并请求建立连接。

- **第三次握手**

  客户端针对服务端请求确认应答。



- **为什么需要3次握手?**

TCP是面对连接的，所以需要双方都确认连接的建立。

TCP的3次握手的漏洞-SYN洪泛攻击

SYN洪泛攻击

定义

通过网络服务所在的端口发送大量伪造原地址的攻击报文，发送到服务端，造成服务端上的半开连接队列被占满，从而阻止其他用户进行访问。

原理

攻击者客户端利用伪造的IP地址向服务端发出请求(第一次握手)，而服务端的响应(第二次握手)的报文将永远发送不到真实的客户端，服务端在等待客户端的第三次握手(永远都不会有的)，服务端在等待这种半开的连接过程中消耗了资源，如果有成千上万的这种连接，主机资源将被耗尽，从而达到攻击的目的。

解决方案

- ✓ 无效连接监控释放
- ✓ 延缓TCB分配方法
- ✓ 防火墙

TCP中的四次挥手（分手）

可能面试点

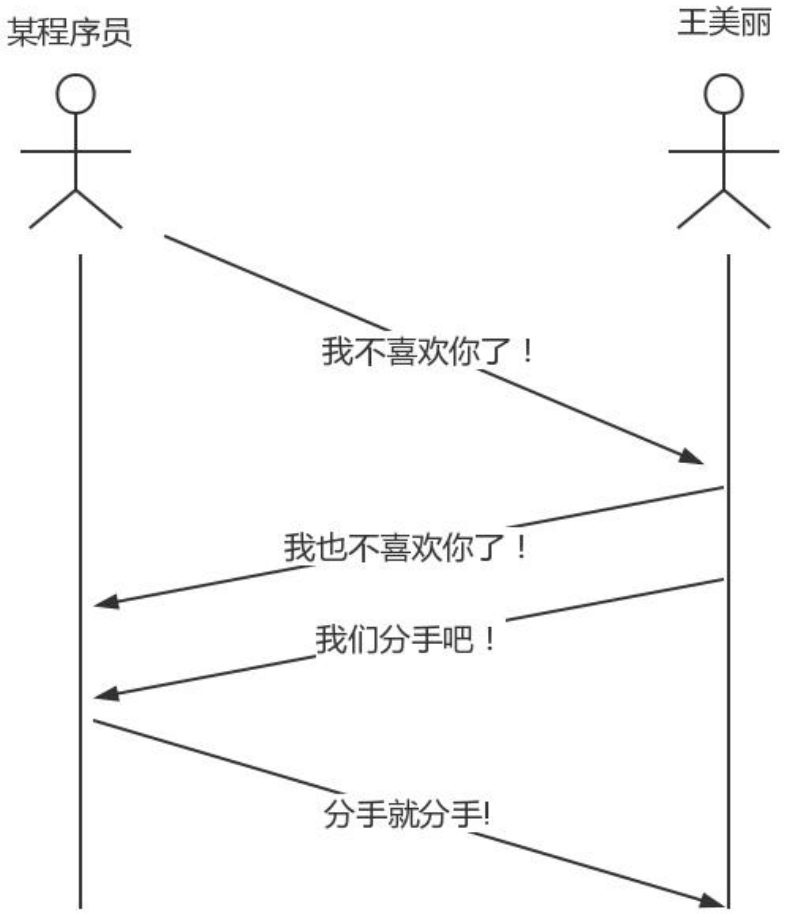
四次挥手

定义

断开一个TCP连接时，需要客户端和服务端总共发送4个包以确认连接的断开。

过程

- ✓ 第一次挥手：客户端发送关闭请求
- ✓ 第二次挥手：服务端响应客户端关闭请求
- ✓ 第三次挥手：服务端发送关闭请求
- ✓ 第四次挥手：客户端发送关闭确认请求



为什么需要四次挥手

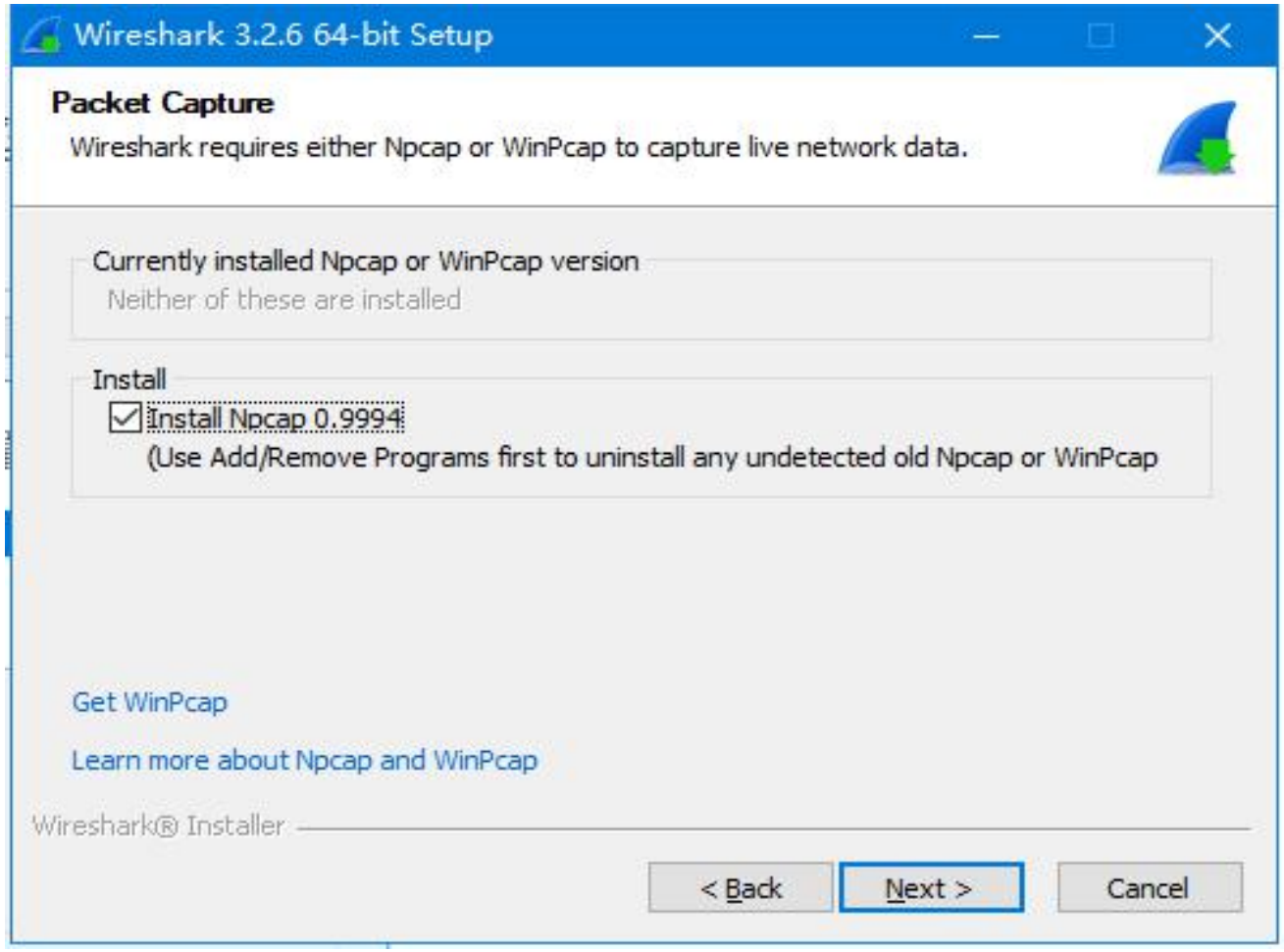
TCP是双全工(即客户端和服务器端可以相互发送和接收请求)，所以需要双方都确认关闭连接。

为什么需要TIME-WAIT状态？

常用的网络工具Wireshark和tcpdump

- 下载与安装

https

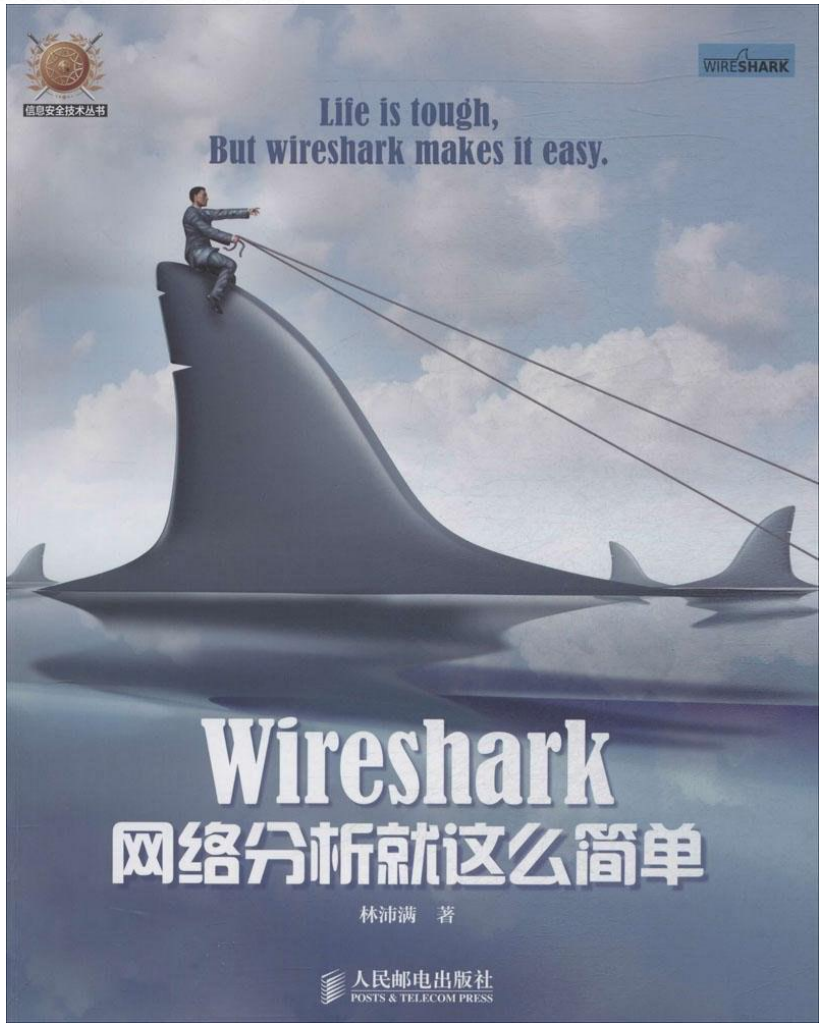


- 数据包的捕获和基本用法

- 实战：用WireShark看看TCP的三次握手

常用的网络工具Wireshark和tcpdump

WIRESHARK 推荐书籍



TCPDUMP详解

tcpdump命令是基于*nix系统的命令行的数据报嗅探工具，可以抓取流动在网卡上的数据包。

```
[root@iZwz9j203ithc4guluvb2wZ ~]# tcpdump --help
tcpdump version 4.9.2
libpcap version 1.5.3
OpenSSL 1.0.2k-fips  26 Jan 2017
Usage: tcpdump [-aAbdDefhHIJKlLnNOpqStuUvxX#] [ -B size ] [ -c count ]
                [ -C file_size ] [ -E algo:secret ] [ -F file ] [ -G seconds ]
                [ -i interface ] [ -j tstamptype ] [ -M secret ] [ --number ]
                [ -Q|-P in|out|inout ]
                [ -r file ] [ -s snaplen ] [ --time-stamp-precision precision ]
                [ --immediate-mode ] [ -T type ] [ --version ] [ -V file ]
                [ -w file ] [ -W filecount ] [ -y datalinktype ] [ -z postrotate-command ]
                [ -Z user ] [ expression ]
```

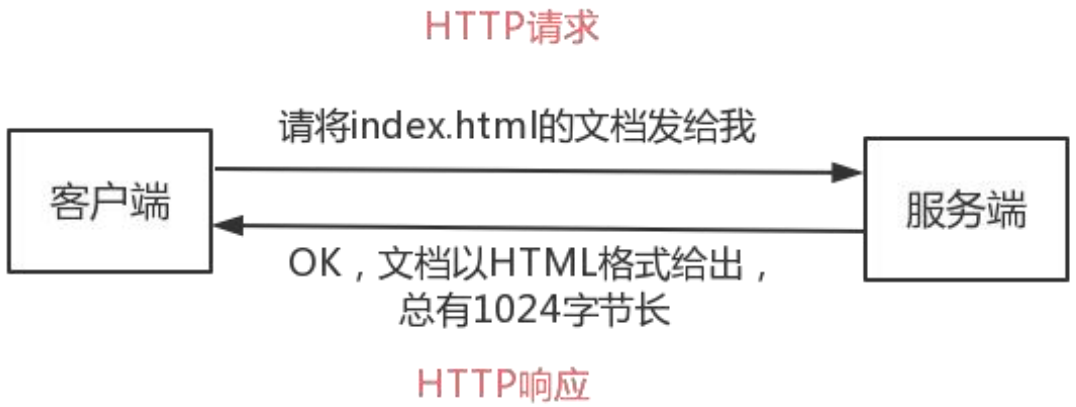
HTTP协议

- HTTP：HTTP协议是Hyper Text Transfer Protocol（超文本传输协议）的缩写,是用于从万维网（WWW:World Wide Web ）服务器传输超文本到本地浏览器的传送协议。

- **web客户端和服务器**
- **URI和URL**

  web服务器资源的名字和用于描述一个网络上资源的地址

  - ✓ schema: http/https/ftp.
  - ✓ host: web服务器的ip地址或者域名
  - ✓ port: 服务端端口， http默认访问的端口是80
  - ✓ path: 资源访问路径
  - ✓ query-string: 查询参数
- **方法：**GET/PUT/DELETE/POST/HEAD



https://tools.ietf.org/html/rfc2616

一次完整http请求的过程

面试点

1、首先进行DNS域名解析（本地浏览器缓存、操作系统缓存或者DNS服务器）

2、三次握手建立 TCP 连接

3、客户端发起HTTP请求

4、服务器响应HTTP请求，

5、客户端解析html代码，并请求html代码中的资源

6、客户端渲染展示内容

7、关闭 TCP 连接

- DNS劫持和HTTP劫持