网络部分讲义

一 网络入门与基础参数

第一部分：网络是什么？为什么要用网络？

1.1 网络的基本概念

什么是网络？
想象一下电话系统：每个人有电话号码，可以通过电话线互相通话。计算机网络也是这样，只不过传输的是数据而不是声音。

网络的本质：

- 连接多台计算机的设备
- 让它们可以共享信息、资源
- 就像现实中的道路网连接各个城市

1.2 为什么需要网络？

没有网络的世界：

- 每台电脑都是孤岛
- 文件需要用U盘拷贝来拷贝去
- 打印机每台电脑都要连接一台
- 信息传递非常麻烦

有网络的世界：

- 资源共享：多台电脑共用一台打印机
- 信息传递：发邮件、传文件瞬间完成
- 远程访问：在家也能访问办公室的电脑
- 协同工作：多人同时编辑同一个文档

第二部分：网络如何工作？——从寄信说起

2.1 用寄信理解网络通信

寄信的过程：

> 写信 → 装信封 → 写地址 → 投递 → 邮局分拣 → 运输 → 收件人邮局 → 投递 → 收信人拆信

网络通信的对应关系：

> 数据 → 数据包 → IP地址 → 发送 → 路由器 → 网络传输 → 目标路由器 → 接收 → 应用程序处理

2.2 网络通信的基本要素

需要解决的核心问题：

1. **地址问题**：数据要发给谁？（IP地址）
2. **路径问题**：数据怎么走？（路由）
3. **可靠问题**：数据是否完整到达？（TCP协议）
4. **识别问题**：哪个程序接收数据？（端口）

第三部分：IP地址详解——网络的"家庭住址"

3.1 什么是IP地址？

通俗理解：

- 就像现实中的家庭住址：**北京市海淀区颐和园路5号**
- 在网络世界中：**192.168.1.100**
- 每台联网设备都需要一个唯一的IP地址

技术定义：

- IP地址是Internet Protocol Address的缩写
- 用于标识网络中的设备位置
- 就像电话号码标识具体的电话机

3.2 IP地址的格式

IPv4地址（目前最常用）：

- 由4个数字组成，每个数字0-255

- 用点分隔：`192.168.1.100`
- 举例：`10.0.0.1`、`172.16.254.1`、`192.168.0.1`

为什么是4个数字？

- 每个IP地址实际上是32位二进制数
- `192.168.1.100` = `11000000.10101000.00000001.01100100`
- 分成4段，每段8位，转换成十进制就是0-255

3.3 IP地址的作用

3.3.1 标识设备身份

```
设备A：IP = 192.168.1.100
设备B：IP = 192.168.1.101
设备C：IP = 192.168.1.102
```

3.3.2 确定设备位置

- **网络部分**：标识设备在哪个网络（就像城市名）
- **主机部分**：标识网络中的具体设备（就像街道门牌号）

举例分析：

```
IP地址：192.168.1.100
网络部分：192.168.1
主机部分：100
```

3.4 IP地址的分类

3.4.1 公有IP vs 私有IP

公有IP地址：

- 在互联网上唯一的地址
- 就像真实的家庭住址，全球唯一
- 举例：`8.8.8.8`（Google的DNS服务器）

私有IP地址：

- 在局域网内部使用
- 不同局域网可以重复使用
- 就像小区内的楼号，只在小区内有效
- 范围：
  - `10.0.0.0 - 10.255.255.255`
  - `172.16.0.0 - 172.31.255.255`
  - `192.168.0.0 - 192.168.255.255`

3.4.2 静态IP vs 动态IP

静态IP：

- 固定不变的地址
- 适用：服务器、打印机、网络设备
- 优点：地址固定，便于管理
- 缺点：需要手动配置

动态IP：

- 自动获取，可能变化
- 适用：个人电脑、手机、平板
- 优点：配置简单，自动分配
- 缺点：地址可能变化

3.5 实际生活中的IP地址例子

家庭网络例子：

```
路由器：192.168.1.1
你的电脑：192.168.1.100
手机：192.168.1.101
打印机：192.168.1.50
```

访问网站的过程：

```
你在浏览器输入：www.baidu.com
↓
DNS服务将域名转换为IP：180.101.49.12
↓
你的电脑（192.168.1.100）向百度服务器（180.101.49.12）发送请求
↓
百度服务器回复数据给你的电脑
```

第四部分：其他重要网络概念

4.1 子网掩码——确定"邻居范围"

作用： 区分IP地址中哪些是网络部分，哪些是主机部分

举例：

```
IP地址：192.168.1.100
子网掩码：255.255.255.0

网络地址：192.168.1.0    （整个小区）
广播地址：192.168.1.255  （向全小区广播）
可用地址：192.168.1.1-254（具体住户）
```

4.2 默认网关——网络的"大门"

作用： 本地网络通往其他网络的出口

举例：

```
你的电脑：192.168.1.100
要访问：8.8.8.8（不在同一个网络）
发送数据给：192.168.1.1（默认网关）
网关负责把数据转发到互联网
```

4.3 DNS服务器——网络的"电话本"

作用： 把域名转换为IP地址

查询过程：

问：www.baidu.com的IP是多少？
答：180.101.49.12

4.4 端口

作用：区分统一设备上的不同应用程序

- 80：HTTP（网页）
- 443：HTTPS（安全网页）
- 21：FTP（文件传输）
- 25：SMTP（邮件发送）
- 22：SSH（安全登录）
- 53：DNS（域名解析）
- 123：NTP（时间同步）
- 161：SNMP（网络管理）
- 520：RIP（路由协议）

第五部分：完整网络通信实例

5.1 访问百度首页的全过程

步骤1：准备阶段

你的电脑配置：
IP地址：192.168.1.100
子网掩码：255.255.255.0
默认网关：192.168.1.1
DNS服务器：8.8.8.8

步骤2：域名解析

你输入：www.baidu.com
电脑问DNS：百度地址是多少？
DNS回答：180.101.49.12

步骤3：发送请求

```
源地址：192.168.1.100（你的电脑）
目标地址：180.101.49.12（百度服务器）
经过：192.168.1.1（网关）转发到互联网
```

步骤4：接收响应

```
百度服务器处理请求
返回网页数据
数据经过互联网到达你的网关
网关转发给你的电脑
浏览器显示网页
```

第六部分：动手实践建议

6.1 查看自己电脑的IP地址

Windows系统：

1. 按 `Win + R`，输入 `cmd`
2. 输入 `ipconfig`
3. 查看显示的网络信息

Mac/Linux系统：

1. 打开终端
2. 输入 `ifconfig` 或 `ip addr`
3. 查看网络配置

6.2 理解看到的信息

```
IPv4 地址：192.168.1.100   ← 你的电脑地址
子网掩码：255.255.255.0    ← 确定网络范围
默认网关：192.168.1.1      ← 路由器地址
```

二 网络连接全过程详解：从开机到上网

第一部分：网络连接的基本流程概览

1.1 连接网络的完整步骤

就像入住新家的过程：

1. 找房子（获取IP地址）
2. 确定地址（IP配置）
3. 找到大门（网关设置）
4. 拿到电话本（DNS配置）
5. 开始通信（正常上网）

技术流程：

物理连接 → 获取IP地址 → 配置网络参数 → 建立通信 → 开始上网

第二部分：有线网络连接详细过程

2.1 物理连接阶段

步骤1：硬件准备

电脑网卡 → 网线 → 墙壁网口 → 楼宇交换机 → 校园网核心

步骤2：链路建立

- **网卡检测**：检测到网线插入
- **协商速率**：与交换机协商传输速度（100M/1000M）
- **链路激活**：物理连接就绪

2.2 自动获取IP地址（DHCP过程）

DHCP四步握手：

就像租房的过程：

你的电脑：有人能租给我一个地址吗？ （DHCP Discover）
DHCP服务器：我可以租给你192.168.1.100 （DHCP Offer）

```
你的电脑：好的，我要租这个地址                （DHCP Request）
DHCP服务器：确认，租期2小时，网关是192.168.1.1（DHCP ACK）
```

技术细节：

1. **DHCP Discover**：广播寻找DHCP服务器
2. **DHCP Offer**：服务器提供IP地址租约
3. **DHCP Request**：客户端确认接受
4. **DHCP ACK**：服务器确认分配

获取到的信息：

- **IP地址**：192.168.1.100
- **子网掩码**：255.255.255.0
- **默认网关**：192.168.1.1
- **DNS服务器**：8.8.8.8
- **租期时间**：通常1-8小时

2.3 手动配置IP地址

适用场景：

- 网络中没有DHCP服务器
- 需要固定IP地址的设备（服务器、打印机）
- 特殊网络环境

配置内容：

```
IP地址：192.168.1.100    （必须是空闲地址）
子网掩码：255.255.255.0  （与网络其他设备一致）
默认网关：192.168.1.1    （路由器地址）
DNS服务器：8.8.8.8       （域名解析服务）
```

第三部分：无线网络连接详细过程

3.1 扫描和发现网络

步骤1：扫描可用网络

```
你的手机/电脑：附近有哪些WiFi?
周围AP回复：我是"PKU-WIFI"，信号强度-50dBm
        我是"PKU-SECURE"，信号强度-60dBm
```

步骤2：选择网络

- 显示所有检测到的无线网络
- 显示信号强度和加密类型
- 用户选择要连接的网络

3.2 获取IP地址

- 过程与有线网络相同（DHCP）
- 通过无线AP获取网络配置
- 无线AP：可以想象为灯泡

第四部分：校园网特殊连接过程

4.1 Web认证流程（常见于宿舍网络）

连接过程：

```
1. 连接WiFi → 获取IP地址
2. 打开浏览器访问任意网站
3. 自动跳转到认证页面
4. 输入学号、密码登录
5. 认证成功，开始计时
6. 可以正常上网
```

技术原理：

- **强制门户**：未认证用户所有HTTP请求都被重定向
- **会话管理**：登录后建立会话，定期检查
- **计费系统**：记录上网时长或流量

第五部分：网络连通性测试

5.1 连接建立后的自检过程

检查网关连通性：

```
你的电脑：网关192.168.1.1，你在吗？  （ping网关）
网关：在的，我在这里              （ping回复）
```

检查DNS解析：

```
你的电脑：DNS服务器，www.baidu.com是多少？  （DNS查询）
DNS服务器：是180.101.49.12          （DNS回复）
```

检查外网连通性：

```
你的电脑：8.8.8.8（Google DNS），能通吗？  （ping测试）
8.8.8.8：能通，网络正常              （ping回复）
```

5.2 常见连接问题排查

问题1：无法获取IP地址

症状：显示"受限连接"或"无Internet访问"
可能原因：

- DHCP服务器故障
- IP地址池耗尽
- 网络线路问题

问题2：能获取IP但无法上网

症状：有IP地址但打不开网页
可能原因：

- DNS配置错误
- 网关故障
- 认证未完成

问题3：无线连接频繁断开

症状：WiFi时断时续
可能原因：

- 信号强度弱
- 无线干扰
- AP负载过高

第六部分：从点击到网页显示的全过程

6.1 完整的数据流路径

```
你的电脑 → 无线AP/交换机 → 楼宇路由器 → 校园网核心 → 互联网 → 目标网站
```

6.2 访问百度首页的详细过程

步骤1：准备阶段

```
你的电脑配置：
IP: 192.168.1.100
网关: 192.168.1.1
DNS: 8.8.8.8
```

步骤2：域名解析

```
你的电脑 → DNS查询：www.baidu.com的IP?
DNS服务器 → 回复：180.101.49.12
```

步骤3：建立TCP连接

```
你的电脑 → SYN → 百度服务器
百度服务器 → SYN-ACK → 你的电脑
你的电脑 → ACK → 百度服务器
（TCP三次握手完成）
```

步骤4：发送HTTP请求

```
你的电脑 → HTTP GET请求 → 百度服务器
请给我首页HTML文件
```

步骤5：接收响应

```
百度服务器 → HTTP响应 + HTML数据 → 你的电脑
```

步骤6：加载资源

```
浏览器解析HTML，发现还需要CSS、图片等
重复步骤2-5获取所有资源
```

步骤7：显示网页

```
浏览器组合所有资源，渲染显示百度首页
```

第七部分：不同设备的连接特点

7.1 电脑连接网络

Windows系统：

- **网络和共享中心**管理连接
- **ipconfig**命令查看状态
- 支持有线、无线、VPN等多种方式

macOS系统：

- **系统偏好设置-网络**
- **ifconfig**命令查看状态
- 自动切换最佳网络

Linux系统：

- **NetworkManager**或**systemd-networkd**
- **ip addr**命令查看状态

- 强大的命令行配置工具

7.2 手机连接网络

Android/iOS：

- **设置-WLAN**管理连接
- 自动记住密码
- 智能选择信号强的网络
- 移动数据与WiFi自动切换

7.3 其他智能设备

物联网设备：

- 简化连接过程
- 可能使用静态IP
- 专有协议通信

8 常见连接问题解决

无法上网时尝试：

1. **重启设备**：解决临时软件问题
2. **重新插拔网线**：解决物理连接问题
3. **忘记网络重新连接**：解决认证问题
4. **手动设置DNS**：如8.8.8.8或114.114.114.114
5. **联系网络管理员**：解决网络侧问题

总结

网络连接的核心步骤：

1. **物理连接**：插网线或连接WiFi
2. **获取地址**：通过DHCP或手动配置
3. **认证登录**：网页认证或客户端认证

4. **开始通信**：通过网关访问互联网

关键概念回顾：

- **IP地址**：设备的网络身份证
- **网关**：本地网络的大门
- **DNS**：域名到IP的翻译官
- **DHCP**：自动分配地址的服务员

四 协议栈（Protocol Stack）完全详解

第一部分：协议栈的基本概念

1.1 什么是协议栈?

协议栈 = 网络通信的"分层翻译官"

生动比喻：

> 把网络通信比作国际商务合作：
>
> 应用层：公司领导（决定要做什么）
> 传输层：外贸部门（打包、编号、确保送达）
> 网络层：物流公司（规划路线、处理跨境）
> 链路层：本地快递员（小区内送货）
> 物理层：运输工具（卡车、飞机）
>
> 协议栈 = 整个从决策到送达的完整流程体系

1.2 正式定义

协议栈（Protocol Stack）：

- 一组网络协议的分层集合
- 每层解决特定的通信问题
- 层与层之间通过接口交互
- 数据从上到下封装，从下到上解封装

第二部分：经典协议栈模型

2.1 OSI七层模型（理论标准）

```
第7层：应用层 - 为用户应用程序提供网络服务
第6层：表示层 - 数据格式转换、加密解密
第5层：会话层 - 建立、管理、终止会话
第4层：传输层 - 端到端连接、可靠性保障
第3层：网络层 - 寻址和路由选择
第2层：数据链路层 - 介质访问、差错控制
第1层：物理层 - 物理介质、信号传输
```

2.2 TCP/IP四层模型（实际标准）

```
应用层：HTTP、FTP、SMTP、DNS等
传输层：TCP、UDP
网络层：IP、ICMP、ARP
网络接口层：以太网、WiFi等
```

2.3 现代五层模型（教学常用）

```
5. 应用层：应用程序数据
4. 传输层：TCP/UDP头部 + 数据
3. 网络层：IP头部 + 传输层数据
2. 数据链路层：帧头 + 网络层数据 + 帧尾
1. 物理层：比特流传输
```

第三部分：协议栈的工作流程

3.1 数据发送过程（封装）

以发送网页请求为例：

步骤1：应用层

```
// 用户输入网址后
生成HTTP请求：
```

```
GET /index.html HTTP/1.1
Host: www.example.com
```

步骤2：传输层

```
// TCP封装
[TCP头部] + [HTTP请求数据]
↓
源端口：随机高位端口
目标端口：80（HTTP）
序列号、确认号、窗口大小...
```

步骤3：网络层

```
// IP封装
[IP头部] + [TCP数据段]
↓
源IP：192.168.1.100
目标IP：93.184.216.34
TTL、协议类型(TCP)...
```

步骤4：数据链路层

```
// 以太网封装
[以太网头] + [IP数据包] + [CRC校验]
↓
源MAC：AA-BB-CC-DD-EE-FF
目标MAC：路由器MAC地址
```

步骤5：物理层

```
// 转换为比特流
10101001010101001010101010...
通过网线/无线电波传输
```

3.2 数据接收过程（解封装）

逆向过程：

```
物理层：接收比特流 → 组帧
数据链路层：检查MAC地址 → 去除帧头帧尾
网络层：检查IP地址 → 路由决策
传输层：检查端口 → 重组数据段
应用层：解析HTTP请求 → 生成响应
```

第四部分：各层协议详解

4.1 应用层协议家族

Web相关：

```
HTTP/1.1 - 传统网页传输
HTTP/2 - 多路复用，头部压缩
HTTP/3 - 基于QUIC，减少延迟
HTTPS - HTTP + TLS加密
```

文件传输：

```
FTP - 文件传输协议
SFTP - SSH文件传输
TFTP - 简单文件传输（基于UDP）
```

邮件协议：

```
SMTP - 发送邮件
POP3 - 接收邮件（下载到本地）
IMAP - 接收邮件（服务器同步）
```

域名解析：

```
DNS - 域名到IP地址转换
使用UDP端口53，TCP用于大型响应
```

4.2 传输层协议对比

TCP（传输控制协议）：

特点：面向连接、可靠传输、流量控制、拥塞控制
头部：20-60字节
适用：网页、邮件、文件传输

UDP（用户数据报协议）：

特点：无连接、尽力而为、简单快速
头部：8字节
适用：DNS、视频流、在线游戏

4.3 网络层核心协议

IP协议：

IPv4：32位地址，约43亿个
IPv6：128位地址，近乎无限
功能：寻址、路由、数据包分片

配套协议：

ICMP：网络控制消息（ping命令）
ARP：IP地址到MAC地址解析
IGMP：组播管理

4.4 数据链路层协议

有线网络：

以太网协议：CSMA/CD
MAC地址：48位硬件地址
帧结构：前导码+目标MAC+源MAC+类型+数据+FCS

无线网络：

```
802.11系列协议
CSMA/CA冲突避免
支持加密：WEP、WPA、WPA2、WPA3
```

第五部分：协议栈的实际实现

5.1 操作系统中的协议栈

Windows协议栈：

```
// 查看协议栈组件
netsh interface ipv4 show config

// 核心组件：
• TCP/IP协议驱动：tcpip.sys
• 网络适配器驱动
• Winsock API：应用程序接口
```

Linux协议栈：

```
# 查看网络栈信息
cat /proc/net/dev
ss -tunlp

# 核心组件：
• 网络设备驱动
• 内核网络栈
• Socket接口
```

5.2 协议栈配置实例

IP地址配置：

```
# Windows
netsh interface ip set address "以太网" static 192.168.1.100 255.255.255.0 1
```

```
92.168.1.1

# Linux
ip addr add 192.168.1.100/24 dev eth0
```

路由配置：

```
# 添加默认路由
route add 0.0.0.0 mask 0.0.0.0 192.168.1.1

# 查看路由表
netstat -r
```

第六部分：协议栈的交互过程

6.1 跨层交互示例

以访问网站为例的完整栈交互：

应用层发起：

```
// 浏览器输入网址
location.href = "<https://www.example.com>"
```

DNS解析（应用层→传输层→网络层...）：

```
应用层：生成DNS查询请求
传输层：UDP端口53
网络层：IP封装，目标DNS服务器
数据链路层：以太网封装
物理层：发送到网关
```

HTTP连接建立：

```
传输层：TCP三次握手
应用层：TLS握手（HTTPS）
```

```
应用层：HTTP请求/响应
```

6.2 协议栈故障排查

分层诊断方法：

```
应用层问题：网站无法访问，但能ping通
传输层问题：特定端口无法连接
网络层问题：无法ping通目标
数据链路层：本地网络连接断开
物理层问题：网线松动，信号中断
```

常用诊断命令：

```
# 网络层测试
ping 8.8.8.8

# 传输层测试
telnet www.example.com 80

# 路由追踪
tracert www.example.com

# DNS测试
nslookup www.example.com
```

第七部分：现代协议栈演进

7.1 新协议栈架构

HTTP/3协议栈：

```
应用层：HTTP/3
传输层：QUIC（基于UDP）
网络层：IP
```

5G协议栈：

```
应用层：各种5G服务
传输层：增强的TCP/UDP
网络层：移动IP、切片网络
数据链路层：NR（新空口）
物理层：毫米波、大规模MIMO
```

7.2 协议栈优化技术

性能优化：

- TCP加速：窗口缩放、选择性确认
- 零拷贝：减少内核到用户空间数据复制
- 多队列网卡：并行处理数据包
- 协议卸载：硬件处理TCP/IP计算

安全增强：

- TLS 1.3：更快更安全的加密
- DoH/DoT：加密的DNS查询
- 网络层安全：IPsec

第八部分：协议栈的实际意义

8.1 对开发者的意义

网络编程基础：

```
# Socket编程就是直接操作协议栈
import socket

# 创建TCP Socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("www.example.com", 80))
```

理解网络性能：

应用层优化：减少HTTP请求
传输层优化：TCP参数调优
网络层优化：路由选择

8.2 对网络工程师的意义

网络设计：

根据应用需求选择协议：
- 实时音视频：UDP + 前向纠错
- 文件传输：TCP + 大窗口
- 物联网：CoAP + UDP

故障诊断：

分层排查：
1. 物理连接是否正常？
2. IP地址配置是否正确？
3. 路由是否可达？
4. 防火墙是否阻挡？
5. 应用服务是否正常？

总结

协议栈的核心价值：

1. 模块化设计

每层专注解决特定问题
下层为上层提供服务
便于开发和维护

2. 标准化接口

层间接口标准化
不同厂商设备可以互操作
促进技术发展

3. 灵活性和扩展性

可以替换某一层协议
新协议可以平滑引入
适应技术演进

关键记住：

- **协议栈是网络通信的分层架构**
- **数据发送时层层封装，接收时层层解封装**
- **每层有特定的职责和协议**
- **理解协议栈是掌握网络技术的基础**

四 TCP和UDP协议完全详解

第一部分：从生活场景理解传输层协议

1.1 为什么需要传输层协议？

想象一个快递公司的两种服务：

服务A（可靠快递）：

- 确认包裹完好无损
- 签收后给发件人回执
- 包裹丢失会重新发送
- 保证包裹按顺序到达

服务B（普通邮寄）：

- 直接投递，不确认是否收到
- 不保证包裹完整性
- 速度快但可能丢失

- 不按顺序也可能送达

对应关系：

- **服务A = TCP**（可靠传输）
- **服务B = UDP**（快速传输）

第二部分：TCP协议深度解析

2.1 TCP是什么？

TCP（Transmission Control Protocol）

- 中文：传输控制协议
- 特点：**可靠、有序、面向连接**
- 就像打电话：建立连接 → 通话 → 挂断

2.2 TCP的核心特性

2.2.1 可靠性保证

确认机制（ACK）：

```
发送方：发送数据包1
接收方：收到，回复"确认1"
发送方：收到确认，发送数据包2
```

超时重传：

```
发送方：发送数据包，启动计时器
如果超时未收到确认 → 重新发送
```

数据校验：

- 每个数据包都有校验和
- 接收方验证数据完整性
- 损坏的数据包会被丢弃并要求重传

2.2.2 流量控制

滑动窗口机制：

```
发送方窗口大小：4个数据包
发送：包1、包2、包3、包4
接收方：确认包1，窗口滑动
发送：包5、包6...
```

作用： 防止发送方发送太快，淹没接收方

2.2.3 拥塞控制

慢启动：

- 开始时发送少量数据
- 成功则逐渐增加发送量
- 就像开车慢慢加速

拥塞避免：

- 网络拥堵时减少发送量
- 避免加重网络负担

2.3 TCP连接建立——三次握手

场景： 就像打电话前的确认

```
客户端：你好，能听到吗？      （SYN）
服务器：能听到，你也能听到吗？  （SYN-ACK）
客户端：能听到，开始通话吧！    （ACK）
```

技术过程：

1. **SYN**：客户端发送连接请求
2. **SYN-ACK**：服务器同意连接，并确认请求
3. **ACK**：客户端确认服务器的同意

为什么需要三次?

- 两次不够：无法确认客户端准备好了
- 四次多余：效率低下

2.4 TCP连接终止——四次挥手

场景： 礼貌地结束通话

```
客户端：我说完了          （FIN）
服务器：好的，我知道了      （ACK）
服务器：我也说完了        （FIN）
客户端：好的，再见        （ACK）
```

技术过程：

1. **FIN**：主动方发送结束请求
2. **ACK**：被动方确认收到
3. **FIN**：被动方也发送结束请求
4. **ACK**：主动方确认，连接关闭

2.5 TCP数据包结构

```
源端口(16位) | 目标端口(16位)
序列号(32位)
确认号(32位)
数据偏移 | 保留 | 标志位 | 窗口大小
校验和(16位) | 紧急指针(16位)
选项(可选)
数据
```

重要字段说明：

- **序列号**：数据的编号，用于排序
- **确认号**：期望收到的下一个数据包编号
- **标志位**：SYN、ACK、FIN等控制标志
- **窗口大小**：接收方能接收的数据量

第三部分：UDP协议深度解析

3.1 UDP是什么？

UDP（User Datagram Protocol）

- 中文：用户数据报协议

- 特点：**快速、简单、无连接**
- 就像寄明信片：写好就寄，不确认是否收到

3.2 UDP的核心特性

3.2.1 无连接

- 发送前不需要建立连接
- 直接发送数据
- 就像广播，不关心谁收到

3.2.2 不可靠传输

- 不保证数据到达
- 不保证数据顺序
- 不进行错误恢复

3.2.3 开销小

- 头部只有8字节（TCP至少20字节）
- 没有复杂的控制机制

3.3 UDP数据包结构

```
源端口(16位) | 目标端口(16位)
长度(16位) | 校验和(16位)
数据
```

字段说明：

- **源端口**：发送方端口（可选）
- **目标端口**：接收方端口
- **长度**：整个数据包的长度
- **校验和**：数据完整性检查（可选）

第四部分：技术细节深入

4.1 TCP的端口概念

端口作用： 区分同一设备上的不同应用程序

常见TCP端口：

- 80：HTTP（网页）
- 443：HTTPS（安全网页）
- 21：FTP（文件传输）
- 25：SMTP（邮件发送）
- 22：SSH（安全登录）

4.2 UDP的端口概念

常见UDP端口：

- 53：DNS（域名解析）
- 123：NTP（时间同步）
- 161：SNMP（网络管理）
- 520：RIP（路由协议）

第五部分：动手实验建议

5.1 使用Wireshark观察协议

观察TCP三次握手：

1. 打开Wireshark，开始抓包
2. 浏览器访问一个网站
3. 过滤：`tcp && ip.addr == 网站IP`
4. 观察SYN、SYN-ACK、ACK数据包

观察UDP通信：

1. 执行`nslookup baidu.com`
2. 在Wireshark中过滤：`udp.port == 53`
3. 观察DNS查询的UDP数据包