课程名称：计算机网络实验

主讲教师：朱怡安

课程代码：U10P31008.02　　第2次

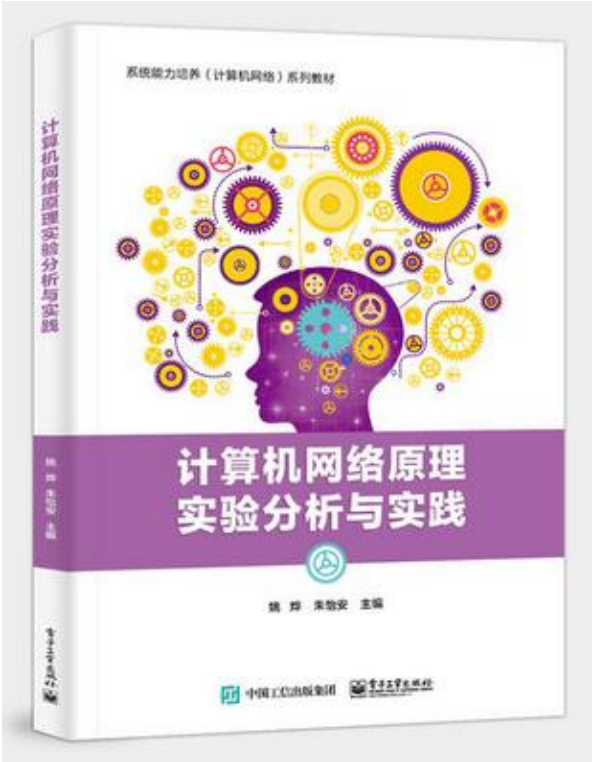
E-MAIL　：zhuya@nwpu.edu.cn

2024 – 2025 学年第一学期

本节实验内容

- 实验内容1：以太网协议分析与实现
  - 以太网协议分析
  - 以太网协议实现
- 实验内容2：TCP端口扫描

（教材-P266）



实验内容1：以太网协议分析与实现

（附参考代码）

实验功能要求：

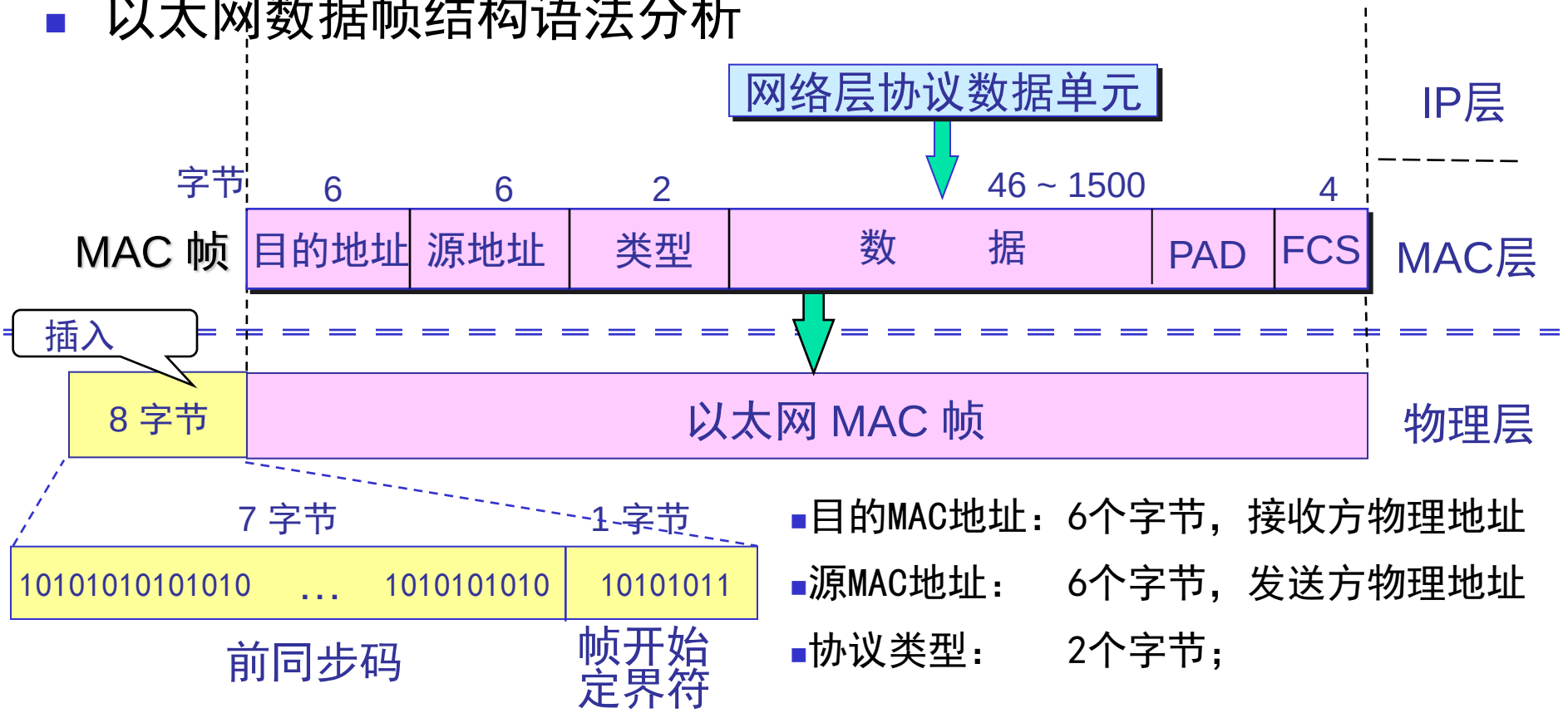
（1）发送端：数据帧构造，发送；

（2）接收端：数据帧接收，解析；

（3）完成一个文件的传输；

实验内容1:协议分析与实现

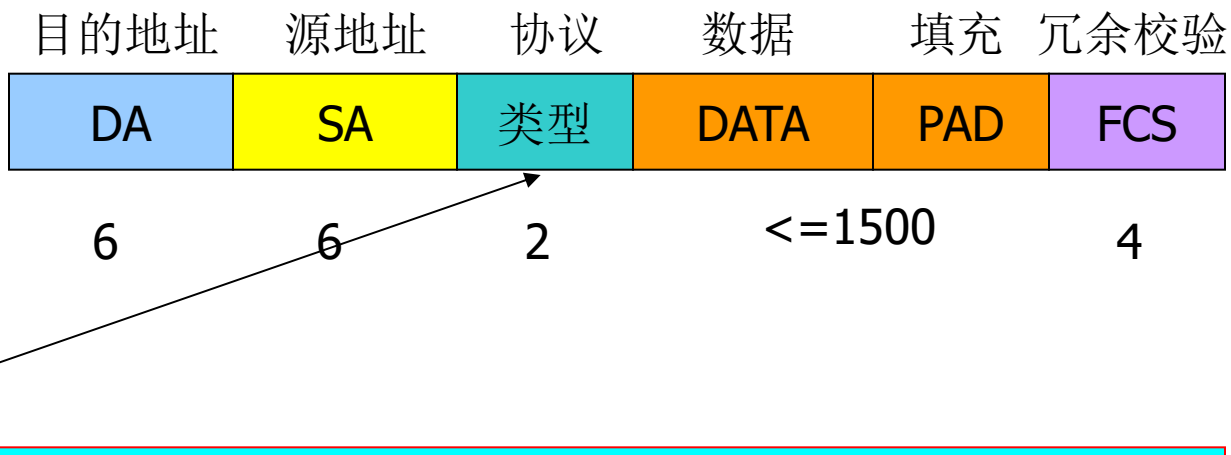
- 一、以太网协议分析
- 以太网数据帧结构语法分析



- 目的MAC地址：6个字节，接收方物理地址
- 源MAC地址：　6个字节，发送方物理地址
- 协议类型：　2个字节；

实验内容1:协议分析与实现

- 一、以太网协议分析
- 以太网数据帧结构语法分析

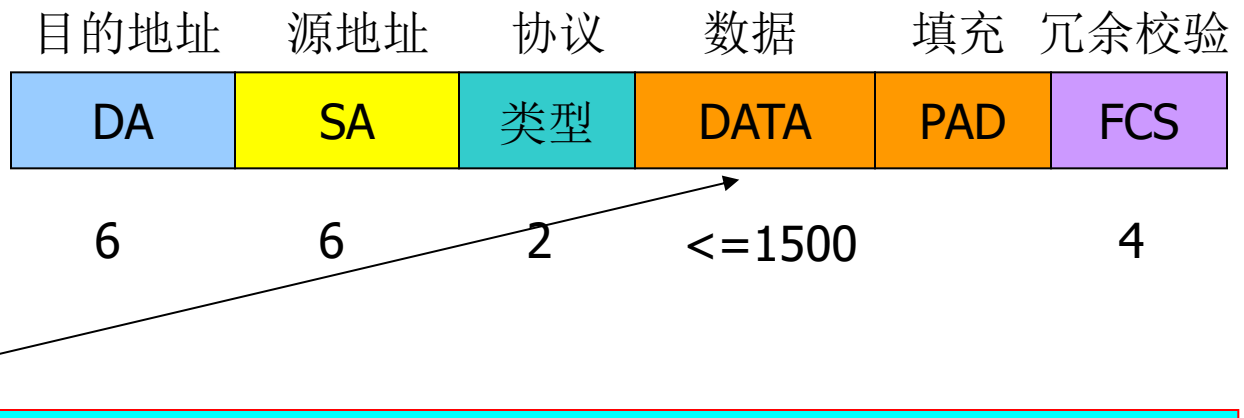


协议类型字段：表示上层协议类型，接收方利用该字段将MAC帧数据（DATA）交付给上层该协议。

- 0X0800：表示上层为IP协议；0X8137：表示IPX协议；0x0806时,表示*ARP协议*；

实验内容1:协议分析与实现

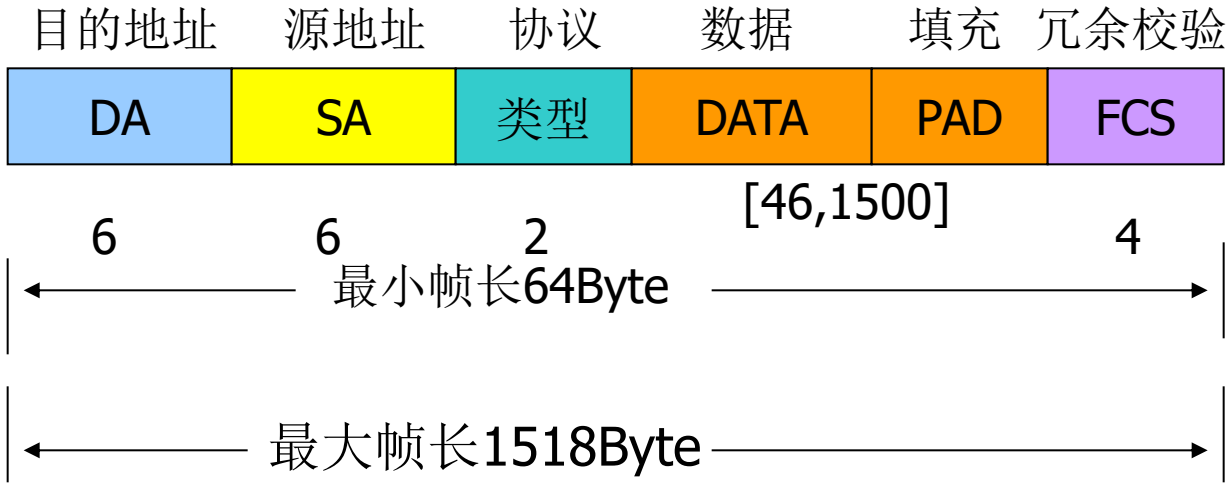
- 一、以太网协议分析
- 以太网数据帧结构语法分析



DATA字段：表示要传送的网络层协议数据单元，网络层协议数据单元应是字节倍数，最大数据长度为1500个字节，最小为46个字节？。

实验内容1:协议分析与实现

- 一、以太网协议分析
- 以太网数据帧结构语法分析



PAD（填充）：

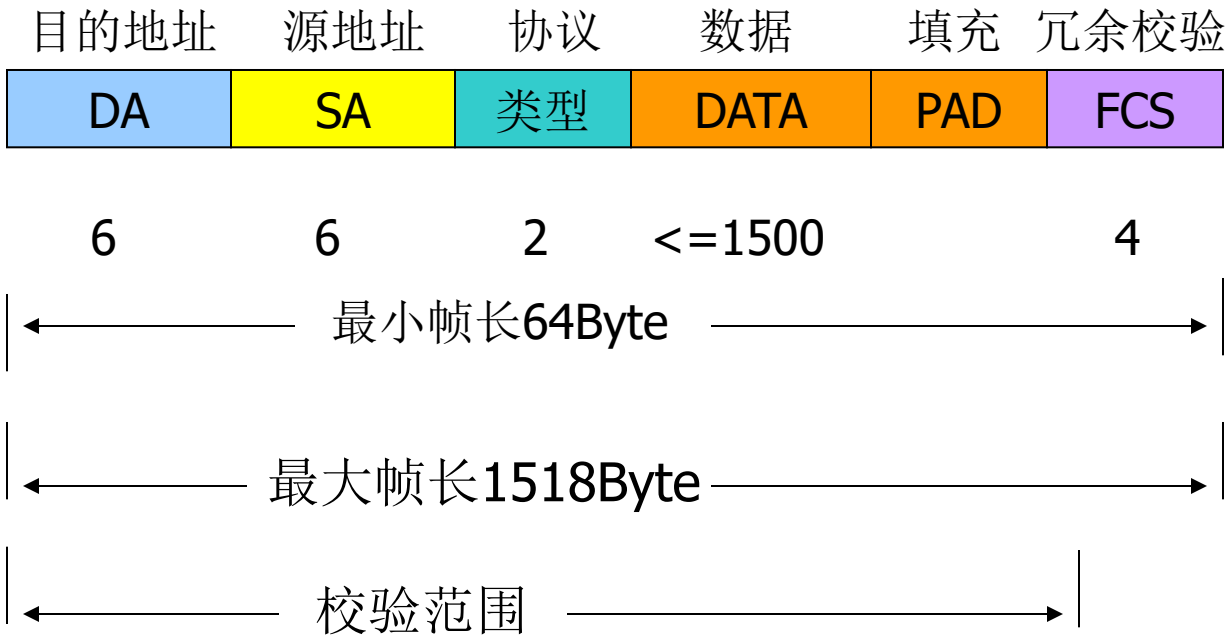
（1）数据帧要求有MTU（46B），最小帧长为64个字节:(18+PDU)

（2）如果实际的PDU数据长度小于46个字节，必须在PAD字段上填充若干字节的0，使PDU和PAD字段的总长度等于46个字节；否则，接收节点会把这个超短帧作为“帧碎片”滤掉，不予接收。

CRC-32　x32+x26+x23+x22+x16+x11+x10+x16+x8+x7+x5+x4+x2+x+1

实验内容1:协议分析与实现

- 一、以太网协议分析
- 以太网数据帧结构语法分析



FCS（帧校验序列）：采用32位CRC校验。
生成多项式：G(X)-32， CRC-32

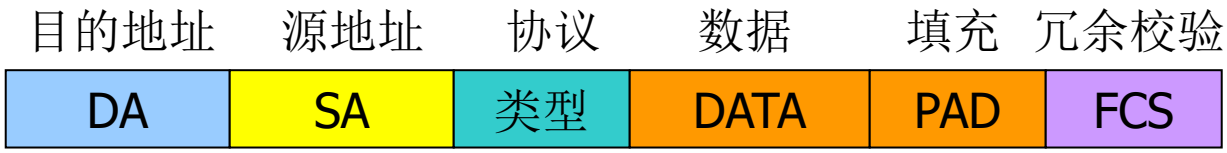
实验内容1:协议分析与实现

- 二、以太网协议实现（依据参考源代码附件）
- 开发接口
  - 实验二需要借助一个软件WINPCAP 来完成。
  - WinPcap是一个网络访问软件。主要功能包括捕获原始数据包、过滤数据包、发送原始数据包以及收集网络通信的统计信息。WinPcap的开发目的是为Win应用程序提供访问网络底层的能力，这意味着它能够直接与网络设备交互，而不需要通过高级的协议栈。
  - WinPcap提供了一个强大的编程接口，使得它在不同操作系统之间容易移植，并且方便程序员进行开发。下载地址http://www.winpcap.org/install/default.htm
  - 但 WinPcap Has Ceased Development，当然也不再对WIN10，win11进行支持了（注意：但不一定不能用，有人也成功用了）。
  - 如用不了，解决方法：1. 使用Npcap， Npcap的下载地址https://npcap.com/#download。2. 仍使用winpcap, 但使用WINDOWS兼容模式对winpcap进行安装 。3.使用老一点的版本，试一把！
  - VC++编译环境或者VISUAL STUDIO环境配置可参考附件。
  - 网络编程高手基本都是使用 WINPCAP或类似软件的高手。

实验内容1:协议分析与实现

- 二、以太网协议实现
- 1、发送方工作流程
- （1）定义数据帧数据结构；
- （2）从文件中读取数据（46-1500字节）；
- （3）计算CRC校验码；
- （4）封装以太网数据帧；
- （5）读取本地网卡列表；
- （7）选择本次通信网卡序号（0,1，2？）；
- （8）初始化本次通信网卡；
- （9）发送数据帧，返回（2），直到文件数据发送完。

```
//ethernet header
struct ethernet_header
{
    u_int8_t dest_mac[6];
    u_int8_t src_mac[6];
    u_int16_t ethernet_type;
};
```



实验内容1:协议分析与实现

- 二、以太网协议实现
- 2、接收方工作流程
- （1）读取本地网卡列表；
- （2）选择本次通信网卡序号（0,1，2？）；
- （3）初始化本次通信网卡；
- （4）接收数据帧；
- （5）数据帧正确性检查：
  - 1）目的地址匹配或者二层广播地址；2）是否碎片帧（小于64B）；3）CRC校验码验证；
- （6）将帧首部及尾部各个字段解析并利用十六进制屏幕打印；
- （7）依据协议类型（0x0800）,将帧数据部分写入文件，返回（4），直到文件接收完成。

检查点1

- 两人协作完成；
- 发送方程序和接收方程序编译成功并运行【先运行接收方程序，后运行发送方程序】；
- 接收方可以接收到数据帧并在屏幕上打印首部和校验字段值；
- 接收方接收的数据帧内容和发送数据帧内容完全相同（首部+数据+尾部）；
- 为了验证传输数据是否正确，选择传输一个视频文件。

助教记录分数

- 两个同学为一组；
- 当完成一个检查点时，主动要求助教检查；
- 助教对检查点完成情况进行记录，记录好完成时间。

实验内容2：TCP端口扫描

（教材–P266）

实验内容2：TCP端口扫描

- 1、实验目的
  - 理解TCP端口扫描的含义及其实现方法，分析并比较采用单进程与多进程实现方法对扫描效率的影响。
- 2、实验原理
  - 基于TCP协议通信模型
  - 端口扫描

int connect(SOCKET Socket, const struct sockaddr FAR* Serveraddr, int Serveraddr-len)。

功能：TCP 客户端向 TCP 服务器发送连接请求，该函数若成功返回，则说明 TCP 连接已经建立完毕，并且仅用于 TCP 客户端。

参数：Socket：表示 TCP 客户端 Socket 套接字。

Serveraddr：表示一个 sockaddr_in 数据结构地址变量，描述服务器端网络地址信息。

Serveraddrlen：表示 sockaddr_in 数据结构变量 Serveraddr 的长度。

返回值：若无错误发生，则 connect()返回 0；否则返回 socket_error 错误，应用程序可通过 WSAGetLastError()函数获取相应错误代码。

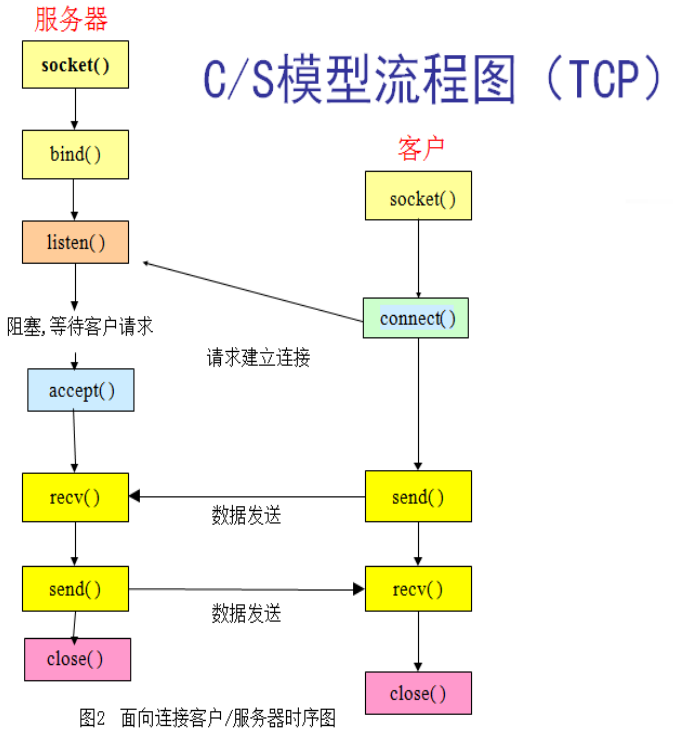


图2 面向连接客户/服务器时序图

实验内容2：TCP端口扫描

- 3、实验原理
  - 端口扫描（单进程编程基本原理）
  - 端口扫描（多进程编程见教材266）

```
int ServePort=20000;
char ServeIP[32]="192.168.1.200";
sockaddr_in  Serveraddr;
Serveraddr.sin_family = AF_INET;
Serveraddr.sin_port = htons(ServePort);
Serveraddr.sin_addr.S_un.S_addr = inet_addr(ServeIP);
if (connect(Socket, (sockaddr*)&addr, sizeof(sockaddr)) == -1)
{
      closesocket(Socket);
       printf("connect failed with error: %d: \n", GetLastError());
       WSACleanup();
       return  -1;
}
```

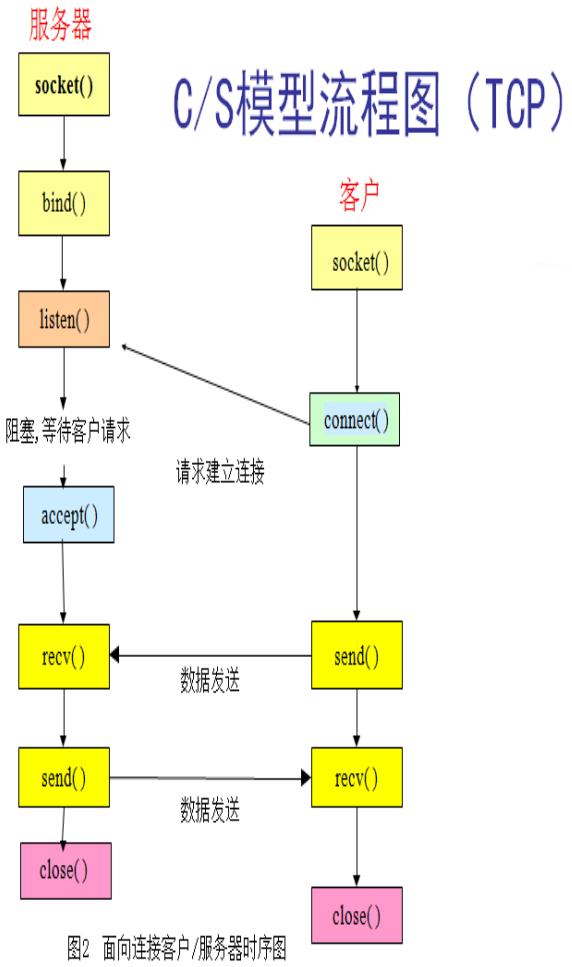


图2 面向连接客户/服务器时序图

助教检查点及记录分数

助教实验检查点

（1）采用单进程完成对0~1024范围内的TCP端口扫描，并检验结果是否正确，将正确的结果在屏幕上打印，统计实验检测完成时间并打印。

（2）采用多进程完成对0~1024范围内的TCP端口扫描，并检验结果是否正确，将正确的结果在屏幕上打印，统计实验检测完成时间并打印。

- 两个同学为一组；
- 当完成一个检查点时，主动要求助教检查，助教对检查点的完成情况进行记录，并记录好完成时间。
- 注意： 不要过多的重复扫描现有的网站，以免被网站或网安人员抓住。

提交实验报告说明

- 1.本次实验课的第一个实验
- 提交：实验报告（参考教材实验模板）+源代码+可执行代码。
- 2.本次实验课的第二个实验
- 提交：实验报告（参考教材实验模板） +源代码+可执行代码。
- 对以上文件压缩(压缩文件名称：姓名-学号-第二次实验)
- 完成时间：下周4晚上12点以前；
- QQ邮箱：　　1312024101 @qq.com