

计算机网络原理实验报告

学院 计算机学院 专业 计算机科学与技术 班级 10012006

学号 2020303245 姓名 夏卓 实验时间: 2022/11/19

一、实验名称:

以太网协议设计与实现

二、实验目的:

了解以太网数据帧结构,掌握数据帧语法设计规则;熟悉以太网数据帧发送和接收的实现流程,能通过编程实现一个简单的以太网协议传输数据;掌握数据帧 CRC 校验方法。

三、实验环境:

Win10, Intelx86, Visual Studio 2019

四、实验内容及步骤:

实验内容:

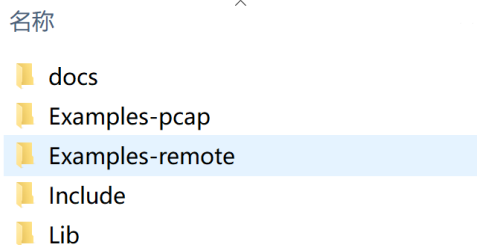
1. 发送方程序和接收方程序编译成功并运行;
2. 接收方可以接收到数据帧并在屏幕上打印首部和校验字段值;
3. 接收方接收的数据帧内容和发送数据帧内容完全相同(首部+数据+尾部)

实验步骤:

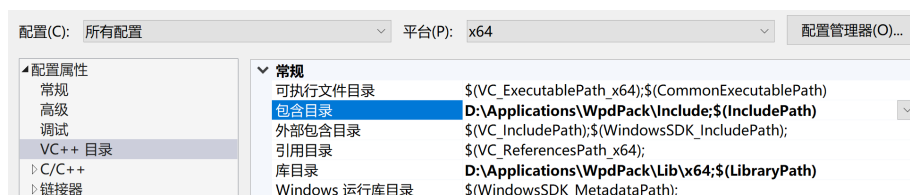
(1) 环境配置

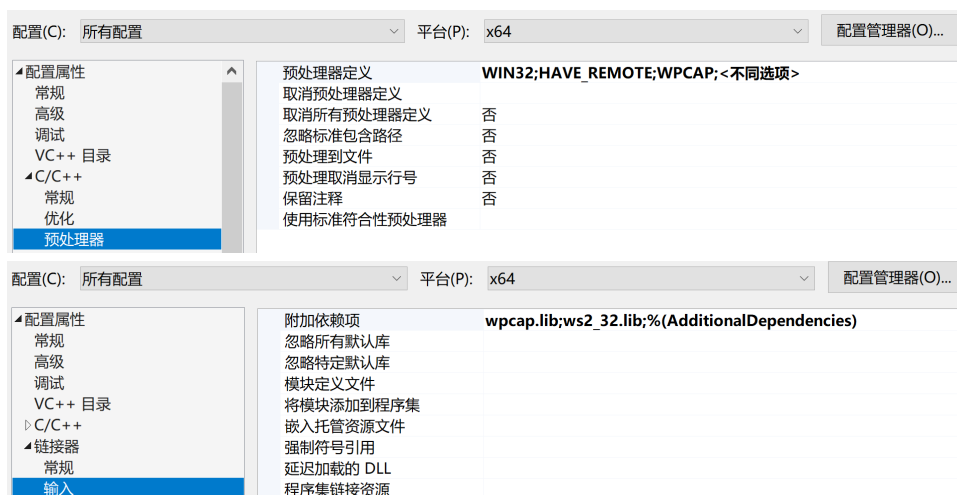
首先安装 WinPcap 运行库, 设置为开机自启。

然后下载 WinPcap 开发包, 得到所需的依赖项以及类库:



配置 VS2019, 在项目属性中引入所需的依赖性, 并添加预处理器命令:

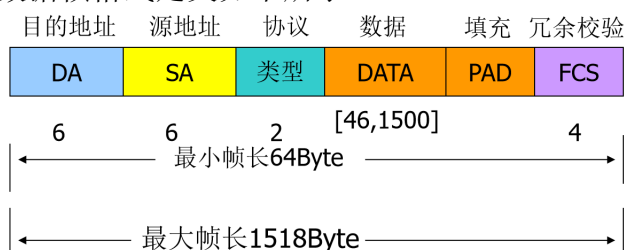




至此，环境配置成功。

(2) 以太网数据帧结构语法设计

数据帧格式定义如下所示：



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，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。

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。

PAD 填充：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 字段的总长度不小于 46 个字节；否则，接收节点会把超短帧作为“帧碎片”滤掉，不予接收。

FCS（帧校验序列）：采用 32 位 CRC 校验。

(3) 理解工业以太网协议实现流程

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；
- ⑥ 选择本次通信网卡序号；
- ⑦ 初始化本次通信网卡；
- ⑧ 发送数据帧，返回②，直到文件数据发送完。

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；

- 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, 包括目的地址匹配、检测是否碎片帧并进行 CRC 校验;
- 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, 将帧数据部分写入文件, 返回④, 直到文件接收完成。

(4) 编写程序并运行

代码参考老师所给程序, 主要进行以下更改:

首先更改数据帧源地址和目的地址为客户端和服务端真实的 MAC 地址
在命令行输入 ipconfig/all, 找到无线局域网适配器 WLAN 的物理地址

无线局域网适配器 WLAN:

```
连接特定的 DNS 后缀 . . . . . :
描述 . . . . . : Intel(R) Wi-Fi 6 AX200 160MHz
物理地址. . . . . : 24-41-8C-F9-1E-74
DHCP 已启用 . . . . . : 是
自动配置已启用. . . . . : 是
```

```
// add destination mac address
hdr->dest_mac[0] = 0x80;
hdr->dest_mac[1] = 0x30;
hdr->dest_mac[2] = 0x49;
hdr->dest_mac[3] = 0xe7;
hdr->dest_mac[4] = 0x64;
hdr->dest_mac[5] = 0xdb;

//add source mac address
hdr->src_mac[0] = 0x24;
hdr->src_mac[1] = 0x41;
hdr->src_mac[2] = 0x8c;
hdr->src_mac[3] = 0xf9;
hdr->src_mac[4] = 0x1e;
hdr->src_mac[5] = 0x74;
```

发送方可以在最后添加输出信息的语句, 比较与接收方信息是否一致

```
struct ethernet_header* ethernet_protocol;
ethernet_protocol = (struct ethernet_header*)buffer;
u_char* mac_string = ethernet_protocol->src_mac;
printf("MAC destination address: %02x:%02x:%02x:%02x:%02x:%02x\n",
    *mac_string, *(mac_string + 1), *(mac_string + 2),
    *(mac_string + 3), *(mac_string + 4), *(mac_string + 5));
mac_string = ethernet_protocol->dest_mac;
printf("MAC source address: %02x:%02x:%02x:%02x:%02x:%02x\n", *mac_string,
    *(mac_string + 1), *(mac_string + 2), *(mac_string + 3),
    *(mac_string + 4), *(mac_string + 5));
printf("Ethernet type: %04x\n", ethernet_protocol->ethernet_type);
printf("data:%s\n", tmp);
printf("crc:%d\n", *(int*)(buffer + size_of_packet - 4));
```

选择网卡时, 需要保持和源 MAC 地址一致, 可以通过在命令行输入 getmac 来获取相应信息:

```
C:\Users\xz276>getmac

物理地址          传输名称
=====
00-E1-7C-68-4B-CD \Device\NPF{787E06B3-4CC8-424D-A01E-2F0E6C0EFBB6}
24-41-8C-F9-1E-74 \Device\NPF{A5DC845C-48EB-4B6F-99C9-761246A1197F}
24-41-8C-F9-1E-78 媒体已断开连接
00-FF-C4-2C-67-FA 媒体已断开连接
0A-00-27-00-00-10 \Device\NPF{A9C4C096-A312-4C92-98B3-B5B128C1981E}
00-50-56-C0-00-01 \Device\NPF{FE4CB657-B901-4C7F-A818-D186E78B3216}
00-50-56-C0-00-08 \Device\NPF{29CF9EE5-C6C6-4E19-AF9E-51E14DA1EB12}
```

五、实验结果:

发送方:

[illegible]

接收方:

[illegible]

可以看到接收方成功接收到了发送方的数据，且进行校验过后，数据并没有丢失，双方 CRC 校验结果一致。

六、实验总结

认识以太网

- “以太网”不是一种具体的网络，而是一种技术标准，它既包含了数据链路层的内容，也包含了一些物理层的内容。例如，以太网规定了网络拓扑结构，访问控制方式，传输速率等。
- 以太网中所有的主机共享一个通信信道，当局域网中的一台主机发出数据后，该局域网中的所有主机都能够收到该数据，只不过每个主机都只关心发送给自己的数据。

认识 MAC 地址

- **MAC 地址**用来识别数据链路层中相连的节点。
- 长度为 48 位，及 6 个字节，一般用 16 进制数字加上冒号的形式来表示，例如：08:00:27:03:fb:19。
- 在网卡出厂时就确定了，不能修改，MAC 地址通常是唯一的（虚拟机中的 MAC 地址不是真实的 MAC 地址，可能会冲突；也有些网卡支持用户配置 MAC 地址）。

MAC 地址和 IP 地址的区别

- IP 地址描述的是路途总体的起点和终点，在路由过程中，源 IP 地址和目的 IP 地址一般是不会变化的。

- MAC 地址描述的是路途上的每一个区间的起点和终点,数据每进行一跳后其源 MAC 地址和目的 MAC 地址都会变化。

教师评语:

成绩: _____ 教师签名: _____ 批阅日期: _____