

廈門大學



信息学院软件工程系

《计算机网络》实验报告

题 目 实验二 利用可见光传输帧的软件

班 级 软件工程 2020 级卓越班

姓 名 庾晓萍

学 号 20420192201952

实验时间 2022 年 3 月 13 日

2022 年 3 月 13 日

填写说明

- 1、本文件为 Word 模板文件，建议使用 Microsoft Word 2019 打开，在可填写的区域中如实填写；
- 2、填表时勿破坏排版，勿修改字体字号，打印成 PDF 文件提交；
- 3、文件总大小尽量控制在 1MB 以下，最大勿超过 5MB；
- 4、应将材料清单上传在代码托管平台上；
- 5、在实验课结束 14 天内，按原文件发送至课程 FTP 指定位置。

1 实验目的

通过完成项目，理解数据链路层传输的基本原理。掌握传输过程中的帧格式设计理念；熟悉传输中的帧与成帧、帧定界符等通信概念，熟悉多方通信中的时分、频分、波分或码分多路复用与解复用等概念，熟悉多方通信中的编址的概念。

2 实验环境

操作系统：Windows 10;

IDE：Visual Studio 2019

编程语言：C++;

3 实验结果

1. 编码

(1) 帧头

```
/*-----帧格式部分-----*/  
  
for (int i = 0; i < length_char; i++) {  
    //一张图的信息区为9184 bits，数据编码的比特数量需要减去以下40 bits，即9144 bits (1143字节)  
    // 帧首定界符(8 bits)、目的地址(8 bits)、源地址(8 bits)、字节计数符(16 bits)  
    if (i == 0 || i % 1143 == 0) {  
        cout << endl << "-----New Frame:-----" << endl;  
        cout << "Your Start Frame Delimiter is :" << endl;  
        for (int k = 0; k < 8; k++) {  
            data.append(1, StartFrameDelimiter[k]); //帧首定界符  
            cout << StartFrameDelimiter[k];  
        }  
        cout << endl << "Your destination address codes are :" << endl;  
        for (int k = 7; k >= 0; k--) {  
            data.append(1, destination_address[k]); //目的地址  
            cout << destination_address[k];  
        }  
        cout << endl << "Your original address codes are :" << endl;  
        for (int k = 7; k >= 0; k--) {  
            data.append(1, original_address[k]); //源地址  
            cout << original_address[k];  
        }  
    }  
}
```

```

cout << endl << "Your byte count notation are : " << endl;
if ( (length_char-i)<1143) {
    int temp= length_char - i;
    //cout << temp << endl; //调试接口
    for (int j = 15; j >= 0; j--) { //字节大端序、位小端序列
        bytecount[j] = temp % 2;
        temp = temp / 2;
    }
    for (int k = 15; k >=0; k--) {
        data.append(1, bytecount[k]); //字节计数符
        cout << bytecount[k];
    }
}
else {
    for (int k = 0; k < 16; k++) {
        data.append(1, fullbytecount[k]); //字节计数符
        cout << fullbytecount[k];
    }
}
}

```

(2) 数据区

```

Binary_Code((unsigned char)input_string[i]); //将字符转换为二进制
for (int j = 0; j < LEN; j++) { //储存在data中, binary_digit的LEN=8
    if (binary_digit[j] == 1) data.append(1, '1');
    else if (binary_digit[j] == 0) data.append(1, '0');
}

```

(3) CRC 编码+二维码绘制

```

//CRC编码+二维码绘制
for (int i = 0; i < img_count - 1; i++) {
    Encode(data.substr(i * (9184), 9184), 0); //第一个参数是code, 对第i段进行编码
}

Encode(data.substr((img_count - 1) * 9184), 1); //第二个参数为1时表示已经读完
cout << "Finish CRC" << endl;
cout << endl;
Get_Video(); //图片转视频
cout << endl << "OK!! " << endl;
system("pause"); //暂停程序执行

```

```

-----New Frame:-----
Your Start Frame Delimiter is :
10101011
Your destination address codes are :
10000000
Your original address codes are :
10000000
Your byte count notation are :
0101110001000000

Finish encode
Data length= 408696
img_count= 45
Finish CRC

Video frame rate defaults to 10

OK!!

```

2. 解码

```

for (; i < srcImages.size(); i += 3) {
    temp = Code_Translate(srcImages[i], verify); //将图片转换为代码
    if (temp.empty()) continue;
    else {
        string headverify2 = temp.substr(0, 40); //头部校验
        Head_Verify(headverify2);
        string final2 = temp.substr(40); //最后存入的真正编码
        code_array.push_back(final2); //将读到的编码存入编码字符串数组中
    }
}

```

```

-----Frame Head-----
Read frame start delimiter successfully
Binary encoding of the byte counter:1110111000100000

-----Frame Head-----
Read frame start delimiter successfully
Binary encoding of the byte counter:1110111000100000

-----Frame Head-----
Read frame start delimiter successfully
Binary encoding of the byte counter:1110111000100000

-----Frame Head-----
Read frame start delimiter successfully
Binary encoding of the byte counter:1110111000100000

-----Frame Head-----
Read frame start delimiter successfully
Binary encoding of the byte counter:1110111000100000

```

4 实验代码

本次实验的代码已上传于以下代码仓库：

https://github.com/ryanregal/Exp_ComputerNetwork

5 实验总结

一、 实验手册中的问题：

1、在实验中的，帧格式是什么？各个字段的作用是什么？长度范围是多少？

这里帧格式指的是二维码数据区的组织格式：

帧首定界符	目的地址	源地址	长度	载荷	CRC
1 字节	1 字节	1 字节	16 bits	0-1144 字节	4 字节

(1) 帧首定界符：0xCA（1010 1011，位小段序，字节大端序）。用于区分每一帧。

(2) 目的地址：用于指示二维码要传输到输出目录下的哪个文件。长度 1 字节，即 8 个比特。

(3) 源地址：用户输入的测试文件名的二进制形式。用于指示这二维码数据来自哪个文件。长度 1 字节，即 8 个比特。

(4) 长度：数据区字节数的二进制形式。用于表示数据区的长度。长度 16 位。

(5) 载荷：数据。长度范围：0~1144 字节。

(6) CRC：校验码，长度 4 字节，采用 CRC32。

2、在实验中的，如何将数据成帧？其中，帧首定界符是什么？

采用帧首定界符+目的地址+源地址+长度+载荷+CRC+结束符的方式成帧。

帧首定界符：0xCA（1010 1011）

3、在实验中的，采用的是何种多路复用算法？简要说明其基本原理。

采用时分多路复用算法。对于给定的文件目录，读取目录下的输入文件并生成视频。对于视频则逐帧读取数据并输出文件。即不同时间处理不同的信息，是时分多路复用算法。

4、在实验中，编址方案是什么？为什么这么设计？

规定输入的文件名字是 0-255 的十进制数字，对应的地址就是八位二进制数，长度 1 字节。这样设计是为了简便的关联文件名和地址的意义，方便转换。而且地址的长度为 1 字节，大小合适，范围合适。

5、你的编码数据载荷大小范围是什么？设定该上限值的依据是什么？

0~1144 字节。

依据：一张二维码的数据区可容纳 9184 个比特，而帧首界定符（8 bits）+目的地址（8 bits）+源地址（8 bits）+长度（12 bits）占据了 36 比特，所以留给载荷和 CRC 的位置有 9148 比特（1148 字节），其中四字节是 CRC，所以载荷大小范围是 0~1144 字节（载荷应以字节为单位）。

二、实验小结

这次实验是第一次实验的延伸，只需要在代码中增加帧格式的设计，相对来说比较简单，但我还是遇到了一些问题，犯了几个错误。首先是在将目的地址、源地址转换成二进制编码时，将多比特下字节大端序、位小段序忘记了，于是又重新回去改了一遍。其次是通过和老师的交流，我体会到了字符填充的重要性。当标志字节出现在待传送的数据信息当中时，就会被误认为是帧的边界，导致丢包。可以在发送端的数据链路层在出现标志字节的数据段前插入另一个转义字符，而在接收端的数据链路层在把数据送往网络层之前删除这个插入的转义字符。

通过本次实验，我对帧格式的设计更加了解。也明白了视频的帧和传输的帧的区别。视频的帧是物理层的概念，每一帧是一张二维码。传输的帧是数据链路层的概念，在实验中指二维码数据区的内容，不包括定位点，因为定位点就是一个纯粹的物理层的部件。从物理层到数据链路层，要剥除物理层的“报文头”。载荷的大小设置依赖于物理层，二维码数据区可容纳的字节数限制了载荷的大小。