

# 廈門大學



## 信息学院软件工程系

### 《计算机网络》实验报告

题    目 实验三  基于 PCAP 库侦听并分析网络流量

班    级 软件工程 2019 级 1 班

姓    名 姬颖超

学    号 22920192204218

实验时间 2021 年 4 月 2 日

2021 年 4 月 2 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，勿超过 5MB；
- 4、应将材料清单上传在代码托管平台上；
- 5、在学期最后一节课前按要求打包发送至 [cni21@qq.com](mailto:cni21@qq.com)。

## 1 实验目的

通过完成实验，理解数据链路层、网络层、传输层和应用层的基本原理。掌握用 Wireshark 观察网络流量并辅助网络侦听相关的编程；掌握用 Libpcap 或 WinPcap 库侦听并处理以太网帧和 IP 报文的方法；熟悉以太网帧、IP 报文、TCP 段和 FTP 命令的格式概念，掌握 TCP 协议的基本机制；熟悉帧头部或 IP 报文头部各字段的含义。熟悉 TCP 段和 FTP 数据协议的概念，熟悉段头部各字段和 FTP 控制命令的指令和数据的含义。

## 2 实验环境

操作系统：Windows 10；

编程语言：C++；

软件：Winshark，VS2019+WinPcap。

## 3 实验结果

### 3.1 用侦听解析软件观察数据格式

(1) 登录 FTP 时，WinShark 捕获的数据

捕获的数据含义：

No.：捕获数据的编号；

Time：捕获数据的相对时间，从开始捕获算为 0.000 秒；

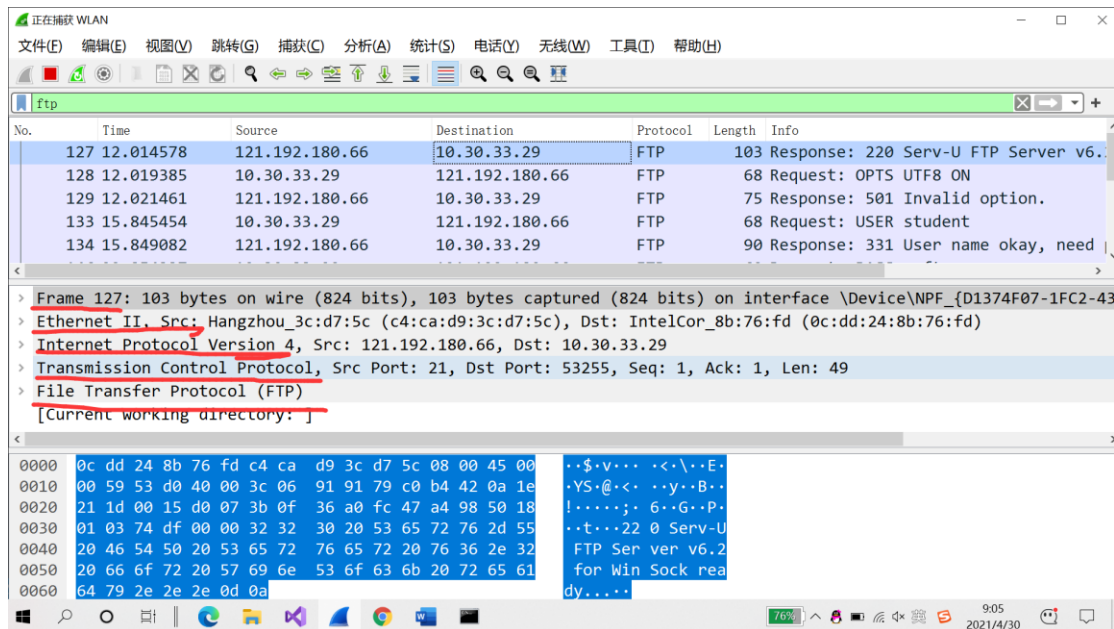
Source：源地址；

Destination：目标地址；

Protocol：协议信息；

Info：数据包的信息。

截图：



Frame：物理层的数据帧概况。

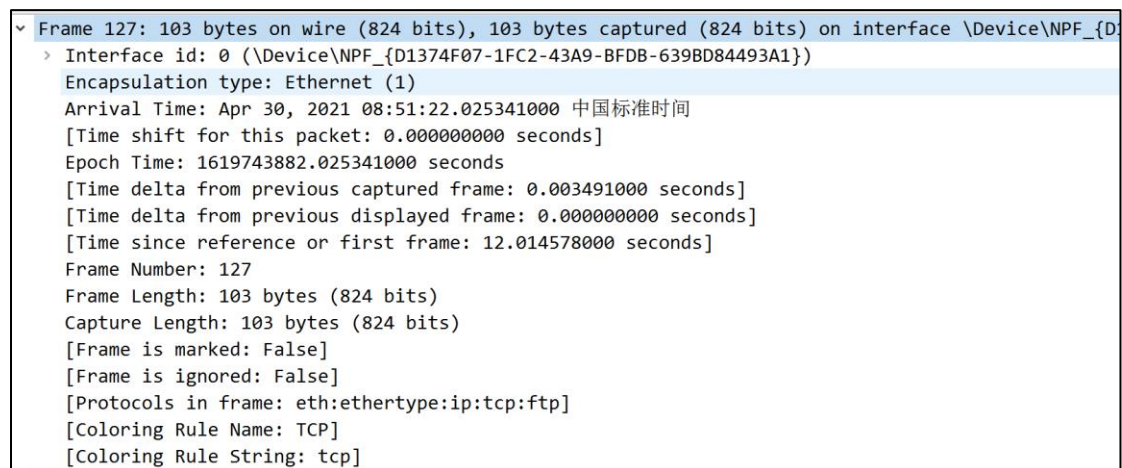
Ethernet II：数据链路层以太网帧头部信息。

Internet Protocol Version 4：互联网层 IP 包头部信息。

Transmission Control Protocol：传输层的数据段头部信息，此处是 TCP 协议。

Hypertext Transfer Protocol：应用层的信息，此处是 HTTP 协议。

## (2) 物理层的数据帧头部信息



## (3) 数据链路层的以太网头部信息

Ethernet II, Src: Hangzhou_3c:d7:5d (c4:ca:d9:3c:d7:5d), Dst: IntelCor_8b:76:fd (0c:dd:24:8b:76:fd)
> Destination: IntelCor_8b:76:fd (0c:dd:24:8b:76:fd)
> Source: Hangzhou_3c:d7:5d (c4:ca:d9:3c:d7:5d)
Type: IPv4 (0x0800)

## (4) 网络层 Ip 头部信息

Internet Protocol Version 4, Src: 121.192.180.66, Dst: 10.30.89.214
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 89
Identification: 0x51c2 (20930)
> Flags: 0x40, Don't fragment
Fragment Offset: 0
Time to Live: 60
Protocol: TCP (6)
Header Checksum: 0x5ae6 [validation disabled]
[Header checksum status: Unverified]
Source Address: 121.192.180.66
Destination Address: 10.30.89.214

## (5) 传输层的 TCP 协议

Transmission Control Protocol, Src Port: 21, Dst Port: 60391, Seq: 1, Ack: 1, Len: 49
Source Port: 21
Destination Port: 60391
[Stream index: 4]
[TCP Segment Len: 49]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 3675804532
[Next Sequence Number: 50 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 3166257754
0101 .... = Header Length: 20 bytes (5)
> Flags: 0x018 (PSH, ACK)
Window: 259
[Calculated window size: 66304]
[Window size scaling factor: 256]
Checksum: 0x0535 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0

## (6) 应用层的 ftp 协议

命令和响应:

```

C:\Users\可可儿>ftp 121.192.180.66
连接到 121.192.180.66。
220 Serv-U FTP Server v6.2 for WinSock ready...
501 Invalid option.
(用户(121.192.180.66:(none)): student
331 User name okay, need password.
密码:
230 User logged in, proceed.
ftp> help
命令可能是缩写的。 命令为:

!          delete          literal          prompt          send
?          debug           ls              put             status
append     dir                  mdelete        pwd             trace
ascii      disconnect         mdir           quit           type
bell       get                mget          quote          user
binary     glob              mkdir         recv          verbose
bye        hash              mls           remotehelp
cd         help              mput          rename
close     lcd              open          rmdir

```

FTP	103 Response: 220 Serv-U FTP Server v6.2 for WinSock ready...
FTP	68 Request: OPTS UTF8 ON
FTP	75 Response: 501 Invalid option.
FTP	68 Request: USER student
FTP	90 Response: 331 User name okay, need password.
FTP	69 Request: PASS software
FTP	84 Response: 230 User logged in, proceed.

### 3.2 用侦听解析软件观察 TCP 机制

用 Wireshark 侦听并观察 TCP 建立和撤除连接的过程

第一次握手数据包：客户端发送一个 TCP，标志位为 SYN，序列号为 0，代表客户端请求建立连接，如下图所示（第一条）：

第二次握手的数据包：服务器发回确认包，标志位为 SYN,ACK. 将确认序号 (Acknowledgement Number) 设置为客户的 ISN 加 1 以.即  $0+1=1$ ，如下图所示（第二条）：

第三次握手的数据包：客户端再次发送确认包(ACK) SYN 标志位为 0,ACK 标志位为 1.并且把服务器发来 ACK 的序号字段+1,放在确定字段中发送给对方。在经过三次握手后和服务器建立了 TCP 连接，如下图所示（第三条）：

TCP	66	61080 → 443	[SYN]	Seq=0	Win=64240	Len=0	MSS=1460	WS=256	SACK
TCP	66	443 → 61080	[SYN, ACK]	Seq=0	Ack=1	Win=65535	Len=0	MSS=1386	
TCP	54	61080 → 443	[ACK]	Seq=1	Ack=1	Win=131584	Len=0		
TLSv1.2	571	Client Hello							
TCP	60	443 → 61080	[ACK]	Seq=1	Ack=518	Win=524544	Len=0		
TCP	1440	443 → 61080	[ACK]	Seq=1	Ack=518	Win=524544	Len=1386	[TCP seg	
TCP	1440	443 → 61080	[ACK]	Seq=1387	Ack=518	Win=524544	Len=1386	[TCP	

### 3.3 用 Libpcap 或 WinPcap 库侦听网络数据

#### (1) VS 与 WinPcap 环境配置

[https://blog.csdn.net/qq\\_17242957/article/details/50954412](https://blog.csdn.net/qq_17242957/article/details/50954412)

#### (2) 关键代码:

##### a. 获取设备列表

```
/*取得列表*/
if (pcap_findalldevs_ex(PCAP_SRC_IF_STRING, NULL, &alldevs, errbuf) == -1)
{
    fprintf(stderr, "Error in pcap_findalldevs:%s\n", errbuf);
    exit(1);
}
```

##### b. 打开设备

```
//打开设备
if ((adhandle = pcap_open_live(d->name, 65536, PCAP_OPENFLAG_PROMISCUOUS, 1000, errbuf)) == NULL)
{
    fprintf(stderr, "\nUnable to open the adapter. %s is not supported by WinPcap\n");
    pcap_freealldevs(alldevs); //释放列表
    return -1;
}
```

##### c. 编译设置过滤器

```
//编译过滤器
bpf_program fcode;
char packet_filter[] = "ip and udp";
if (pcap_compile(adhandle, &fcode, packet_filter, 1, netmask) < 0)
{
    fprintf(stderr, "\nUnable to compile the packet filter. Check the syntax.\n");
    pcap_freealldevs(alldevs);
    return -1;
}
//设置过滤器
if (pcap_setfilter(adhandle, &fcode) < 0)
{
    fprintf(stderr, "\nError setting the filter.\n");
    pcap_freealldevs(alldevs);
    return -1;
}
```

## d. 捕获数据

```
//开始捕捉
while (1)
{
    pcap_loop(adhandle, 1, packet_handler, NULL);
    Sleep(10000); //每10秒捕获一个文件
}
```

## e. 处理数据

```
//获取源MAC、源IP、目标MAC、目标IP、帧长度
string smaddr=bytearray2hex( mh->src_addr ,6);
string siaddr = bytearray2dec(ih->saddr, 4);
string dmaddr = bytearray2hex(mh->dest_addr, 6);
string diaddr = bytearray2dec(ih->daddr, 4);
```

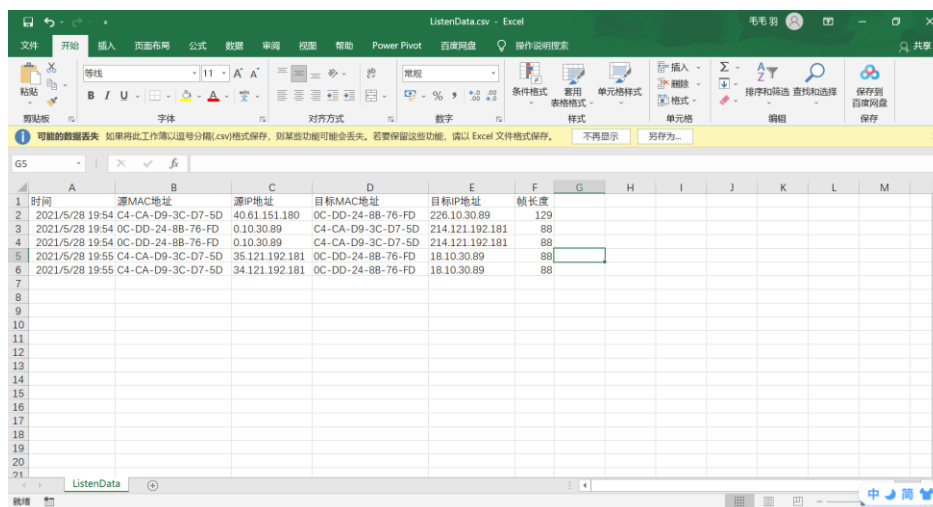
## (3) 实验结果

控制台的输出数据：

```
listening on Network adapter 'Microsoft' on local host...
0C DD 24 8B 76 FD C4 CA D9 3C D7 5D 08 00 45 00
00 73 B9 E4 40 00 34 11 36 28 3D 97 B4 E2 0A 1E
59 D6 1F 40 0F B7
mac_header:
    dest_addr: 0C DD 24 8B 76 FD
    src_addr : C4 CA D9 3C D7 5D
    type: F045
ip_header
    ver_ihl   : 45
    tos       : 00
    tlen      : 0073
    identification: 4000
    flags_fo  : 4000
    ttl       : 34
    proto     : 11
    crc       : 3600
    op_pad    : 00001F40
    saddr     : 28 3D 97 B4  40.61.151.180.
    daddr     : E2 0A 1E 59  226.10.30.89.
```

csv 文件的数据（每十秒捕获一个包）：





	A	B	C	D	E	F	G	H	I	J	K	L	M
1	时间	源MAC地址	源IP地址	目标MAC地址	目标IP地址	帧长度							
2	2021/5/28 19:54	C4-CA-D9-3C-D7-5D	40.61.151.180	0C-DD-24-8B-76-FD	226.10.30.89	129							
3	2021/5/28 19:54	0C-DD-24-8B-76-FD	0.10.30.89	C4-CA-D9-3C-D7-5D	214.121.192.181	88							
4	2021/5/28 19:54	0C-DD-24-8B-76-FD	0.10.30.89	C4-CA-D9-3C-D7-5D	214.121.192.181	88							
5	2021/5/28 19:55	C4-CA-D9-3C-D7-5D	35.121.192.181	0C-DD-24-8B-76-FD	18.10.30.89	88							
6	2021/5/28 19:55	C4-CA-D9-3C-D7-5D	34.121.192.181	0C-DD-24-8B-76-FD	18.10.30.89	88							
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													
20													
21													

### 3.4 解析侦听到的网络数据

#### (1) 用户名、密码所在报文的特征

一般登录名以“USER”开头，口令以“PASS”开头，登录成功以“230”开头，失败以“530”开头。

#### (2) 关键代码：

提取用户名和密码以及成功与否的信息：

```
//选择出command为用户和pass的包
string com;//读取
for (int i = 0; i < 4; i++)
    com+= (char)pkt_data[head + i];
if (com == "USER")
{
    ostringstream sout;
    for (int i = head + 5; pkt_data[i] != 13; i++)
    {
        sout << pkt_data[i];
    }
    ftp.user = sout.str();
}
else if (com == "PASS")
{
    ostringstream sout;
    for (int i = head + 5; pkt_data[i] != 13; i++)
    {
        sout << pkt_data[i];
    }
    ftp.pass = sout.str();
}
```

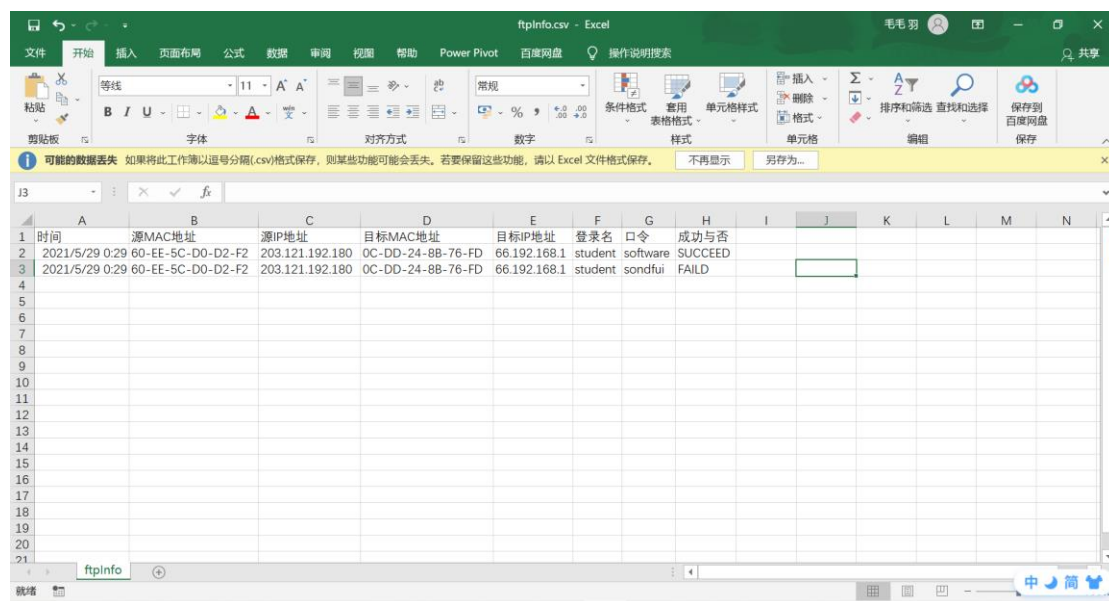
```
else if (com == "230 ")
{
    ftp.info = "SUCCEED";
    print(pkt_data);
}
else if (com == "530 ")
{
    ftp.info = "FAILED";
    print(pkt_data);
}
```

### (3) 结果

控制台输出：

```
listening on Network adapter 'Microsoft' on local host...
2021-05-29 00:29:00, 60-EE-5C-D0-D2-F2, 203.121.192.180, 0C-DD-24-8B-76-FD, 66.192.168.1, student, software, SUCCEED
2021-05-29 00:29:26, 60-EE-5C-D0-D2-F2, 203.121.192.180, 0C-DD-24-8B-76-FD, 66.192.168.1, student, sondfui, FAILED
```

csv 文件：



时间	源MAC地址	源IP地址	目标MAC地址	目标IP地址	登录名	口令	成功与否
2021/5/29 0:29	60-EE-5C-D0-D2-F2	203.121.192.180	0C-DD-24-8B-76-FD	66.192.168.1	student	software	SUCCEED
2021/5/29 0:29	60-EE-5C-D0-D2-F2	203.121.192.180	0C-DD-24-8B-76-FD	66.192.168.1	student	sondfui	FAILED

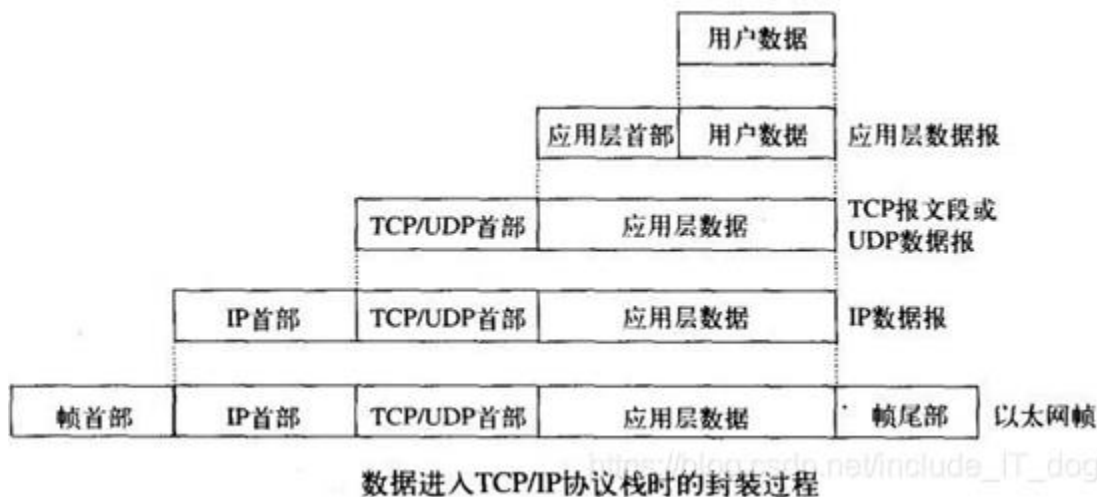
## 4 实验代码

本次实验的代码已上传于以下代码仓库：

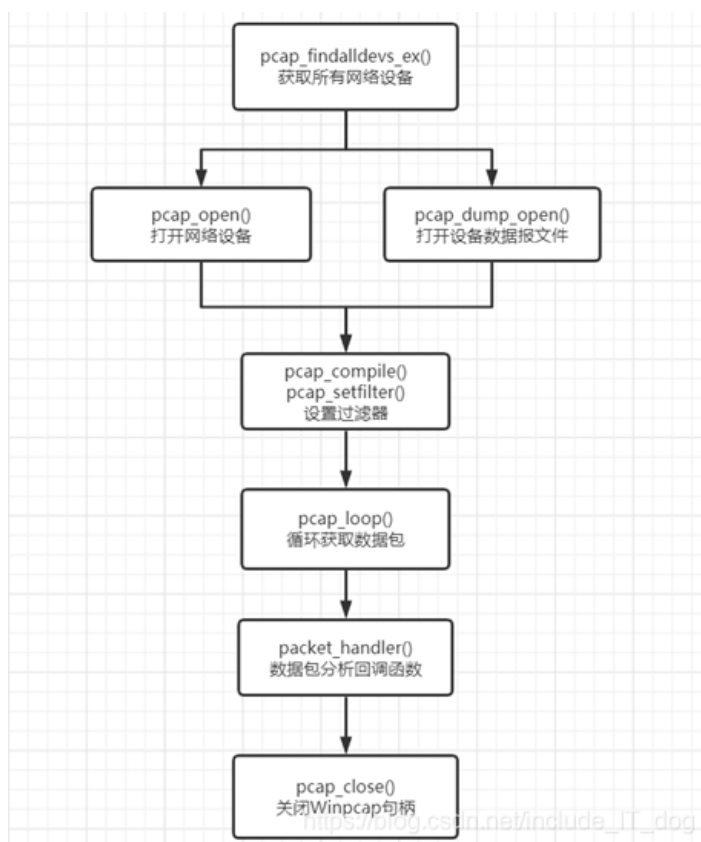
[https://github.com/ikekeer/ComputerNetwork/tree/main/E3\\_4218](https://github.com/ikekeer/ComputerNetwork/tree/main/E3_4218)

## 5 实验总结

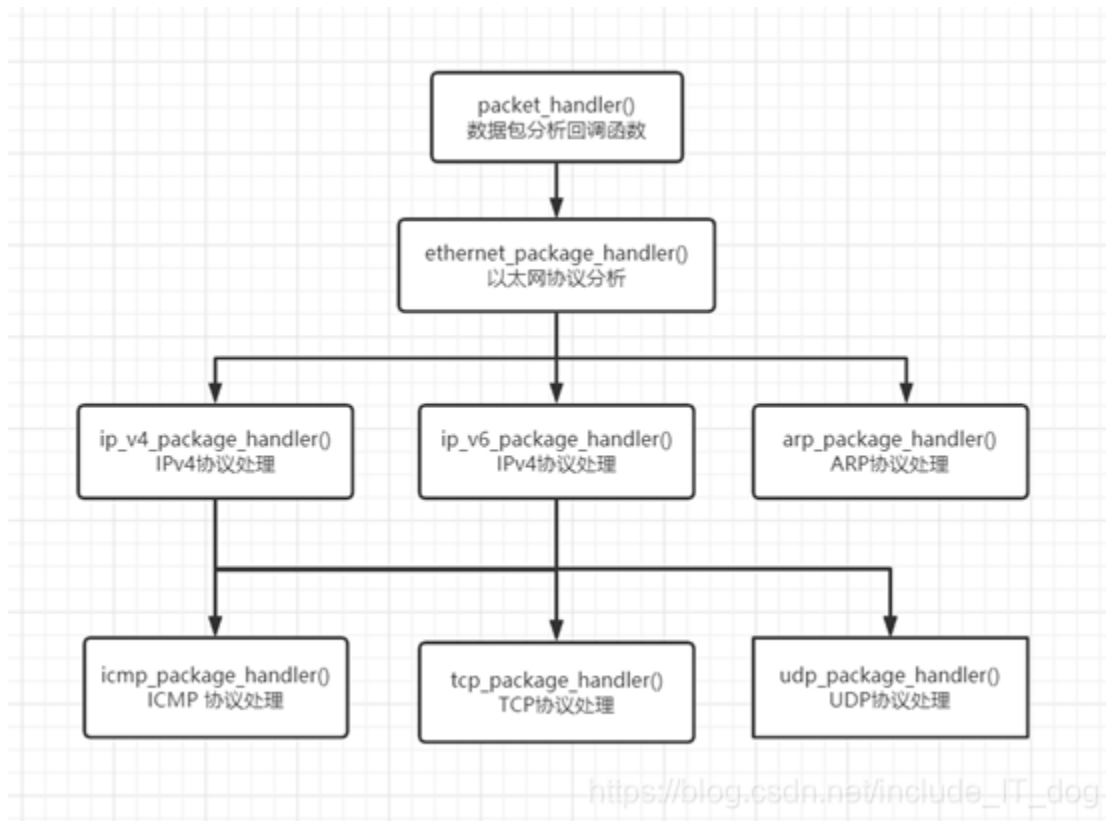
(1) 通过使用 WinShark 捕获数据，进一步熟悉了以太网帧、IP 报文、TCP 段等头部各字段的含义。理解数据封装过程：



(2) 学习了 Winpcap 系统的处理流程：



协议分析流程：



(3) 接触到解析各协议的函数：

`Ethernet_package_handler()`：用来解析以太网帧  
`IP_v4_package_handler()`：用来解析IPv4数据报  
`IP_v6_package_handler()`：用来解析IPv6数据报  
`Arp_package_handler()`：用来解析ARP协议  
`ICMP_package_handler()`：用来解析ICMP协议  
`Tcp_package_handler()`：用来解析TCP报文段  
`Udp_package_handler()`：用来解析UDP报文段

(4) 编写解析 IP 报文、TCP 报文段、以及 FTP 报文的程序。