厦門大學

信息学院软件工程系

《计算机网络》实验报告

题　　目　实验三　侦听以太网和IP报文

班　　级　软件工程2018级1班

姓　　名　方浩南

学　　号　24320182203188

实验时间　2020年3月11日

2020 年　3　月　25　日

1 实验目的

用WinPCAP或libpcap监听并分析以太网的帧，记录目标与源MAC和IP地址。

2 实验环境

操作系统：Ubuntu 18.04、编程语言：C。

3 实验结果

查找并打开一个网卡，名称是字符串而句柄则类似指针。

```
/* find a network interface */
dev = pcap_lookupdev(errbuf);
if (dev == NULL)
{
    printf("Error finding device: %s\n", errbuf);
    return 1;
}
/* open device */
handle = pcap_open_live(dev, BUFSIZ, 1, 0, errbuf);
if (handle == NULL)
{
    printf("Error opening device[%s]: %s\n", dev, errbuf);
    return 2;
}
printf("Opening device[%s] for capture, press 'q' to quit.\n\n", dev);
```

在开始抓取之前先做其它初始化：设置时间和日志目录。

```
/* open log file for writing */
fp = fopen("log.csv", "a");
/* initial time stamp */
len = 0;
gettimeofday(&lasttv, NULL);
```

开始抓包：设置为0/-1代表不限个数，每次抓到的包会进入回调函数。

```
/* capture packets */
pcap_loop(handle, 0, logger, NULL);
```

利用C语言的结构体和指针可以很方便转换和读取以太网的帧。

以太网报头：6字节目的MAC、6字节源MAC、2字节类型。

```
/* 10Mb/s ethernet header */
struct ether_header
{
  uint8_t  ether_dhost[ETH_ALEN];   /* destination eth addr */
  uint8_t  ether_shost[ETH_ALEN];   /* source ether addr */
  uint16_t ether_type;             /* packet type ID field  */
} __attribute__ ((__packed__));
```

在以太网的数据段开始有IP报头：

```
struct iphdr
  {
#if __BYTE_ORDER == __LITTLE_ENDIAN
    unsigned int ihl:4;
    unsigned int version:4;
#elif __BYTE_ORDER == __BIG_ENDIAN
    unsigned int version:4;
    unsigned int ihl:4;
#else
# error "Please fix <bits/endian.h>"
#endif
    uint8_t tos;
    uint16_t tot_len;
    uint16_t id;
    uint16_t frag_off;
    uint8_t ttl;
    uint8_t protocol;
    uint16_t check;
    uint32_t saddr;
    uint32_t daddr;
    /*The options start here. */
  };
```

回调函数的定义：如果类型是IP包则记录。

```
/* handle packet */
void logger(unsigned char *user,
            const struct pcap_pkthdr *pkthdr,
            const unsigned char *packet)
{
    ehdr = (struct ether_header *)packet;
    if (ntohs(ehdr->ether_type) == ETHERTYPE_IP)
    {
```

利用系统相关函数把IP/MAC/时间方便地转为可读形式：

```
/* get packet time stamp in ASCII */
pkttv = pkthdr->ts;
strftime(tm_fmt, sizeof tm_fmt, "%Y-%m-%d %H:%M:%S", localtime(&pkttv.tv_sec));
/* get packet mac address in ASCII */
strcpy(smac, ether_ntoa((struct ether_addr *)ehdr->ether_shost));
strcpy(dmac, ether_ntoa((struct ether_addr *)ehdr->ether_dhost));
/* get packet ip address in ASCII */
ihdr = (struct iphdr *)(packet + ETH_HLEN);
strcpy(sip, inet_ntoa((struct in_addr){ihdr->saddr}));
strcpy(dip, inet_ntoa((struct in_addr){ihdr->daddr}));
fprintf(fp, "%s,%s,%s,%s,%s,%d\n", tm_fmt, smac, sip, dmac, dip, pkthdr->caplen);
```

以秒为单位，累加一秒（或若干）的所有包长度并记录时间戳，计算速率。

```
if (pkttv.tv_sec > lasttv.tv_sec)
{
    rate = (double)len * 8.0 /
           ((double)(pkttv.tv_sec - lasttv.tv_sec) +
            (double)(pkttv.tv_usec - lasttv.tv_usec) / 1000000.0);
    lasttv = pkttv, len = 0;
    if (rate > MAXRATE)
    {
        printf("[%s][%s,%s] SEND %d bytes, current speed: %.2lf Kbps\n", tm_fmt, s
        printf("[%s][%s,%s] RECV %d bytes, current speed: %.2lf Kbps\n", tm_fmt, d
    }
}
len += pkthdr->caplen;
```

通过设置非阻塞的终端输入可以在每次回调时检测输入以退出侦听。

运行结果：

```
Opening device[wlp0s20f3] for capture, press 'q' to quit.

[2020-03-23 22:53:10][d4[redacted]:a4,19[redacted]184] SEND 595 bytes, current speed: 16554.03 Kbps
[2020-03-23 22:53:10][d4[redacted]:6c,58[redacted]] RECV 595 bytes, current speed: 16554.03 Kbps
[2020-03-23 22:53:15][d4[redacted]:6c,19[redacted]1] SEND 87 bytes, current speed: 22651.05 Kbps
[2020-03-23 22:53:15][d4[redacted]:a4,19[redacted]184] RECV 87 bytes, current speed: 22651.05 Kbps
[2020-03-23 22:53:16][d4[redacted]:a4,19[redacted]184] SEND 1270 bytes, current speed: 12441.26 Kbps
[2020-03-23 22:53:16][d4[redacted]:6c,12[redacted]69] RECV 1270 bytes, current speed: 12441.26 Kbps

Device[wlp0s20f3] closed.
```

	A	B	C	D	E	F
1	2020-03-23 22:53:03	d4[redacted]	6c 1[redacted]	d4[redacted]	a4 19[redacted]	184	490
2	2020-03-23 22:53:03	d4[redacted]	a4 1[redacted]	84 d4[redacted]	6c 19[redacted]	2	66
3	2020-03-23 22:53:03	d4[redacted]	a4 1[redacted]	84 d4[redacted]	6c 19[redacted]	2	66
4	2020-03-23 22:53:03	d4[redacted]	a4 1[redacted]	84 d4[redacted]	6c 19[redacted]	1	77
5	2020-03-23 22:53:04	d4[redacted]	6c 1[redacted]	d4[redacted]	a4 19[redacted]	184	77
6	2020-03-23 22:53:04	d4[redacted]	a4 1[redacted]	84 d4[redacted]	6c 19[redacted]	2	74
7	2020-03-23 22:53:04	d4[redacted]	6c 1[redacted]	d4[redacted]	a4 19[redacted]	184	66
8	2020-03-23 22:53:04	d4[redacted]	a4 1[redacted]	84 d4[redacted]	6c 19[redacted]	2	66
9	2020-03-23 22:53:04	d4[redacted]	a4 1[redacted]	84 d4[redacted]	6c 20[redacted]	20	66
10	2020-03-23 22:53:04	d4[redacted]	a4 1[redacted]	84 d4[redacted]	6c 20[redacted]	11	66
11	2020-03-23 22:53:04	d4[redacted]	a4 1[redacted]	84 d4[redacted]	6c 19[redacted]	2	66
12	2020-03-23 22:53:04	d4[redacted]	6c 2[redacted]	0 d4[redacted]	a4 19[redacted]	184	66
13	2020-03-23 22:53:04	d4[redacted]	6c 2[redacted]	1 d4[redacted]	a4 19[redacted]	184	66
14	2020-03-23 22:53:04	d4[redacted]	6c 1[redacted]	d4[redacted]	a4 19[redacted]	184	66
15	2020-03-23 22:53:04	d4[redacted]	6c 1[redacted]	d4[redacted]	a4 19[redacted]	184	473
16	2020-03-23 22:53:04	d4[redacted]	a4 1[redacted]	84 d4[redacted]	6c 19[redacted]	2	66
17	2020-03-23 22:53:04	d4[redacted]	a4 1[redacted]	84 d4[redacted]	6c 19[redacted]	2	66
18	2020-03-23 22:53:04	d4[redacted]	a4 1[redacted]	84 d4[redacted]	6c 19[redacted]	1	77
19	2020-03-23 22:53:04	d4[redacted]	6c 1[redacted]	d4[redacted]	a4 19[redacted]	184	77

4 实验总结

通过这次实验不仅学习了libpcap的使用，更借助结构体切实地理解了以太网的帧格式、报头和地址等等。