实验二 RAW SOCKET 编程与以太网帧分析基础

161130118 尹浚宇

实验目的

学会利用 RAW SOCKET 编程捕获网络上的数据
学会分析捕获到的数据
学会自己构造 icmp 包头并利用套接字编程模仿 OS 的 ping 程序

数据结构说明

本次实验无自定义的数据结构，所用到的系统库提供的结构体如下:

```
struct hostent
{
    char *h_name;  //主机名，即官方域名
    char **h_aliases;  //主机所有别名构成的字符串数组，同一 IP 可绑定多个域名
    int h_addrtype; //主机 IP 地址的类型，例如 IPV4（AF_INET）还是 IPV6
    int h_length;  //主机 IP 地址长度，IPV4 地址为 4，IPV6 地址则为 16
    char **h_addr_list;  /* 主机的 ip 地址，以网络字节序存储。若要打印出这个 IP，
需要调用 inet_ntoa()。*/
};
```

这里该结构体是用于配合函数 srtuct host* gethostbyname(const char* hostname)以达到解析域名获取 IP 地址的作用，即为了使 ping 程序能够 ping 外网所设计.

```
struct sockaddr_in
{
short int sin_family;                    /* 协议族 */
unsigned short int sin_port;          /* 端口号 */
struct in_addr sin_addr;                 /* ip 地址 */
unsigned char sin_zero[8];            /* 为了让 sockaddr 与 sockaddr_in 两个数据结构
保持大小相同而保留的空字节 */
};
```

这里使用该结构体配合 sendto 函数从而给目的主机发送 icmp 包.

```
struct ip
{
u_int ip_v:4; //version(版本)
u_int ip_hl:4; //header length(报头长度)
u_char ip_tos;
u_short ip_len;
u_short ip_id;
u_short ip_off;
u_char ip_ttl;
u_char ip_p;
```

```
u_short ip_sum;
struct in_addr ip_src;
struct in_addr ip_dst;
};
```
该结构体是用于解析收到的 ICMP 所用，其定义形式就是将 ip 协议规定的 ip 头结构转换为 C 语言结构体形式.

struct icmp
```
{
u_char icmp_type; //报文类型
u_char icmp_code; //报文类型子码
u_short icmp_cksum;
u_short icmp_id;
u_short icmp_seq;
char icmp_data[1];
};
```
该结构体对应于 icmp 包头的结构，在填充和解析 ICMP 包头的过程中都有使用.

struct timeval
```
{
time_t          tv_sec;         /* seconds */
suseconds_t     tv_usec; /* microseconds */
};
```
该结构体对应一个时间戳，程序中用于计算 rtt.

程序设计思路及运行流程

抓包程序

1. 创建 RAW SOCKET 套接字
2. 接收网卡上捕获的以太网帧
3. 解析并打印 MAC 地址信息
4. 解析类型信息
    - 4.1 若是 IP 类型，进行 IP 解析
        - 4.1.1 若是 TCP 类型，进行 TCP 解析
        - 4.1.2 若是 UDP 类型，进行 UDP 解析
        - 4.1.3 若是 ICMP 类型，进行 ICMP 解析
    - 4.2 若是 ARP 类型，进行 ARP 解析
    - 4.3 若未识别类型，不进行处理
5. 回到第 2 步

ping 程序

1. 创建 host entry，创建 RAW SOCKET.

2. 利用 hostentry 解析域名获得目的 ip 地址
3. 封装一个 icmp 包头，并向目的 ip 发送
4. 接收目的 ip 传回的 ip 包并解析，计算 rtt 等参数
5. 重复 3-4 步 4 次
6. 打印 ping 结果的统计信息

运行结果截图

抓包程序

```
MAC address: 00:0c:29:cb:88:0d ==> 00:50:56:e7:34:b3
type: IP (0x0800)
Version: 4
Internet Header Length: 20 bytes
Type Of Service: 0x00
Total Length: 0x0054
Identification: 0x8820
Flags: 0b010
Fragment Offset: 0x0000
Time To Live(TTL): 64
Protocol: 1 (icmp)
Header Checksum: 0x58a0
Source: 192.168.72.135
Destination: 119.75.217.109
Type: 8 (Echo Request)
Code: 0
Checksum: 0xf94a
Identifier: 0x0749
Sequence Number: 0x0002
ICMP Data Length: 56 bytes
```

ping 程序

```
user1@ubuntu:~/lab2$ sudo ./ping www.nju.edu.cn
Ping www.nju.edu.cn(202.119.32.7): 56(84) bytes of data.
26 bytes from 202.119.32.7:icmp_seq=0 ttl=128 rtt=1.397 ms
26 bytes from 202.119.32.7:icmp_seq=1 ttl=128 rtt=1.341 ms
26 bytes from 202.119.32.7:icmp_seq=2 ttl=128 rtt=1.552 ms
26 bytes from 202.119.32.7:icmp_seq=3 ttl=128 rtt=1.026 ms
--- 202.119.32.7 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss
```

相关参考资料

Linux 网络编程, socket(PPT)，实验讲义

对比样例程序

抓包程序无对比; ping 程序参考了网上常见的 ping 程序的写法

代码个人创新以及思考

抓包程序解析了多种 IP 包; ping 程序实现了 ping 外网的功能.