Lab6 VPN 设计，实现与分析实验报告

151220098 汤聪

一. 实验目的

本实验主要目的是设计和实现一个简单的虚拟专用网络的机制， 并与已有的标准实现（如 PPTP）进行比较， 进而让学生进一步理解 VPN 的工作原理和内部实现细节。

二. 数据结构说明

1. 以太网头部结构
   包括源 mac 地址和目的 mac 地址和 type

```
typedef struct machead
{
   unsigned char desmac[6];
   unsigned char sourcemac[6];
   unsigned short type:16;
}MAC_HEAD;
```

2. 路由表信息
   包括目的地址，网关，掩码，设备号

```
typedef struct route_item{
   char destination[16];
   char gateway[16];
   char netmask[16];
   char interface[16];
}ROUTE_INFO;
```

3. ARP 表信息
   包括 ip 地址和 mac 地址

```
typedef struct arp_table_item{
   char ip_addr[16];
   char mac_addr[18];
}ARP_TABLE;
```

4. IP 头结构
   和之前实验的 IP 头结构一样

```
typedef struct iphdr
{
  unsigned char ver_hlen;
  unsigned char tos;
  unsigned short data_length;
  unsigned short ident;
  unsigned int flags:3;
  unsigned int offset:13;
  unsigned char ttl;
  unsigned char proto;
  unsigned short checksum;
  unsigned char sourceip[4];
  unsigned char destip[4];
}IP_HEADER;
```

5. ICMP 头结构
   和之前实验的 ICMP 头结构一样

```
typedef struct icmpdata
{
  unsigned char type;
  unsigned char code;
  unsigned short checksum;
  union
  {
    struct
    {
      unsigned short id;
      unsigned short sequence;
    }echo;
    unsigned int gateway;
    struct
    {
      unsigned short unsed;
      unsigned short mtu;
    }frag;
  }un;
  unsigned char data[0];
  #define icmp_id un.echo.id
  #define icmp_seq un.echo.sequence
}ICMP_HEAD;
```

6. 设备结构
   包括设备号和相对应的 mac 地址

```
typedef struct device_item{
  char interface[14];
  char mac_addr[18];
}DEVICE;
```

三. 配置文件说明

1. server1 的配置

IP 表

Eth0	10.0.0.1	00:0c:29:ca:21:ed
Eth1	192.168.0.2	00:0c:29:ca:21:f7

ARP 表

Pc1eth0	10.0.0.1	00:0c:29:ca:21:ed
Server1 eth1	192.168.0.2	00:0c:29:ca:21:f7
Server1 eth0	10.0.0.2	00:0c:29:30:65:bb
Net eth0	192.168.0.1	00:0c:29:55:5b:4d

路由表

172.0.0.2	192.168.0.1	Eth1	192.168.0.2
10.0.0.2	10.0.0.2	Eth0	10.0.0.1

Vpn 表

Pc1	10.0.0.2	10.0.0.1
PC2	10.0.1.2	172.0.0.2

2. server2 的配置

IP 表

Eth0	172.0.0.2	00:0c:29:22:18:c9
Eth1	10.0.1.1	00:0c:29:22:18:d3

ARP 表

172.0.0.2	00:0c:29:22:18:c9	Server eth0
10.0.1.1	00:0c:29:22:18:d3	Server eth1
10.0.1.2	00:0c:29:55:85:57	Pc2 eth0
172.0.0.1	00:0c:29:55:5b:57	Net eth1

路由表

192.168.0.2	172.0.0.1	Eth0	172.0.0.2
10.0.1.2	10.0.1.2	Eth1	10.0.1.1

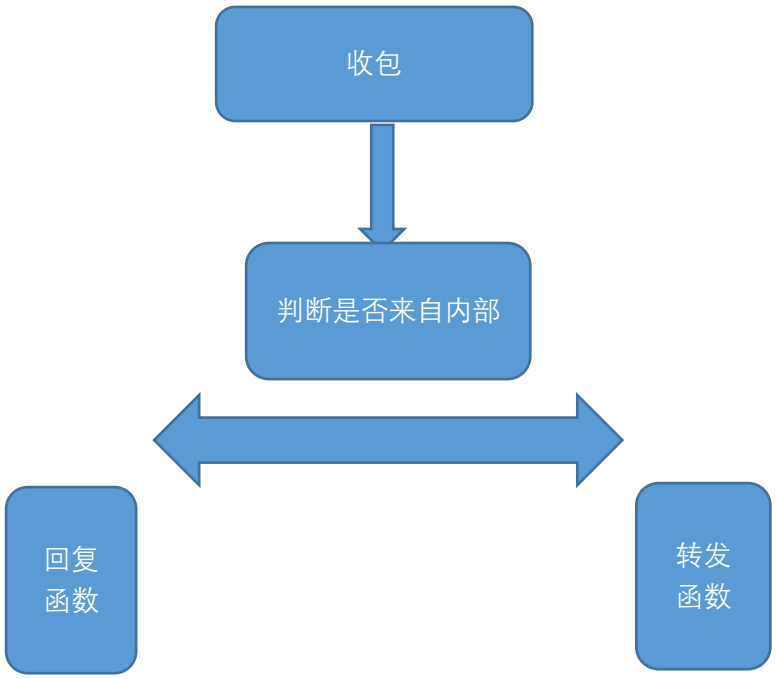
Vpn 表

Pc1	10.0.0.2	10.0.0.1
Pc2	10.0.1.2	172.0.0.2

四. 程序设计的思路以及运行流程

先对各个表项初始化，然后将配置文件读入，然后进行收包，根据包的类型进行转发或者回复。转发函数将收到的包再用一层以太网帧进行封装，并通过相应的网口广播。回

复函数解开一层封装后，通过相应网口广播。



五. 运行结果截图

```
root@ubuntu:~# ping 10.0.1.2
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.
64 bytes from 10.0.1.2: icmp_req=1 ttl=64 time=3.01 ms
64 bytes from 10.0.1.2: icmp_req=2 ttl=64 time=3.26 ms
64 bytes from 10.0.1.2: icmp_req=3 ttl=64 time=2.70 ms
64 bytes from 10.0.1.2: icmp_req=4 ttl=64 time=2.63 ms
64 bytes from 10.0.1.2: icmp_req=5 ttl=64 time=2.62 ms
64 bytes from 10.0.1.2: icmp_req=6 ttl=64 time=2.55 ms
64 bytes from 10.0.1.2: icmp_req=7 ttl=64 time=2.88 ms
64 bytes from 10.0.1.2: icmp_req=8 ttl=64 time=2.66 ms
^C
--- 10.0.1.2 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7075ms
rtt min/avg/max/mdev = 2.552/2.793/3.268/0.236 ms
```

六. 相关参考资料

实验讲义和同学帮助

七. 该程序的应用场景

用于创建特定的私有网络