计算机网络实验报告

**实验六**

**VPN设计、实现与分析**

学号：141220065

姓名：刘博

时间：2016.5.23

**1. 实验目的：**

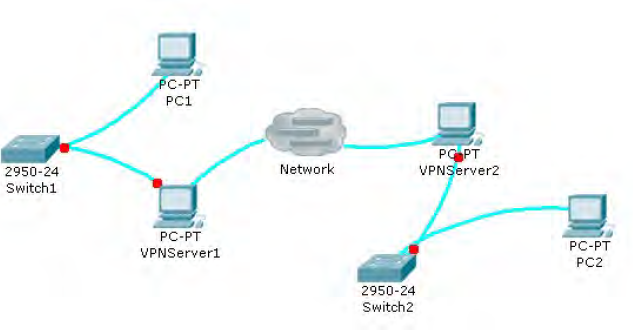
设计和实现一个简单的虚拟专用网络机制，并与已有的标准实现进行比较，进而进一步理解VPN的工作原理和内部实现细节。

**2. 实验环境：**

Ubuntu-12.04（自己电脑）

**3. 网络拓扑配置：**

|  |  |  |  |
| --- | --- | --- | --- |
| 节点名 | 虚拟设备名 | IP | 子网掩码 |
| PC1 | Pc1 | 10.0.0.2 | 255.255.255.0 |
| VPN\_server1 | VPN1 | 10.0.0.1  192.168.0.2 | 255.255.255.0  255.255.255.0 |
| Network | Network | 192.168.0.1  172.0.0.1 | 255.255.255.0  255.255.255.0 |
| VPN\_server2 | VPN2 | 172.0.0.2  10.0.1.1 | 255.255.255.0  255.255.255.0 |
| PC2 | Pc2 | 10.0.1.2 | 255.255.255.0 |



**4. 路由配置文件：**

为所有的节点设置ip地址和路由规则：

PC1：ifconfig eth0 10.0.0.2 netmask 255.255.255.0

Route add default gw 10.0.0.1

VPN\_server1:ifconfig eth0 10.0.0.1 netmask 255.255.255.0

Ifconfig eth1 192.168.0.2 netmask 255.255.255.0

Route add default gw 192.168.0.1

Network:ifconfig eth0 192.168.0.2 netmask 255.255.255.0

Ifconfig eth1 172.0.0.1 netmask 255.255.255.0

Echo 1 > /proc/sys/net/ipv4/ip\_forward

VPN\_server2:ifconfig eth0 172.0.0.2 netmask 255.255.255.0

Ifconfig eth1 10.0.1.1 netmask 255.255.255.0

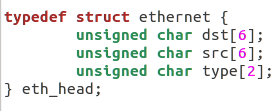
Route add default gw 172.0.0.1

PC2:ifconfig eth0 10.0.1.2 netmask 255.255.255.0

Route add default gw 10.0.1.1

**5. 数据结构说明：**

以太网帧头部：

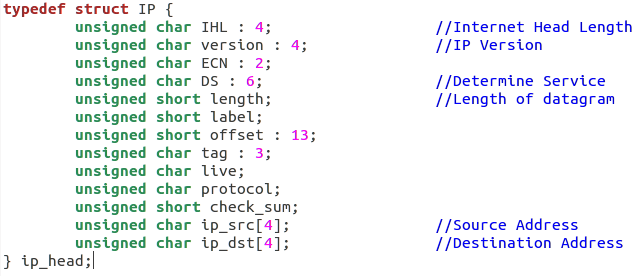


Dst：目的mac地址；

Src：源mac地址；

Type：上层协议类型；

Ip头部：



IHL：网络头部长度；

Version：ip版本号；

Live：ttl数据报生存期；

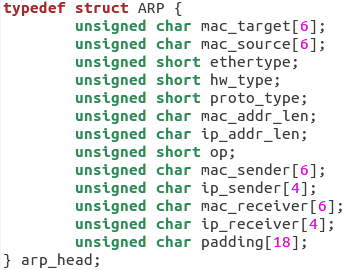
Protocol：上层协议类型；

Check\_sum：校验和；

Ip\_src：源ip地址；

Ip\_dst：目的ip地址；

ARP头部：



Mac\_target：目标mac地址；

Mac\_source：源mac地址；

Hw\_type：硬件类型（网卡编号）；

Proto\_type：协议类型；

Mac\_addr\_len：mac地址长度；

Ip\_addr\_len：ip地址长度；

Op：操作类型；

Mac\_sender：mac发送者地址；

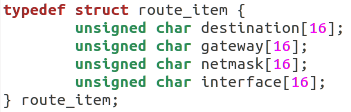
Ip\_sender：ip发送者地址；

Mac\_receiver：mac接受者地址；

Ip\_receiver：ip接受者地址；

Padding：附加段；

路由表项：



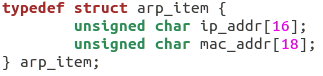
Destination：目的ip地址；

Gateway：网关地址；

Netmask：网络掩码；

Interface：网络接口（网卡名）；

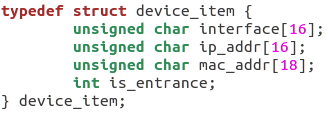
Arp表项：



Ip\_addr：ip地址；

Mac\_addr：mac地址；

Device表项：



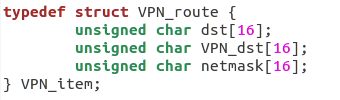
Interface：网卡接口（网卡名）；

Ip\_addr：网卡ip地址；

Mac\_addr：网卡mac地址；

Is\_entrance:网卡是否是外部接口；

VPN路由表项：



1. **配置文件说明：**

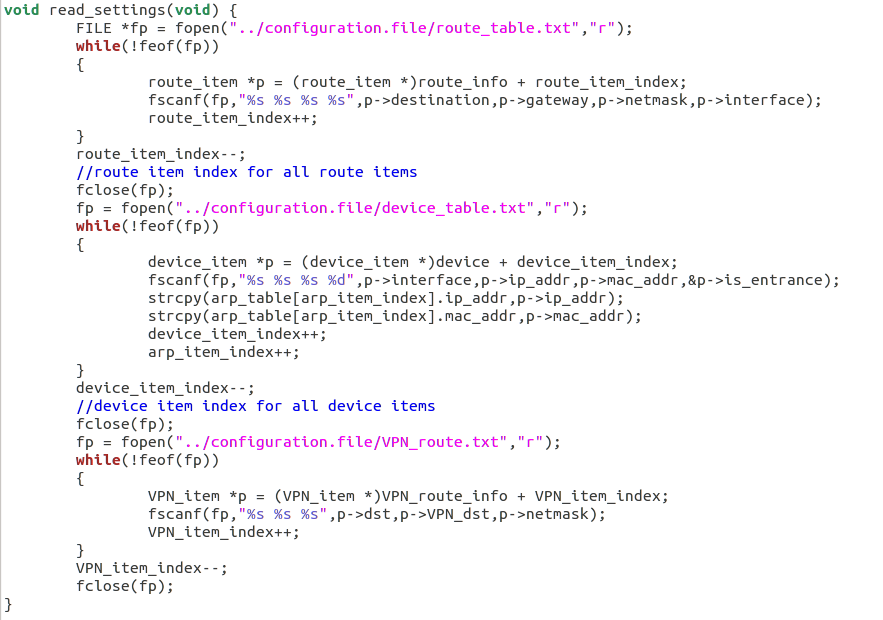
所有配置文件都在configuration文件夹下，其中VPN\_server分别有独立的配置文件；

Route，device表的内部与实验四基本相同，新加入了VPN表，用来保存VPN的虚拟路由信息。

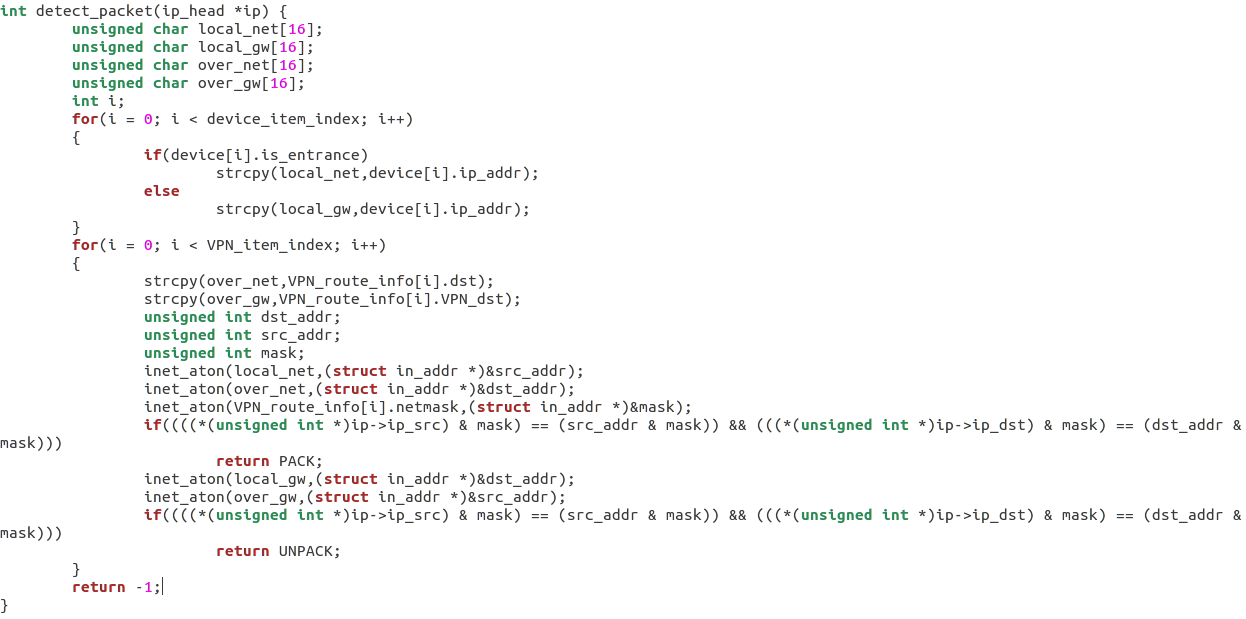
1. **程序设计思路及运行流程：**

设计思路大体与实验四相似，只是在实验四的静态路由转发基础上加上一层VPN的封装即可；

首先路由器进行初始化，直接将路由表和设备表进行读取和赋值，同时加上读取VPN表并保存：



初始化完成后，判断收到的包裹是否是VPN包裹，如果是则进行解包，如果不是，则进行封装后选择转发或不转发；



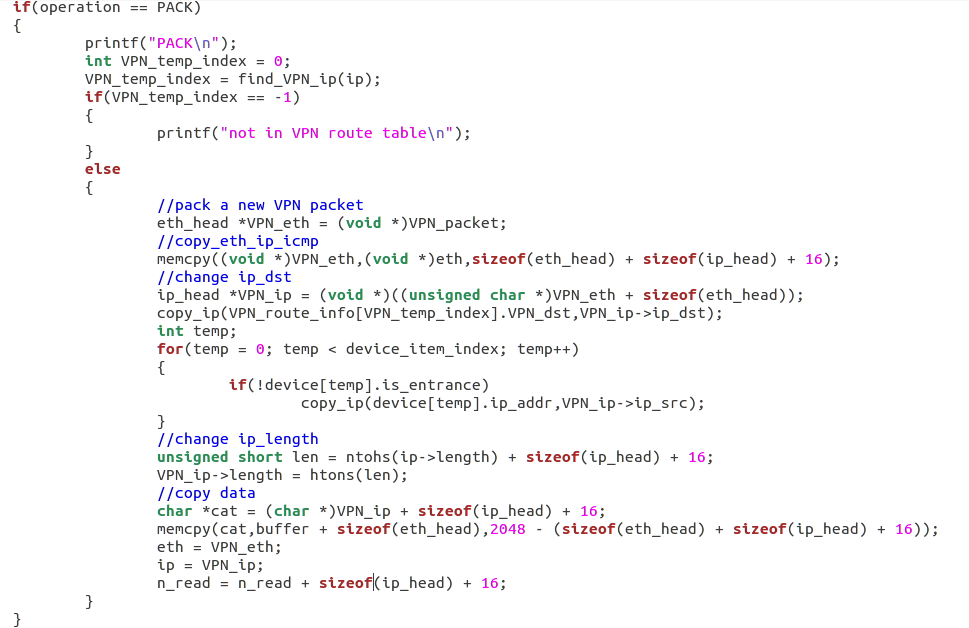
判断方式就是看包裹的目的IP和源IP，如果目的IP和源IP都是在两个VPN子网之内的（例如10.0.0.2和10.0.1.2）那么就判断是封装前的VPN包，将其封装成VPN包即可；

如果目的IP和源IP都是两个VPN的网关地址（例如192.168.0.2和172.0.0.2）那么就判断是封装后的VPN包，则将其解包即可；

如果目的IP和源IP不满足以上两个条件，则判断该包与VPN无关，则正常转发即可。

下面看看如何封装与解封：

1. 封装：



首先明确我们需要将一个原来的数据报（ping 包）作为新的VPN数据包的负载，即将原来的ping的IP包放入新的VPN包的DATA部分，然后添加新的mac地址和ip地址，所以我们可以对原来的数据包进行改造，适当的扩展即可完成：

首先mac部分由于与VPN的关系不大，所以mac地址部分不要进行额外的修改，所以可以将原来的数据包的mac地址完整拷贝到新的数据包的mac地址；

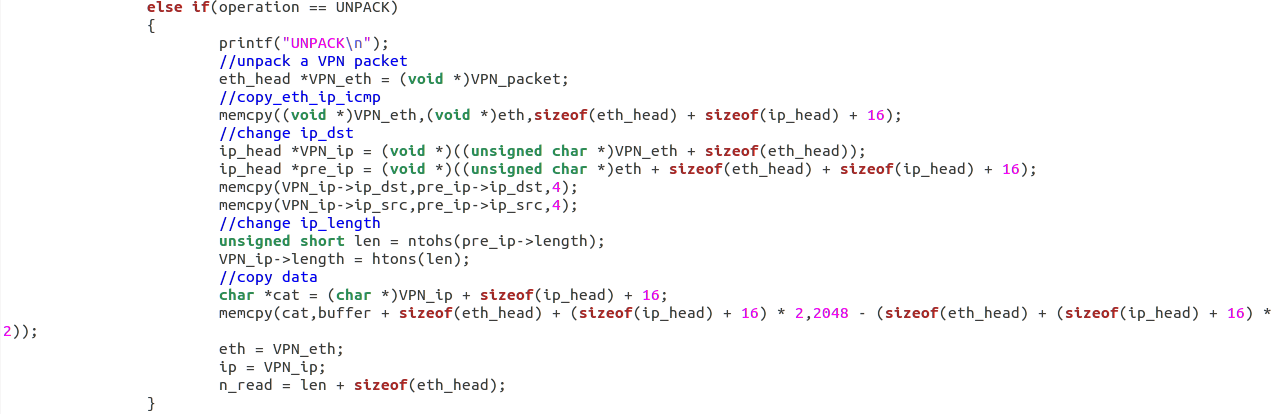
对于ip地址的改造是这里改造的重点，首先在网络上跑的数据包其ip地址均应该是网络上可寻址的ip地址，内网地址需要更改，所以将ip部分的src改成本地VPN的网关地址，将dst改成目标ip的VPN的网关地址；

由于对ip包进行了修改（数据部分增加）所以需要更改ip头部的length部分；

其次需要修改所有ip内部的校验和信息，生存时间（包括icmp的校验和）

然后将生成的完整的VPN包按照标准的路由数据包转发即可（利用实验四已有的部分）注意数据发送的大小已经改变；

1. 解封：



如果我们收到了一个来自外部的VPN数据包，那么我们就需要将它进行解封，我们需要的数据包实际上是该VPN包的负载；所以我们只需要将这个VPN外部的东西舍弃，然后将其负载重新提取；

首先需要将负载部分的ip，icmp部分提取出来并且拷贝到真正的ip，icmp处，然后修改ip长度，校验和，生存时间等信息，将新的ip包作为真正的ip包进行转发即可；

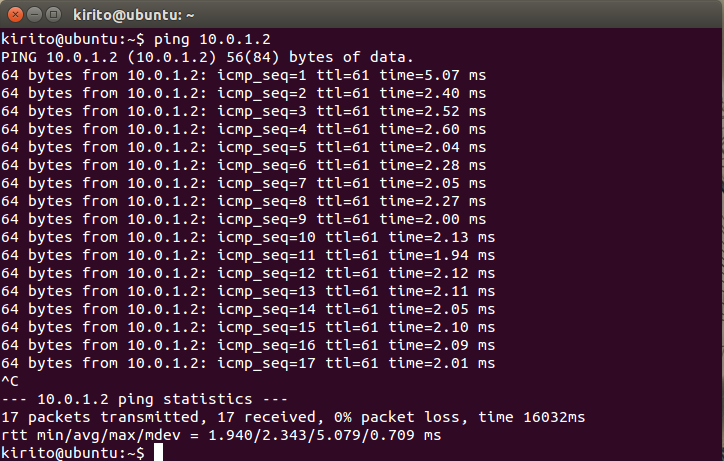
当VPN的封装和解封结束后，剩下的就是正常的路由转发功能，则可以直接利用实验四的代码即可；这里不再进行赘述。

**8．运行结果截图：**

首先将VPN1和VPN2的router程序运行；

然后从pc1 ping pc2：

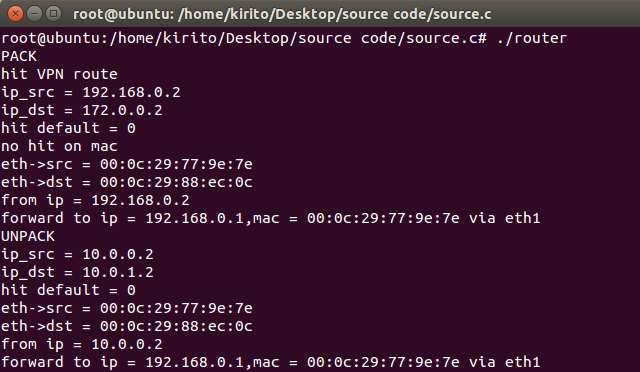
运行结果如下：



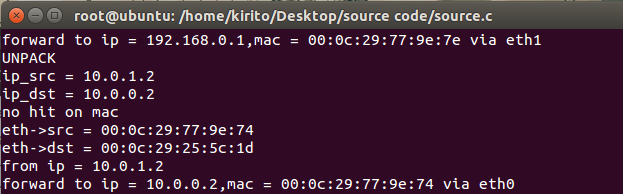
可以看到pc1（10.0.0.2）可以成功ping到pc2（10.0.1.2）

并且从两个pc处看起来所有的数据包就像是直接互相发送的而没有进行修改，实际上是通过了VPN之后才可以建立的连接；

下面看一下VPN上的程序输出：



首先程序PACK一个VPN包，PACK后的VPN包ip地址是192.168.0.2->172.0.0.2然后通过网关192.168.0.1成功发送到了network上；



过了一会程序收到了来自外部的（ping reply）VPN包，所以先UNPACK该包，UNPACK后该ip包的ip地址是10.0.0.2->10.0.1.2然后成功发送到10.0.0.2（pc1）上。

**9. 相关参考资料：**

（百度，google等）；

**10. 对比样例程序：**

本次实验无任何样例程序；

**11. 代码个人创新及思考：**

这次程序主要是注意修改ip地址时需要进行注意，如果仅仅只是修改ip地址而不去修改其他的icmp校验和，ip数据长等信息，则生成的数据包不会通过转发，会变成以太网错误帧，所以要注意进行相应的ip头部，icmp头部改变；

其次对于数据包判断，如何判断一个数据包是否是VPN数据包，对于收到的和发送的数据包如何进行封装和解封都是关键；

其次，如果仅仅只是修改新的VPN包的目的ip地址，而不修改源ip的地址，那么在network上不会进行转发，原因是路由表内没有源ip地址的路由信息，所以必须将新VPN包的ip源地址也改成网关地址（借用），这样才能直接在network上跑。