计算机网络LAB2: RAW SOCKET编程与以太网帧分析

计算机科学与技术系 161220071 李杨

实验要求：

1编写一个抓包程序raw\_socket，抓取ip数据包和arping数据包。

2.编一个ping程序raw\_socket\_ping

实验目的：

初步解raw socket封装和发送以太网帧的功能，在了解icmp包的结构的基础上，实现ICMP包的发送和接收。

数据结构说明：以下皆为调用的已封装好了的结构体

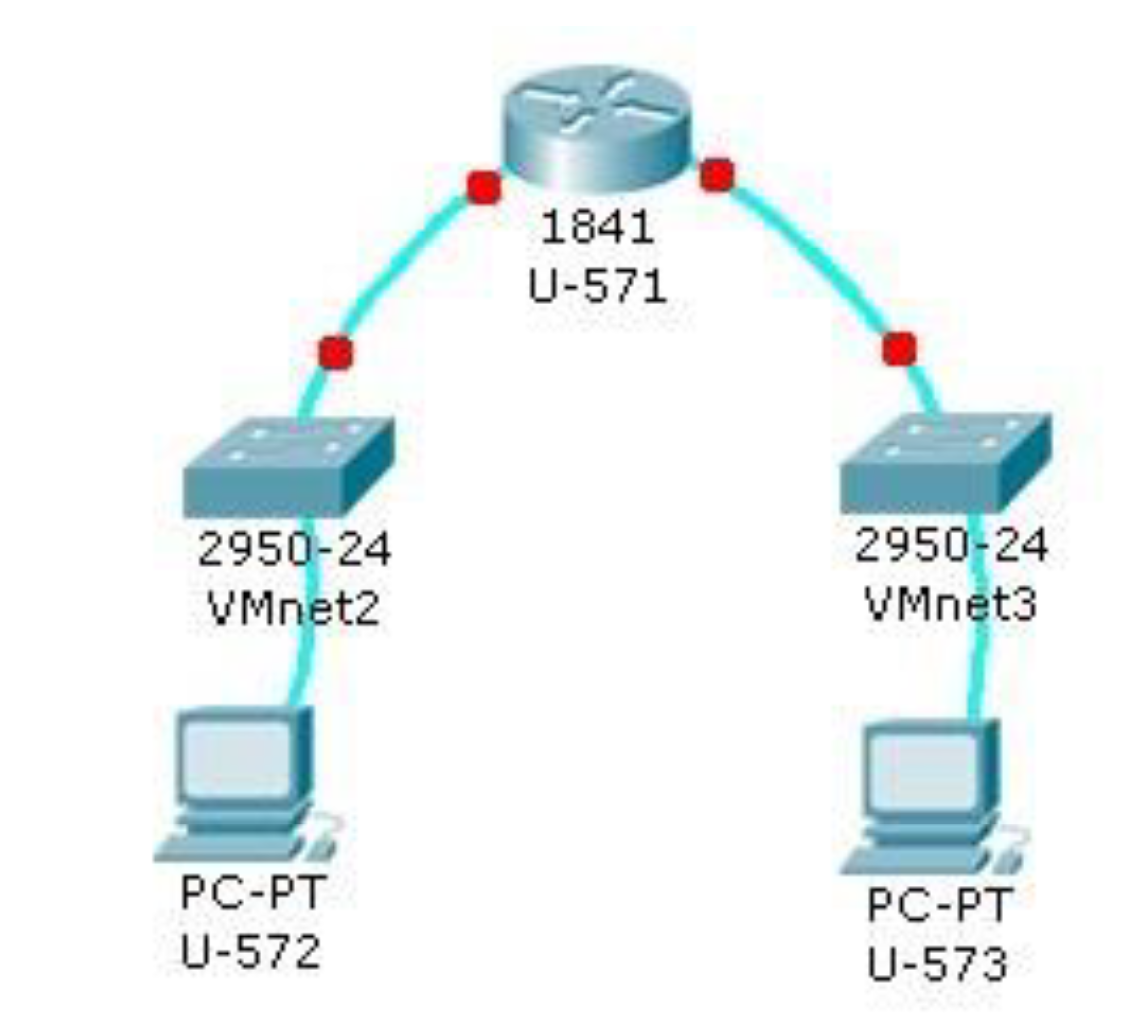
Struct ifreq：用来保存某个接口信息，在get\_local\_ip函数里面使用了这个结构体，目的是获得本地的ip地址.

Struct ockaddr\_in:这是网络通信常用的结构体，在get\_local\_ip和call函数里使用，IPv4专用socket地址,保存目的地址

Structicmp：使用的Linux中ICMP数据结构(<netinet/ip\_icmp.h>)，用来对icmp的设置以及发送与接收

Struct timeval：查询获取系统时间的结构体，用来获取每次收发包所用的时间

环境配置：

 vmnet2为子网2，vmnet3为子网3配置如图所示。

程序设计的思路以及流程：

Raw\_socket程序（抓包）：

按照给定的程序样例，通过socket函数建立连接，并通过recvfrom函数从（已连接）套接口上接收数据，并捕获数据发送源的地址，进而可以打印得到收发端的MAC地址，易知buffer的前12位为mac地址，所以对第十三,十四位进行判断，由信息可知，ip、arp、rarp的类型分别0800,0806,8035，所以可以直接对第十四位进行判断，从而判断出数据包的类型，然后依次打印出各个数据段的内容。

Raw\_socket\_ping程序（ping）：

现在可以先初步搭一个框架出来：检测参数、取（转换）目标IP地址、发报文、接收报文、打印信息。使用gethostbyname ()函数来通过主机名获得ip地址(要发当然需要知道目标IP地址)。要实现发送报文的函数，首先得有报文才行，所以要先实现一个设置ICMP报文的函数。即seticmp函数，这里还要再单独实现一个计算校验和的函数，这样的话，就差rtt的值不知道了，所以再单独写一个getrtt的函数，用来获取时间间隔。这样的话，一个icmp包就包装完成了，下面就是发送了，这里使用sendto函数，就可以将存放在buf里面的报文往sockfd写数据。接着就是收取包了，recvfrom读取sockfd上的数据，buf和len参数分别指定度缓冲区的位置和大小。ICMP数据报是封装IP报文里发送的，IP协议是无连接的，不可靠的协议。所以每次读取数据都要获取发送端的socket地址，即参数src\_addr所指的内容，addlen参数则指定该地址的长度。

当调用recvfrom时，需要设置addlen参数指向一个整数，该整数包含addr所指向的套接字缓冲区的字节长度。返回时，该整数设为该地址的实际字节长度

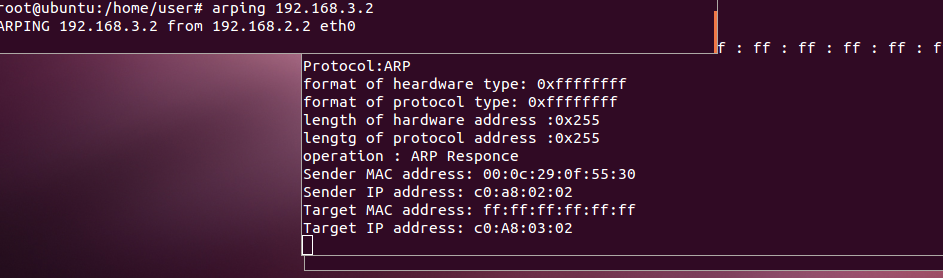
GetRtt用来获取两个时间戳的差，结果保存在第一个参数所指空间处。

Unpack函数用来ip和icmp报头。最后将每个函数整合起来，就可以完成ping程序了。由于要打印本地ip地址，所以在每次unpack打印数据时，调用一次get\_local\_ip函数，来获得本地的ip地址。

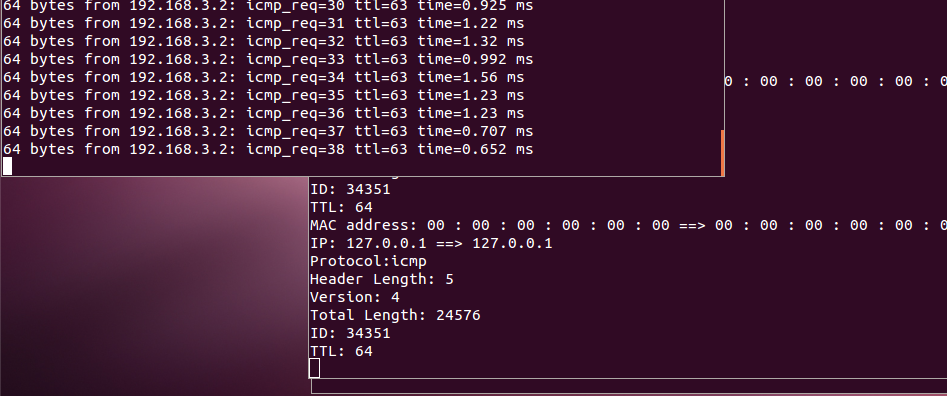
运行结果：

Raw\_socket:

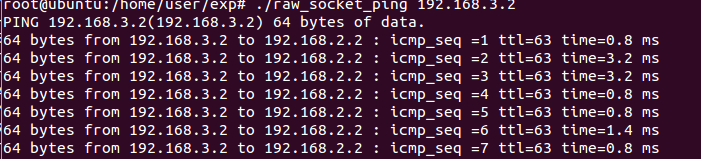
Arping:



Ping：

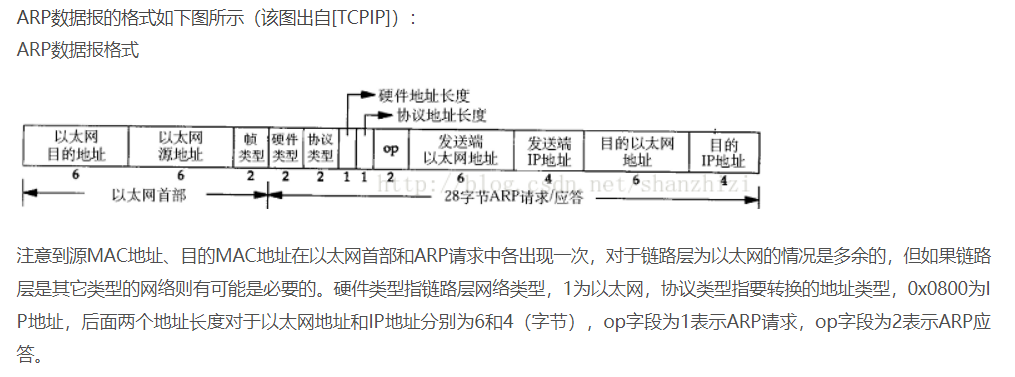


Raw\_socket\_ping:



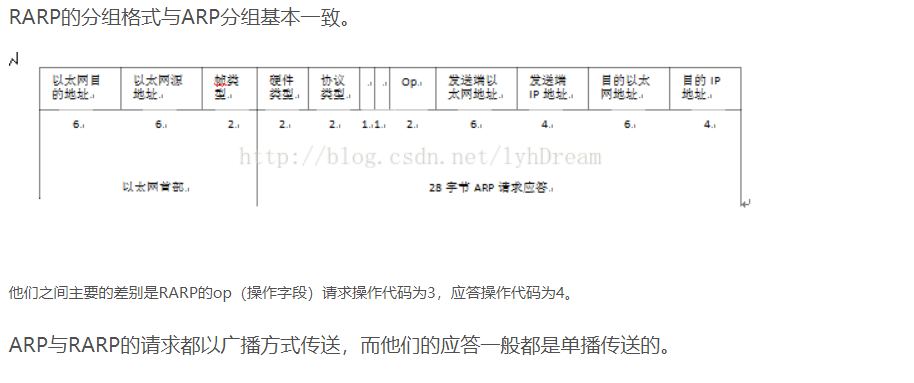
参考资料：

ARP结构的参考：



网址：<https://blog.csdn.net/shanzhizi/article/details/9995489>

RARP结构的参考：



Ip结构以及iphdr结构体详解：

<https://blog.csdn.net/beginning1126/article/details/14057087>

、

ICMP协议的报文格式相关资料：

<https://blog.csdn.net/u011784495/article/details/71743516>

ICMP消息类型：



时间间隔的计算：

<https://blog.csdn.net/u011006622/article/details/52459188>

使用第二种方法。