LAB4 静态路由编程实现

161220071 李杨

实验拓扑图：

3.1/4.2

2.1/4.1

Vm3

Vm4

Vm2

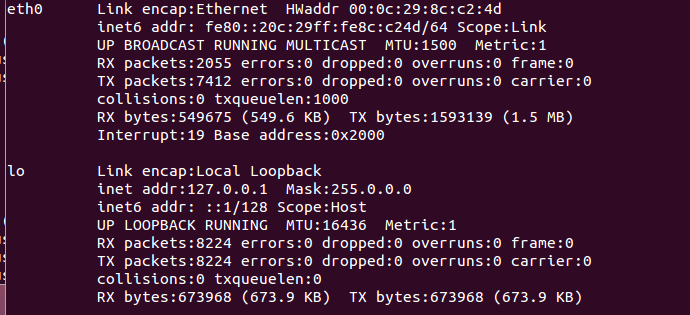
Router1

Router0

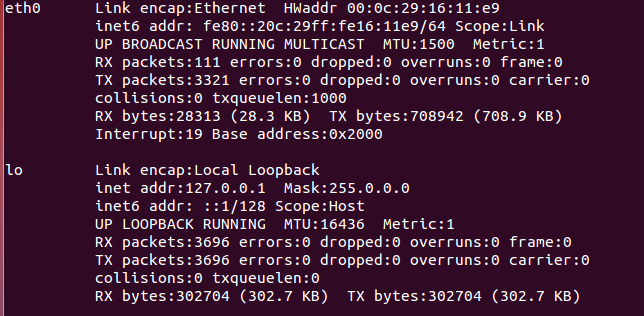
(

IP地址截屏：

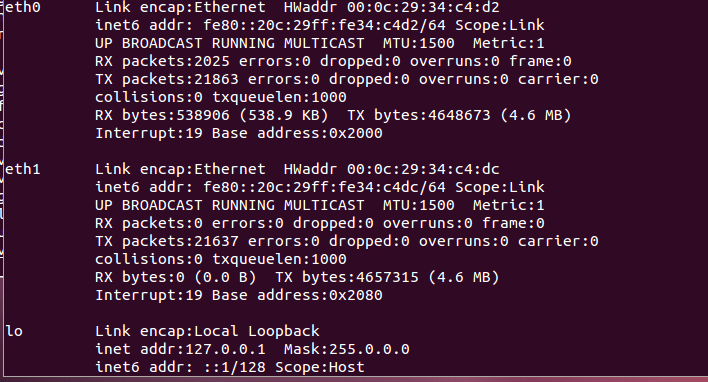
PC0:



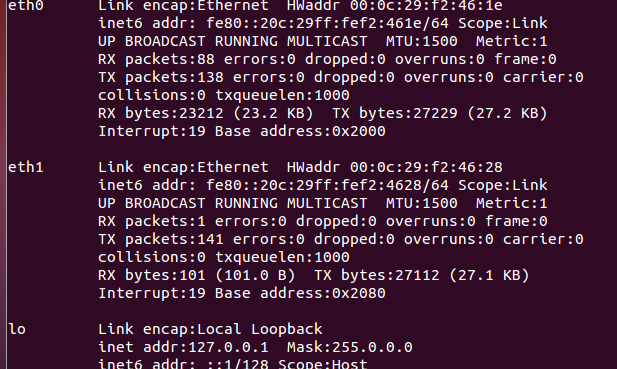
PC1:



ROUTER0:



ROUTER1:



实验目的：

在Router1、Router2上实现静态路由的编程，PC1、PC2实现ICMP的收发程序

数据结构说明：

struct route\_item{

char destination[16];

char gateway[16];

char netmask[16];

char interface[16];

}route\_info[MAX\_ROUTE\_INFO];

Destination,Gateway,netmask,Interface分别对应于目的网络，网关，

子网掩码，对应的网络接口.

struct arp\_table\_item{

char ip\_addr[16];

unsigned char mac\_addr[6];

}arp\_table[MAX\_ARP\_SIZE];

Ip\_addr:IP地址：

Mac\_addr:ip地址所对应的mac地址

实验思路：

对于两台pc机，使用相同的raw\_socket\_ping.c文件，两台router机，使用相同的router.c文件，然后分别有各自的路由表以及arp表。

Raw\_socket\_ping程序，与实验2相似，也就是写一个ping的程序。对于checksum函数，与lab2的内容一样，然后在程序中，分别对pc0,pc1的ip地址写死，因为令pc0的IP地址为192.168.2.2，然后并写出下一跳的mac地址，即与其在同一个子网的router0上的接口，为00:0c:29:34:c4:d2.对于要发送的信息进行装包打包，然后进行发送，这与lab2的操作相同，在这之后，需要一个recv功能的函数，用来对ping的指令的包进行获取，这里只进行了简单的过滤，ppt上说可以通过aockaddr\_ll的某些字段进行判断，但在网上看了半天没看懂，但在wireshark里进行转包的时候发现大部分的ip地址都是0.0.0.0或者是255.255.255.0，所以直接用一个粗暴的方法，也就是直接进行比较，如果ip的地址为上述的两种情况，则直接进行舍弃。对于获得的包，对其icmp\_header->icmp进行判断，当等于8时，则表示请求回显，若为零，则为reply,不需要进行额外操作。对于等于8的情况，需要返回一个数据包，具体的操作与装包时相似，不赘述了。

Router程序：主要的功能就是收发包，首先根据所写的路由表进行初始化，路由表的详细内容会附在报告末尾，首先使用recvfrom函数，对所有的包进行获取，这里需要判断src.dst是否为127.0.0.1，如果不为0，这说明是正确的src,dst地址，将其打印出来，然后将本地ip地址与目的Ip地址进行判断，也就是判断是否是ping的本机，如果是的话，则直接将src与dst的IP地址进行交换，由下面的程序直接进行回发。

后面进行的就是发送ping的request或者是将包发送到下一跳。对检阅路由表，将destination与netmask进行取与操作，可以获得对应的是哪个子网，然后查路由表，可以获得对应的gateway.这里要进行一个判断，判断的内容是进行的下一跳是大于一跳，还是一跳，如果是一跳，则直接查ip地址的mac地址，否则则查gateway的ip地址，然后进行发送。

路由表：

router0

3

192.168.2.0 192.168.2.1 255.255.255.0 eth0

192.168.4.0 192.168.4.1 255.255.255.0 eth1

192.168.3.0 192.168.4.2 255.255.255.0 eth1

4

192.168.2.1 00:0c:29:f2:46:28

192.168.4.1 00:0c:29:34:c4:d2

192.168.4.2 00:0c:29:34:c4:dc

192.168.2.2 00:0c:29:8c:c2:4d

路由表2：

router1

3

192.168.3.0 192.168.3.1 255.255.255.0 eth0

192.168.4.0 192.168.4.2 255.255.255.0 eth1

192.168.2.0 192.168.4.1 255.255.255.0 eth1

4

192.168.3.1 00:0c:29:f2:46:1e

192.168.4.2 00:0c:29:f2:46:28

192.168.4.1 00:0c:29:34:c4:dc

192.168.3.2 00:0c:29:16:11:e9

实验截图：

