

实验四 静态路由编程实现

141220132 银琦 141220132@smail.nju.edu.cn

实验目的

本实验的主要目的是设计和实现一个简单的静态路由机制，用以取代Linux自身通过ip forwarding实现的静态路由方式，进而加深对二三层协议衔接及静态路由的理解。

网络拓扑配置

节点名	Ip	Netmask	Mac地址
Router1	Eth0:192.168.0.	255.255.255.	00:0c:29:49:d9:5
	1	0	6
	Eth1:192.168.1.	255.255.255.	00:0c:29:49:d9:6
Router2	1	0	0
	Eth0:192.168.2.	255.255.255.	00:0c:29:c0:65:1
	Eth1:192.168.1.	255.255.255.	00:0c:29:c0:65:2
PC1	2	0	1
	192.168.0.2	255.255.255.	00:0c:29:95:57:c
PC2		0	f
	192.168.2.2	255.255.255.	00:0c:29:4f:38:0
		0	e

设置IP及网关

PC1:

```
sudo ifconfig eth0 192.168.0.2 netmask 255.255.255.0
```

```
sudo route add default gw 192.168.0.1
```

PC2:

```
sudo ifconfig eth0 192.168.2.2 netmask 255.255.255.0
```

```
sudo route add default gw 192.168.2.1
```

router1

```
sudo ifconfig eth0 192.168.0.1 netmask 255.255.255.0
```

```
sudo ifconfig eth1 192.168.1.1 netmask 255.255.255.0
```

router2

```
sudo ifconfig eth0 192.168.2.1 netmask 255.255.255.0
```

```
sudo ifconfig eth1 192.168.1.2 netmask 255.255.255.0
```

路由规则配置

router1

打开/etc/sysctl.conf文件，修改net.ipv4.ip_forward为0。

router2

```
sudo ip route add 192.168.0.0/24 via 192.168.1.1
```

数据结构说明

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

该结构描述了路由表，其中destination为目标地址，gateway为路由网关，netmask为掩码，interface为以太网端口。

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

该结构描述了arp缓存，其中ip为ip地址，mac_addr为ip下一跳mac地址。

```
struct device_info {  
    char mac[18];  
    int interface;
```

```
}
```

该结构描述了路由器网络配置，其中mac为mac地址，interface为对应的端口号。

配置文件说明

arp_table_info:

```
192.168.1.1 00:0c:29:c0:65:21
```

```
192.168.0.1 00:0c:29:95:57:cf
```

arp缓存，共有两项，可以使得PC1PC2互相ping通。前面为router1的ip地址，后面为ip地址下一跳的mac地址，即192.168.1.1的下一跳为192.168.1.2，192.168.0.1的下一跳为192.168.0.1。

device_item_info:

```
00:0c:29:49:d9:56 0
```

```
00:0c:29:49:d9:60 1
```

路由器网络配置，即router1的两个mac地址和其端口号。

route_table_info:

```
192.168.0.1 192.168.0.1 255.255.255.0 0
```

```
192.168.0.2 192.168.0.1 255.255.255.0 0
```

```
192.168.1.1 192.168.1.1 255.255.255.0 1
```

```
192.168.1.2 192.168.1.1 255.255.255.0 1
```

```
192.168.2.1 192.168.1.1 255.255.255.0 1
```

```
192.168.2.2 192.168.1.1 255.255.255.0 1
```

静态路由表，包括6项，每项四个值为目标地址，网关，子网掩码和端口号，保证四台虚拟机之间可以进行路由转发。

程序设计的思路以及运行流程

运行结果截图

PC1发送包

PC2返回包

```
The MAC ADDRESS of the next hop is 00:0c:29:49:d9:60
check the destination mac in device
The des_mac is in device,start receive data...
TYPE: 0800      packet is IP protocol
the des_ip is: 192.168.0.2
Check and find the des_ip is in the route item table. index=1
check and find that next_hop is in the arp table. index=1
resend data src_mac: 00:0c:29:49:d9:60
resend data des_mac: 00:0c:29:95:57:cf
the interface is eth0
start to send data to next_hop.....
transmitted successfully!
one package transmitted successfully, begin to get the next one...

64 bytes from 192.168.2.2: icmp_req=8 ttl=63 time=5.23 ms
64 bytes from 192.168.2.2: icmp_req=9 ttl=63 time=2.79 ms
64 bytes from 192.168.2.2: icmp_req=10 ttl=63 time=2.17 ms
64 bytes from 192.168.2.2: icmp_req=11 ttl=63 time=1.01 ms
64 bytes from 192.168.2.2: icmp_req=12 ttl=63 time=2.73 ms
64 bytes from 192.168.2.2: icmp_req=13 ttl=63 time=3.07 ms
64 bytes from 192.168.2.2: icmp_req=14 ttl=63 time=2.43 ms
^C
--- 192.168.2.2 ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13053ms
rtt min/avg/max/mdev = 1.019/3.238/7.168/1.501 ms
user@ubuntu:~$
```



PC1 ping PC2

相关参考资料

静态路由：

<http://baike.baidu.com/view/911.htm>

编程实现静态路由：

http://www.pudn.com/downloads245/sourcecode/unix_linux/network/detail1141871.html

对比样例程序

基本上均参考了网络编程实现，代码基本为参考资料第二条给出的文件。在对代码理解后根据自己对静态路由的理解对代码进行了一些小的改动，然后手动进行了配置文件的设置。

代码个人创新及思考

虽然这一次实验个人代码编写量非常少（绝大多数参考网络代码的缘故），但是阅读网络代码，了解各个配置文件功能和配置方法还是花了自己不少的时间。当然在修改网络代码以适应自身实现的过程中也遇到了不少的问题，最常见的就是网络端口号出错导致一边可以ping通另一边却接收不到返回包的情况。

在实验中，也有许多同学之间的帮助，才得以更加快速地解决问题。

遇到的问题

1. 一开始运行时没有切入root账户，于是出现了“create raw socket error”，后来仔细查看讲义才发现要切入root账户运行。
2. 在测试配置文件的时候，没有注意细节，根据代码读入的方式，每个节点是由空格区分的，一开始在行末没有加入空格，导致读入的数据有误，通过在代码中增加printf输出结果后发现。
3. 配置文件中多了空行，导致读入产生错误，最后产生了段错误。
4. 一开始没有弄懂各个配置文件中内容的含义，所以一直没有传输成功，后来静下心来查阅资料，弄清每一项的用途，最后得出了正确的结果。
5. 在一开始配置环境的时候，忽略了要给router2增加路由转发规则，所以始终ping不通，后来增加了路由转发规则后就可以ping通了。