

南京大学本科生实验报告

课程名称：计算机网络 任课教师：田臣/李文中 助教：方毓楚、郑浩、陈伟等（排名不分先后）

学院	地理与海洋科学学院	专业	地理信息科学
学号	191830173	姓名	徐嵩
Email	song.xv@outlook.com	开始/完成日期	2022/3/24

1. 实验名称

Lab 5: Respond to ICMP

2. 实验目的

构建一个具有 ICMP 功能的 IPv4 路由器。

3. 实验内容

- A. 响应 ICMP 报文
- B. 生成 ICMP 错误信息

4. 实验结果

Show how you implement the logic of responding to ICMP echo requests.

当一个链路层帧到达时，首先判断其是否是 IPv4 分组，是则检查目的 IP 是否是路由器接口之一，是则检查是否有 ICMP 报头，是则判断 ICMP 类型是否是 EchoRequest，是则构造一个合适的 ICMP EchoReply 并发送。

In the report, show how you implement the logic of generating ICMP error messages.

在确保该链路层帧包含 IPv4 分组后，检查该分组目的 IP 是否是路由器自身的 IP 之一且不是 ICMP EchoRequest 报文，满足条件则发送一个 ICMP 错误报文目的端口不可达；然后检查目的 IP 是否在路由器转发表中，若不在则构造 ICMP 错误报文目的网络不可达；然后将该 IP 报头的 TTL 减 1，若等于 0，则发送 ICMP 错误报文超时；若因为无法通过 ARP 协议获取下一跳 IP 对应的 MAC 地址，则发送 ICMP 错误报文目的主机不可达。

In the report, show the test result of your router.

测试结果如下：

```
Results for test scenario IP forwarding and ARP requester tests: 28 passed, 0 failed, 0 pending

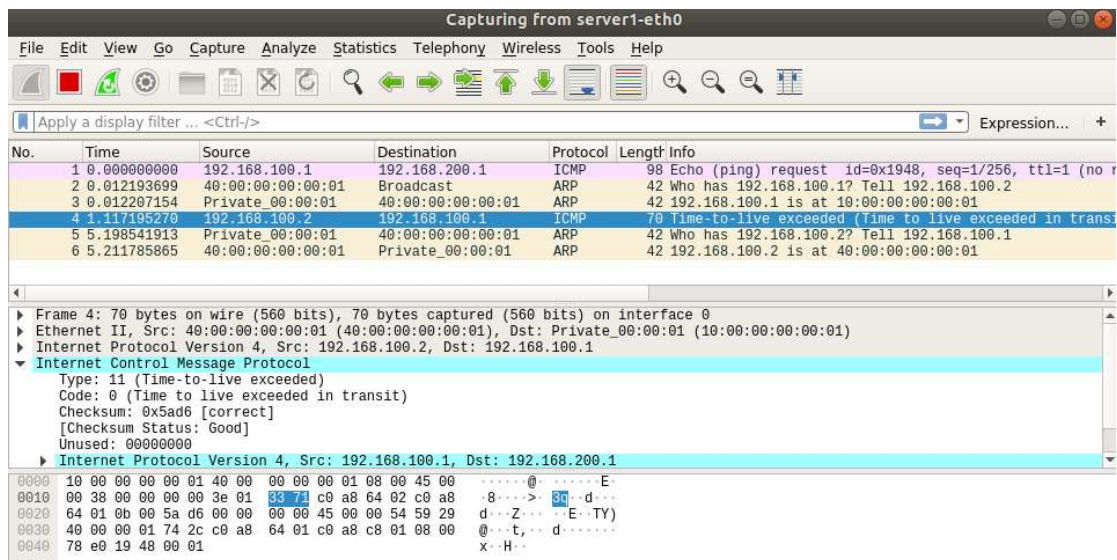
Passed:
1 ICMP echo request (PING) for the router IP address
  192.168.1.1 should arrive on router-eth0. This PING is
  directed at the router, and the router should respond with
  an ICMP echo reply.
2 Router should send an ARP request for 10.10.1.254 out
  router-eth1.
3 Router should receive ARP reply for 10.10.1.254 on router-
  eth1.

26 Router should send an ARP request for 192.168.1.239.
27 Router should receive ARP reply for 192.168.1.239.
28 Router should send an ICMP host unreachable error to
  192.168.1.239.

All tests passed!
```

Write the procedure and analysis in your report with screenshots.

打开 server1 链路的 wireshark，使用命令 `ping -c1 -t1 192.168.200.1`，由于 ttl 设置为 1，因此在到达路由器后返回 ICMP 报文超时错误；



然后使用命令 `ping -c1 192.168.200.3`，由于目标网络 192.168.200.0 存在但没有 3 号主机，因此路由器返回 ICMP 报文目的主机不可达，如下图所示：

Capturing from server1-eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
4	1.117195270	192.168.100.2	192.168.100.1	ICMP	70	Time-to-live exceeded (Time to live exceeded in tran
5	5.198541913	Private_00:00:01	40:00:00:00:00:01	ARP	42	Who has 192.168.100.2? Tell 192.168.100.1
6	5.211785865	40:00:00:00:00:01	Private_00:00:01	ARP	42	192.168.100.2 is at 40:00:00:00:00:01
7	408.560281184	192.168.100.1	192.168.200.3	ICMP	98	Echo (ping) request id=0x19ec, seq=1/256, ttl=64 (n
8	413.605472072	192.168.100.2	192.168.100.1	ICMP	70	Destination unreachable (Host unreachable)
9	413.774020482	Private_00:00:01	40:00:00:00:00:01	ARP	42	Who has 192.168.100.2? Tell 192.168.100.1
10	413.799521961	40:00:00:00:00:01	Private_00:00:01	ARP	42	192.168.100.2 is at 40:00:00:00:00:01

Frame 8: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0

Ethernet II, Src: 40:00:00:00:00:01 (40:00:00:00:00:01), Dst: Private_00:00:01 (10:00:00:00:00:01)

Internet Protocol Version 4, Src: 192.168.100.2, Dst: 192.168.100.1

Internet Control Message Protocol

Type: 3 (Destination unreachable)

Code: 1 (Host unreachable)

Checksum: 0xa363 [correct]

[Checksum Status: Good]

Unused: 00000000

Internet Protocol Version 4, Src: 192.168.100.1, Dst: 192.168.200.3

```

0000 10 00 00 00 00 01 40 00 00 00 00 01 08 00 45 00  .....@:  .... E
0010 00 38 00 00 00 00 3f 01 32 71 c0 a8 64 02 c0 a8  8....?  2q..d...
0020 64 01 03 01 a3 63 00 00 00 00 45 00 00 54 07 1c  d....c...  E..T..
0030 40 00 3f 01 87 37 c0 a8 64 01 c0 a8 c8 03 00 00  @?...7... d....
0040 37 ae 19 ec 00 01 7.....

```

5. 核心代码

```

def IPv4_handler(self, ip, recv):
    timestamp, ifaceName, packet = recv
    # TODO: 查表, 验证目的 IP 是否是自身接口
    # 如果是, 且是 ICMP echo 则回复 echo reply
    if ip.dst in self.ips:
        if packet.has_header(ICMP):
            icmp = packet.get_header(ICMP)
            if icmp.icmptype == ICMPType.EchoRequest:
                self.icmp_echo_reply(ip, recv)
                return
        self.icmp_error_handle(33, ip, recv)
        return
    next_hop, outport = self.look_up_forwarding_table(ip.dst)
    if next_hop != '-1':
        # 检查 ttl
        ip.ttl = ip.ttl - 1
        if ip.ttl == 0:
            self.icmp_error_handle(110, ip, recv)
            return
        # TODO: 找到下一跳 MAC
        # 如果找不到自动加入转发队列, 并发送 ARP request, 返回查询 MAC 结
        next_hop_mac_addr = self.look_up_arp_table(next_hop, outport, ip, recv)
        if next_hop_mac_addr != 'ff-ff-ff-ff-ff-ff':
            # TODO: 准备以太网包
            eth_header = Ethernet()
            eth_header.src,xxx = self.find_mac_ip_by_port(outport)
            eth_header.dst = next_hop_mac_addr
            eth_header.ethertype = EtherType.Ipv4
            if packet.has_header(Ethernet):
                del packet[Ethernet]
            packet.insert_header(0, eth_header)

            # TODO: 发送 IP 包
            self.net.send_packet(outport, packet)
        else:
            self.icmp_error_handle(30, ip, recv)

```

果

6. 总结与感想

通过 Lab 3-5, 我基本完整实现了一个 IPv4 路由器, 在这个过程中对网络层和 IP 协议理解更加深入。