

南京大学计算机网络实验报告

任课教师:田臣

实验五 Respond to ICMP

计算机科学与技术系

181860055 刘国涛

邮箱: 181860055@smail.nju.edu.cn

2020年5月1日

实验目的

- 学习并实现路由对ICMP的响应
- 实现ICMP错误的生成

实验内容

TASK 2 Responding to ICMP echo requests

任务概述 为属于路由的ICMP echo request的数据包构造echo reply, 并发送

任务实现

构造echo reply的数据包需要分别构造：

1. ICMP header
2. IP header
3. Ethernet header

1、ICMP header

构造ICMP header需要设置以下参数：

- icmptype设为EchoReply
- icmpdata的 **sequence**、**identifier**、**data** 设为与request相同

```
1 request_icmp_header = request.get_header(ICMP)
2 icmp = ICMP()
3 icmp.icmptype = ICMPType.EchoReply
4 icmp.icmpdata.sequence =
  request_icmp_header.icmpdata.sequence
5 icmp.icmpdata.identifier =
  request_icmp_header.icmpdata.identifier
6 icmp.icmpdata.data = request_icmp_header.icmpdata.data
```

2、IP header

构造IP header需要设置以下参数:

- 目的IP为request的源IP, 源IP为request的目的IP
- protocol为ICMP
- ttl>8即可

```
1 request_ipv4_header = request.get_header(IPv4)
2 ipv4 = IPv4()
3 ipv4.dst = request_ipv4_header.src
4 ipv4.src = request_ipv4_header.dst
5 ipv4.protocol = IPPROTO.ICMP
6 ipv4.ttl = 64
7 ipv4.ipid = 0
```

3、Ethernet header

```
1 eth = Ethernet()
2 eth.ethertype = EtherType.IP
3 eth.src = interface.ethaddr # 发送端口的MAC地址
4 eth.dst = next_hop.ethaddr # 下一跳的MAC地址
```

这里下一跳的ethaddr的获取与lab4相同, 过程如下:

1. 从forwarding table中通过最长前缀匹配得到下一跳的IP
2. 在ARP缓存表中查找下一跳IP对应的MAC地址
3. 如果ARP缓存表中没有, 则发起ARP请求, 并将echo reply packet 加入等待队列中, 知道接收到ARP答复才装填eth.dst, 并进行echo reply

TASK 3 Generating ICMP error messages

任务概述 考虑四种ICMP错误情况, 并将错误发送回源数据包的source IP

任务实现

需要考虑的错误:

1. ICMP destination network unreachable -- 转发表中匹配不到目的IP
2. ICMP time exceeded -- 转发数据包的ttl-1后为0

3. ICMP destination host unreachable -- 对下一跳或目标主机的ARP请求失败（超过5次）
4. ICMP destination port unreachable -- ICMP类型不是echo request

首先需要定义一个异常类：

```
1 class NetworkException(Exception):
2     def __init__(self, errInfo, args):
3         super().__init__(self)
4         self.errInfo = errInfo
5         self.args = args
6     def __str__(self):
7         return self.errInfo
```

以上四种错误都将通过这种方式抛出：

```
1 raise NetworkException(errInfo, args)
```

并通过

```
1 try:
2     pass # do something here
3 except NetworkException as ne:
4     self.handle_icmperror(ne, ne.args)
```

来捕获四种错误，并转到处理函数 `handle_icmperror` 中处理

这里关注 `handle_icmperror`，在处理函数中，分别处理四种错误：

```
1 class Router():
2     def handle_icmperror(self, ne, args):
3         kind = str(ne)
4         if kind == "destination network unreachable":
5             pass # handle
6         elif kind == "destination host unreachable":
7             pass # handle
8         elif kind == "time exceeded":
9             pass # handle
10        elif kind == "destination port unreachable":
11            pass # handle
```

四种错误的处理方式大致相同，以 `destination network unreachable` 为例

首先是构造数据包，同样的需要分别构造ICMP header,IP header,Ethernet header

- 构造ICMP header
 - `icmptype = DestinationUnreachable`
 - `icmpdata.data`拷贝原数据包的前28个字节以达到有效载荷
 - `icmpdata.origdgramlen`为原数据报的长度（去掉Ethernet header后）
 - `icmpcode = 0`（network unreachable）
- 构造IP header
 - `ttl = 20` (大于8即可)
 - `dst`为原数据包的source IP
 - `src`为发送端口IP
 - `protocol`为ICMP
- 构造Ethernet header
 - `ethertype`为IP
 - `src`为发送端口MAC地址
 - `dst`结合forwarding table和ARP请求得到

然后将构造好的数据包从接收源数据包的端口发出

实验结果

TASK 3

Testing:

```
8 Router should send ARP request for 10.10.123.123 out router-eth1.
9 Router should receive ARP reply for 10.10.123.123 on router-eth1.
10 Router should send ICMP time exceeded error back to 10.10.123.123 on router-eth1.
11 A packet to be forwarded to 1.2.3.4 should arrive on router-eth1. The destination address 1.2.3.4 should not match any entry in the forwarding table.
12 Router should send an ICMP destination network unreachable error back to 10.10.123.123 out router-eth1.
13 A UDP packet addressed to the router's IP address 192.168.1.1 should arrive on router-eth1. The router cannot handle this type of packet and should generate an ICMP destination port unreachable error.
14 The router should send an ICMP destination port unreachable error back to 172.16.111.222 out router-eth1.
15 An IP packet from 192.168.1.239 for 10.10.50.250 should arrive on router-eth0. The host 10.10.50.250 is presumed not to exist, so any attempts to send ARP requests will eventually fail.
16 Router should send an ARP request for 10.10.50.250 on router-eth1.
17 Router should try to receive a packet (ARP response), but then timeout.
18 Router should send an ARP request for 10.10.50.250 on router-eth1.
19 Router should try to receive a packet (ARP response), but then timeout.
20 Router should send an ARP request for 10.10.50.250 on router-eth1.
21 Router should try to receive a packet (ARP response), but then timeout.
22 Router should send an ARP request for 10.10.50.250 on router-eth1.
23 Router should try to receive a packet (ARP response), but then timeout.
24 Router should send an ARP request for 10.10.50.250 on router-eth1.
25 Router should try to receive a packet (ARP response), but then timeout. At this point, the router should give up and generate an ICMP host unreachable error.
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!

```
(syenv) njucs@njucs-VirtualBox:~/switchyard/lab_5$
```

Deploying:

步骤:

在客户端 `ping -c 2 192.168.100.2`

```

root@njucs-VirtualBox:~/switchyard/lab_5# ping -c 2 192.168.100.2
PING 192.168.100.2 (192.168.100.2) 56(84) bytes of data.
64 bytes from 192.168.100.2: icmp_seq=1 ttl=64 time=144 ms
64 bytes from 192.168.100.2: icmp_seq=2 ttl=64 time=80.0 ms

--- 192.168.100.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 80.042/112.216/144.391/32.176 ms
root@njucs-VirtualBox:~/switchyard/lab_5#

```

client的抓包结果

Capturing from router-eth2						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.1.1.1	192.168.100.2	ICMP	98	Echo (ping) request id=0x15cd, seq=1/256, ttl=64 (reply in 4)
2	0.048454814	40:00:00:00:00:03	Broadcast	ARP	42	Who has 10.1.1.1? Tell 10.1.1.2
3	0.048469657	30:00:00:00:00:01	40:00:00:00:00:03	ARP	42	10.1.1.1 is at 30:00:00:00:00:01
4	0.144361081	192.168.100.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x15cd, seq=1/256, ttl=64 (request in 1)
5	1.001648171	10.1.1.1	192.168.100.2	ICMP	98	Echo (ping) request id=0x15cd, seq=2/512, ttl=64 (reply in 6)
6	1.081656249	192.168.100.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x15cd, seq=2/512, ttl=64 (request in 5)
7	5.110132674	30:00:00:00:00:01	40:00:00:00:00:03	ARP	42	Who has 10.1.1.2? Tell 10.1.1.1
8	5.152649982	40:00:00:00:00:03	30:00:00:00:00:01	ARP	42	10.1.1.2 is at 40:00:00:00:00:03

Frame 6: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
 Ethernet II, Src: 40:00:00:00:00:03 (40:00:00:00:00:03), Dst: 30:00:00:00:00:01 (30:00:00:00:00:01)
 Internet Protocol Version 4, Src: 192.168.100.2, Dst: 10.1.1.1
 Internet Control Message Protocol

说明能够ping通router上的接口

在客户端 `ping -c 1 -t 1 192.168.100.1`

```

root@njucs-VirtualBox:~/switchyard/lab_5# ping -c 1 -t 1 192.168.100.1
PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
From 10.1.1.2 icmp_seq=1 Time to live exceeded

--- 192.168.100.1 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms
root@njucs-VirtualBox:~/switchyard/lab_5#

```

得到抓包结果：

Capturing from client-eth0						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.1.1.1	192.168.100.2	ICMP	98	Echo (ping) request id=0x15cd, seq=1/256, ttl=64 (reply in 4)
2	0.048462237	40:00:00:00:00:03	Broadcast	ARP	42	Who has 10.1.1.1? Tell 10.1.1.2
3	0.048471634	30:00:00:00:00:01	40:00:00:00:00:03	ARP	42	10.1.1.1 is at 30:00:00:00:00:01
4	0.144368134	192.168.100.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x15cd, seq=1/256, ttl=64 (request in 1)
5	1.001647470	10.1.1.1	192.168.100.2	ICMP	98	Echo (ping) request id=0x15cd, seq=2/512, ttl=64 (reply in 6)
6	1.081655371	192.168.100.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x15cd, seq=2/512, ttl=64 (request in 5)
7	5.110132114	30:00:00:00:00:01	40:00:00:00:00:03	ARP	42	Who has 10.1.1.2? Tell 10.1.1.1
8	5.152657300	40:00:00:00:00:03	30:00:00:00:00:01	ARP	42	10.1.1.2 is at 40:00:00:00:00:03
9	179.150294264	10.1.1.1	192.168.100.1	ICMP	98	Echo (ping) request id=0x15d5, seq=1/256, ttl=1 (no response found!)
10	179.259487244	40:00:00:00:00:03	Broadcast	ARP	42	Who has 10.1.1.1? Tell 10.1.1.2
11	179.259500446	30:00:00:00:00:01	40:00:00:00:00:03	ARP	42	10.1.1.1 is at 30:00:00:00:00:01
12	179.359213712	10.1.1.2	10.1.1.1	ICMP	70	Time-to-live exceeded (time to live exceeded in transit)
13	184.310393039	30:00:00:00:00:01	40:00:00:00:00:03	ARP	42	Who has 10.1.1.2? Tell 10.1.1.1
14	184.399296851	40:00:00:00:00:03	30:00:00:00:00:01	ARP	42	10.1.1.2 is at 40:00:00:00:00:03

Frame 12: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
 Ethernet II, Src: 40:00:00:00:00:03 (40:00:00:00:00:03), Dst: 30:00:00:00:00:01 (30:00:00:00:00:01)
 Internet Protocol Version 4, Src: 10.1.1.2, Dst: 10.1.1.1
 Internet Control Message Protocol

router返回了Time exceeded

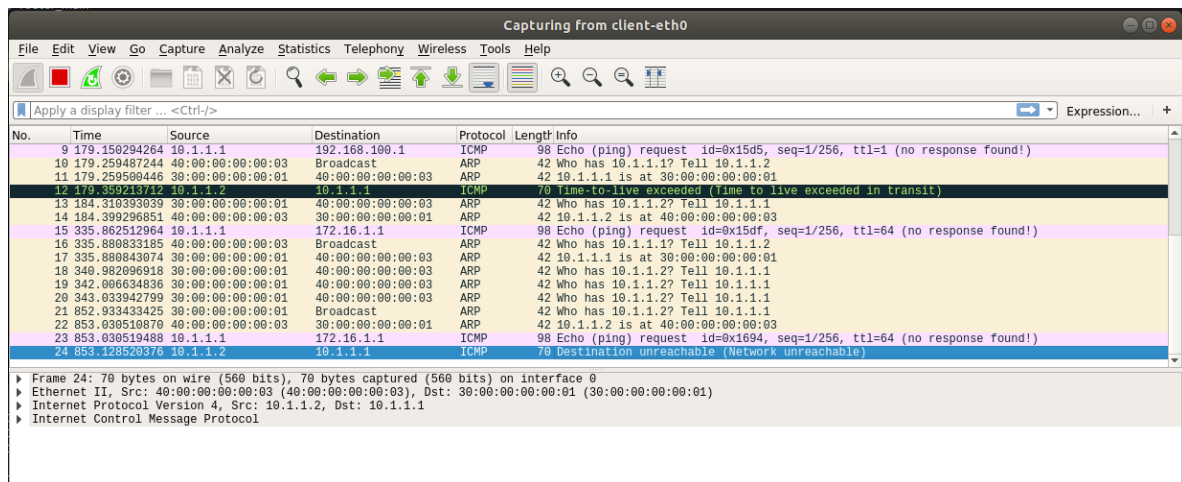
在客户端 `ping -c 1 172.16.1.1`

```
root@njucs-VirtualBox:~/switchyard/lab_5# ping -c 1 172.16.1.1
PING 172.16.1.1 (172.16.1.1) 56(84) bytes of data.
From 10.1.1.2 icmp_seq=1 Destination Net Unreachable

--- 172.16.1.1 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

root@njucs-VirtualBox:~/switchyard/lab_5#
```

得到抓包结果



如蓝色条目所示，router返回了Network unreachable

使用traceroute测试

```
root@njucs-VirtualBox:~/switchyard/lab_5# traceroute -N 1 -n 192.168.100.1
traceroute to 192.168.100.1 (192.168.100.1), 30 hops max, 60 byte packets
 1 10.1.1.2 112.964 ms 104.048 ms 110.031 ms
 2 192.168.100.1 308.149 ms 208.708 ms 208.421 ms
root@njucs-VirtualBox:~/switchyard/lab_5#
```

得到抓包结果：

Capturing from client-eth0						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
25	2015.2790199	10.1.1.1	192.168.100.1	UDP	74	58899 → 33434 Len=32
26	2015.2915079	40:00:00:00:00:03	Broadcast	ARP	42	Who has 10.1.1.1? Tell 10.1.1.2
27	2015.2915170	30:00:00:00:00:01	40:00:00:00:00:03	ARP	42	10.1.1.1 is at 30:00:00:00:00:01
28	2015.3928668	10.1.1.2	10.1.1.1	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
29	2015.3929556	10.1.1.1	192.168.100.1	UDP	74	33419 → 33435 Len=32
30	2015.4969938	10.1.1.2	10.1.1.1	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
31	2015.4970872	10.1.1.1	192.168.100.1	UDP	74	48251 → 33436 Len=32
32	2015.6071070	10.1.1.2	10.1.1.1	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
33	2015.6072148	10.1.1.1	192.168.100.1	UDP	74	43998 → 33437 Len=32
34	2015.9153499	192.168.100.1	10.1.1.1	ICMP	102	Destination unreachable (Port unreachable)
35	2015.9154344	10.1.1.1	192.168.100.1	UDP	74	54805 → 33438 Len=32
36	2016.1241323	192.168.100.1	10.1.1.1	ICMP	102	Destination unreachable (Port unreachable)
37	2016.1242806	10.1.1.1	192.168.100.1	UDP	74	60777 → 33439 Len=32
38	2016.3326898	192.168.100.1	10.1.1.1	ICMP	102	Destination unreachable (Port unreachable)
39	2020.3460071	30:00:00:00:00:01	40:00:00:00:00:03	ARP	42	Who has 10.1.1.2? Tell 10.1.1.1
40	2020.4154056	40:00:00:00:00:03	30:00:00:00:00:01	ARP	42	10.1.1.2 is at 40:00:00:00:00:03
▶ Frame 24: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0 ▶ Ethernet II, Src: 40:00:00:00:00:03 (40:00:00:00:00:03), Dst: 30:00:00:00:00:01 (30:00:00:00:00:01) ▶ Internet Protocol Version 4, Src: 10.1.1.2, Dst: 10.1.1.1 ▶ Internet Control Message Protocol						

发送udp后成功产生port unreachable错误

总结与感想

本次实验进一步了解了在网络中router错误的产生和通知，以及router对ICMP的响应方式