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：

学院	计算机科学与技术系	专业	计算机科学与技术
学号	185220001	姓名	磯田智明
Email	185220001@smail.nju.edu.cn	开始/完成日期	2021.04.28/2021.05.05

1. 实验名称

IPv4 Router-Respond to ICMP

2. 实验目的

- 响应ICMP echo request
- 根据不同情况生成ICMP错误信息

3. 实验内容

Task 1: Preparation

配置实验环境

Task 2: Responding to ICMP echo requests

响应ICMP echo request

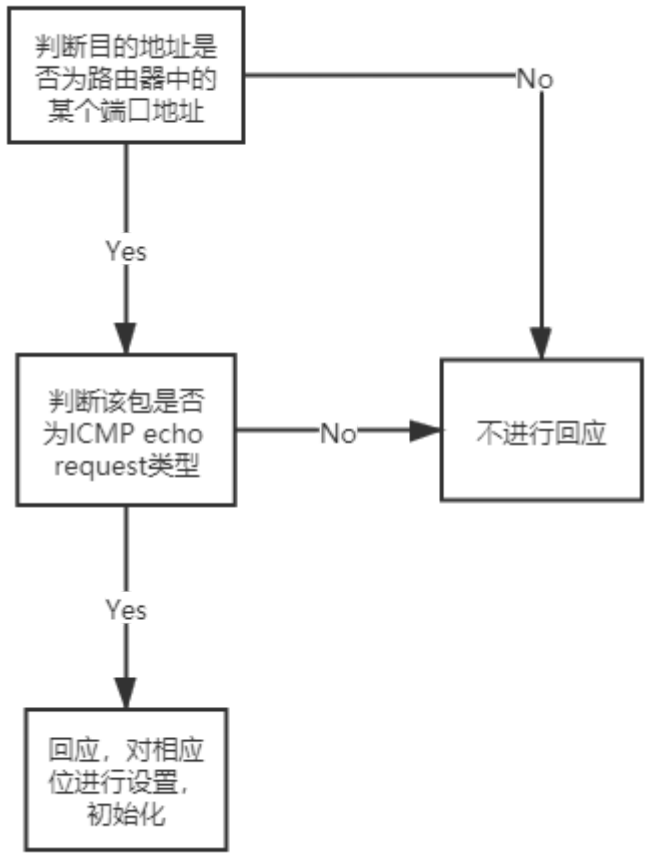
Task 3: Generating ICMP error messages

根据不同情况发送ICMP错误信息，如：目标网络不可达，超时等

4. 实验结果

Task 2: Responding to ICMP echo requests

The logic of responding to ICMP echo requests



首先判断目的地址是否为路由器上的某个端口，并且该包还是 `echo request` 类型，则需要进行回应。

其中需要对ICMP包头进行如下处理：

- 将 `icmptype` 和 `icmpcode` 都设置为0，即为 `echo reply` 类型；
- 将 `echo request` 的 `sequence` ，`identifier` 和 `data` 部分复制到新的ICMP包头中；

还需要对IPv4包头进行如下处理：

- 将源地址设置成路由器的端口地址（`echo request` 到达端口）；
- 将目的地址设置成 `echo request` 的源地址。

最后再将设置好的包转发出去即可

Task 3: Generating ICMP error messages

The logic of generating ICMP error messages

在本节中需要处理四种错误类型，分别是：

- `ICMP time exceeded` 是判断IPv4包头中的 `ttl` 是否为0；如果为0则需要创建一个 `ICMP time exceeded` 错误信息； `icmptype` 为11， `icmpcode` 为0。
- `ICMP destination port unreachable` 是数据包的目的地址是路由器的端口，但是该包并不是 `echo request` 类型，则会触发该错误； `icmptype` 为3， `icmpcode` 为3。
- `ICMP destination network unreachable` 是如果数据包的目的地址与转发表中的信息匹配不上，则会报错； `icmptype` 为3， `icmpcode` 为0。
- `ICMP destination host unreachable` 是发送ARP request超过5次但是没有回应，则会触发该报错； `icmptype` 为3， `icmpcode` 为1。

其中目的ip地址为原数据包的源地址所对应转发表中的目的地址，源ip地址设置为原数据包到达路由器的端口

Running in the Test Environment

在terminal中执行：

```
swyard -t testcases/router3_testscenario.srpy myrouter.py
```

```
(syenv) njucs@njucs-VirtualBox:~/sy/lab-5-vectormoon$ swyard -t testcases/router3_testscenario.srpy myrouter.py
17:22:40 2021/05/16     INFO Starting test scenario testcases/router3_testscenario.srpy
17:22:40 2021/05/16     INFO ------------------------forwarding_table_info------------------------
17:22:40 2021/05/16     INFO         address: 192.168.1.0;  other: [IPv4Address('255.255.255.0'), '0.0.0.0', 'router-eth0'].
17:22:40 2021/05/16     INFO         address: 10.10.0.0;  other: [IPv4Address('255.255.0.0'), '0.0.0.0', 'router-eth1'].
17:22:40 2021/05/16     INFO         address: 172.16.42.0;  other: [IPv4Address('255.255.255.252'), '0.0.0.0', 'router-eth2'].
17:22:40 2021/05/16     INFO ------------------------------------------------------------------
17:22:40 2021/05/16     INFO ------------------------forwarding_table_info------------------------
17:22:40 2021/05/16     INFO         address: 192.168.1.0;  other: [IPv4Address('255.255.255.0'), '0.0.0.0', 'router-eth0'].
17:22:40 2021/05/16     INFO         address: 10.10.0.0;  other: [IPv4Address('255.255.0.0'), '0.0.0.0', 'router-eth1'].
17:22:40 2021/05/16     INFO         address: 172.16.42.0;  other: [IPv4Address('255.255.255.252'), '0.0.0.0', 'router-eth2'].
17:22:40 2021/05/16     INFO         address: 172.16.0.0;  other: [IPv4Address('255.255.0.0'), IPv4Address('192.168.1.2'), 'router-eth0'].
17:22:40 2021/05/16     INFO         address: 172.16.128.0;  other: [IPv4Address('255.255.192.0'), IPv4Address('10.10.0.254'), 'router-eth1'].
17:22:40 2021/05/16     INFO         address: 172.16.64.0;  other: [IPv4Address('255.255.192.0'), IPv4Address('10.10.1.254'), 'router-eth1'].
17:22:40 2021/05/16     INFO         address: 10.100.0.0;  other: [IPv4Address('255.255.0.0'), IPv4Address('172.16.42.2'), 'router-eth2'].
17:22:40 2021/05/16     INFO ------------------------------------------------------------------
17:22:40 2021/05/16     INFO handle_packet
17:22:41 2021/05/16     INFO Sending arp request Ethernet 10:00:00:00:00:02->ff:ff:ff:ff:ff:ff ARP | Arp 10:00:00:00:00:02:10.10.0.1 ff:ff:ff:ff:ff:ff:10.10.1.254 to router-eth1
17:22:41 2021/05/16     INFO handle_packet
17:22:41 2021/05/16     INFO -------------------------arp_table_info-------------------------
17:22:41 2021/05/16     INFO         IP: 10.10.1.254; MAC Address: ab:cd:00:00:00:01
17:22:41 2021/05/16     INFO -----------------------------------------------------------
17:22:41 2021/05/16     INFO Forwarding packet Ethernet 10:00:00:00:00:02->ab:cd:00:00:00:01 IP | IPv4 192.168.1.1->172.16.111.222 ICMP | ICMP EchoReply 0 42 (12 data bytes) to router-eth1
17:22:41 2021/05/16     INFO handle_packet
17:22:41 2021/05/16     INFO Forwarding packet Ethernet 10:00:00:00:00:02->ab:cd:00:00:00:01 IP | IPv4 10.10.0.1->172.16.111.222 ICMP | ICMP EchoReply 0 42 (0 data bytes) to router-eth1
17:22:41 2021/05/16     INFO handle_packet
17:22:42 2021/05/16     INFO Sending arp request Ethernet 10:00:00:00:00:02->ff:ff:ff:ff:ff:ff ARP | Arp 10:00:00:00:00:02:10.10.0.1 ff:ff:ff:ff:ff:ff:10.10.123.123 to router-eth1
17:22:42 2021/05/16     INFO handle_packet
17:22:42 2021/05/16     INFO -------------------------arp_table_info-------------------------
17:22:42 2021/05/16     INFO         IP: 10.10.1.254; MAC Address: ab:cd:00:00:00:01
17:22:42 2021/05/16     INFO         IP: 10.10.123.123; MAC Address: be:ef:00:00:00:01
17:22:42 2021/05/16     INFO -----------------------------------------------------------
17:22:42 2021/05/16     INFO Forwarding packet Ethernet 10:00:00:00:00:02->be:ef:00:00:00:01 IP | IPv4 10.10.0.1->10.10.123.123 ICMP | ICMP TimeExceeded:TTLExpired 28 bytes of raw payload (b'E\x00\x00\x1c
\x00\x00\x00\x00\x00\x01') OrigDgramLen: 0 to router-eth1
17:22:42 2021/05/16     INFO handle_packet
17:22:42 2021/05/16     INFO Forwarding packet Ethernet 10:00:00:00:00:02->be:ef:00:00:00:01 IP | IPv4 10.10.0.1->10.10.123.123 ICMP | ICMP DestinationUnreachable:NetworkUnreachable 28 bytes of raw payloa
d (b'E\x00\x00\x1c\x00\x00\x00\x00\x01') NextHopMTU: 0 to router-eth1
17:22:42 2021/05/16     INFO handle_packet
17:22:42 2021/05/16     INFO Forwarding packet Ethernet 10:00:00:00:00:02->ab:cd:00:00:00:01 IP | IPv4 10.10.0.1->172.16.111.222 ICMP | ICMP DestinationUnreachable:PortUnreachable 28 bytes of raw payload
(b'E\x00\x00!\x00\x00\x00\x00\x00$\x11') NextHopMTU: 0 to router-eth1
17:22:42 2021/05/16     INFO handle_packet
17:22:43 2021/05/16     INFO Sending arp request Ethernet 10:00:00:00:00:02->ff:ff:ff:ff:ff:ff ARP | Arp 10:00:00:00:00:02:10.10.0.1 ff:ff:ff:ff:ff:ff:10.10.50.250 to router-eth1
17:22:44 2021/05/16     INFO Sending arp request Ethernet 10:00:00:00:00:02->ff:ff:ff:ff:ff:ff ARP | Arp 10:00:00:00:00:02:10.10.0.1 ff:ff:ff:ff:ff:ff:10.10.50.250 to router-eth1
17:22:46 2021/05/16     INFO Sending arp request Ethernet 10:00:00:00:00:02->ff:ff:ff:ff:ff:ff ARP | Arp 10:00:00:00:00:02:10.10.0.1 ff:ff:ff:ff:ff:ff:10.10.50.250 to router-eth1
17:22:47 2021/05/16     INFO Sending arp request Ethernet 10:00:00:00:00:02->ff:ff:ff:ff:ff:ff ARP | Arp 10:00:00:00:00:02:10.10.0.1 ff:ff:ff:ff:ff:ff:10.10.50.250 to router-eth1
17:22:49 2021/05/16     INFO Sending arp request Ethernet 10:00:00:00:00:02->ff:ff:ff:ff:ff:ff ARP | Arp 10:00:00:00:00:02:10.10.0.1 ff:ff:ff:ff:ff:ff:10.10.50.250 to router-eth1
```

```
17:22:50 2021/05/16     INFO Delete packet IPv4 172.16.42.1->192.168.1.239 ICMP | ICMP EchoRequest 0 42 (0 data bytes)
17:22:51 2021/05/16     INFO Sending arp request Ethernet 10:00:00:00:00:01->ff:ff:ff:ff:ff:ff ARP | Arp 10:00:00:00:00:01:192.168.1.1 ff:ff:ff:ff:ff:ff:192.168.1.239 to router-eth0
17:22:51 2021/05/16     INFO handle_packet
17:22:51 2021/05/16     INFO -------------------------arp_table_info-------------------------
17:22:51 2021/05/16     INFO         IP: 10.10.1.254; MAC Address: ab:cd:00:00:00:01
17:22:51 2021/05/16     INFO         IP: 10.10.123.123; MAC Address: be:ef:00:00:00:01
17:22:51 2021/05/16     INFO         IP: 192.168.1.239; MAC Address: 00:01:02:03:04:05
17:22:51 2021/05/16     INFO -----------------------------------------------------------
17:22:51 2021/05/16     INFO Forwarding packet Ethernet 10:00:00:00:00:01->00:01:02:03:04:05 IP | IPv4 192.168.1.1->192.168.1.239 ICMP | ICMP DestinationUnreachable:HostUnreachable 28 bytes of raw payload
 (b'E\x00\x00\x1c\x00\x00\x00\x00?\x01') NextHopMTU: 0 to router-eth0

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.
4   Router should send ICMP echo reply (PING) to 172.16.111.222
    out router-eth1 (that's right: ping reply goes out a
    different interface than the request).
5   ICMP echo request (PING) for the router IP address 10.10.0.1
    should arrive on router-eth1.
6   Router should send ICMP echo reply (PING) to 172.16.111.222
    out router-eth1.
7   ICMP echo request (PING) for 10.100.1.1 with a TTL of 1
    should arrive on router-eth1.  The router should decrement
    the TTL to 0 then see that the packet has "expired" and
    generate an ICMP time exceeded error.
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!
```

Running in the Mininet

在终端中输入以下指令启动mininet

```
$ sudo python start_mininet.py
```

在 `client` 中使用traceroute测试，使用如下指令：

```
client# traceroute -N 1 -n 192.168.100.1
```

```
(syenv) root@njucs-VirtualBox:~/sy/lab-5-vectormoon# 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  21.864 ms *  35.953 ms
 2  192.168.100.1  111.045 ms  103.605 ms  103.113 ms
```

在mininet中启动 `server1` 和 `router`

```
mininet> xterm server1
mininet> xterm router
```

利用wireshark抓 `router-eth0` 和 `server1` 的包

```
router# wireshark -i router-eth0
```

在 `router` 的xterm中打开虚拟环境并且启动 `my_router.py`

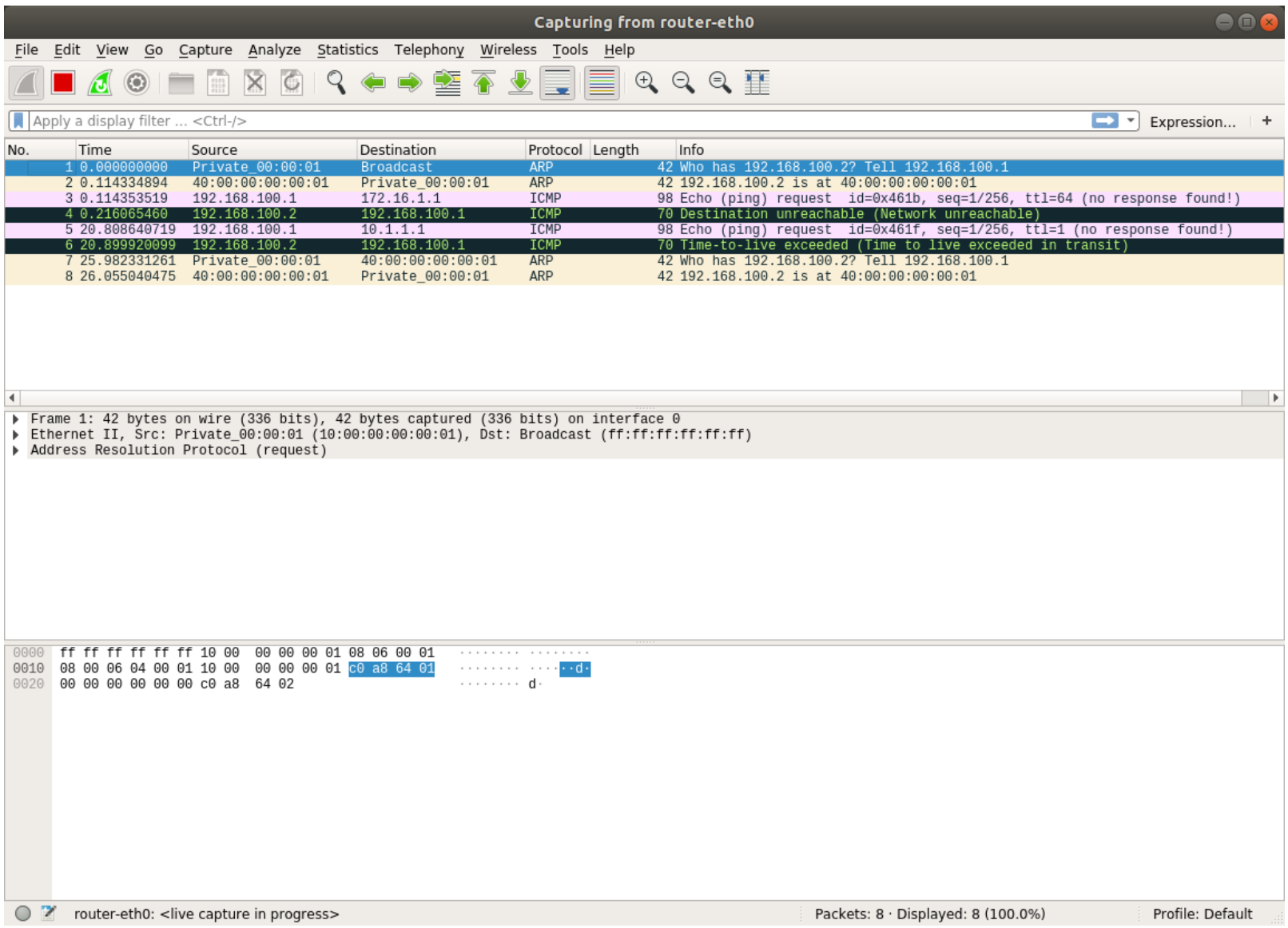
```
(syenv) router# swyard myrouter.py
```

在 `server1` 的xterm中，使用以下指令测试：

```
server1# ping -c 1 172.16.1.1
server1# ping -c 1 -t 1 client
```

得到如下结果：

wireshark中的结果



在 `start_mininet.py` 的 `server1` 的配置中增加了一个新的路径172.16.1.1，但是没有连接到任何主机中，所以在执行第一条指令的时候，所以会显示 `Destination unreachable`；在执行第二条指令的时候，将 `ttl` 设置为1，因为 `server1` 到 `server2` 需要两跳才能到达目的地，所以会显示 `Time-to-live exceeded`

5. 核心代码

Task 2: Responding to ICMP echo requests

```
if (targetip_exist_flag != -1) and (icmp.icmptype == 8 and icmp.icmpcode ==
0):
    icmp_reply = ICMP()
    # echo reply(ping)
    icmp_reply.icmptype = 0
    icmp_reply.icmpcode = 0
    # copy info from icmp request to icmp reply
    icmp_reply.icmpdata.sequence = icmp.icmpdata.sequence
    icmp_reply.icmpdata.identifier = icmp.icmpdata.identifier
    icmp_reply.icmpdata.data = icmp.icmpdata.data
```

```
    # 2=ICMP
    packet[2] = icmp_reply
    # construct IP header
    ipv4.dst = ipv4.src
    ipv4.src = self.ip_list[targetip_exist_flag]
```

判断目的地址是否为路由器上的某个端口，并且该包还是 `echo request` 类型，则需要进行回应。其中需要对ICMP包头和IPv4的包头需要做处理

Task 3: Generating ICMP error messages

```
def icmp_error(self, origpkt, type_of_error, icmp_code, srcip, dstip):
    # origpkt = Ethernet() + IPv4() + ICMP()
    i = origpkt.get_header_index(Ethernet)
    del origpkt[i]

    eth = Ethernet()

    icmp = ICMP()
    icmp.icmptype = type_of_error
    icmp.icmpcode = icmp_code
    icmp.icmpdata.data = origpkt.to_bytes()[:28]

    ip = IPv4()
    ip.protocol = IPProtocol.ICMP
    ip.ttl = 64
    ip.src = srcip
    ip.dst = dstip

    pkt = eth + ip + icmp
    return pkt
```

6. 总结与感想

对路由器和ICMP协议有了一定的了解，但是对包头的结构没有深刻的认识