Lab4实验报告

学院	化学化工学院	专业	应用化学
学号	221850092	姓名	周辉
Email	1664860843@qq.com		

1.实验名称

Lab4

2.实验目的

完成路由器的接收数据包，将其目标地址与转发表匹配，并将它们转发到正确的接口的功能

3.实验内容

克隆lab4仓库，复制`myrouter.py`到目录中, 修改`myrouter.py`实现转发数据包的功能

4.实验结果

Task2 建立转发表

实验要求从两个来源构建转发表:`net.interfaces()`和`forwarding_table.txt`

建立了一个新的类`forwarding_table_entry`用来表示转发表的表项

```
class forwarding_table_entry(object):
    def __init__(self, prefix, mask, next_ip, target):
        self.prefix = prefix
        self.mask = mask
        self.next_ip = next_ip
        self.target = target
```

`net.interfaces()`部分

```
for interface in self.interfaces:
ipaddr = IPv4Address(interface.ipaddr)
netmask = IPv4Address(interface.netmask)
prefix = IPv4Address(int(ipaddr) & int(netmask))
entry = forwarding_table_entry(prefix, netmask, None, interface)
self.forwarding_table.append(entry)
```

`forwarding_table.txt`部分

```
f = open("forwarding_table.txt")
lines = f.readlines()
for line in lines:
    line = line.strip('\n')
    items = line.split(" ")
    ipaddr = IPv4Address(items[0])
    netmask = IPv4Address(items[1])
    next_ip = IPv4Address(items[2])
    for interface in self.interfaces:
        if interface.name == items[3]:
            target = interface
            break
    else:
        target = None #如果target真的是None其实完全没有处理，会直接崩溃
    prefix = IPv4Address(int(ipaddr) & int(netmask))
    entry = forwarding_table_entry(prefix, netmask, next_ip, target)

    self.forwarding_table.append(entry)
```

这里使用了readlines()来读取文件数据

经过这两个过程, 就成功建立了转发表

如何匹配目标IP地址:

```
for entry in self.forwarding_table:
    entry:forwarding_table_entry
    prefix = int(head.dst) & int(entry.mask)
    prefixnet = IPv4Network(f"{entry.prefix}/{entry.mask}")
    if prefix == int(entry.prefix) and prefixnet.prefixlen > prefix_len: #匹配
        prefix_len = prefixnet.prefixlen
        matched_entry = entry
        matched = True
```

这里使用了`prefix == int(entry.prefix)`来判断是否匹配, 并且取前缀长度最大的匹配项

Task3 转发数据包和ARP

新建了一个`Packet_queue`类来表示待处理的数据包队列(实际上更像是目标IP的队列), 其类方法`handle(*self*, *packet*:Packet, *table_entry*, *net*)`用于发送ARP请求和记录ARP请求次数及时间

当收到新的数据包时, 按如下逻辑处理

```
if packet.has_header(IPv4):
    head: IPv4 = packet.get_header(IPv4)
    head.ttl -= 1

    #匹配过程#

    if matched:#匹配成功
        entry = matched_entry
        log_info(f"{packet} matched {entry.target}")
        packet_queue.handle(packet, entry, self.net)#交由packet_queue.handle处理(加入待处理队列)
```

如果应该生成并发送ARP请求, 则

```
taget_ip = table_entry.next_ip if table_entry.next_ip else packet[IPv4].dst
arp = Arp(operation = ArpOperation.Request,
        senderhwaddr = table_entry.target.ethaddr,
        senderprotoaddr = table_entry.target.ipaddr,
        targethwaddr = "ff:ff:ff:ff:ff:ff",
        targetprotoaddr = taget_ip)#建立ARP header
arp_request = Ethernet(src=table_entry.target.ethaddr,#Ethernet header
                        dst="ff:ff:ff:ff:ff:ff",
                        ethertype=EtherType.ARP) + arp#两个header相加，形成ARP包
net.send_packet(table_entry.target.name, arp_request)#发送ARP包
entry.last_request_time = time.time()
entry.request_nums += 1#记录请求时间和次数
```

测试结果

`testscenario2.srpy`

```
Passed:
1   IP packet to be forwarded to 172.16.42.2 should arrive on
    router-eth0
        Expected event: recv_packet Ethernet
        10:00:00:00:00:03->30:00:00:00:00:01 IP | IPv4
        192.168.1.100->172.16.42.2 ICMP | ICMP EchoRequest 0 42 (0
        data bytes) on router-eth0
2   Router should send ARP request for 172.16.42.2 out router-
    eth2 interface
        Expected event: send_packet(s) Ethernet
        10:00:00:00:00:03->ff:ff:ff:ff:ff:ff ARP | Arp
        10:00:00:00:00:03:172.16.42.1 ff:ff:ff:ff:ff:ff:172.16.42.2
        out router-eth2
3   Router should receive ARP response for 172.16.42.2 on
    router-eth2 interface
        Expected event: recv_packet Ethernet
        30:00:00:00:00:01->10:00:00:00:00:03 ARP | Arp
        30:00:00:00:00:01:172.16.42.2 10:00:00:00:00:03:172.16.42.1
        on router-eth2
4   IP packet should be forwarded to 172.16.42.2 out router-eth2
        Expected event: send_packet(s) Ethernet
        10:00:00:00:00:03->30:00:00:00:00:01 IP | IPv4
        192.168.1.100->172.16.42.2 ICMP | ICMP EchoRequest 0 42 (0
        data bytes) out router-eth2
5   IP packet to be forwarded to 192.168.1.100 should arrive on
    router-eth2
        Expected event: recv_packet Ethernet
        10:00:00:00:00:01->20:00:00:00:00:01 IP | IPv4
        172.16.42.2->192.168.1.100 ICMP | ICMP EchoReply 0 42 (0
        data bytes) on router-eth2
6   Router should send ARP request for 192.168.1.100 out router-
    eth0
        Expected event: send_packet(s) 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:192.168.1.100 out router-eth0
7   Router should receive ARP response for 192.168.1.100 on
    router-eth0
```

```
        Expected event: recv_packet Ethernet
        20:00:00:00:00:01->10:00:00:00:00:01 ARP | Arp
        20:00:00:00:00:01:192.168.1.100
        10:00:00:00:00:01:192.168.1.1 on router-eth0
8   IP packet should be forwarded to 192.168.1.100 out router-
    eth0
        Expected event: send_packet(s) Ethernet
        10:00:00:00:00:01->20:00:00:00:00:01 IP | IPv4
        172.16.42.2->192.168.1.100 ICMP | ICMP EchoReply 0 42 (0
        data bytes) out router-eth0
9   Another IP packet for 172.16.42.2 should arrive on router-
    eth0
        Expected event: recv_packet Ethernet
        10:00:00:00:00:03->30:00:00:00:00:01 IP | IPv4
        192.168.1.100->172.16.42.2 ICMP | ICMP EchoRequest 0 42 (0
        data bytes) on router-eth0
10  IP packet should be forwarded to 172.16.42.2 out router-eth2
    (no ARP request should be necessary since the information
    from a recent ARP request should be cached)
        Expected event: send_packet(s) Ethernet
        10:00:00:00:00:03->30:00:00:00:00:01 IP | IPv4
        192.168.1.100->172.16.42.2 ICMP | ICMP EchoRequest 0 42 (0
        data bytes) out router-eth2
11  IP packet to be forwarded to 192.168.1.100 should arrive on
    router-eth2
        Expected event: recv_packet Ethernet
        10:00:00:00:00:01->20:00:00:00:00:01 IP | IPv4
        172.16.42.2->192.168.1.100 ICMP | ICMP EchoReply 0 42 (0
        data bytes) on router-eth2
12  IP packet should be forwarded to 192.168.1.100 out router-
    eth0 (again, no ARP request should be necessary since the
    information from a recent ARP request should be cached)
        Expected event: send_packet(s) Ethernet
        10:00:00:00:00:01->20:00:00:00:00:01 IP | IPv4
        172.16.42.2->192.168.1.100 ICMP | ICMP EchoReply 0 42 (0
        data bytes) out router-eth0
13  An IP packet from 10.100.1.55 to 172.16.64.35 should arrive
    on router-eth1
        Expected event: recv_packet Ethernet
        10:00:00:00:00:02->11:22:33:44:55:66 IP | IPv4
        10.100.1.55->172.16.64.35 ICMP | ICMP EchoRequest 0 42 (0
        data bytes) on router-eth2
14  Router should send an ARP request for 10.10.1.254 on router-
    eth1
        Expected event: send_packet(s) 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
        out router-eth1
15  Application should try to receive a packet, but then timeout
        Expected event: Timeout after 1.5s on a call to recv_packet
16  Router should send another an ARP request for 10.10.1.254 on
    router-eth1 because of a slow response
        Expected event: send_packet(s) 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
        out router-eth1
```

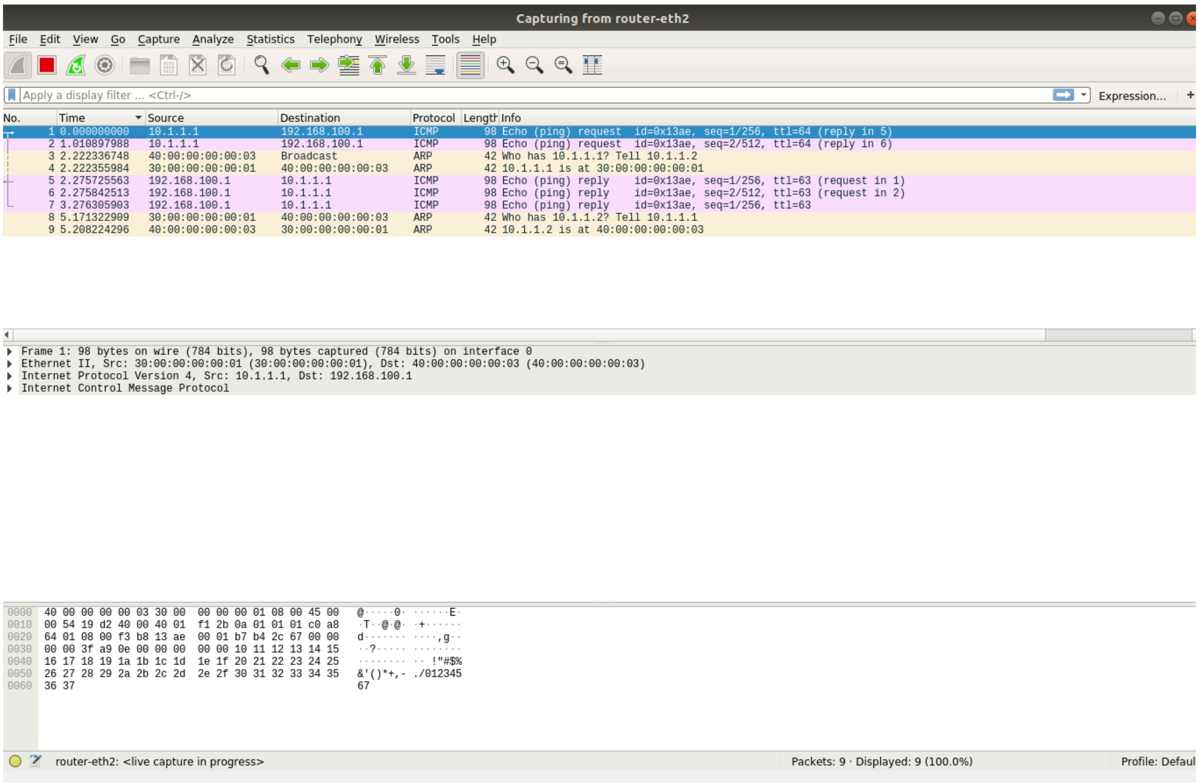
```
17  Router should receive an ARP response for 10.10.1.254 on
    router-eth1
        Expected event: recv_packet Ethernet
        11:22:33:44:55:66->10:00:00:00:00:02 ARP | Arp
        11:22:33:44:55:66:10.10.1.254 10:00:00:00:00:02:10.10.0.1 on
        router-eth1
18  IP packet destined to 172.16.64.35 should be forwarded on
    router-eth1
        Expected event: send_packet(s) Ethernet
        10:00:00:00:00:02->11:22:33:44:55:66 IP | IPv4
        10.100.1.55->172.16.64.35 ICMP | ICMP EchoRequest 0 42 (0
        data bytes) out router-eth1
19  An IP packet from 192.168.1.239 for 10.200.1.1 should arrive
    on router-eth0.  No forwarding table entry should match.
        Expected event: recv_packet Ethernet
        ab:cd:ef:ab:cd:ef->10:00:00:00:00:01 IP | IPv4
        192.168.1.239->10.200.1.1 ICMP | ICMP EchoRequest 0 42 (0
        data bytes) on router-eth0
20  An IP packet from 192.168.1.239 for 10.10.50.250 should
    arrive on router-eth0.
        Expected event: recv_packet Ethernet
        ab:cd:ef:ab:cd:ef->10:00:00:00:00:01 IP | IPv4
        192.168.1.239->10.10.50.250 ICMP | ICMP EchoRequest 0 42 (0
        data bytes) on router-eth0
21  Router should send an ARP request for 10.10.50.250 on
    router-eth1
        Expected event: send_packet(s) 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
        out router-eth1
22  Router should try to receive a packet (ARP response), but
    then timeout
        Expected event: Timeout after 1.5s on a call to recv_packet
23  Router should send an ARP request for 10.10.50.250 on
    router-eth1
        Expected event: send_packet(s) 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
        out router-eth1
24  Router should try to receive a packet (ARP response), but
    then timeout
        Expected event: Timeout after 1.5s on a call to recv_packet
25  Router should send an ARP request for 10.10.50.250 on
    router-eth1
        Expected event: send_packet(s) 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
        out router-eth1
26  Router should try to receive a packet (ARP response), but
    then timeout
        Expected event: Timeout after 1.5s on a call to recv_packet
27  Router should send an ARP request for 10.10.50.250 on
    router-eth1
        Expected event: send_packet(s) 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
        out router-eth1
28  Router should try to receive a packet (ARP response), but
    then timeout
        Expected event: Timeout after 1.5s on a call to recv_packet
29  Router should send an ARP request for 10.10.50.250 on
    router-eth1
        Expected event: send_packet(s) 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
        out router-eth1
30  Router should try to receive a packet (ARP response), but
    then timeout
        Expected event: Timeout after 1.5s on a call to recv_packet
31  Router should try to receive a packet (ARP response), but
    then timeout
        Expected event: Timeout after 1.5s on a call to recv_packet


All tests passed!
```

部署

在client运行`ping -c2 192.168.100.1`即client ping server1

router和server1的抓包结果分别如下:

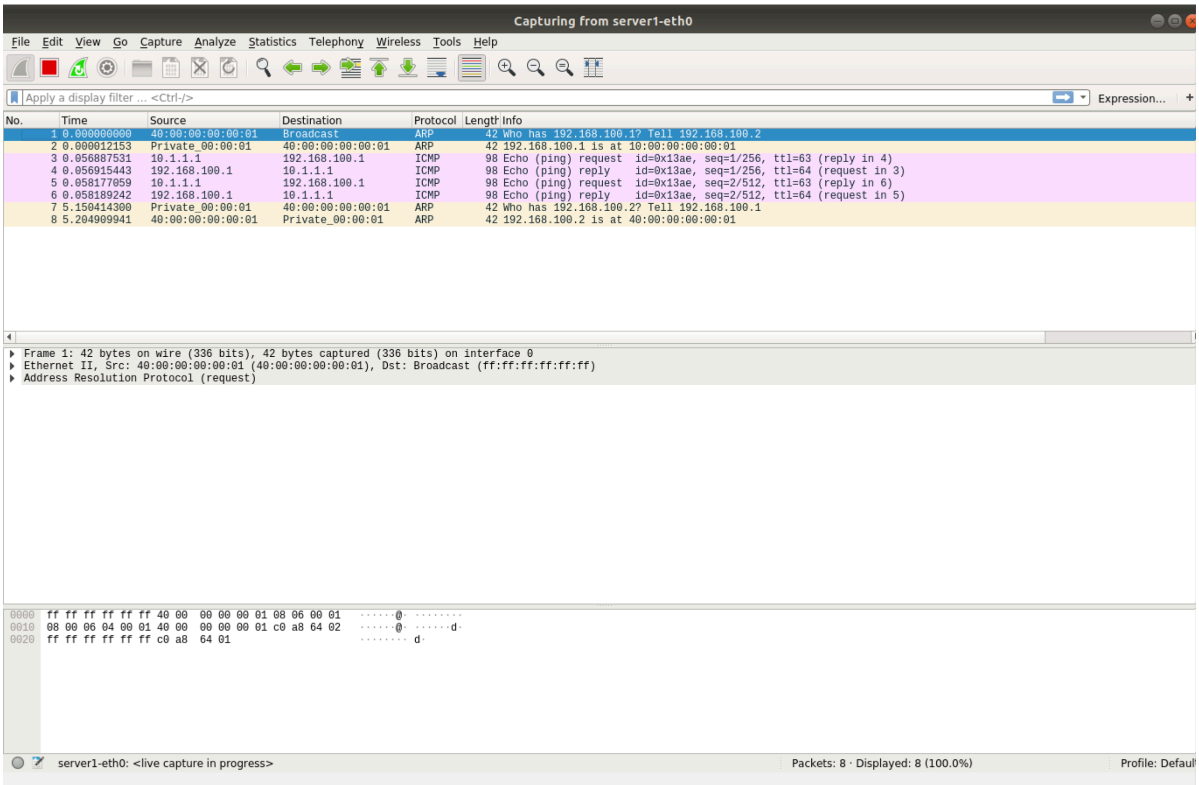

Capturing from router-eth2
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
+

```
No.  Time         Source             Destination        Protocol Length Info
   1 0.000000000  10.1.1.1           192.168.100.1      ICMP      98 Echo (ping) request  id=0x13ae, seq=1/256, ttl=64 (reply in 5)
   2 1.010897988  10.1.1.1           192.168.100.1      ICMP      98 Echo (ping) request  id=0x13ae, seq=2/512, ttl=64 (reply in 6)
   3 2.222336748  40:00:00:00:00:03  Broadcast          ARP       42 Who has 10.1.1.1? Tell 10.1.1.2
   4 2.222355984  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
   5 2.275725563  192.168.100.1      10.1.1.1           ICMP      98 Echo (ping) reply    id=0x13ae, seq=1/256, ttl=63 (request in 1)
   6 2.275842513  192.168.100.1      10.1.1.1           ICMP      98 Echo (ping) reply    id=0x13ae, seq=2/512, ttl=63 (request in 2)
   7 3.276305903  192.168.100.1      10.1.1.1           ICMP      98 Echo (ping) reply    id=0x13ae, seq=1/256, ttl=63
   8 5.171322909  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
   9 5.208224296  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 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▸ Ethernet II, Src: 30:00:00:00:00:01 (30:00:00:00:00:01), Dst: 40:00:00:00:00:03 (40:00:00:00:00:03)
▸ Internet Protocol Version 4, Src: 10.1.1.1, Dst: 192.168.100.1
▸ Internet Control Message Protocol
```

```
0000  40 00 00 00 00 03 30 00  00 00 00 01 08 00 45 00   @·····0· ······E·
0010  00 54 19 d2 40 00 40 01  f1 2b 0a 01 01 01 c0 a8   ·T··@·@· ·+······
0020  64 01 08 00 f3 b8 13 ae  00 01 b7 b4 2c 67 00 00   d······· ····,g··
0030  00 00 3f a9 0e 00 00 00  00 00 10 11 12 13 14 15   ··?····· ········
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ········ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67
```

router-eth2: <live capture in progress>   Packets: 9 · Displayed: 9 (100.0%)   Profile: Defaul



Capturing from server1-eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…
+

```
No.    Time         Source              Destination         Protocol Length Info
     1 0.000000000  40:00:00:00:00:01   Broadcast           ARP       42 Who has 192.168.100.1? Tell 192.168.100.2
     2 0.000012153  Private_00:00:01    40:00:00:00:00:01   ARP       42 192.168.100.1 is at 10:00:00:00:00:01
     3 0.056887531  10.1.1.1            192.168.100.1       ICMP      98 Echo (ping) request  id=0x13ae, seq=1/256, ttl=63 (reply in 4)
     4 0.056915443  192.168.100.1       10.1.1.1            ICMP      98 Echo (ping) reply    id=0x13ae, seq=1/256, ttl=64 (request in 3)
     5 0.058177059  10.1.1.1            192.168.100.1       ICMP      98 Echo (ping) request  id=0x13ae, seq=2/512, ttl=63 (reply in 6)
     6 0.058189242  192.168.100.1       10.1.1.1            ICMP      98 Echo (ping) reply    id=0x13ae, seq=2/512, ttl=64 (request in 5)
     7 5.150414300  Private_00:00:01    40:00:00:00:00:01   ARP       42 Who has 192.168.100.2? Tell 192.168.100.1
     8 5.204909941  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 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▸ Ethernet II, Src: 40:00:00:00:00:01 (40:00:00:00:00:01), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▸ Address Resolution Protocol (request)
```

```
0000  ff ff ff ff ff ff 40 00  00 00 00 01 08 06 00 01   ······@· ········
0010  08 00 06 04 00 01 40 00  00 00 00 01 c0 a8 64 02   ······@· ······d·
0020  ff ff ff ff ff ff c0 a8  64 01                     ········ d·
```

server1-eth0: <live capture in progress>   Packets: 8 · Displayed: 8 (100.0%)   Profile: Defaul

可以看到, router正常发送了ARP request并且受到回复后转发了Echo包, 运行如预期