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-Forwarding Packets

2. 实验目的

- 转发目的地址为其他主机的包，通过最长前缀匹配找到需要转发的子网
- 通过arp请求得到对应要转发地址的mac地址，并转发

3. 实验内容

Task 1: Preparation

配置实验环境

Task 2: IP Forwarding Table Lookup

得到转发表得到从哪个端口转发的信息

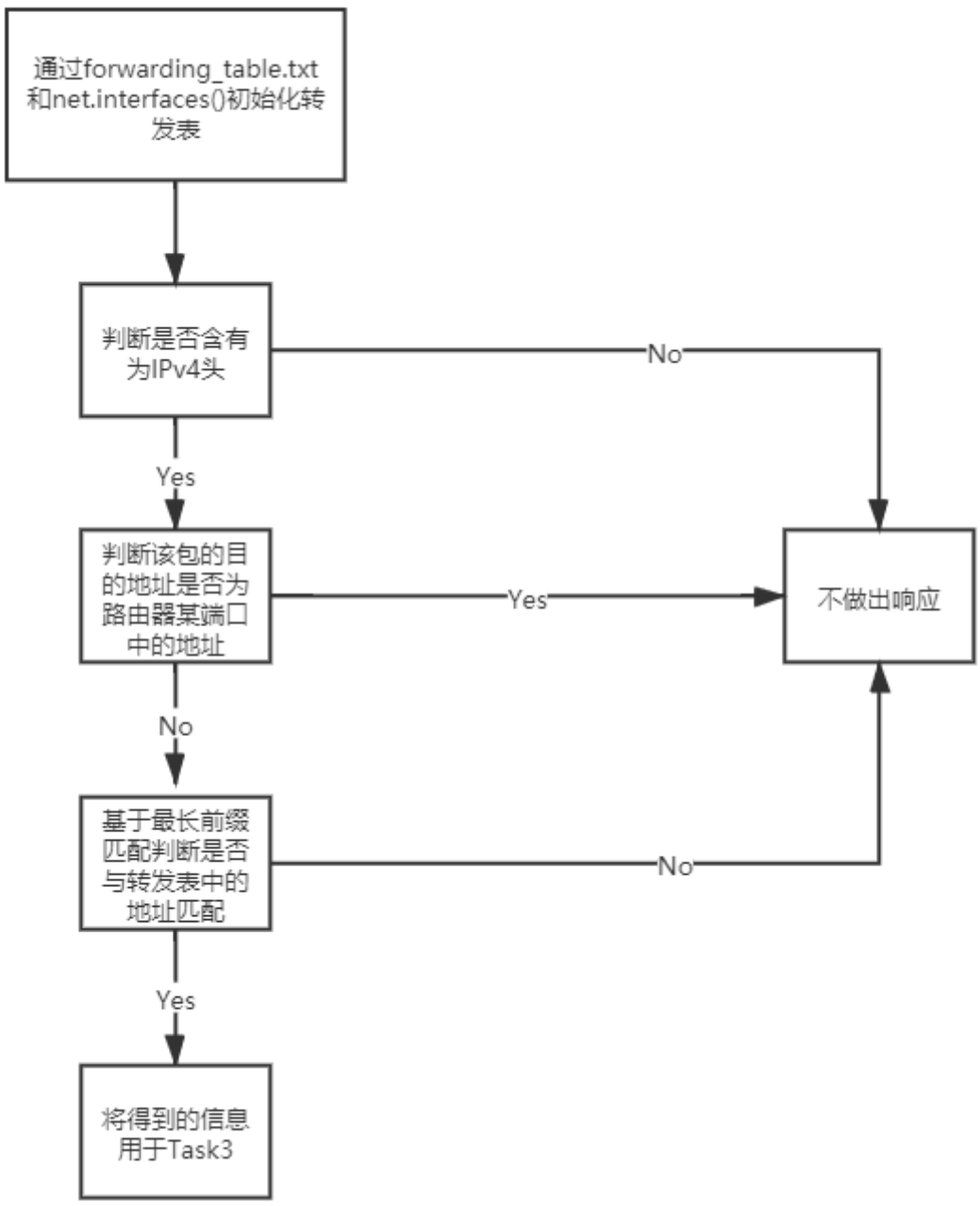
Task 3: Forwarding the Packet and ARP

通过ARP请求得到消息转发包

4. 实验结果

Task 2: IP Forwarding Table Lookup

the logic of building IP forwarding table and matching the destination IP addresses



通过 `forwarding_table.txt` 和 `net.interfaces()` 初始化转发表：

- 利用 `forwarding_table.txt` 初始化，以空格为分割字符直接读 `forwarding_table.txt` 即可；
- 利用 `net.interfaces()` 初始化，可以得到路由器各个端口对应的子网掩码、端口对应的子网地址以及端口信息。

其中对应下一跳地址：

- 如果下一跳地址不为 `0.0.0.0` 则表示还需要将该包转发到下一跳路由器中，并将目的ip地址改为下一跳地址；

```
dstip = IPv4Address(self.forwarding_table[match_subnet][1])
```

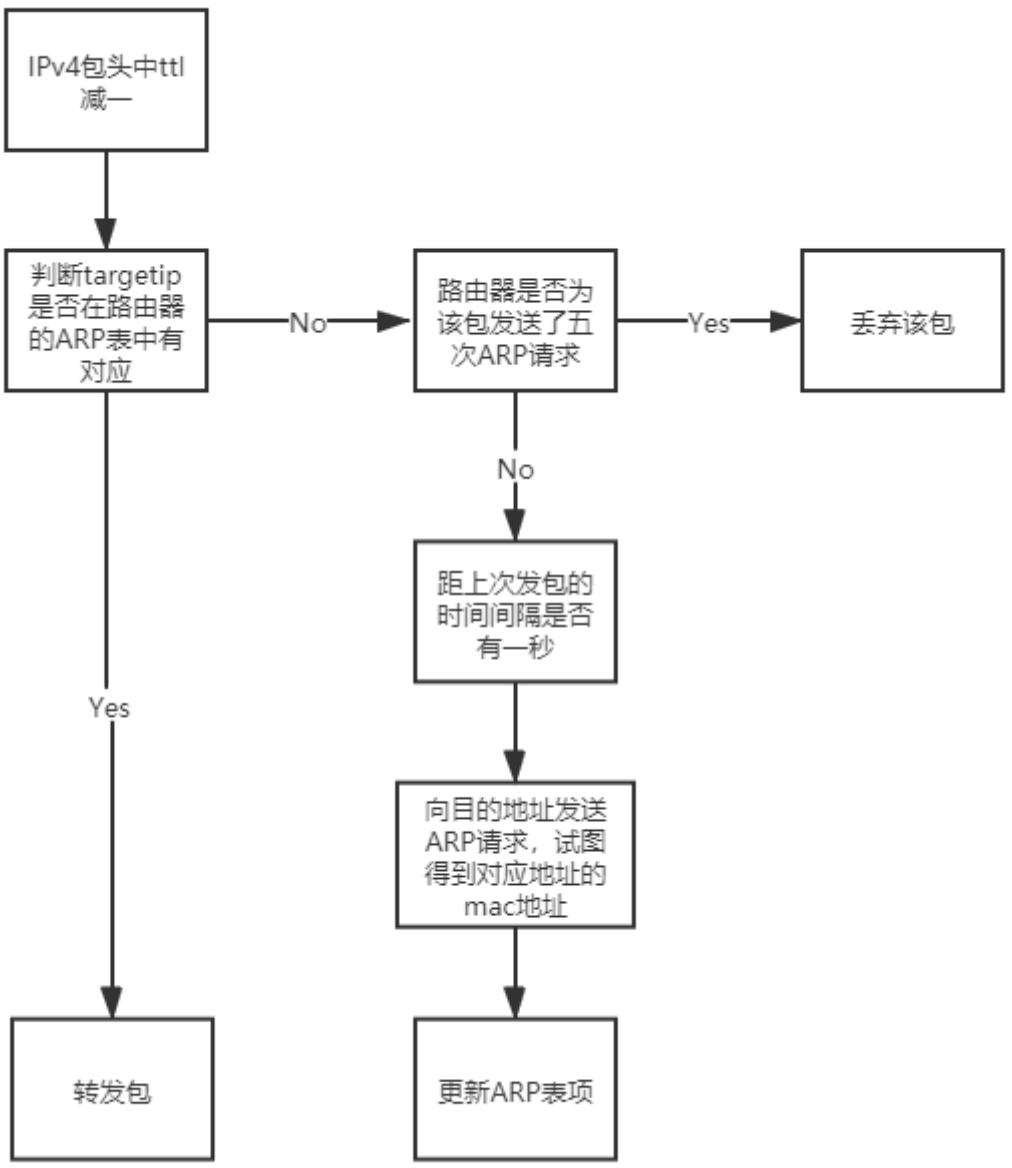
- 如果为 `0.0.0.0` ，因为包与路由器端口的子网地址匹配意味着，该端口对应的子网即为目的子网，目的ip地址设置为IPv4包中的IP地址即可。

```
dstip = IPv4Address(ipv4.dst)
```

当运行时，接收到发送给其他节点的包，查询转发表中匹配的内容得到要转发的子网地址，从而得知该包需要从哪个端口转发

Task 3: Forwarding the Packet and ARP

the logic of forwarding the packet and ARP



在上一节中已经得知要从哪个端口转发，现在已经知道目的ip地址和路由器的mac地址以及源ip地址，还缺少一个目的地址的mac地址即可转发这个包。所以先查询路由器的arp表，如果表中有对应的信息，直接转发即可；如果没有还需要发送arp请求得到目的地址的mac地址才可转发。

其中转发的过程应该在 `start` 函数的第三行位置处运行：

```
1  def start(self):
2      while True:
3          self.forwarding()
4          try:
5              recv = self.net.recv_packet(timeout=1.0)
6          except NoPackets:
7              continue
8          except Shutdown:
9              break
10         self.handle_packet(recv)
11         self.stop()
```

如果 `forwarding()` 函数在try语句块之后运行会导致只有在接收到新的包之后才会转发，所以 `forwarding()` 函数应该在try语句块之前运行。

Running in the Test Environment

在terminal中执行：

```
1  swyard -t testcases/myrouter2_testscenario.srpy myrouter.py
```

```
(syenv) njucs@njucs-VirtualBox:~/sy/lab-4-vectormoon$ swyard -t testcases/myrouter2_testscenario.srpy myrouter.py
14:53:08 2021/05/06     INFO Starting test scenario testcases/myrouter2_testscenario.srpy
14:53:08 2021/05/06     INFO ------------------------forwarding_table_info-----------------------
14:53:08 2021/05/06     INFO         address: 192.168.1.0;  other: [IPv4Address('255.255.255.0'), '0.0.0.0', 'router-eth0'].
14:53:08 2021/05/06     INFO         address: 10.10.0.0;  other: [IPv4Address('255.255.0.0'), '0.0.0.0', 'router-eth1'].
14:53:08 2021/05/06     INFO         address: 172.16.42.0;  other: [IPv4Address('255.255.255.252'), '0.0.0.0', 'router-eth2'].
14:53:08 2021/05/06     INFO ------------------------------------------------------------------
14:53:08 2021/05/06     INFO ------------------------forwarding_table_info-----------------------
14:53:08 2021/05/06     INFO         address: 192.168.1.0;  other: [IPv4Address('255.255.255.0'), '0.0.0.0', 'router-eth0'].
14:53:08 2021/05/06     INFO         address: 10.10.0.0;  other: [IPv4Address('255.255.0.0'), '0.0.0.0', 'router-eth1'].
14:53:08 2021/05/06     INFO         address: 172.16.42.0;  other: [IPv4Address('255.255.255.252'), '0.0.0.0', 'router-eth2'].
14:53:08 2021/05/06     INFO         address: 172.16.0.0;  other: [IPv4Address('255.255.0.0'), IPv4Address('192.168.1.2'), 'router-eth0'].
14:53:08 2021/05/06     INFO         address: 172.16.128.0;  other: [IPv4Address('255.255.192.0'), IPv4Address('10.10.0.254'), 'router-eth1'].
14:53:08 2021/05/06     INFO         address: 172.16.64.0;  other: [IPv4Address('255.255.192.0'), IPv4Address('10.10.1.254'), 'router-eth1'].
14:53:08 2021/05/06     INFO         address: 10.100.0.0;  other: [IPv4Address('255.255.0.0'), IPv4Address('172.16.42.2'), 'router-eth2'].
14:53:08 2021/05/06     INFO ------------------------------------------------------------------
14:53:08 2021/05/06     INFO handle_packet
14:53:09 2021/05/06     INFO Sending arp request 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 to router-eth2
14:53:09 2021/05/06     INFO handle_packet
14:53:09 2021/05/06     INFO ------------------------arp_table_info-----------------------
14:53:09 2021/05/06     INFO         IP: 172.16.42.2; MAC Address: 30:00:00:00:00:01
14:53:09 2021/05/06     INFO ------------------------------------------------------------
14:53:09 2021/05/06     INFO Forwarding 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) to router-eth2
14:53:09 2021/05/06     INFO handle_packet
14:53:10 2021/05/06     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:192.168.1.100 to router-eth0
14:53:10 2021/05/06     INFO handle_packet
14:53:10 2021/05/06     INFO ------------------------arp_table_info-----------------------
14:53:10 2021/05/06     INFO         IP: 172.16.42.2; MAC Address: 30:00:00:00:00:01
14:53:10 2021/05/06     INFO         IP: 192.168.1.100; MAC Address: 20:00:00:00:00:01
14:53:10 2021/05/06     INFO ------------------------------------------------------------
14:53:10 2021/05/06     INFO Forwarding 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) to router-eth0
14:53:10 2021/05/06     INFO handle_packet
14:53:10 2021/05/06     INFO Forwarding 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) to router-eth2
14:53:10 2021/05/06     INFO handle_packet
14:53:10 2021/05/06     INFO Forwarding 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) to router-eth0
14:53:10 2021/05/06     INFO handle_packet
14:53:11 2021/05/06     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
14:53:13 2021/05/06     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
14:53:13 2021/05/06     INFO handle_packet
14:53:13 2021/05/06     INFO ------------------------arp_table_info-----------------------
14:53:13 2021/05/06     INFO         IP: 172.16.42.2; MAC Address: 30:00:00:00:00:01
14:53:13 2021/05/06     INFO         IP: 192.168.1.100; MAC Address: 20:00:00:00:00:01
14:53:13 2021/05/06     INFO         IP: 10.10.1.254; MAC Address: 11:22:33:44:55:66
14:53:13 2021/05/06     INFO ------------------------------------------------------------
14:53:13 2021/05/06     INFO Forwarding 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) to router-eth1
14:53:13 2021/05/06     INFO handle_packet
14:53:13 2021/05/06     INFO handle_packet
14:53:14 2021/05/06     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
14:53:15 2021/05/06     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
14:53:17 2021/05/06     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
14:53:18 2021/05/06     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
14:53:20 2021/05/06     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
14:53:21 2021/05/06     INFO Delete 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)
```

```
Results for test scenario IP forwarding and ARP requester tests: 31 passed, 0 failed, 0 pending


Passed:
1   IP packet to be forwarded to 172.16.42.2 should arrive on
    router-eth0
2   Router should send ARP request for 172.16.42.2 out router-
    eth2 interface
3   Router should receive ARP response for 172.16.42.2 on
    router-eth2 interface
4   IP packet should be forwarded to 172.16.42.2 out router-eth2
5   IP packet to be forwarded to 192.168.1.100 should arrive on
    router-eth2
6   Router should send ARP request for 192.168.1.100 out router-
    eth0
7   Router should receive ARP response for 192.168.1.100 on
    router-eth0
8   IP packet should be forwarded to 192.168.1.100 out router-
    eth0
9   Another IP packet for 172.16.42.2 should arrive 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)
11  IP packet to be forwarded to 192.168.1.100 should arrive 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)
13  An IP packet from 10.100.1.55 to 172.16.64.35 should arrive
    on router-eth1
14  Router should send an ARP request for 10.10.1.254 on router-
    eth1
15  Application should try to receive a packet, but then timeout
16  Router should send another an ARP request for 10.10.1.254 on
    router-eth1 because of a slow response
17  Router should receive an ARP response for 10.10.1.254 on
    router-eth1
18  IP packet destined to 172.16.64.35 should be forwarded on
    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.
20  An IP packet from 192.168.1.239 for 10.10.50.250 should
    arrive on router-eth0.
21  Router should send an ARP request for 10.10.50.250 on
    router-eth1
22  Router should try to receive a packet (ARP response), but
    then timeout
23  Router should send an ARP request for 10.10.50.250 on
    router-eth1
24  Router should try to receive a packet (ARP response), but
    then timeout
25  Router should send an ARP request for 10.10.50.250 on
    router-eth1
26  Router should try to receive a packet (ARP response), but
    then timeout
27  Router should send an ARP request for 10.10.50.250 on
    router-eth1
28  Router should try to receive a packet (ARP response), but
    then timeout
29  Router should send an ARP request for 10.10.50.250 on
    router-eth1
30  Router should try to receive a packet (ARP response), but
    then timeout
31  Router should try to receive a packet (ARP response), but
    then timeout


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mininet中启动 `server1` 和 `router`

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

利用wireshark抓 `router-eth0` 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`

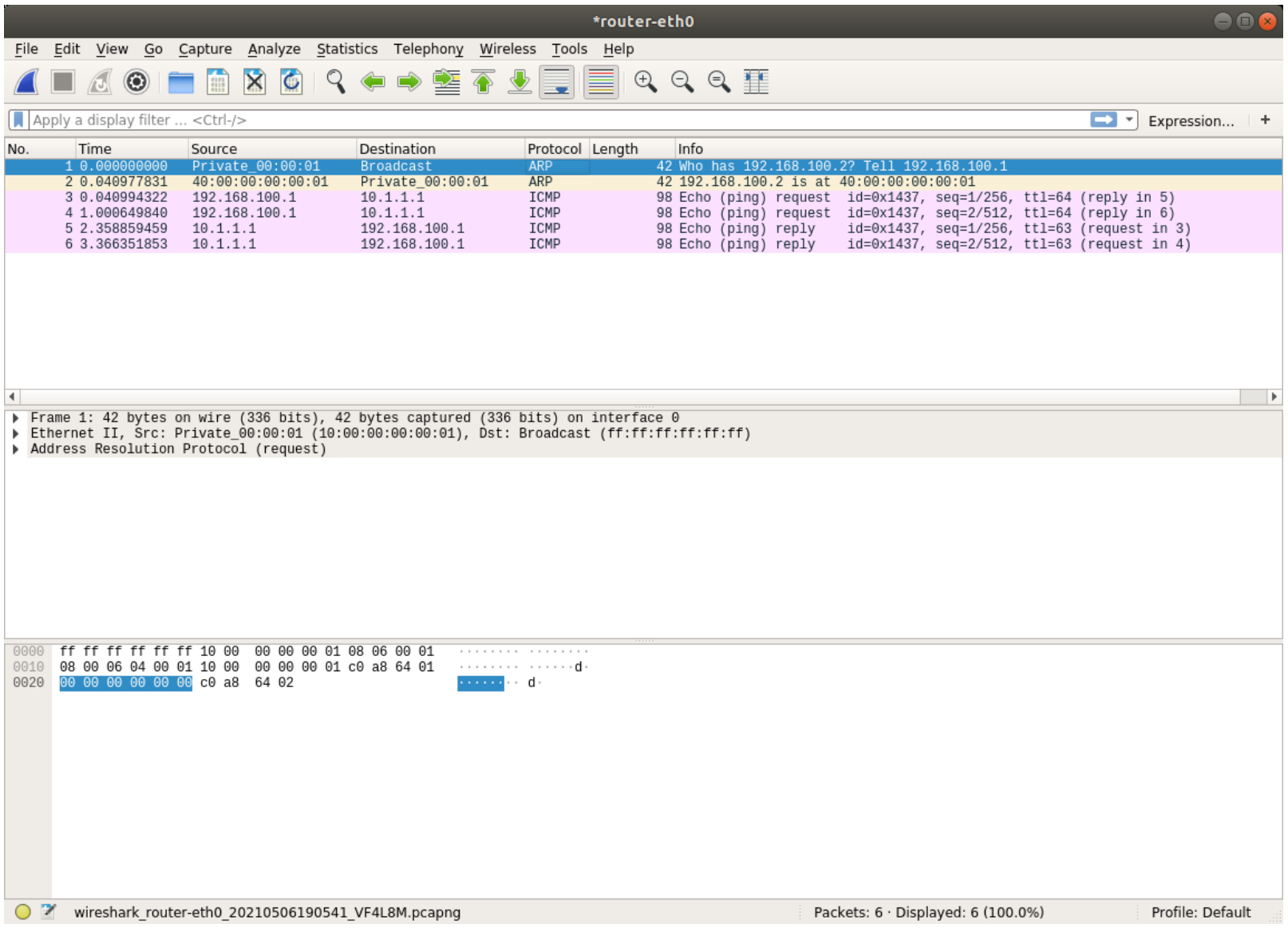
```
(syenv) router# swyard myrouter.py
```

在 `server1` 的xterm中，使用以下指令测试

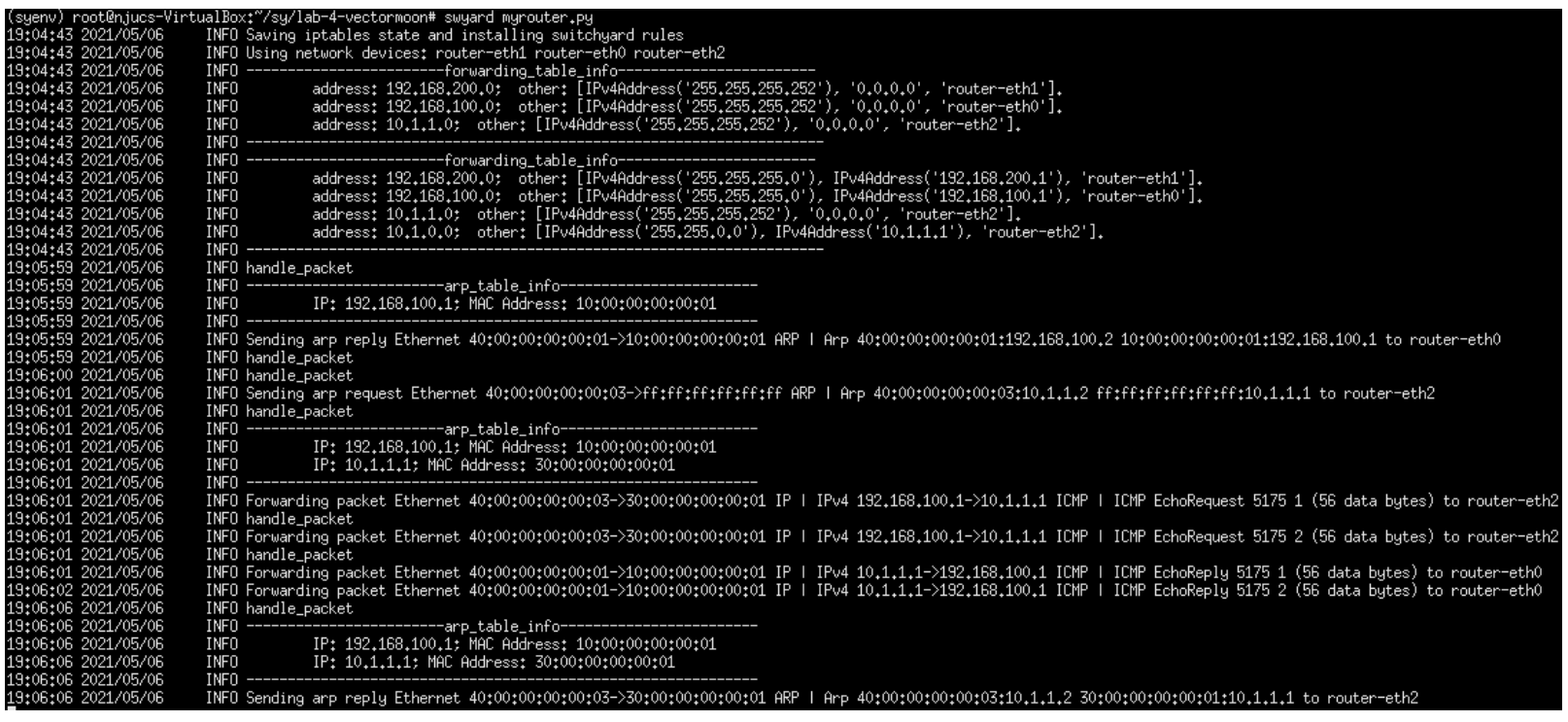
```
server1# ping -c2 10.1.1.1
```

得到如下结果：

wireshark中的结果



xterm router中结果



`server1` 首先要将数据包发给 `router` 所以向 `router` 发送ARP请求试图获取 `router`对应端口的mac地址；随后 `router` 向 `server1` 发送ARP包回复；在上面xterm的结果可以得知，`40:00:00:00:00:03` 端口回发送ARP request，试图得到ip地址为 `10.1.1.1` 的mac地址(也即 `client` 的mac地址)。在得到ARP reply之后会更新表项，在那之后因为已经得到转发所需要的全部消息，只需要转发即可。

5. 核心代码

Task 2: IP Forwarding Table Lookup

```
for address in self.forwarding_table.keys():
    prefix = IPv4Address(address)
    destaddr = ipv4.dst
    matches = (int(prefix) & int(destaddr)) == int(prefix)
    if (matches):
        netaddr = IPv4Network(str(address) + '/' +
str(self.forwarding_table[address][0]))
        if (netaddr.prefixlen > prefix_len):
            prefix_len = netaddr.prefixlen
            match_subnet = IPv4Address(address)
```

最长子网匹配，通过第四行代码确定目的地址和转发表中的地址是否匹配；第七行代码来判断是否为当前最长匹配，如果不是则更新最长匹配子网。

Task 3: Forwarding the Packet and ARP

```
def forwarding(self):
    if (targetipaddr in self.arp_table.keys()):
        # search arp table
        self.forwarding_packet(my_packet, router_send_to_host_port_name,
targetipaddr, router_forwarding_port_info)
    elif (handle_pkt.get_num_of_retries() < 5):
        # send arp request
        self.send_arp_request(handle_pkt, router_forwarding_port_info,
targetipaddr, router_send_to_host_port_name)
    elif (handle_pkt.get_num_of_retries() >= 5):
        # delete
        log_info (f"Delete packet {self.packet_queue[0].get_packet()}")
        del(self.packet_queue[0])
```

首先需要判断目的地址在路由器的ARP表项中是否有对应项，如果有的话直接转发即可；如果没有则需要发送ARP请求得到目的地址的mac地址，且会在发送五次ARP请求之后没有响应丢弃该包。

6. 总结与感想

在本实验中对路由器转发机制理解更加深刻，一子网的主机要向另一子网发送数据包则需要先将包发送到路由器中，在路由器中进行适当的处理，得到相应的消息之后，再将路由器中包转发给下一跳路由器或者目标子网对应的主机中。