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	**工程管理学院**	专业（方向）	**计算机金融**
学号	**211275041**	姓名	**杨晨毅**
Email	**2653024890@qq.com**	开始/完成日期	**5.1-5.4**

1. 实验名称

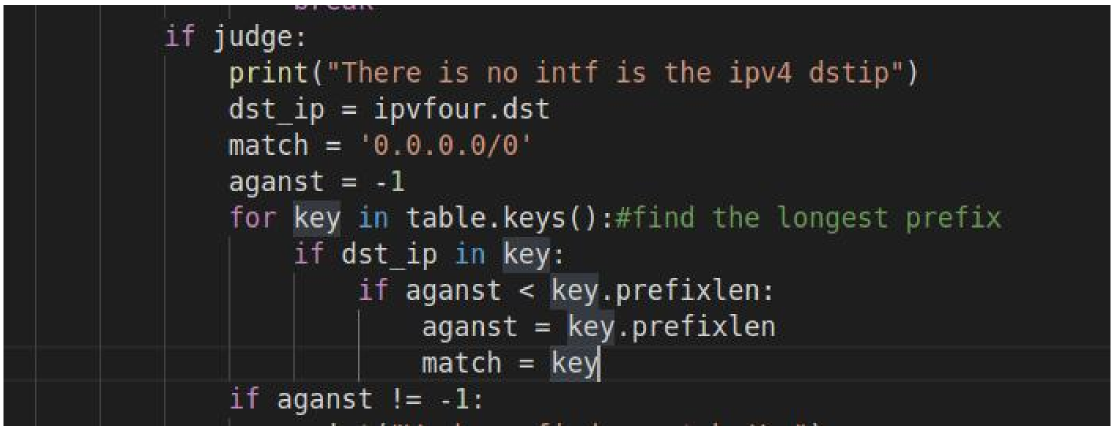
Forwarding TableForwarding Table

2. 实验目的

完成路由表的构建、匹配，以及包的转发

3. 实验内容

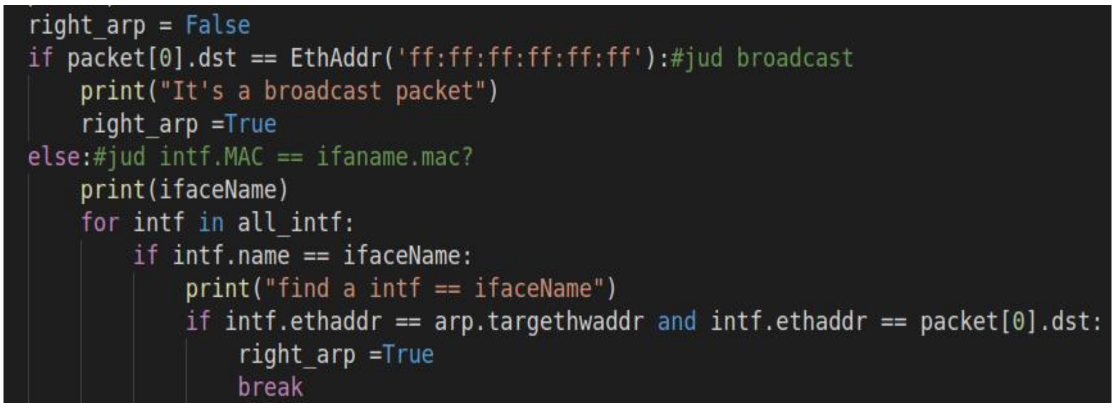
匹配目的 ip 地址



```
        if judge:
            print("There is no intf is the ipv4 dstip")
            dst_ip = ipvfour.dst
            match = '0.0.0.0/0'
            aganst = -1
            for key in table.keys():#find the longest prefix
                if dst_ip in key:
                    if aganst < key.prefixlen:
                        aganst = key.prefixlen
                        match = key
            if aganst != -1:
```

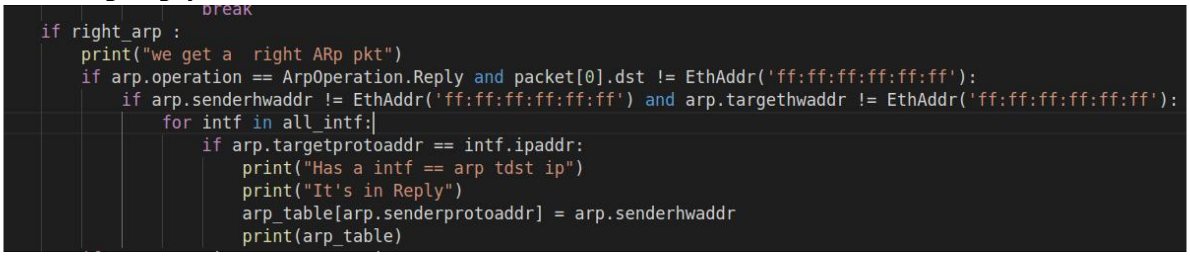
收发包与 ARP 请求发送

1. 收包时候判断 Ethnet 地址：



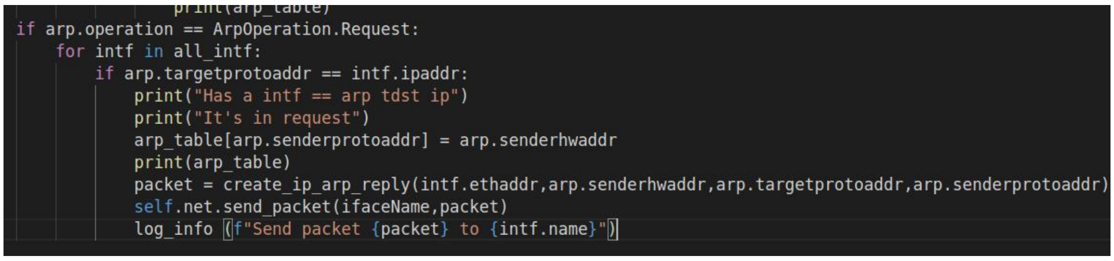
```
right_arp = False
if packet[0].dst == EthAddr('ff:ff:ff:ff:ff:ff'):#jud broadcast
    print("It's a broadcast packet")
    right_arp =True
else:#jud intf.MAC == ifaname.mac?
    print(ifaceName)
    for intf in all_intf:
        if intf.name == ifaceName:
            print("find a intf == ifaceName")
            if intf.ethaddr == arp.targethwaddr and intf.ethaddr == packet[0].dst:
                right_arp =True
                break
```

对于 **arp reply** 包：判断是否有地址是 **broadcast**，若没有，存表



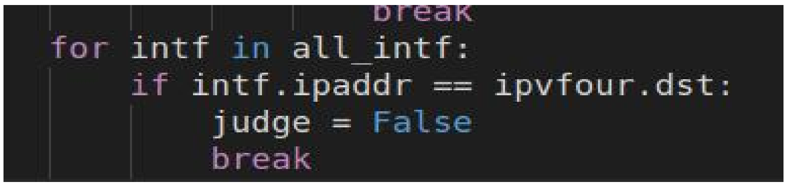
```
                    break
    if right_arp :
        print("we get a  right ARp pkt")
        if arp.operation == ArpOperation.Reply and packet[0].dst != EthAddr('ff:ff:ff:ff:ff:ff'):
            if arp.senderhwaddr != EthAddr('ff:ff:ff:ff:ff:ff') and arp.targethwaddr != EthAddr('ff:ff:ff:ff:ff:ff'):
                for intf in all_intf:
                    if arp.targetprotoaddr == intf.ipaddr:
                        print("Has a intf == arp tdst ip")
                        print("It's in Reply")
                        arp_table[arp.senderprotoaddr] = arp.senderhwaddr
                        print(arp_table)
```

对于 **arp request** 包：则从传入端口发回 **arp reply** 包



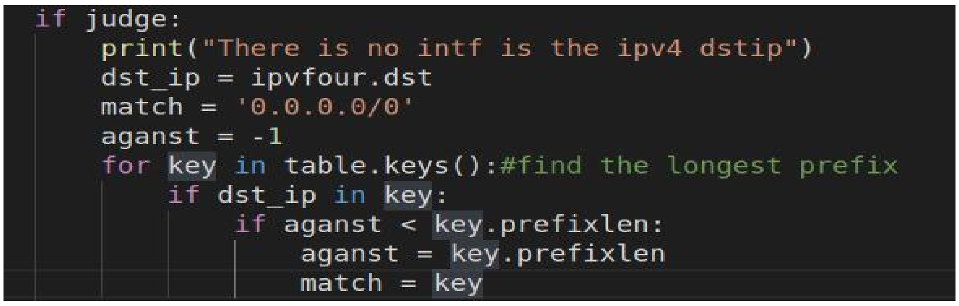
```
                print(arp_table)
if arp.operation == ArpOperation.Request:
    for intf in all_intf:
        if arp.targetprotoaddr == intf.ipaddr:
            print("Has a intf == arp tdst ip")
            print("It's in request")
            arp_table[arp.senderprotoaddr] = arp.senderhwaddr
            print(arp_table)
            packet = create_ip_arp_reply(intf.ethaddr,arp.senderhwaddr,arp.targetprotoaddr,arp.senderprotoaddr)
            self.net.send_packet(ifaceName,packet)
            log_info (f"Send packet {packet} to {intf.name}")
```

对 **ipv4** 包：首先判断目的 **ip** 是否是端口 **ip**，若是，则丢弃



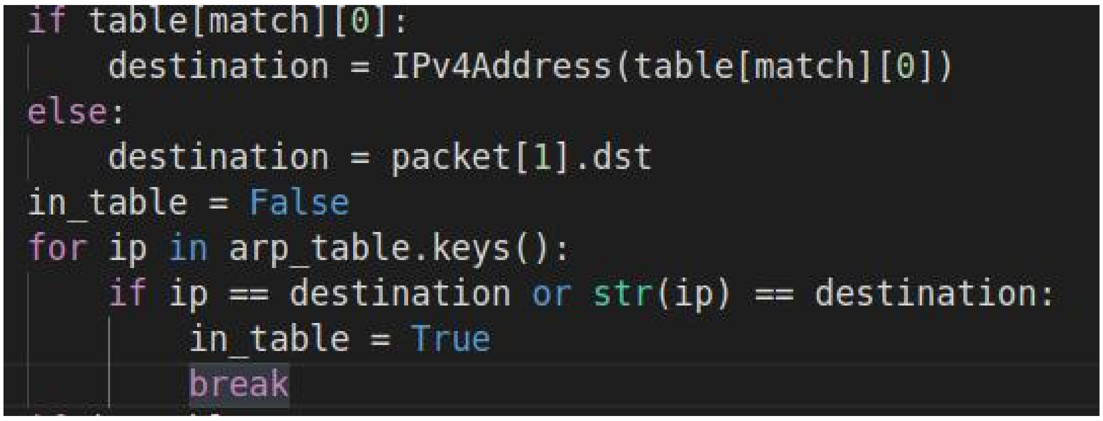
```
                    break
    for intf in all_intf:
        if intf.ipaddr == ipvfour.dst:
            judge = False
            break
```

然后开始最长前缀匹配



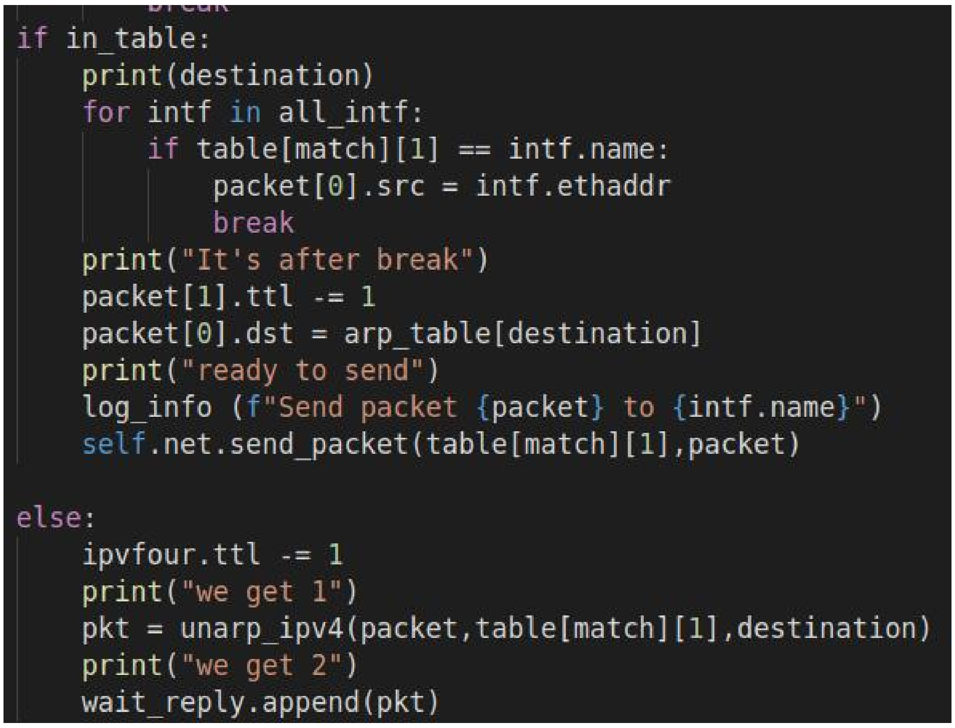
```
if judge:
    print("There is no intf is the ipv4 dstip")
    dst_ip = ipvfour.dst
    match = '0.0.0.0/0'
    aganst = -1
    for key in table.keys():#find the longest prefix
        if dst_ip in key:
            if aganst < key.prefixlen:
                aganst = key.prefixlen
                match = key
```

改变目标 **ip** 同时判断 **arp** 表中有无对应



```
if table[match][0]:
    destination = IPv4Address(table[match][0])
else:
    destination = packet[1].dst
in_table = False
for ip in arp_table.keys():
    if ip == destination or str(ip) == destination:
        in_table = True
        break
```

若在 arp 表里，择改包并发包，若不在，则进缓存



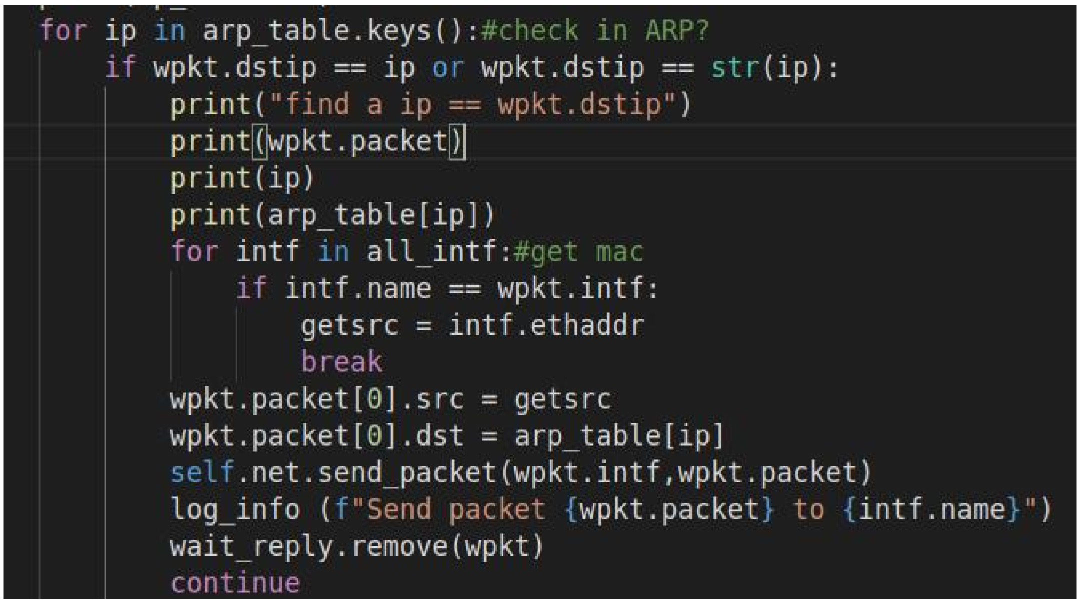
```
if in_table:
    print(destination)
    for intf in all_intf:
        if table[match][1] == intf.name:
            packet[0].src = intf.ethaddr
            break
    print("It's after break")
    packet[1].ttl -= 1
    packet[0].dst = arp_table[destination]
    print("ready to send")
    log_info (f"Send packet {packet} to {intf.name}")
    self.net.send_packet(table[match][1],packet)

else:
    ipvfour.ttl -= 1
    print("we get 1")
    pkt = unarp_ipv4(packet,table[match][1],destination)
    print("we get 2")
    wait_reply.append(pkt)
```

之后进入 **self.check()**对缓存表进行处理

若没有包，则进 **self.check()**进行处理

Self.check()：对每个 **waitingpacket** 处理，建立 **ip** 为 **key** 的字典让所有同 **ip** 的包共享 **count** 与 **time**

1. 判断是否在 **arp** 表里：若有则改包后直接发包

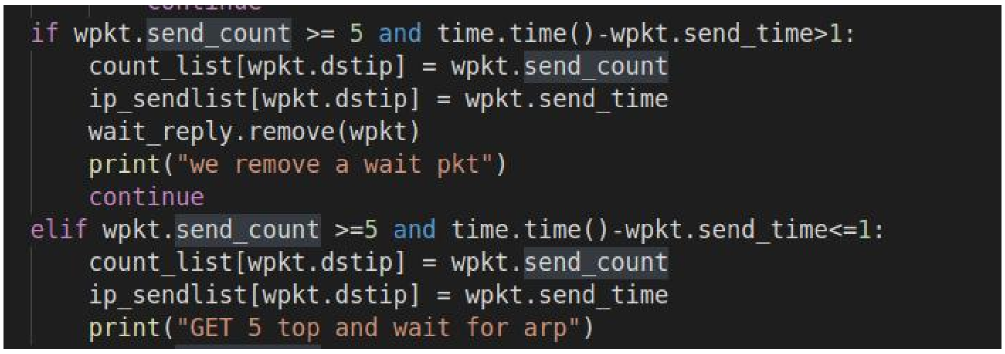


```
for ip in arp_table.keys():#check in ARP?
    if wpkt.dstip == ip or wpkt.dstip == str(ip):
        print("find a ip == wpkt.dstip")
        print(wpkt.packet)
        print(ip)
        print(arp_table[ip])
        for intf in all_intf:#get mac
            if intf.name == wpkt.intf:
                getsrc = intf.ethaddr
                break
        wpkt.packet[0].src = getsrc
        wpkt.packet[0].dst = arp_table[ip]
        self.net.send_packet(wpkt.intf,wpkt.packet)
        log_info (f"Send packet {wpkt.packet} to {intf.name}")
        wait_reply.remove(wpkt)
        continue
```

2. 若不在

（1）count==5 && time>=1：更新 wait 表并删包

count ==5 && time<1：更新 wait 表

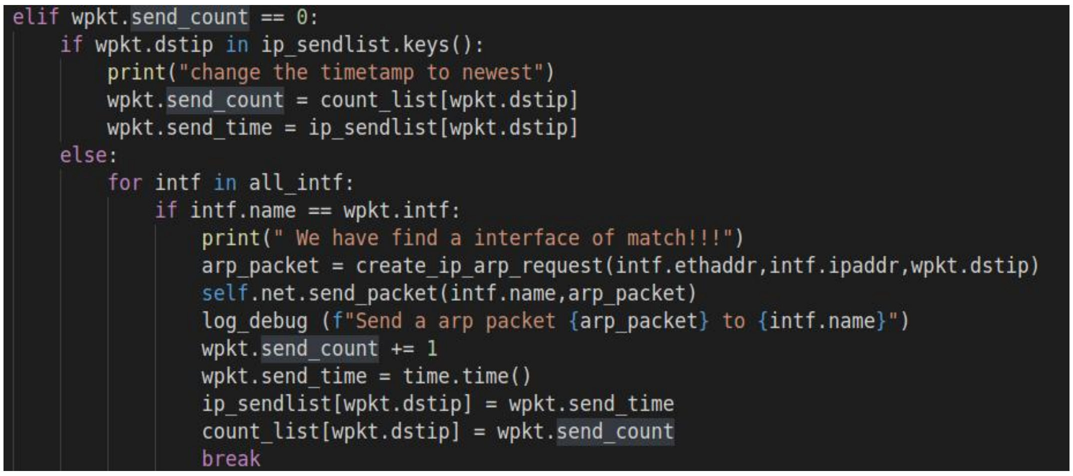


```
if wpkt.send_count >= 5 and time.time()-wpkt.send_time>1:
    count_list[wpkt.dstip] = wpkt.send_count
    ip_sendlist[wpkt.dstip] = wpkt.send_time
    wait_reply.remove(wpkt)
    print("we remove a wait pkt")
    continue
elif wpkt.send_count >=5 and time.time()-wpkt.send_time<=1:
    count_list[wpkt.dstip] = wpkt.send_count
    ip_sendlist[wpkt.dstip] = wpkt.send_time
    print("GET 5 top and wait for arp")
```

（2）count == 0 :

若 wait 表中有对应 ip 则更新包

若 wait 表中无 则发 arp 包并更新包更新 wait 表



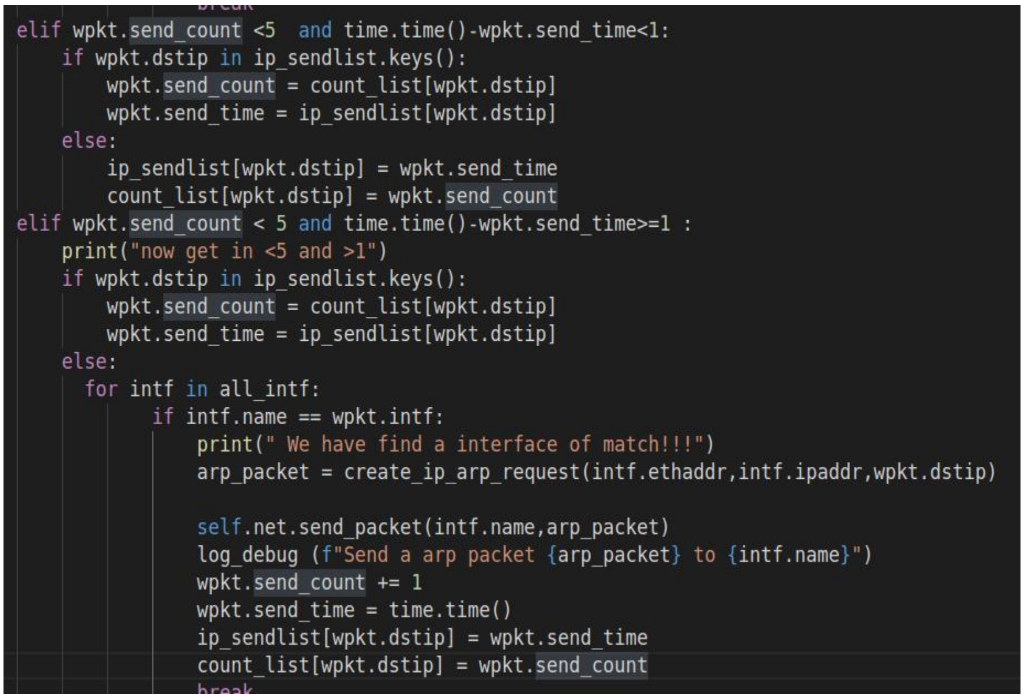
```
elif wpkt.send_count == 0:
    if wpkt.dstip in ip_sendlist.keys():
        print("change the timetamp to newest")
        wpkt.send_count = count_list[wpkt.dstip]
        wpkt.send_time = ip_sendlist[wpkt.dstip]
    else:
        for intf in all_intf:
            if intf.name == wpkt.intf:
                print(" We have find a interface of match!!!")
                arp_packet = create_ip_arp_request(intf.ethaddr,intf.ipaddr,wpkt.dstip)
                self.net.send_packet(intf.name,arp_packet)
                log_debug (f"Send a arp packet {arp_packet} to {intf.name}")
                wpkt.send_count += 1
                wpkt.send_time = time.time()
                ip_sendlist[wpkt.dstip] = wpkt.send_time
                count_list[wpkt.dstip] = wpkt.send_count
                break
```

（3）count <5 && time<1:若 wait 表中有 ip 则更新包

count <5 && time >1 :

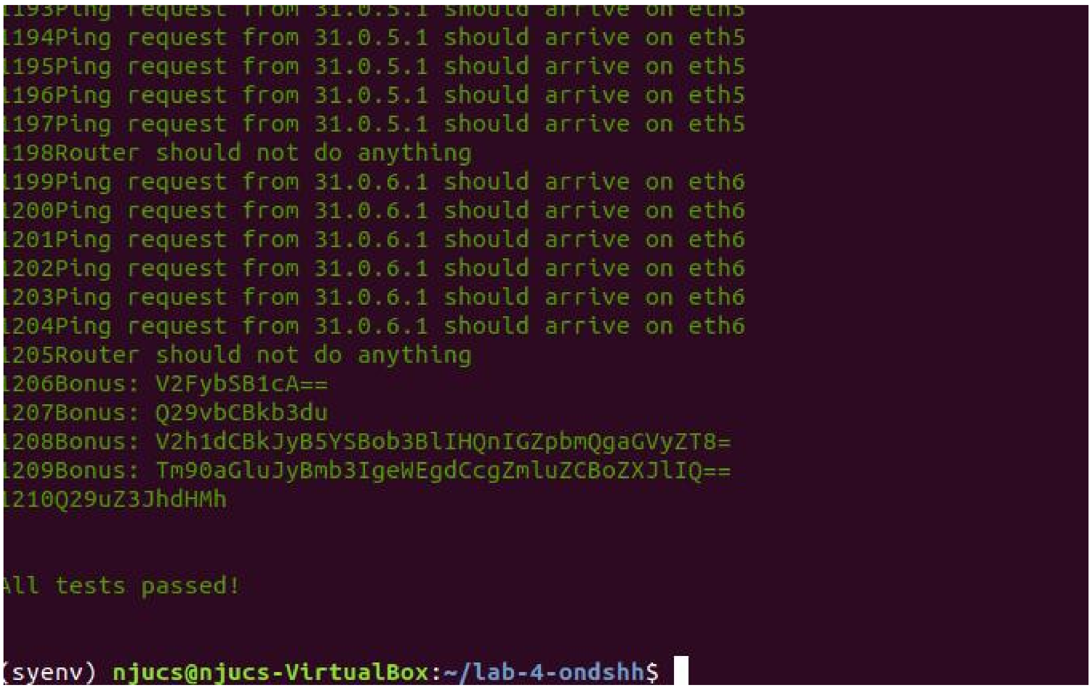
若 wait 表中有 ip 则更新包

若无则发 Arp 包并更新 wait 表



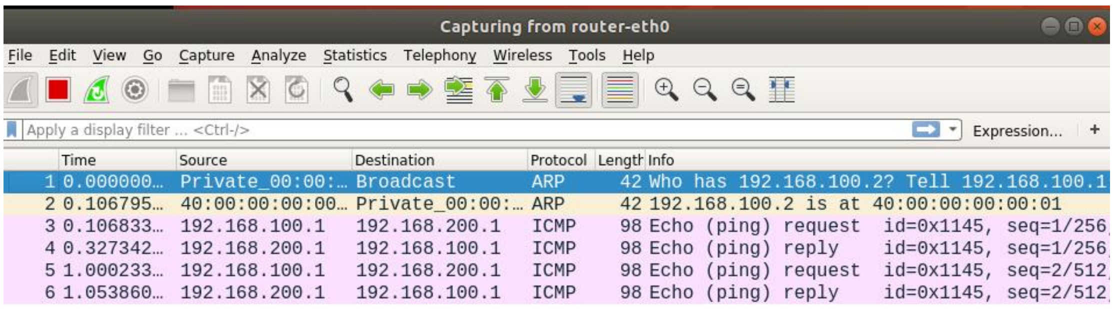
```
                break
elif wpkt.send_count <5  and time.time()-wpkt.send_time<1:
    if wpkt.dstip in ip_sendlist.keys():
        wpkt.send_count = count_list[wpkt.dstip]
        wpkt.send_time = ip_sendlist[wpkt.dstip]
    else:
        ip_sendlist[wpkt.dstip] = wpkt.send_time
        count_list[wpkt.dstip] = wpkt.send_count
elif wpkt.send_count < 5 and time.time()-wpkt.send_time>=1 :
    print("now get in <5 and >1")
    if wpkt.dstip in ip_sendlist.keys():
        wpkt.send_count = count_list[wpkt.dstip]
        wpkt.send_time = ip_sendlist[wpkt.dstip]
    else:
      for intf in all_intf:
            if intf.name == wpkt.intf:
                print(" We have find a interface of match!!!")
                arp_packet = create_ip_arp_request(intf.ethaddr,intf.ipaddr,wpkt.dstip)

                self.net.send_packet(intf.name,arp_packet)
                log_debug (f"Send a arp packet {arp_packet} to {intf.name}")
                wpkt.send_count += 1
                wpkt.send_time = time.time()
                ip_sendlist[wpkt.dstip] = wpkt.send_time
                count_list[wpkt.dstip] = wpkt.send_count
                break
```

show the result of testscenario2_advanced.srpy



```
1193Ping request from 31.0.5.1 should arrive on eth5
1194Ping request from 31.0.5.1 should arrive on eth5
1195Ping request from 31.0.5.1 should arrive on eth5
1196Ping request from 31.0.5.1 should arrive on eth5
1197Ping request from 31.0.5.1 should arrive on eth5
1198Router should not do anything
1199Ping request from 31.0.6.1 should arrive on eth6
1200Ping request from 31.0.6.1 should arrive on eth6
1201Ping request from 31.0.6.1 should arrive on eth6
1202Ping request from 31.0.6.1 should arrive on eth6
1203Ping request from 31.0.6.1 should arrive on eth6
1204Ping request from 31.0.6.1 should arrive on eth6
1205Router should not do anything
1206Bonus: V2FybSB1cA==
1207Bonus: Q29vbCBkb3du
1208Bonus: V2h1dCBkJyB5YSBob3BlIHQnIGZpbmQgaGVyZT8=
1209Bonus: Tm90aGluJyBmb3IgeWEgdCcgZmluZCBoZXJlIQ==
1210Q29uZ3JhdHMh


All tests passed!


(syenv) njucs@njucs-VirtualBox:~/lab-4-ondshh$
```

Server1 ping Server2



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

```
    Time       Source            Destination       Protocol Length Info
  1 0.000000…  Private_00:00:…   Broadcast         ARP      42 Who has 192.168.100.2? Tell 192.168.100.1
  2 0.106795…  40:00:00:00:00…   Private_00:00:…   ARP      42 192.168.100.2 is at 40:00:00:00:00:01
  3 0.106833…  192.168.100.1     192.168.200.1     ICMP     98 Echo (ping) request  id=0x1145, seq=1/256
  4 0.327342…  192.168.200.1     192.168.100.1     ICMP     98 Echo (ping) reply    id=0x1145, seq=1/256
  5 1.000233…  192.168.100.1     192.168.200.1     ICMP     98 Echo (ping) request  id=0x1145, seq=2/512
  6 1.053860…  192.168.200.1     192.168.100.1     ICMP     98 Echo (ping) reply    id=0x1145, seq=2/512
```

eth0 为 server1 与 router 的接口，因此先收到 arp 包，router 查询后进行回复，之后把 icmp 包转发给 server2，server2->router->server1

4. 总结与感想

加深学习了路由表的匹配与路由包的转发