

# 计算机网络实验 LAB4

task 1: 准备工作，将文件复制到当前的 lab4 文件夹

task 2: 建立 forwarding\_table 和进行最长匹配

1>建立 forwarding\_table，我按照要求，在 myrouter 初始化时就将 forwarding\_table.txt 的内容读入到一个列表中，并调用 interfaces 函数，将相应的 router 端口的 ip 地址和掩码都加入表中，对于 router 的 ip 我还无法确定其 next\_hop 的 ip 地址，所以暂时将其置位 None

2>最长前缀匹配，我调用了 switchyard.lib.address 中的函数，在每次匹配时若成功，则计算前缀长度，最终挑选最长的，返回其序号

3>函数返回后，若值为 None，证明没有匹配，或为 router\_ip 中的值，或指向其它 ip，对每种情况进行相应的处理即可（在 task3 中进行）

4>若在 netaddr 没有得到符合条件的匹配，证明 router 端口相连的子网中没有目的 ip 的匹配，这时就需要对相连的 next\_hop 的 IP 进行匹配，如果仍没有匹配，再返回 None

task3:

首先阐述我的 router 算法逻辑：

数据结构：

arp\_queue:{}字典，记录等待 arp\_request 的包的信息

cache\_table:{}字典，记录 router 收到的 arp\_request 的 ip-mac 对

forward\_table:[]记录通过该 router 可以到达的网络及其通过的端口

每次循环

1>检查 arp\_queue 每个表项，若时间大于 1 秒，且重发超过五次，丢弃  
若未超过五次，则重发 arp\_request，进入 2>

2>收到包，分为三种情况，Arp 包，到 3>，IPv4 包，到 4>，其它包，丢弃之

3>Arp 包：

若是 Arp 请求，则遍历 router 的端口，若为其端口的 ip，则返回一个相应端口的 mac 给收到信息的端口

若是 Arp 回复，则遍历 arp\_queue，若相应 ip 在其中 key 值，证明得到了该 ip 对应的 mac，组装包，并发送到收到回复的端口中，更新 arp\_queue

4>IPv4 包：

首先进行最大匹配，若匹配到，则查看目标 ip 是否在 cache\_table 中，若在，则组装包直接发送之。若不在，则将包的信息记录在 arp\_queue 中，并发送 arp 请求到相应端口。（收到 Arp 回复时才能组装发送该包）

按照上述思路，我通过了相应的 test 文件，截图如下：

Results for test scenario IP forwarding and ARP requester tests: 31 passed, 0 fail

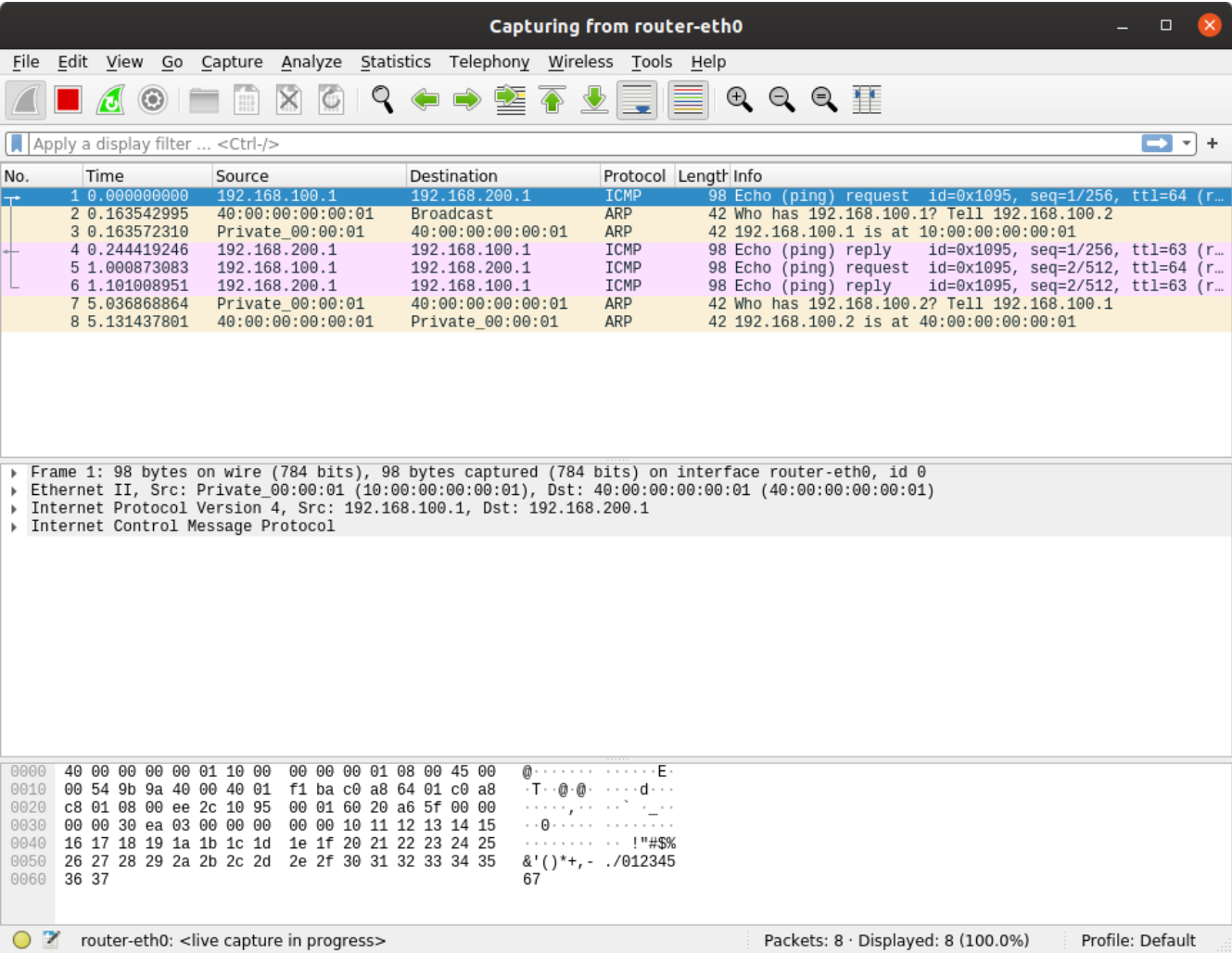
Passed:

- 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
- 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!

部署到 mininet 中的验证与分析：

我在 server1 中执行 ping -c2 192.168.200.1(server2)，wireshark 在 router-eth0 中抓包，因为 eth0 是直接和 server1 相连的包（参考讲解中的例子）



简要分析一下包的发送过程：

首先 server1 向 router 发送 arp 包，得到 router-eth0 的 mac 地址后，向其发送 ICMP 包，router 识别其是 IPv4 包，经过最长匹配，得到 forwarding\_table 的一个表项[‘192.268.200.1’, ...,None, router-eth1]

因为 next\_hop 段为 None，router 必须将包的目的 ip192.168.200.2 作为下一个要发送的站点，但它不在 cache\_table 中，所以必须发送 arp 请求，得到其 mac 地址后，router 将整个包打包发送至 server2，此时第一个

ICMP 请求才得以发送。第二次 ICMP 请求过程类似，只是 router 的 cache\_table 有了 server2 的 ip-mac 对，故少了一次 arp 请求。以上过程根据 wireshark 的抓包结果和 router 中的 log\_info 信息得到。

本次实验到这里就完成了，我学到了 router 的基本工作原理，并能将其与 switch 进行对比，受益匪浅。