

lab6_report_router

姓名	学号	邮箱	完成时间
陈嘉昀	211220137	jjayunchen@smail.nju.edu.cn	2023年12月28日

1 代码设计

1-1 arp

- arp_send_request: 从当前端口发送一个查询dst_ip的mac地址的arp包
- arp_send_reply: 回复需要查询的mac
- handle_arp_packet: 根据收到的arp包的类型, arp_reply或者arp_insert(在arpcache中介绍)
- iface_send_by_arp:(Tricky!!)
 - 查询cache中是否有dst_ip的mac:
 - 若有, 修改包中的dst_mac, 通过iface_send_packet发送
 - 若无, 缓存此包, 并发送查询arp的包。

1-2 arpcache

- arp_lookup: 查询某个dst_ip对应的mac
- arp_append_packet:
 - 缓存某个暂时不知道dst_mac设置为什么的包, 并登记:
 - 若已有相同的查询登记, 则直接缓存此包在对应登记条目上
 - 否则, 创建新的登记条目, 缓存此包, 并发送查询arp的包
- arp_insert:
 - 收到一个ip对应的mac, 加入cache
 - 检查是否有此ip的查询登记, 若有, 则继续登记的包的发送
- arp_sweep: 清除过期/未应答的等级条目

1-3 icmp

- icmp_send_packet: 设置好icmp包, 通过ip_send_packet发送

1-4 ip_base

- ip_send_packet: 在此实验只发送icmp, 使用ip和ether协议包装好icmp包
- longest_prefix: 查询路由表, 对dst_ip进行匹配

1-5 ip

- handle_ip_packet:
 - 如果是发给router, 则回复icmp
 - 如果不是:
 - ttl -= 1 后, 判断
 - 若ttl == 0, 发送icmp
 - 否则, 重新计算checksum, 并转发

2 截图

拓扑如下：

```
      h3
      |
h1 -- r1 -- h2
```

运行结果如下： ping：

```
(base) ubuntu@VM-4-8-ubuntu:~/2023-fall/CN/lab6-2023autumn-Chen-Jiayun/02-router$ sudo ./router_topo.py
mininet> h1 ping h2
PING 10.0.2.22 (10.0.2.22) 56(84) bytes of data.
^C
--- 10.0.2.22 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1007ms

mininet> r1 ./router &
DEBUG: find the following interfaces: r1-eth0 r1-eth1 r1-eth2.
mininet> h1 ping h2
PING 10.0.2.22 (10.0.2.22) 56(84) bytes of data.
64 bytes from 10.0.2.22: icmp_seq=1 ttl=63 time=0.191 ms
64 bytes from 10.0.2.22: icmp_seq=2 ttl=63 time=0.154 ms
64 bytes from 10.0.2.22: icmp_seq=3 ttl=63 time=0.150 ms
^C
--- 10.0.2.22 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2039ms
rtt min/avg/max/mdev = 0.150/0.165/0.191/0.018 ms
```

traceroute:

```
mininet> h1 traceroute h2
traceroute to 10.0.2.22 (10.0.2.22), 30 hops max, 60 byte packets
 1  10.0.1.1 (10.0.1.1)  0.085 ms  0.054 ms  0.051 ms
 2  * 10.0.2.22 (10.0.2.22)  0.158 ms  0.164 ms
```

新的拓扑：

```
h1 -- r1 -- r2 -- h2
```

网络路由配置如下：

```
# mul_router_topo.py
h1.cmd('ifconfig h1-eth0 10.0.1.11/24')
h2.cmd('ifconfig h2-eth0 10.0.4.33/24')

h1.cmd('route add default gw 10.0.1.1')
h2.cmd('route add default gw 10.0.4.1')

r1.cmd('ifconfig r1-eth0 10.0.1.1/24')
r1.cmd('ifconfig r1-eth1 10.0.2.1/24')
r1.cmd('ip route add default via 10.0.2.2 dev r1-eth1')
r1.cmd('ip route add 10.0.1.0/24 dev r1-eth0')

r2.cmd('ifconfig r2-eth0 10.0.2.2/24')
r2.cmd('ifconfig r2-eth1 10.0.4.1/24')
r2.cmd('ip route add default via 10.0.2.1 dev r2-eth0')
r2.cmd('ip route add 10.0.4.0/24 dev r2-eth1')
```

traceroute:

```
mininet> h1 traceroute h2
traceroute to 10.0.4.33 (10.0.4.33), 30 hops max, 60 byte packets
 1  10.0.1.1 (10.0.1.1)  0.340 ms  0.296 ms  0.289 ms
 2  10.0.2.2 (10.0.2.2)  0.397 ms  0.388 ms  0.378 ms
 3  10.0.4.33 (10.0.4.33)  0.455 ms  0.451 ms  0.445 ms
mininet>
```