

Lab 2: Router

孙梓健 2017010428 软81班

1 实验目的

- 掌握路由器的工作原理
- 了解以太帧、ARP数据包、IPv4数据报和ICMP报文的结构
- 熟悉各种数据包的操作方法

2 实验环境

- 操作系统：Ubuntu 16.04.7
- 编译平台：Visual Studio Code & g++ 5.4.0

3 项目流程

3.1 数据包处理

对于每种数据包，先从大小、类型、校验和等属性判断其合法性，再按照其类型的不同分别进行处理。

1. 以太帧：分为IPv4包和ARP包
2. ARP包：分为request和reply
 - request：如果目的IP是本路由的IP，发送reply，否则忽略
 - reply：查找arp cache中是否目的地址有对应的IP/MAC对，如果有，对相应的包执行handlePacket，否则将IP/MAC对加入arp cache
3. IPv4：分为指向本路由的包和指向其他路由的包
 - 指向本路由：分为ICMP包和TCP&UDP包，如果是后者，返回响应消息Port Unreachable
 - 指向其他路由：如果TTL为1，判定为超时，返回响应消息Time Exceeded，否则进行转发
4. ICMP：查找arp cache中是否目的地址有对应的IP/MAC对，如果有，返回reply，否则将request加入arp cache

3.2 ARP Cache

1. 记录发送次数为5的request，对于这些request，返回响应消息Host Unreachable，对于其他request，发送arp request
2. 删除发送超过5次的request
3. 删除invalid的cache entry

4 实验结果

4.1 ping

```
mininet> client ping server1
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=63 time=1727 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=63 time=759 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=63 time=35.1 ms
64 bytes from 192.168.2.2: icmp_seq=4 ttl=63 time=10.6 ms
64 bytes from 192.168.2.2: icmp_seq=5 ttl=63 time=37.0 ms
64 bytes from 192.168.2.2: icmp_seq=6 ttl=63 time=9.46 ms
64 bytes from 192.168.2.2: icmp_seq=7 ttl=63 time=30.1 ms
64 bytes from 192.168.2.2: icmp_seq=8 ttl=63 time=58.7 ms
64 bytes from 192.168.2.2: icmp_seq=9 ttl=63 time=32.4 ms
64 bytes from 192.168.2.2: icmp_seq=10 ttl=63 time=23.1 ms
^C
--- 192.168.2.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9024ms
rtt min/avg/max/mdev = 9.468/272.454/1727.923/531.929 ms, pipe 2
```

4.2 traceroute

```
mininet> client traceroute server1
traceroute to 192.168.2.2 (192.168.2.2), 30 hops max, 60 byte packets
 1  10.0.1.1 (10.0.1.1)  1019.272 ms  1020.204 ms  1020.216 ms
 2  192.168.2.2 (192.168.2.2)  1071.343 ms  1071.398 ms  1071.396 ms
```

4.3 wget

```
mininet> client wget http://192.168.2.2/111
--2021-12-07 20:28:04-- http://192.168.2.2/111
正在连接 192.168.2.2:80... 已连接。
已发出 HTTP 请求，正在等待回应... 404 File not found
2021-12-07 20:28:05 错误 404: File not found.
```

ps: 我也不知道有什么文件

4.4 autograde

```
Your tests score is 45/45 pts
THIS IS NOT YOUR FINAL SCORE !!!
```

5 问题总结

1. 在periodicCheckArpRequestsAndCacheEntries函数中使用了std::mutex，导致经常出现死锁现象。了解到std::recursive_mutex允许同一个线程对互斥量对象获取多级所有权，将arp_cache.cpp文件中的所有std::mutex改成std::recursive_mutex之后得到解决。
2. 对代码框架阅读不熟悉，在获取以太网帧类型的时候，调用了ethertype之后对type进行了二次顺序码转换，导致ARP和IPv4的类型判断错误。
3. 检查ICMP报文校验和的时候，没有正确理解cksum函数，以为校验和应该是针对整个包。实际上应该减去以太网帧和IP包头大小。
4. 查找arp entry有时会错误地没有找到对象，导致ping卡死，原因是在某些地方调用queueRequest函数时，将源IP和目的IP混淆，修改正确即可。

6 实验感想

通过本次实验，我增加了对路由器工作原理的了解，对以太帧、ARP、IPv4、ICMP等数据包和协议的理解更加深入，在了解这些协议的基础上，更加熟悉了网络的五层结构和数据的传输流程。