

计算机网络实验2-3 设计文档

实验二 抓包程序

设计实现

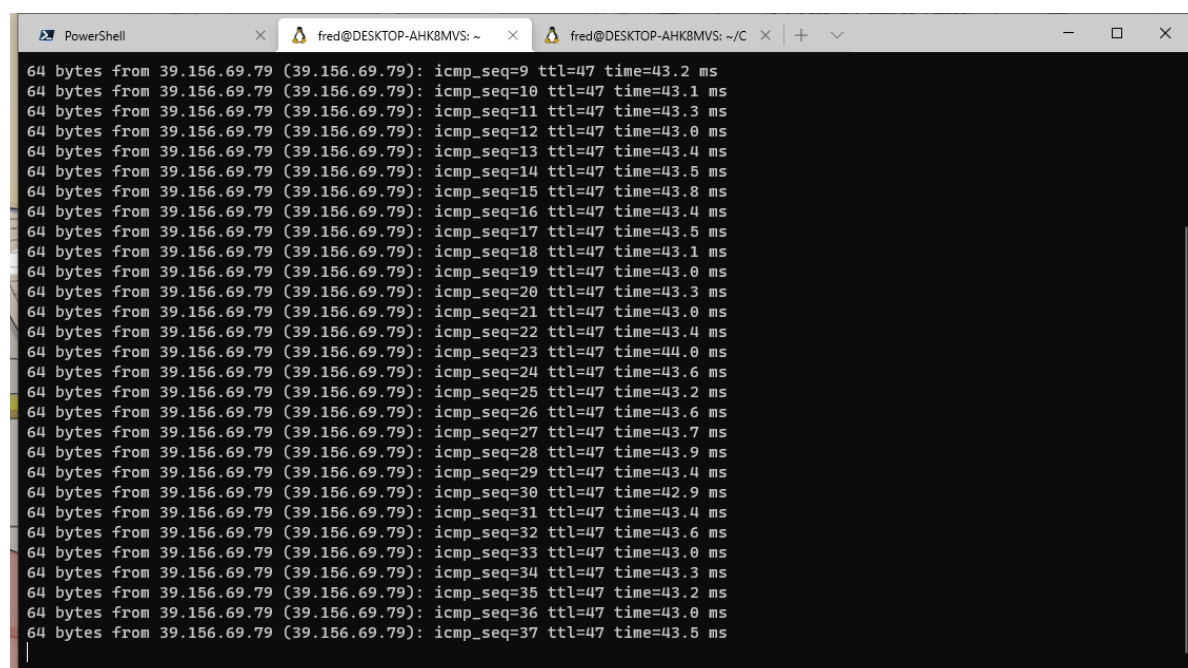
通过类型字段判断是ARP或IP数据包。

IP数据包->判断是ICMP、TCP或UDP包

ARP数据包->判断是request或reply

然后按照Wireshark的格式输出

结果截图



```
PowerShell
fred@DESKTOP-AHK8MVS: ~
fred@DESKTOP-AHK8MVS: ~/C

64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=9 ttl=47 time=43.2 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=10 ttl=47 time=43.1 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=11 ttl=47 time=43.3 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=12 ttl=47 time=43.0 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=13 ttl=47 time=43.4 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=14 ttl=47 time=43.5 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=15 ttl=47 time=43.8 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=16 ttl=47 time=43.4 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=17 ttl=47 time=43.5 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=18 ttl=47 time=43.1 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=19 ttl=47 time=43.0 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=20 ttl=47 time=43.3 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=21 ttl=47 time=43.0 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=22 ttl=47 time=43.4 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=23 ttl=47 time=44.0 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=24 ttl=47 time=43.6 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=25 ttl=47 time=43.2 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=26 ttl=47 time=43.6 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=27 ttl=47 time=43.7 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=28 ttl=47 time=43.9 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=29 ttl=47 time=43.4 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=30 ttl=47 time=42.9 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=31 ttl=47 time=43.4 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=32 ttl=47 time=43.6 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=33 ttl=47 time=43.0 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=34 ttl=47 time=43.3 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=35 ttl=47 time=43.2 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=36 ttl=47 time=43.0 ms
64 bytes from 39.156.69.79 (39.156.69.79): icmp_seq=37 ttl=47 time=43.5 ms
```

ping

```
Protocol: ICMP
Source Address: 192.168.78.137
Destination Address: 183.232.231.172
Type: 8, Code: 0
id = 0x0ea5, seq = 1/256, ttl = 64
-----
Protocol: ICMP
Source Address: 183.232.231.172
Destination Address: 192.168.78.137
Type: 0, Code: 0
id = 0x0ea5, seq = 1/256, ttl = 128
```

结果

实验三 自己动手写协议栈之以太网收发实现

设计实现

ethernet_out

添加以太网包头

添加目的地址、原地址、协议类型

调用driver_send()

ethernet_in

通过类型字段判断IP协议或ARP协议

去掉以太网包头

IP协议->调用ip_in() ARP协议->调用arp_in()

结果截图

```
$ make test_eth_in
gcc eth_in_test.c ../src/ethernet.c faker/arp.c faker/ip.c faker/driver.c global.c ../src/utils.c -o eth_in_test -lpcap -I../include/
./eth_in_test
Test start
Feeding input 17
Sample input all processed, checking output
Checking log file(compare with demo).
Round 1: no differences
Round 2: no differences
Round 3: no differences
Round 4: no differences
Round 5: no differences
Round 6: no differences
Round 7: no differences
Round 8: no differences
Round 9: no differences
Round 10: no differences
Round 11: no differences
Round 12: no differences
Round 13: no differences
Round 14: no differences
Round 15: no differences
Round 16: no differences
Round 17: no differences
====> All log rounds are the same to the demo.
```

```
$ make test_eth_out
gcc eth_out_test.c ../src/ethernet.c faker/arp.c faker/ip.c faker/driver.c global.c ../src/utls.c -o eth_out_test -lpcap -I../include/
./eth_out_test
Test start
Feeding input 17
Sample input all processed, checking output
Checking log file(compare with demo).
Round 1: no differences
Round 2: no differences
Round 3: no differences
Round 4: no differences
Round 5: no differences
Round 6: no differences
Round 7: no differences
Round 8: no differences
Round 9: no differences
Round 10: no differences
Round 11: no differences
Round 12: no differences
Round 13: no differences
Round 14: no differences
Round 15: no differences
Round 16: no differences
Round 17: no differences
====> All log rounds are the same to the demo.
Checking pcap output file(compare with demo).
Packet 1: no differences
Packet 2: no differences
Packet 3: no differences
Packet 4: no differences
Packet 5: no differences
Packet 6: no differences
Packet 7: no differences
Packet 8: no differences
Packet 9: no differences
Packet 10: no differences
Packet 11: no differences
Packet 12: no differences
Packet 13: no differences
Packet 14: no differences
Packet 15: no differences
Packet 16: no differences
Packet 17: no differences
====> All packets are the same to the demo.
```