

廈門大學



信息学院软件工程系

《计算机网络》实验报告

题 目 实验三 捕获并分析帧和 IP 报文

班 级 软件工程 2018 级 1 班

姓 名 林坚

学 号 24320182203232

实验时间 2020 年 3 月 12 日

2020 年 03 月 12 日

1 实验目的

捕获并分析以太网的帧，获取目标与源网卡的 MAC 地址，获取远端 MAC 地址

2 实验环境

Eclipse VS WinSock 的 GetAddress 命令 WinPCAP

3 实验结果

1 用 IPCONFIG.EXE 显示计算机中网络适配器的 IP 地址、子网掩码及默认

```
PS C:\Windows\System32> ipconfig /all

Windows IP 配置

   主机名 . . . . . : DESKTOP-SEJ1B3J
   主 DNS 后缀 . . . . . :
   节点类型 . . . . . : 混合
   IP 路由已启用 . . . . . : 否
   WINS 代理已启用 . . . . . : 否

无线局域网适配器 本地连接* 1:

   媒体状态 . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . :
   描述. . . . . : Microsoft Wi-Fi Direct Virtual Adapter
   物理地址. . . . . : 84-FD-D1-75-1C-DD
   DHCP 已启用 . . . . . : 是
   自动配置已启用. . . . . : 是

无线局域网适配器 本地连接* 2:

   媒体状态 . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . :
   描述. . . . . : Microsoft Wi-Fi Direct Virtual Adapter #2
   物理地址. . . . . : 86-FD-D1-75-1C-DC
   DHCP 已启用 . . . . . : 是
   自动配置已启用. . . . . : 是

无线局域网适配器 WLAN:

   连接特定的 DNS 后缀 . . . . . :
   描述. . . . . : Intel(R) Wireless-AC 9560 160MHz
   物理地址. . . . . : 84-FD-D1-75-1C-DC
   DHCP 已启用 . . . . . : 是
   自动配置已启用. . . . . : 是
   本地链接 IPv6 地址. . . . . : fe80::508c:b73d:562:9062%14(首选)
   IPv4 地址 . . . . . : 192.168.1.7(首选)
   子网掩码 . . . . . : 255.255.255.0
   获得租约的时间 . . . . . : 2020年3月14日 8:54:32
   租约过期的时间 . . . . . : 2020年3月15日 8:54:31
   默认网关 . . . . . : 192.168.1.1
   DHCP 服务器 . . . . . : 192.168.1.1
   DHCPv6 IAID . . . . . : 126156241
   DHCPv6 客户端 DUID . . . . . : 00-01-00-01-24-21-48-44-84-FD-D1-75-1C-DC
   DNS 服务器 . . . . . : 192.168.1.1
   TCP/IP 上的 NetBIOS . . . . . : 已启用

网关
```

2 获得本机 MAC 地址

```

package mynet;

import java.net.InetAddress;

public class Test {
    @SuppressWarnings("static-access")
    public static void main(String[] args) throws Exception {
        InetAddress ia=null;
        try {
            ia=ia.getLocalHost();
            String localname=ia.getHostName();
            String localip=ia.getHostAddress();
            System.out.println("本机名: "+ localname);
            System.out.println("本机ip: "+localip);
        } catch (Exception e) {
            e.printStackTrace();
        }
        InetAddress ia1 = InetAddress.getLocalHost();//获取本地IP对象
        System.out.println("MAC ."+getMACAddress(ia1));
    }
    //获取MAC地址的方法
    private static String getMACAddress(InetAddress ia)throws Exception{
        byte[] mac = NetworkInterface.getByInetAddress(ia).getHardwareAddress();//获得网络接口对象（即网卡），并得到mac地址，mac地址存在于一个byte数组
        StringBuffer sb = new StringBuffer(); //下面代码是把mac地址拼装成String
        for(int i=0;i<mac.length;i++){
            if(i!=0){
                sb.append("-");
            }
            String s = Integer.toHexString(mac[i] & 0xFF); //mac[i] & 0xFF 是为了把byte转化为正整数
            System.err.println(s);
            sb.append(s.length()==1?0+s:s);
        }
        return sb.toString().toUpperCase(); //把字符串所有小写字母改为大写成为正规的mac地址并返回
    }
}

```

terminated> test java Application D:\mycompse\p2\src>

本机名: DESKTOP-SEJ1B3J

本机ip: 192.168.1.7

84

fd

d1

75

MAC .84-FD-D1-75-1C-DC

1c

dc

3 获取远端 MAC 地址

```

1. \Device\NPF_{9390F67C-DC76-4EA9-B7BB-4CD5661D23DC} (Microsoft)
2. \Device\NPF_{E82036A1-F388-47CF-9880-1D8354042C31} (Microsoft)
3. \Device\NPF_{964524A1-6224-4C1B-9C84-B6A400F40A75} (Microsoft)
Enter the interface number (1-3):3

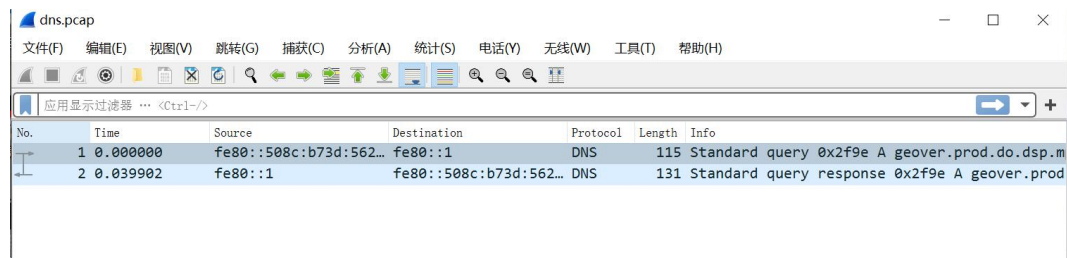
*listening on Microsoft...
*12:01:08.520898 len:129 84 FD D1 75 1C DC 74 B9 EB A5 F8 79 08 00 45 00
*00 73 10 9B 40 00 34 11 FD 2E DF A6 97 5A C0 A8
*01 07 1F 40 0F A6
*mac_header:
*   dest_addr: 84 FD D1 75 1C DC
*   src_addr: 74 B9 EB A5 F8 79
*   type: 0800
*ip_header
*   ver_ihl : 45
*   tos : 00
*   tlen : 0073
*   identification: 109B
*   flags_fo : 4000
*   ttl : 34
*   proto : 11
*   crc : FD2E
*   op_pad : 00001F40
*   saddr: : DF A6 97 5A DF A6 97 5A
*   daddr: : C0 A8 01 07 C0 A8 01 07
*12:01:10.670165 len:48 FF FF FF FF FF FF F0 6D 78 4E 67 0D 08 00 45 00
*00 22 7B 59 40 00 40 11 3B 1F C0 A8 01 03 C0 A8
*01 FF B6 A2 EA 60
//开始捕获
void packet_handler(u_char *param, const struct pcap_pkthdr *header, const u_char *pkt_data)
{
    mac_header *mh;
    ip_header *ih;

    int length = sizeof(mac_header) + sizeof(ip_header);
    for (int i = 0; i < length; i++) {
        printf("%02X ", pkt_data[i]);
        if ((i & 0xF) == 0xF)
            printf("\n");
    }
    printf("\n");

    mh = (mac_header*)pkt_data;
    printf("mac_header:\n");
    printf("\tdest_addr: ");
    for (int i = 0; i < 6; i++) {
        printf("%02X ", mh->dest_addr[i]);
    }
}

```

测试:



The screenshot shows the Wireshark interface with a packet capture of DNS traffic. The packet list pane shows two packets: a standard query and a standard query response. The packet details pane shows the structure of the DNS message, including the query ID, flags, and the question section.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::508c:b73d:562...	fe80::1	DNS	115	Standard query 0x2f9e A geover.prod.do.dsp.m
2	0.039902	fe80::1	fe80::508c:b73d:562...	DNS	131	Standard query response 0x2f9e A geover.prod

4. 实验总结

在使用wireshark测试的时候出现了运行结果和导出的dns.pcap里的两条报文不一样的情况，不知道是什么原因

大体在视频和ppt的教学下学会了如何获取本机ip和mac地址以及远端mac地址。