实验报告

实验（二）

姓　　名	熊恪峥
学　　号	22920202204622
日　　期	2022年10月18日
学　　院	信息学院
课程名称	计算机网络

实验（二）

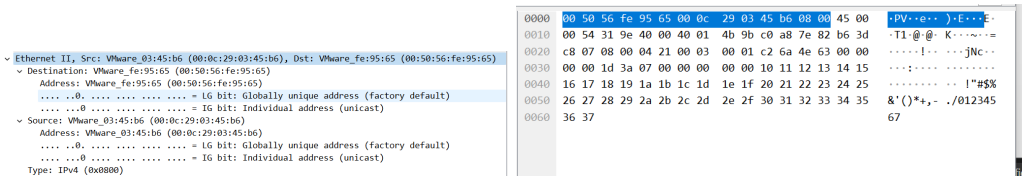
目录

1 捕获和分析有线以太网数据包

1.1 分析MAC帧

首先在Wireshark中查看帧内容如图 1

图 1: MAC帧内容



(a) 内容 (b) 原始数据

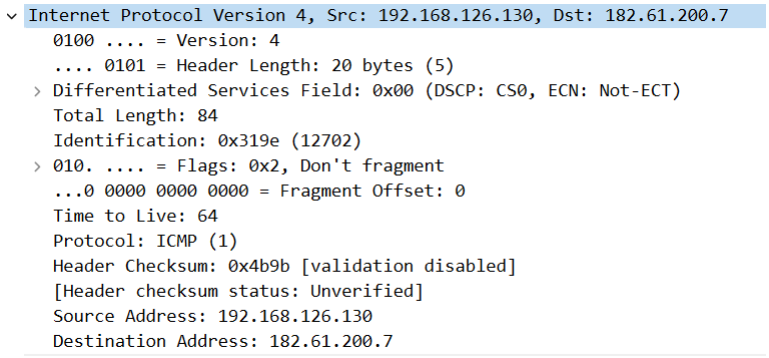
根据MAC帧格式，前12字节分别是目的地址和源地址。因此目的地址是 $00:50:56:fe:95:95$，源地址是$00:0c:29:03:45:b6$。类型字段的值是$0x0800$，表示下一层是IP协议。

EUI-48地址，也就是MAC地址，是一个48位的二进制数。它的前24位是厂商的OUI。其中第一个字节的后两位分别表示了是否是全局唯一的地址和是否是多播地址。例如目的地址的前两位是00，表示是全局唯一的地址，进行单播；源地址也是如此。

1.2 分析IP数据报首部

IP数据报的内容如图 2。

图 2: IP数据报



前4位为版本号，说明这个报文是IPv4；其后4位是首部长度，本报首部长度为20字节；然后是区分服务（Differentiated services）DS，占1字节。 IP报总长度，占4位，本数据报总长度为84字节。然后是标识，占16位；之后是标志，占3位，有禁止分段的标志。然后是偏移，占13位，本请求帧没有分片，所以片偏移量为0；生存时间，占8位，表明数据报在互联网中至多可经过多少个路由器。本数据报TTL为64；然后是协议字段，占8位，表明此数据报携带的数据是使用何种协议，本数据报使用ICMP 之后是首部检验和，占16位。最后源地址和目的地址各占32位。

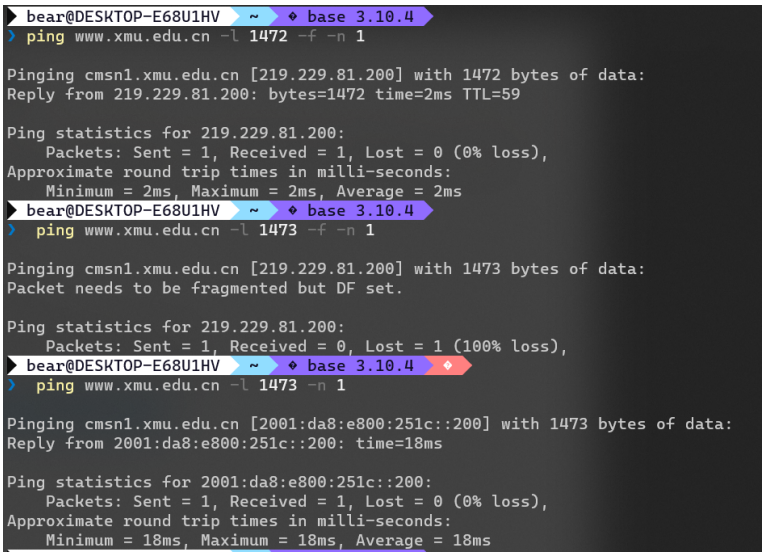
帧长比数据报长要长，因为在不同的层会加入额外的首部。

1.3 观察IP分片

根据实验内容，(a)是普通的ping请求。(b)、(c)、(d)用不同长度的数据包进行ping请求。其中(b)、(c)发送了禁止分段标记，(c)、(d)形成的数据长度超过了MTU，因此ping的效果如图 3所示。

可见，(c)无法成功，因为应当分片，但由于禁止分片标记而不能分段，因此无法成功。

图 3: Ping结果



观察数据包的长度，如图 4a，发现当包长度超过1472时ICMP请求变短了。改变过滤条件，发现出现了IP分片数据报，如图 4b所示。因此，当数据变长时会产生分片。这里测试了1473、1479两种长度，可以发现分片的所有数据报长度之和是正常的。

图 4: 分片数据报

Protocol	Length	Info
ICMP	1514	Echo (ping) request
ICMP	1514	Echo (ping) reply
ICMP	41	Echo (ping) request
ICMP	41	Echo (ping) reply
ICMP	35	Echo (ping) request
ICMP	35	Echo (ping) reply

(a) 长度

Protocol	Length	Info
ICMP	1514	Echo (ping) request
ICMP	1514	Echo (ping) reply
IPv4	1514	Fragmented IP protocol
ICMP	41	Echo (ping) request
IPv4	1514	Fragmented IP protocol
ICMP	41	Echo (ping) reply
IPv4	1514	Fragmented IP protocol
ICMP	35	Echo (ping) request
IPv4	1514	Fragmented IP protocol
ICMP	35	Echo (ping) reply
ICMP	1514	Echo (ping) request
ICMP	1514	Echo (ping) reply

(b) 分片数据报

1.4 ICMP协议分析

在Windows下，一次Ping命令会进行4次ICMP请求，因此有8个数据报，其中四次请求四次回应。如图5所示。

数据包内容如图 6，ICMP请求和回应的Code分别为8-0和0-0。 IP部分首先两者的源地址和目的地址相反，其次TTL也不一样，可能是由于操作系统不同。

1.5 tracert工作原理分析

tracert程序的设计利用ICMP及TTL来列出所有经过的节点。

首先，tracert送出一个TTL是1的数据报到目的地，当路径上的第一个路由器收到时，它将TTL减1。此

图 5: Ping产生的数据报

Destination	Protocol	Length	Info
223.252.214.105	ICMP	74	Echo (ping) request
192.168.1.102	ICMP	74	Echo (ping) reply
223.252.214.105	ICMP	74	Echo (ping) request
192.168.1.102	ICMP	74	Echo (ping) reply
223.252.214.105	ICMP	74	Echo (ping) request
192.168.1.102	ICMP	74	Echo (ping) reply
223.252.214.105	ICMP	74	Echo (ping) request
192.168.1.102	ICMP	74	Echo (ping) reply

图 6: ICMP请求

```
Internet Protocol Version 4, Src: 223.252.214.105, Dst: 192.168.1.102
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 60
    Identification: 0xa964 (43364)
  > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 55
    Protocol: ICMP (1)
    Header Checksum: 0x61e8 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 223.252.214.105
    Destination Address: 192.168.1.102
Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x5325 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 566 (0x0236)
    Sequence Number (LE): 13826 (0x3602)
    [Request frame: 117]
    [Response time: 18.344 ms]
  > Data (32 bytes)
      Data: 6162636465666768696a6b6c6d6e6f7071727374757677616263646566676869
      [Length: 32]
```

(a) 请求

```
Internet Protocol Version 4, Src: 223.252.214.105, Dst: 192.168.1.102
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 60
    Identification: 0xa964 (43364)
  > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 55
    Protocol: ICMP (1)
    Header Checksum: 0x61e8 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 223.252.214.105
    Destination Address: 192.168.1.102
Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x5325 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 566 (0x0236)
    Sequence Number (LE): 13826 (0x3602)
    [Request frame: 117]
    [Response time: 18.344 ms]
  > Data (32 bytes)
      Data: 6162636465666768696a6b6c6d6e6f7071727374757677616263646566676869
      [Length: 32]
```

(b) 回复

时，TTL变为0，所以该路由器会将数据报丢掉，并送回一个ICMP包，这包括发IP包的源地址，IP包的所有内容及路由器的IP地址，tracert收到这个消息后便知道这个路由器存在于这个路径上，接着tracert再送出另一个TTL是2的数据报，以此类推，traceroute每次将送出的数据报的TTL加1来发现下一个路由器，这个重复的动作一直持续到某个数据报抵达目的地。这时，该主机并不会送回ICMP消息，因为它已是目的地了。

过程大致如图 7。

图 7: tracert原理

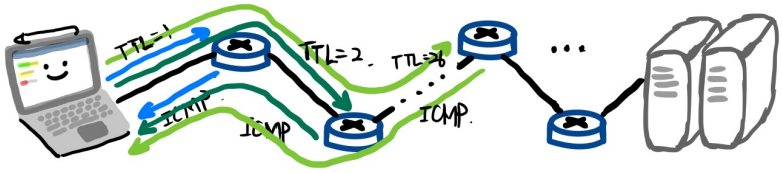
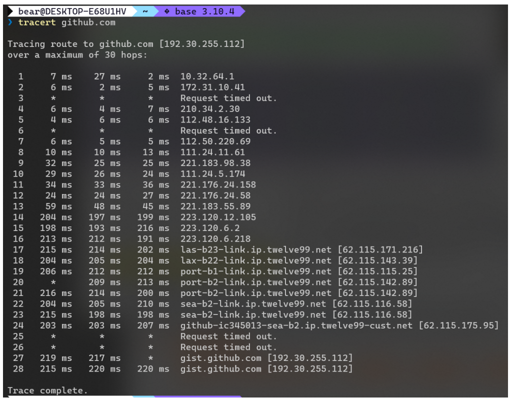
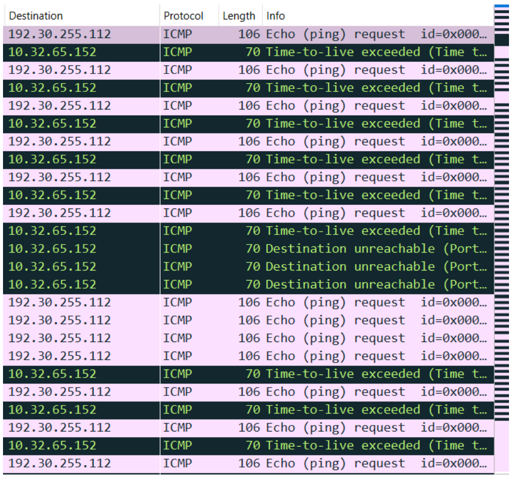


图 8是使用tracert跟踪GitHub的结果。可以看到，tracert会发送多个数据报，每个数据报的TTL不同，这样就可以跟踪到路径上的所有节点。到达GitHub需要28跳，因为服务器在美国，所以需要经过很多跳的路由器。因此这个结果符合实际情况。

图 8: tracert



(a) tracert输出



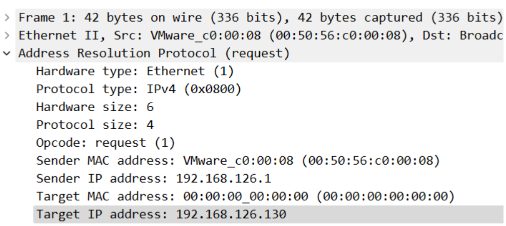
(b) 产生的数据报

1.6 ARP协议分析

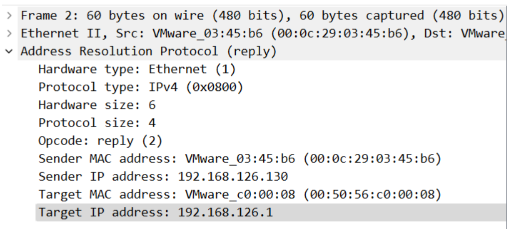
实验要求ping到局域网内的主机和远程主机，分析数据包内容。由于使用WIFI时不能保证两人一定在同一局域网中，因此我使用了另一种方法。

当虚拟机网络模式为NAT时，虚拟机和宿主机可以认为是在同一局域网中的主机，因此从虚拟机Ping宿主机的时候可以得到和Ping局域网内的主机一样的效果。如图 9

图 9: ARP请求



(a) 请求

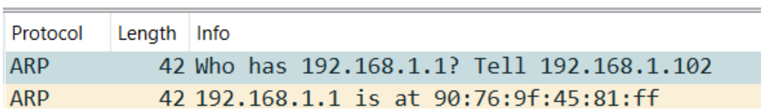


(b) 回复

ARP请求如图 9a，其中目的MAC地址是0，说明在进行广播。此时，位于192.168.126.130的虚拟机收到了ARP请求，并进行了回复，如图 9b。这样就知道了虚拟机的MAC地址。然后会加入ARP表中。

而Ping远程主机时则会产生对网关的ARP请求，如图 10。

图 10: Ping远程主机产生的ARP



因为根据ARP的工作方式，所有帧都必须传送到本地网段中的节点。如果目的 IPv4 主机在本地网络上，帧将使用此设备的 MAC 地址作为目的 MAC 地址。如果目的 IPv4 主机不在本地网络上，则源节点需要将帧传送到作为网关的路由器接口，或用于到达该目的地的下一跳。源节点将使用网关的 MAC 地址作为帧（其中含有发往其它网络上主机的 IPv4 数据包）的目的地址。因此会产生这种现象。

2 捕获和分析802.11数据

在按照实验要求进行实验时，需要注意：在Linux中启动Wireshark需要root权限。否则无法访问虚拟网卡。

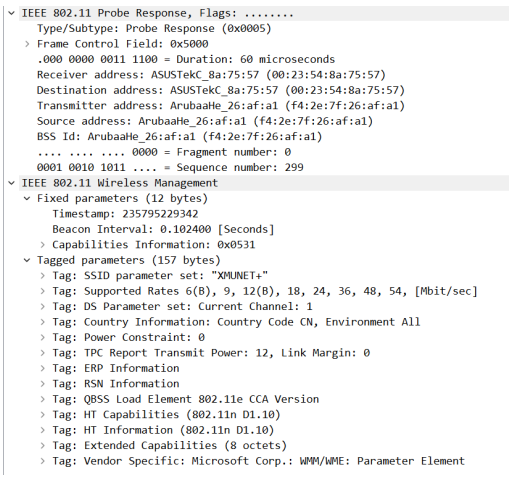
2.1 管理帧

在抓到的包中可以发现Probe Request和Beacon两种管理帧。如图 11。 Beacon帧主要来声明网络的存在。定期传送的信标可让移动式工作站该网络的存在，从而调整加入该网络所必需的参数。设备通过Probe Request帧来扫描所在区域内的802.11网络。若Probe Request帧探查的网络与之兼容，该网络就会回复Probe Response帧给予响应。

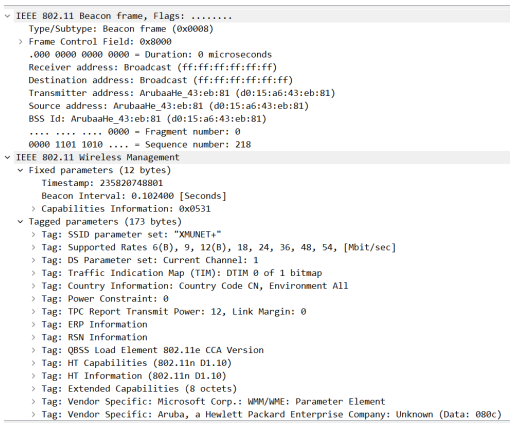
其中Beacon帧广播了SSID、信道、BSSID等信息，如图 11b。

可以看出SSID是XMUNET+，信道是1，BSSID是d0:15:a6:43:eb:81。

图 11: 管理帧



(a) Probe Response



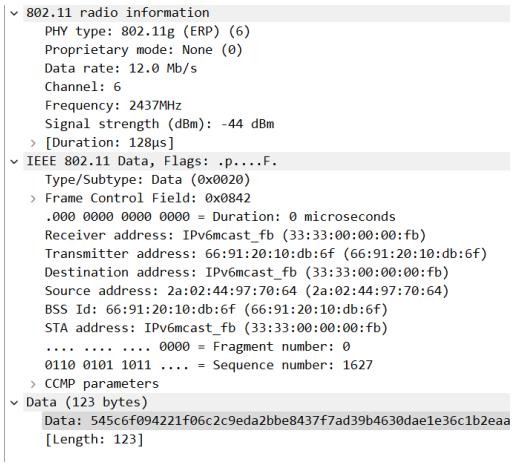
(b) Beacon

2.2 数据帧

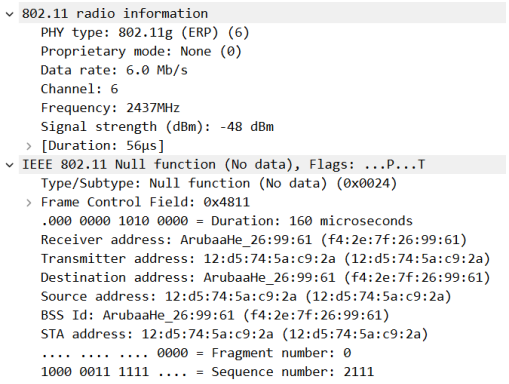
在抓到的包中可以发现Data、NULL Data、QoS Data和NULL QoS Data4种数据帧。如图 12。

数据帧会将上层协议的数据置于帧主体加以传递。

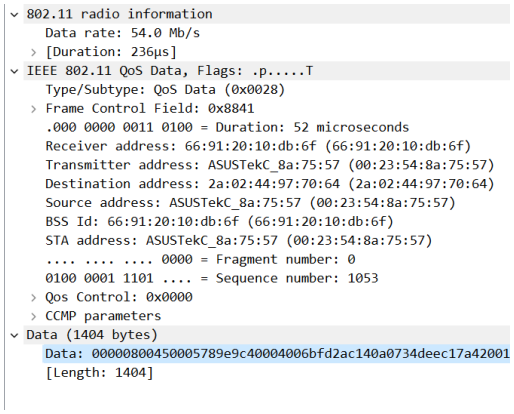
图 12: 数据帧



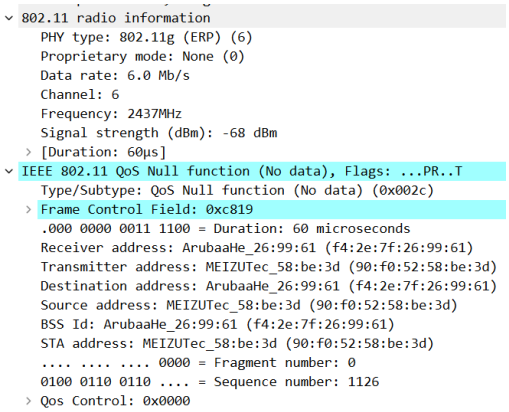
(a) Data



(b) Null Function



(c) QOS Data



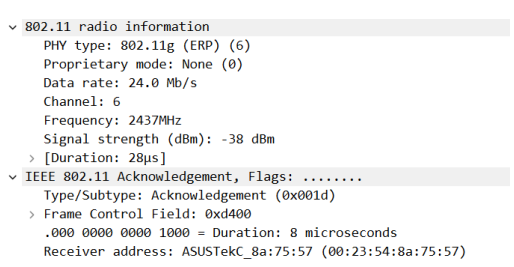
(d) QOS Null Function

2.3 控制帧

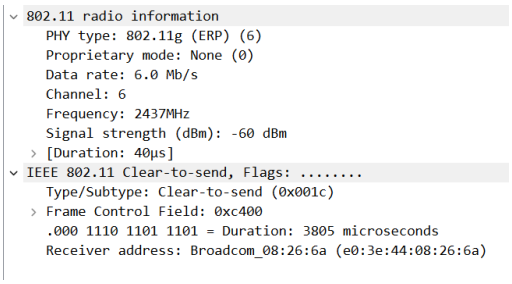
在抓到的包中可以发现Acknowledgment和Clear to Send两种控制帧。如图 13。每个发送的单播报文，接收者在成功接收到发送报文后，都要发送一个应答ACK进行确认。它的Duration是8微秒，能够反映出ACK信号在整个帧交换过程中位居何处。

目的客户端收到RTS后，发送一个CTS报文，这样在客户端覆盖范围内所有的设备都会在指定的时间内不发送数据。这里指定的时间是3805微秒。

图 13: 控制帧



(a) Acknowledgment



(b) Clear to Send

3 探索Wireshark更多功能和其它抓包工具

3.1 数据流追踪

向百度图片搜索发送图片，可以使用数据流追踪来追踪数据流的传输过程。如图 14。

图 14: 数据流追踪

3.2 协议分层统计

以802.11为例，可以通过协议分层统计统计使用到了哪些协议，如图 15。

可以发现所有的包都包含在802.11中。

图 15: 协议分层统计