数据通信与计算机网络 Lab5:数字链路层

王傲

15300240004

数据通信与计算机网络

COMP130017.01

指导教师：肖晓春

1. 地址解析协议

实验内容：

根据 MAC 地址是否在 ARP 缓存中查看地址解析流程的不同。发送方 IP 地址为 10.141.208.237，Ping 的主机地址为 10.141.208.24。

实验分析：

1. 观察本机开始发送的分组和响应分组，分析源地址和目的地址。分别记下其 MAC 地址。

由于是在服务器上进行捕获的，没有开启混杂模式的情况下收到的分组也比较繁杂。Ping 指令执行了两次，每次两组。第一次开始的包在第 24，第二次开始的包在 415。第一次 Ping 指令没有 ARP 请求和应答分组，第二次 Ping 指令对应的 ARP 请求分组在第 413，ARP 应答分组在第 414。

本机发送的第一个 ICMP 请求分组如图：

```
▶ Frame 24: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▶ Ethernet II, Src: Dell_0c:46:78 (44:a8:42:0c:46:78), Dst: Dell_34:e2:48 (f8:bc:12:34:e2:48)
▶ Internet Protocol Version 4, Src: 10.141.208.237, Dst: 10.141.208.24
▼ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4ce8 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence number (BE): 115 (0x0073)
    Sequence number (LE): 29440 (0x7300)
    [Response frame: 25]
  ▶ Data (32 bytes)
```

这个 ICMP 请求分组的响应分组如下：

```
▶ Frame 25: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▶ Ethernet II, Src: Dell_34:e2:48 (f8:bc:12:34:e2:48), Dst: Dell_0c:46:78 (44:a8:42:0c:46:78)
▶ Internet Protocol Version 4, Src: 10.141.208.24, Dst: 10.141.208.237
▼ Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x54e8 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence number (BE): 115 (0x0073)
    Sequence number (LE): 29440 (0x7300)
    [Request frame: 24]
    [Response time: 0.228 ms]
  ▶ Data (32 bytes)
```

可以看见，源主机的 IP 地址为 10.141.208.237，MAC 地址为 44:a8:42:0c:46:78；目的主机的 IP 地址为 10.141.208.24,MAC 地址为 f8:bc:12:34:e2:48。

此外，将 MAC 地址的前三个字节转换为生产厂商的字符串的话，源主机是 Dell_0c，目的主机是 Dell_34。后面三个字节不变。

2. 观察目的IP地址已经在ARP高速缓存里的情况下，捕获的分组的应答情况，将这几个分组截图。

IP地址已在ARP缓存中时，捕获分组的情况：

21	1.743510	SuperMic_c5:a1…	Broadcast	ARP	60	Who has 10.141.209.112? Tell 10.141.209.108
22	1.776350	TyanComp_de:3e…	Broadcast	ARP	60	Who has 10.141.210.151? Tell 10.141.210.17
23	1.901512	AsrockIn_e1:87…	Broadcast	ARP	60	Who has 192.168.1.32? Tell 192.168.1.195
24	1.940138	10.141.208.237	10.141.208.24	ICMP	74	Echo (ping) request id=0x0001, seq=115/29440, ttl=128 (reply in 25)
25	1.940366	10.141.208.24	10.141.208.237	ICMP	74	Echo (ping) reply id=0x0001, seq=115/29440, ttl=64 (request in 24)
32	2.658502	TyanComp_dd:ed…	Broadcast	ARP	60	Who has 10.141.210.151? Tell 10.141.210.22
33	2.742938	SuperMic_c5:a1…	Broadcast	ARP	60	Who has 10.141.209.112? Tell 10.141.209.108
34	2.901496	AsrockIn_e1:87…	Broadcast	ARP	60	Who has 192.168.1.32? Tell 192.168.1.195
35	2.948905	10.141.208.237	10.141.208.24	ICMP	74	Echo (ping) request id=0x0001, seq=116/29696, ttl=128 (reply in 36)
36	2.949127	10.141.208.24	10.141.208.237	ICMP	74	Echo (ping) reply id=0x0001, seq=116/29696, ttl=64 (request in 35)

可以看见，当IP地址已经在ARP缓存中时，在ICMP分区前是没有相应的ARP查询和响应的（图中的ARP查询与ICMP查询和应答无关）。

3. 再次观察目的IP地址不在ARP高速缓存里的情况下，捕获的分组的应答情况，将这几个分组截图。

IP地址不在ARP缓存中时，捕获分组的情况：

413	9.641173	Dell_0c:46:78	Broadcast	ARP	42	Who has 10.141.208.24? Tell 10.141.208.237
414	9.641405	Dell_34:e2:48	Dell_0c:46:78	ARP	60	10.141.208.24 is at f8:bc:12:34:e2:48
415	9.641433	10.141.208.237	10.141.208.24	ICMP	74	Echo (ping) request id=0x0001, seq=117/29952, ttl=128 (reply in 416)
416	9.641663	10.141.208.24	10.141.208.237	ICMP	74	Echo (ping) reply id=0x0001, seq=117/29952, ttl=64 (request in 415)
424	10.456951	TyanComp_de:3e…	Broadcast	ARP	60	Who has 10.141.210.151? Tell 10.141.210.9
425	10.484419	TyanComp_dd:f2…	Broadcast	ARP	60	Who has 10.141.210.151? Tell 10.141.210.13
426	10.520634	Vmware_60:f9:f0	Broadcast	ARP	60	Who has 10.141.210.46? Tell 10.141.210.61
427	10.659771	10.141.208.237	10.141.208.24	ICMP	74	Echo (ping) request id=0x0001, seq=118/30208, ttl=128 (reply in 428)
428	10.660001	10.141.208.24	10.141.208.237	ICMP	74	Echo (ping) reply id=0x0001, seq=118/30208, ttl=64 (request in 427)

可以看见，第一次IP地址不在ARP缓存中时，主机会通过ARP进行广播，询问目的主机IP地址对应的MAC地址；相应目的主机收到广播消息后，会将自己的MAC地址单播给请求主机。而再次发送分组的时候，相应的IP地址就已经在ARP缓存中了，所以不需要再次进行广播。

4. 分析以上两种情况的不同点。

不同点就是当在同一局域网的目的主机的IP地址不在ARP缓存中时，请求主机会进行ARP广播，获得目的主机的响应，得到MAC地址并记录在ARP缓存中。若目的主机的IP地址已经在ARP缓存中时，则封装帧的时候会直接利用缓存中的地址而不再进行广播。

每台主机以APR高速缓存的形式维护一张已知IP地址到MAC地址的转换表。发送一个IP分组到具体的目的主机之前，都要访问ARP高速缓存。如果找到一个对应的MAC地址，那么

IP分组就放在链路层帧的数据部分，而帧的目的地址将被设置为ARP高速缓存中找到的MAC地址。如果没有发现IP地址的转换项，那么本机将广播一个报文，要求具有此IP地址的主机用它的MAC地址作出响应。具有该IP地址的主机直接应答请求方，并且把新的映射项填入ARP高速缓存。

2. MAC地址欺骗

实验分析：

从获得的信息可以看出：

- A的IP地址为10.141.255.198，MAC地址为50:78:8b:80:44:e5
- B的IP地址为10.141.255.196，MAC地址为50:78:8b:80:44:e5
- C的IP地址为10.141.255.194，MAC地址为50:78:8b:80:45:35
- D的IP地址为10.141.255.192，MAC地址为50:78:8b:80:44:ce

可以看见，A和B的MAC地址相同。

按照实验设计，B和A的MAC地址相同，在同一局域网中B会收到C发向A的数据。但是，根据给出的数据，B并没有收到C发向A的UDP数据，可能存在一些问题。

1. 观察A、B、D中的分组，找出相关的通信数据报文，分析此时请求分组（或响应分组）中源地址和目的地址，将该地址截图。分析产生A、B、D数据包的原因。

A中：

```
46 15.755793 10.141.255.194  10.141.255.198  IPv4     1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=7666) [Reassembled in #51]
47 15.755944 10.141.255.194  10.141.255.198  IPv4     1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=7666) [Reassembled in #51]
48 15.756095 10.141.255.194  10.141.255.198  IPv4     1514 Fragmented IP protocol (proto=UDP 17, off=2960, ID=7666) [Reassembled in #51]
49 15.756246 10.141.255.194  10.141.255.198  IPv4     1514 Fragmented IP protocol (proto=UDP 17, off=4440, ID=7666) [Reassembled in #51]
50 15.756406 10.141.255.194  10.141.255.198  IPv4     1514 Fragmented IP protocol (proto=UDP 17, off=5920, ID=7666) [Reassembled in #51]
51 15.756472 10.141.255.194  10.141.255.198  UDP       834 1097 → 5001 Len=8192
```

MAC地址为：

```
▶ Ethernet II, Src: 50:78:8b:80:45:35 (50:78:8b:80:45:35), Dst: 50:78:8b:80:44:e5 (50:78:8b:80:44:e5)
▶ Internet Protocol Version 4, Src: 10.141.255.194, Dst: 10.141.255.198
```

B中没有应该有的通信数据报文。

B的地址：

```
▶ Ethernet II, Src: Hangzhou_34:ae:74 (00:23:89:34:ae:74), Dst: 50:78:8b:80:44:e5 (50:78:8b:80:44:e5)
▶ Internet Protocol Version 4, Src: 10.6.0.247, Dst: 10.141.255.196
```

D的MAC地址和A、B不同，没有收到C发送的信息。

D的地址：

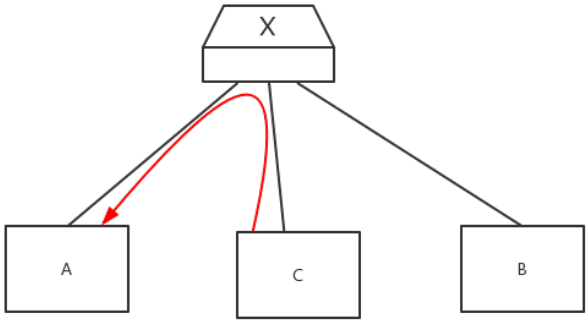
```
▶ Ethernet II, Src: Hangzhou_34:ae:74 (00:23:89:34:ae:74), Dst: 50:78:8b:80:44:ce (50:78:8b:80:44:ce)
▶ Internet Protocol Version 4, Src: 204.245.6.34, Dst: 10.141.255.192
```

从获得的信息可以看出：

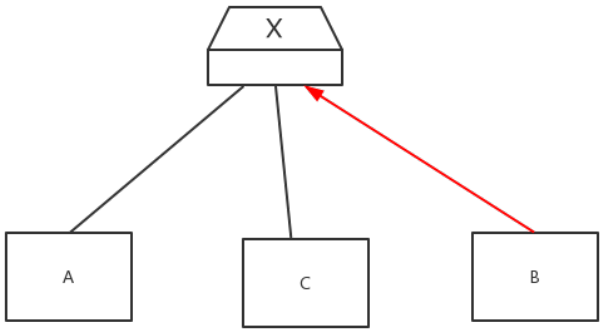
- A的IP地址为10.141.255.198，MAC地址为50:78:8b:80:44:e5
- B的IP地址为10.141.255.196，MAC地址为50:78:8b:80:44:e5
- C的IP地址为10.141.255.194，MAC地址为50:78:8b:80:45:35
- D的IP地址为10.141.255.192，MAC地址为50:78:8b:80:44:ce

可以看见，A和B的MAC地址相同。

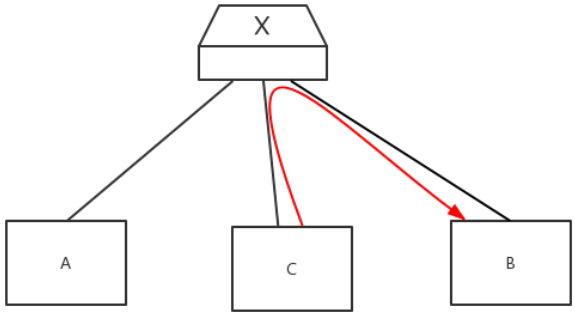
正常情况下，出现A、B、C数据包的原因是：



A、B、C连在同一台交换机上，C将数据发送给交换机，交换机根据转发表中MAC地址和端口的对应关系将数据发送给A对应的端口。



这时，B可能会修改自己的MAC地址，并通过传送数据来修改交换机中端口和MAC地址的对应关系，即将自己MAC的地址（也就是A的MAC地址）对应到自己连接的交换机的端口。



这样，C再次发送数据的话，交换机就会把数据转发到B的端口，B就会收到A本应收到的数据。

而D则因为MAC地址不同，不会收到C的信息。

3. 以太网帧

实验内容：

1. 你所捕获的帧是802.3帧，还是DIX V2帧？为什么？

捕获的是DIX V2的帧。

区别802.3帧和以太网II的帧的方法在于类型/长度。802.3帧的长度不会超过1500，而以太网II的帧中的类型值均超过了1500（0x05DC），比如IPv4的类型值是0x0800。

任取几个分组，查看以太网层：

```
▼ Ethernet II, Src: TyanComp_dd:ed:42 (00:e0:81:dd:ed:42), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Source: TyanComp_dd:ed:42 (00:e0:81:dd:ed:42)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
```

```
▼ Ethernet II, Src: Dell_0c:46:78 (44:a8:42:0c:46:78), Dst: Dell_34:e2:48 (f8:bc:12:34:e2:48)
  ▶ Destination: Dell_34:e2:48 (f8:bc:12:34:e2:48)
  ▶ Source: Dell_0c:46:78 (44:a8:42:0c:46:78)
    Type: IPv4 (0x0800)
```

可以看见，一个为0x0806，一个为0x0800，均超过了1500，所以是以太网II的帧。并且，根据Wireshark的分析结果，显示捕获的是以太网II的帧，证明了这个结果。

2. 注意帧的类型字段，指出包含ARP分组，IP分组的帧的类型字段不同点。

由上面的图可以看出，包含ARP分组的帧的类型为0x0806,包含IP分组的帧的类型是

0x0800，两者不同且均大于1500，可以进行区分。

4. 讨论与研究

1. 暂时不需要。MAC地址有48位，并且根据MAC地址的申请规则，有几十万亿个地址，而全球才60亿人。未来还是存在用完的可能性的，但是MAC地址不一定需要像IP地址一样全球唯一，在一个局域网内不同就可以了。

MAC地址是厂商向IEEE申请后，自己分配的。根据IEEE公布的IEEE OUI和公司ID分配文件，共有约33260个生产厂家，按每个生产厂家均使用了所有的MAC地址来计算，约有5千亿已被分配。实际数量肯定是小于这个值的，所以还有大量的MAC地址剩余。

2. 仅仅注册MAC地址并不可行，因为存在MAC地址欺骗的情况。

可以尝试将IP地址与MAC地址绑定，两者均合法且匹配的才予以授权。

会产生冲突，MAC地址相同的计算机其中一台无法使用网络资源。在不同的子网中就不会有问题。

保护网络免受攻击的物理安全十分重要，因为物理层面是网络的基础，物理层面不安全，其他安全就无从谈起。