实验二 ARP与DNS协议分析实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号： | 7-1 |  |  |  |  |
| 姓名： | 白佳兴 | 学号： | 2204311549 | 班级： | 计算机2105 |
| 姓名： |  | 学号： |  | 班级： |  |
|  |  |  |  |  |  |

1. 实验目的

分析ARP协议报文首部格式以及在同一网段内和不同网段间的解析过程，分析DNS协议的工作过程。

1. 实验内容

（1）利用校园网及云服务器搭建内网、外网环境；

（2）用Wireshark截获ARP报文，分析报文结构及ARP协议在同一网段和不同网段间的解析过程；

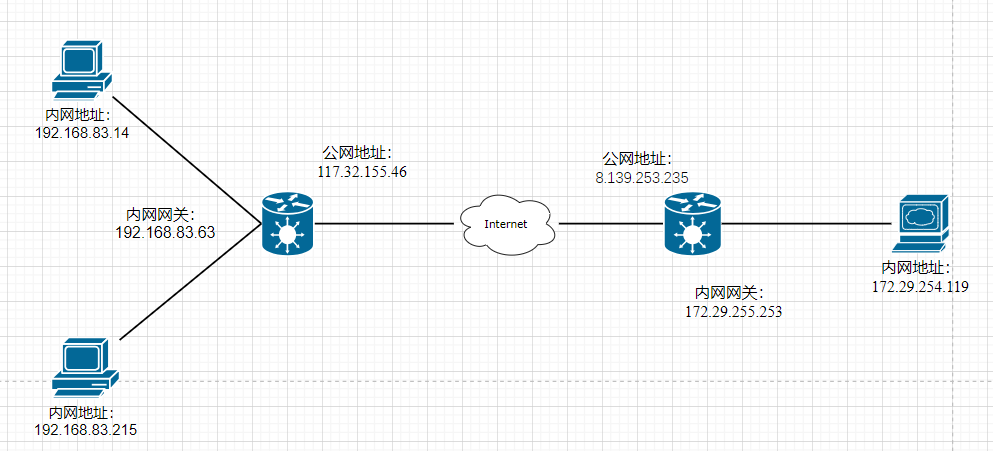
（3）用Wireshark截取DNS报文，分析DNS工作过程。

1. 实验环境与分组

每2名同学一组，以现有的校园网络环境及云服务器搭建内网、外网网络。

1. 实验网络拓扑皆否

按照实际网络情况绘制拓扑图【标注出内、外地址】。



1. 实验过程及结果分析

【过程记录应当详尽，截图并加以说明。以下过程和表格仅供参考。】

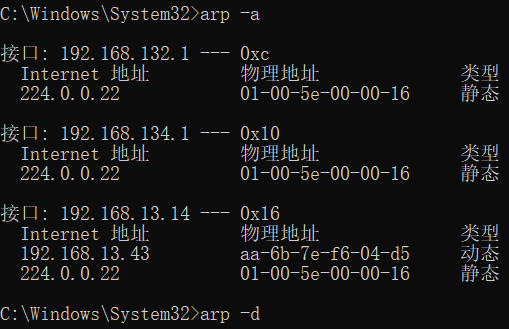
* 1. ARP协议分析

（一）同一网段内IP的ARP协议分析：

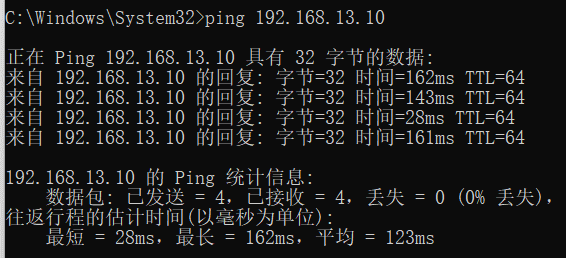
步骤1：在计算机终端的命令行窗口执行命令：

执行“arp –a”观察arp缓存；

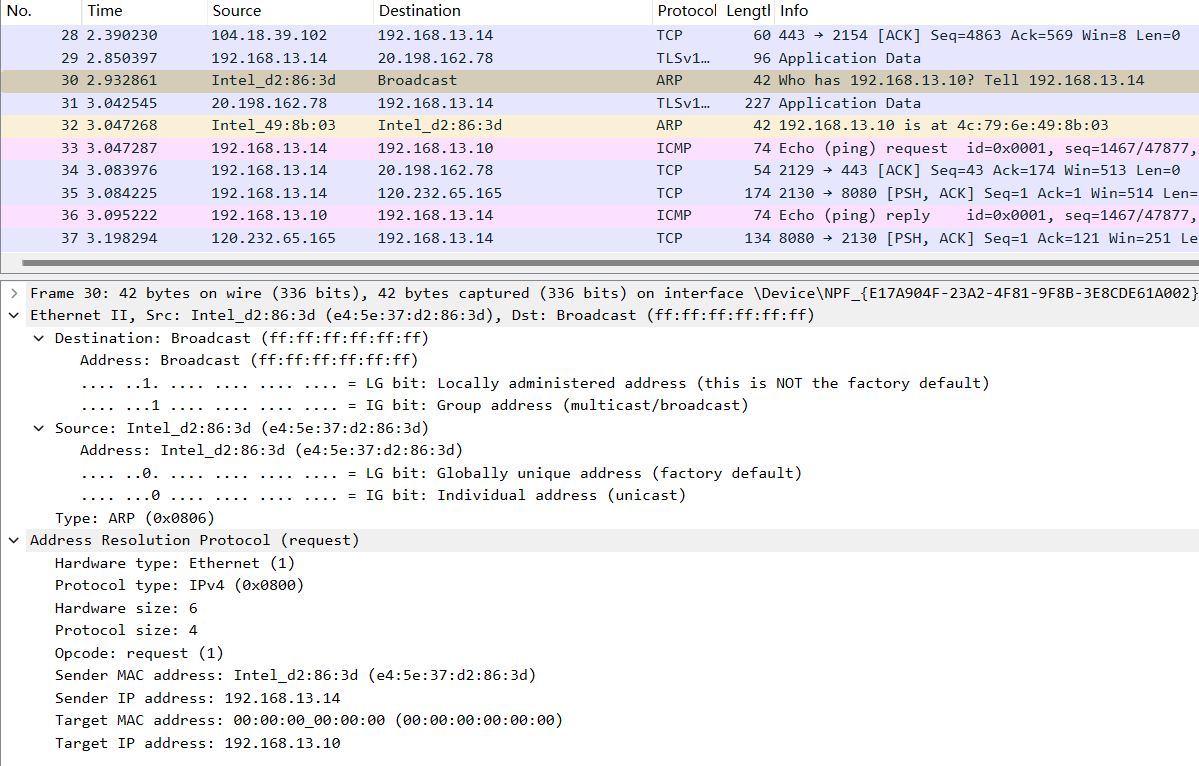
执行“arp –d”命令清空arp缓存。



步骤2：在计算机终端上运行Wireshark截获报文，在命令行窗口ping同一网段的另一设备地址。执行完后停止报文截获，筛选出相关的arp和icmp报文进行分析（源IP地址/MAC地址、目的IP地址/MAC地址等）。



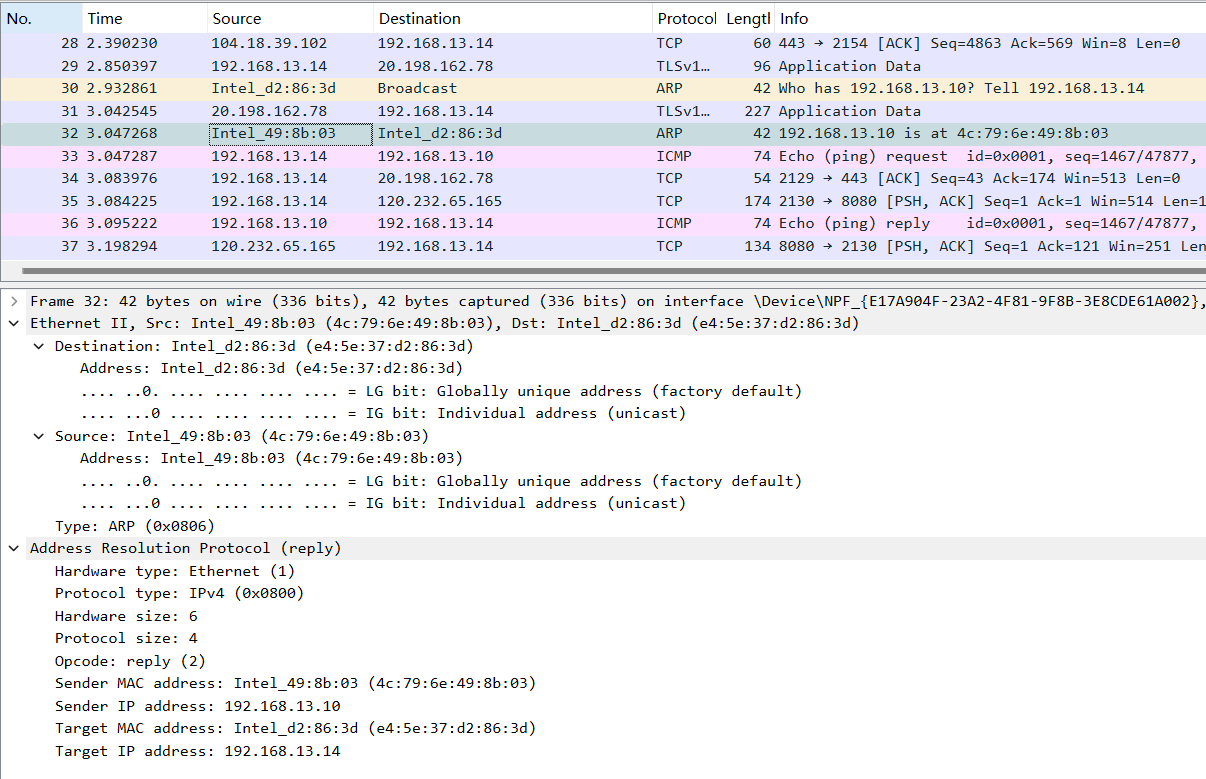
arp请求报文：



可以看出这是一个 ARP 请求报文，其目的 MAC 地址由于暂时不知道，被设置为 ff:ff:ff:ff:ff:ff。第一行由于是以太网连接，故硬件类型的值为 1。第二行标明使用 IP 协议。第三行标明硬件地址长度为 6 个字节。第四行标明协议地址长度为 4 个字节。第五行是操作类型字段，表示是 ARP 请求。第六、七行分别是发送方（主机 1）的 MAC 地址和 IP 地址。第八行是目的 MAC 地址，由于现在暂时不知道故全部填充为0。第九行是目的 IP 地址，就是该ARP报文要找的 MAC 地址对应的 IP 地址（即主机 2的IP地址）。

请求报文的具体内容是“Who has 192.168.13.10?Tell 192.168.13.14” 。本机ip是192.168.13.14，本机正在通过广播的形式寻找主机ip为192.168.13.10的MAC地址

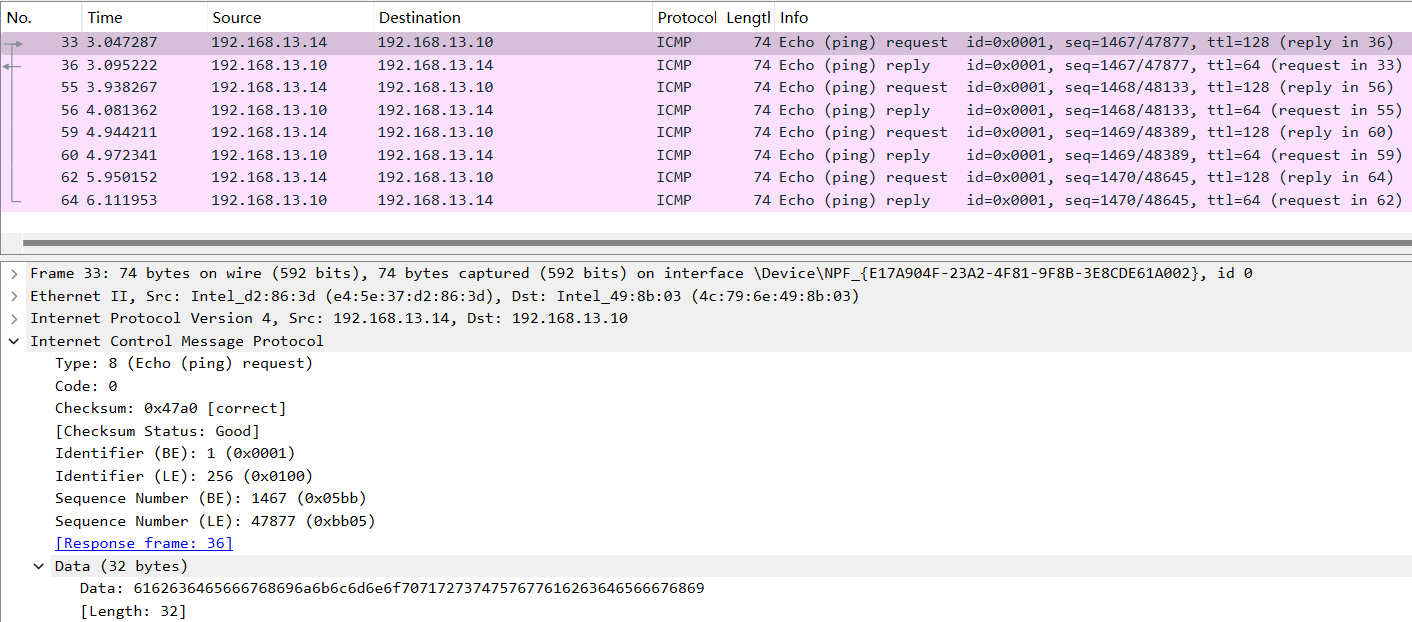
arp应答报文：



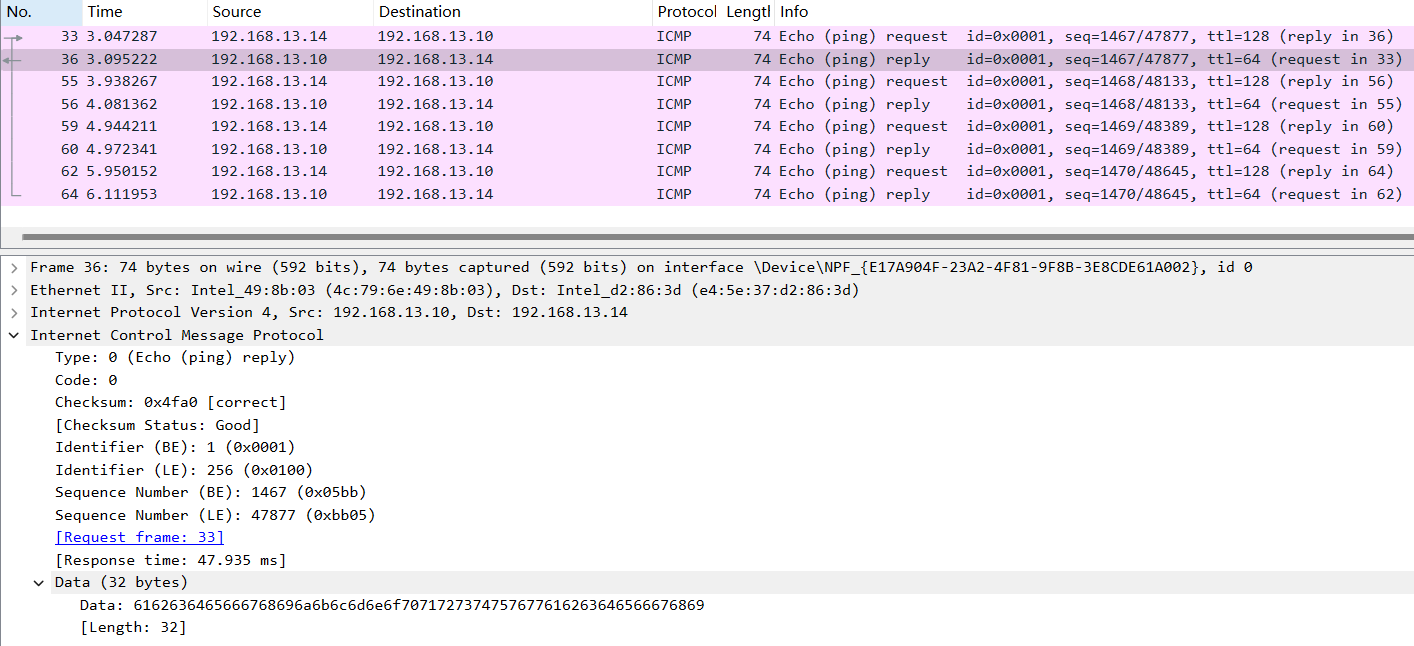
该报文和 ARP 请求报文在格式上相同，都是以太网连接，ipv4，6字节的硬件长度，4字节的协议长度，可以看出操作码字段的值为 2，代表这是一个reply响应报文。同时，该报文的收发双方的 IP 地址和 MAC 地址均为有效值，说明通过 ARP 协议我们已经正确得到了要找的 IP 地址所对应的 MAC 地 址，ARP 协议正确的发挥了它的功能。

该报文的具体内容是“192.168.13.10 is at 4c:79:6e:49:8b:03“，目标主机接受到本机的请求报文之后发送arp应答报文告诉其具体的Mac地址，当两个主机在同一网段时，主机A能直接向主机B广播ARP 报文，主机B在收到 ARP 报文后也能直接返回 ARP 响应报文，整个过程不需要网关参与，主机A得到的就是主机B的 MAC 地址。

ICMP报文

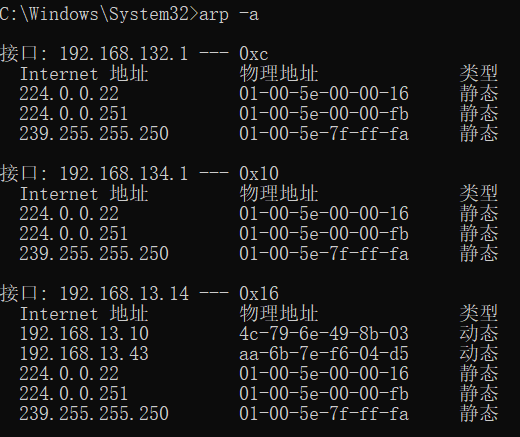


该报文为33号帧，type字段的值为8，code字段的值为0，表示回显请求类报文。Checksum字段的值为0x47a0，表示校验和。Identifier字段表示的是标识符，BE表示Big end，即按照大端的顺序表示为0x0001，LE表示Litter end,即按照小端的顺序表示为0x1000,可以发现这两个信息指向的是同一个二进制报文内容，只不过在不同的系统里采用的方式不同，类似的Sequence Number也分为大端BE和小段HE，分别为0x05bb和0xbb05。数据长度32字节。该报文的响应帧为36号帧。可以看出报文其实就是一些比特流，网络协议就是解释它的东西。ICMP在IP协议之上，用于在IP主机、路由器之间传递控制消息。



具体分析36号报文，发现其内容大部分与33号请求报文一致，不同的是type字段的值为0，表明该报文的类型为应答报文，还有Request frame为33，表明其对应的请求报文为33号报文，Response time响应时间为47.935ms.

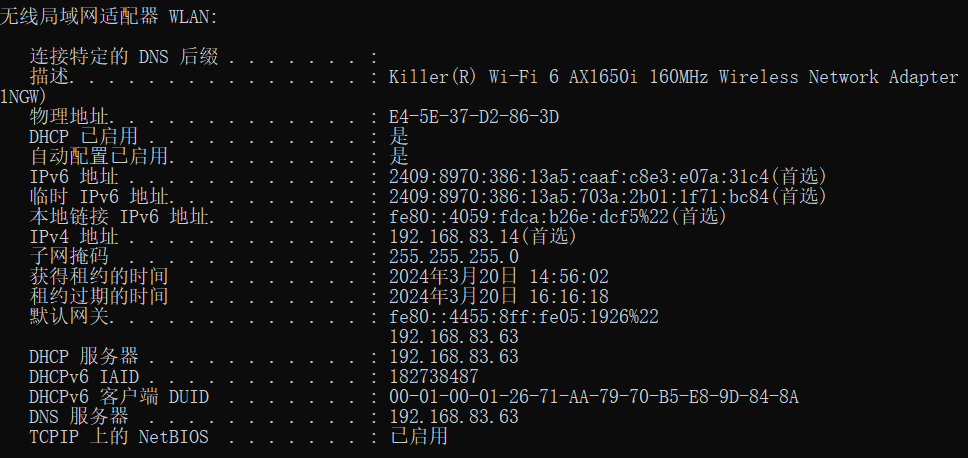
步骤3：在命令行窗口执行“arp –a”，记录结果。

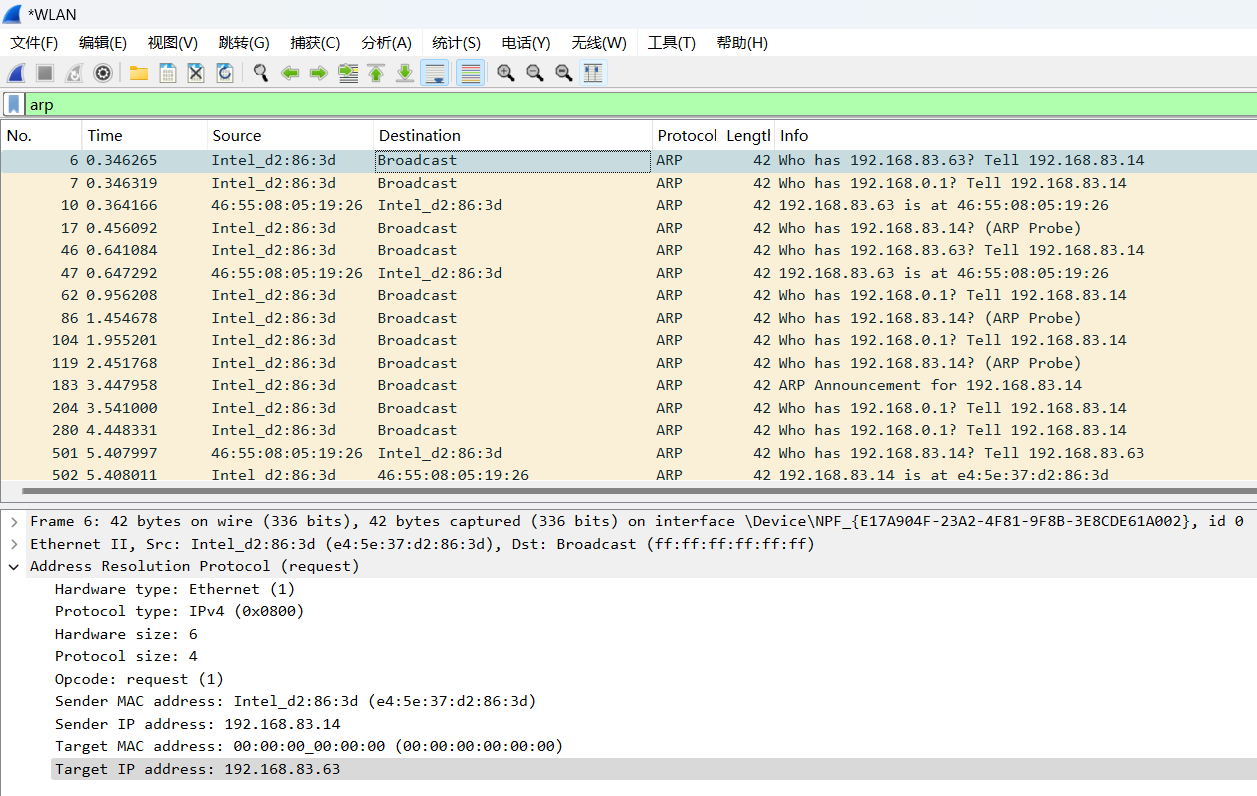


对比步骤1中的arp信息我们可以发现，ping成功之后会在本地计算机内保存目标计算机的MAC地址作为缓存。

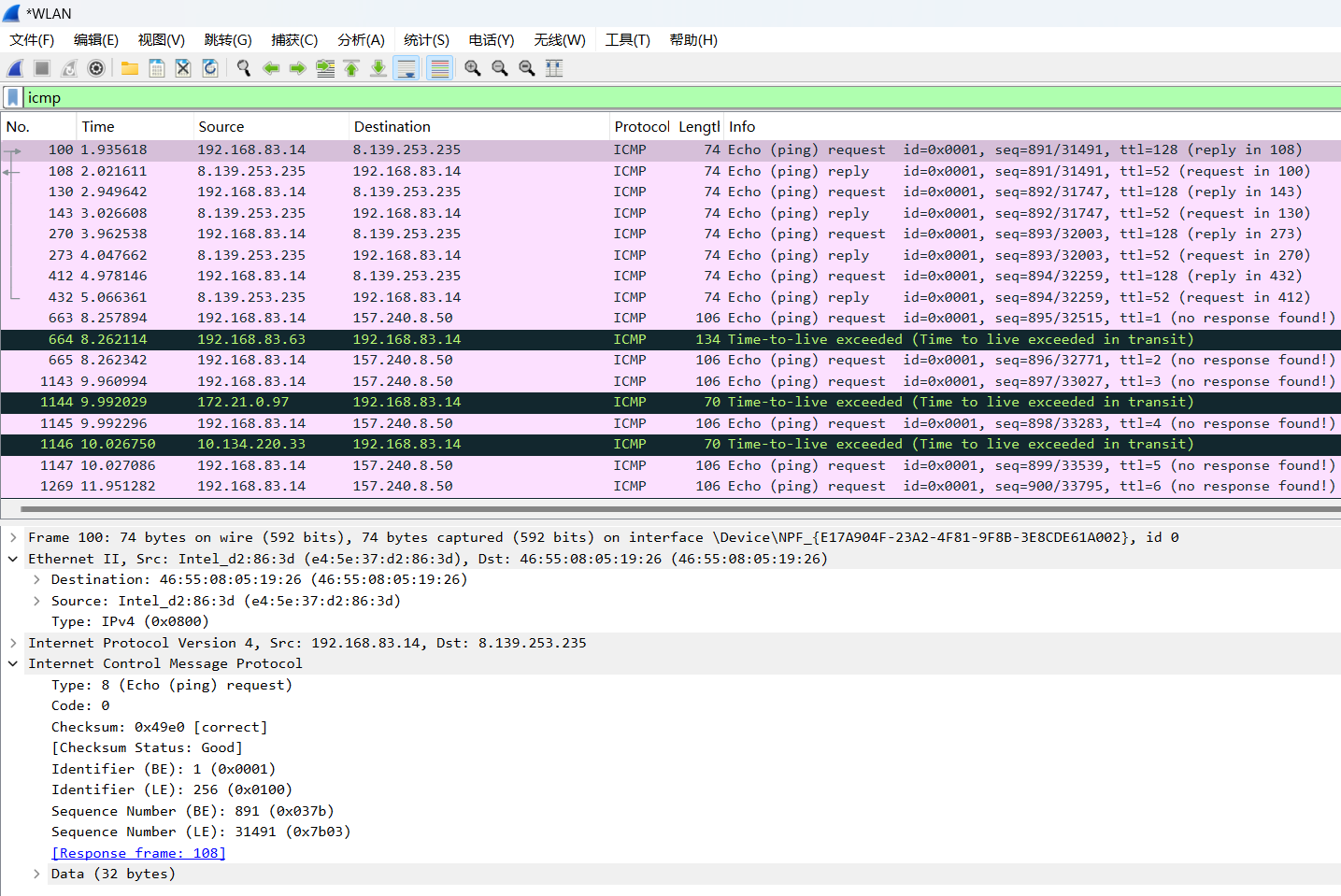
（二）不同网段的ARP协议分析

步骤1：在本地计算机和云服务器执行“arp –d”清空缓存，运行Wireshark捕获报文，在本地计算机ping云服务器地址。执行完后停止报文截获，筛选出相关的arp和icmp报文进行分析（arp与icmp报文的顺序，报文源IP地址/MAC地址、目的IP地址/MAC地址及其对应的主机等）。



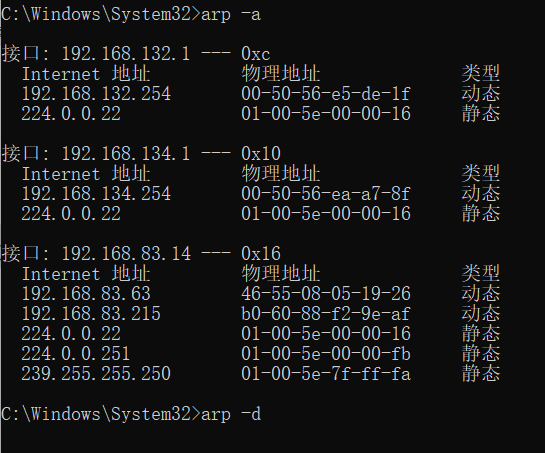


可以看出这也是一条 ARP 请求报文，和同一网段内的请求报文的区别在于它所请求的是默认网关的 MAC 地址。因为我们要访问的服务器在外网，不在同一网段中，因此主机无法直接向服务器广播 ARP 报文，只能先向网关发送 ARP报文，获取网关的 MAC 地址，然后将要发送的 ICMP 协议包交给网关，让网关代其向服务器发送。



本机在获得网关的MAC地址后往目标服务器发送数据包，可以发现ICMP数据包的发送顺序在ARP协议之后。

步骤2：执行“arp –a”命令，记录结果。



步骤3：分析捕获的报文，选中第一条ARP请求报文和第一条应答报文，填写2-1表。

表2-1 ARP请求报文和应答报文的字段信息

|  |  |  |
| --- | --- | --- |
| 字段 | 请求报文 | 请求报文 |
| Ethernet II Dst: | Broadcast ff:ff:ff:ff:ff:ff | Intel\_d2:86:3d (e4:5e:37:d2:86:3d) |
| Ethernet II Src: | Intel\_d2:86:3d(e4:5e:37:d286:3d) | 46:55:08:05:19:26 (46:55:08:05:19:26) |
| ARP Sender MAC address: | Intel\_d2:86:3d (e4:5e:37:d2:86:3d) | 46:55:08:05:19:26 (46:55:08:05:19:26) |
| ARP Sender IP address: | 192.168.83.14 | 192.168.83.63 |
| ARP Target MAC address: | 00:00:00\_00:00:00 (00:00:00:00:00:00) | Intel\_d2:86:3d (e4:5e:37:d2:86:3d) |
| ARP Target IP address: | 192.168.83.63 | 192.168.83.14 |

分析捕获的报文，选中第一条ICMP请求报文和第一条应答报文，填写表2-2。（对应主机填写本机、本地网关、服务器等）

表2-2 ICMP请求报文和应答报文的字段信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 请求报文 | 对应主机 | 应答报文 | 对应主机 |
| Ethernet II Src: | Intel\_d2:86:3d (e4:5e:37:d2:86:3d) | 本机 | 46:55:08:05:19:26 (46:55:08:05:19:26) | 服务器 |
| IP Src: | 192.168.83.14 | 本机 | 8.139.253.235 | 服务器 |
| Ethernet II Dst: | 46:55:08:05:19:26 (46:55:08:05:19:26) | 服务器 | Intel\_d2:86:3d (e4:5e:37:d2:86:3d) | 本机 |
| IP Dst: | 8.139.253.235 | 服务器 | 192.168.83.14 | 本机 |

步骤4：比较ARP协议在不同网段和相同网段内解析过程的异同。

相同点: ARP协议都是用来通过解析同一局域网内的IP地址获得对应的MAC地址，以便在链路层进行数据传输。基本流程:主机A需要向主机B发送数据，但不知道B的MAC地址，主机A会向本网段的所有主机广播一个ARP请求报文，其中包含A的IP地址和MAC地址，以及B的IP地址，所有收到ARP请求报文的主机都会检查报文中的IP地址是否与自己的IP地址匹配。如果IP地址不匹配，那么直接忽略。如果IP地址匹配，则该主机会向主机A发送一个ARP响应报文，其中包含B的MAC地址。主机A收到ARP响应报文后，会将B的MAC地址和IP地址缓存起来，以便下次直接使用。

不同点：在相同网段内，ARP解析直接发送广播，目标主机直接响应ARP请求，ARP请求报文是广播在本网段内所有主机，整个过程都不需要网关的参与。而在不同网段内，需要通过网关进行转发，最终由目标主机响应，响应经过网关转发后才到达源主机，ARP请求报文需要先发送给网关，由网关再转发给目标网段的所有主机进而让目标主机获得请求报文信息

* 1. DNS协议分析

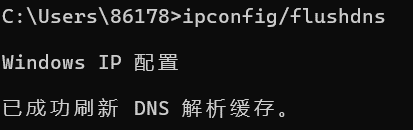
（一）默认DNS域名解析

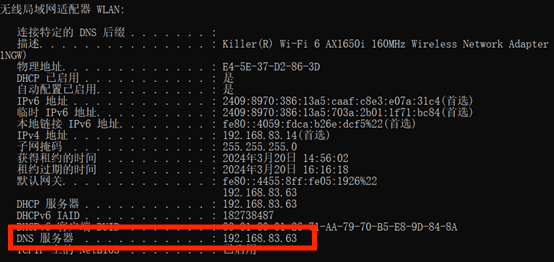
步骤1：在命令窗口执行命令：

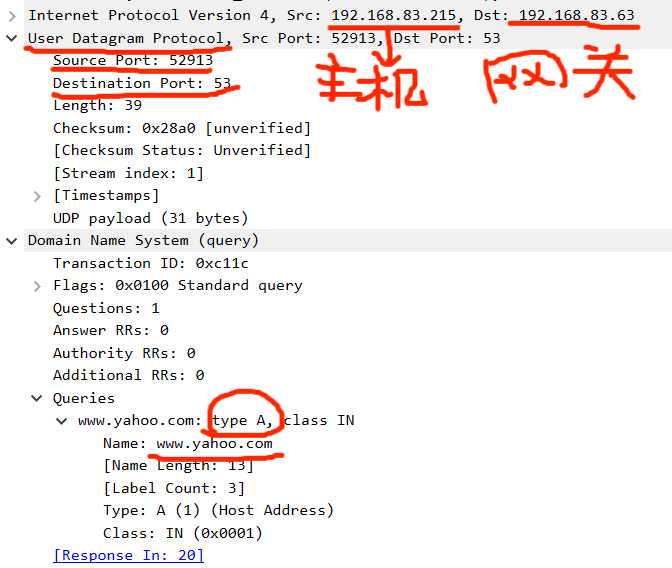
执行“ipconfig /displaydns”观察本地DNS缓存；

执行“ipconfig /flushdns”清除本地DNS缓存。

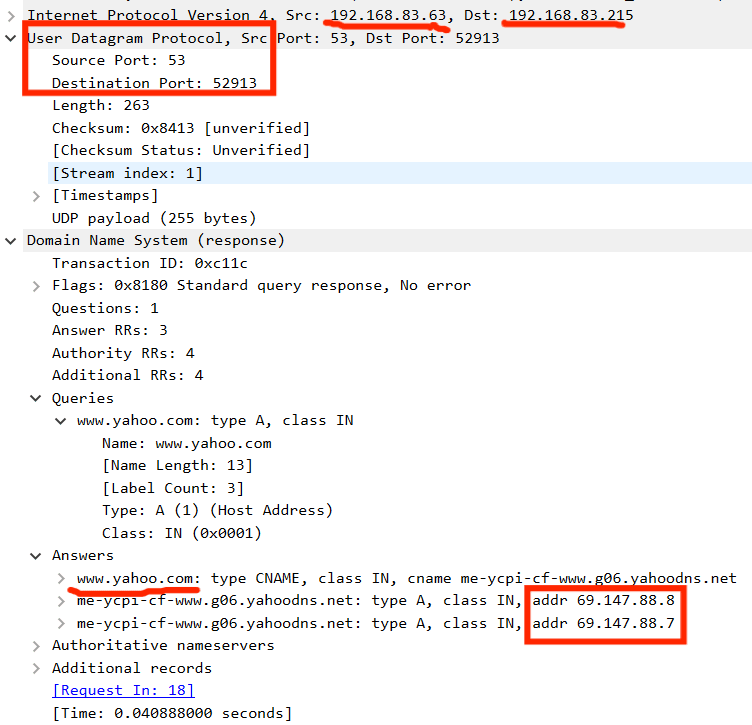




步骤2：在计算机终端上运行Wireshark截获报文，浏览器访问域名（如http://www.yahoo.com），网站打开后停止报文截获，观察分析DNS查询、回复报文分别包含哪些主要内容（UDP还是TCP、目的地址与本机默认DNS是否相同、源端口和目的端口、域名解析记录类型、解析出的IP地址等）。



查询报文中的192.168.83.215即为本机ip地址,192.168.83.63即为DNS服务器的ip地址，可以发现查询报文为UDP类型，目的地址就是DNS服务器的地址，源端口为52913，目的端口为53，域名解析记录类型为type A,class IN表示互联网地址查询的域名为[www.yahoo.com](http://www.yahoo.com)。



回复报文的格式和查询报文大致相同，这里查询一个域名返回了两个 IP 地址，分别为69.147.88.8和69.147.88.7。通过查阅资料得知这是使用了 DNS负载均衡技术。基于 DNS 的负载平衡是一种特定类型的负载平衡，它使用 DNS 在多个服务器之间分配流量。它通过提供不同的 IP 地址来响应 DNS 查询来做到这一点。负载平衡器可以使用各种方法或规则来选择分享哪个 IP 地址以响应 DNS 查询。这样就可以将流量分配到多台服务器上以提高性能和可用性，组织使用不同形式的负载平衡来加快网站和专用网络的速度。如果没有负载平衡，大多数互联网应用程序和网站将无法有效处理流量或正常运行。

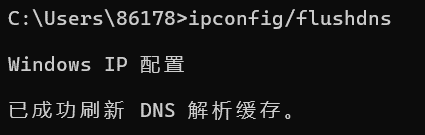
（二）指定DNS域名解析

步骤1：在命令窗口执行命令：

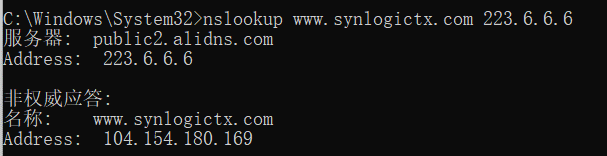
执行“ipconfig /displaydns”观察本地DNS缓存；

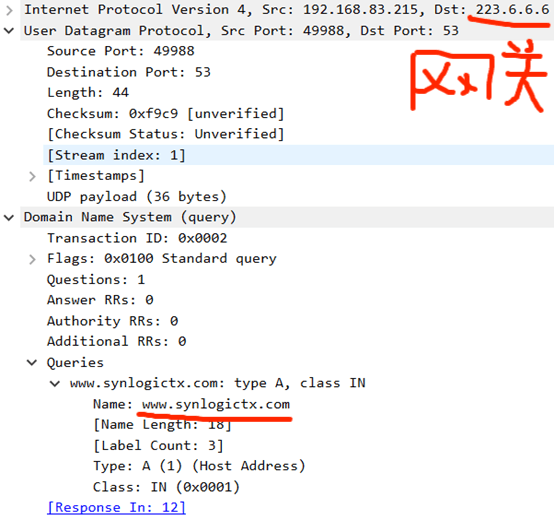
执行“ipconfig /flushdns”清除本地DNS缓存。



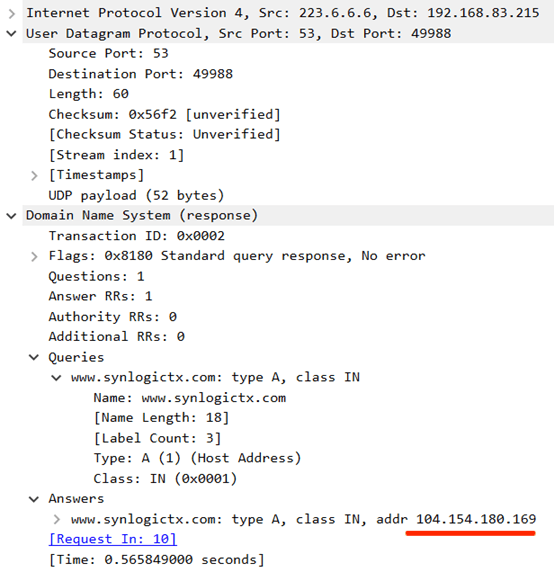


步骤2：在计算机终端上运行Wireshark截获报文，在命令窗口执行指定DNS服务器解析域名命令（如nslookup www.synlogictx.com 223.6.6.6），解析完毕后停止报文截获，观察分析DNS查询、回复报文分别包含哪些主要内容（UDP还是TCP、目的地址与本机默认DNS是否相同、源端口和目的端口、域名解析记录类型、解析出的IP地址等）。





查询报文的格式与之前默认DNS服务器的查询报文差不多，可以发现该报文的目的DNS服务器地址变成了我们设定的233.6.6.6。可以看出报文采用 UDP 协议，响应报文的源端口 49988，目的端口 53。



可以看到，源端口53，目的端口49922，由网关233.6.6.6发出，本机192.168.83.215接收数据，数据包为DNS类型，解析出www.synlogictx.co 的ip地址为104.154.180.169

* 1. 互动讨论主题

1. 发送方与接收方ARP与ICMP报文出现的次序成因；

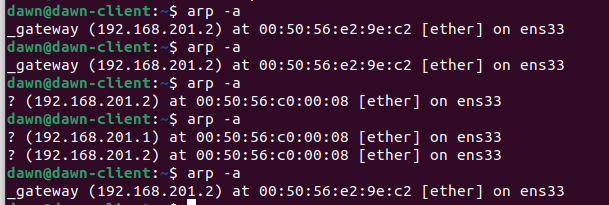
所有数据发送前都需要知道目的主机的 IP 和 MAC 地址。在清除了本地缓存的情况下，所有数据的发送都需要先利用 ARP 协议获得目的 IP 地址对应的 MAC 地址。可以明显地观察到，在两台主机通信的过程中ARP报文发送之后ICMP报文才开始发送，ICMP报文经常配合IP报文一起进行发送。

如果通信双方在同一网段中，如果发送方的ARP缓存中有接收方的MAC地址，那么直接发送即可，如果没有则在局域网内通过发送ARP报文获取接收方的MAC地址，获取之后才能发送ICMP报文进行正常的数据通信。

如果通信双方在不同的网段之中，则发送方首先需要在局域网内发送ARP报文以此来获取内部网关的MAC地址，然后将需要发送的ICMP报文转交给网关，由网关去公网中寻找对应IP的MAC地址，然后完成数据通信。

所以不论哪一种情况，都是ARP报文的发送顺序在ICMP报文之前

1. ARP的安全性问题；



由于 ARP 协议没有安全验证机制，因此存在 ARP 欺骗问题。例如假设某网段存在三台主机 PC1、PC2、PC3,他们的 IP 地址与 MAC 地址分别为：（IP1，MAC1）, （IP2，MAC2）, （IP3，MAC3）。如果 PC1 为黑客，那么 PC1 可以通过不断向 PC2 发送 ARP 响应报文，内容为（IP3,MAC1）,从而最终使 PC2 缓存中存放错误的 ARP 条目，这样 PC2在给 PC3 发送数据的时候就会错误的发给 PC1，这样就造成了 ARP 攻击与欺骗。 使用虚拟机作为目标机，可以看到在启动程序后 arp记录中网关对应的MAC地址被修改 在终止程序后又恢复

（3）DNS的欺骗带来的安全性问题；

在DNS欺骗中首先欺骗者向目标机器发送构造好的ARP应答数据包，ARP欺骗成功后，嗅探到对方发出的DNS请求数据包，分析数据包取得IP和端口号后，向目标发送自己构造好的一个DNS返回包，对方收到DNS应答包后，发现IP和端口号全部正确，即把返回数据包中的域名和对应的IP地址保存进DNS缓存表中，而后来的当真实的DNS应答包返回时则被丢弃。

对于避免DNS欺骗所造成危害，有以下几种措施

因为DNS欺骗前提也需要ARP欺骗成功。所以首先做好对ARP欺骗攻击的防范。不要依赖于DNS，尽管这样会很不方便，可以使用hosts文件来实现相同的功能。使用安全检测软件定期检查系统是否遭受攻击

* 1. \*进阶自设计

Scapy是一个 Python程序，它允许用户发送、嗅探、分析和伪造网络包。这种能力允许构建能够探测、扫描或攻击网络的工具。换句话说，Scapy是一个强大的交互式包操作程序。它能够伪造或解码大量协议的数据包，在网络上发送它们，捕获它们，匹配请求和响应，等等。Scapy可以轻松地处理大多数经典任务，如扫描、跟踪、探测、单元测试、攻击或网络发现。它可以代替hping、arpsoof、arp-sk、arping、p0f甚至Nmap、tcpdump和tshark的某些部分。

（1）使用scapy在Linux下写程序来模拟完成一个简单的ARP欺骗。

（2）使用scapy在Linux下写程序来模拟完成一个简单的DNS欺骗。完整的攻击实现工作量和难度都很大。为了降低难度，可以不实现中间人攻击，而是直接让受害者把DNS服务器修改为欺骗者的地址。