实验二 ARP与DNS协议分析实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号： | 3-1 |  |  |  |  |
| 姓名： | 陈实 | 学号： | 2215015058 | 班级： | 计算机2101 |
| 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；

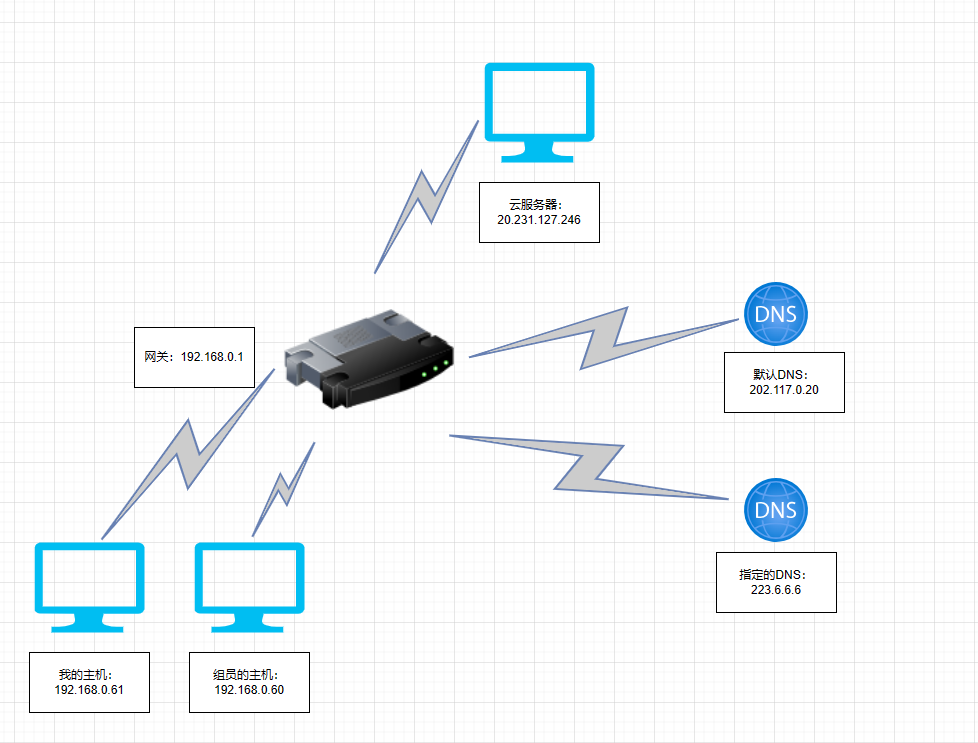
（3）用Wireshark截取DNS报文，分析DNS工作过程。

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

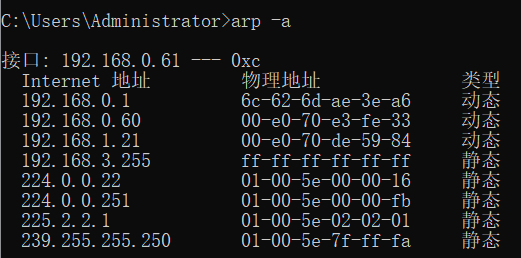
【过程记录应当详尽，截图并加以说明。以下过程和表格仅供参考。】

* 1. ARP协议分析

（一）同一网段内IP的ARP协议分析：

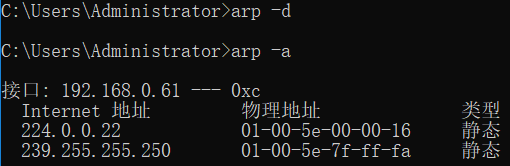
步骤1：在计算机终端的命令行窗口执行命令：

执行“arp –a”观察arp缓存；



图中显示的是ARP缓存，记录的是ip地址与物理地址之间的对应关系。ARP协议：ARP（Address Resolution Protocol）是地址解析协议，它是一种在网络中将IP地址转换为物理地址（如MAC地址）的协议。ARP的主要功能是在给定一个IP地址的情况下，查询该IP地址对应的物理地址

执行“arp –d”命令清空arp缓存。

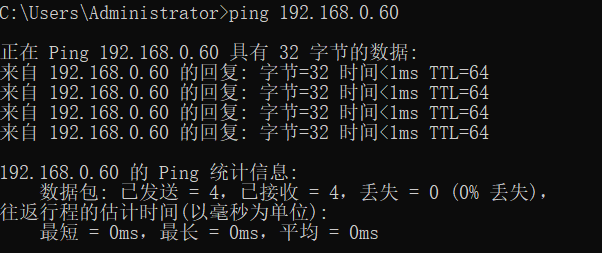


清空后仍然有表项存在：属于特殊的表项

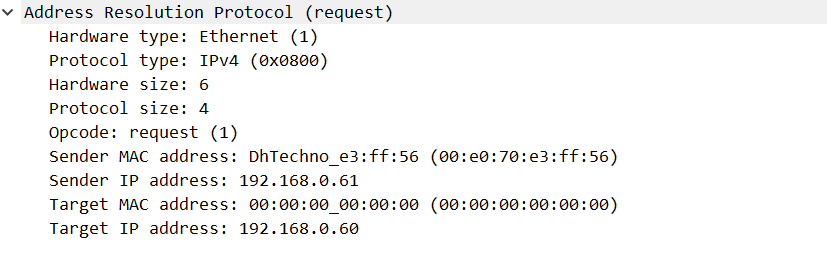
ARP缓存中的IP地址235.255.255.250和224.0.0.22都属于特殊的IP地址范围，有特定的用途：235.255.255.250是一个链路本地多播地址址，用于在本地网段内发送多播数据包，224.0.0.22是协议使用的一个特殊多播地址。

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：







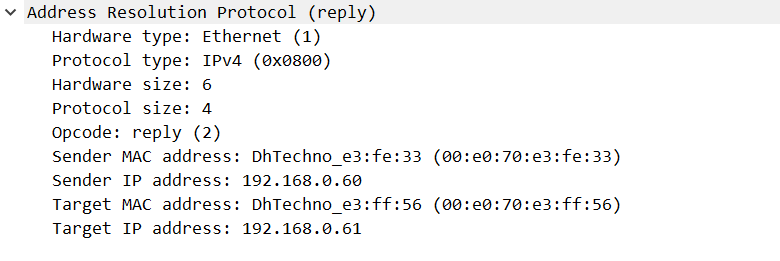
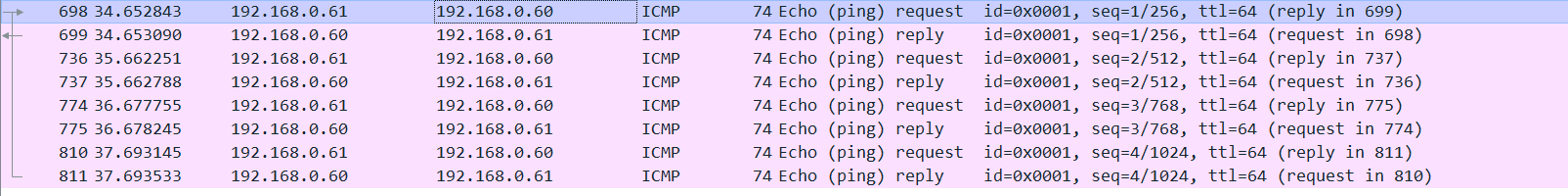


表2-1 ARP请求报文和应答报文的字段信息

|  |  |  |
| --- | --- | --- |
| 字段 | 请求报文 | 应答报文 |
| Ethernet II Dst: | ff:ff:ff:ff:ff:ff | 00:e0:70:e3:ff:56 |
| Ethernet II Src: | 00:e0:70:e3:ff:56 | 00:e0:70:e3:fe:33 |
| ARP Sender MAC address: | 00:e0:70:e3:ff:56 | 00:e0:70:e3:fe:33 |
| ARP Sender IP address: | 192.168.0.61 | 192.168.0.60 |
| ARP Target MAC address: | 00:00:00:00:00:00 | 00:e0:70:e3:ff:56 |
| ARP Target IP address: | 192.168.0.60 | 192.168.0.61 |

两台主机查询路由发现在同一局域网内，所以可以直接通信，本机（192.168.0.61）广播查询192.168.0.60的mac地址，报文中包含自己的ip和mac地址以及目标的ip地址，192.168.0.60收到后，根据包中的ip和Mac地址，向本机返回一个包含自己mac地址的arp包

ICMP



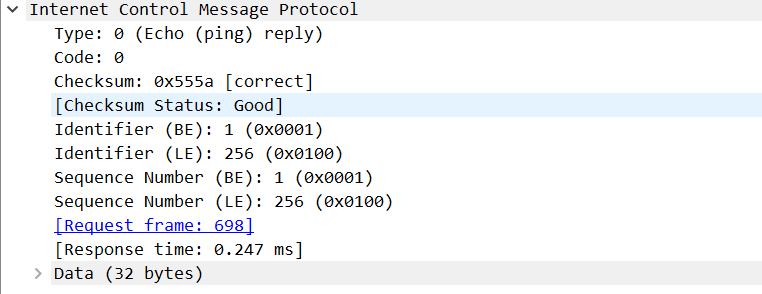
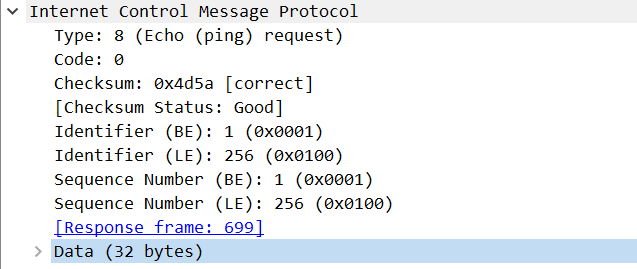
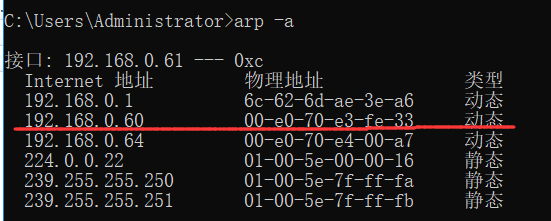


表2-2 ICMP请求报文和应答报文的字段信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 请求报文 | 对应主机 | 应答报文 | 对应主机 |
| Ethernet II Src: | 00:e0:70:e3:ff:56 | 本机 | 00:e0:70:e3:fe:33 | 组员主机 |
| IP Src: | 192.168.0.61 | 本机 | 192.168.0.60 | 组员主机 |
| Ethernet II Dst: | 00:e0:70:e3:fe:33 | 组员主机 | 00:e0:70:e3:ff:56 | 本机 |
| IP Dst: | 192.168.0.60 | 组员主机 | 192.168.0.61 | 本机 |

两台主机处于同一个局域网内，通过前面的arp查询，相互已经知道了对方的mac地址，可以不通过网关直接通信，所以Ethernet和IP都源和目的地址分别为对方

步骤3：在命令行窗口执行“arp –a”，记录结果。



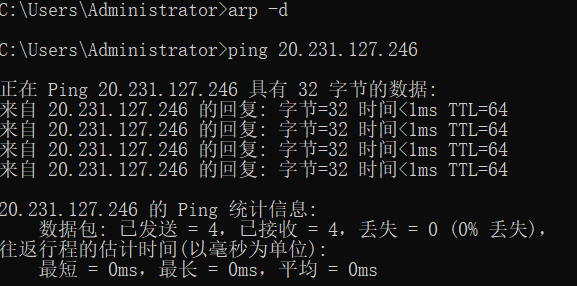
前面发送的ARP包收到回应后，记录到ARP缓存中，所以出现了192.168.0.60-00:e0:70:e3:fe:33的表项

（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）。

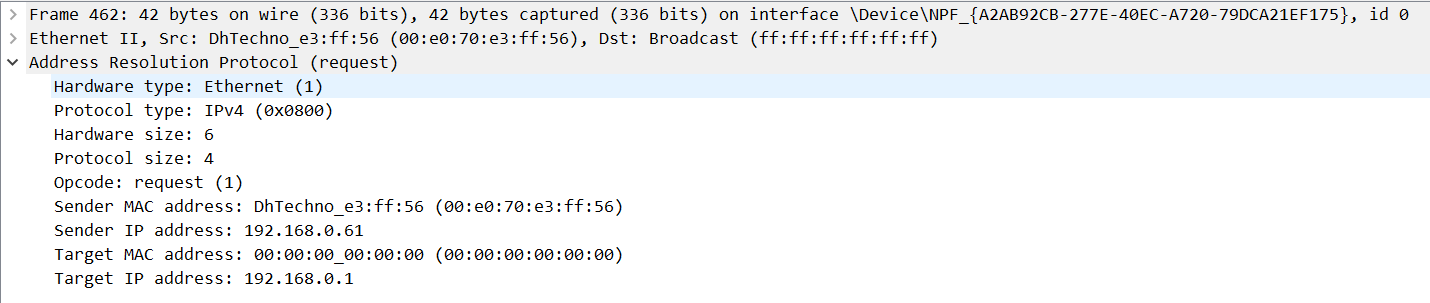
【如果网卡自动解析默认网关的MAC地址，可以删除默认网关设置，添加外网路由后再试。参考命令：route delete 0.0.0.0， route add 202.0.0.0 MASK 255.0.0.0 192.168.0.1】

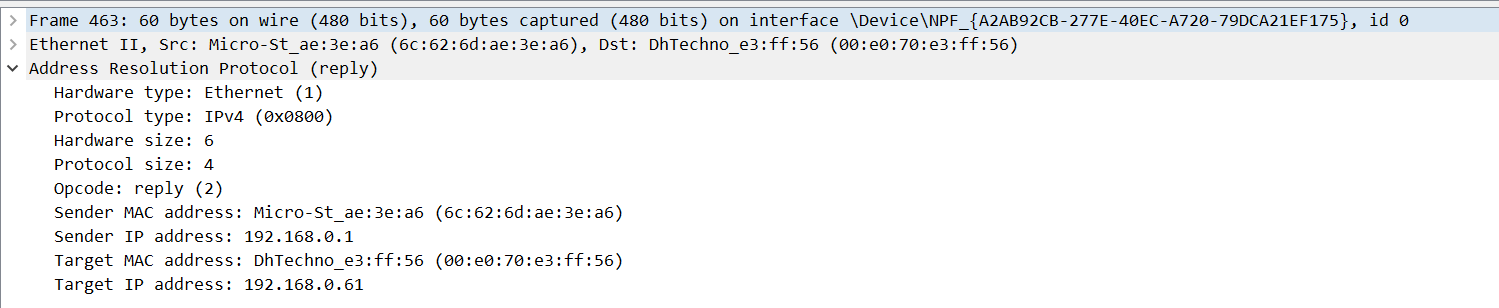
云服务器的公网ipv4地址为20.231.127.246，是服务器厂商提供出来用于让用户进行远程连接的，云服务器监听此地址来接受链接服务



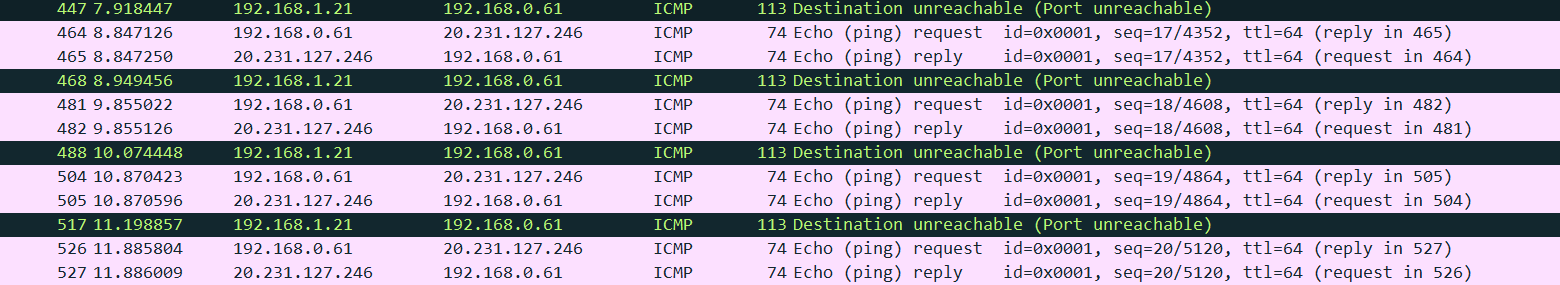
ARP:

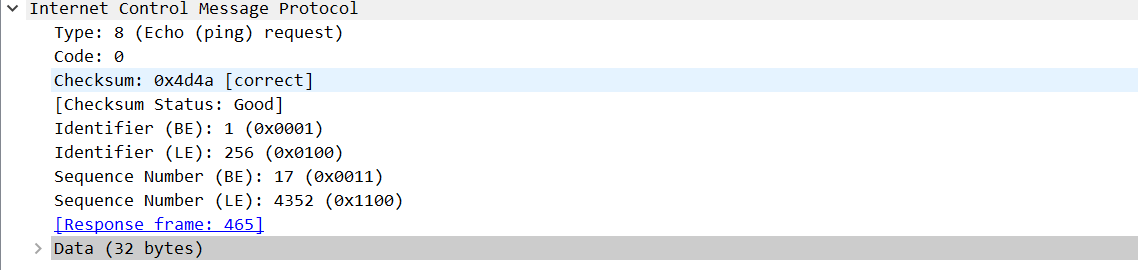


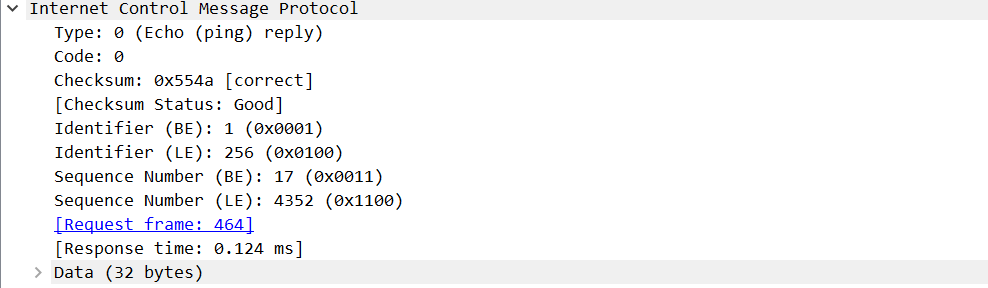




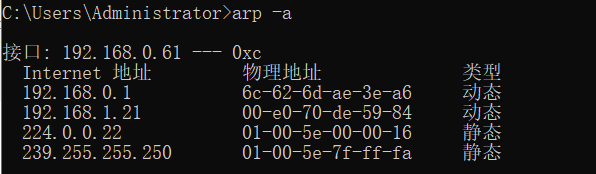
ICMP:







步骤2：执行“arp –a”命令，记录结果。



与前面的实验不同，没有云服务器IP对应的ARP表项

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 | 应答报文 |
| Ethernet II Dst: | ff:ff:ff:ff:ff:ff | 00:e0:70:e3:ff:56 |
| Ethernet II Src: | 00:e0:70:e3:ff:56 | 6c:62:6d:ae:3e:a6 |
| ARP Sender MAC address: | 00:e0:70:e3:ff:56 | 6c:62:6d:ae:3e:a6 |
| ARP Sender IP address: | 192.168.0.61 | 192.168.0.1 |
| ARP Target MAC address: | 00:00:00:00:00:00 | 00:e0:70:e3:ff:56 |
| ARP Target IP address: | 192.168.0.1 | 192.168.0.61 |

此ARP请求报文是用于获取192.168.0.1的mac地址，192.168.0.1是本机的默认网关，云服务器与本机不处于同一局域网中，所以在与路由表匹配中与默认路由匹配，对应的下一跳的目标IP为192.168.0.1，，ARP缓存中没有192.168.0.1 的表项，所以发出ARP请求

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: | 00:e0:70:e3:ff:56 | 本机 | 6c:62:6d:ae:3e:a6 | 本地网关 |
| IP Src: | 192.168.0.61 | 本机 | 20.231.127.246 | 服务器 |
| Ethernet II Dst: | 6c:62:6d:ae:3e:a6 | 本地网关 | 00:e0:70:e3:ff:56 | 本机 |
| IP Dst: | 20.231.127.246 | 服务器 | 192.168.0.61 | 本机 |

两台主机位于不同的局域网内，本机发出的ICMP报文会发往本地网关，所以Ethernet II Dst:为本地网关，通过NAT转换发送给目的ip，接收的ICMP报文也是，先由路由器接收，经过NAT转换发送给本机

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。

1. 相同网段内
2. 主机发送ARP请求广播包到本地网段,查询目标IP地址对应的MAC地址
3. 如果目标主机在同一网段,它就会回复一个ARP响应包,包含自己的MAC地址
4. 发送主机收到响应后就能缓存IP-MAC地址映射,用于后续通信
5. 不同网段内解析
6. 主机首先通过ARP查询默认网关的MAC地址
7. 发送主机将数据发送到默认网关
8. 网关根据路由表,将数据发送到下一条路径
9. 最终数据被转发到目标网段,然后在该网段内通过ARP解析目标主机MAC地址

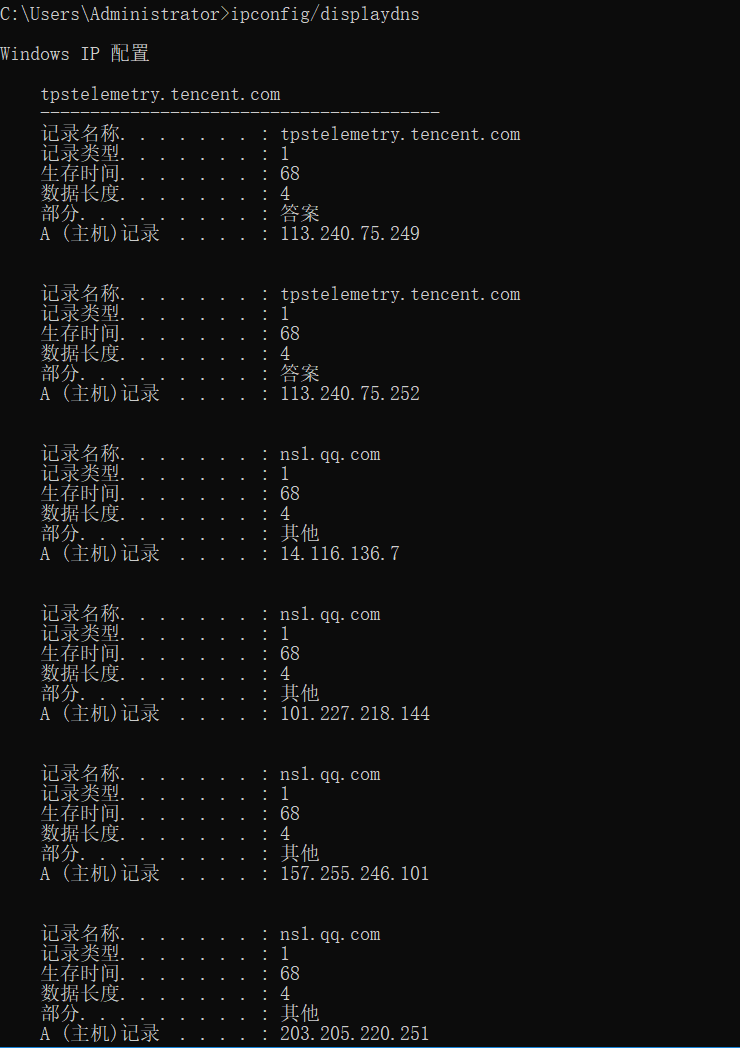
总的来说,在相同网段内ARP只需一次请求-响应过程;而不同网段需要先通过ARP获取网关MAC地址,然后通过网关转发到其它网段,最后在目标网段内再次使用ARP解析目标主机MAC地址。

* 1. DNS协议分析

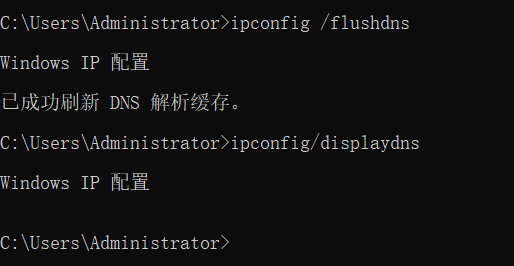
（一）默认DNS域名解析

步骤1：在命令窗口执行命令：

执行“ipconfig /displaydns”观察本地DNS缓存；



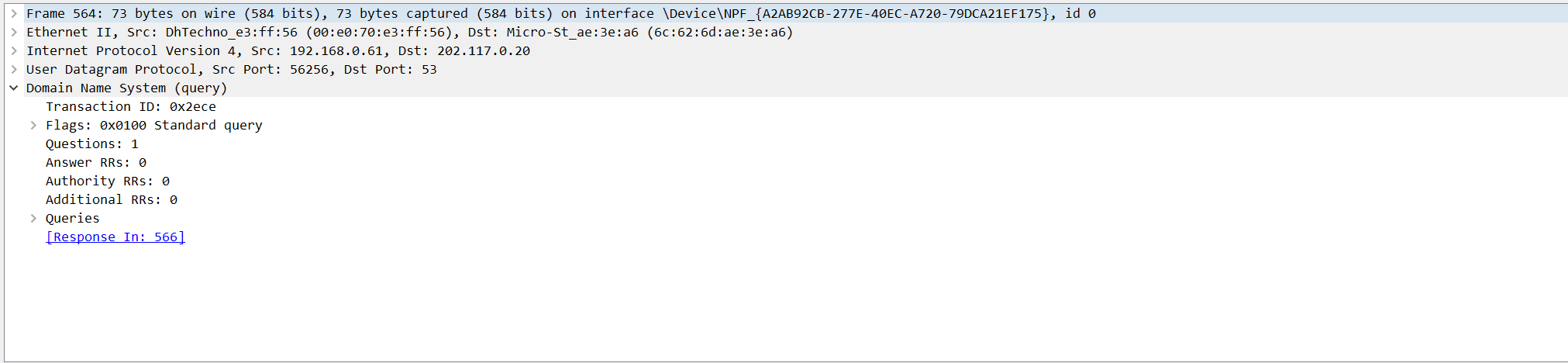
执行“ipconfig /flushdns”清除本地DNS缓存。

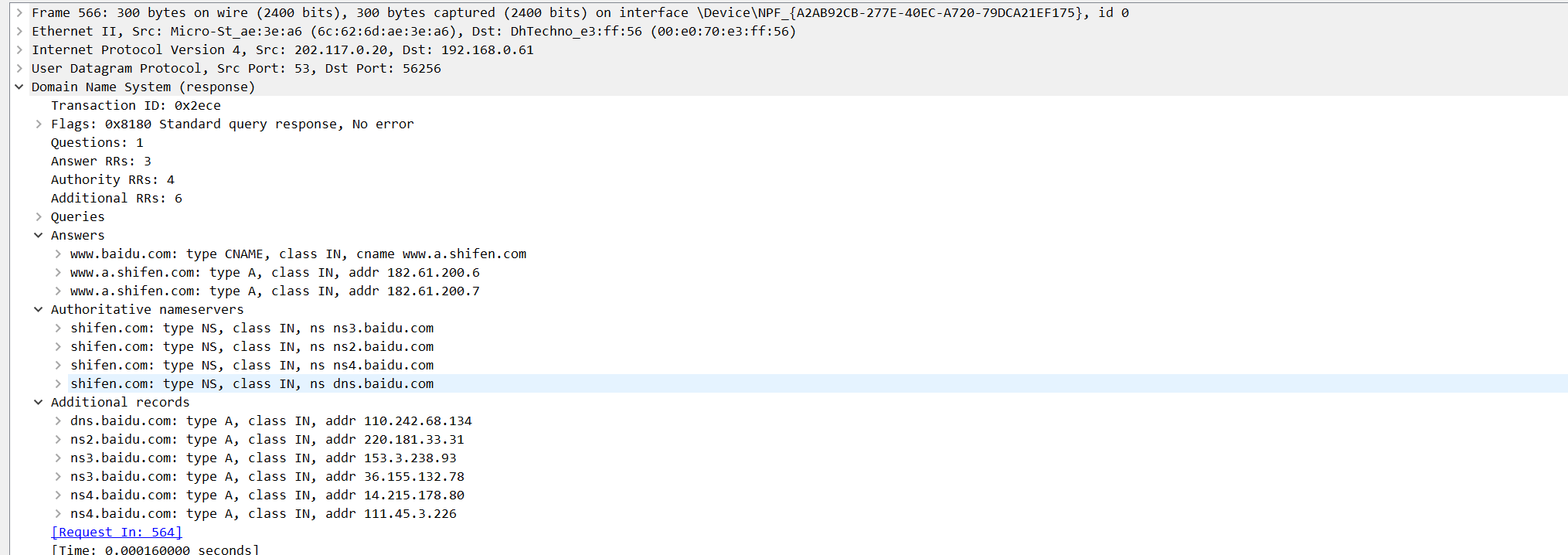


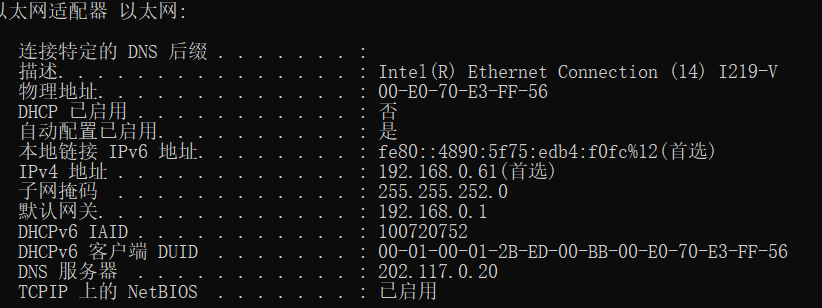
步骤2：在计算机终端上运行Wireshark截获报文，浏览器访问域名（如http://www.yahoo.com），网站打开后停止报文截获，观察分析DNS查询、回复报文分别包含哪些主要内容（UDP还是TCP、目的地址与本机默认DNS是否相同、源端口和目的端口、域名解析记录类型、解析出的IP地址等）。

1. DNS查询使用UDP协议
2. 查询的DNS服务器为202.117.0.20，与默认DNS一致，
3. 请求报文中：源端口：56256，目的端口：53，53是UDP协议中为DNS服务保留的端口号
4. 域名解析记录类型有A和HTTPS，A代表查询Address，HTTPS：当浏览器访问一个使用 HTTPS 协议的网站时，它会查找 HTTPS 记录以确定要使用的端口号。默认情况下，HTTPS 协议使用端口 443，但可以通过 HTTPS 记录指定不同的端口。HTTPS 记录可确保网站通过加密连接提供内容，从而保护用户数据免受窃听和篡改。
5. 解析出的IP地址为182.61.200.6和182.61.200.7









（二）指定DNS域名解析

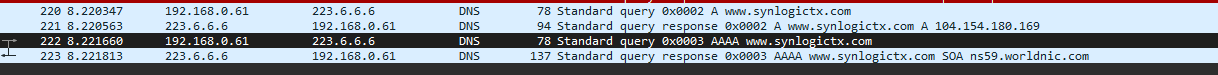
步骤1：在命令窗口执行命令：

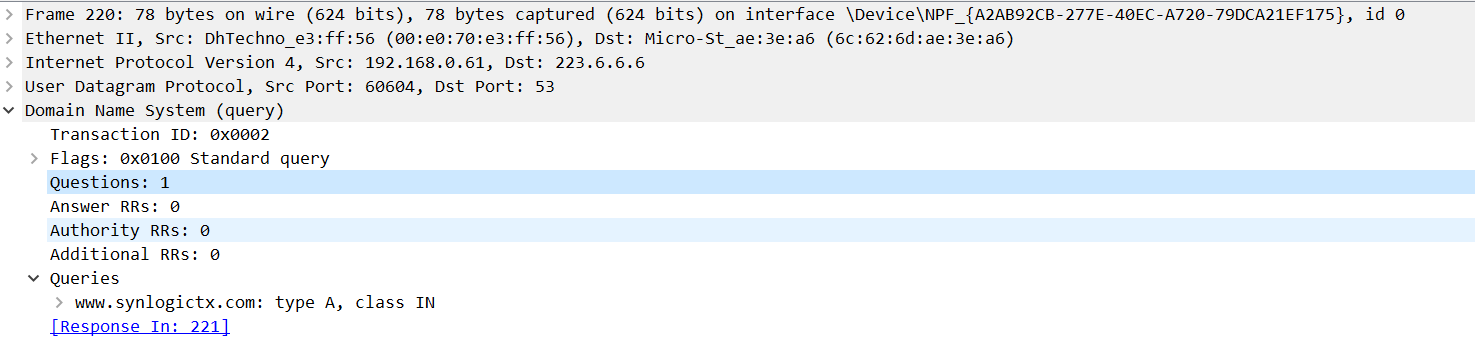
执行“ipconfig /displaydns”观察本地DNS缓存；

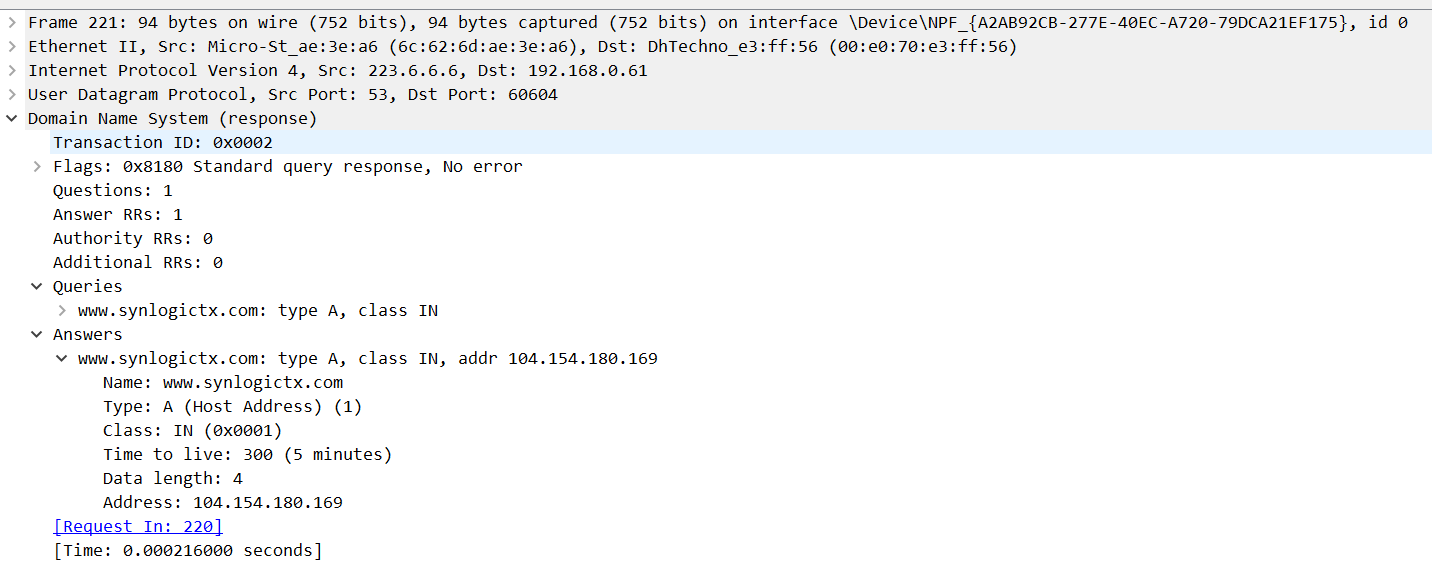
执行“ipconfig /flushdns”清除本地DNS缓存。

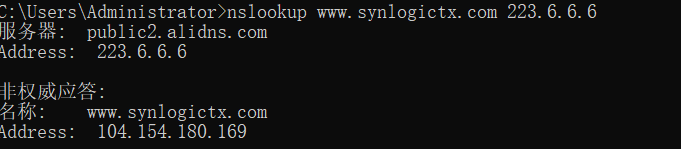
步骤2：在计算机终端上运行Wireshark截获报文，在命令窗口执行指定DNS服务器解析域名命令（如nslookup www.synlogictx.com 223.6.6.6），解析完毕后停止报文截获，观察分析DNS查询、回复报文分别包含哪些主要内容（UDP还是TCP、目的地址与本机默认DNS是否相同、源端口和目的端口、域名解析记录类型、解析出的IP地址等）。

1. DNS查询使用UDP协议
2. 查询的DNS服务器为223.6.6.6，与默认DNS不一致，变为指定的DNS服务器
3. 请求报文中：源端口：60604，目的端口：53，53是UDP协议中为DNS服务保留的端口号
4. 域名解析记录类型有A和AAAA，分别对应查询域名对应的IPv4地址和IPv6地址
5. 解析出的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；
2. ARP 和 ICMP 协议在不同的层工作。ARP 工作在链路层，而 ICMP 工作在网络层。在发送 ICMP 报文之前，主机需要先建立链路层连接，而这需要使用 ARP 协议。为此，为了确保 ICMP 报文能够正确传输，主机需要先使用 ARP 协议解析目标主机的 IP 地址到 MAC 地址
3. ARP的安全性问题；
4. ARP报文是采用广播的方式，进行查询，并且没有检验，所以任何机器都可以对此报文进行响应，所以攻击者可以送伪造的 ARP 响应报文，将自己的 MAC 地址与目标 IP 地址关联起来。这会导致收到响应的主机将数据包发送到攻击者的计算机，而不是目标主机。如果数据包没加密，这信息将全部被窃取，同时，攻击者可以作为中间人进行伪装，使得通信显得正常，能持续攻击不被发现。
5. 攻击者可以向网络中广播大量 ARP 请求报文，从而导致网络拥塞和主机无法正常通信。
6. DNS的欺骗带来的安全性问题；
7. 欺骗的思想是将正确的域名与错误的IP建立联系，将用户导向错误的IP地址
8. DNS 欺骗：攻击者通过劫持 DNS 服务器或修改 DNS 记录，在用户发出DNS查询的时候返回错误的ip地址。
9. DNS 缓存中毒：DNS服务器会缓存域名和IP地址的映射关系，当用户访问一个域名时，DNS服务器会首先查找缓存，如果找到了对应的IP地址，就会直接返回该IP地址。攻击者通过向 DNS 服务器发送伪造的 DNS 响应，将虚假 IP 地址与合法域名关联起来。
   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。