# **实验1-2报告模板**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 姓名： |  | 学号： |  | 班级： |  |
|  |  |  |  |  |  |

# **实验一 常用网络命令及工具**

## 1.1 实验目的

能够使用常用的网络命令完成PC机的特定网络功能查看、设置、测试等，能够使用常用的网络报文分析工具分析网络协议报文，能够使用Python（或其它）语言编写简单的TCP客户端程序。

## 1.2 实验内容

（1）常用网络命令练习；

（2）网络分析工具练习；

（3）简单的TCP客户端网络编程。

## 1.3 实验步骤

### 1.3.1 实验设备环境

每人一台PC，独立完成实验并提交实验报告。

【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。】

### 1.3.2 常用网络命令练习

步骤1：以命令行方式查看并记录本机的网络配置信息，查看本机共有几个网卡，哪些是物理网卡，哪些是虚拟网卡；用不同方法查看本机的公网ip地址，查看机房内其他PC公网地址，是否相同并分析原因。【参考命令：ipconfig /all、nslookup】

本机上网时用的是哪一个网卡，记录相关信息？

|  |  |
| --- | --- |
| 字段 | 配置值 |
| 以太网适配器 | 本地连接2 |
| 描述 | Red Hat VirtIO Ethernet Adapter |
| 物理地址 | 12-0.-FE-01-39-11 |
| DHCP已启用 | 是 |
| IPv4地址 | 192.168.1.182（首选） |
| 子网掩码 | 255.255.254.0 |
| 默认网关 | 192.168.0.1  1.2.4.8 |
| DNS服务器 | 202.117.0.20 |

获取公网地址方式：

* 使用nslookup myip.opendns.com resolver1.opendns.com或curl ifconfig.me等命令。
* 访问<https://ip138.com/>、<https://ipinfo.io/>等网站；
* 利用HFS软件的“IP地址->查找外部地址”、Wireshark抓包分析等。

|  |  |
| --- | --- |
| 方法 | 外网地址 |
| nslookup命令 | 208.67.222.222 |
| 访问<https://ip138.com/> | 123.138.191.140 |
| HFS软件 | 202.200.231.62 |
| …… |  |

不同方式获取的外网地址是否相同？如果不同请分析可能的原因。

不同

同一种方式获的机房内其他PC公网地址是否相同？分析原因。

相同

步骤2：用命令行修改本机IP地址和DNS服务器地址的获取方式（原来是自动获取方式则改为手动设置，原来为手动设置地址则改为自动获取）查看并记录网卡配置信息，与手动设置地址时的配置有什么不同（注意观察租约时间）？

【参考命令：

IP地址手动设置命令：netsh interface ip set address name="以太网 2" static 192.168.1.101 255.255.255.0 192.168.1.1；

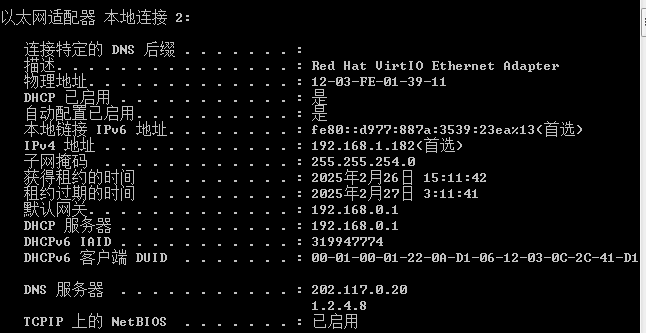
DNS服务器地址手动设置命令：netsh interface ip set dns name="以太网 2" source=static add=202.117.1.21；

IP地址自动获取命令：netsh interface ip set address name="以太网 2" source=dhcp；

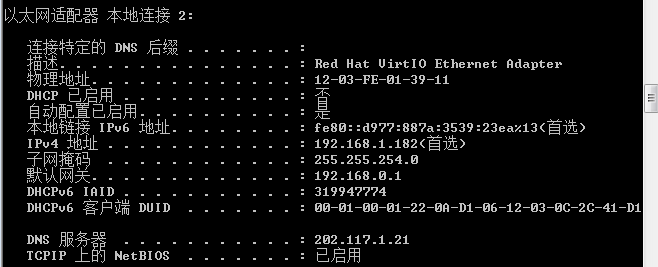
DNS服务器地址自动获取设置命令：netsh interface ip set dns name="以太网 2" source=dhcp。

（注意：实际参数根据步骤1记录进行修改，“以太网 2”改为“以太网适配器”配置值，IP地址/子网掩码/默认网关改为步骤1记录的配置值。连接手机热点上网时，手动设置地址可能会失败。）】

自动配置：



手动配置：



在手动配置以后，不再有租约时间相关的配置信息。

步骤3：查看并记录本机的IPv4路由表，标记出默认路由。用命令行删除默认路由，看看本机还能否上网并分析原因（如果还能上网，查看是否开启了IPv6，可禁用后再试）。查看网卡的默认网关配置是否还在？【参考命令：route print，route delete，ipconfig】

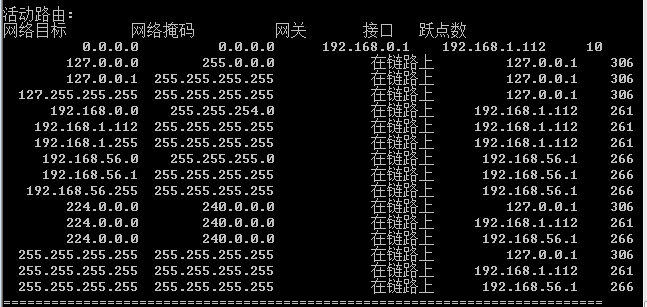
找到默认路由：



删除路由后，无法联网，默认网关配置也不存在：

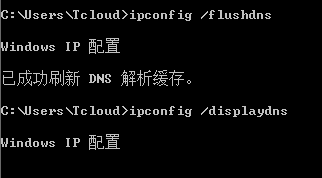


步骤4：分别用route add和route add -p增加一条默认路由，看看它们会出现在IPv4路由表的哪个部分，这两个路由表中的路由有什么不同？



步骤5：在命令行运行ipconfig /flushdns清除本地DNS缓存，用ipconfig /displaydns查看是否清空；ping通一个网址（如www.xjtu.edu.cn）后，再用ipconfig /displaydns查看本地DNS缓存，记录缓存的域名与IP地址。

清空本地DNS缓存：



Ping通以后的DNS缓存：



步骤6：把网卡的DNS服务器地址修改为无效DNS地址（无法ping通），分别ping域名和IP地址看能否ping通，查看本地DNS缓存，记录结果并分析原因。【参考命令：netsh interface ip set dns name="以太网 2" source=static add=192.168.0.254，配置后会提示“配置的DNS服务器不正确或不存在”】

Ping域名不通，ping IP地址能通。因为DNS服务器地址无效，因此ping域名时无法通过DNS协议将域名转换为IP地址。直接ping IP地址不需要用到DNS协议，可以直接实现网络寻址。



### 1.3.3 网络分析工具练习

步骤1：将网卡禁用后再启用，打开Wireshark软件抓包，能够正常上网后（打开网页、登录微信成功等）停止抓包。查看捕获的数据包及涉及到的协议，选择2种协议（如DHCP，ARP等，利用协议过滤筛选出该协议报文），分析协议的功能及关键交互数据。

|  |  |  |
| --- | --- | --- |
| 协议名 | 描述项 | 配置值 |
| *ARP* | *协议功能* | *IP地址对应MAC地址解析* |
| *源地址-目的地址* | *c2:4f:12:da:58:c1 - Broadcast* |
| *请求/应答信息* | *Who has 192.168.3.2? Tell 192.168.3.1* |
| DHCP | 协议功能 | 为客户机动态指定IP地址和参数配置 |
| 源地址-目的地址 | 192.168.3.1 - 192.168.3.2 |
| 请求/应答信息 | DHCP ACK Transaction ID 0x422c913e |
| ARP | 协议功能 | *IP地址对应MAC地址解析* |
| 源地址-目的地址 | Intel\_fd:8d:51 - c2:4f:12:da:58:c1 |
| 请求/应答信息 | 192.168.3.2 is at 94:e7:0b:fd:8d:51 |
| DHCP | 协议功能 | 为客户机动态指定IP地址和参数配置 |
| 源地址-目的地址 | 0.0.0.0 - 255.255.255.255 |
| 请求/应答信息 | DHCP Request - Transaction ID 0x422c913e |

步骤2：清除本机的DNS缓存【参考命令：ipconfig /flushdns】，运行Wireshark截获报文，浏览器访问网站（<https://www.python.org/>，浏览新闻，下载软件等），利用IP地址过滤筛选出访问该网站的报文，查看访问该网站时，都用到了哪些协议，主要作用是什么？【域名解析为IP地址方法：ping 域名，或nslookup 域名】

|  |  |  |
| --- | --- | --- |
| 协议名 | 描述项 | 配置值 |
| *TCP* | *协议功能* | *传输控制协议,在不可靠的互联网络上提供可靠的端到端传输。* |
| *源地址-目的地址* | *192.168.3.2 - 202.117.1.13* |
| *请求/应答信息* | *[SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK\_PERM* |
| TLS | 协议功能 | 数据加密和身份认证 |
| 源地址-目的地址 | 192.168.3.2 - 202.117.1.13 |
| 请求/应答信息 | Client Hello (SNI=www.xjtu.edu.cn) |
| HTTP | 协议功能 | 信息传输和资源共享，支持客户端-服务器交互 |
| 源地址-目的地址 | 192.168.3.2 - 202.117.1.13 HTTP2 |
| 请求/应答信息 | HEADERS[17]: GET /system/resource/js/counter.js |

### 1.3.4 TCP客户端编程

步骤1：编写TCP客户端程序。

步骤2：运行TCP客户端程序，连接到校园网内的202.117.10.144上TCP服务（端口8888）测试，进行信息交互。记录运行测试过程。

要求输入至少5行文字，并且第一行输入名字、学号和班级。

TCP客户端程序代码如下：

运行结果：

## 1.4 互动讨论主题

本地计算机接入网络之后，需要通过哪些设置、启用哪些协议之后才能上网。

首先要配置正确的IP地址（可以是自动获取DHCP或者手动设置静态IP）、子网掩码、默认网关和DNS服务器地址；需要启用网络协议如TCP/IP协议族（其中包含诸如IP协议、TCP协议、UDP协议等是上网所必需的）；如果是通过无线网络连接，要输入正确的Wi - Fi密码以连接到接入点。

# **实验二 ARP与DNS协议分析**

## 2.1 实验目的

能够分析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，能够分析DNS协议的工作过程，能编写简单的TCP服务器程序

## 2.2 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。

（4）编写简单的TCP服务器程序，并用上次的TCP客户端程序连接测试。

## 2.3 实验步骤

### 2.3.1 实验环境与分组

每人独立完成实验并提交实验报告。以现有的校园网络环境及外部服务器搭建内网、外网网络环境。

### 2.3.2 ARP协议分析

同一网段的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；

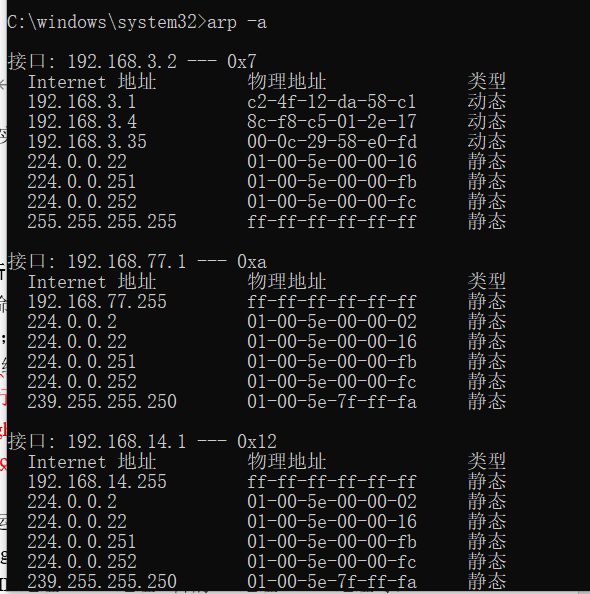
执行“arp –d”命令清空arp缓存。

注：如果arp命令无法运行，可使用以下命令代替：

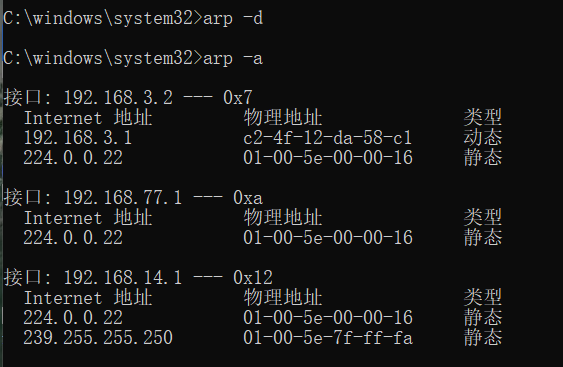
netsh interface IP show neighbors

netsh interface IP delete arpcache

查看arp缓存：



清除arp缓存后：

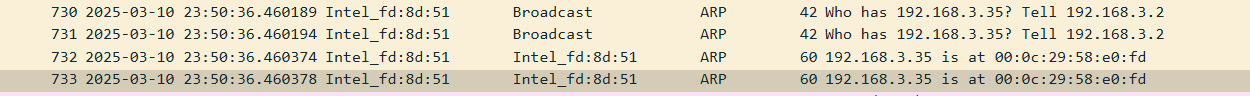


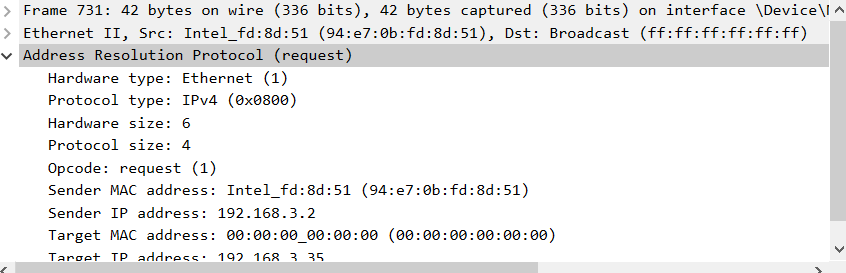
步骤2：在计算机终端上运行Wireshark截获报文，在命令行窗口ping同一网段的另一PC地址（不要ping网关）。执行完后停止报文截获，筛选出相关的arp和icmp报文进行分析（源IP地址/MAC地址、目的IP地址/MAC地址等）。

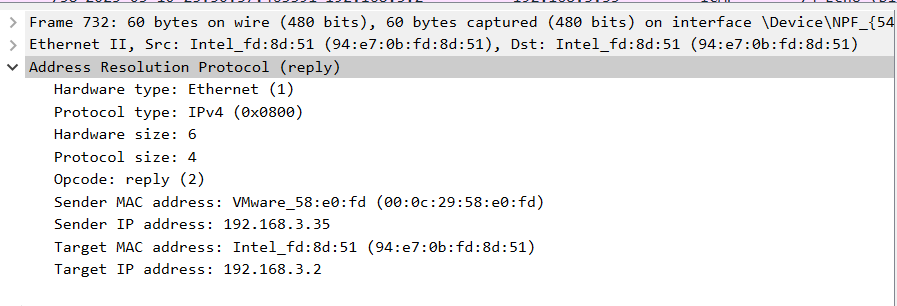
Arp请求和响应报文：

请求报文源MAC地址为Intel\_fd:8d:51即94:e7:0b:fd:8d:51，目的MAC地址为broadcast即ff:ff:ff:ff:ff:ff。请求报文中显示，想要查询的MAC地址属于的IP是192.168.3.35（Target IP Address）

响应报文源和目的MAC地址都是94:e7:0b:fd:8d:51，因为这里使用物理机与虚拟机进行通信，虚拟机网络适配器采用桥接模式。响应报文中给出了192.168.3.35虚拟机的虚拟网卡MAC地址，为00:0c:29:58:e0:fd



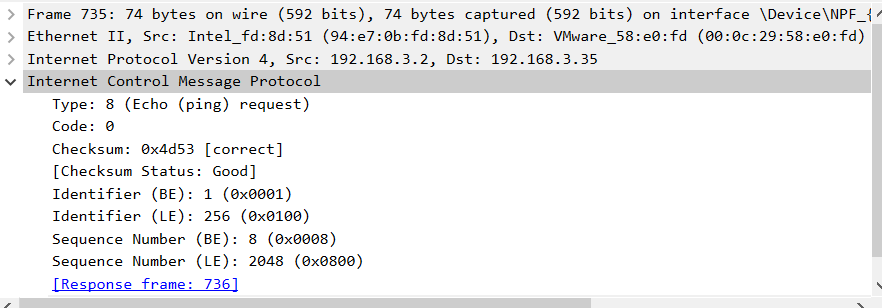




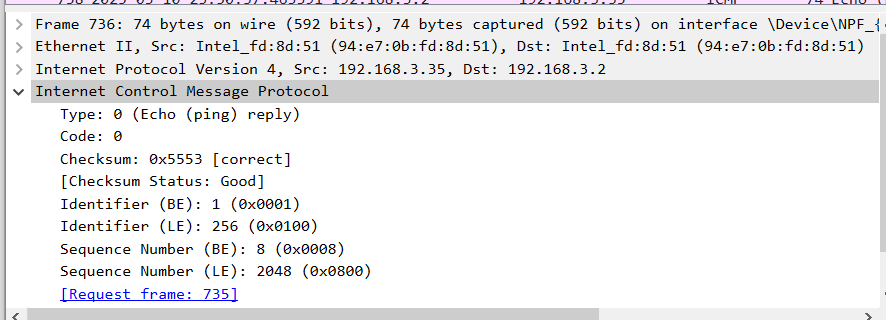
ICMP请求和响应：

请求报文源IP：192.168.3.2，源MAC地址94:e7:0b:fd:8d:51.目的IP和目的MAC地址为上面arp请求和响应查询到的虚拟机IP和对应MAC地址。

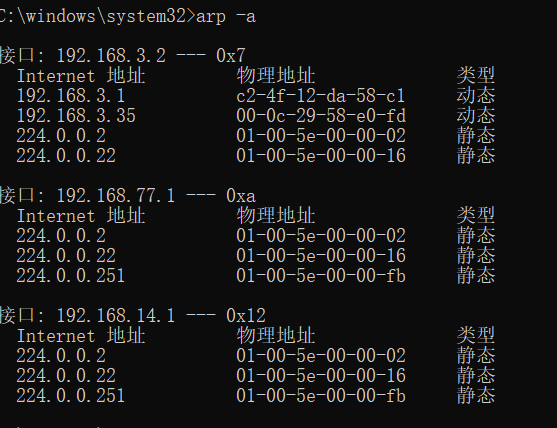




响应报文各层协议的源地址和目的地址在IP层与请求恰好相反。MAC层地址由于本机上运行虚拟机网络适配器用桥接模式，因此看到的源地址和目的地址都是物理机的MAC地址。



步骤3：在命令行窗口执行“arp –a”，记录结果。

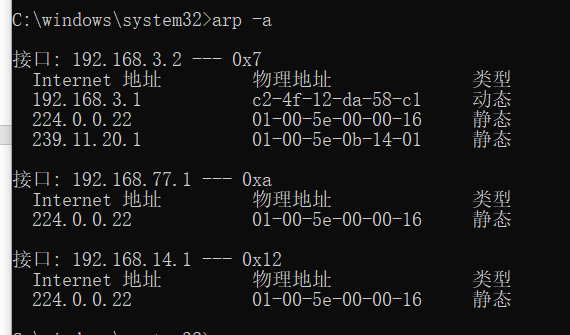


不同网段的ARP协议分析：

步骤1：在本地计算机运行Wireshark捕获报文，执行“arp –d”清空缓存，在本地计算机ping外网服务器域名/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）。

【网卡会自动解析默认网关的MAC地址，如果抓到的数据包中ARP与ICMP数据包间隔较远，可以删除默认网关设置，添加外网路由后，再重复步骤1。参考命令：route delete 0.0.0.0， route add 外网服务器地址 MASK 255.0.0.0 本地网关地址】

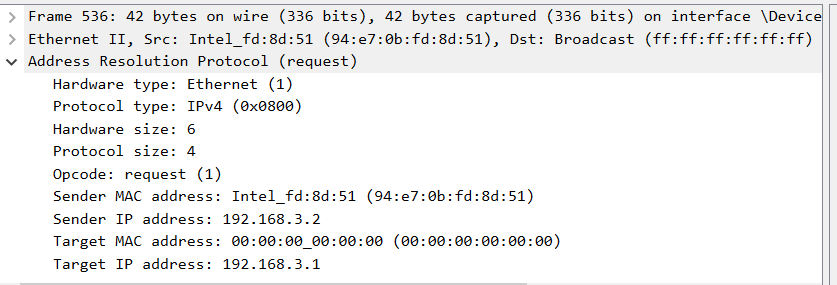
步骤2：执行“arp –a”命令，记录结果。



步骤3：分析捕获的报文，选中**第一对**ARP请求（**广播报文**）—应答报文，填写下表。



请求报文：



应答报文：

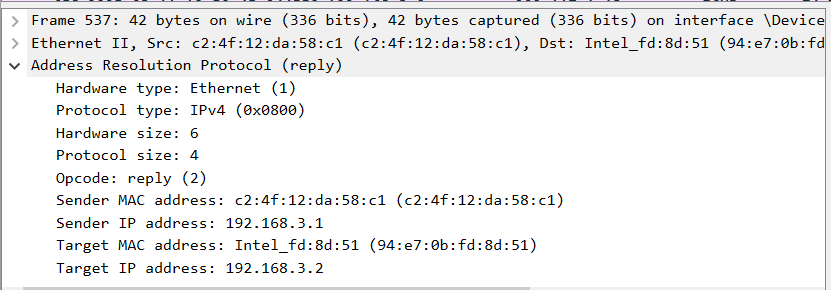


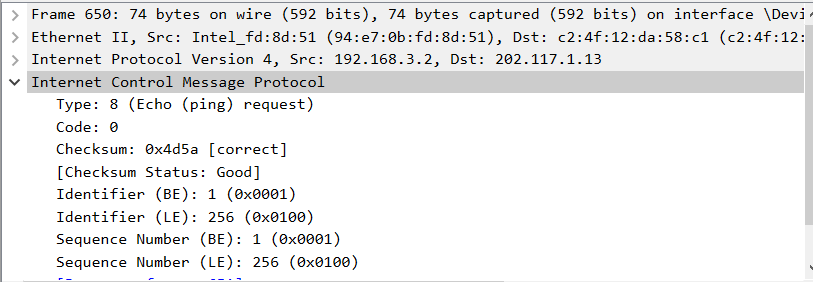
表2-1 ARP请求报文和应答报文的字段信息

|  |  |  |
| --- | --- | --- |
| 字段 | 请求报文 | 应答报文 |
| Ethernet II Dst: | Broadcast(ff:ff:ff:ff:ff:ff) | 94:e7:0b:fd:8d:51 |
| Ethernet II Src: | 94:e7:0b:fd:8d:51 | c2:4f:12:da:58:c1 |
| ARP Sender MAC address: | 94:e7:0b:fd:8d:51 | c2:4f:12:da:58:c1 |
| ARP Sender IP address: | 192.168.3.2 | 192.168.3.1 |
| ARP Target MAC address: | 00:00:00:00:00:00 | 94:e7:0b:fd:8d:51 |
| ARP Target IP address: | 192.168.3.1 | 192.168.3.2 |

分析捕获的报文，选中一对ICMP请求—应答报文，填写下表。（**对应主机**填写**本机、本地网关、外网服务器**）



请求报文：



应答报文：

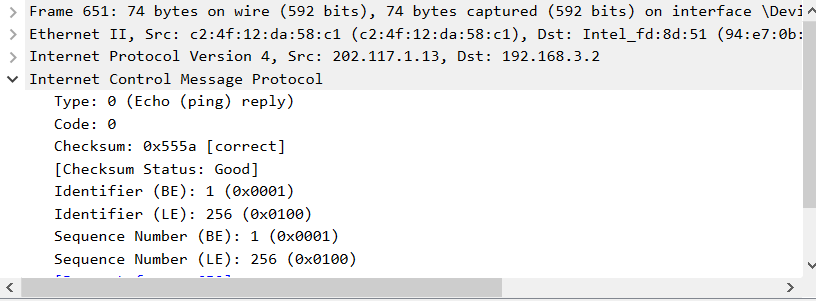


表2-2 ICMP请求报文和应答报文的字段信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 请求报文 | 对应主机 | 应答报文 | 对应主机 |
| Ethernet II Src: | 94:e7:0b:fd:8d:51 | 本机 | c2:4f:12:da:58:c1 | 本地网关 |
| IP Src: | 192.168.3.2 | 本机 | 202.117.1.13 | 外网服务器 |
| Ethernet II Dst: | c2:4f:12:da:58:c1 | 本地网关 | 94:e7:0b:fd:8d:51 | 本机 |
| IP Dst: | 202.117.1.13 | 外网服务器 | 192.168.3.2 | 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。

在相同网段内，主机之间进行通信时，发送方主机如果不知道目的主机的MAC地址，就会发送ARP请求广播，这个广播会在本网络段内传播，网络段内的所有主机都会收到这个请求，但只有目的主机才会回复自己的MAC地址给发送方主机。

而在不同网段内，源主机需要通过ARP来获取默认网关的MAC地址，发送ARP请求获取网关MAC地址的过程和同一网段内获取其他主机MAC地址的过程相同，都是通过广播请求得到相应回复。不同之处在于，同一网段是主机与主机之间的MAC地址解析，而不同网段是主机与网关之间的MAC地址解析，数据包也是先送到默认网关。

### 2.3.3 DNS协议分析

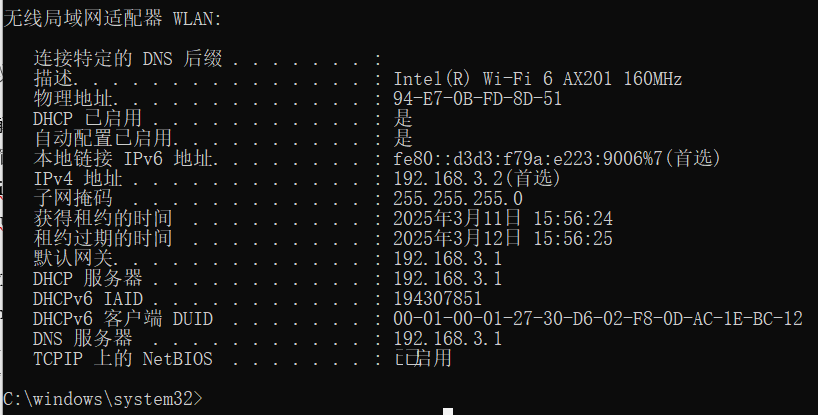
默认DNS域名解析：

步骤1：在命令窗口执行命令：

执行“ipconfig /displaydns”观察本地缓存；

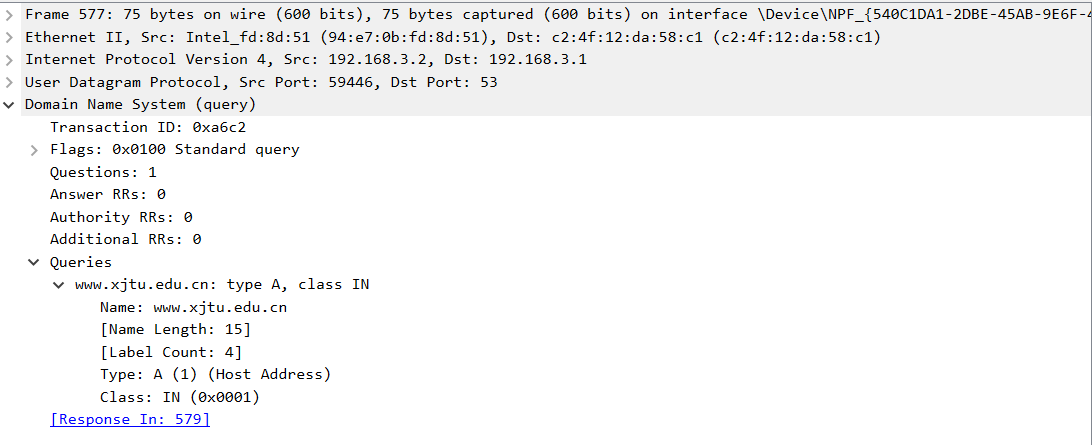
执行“ipconfig /flushdns”清除本地缓存。

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、域名解析记录类型及含义、解析出的IP地址等）。

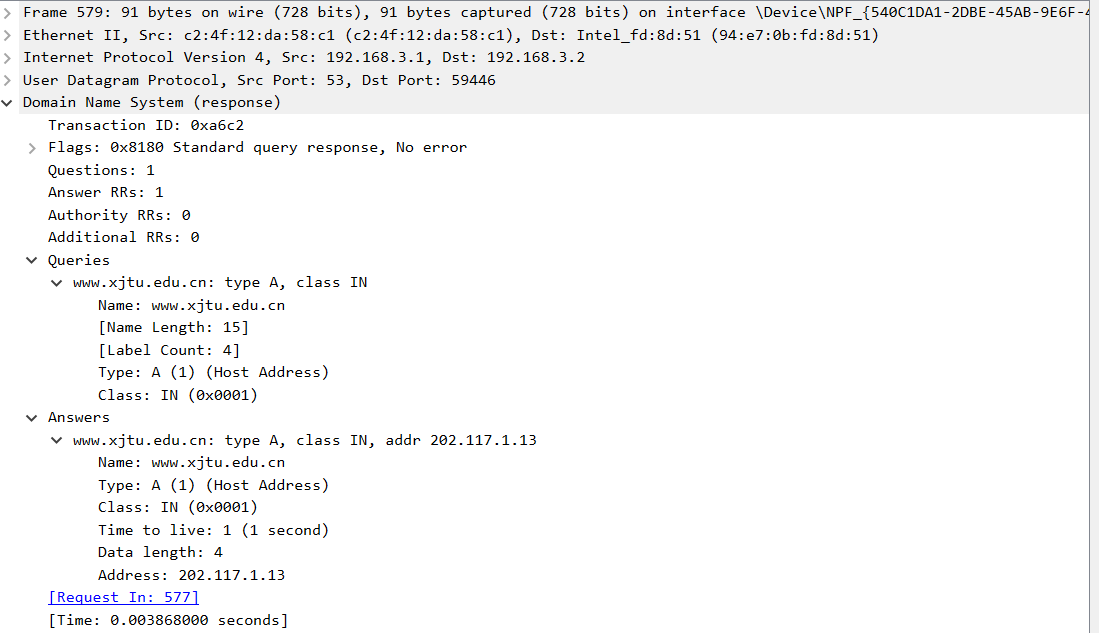


本机默认DNS为192.168.3.1

DNS查询报文：传输层使用UDP协议，源地址为本机，目的地址为本机默认DNS，源端口为59446，目的端口53，A记录，请求解析的域名为www.xjtu.edu.cn



DNS回复报文：传输层仍然使用UDP协议，源地址为本机默认DNS，目的地址为本机。源端口、目的端口与请求报文相反，A记录，解析出域名www.xjtu.edu.cn的IP地址是202.117.1.13



DNS域名解析记录类型：

A记录：将域名指向一个IPv4地址；

CNAME记录：将域名指向一个域名，实现与被指向域名相同的访问效果，这个域名一般是主机服务商提供的一个域名；

MX记录：建立电子邮箱服务，将指向邮件服务器地址；

NS记录：域名解析服务器记录，如果要将子域名指定某个域名服务器来解析，需要设置NS记录；

TXT记录：可任意填写，可为空。一般做一些验证记录时会使用此项，如：做SPF（反垃圾邮件）记录；

AAAA记录：将主机名（或域名）指向一个IPv6地址；

SOA记录：SOA叫做起始授权机构记录，NS用于标识多台域名解析服务器，SOA记录用于在众多NS记录中那一台是主服务器；

PTR记录：PTR记录是A记录的逆向记录，又称做IP反查记录或指针记录，负责将IP反向解析为域名；

显性URL转发记录：将域名指向一个http(s)协议地址，访问域名时，自动跳转至目标地址。

隐性URL转发记录：将域名指向一个http(s)协议地址，访问域名时，自动跳转至目标地址，隐性转发会隐藏真实的目标地址。

指定DNS域名解析：

步骤1：在命令窗口执行命令：

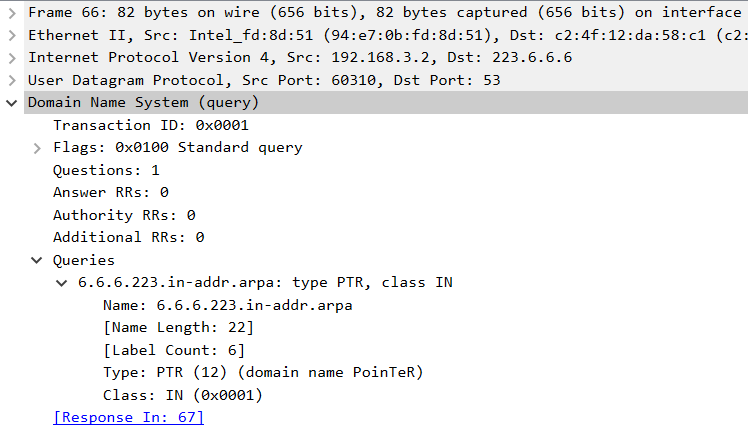
执行“ipconfig /displaydns”观察本地缓存；

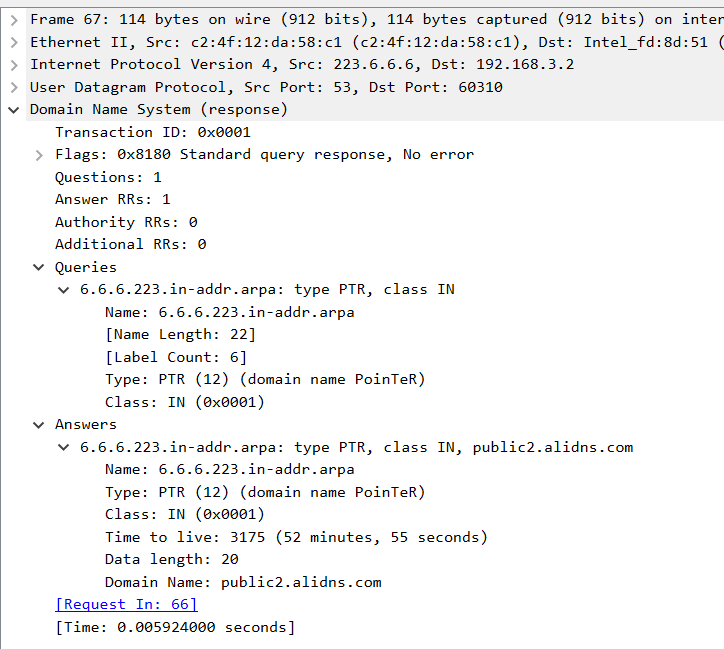
执行“ipconfig /flushdns”清除本地缓存。

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.github.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、注意**PTR**类型的作用、解析出的IP地址等）。

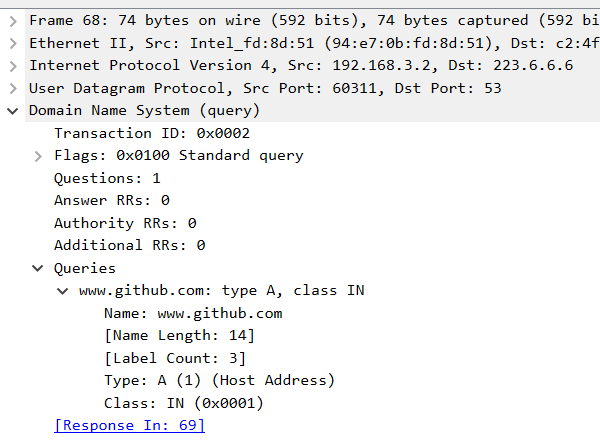
共有三组请求和应答报文，请求源IP为本机192.168.3.2，目的IP为223.6.6.6（不是本机默认DNS），应答报文恰好相反，传输层协议都是UDP。本机通信端口为53，而DNS服务器通信端口依次为60310、60311、60312.

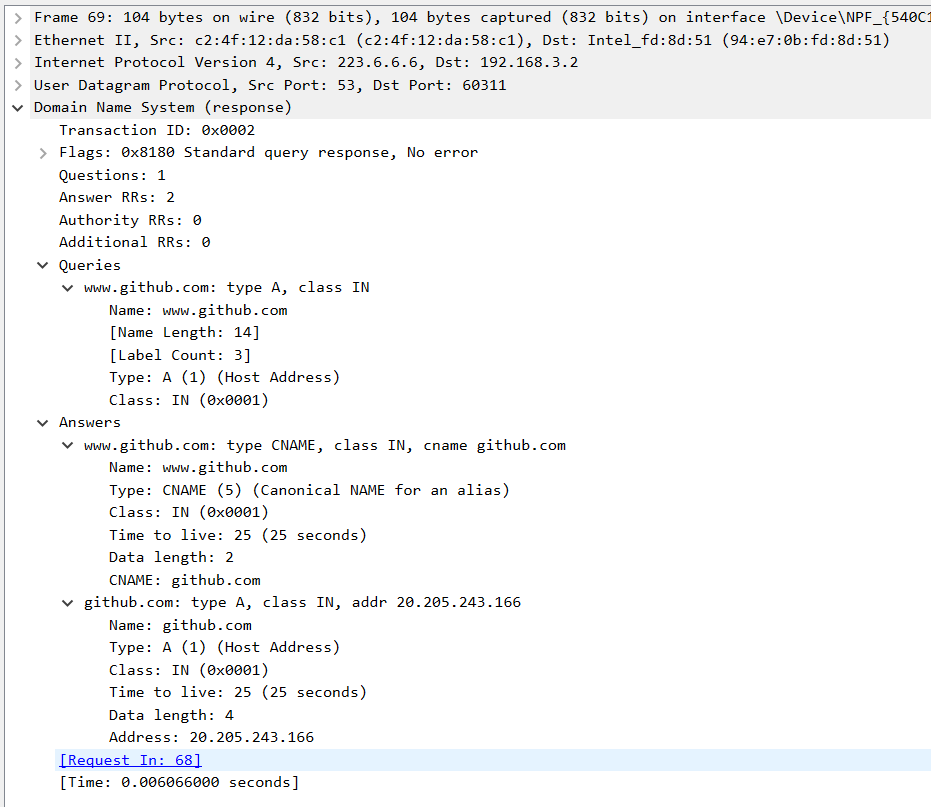
第一对请求应答报文：请求是PTR类型，因此应答将IP地址223.6.6.6反向解析为域名public2.alidns.com。



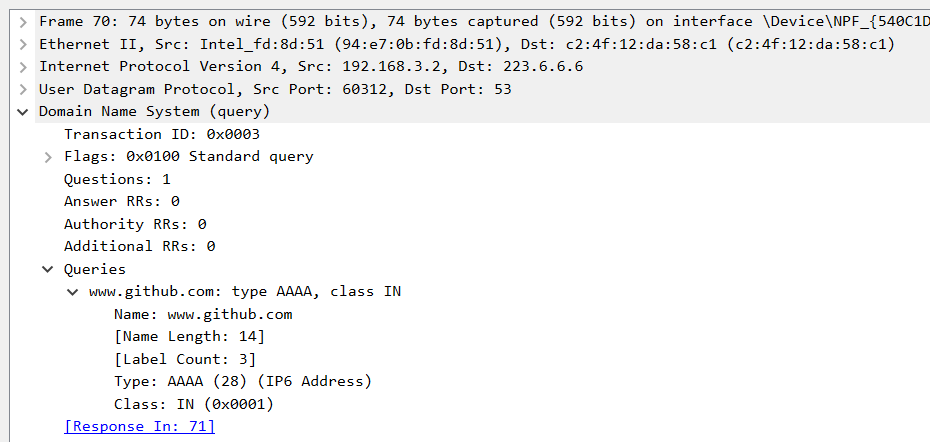


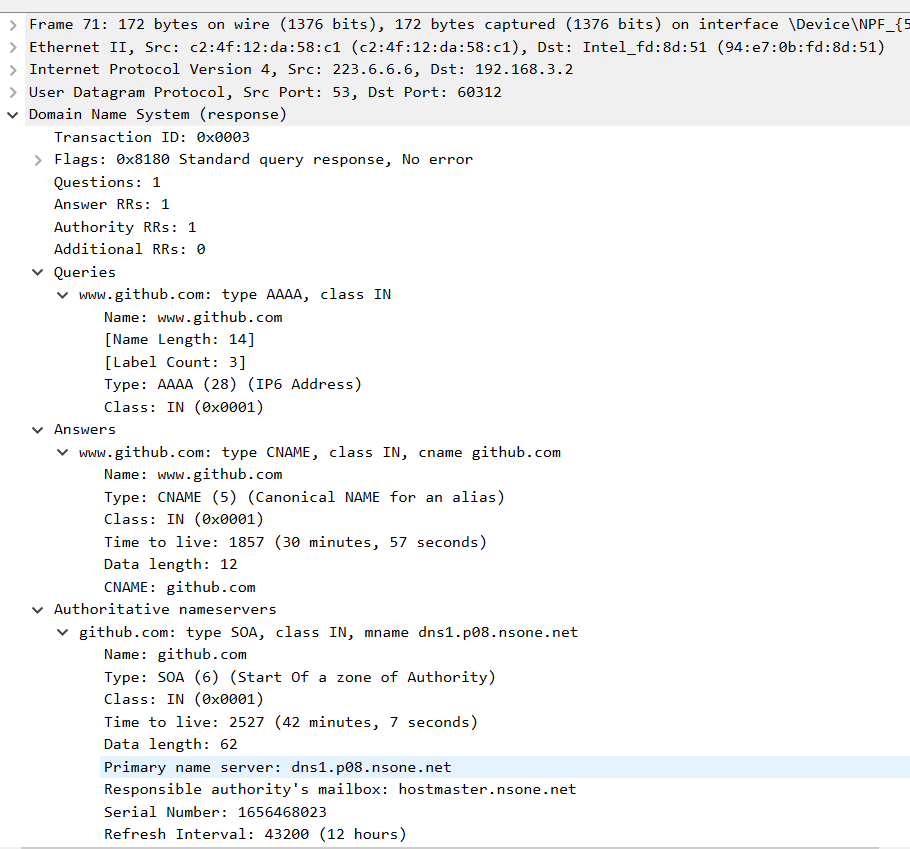
第二对：请求是A类型，应答中不仅给出了这个域名的一个别名github.com，而且解析出对应的IP为20.205.243.166





第三对：请求是AAAA类型，希望解析为IPv6地址，但是应答为SOA类型，给出起始授权记录。推测我们指定的DNS服务器不能将该域名解析为IPv6地址。





### TCP服务器编程

（1）编写简单的TCP服务器程序，比如可以把客户端发送来的消息原样返回给客户端；

（2）在一台机器上运行该TCP服务器程序，在另外一台机器上运行简单的TCP客户端程序进行测试。

TCP客户端程序类似实验一。在Ubuntu虚拟机上编写TCP服务器程序：



和客户端程序配合，客户端输入姓名学号班级以及一些其他的信息，服务器建立连接后提示对方IP，之后将客户端的信息回显给客户端。运行结果：

客户端：



服务器：

## 2.4 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；

在正常的网络通信中，发送方先发送ARP请求是为了获取接收方的MAC地址，因为在以太网中数据帧的传输需要知道目的MAC地址。如果发送方不知道接收方MAC地址，它会以广播形式发送ARP请求，接收方收到后会回复自己的MAC地址。只有在得到接收方MAC地址后，发送方才会构建并发送包含ICMP报文等实际通信数据的以太网帧。如果发送方已经知道接收方MAC地址（可能之前有过通信，ARP表项还未过期等情况），那么发送方可以直接发送ICMP报文，ICMP报文的出现次序取决于MAC地址是否已知等因素。本实验在ping之前都清空了arp缓存，因此先出现arp报文再出现ICMP报文。

（2）ARP的安全性问题；

ARP欺骗：攻击者可以发送虚假的ARP消息，将自己的MAC地址与目标IP地址关联起来。这样，网络中的其他设备会将发往目标IP的流量发送给攻击者，导致中间人攻击。攻击者可以在不被察觉的情况下窃取敏感信息，如登录凭证、信用卡信息等。

ARP缓存中毒：通过不断发送虚假的ARP消息，攻击者可以让目标设备的ARP缓存中存储错误的MAC - IP映射关系。这会影响网络的正常通信，导致数据传输错误或中断。

缺乏身份验证机制：ARP协议本身没有对发送ARP消息的设备进行身份验证，任何设备都可以发送ARP消息，这使得攻击者容易进行伪造。

广播特性的滥用：ARP使用广播来发现MAC地址，这使得攻击者可以监听广播并获取网络中的MAC - IP映射关系信息，从而为攻击提供便利

（3）DNS的欺骗带来的安全性问题；

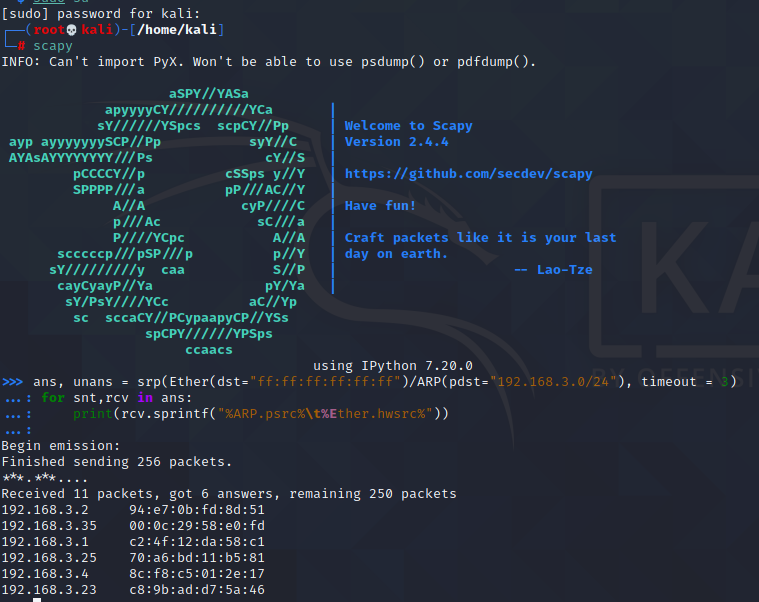
信息窃取，攻击者可将用户引导至恶意网站，从而获取用户在该网站输入的账号、密码等敏感信息。其次是网络访问控制失效，比如将正常的网站流量导向恶意的服务器，绕过企业或组织设置的网络访问限制，可能使内部网络遭受攻击。再者，可能影响整个网络的通信正常性，被欺骗的DNS服务器可能返回错误的IP地址，导致大量的网络连接失败或者被导向错误的目的地。

## 2.5 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。

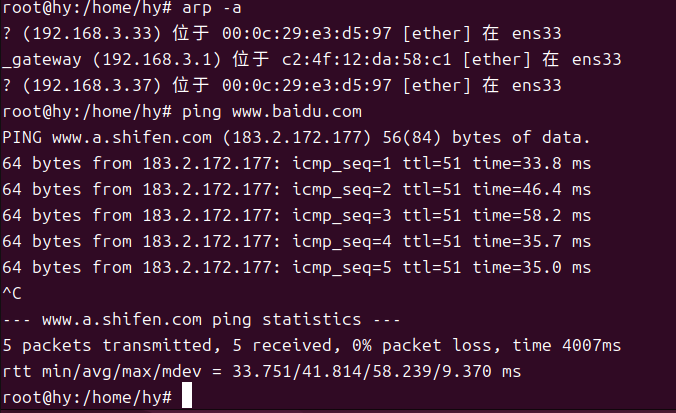
（1）使用scapy编程，探测本局域网内主机的IP及ARP对应关系，并输出IP/MAC对应表。

在kali Linux系统中使用scapy，使用交互式的方法，输入代码指令：

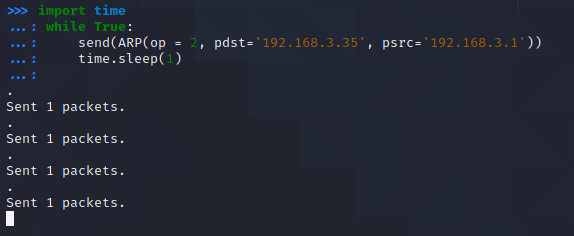


（2）使用scapy在Linux下写程序来模拟完成一个简单的ARP欺骗。

Kali虚拟机进行ARP欺骗，Ubuntu虚拟机作为靶机（192.168.3.35）。欺骗前，Ubuntu虚拟机ARP缓存正常，可以与外网通信：



进行ARP欺骗：持续向靶机发送ARP应答报文，使得靶机误将本机当作网关。



靶机ARP缓存异常，不能与外网通信。

