小实验 2 实验指导书

1、实验要求

本次小实验由 4 个抓包观察实验(第二部分)和 4 个简述题(第三部分)构成。所有抓包实验都需要给出简短的实验报告，内容包括：

(1)文字回答：回答指导书中提出的问题。要求简短，每个实验回答总计不宜超过 100 字。有参考答案的题目可不做文字回答。

(2)过程截图：展示关键实验过程，也可作为文字回答的补充。若无特别要求，则每个实验至少一张截图，展示实验过程。若有具体截图要求([]为要求)，则按照[]中要求给出对应截图。

注意 1：本次小实验中，抓包实验 1（4）、抓包实验 4 为选做实验，不完成不扣分，完成也没有额外的加分

2、实验题目及指导

抓包实验 1：观察以太网帧实验

依次执行以下操作：

- 开启 Wireshark 捕获数据帧
- 利用 arp -d 命令删除主机上的 ARP 表项（只删除网关亦可）
- 用浏览器浏览网页，触发 ARP 过程

注 1：windows 用户注意用管理员权限打开 CMD

注 2：macOS 用户删除主机所有 ARP 表项的命令是 sudo arp -a -d

观察 Wireshark 中捕获的数据包，并回答以下问题。

（1）观察捕获到的数据包，并结合课上介绍的经典以太网帧格式。Wireshark 捕获的数据包含哪些 MAC 层字段？不包含哪些 MAC 层字段？**[截图展示以太网帧头]**

参考答案：包含目的地址，源地址，类型，数据字段。不包含前同步码，帧开始定界符，校验和。

（2）找到捕捉的数据帧中由实验主机发出的 ARP 请求帧，辨认其目的地址和源地址域。它的目的 MAC 地址是多少？再用 ipconfig -all 命令查看实验主机的 MAC 地址，看看是否和该帧中的源地址一致。**[截图展示该 ARP 请求帧和实验主机 MAC 地址]**

参考答案：目的 MAC 地址为广播地址 FF:FF:FF:FF:FF:FF，源 MAC 地址与实验主机一致。（注意题目关注的是以太网头的 MAC，而不是 ARP 头中的 MAC）
注：macOS 用户查看 MAC 地址的命令是：ifconfig -a

（3）封装 IP 分组和 ARP 分组的帧有不同的类型字段号，他们分别是？**[截图展示 ARP 分组帧类型号和 IP 分组帧类型号]**

参考答案：0x0806（ARP），0x0800（IPv4）

（4）目的地址为实验主机的数据帧中，长度最长的是多大？长度最短的是多大？为什么？**（选做）**

参考答案：最长 1514 字节，最短 60 字节。最长的情况：6 字节目的地址+6 字节源地址+2 字节类型+1500 字节数据，不包含 4 字节的校验和，满足最大 1518 字节的标准。最短的情况：数据字段过短，数据帧被 PADDING 填充至 60 字节，不包含 4 字节的校验和，满足最小 64 字节的标准。

注 1：需以目的地址为实验主机过滤，发送的帧在被 wireshark 捕获时还未经过网卡 PADDING，可能不满足最小帧长

注 2：网卡，路由器等设备的配置均可能影响观察到的最大最小帧长。由于硬件区别难以避免，故作为选做。

注 3：生成最大帧长的数据包：用 ping -l 3000 www.baidu.com（Linux 和 macOS 为 -s 选项）。生成最小帧长数据包：按抓包实验 1 描述的方法构造 ARP 数据包

抓包实验 2：观察通常的 IPv4 分组

- 浏览网页，观察过程中捕获的 IPv4 报文。

（1）Version 字段的值是多少？IHL 字段的值一般是多少？结合 IPv4 分组头部格式可以看出 IHL 的单位是什么？

参考答案：Version 字段的值是 4。IHL 字段的值一般是 5，IHL 的单位是 4 字节。

（2）TCP、UDP、ICMP 对应的 Protocol 值分别是什么？ **[截图展示 TCP，UDP，ICMP 类型 IPv4 分组的 Protocol 值]**

参考答案：0x06，0x11，0x01

- 用 tracert /traceroute 探测 www.baidu.com

注 1：linux 和 MacOS 环境使用的命令是 traceroute，window CMD 使用的是 tracert。

注 2：tracert 使用 ICMP 发送探测，traceroute 使用 UDP 发送探测

（3）观察 tracert 程序发送的分组中 TTL 字段，它们有何特点？ **[截图展示 TTL=0,1,2,3 的数据包各一个]**

参考答案：TTL 值从 1 开始递增

- 用 ping -f 命令（强制不分段）探测 www.baidu.com。

（4）Windows 操作系统中初始的 TTL 值多为 128，而 Linux 的多为 64。利用这一点大体判断一下 www.baidu.com 运行哪一类操作系统，以及分组到达 www.baidu.com 之前经过了多少个路由器。

注 1：linux 环境强制不分段命令为 ping -M do www.baidu.com

注 2：MacOS 环境强制不分段命令为 ping -D www.baidu.com

（5）用 ping -f 发送的分组的 DF 值是多少？**[截图展示分组 DF 值]**

参考答案：DF = 1

抓包实验 3：观察 IPv4 分段与重组

依次执行以下操作：

- 开启 Wireshark 捕获数据帧；
- 生成数据长度超过以太网最大长度的数据包，触发 IPv4 分段（ping -l 3000 www.baidu.com）

注 1：Linux 和 MacOS 下的命令是 ping -s 3000 www.baidu.com

注 2：可能出现收不到 reply 的情况，观察 request 包结构即可。

注 3：不要用 ICMP 过滤数据包，前两个分组会被识别为 IPv4

观察 Wireshark 中捕获的数据包，并回答以下问题。

（1）三个分组的 Identification 值是多少？是否相等？ **[截图展示三个分组的 Identification 值]**

参考答案：都为 0xXXXX。相等。

（2）前两个分组的 Flag 字段，DF 和 MF 的值分别是？表示什么意思？**[截图展示前两个分组的 DF，MF 值]**

参考答案：DF 值为 0，MF 值为 1。还有更多分段。

（3）第三个分组的 Flag 字段，DF 和 MF 的值分别是？表示什么意思？**[截图展示第三个分组的 DF，MF 值]**

参考答案：DF 值为 0，MF 值为 0。表明这已经是最后一个分段。

（4）三个分组 Fragment offset 的值依次为多少，以确保在乱序到达时也能正确重组出原来的分组？ **[截图展示三个分组的 offset]**

参考答案：0、1480 字节(包中值 185)、2960 字节(包中值 370)

（5）三个分段的总的数据长度为 1500+1500+68-3*20=3008，比 ping 命令中的参数 3000 多了 8，为什么？

抓包实验 4：观察 IP 选项的使用(选做)

依次执行以下操作：

- 开启 Wireshark 捕获数据帧
- CMD 中运行 ping -r 8 www.baidu.com

注 1：Linux 环境命令为 ping -R www.baidu.com。（Linux 中默认为记录 9 跳，与标答 8 跳结果不同）

注 2：MacOS 环境 ping 命令不能实现-R 操作

观察 Wireshark 中捕获的数据包，并回答以下问题。

（1）在 ping 请求分组中，IPv4 选项 code 值为多少？表示什么？

参考答案：0x07，表示记录路径。

（2）在 ping 请求分组中，len 值为多少？表示可以记录多少条 IPv4 地址？当前指针的值是多少？表示下一条记录是第几条记录？

参考答案：0x23，即 35。8 条，因为 3+8*4=35。当前指针的值是 4。下一条记录是第 1 条，因为 4/4=1

（3）在 ping 应答分组中，分组经过路径上 8 跳内的路由器出口地址被记录下来，当前指针的值是多少？表示下一条记录是第几条？

参考答案：当前指针的值是 36。下一条记录是第 9 条，因为 36/4=9

Wireshark 过滤语句参考

（1）IP 包目的地址过滤（ip.dst == x.x.x.x）

（2）IP 包源地址过滤(ip.src == x.x.x.x)

（3）ICMP 协议过滤（icmp）

（4）ARP 包 IP 地址字段过滤(arp.dst.proto_ipv4 == x.x.x.x or arp.src.proto_ipv4 == x.x.x.x)

3、简述题

（1）实验过程中，如果选择由实验主机发出的数据帧，则会发现帧长度最小 42 字节，不满足最小帧长要求。试分析这种帧的各个域，并解释这一现象。

（2）上网查找资料，看看除了 IP 和 ARP 之外，还有哪些 IEEE 802.3 协议支持的网络层分组类型，类型编码分别是什么？

（3）什么情况下 IPv4 分组需要分段？在哪里分段？又是在哪里重新组装起来的？

（4）拒绝服务（Denial of Service，DoS）攻击，通过消耗目标主机设备的某种资源，导致其网络服务不能被正常用户使用。IP 数据报分片机制可能被攻击者利用来构建拒绝服务攻击。试设计一种利用 IP 数据报分片机制发动 DOS 攻击的方法，并提出防御的思路。

4、参考资料

以太网帧结构：网络学堂 CH05-01.pdf,44 页

RFC 791(IPv4 报文格式)： https://www.rfc-editor.org/rfc/rfc791