电子科技大学信息与软件工程学院

实 验 报 告

学　　号 2018091618008

姓　　名 袁昊男

（实验）课程名称 网络安全攻防技术

理论教师 王瑞锦

实验教师 王瑞锦

电子科技大学教务处制表

电 子 科 技 大 学

实　　验　　报　　告

学生姓名：袁昊男　　学号：2018091618008　　指导教师：王瑞锦

实验地点：信软楼 306　　实验时间：2020.10.21

一、实验室名称：信息与软件工程学院实验中心

二、实验名称：网络侦听实验

三、实验学时：2 学时

四、实验原理：

（一）地址解析协议（ARP）实验

本实验中，所有计算机位于一个物理网络中：所有计算机通过以太网交换机连接在一个以太网中。该物理网络中没有连接路由器。同时，所有计算机也位于同一个 IP 网络中。

IP 分组在以太网中发送时，除了要有接收站的 IP 地址（IP 分组中的目的 IP 地址）外，还需要接收站的 MAC 地址（以太网帧中的目的 MAC 地址）。ARP 协议将 IP 地址（逻辑地址）动态映射为 MAC 地址（物理地址）。

实验中两人一组，在“未知”（使用命令 ***arp　-d　**** 清空 ARP 缓存表）和“已知”IP 网络内通信时所需地址映射（目的 IP 地址，目的 MAC 地址）这两种情况下，先后使用计算机上的通信测试命令（***ping***）发起一次通信过程，并通过使用 wireshark 软件捕获通信过程中通信双方的交互信息。比较两次通信过程中所捕获的分组数量、分组类型和分组内容，分析 ARP 协议的工作原理，包括：ARP 分组（ARP 请求分组和 ARP 应答分组）的产生条件、具体内容和传输方式。

每个实验者使用计算机上的 ARP 缓存表查看命令（***arp　-a***），查看本小组的 ARP 协议操作结果和 ARP 缓存表内容，了解 ARP 缓存表的形成及其在 ARP 协议操作过程中的作用。

（二）网络路径跟踪（TRACE）实验

本实验中，每个实验小组中的计算机分别连接在两个以太网中，每个以太网被配置为一个 IP 子网，4 台路由器按照实验拓扑结构互连这两个 IP 子网。

ICMP 协议作为 IP 协议的辅助协议，提供差错报告和查询机制。

实验者在计算机上使用路径跟踪命令（***tracert***）查看子网 A 和子网 B 之间的

通信路径，理解并掌握命令的用途和使用方法，结合 IP 协议、ICMP 协议分析命令的工作原理。

实验者通过更改 ***tracert*** 命令参数，结合 Sniffer 软件所捕获的数据报文和 ICMP 的差错报告机制，考察 IP 分组生存时间（TTL）的含义及其对网络间 IP 分组交付的影响，了解并体会 ***tracert*** 命令的工作原理。

（三）TCP 连接实验

本实验中，所有计算机位于一个物理网络中：所有计算机通过以太网交换机连接在一个以太网中。该物理网络中没有连接路由器，有一台 FTP 服务器。所有计算机和 FTP 服务器位于同一个 IP 网络中。

TCP 协议是一个面向连接的、可靠的运输层协议，通过连接建立和连接终止这两个过程完成面向连接的传输。

FTP 协议是一个用于文件传输的应用层协议，采用客户/服务器模式实现文件传输功能，使用 TCP 协议提供的面向连接的可靠传输服务。FTP 客户和服务器之间需要建立两条 FTP 连接：控制连接（端口 21）和数据连接（端口 20）。

实验者的计算机作为 FTP 客户，通过 ***ftp*** 命令与 FTP 服务器进行一次 FTP 会话活动。使用 Sniffer 软件捕获通信双方的交互信息，考察 TCP 协议的连接建立过程和连接终止过程。

分析 TCP 连接建立和连接终止过程中所捕获的 TCP 报文段，掌握 TCP 报文段首部中的端口地址、序号、确认号和各个码元比特的含义和作用。结合 FTP 操作，体会网络应用程序间的交互模式——客户/服务器（C/S）模式。

五、实验目的：

通过使用 wireshark（嗅探）工具，实现捕捉 ARP、ICMP、FTP 等协议的数据包，以理解 TCP/IP 协议栈中多种协议的数据结构、会话连接建立和终止的过程、TCP 序列号、应答序号的变化规律。并且通过实验了解 FTP、HTTP 等协议明文传输的特性，以建立安全意识，防止 FTP、HTTP 等协议由于传输明文密码造成的泄密。

六、实验内容：

1、 地址解析协议（ARP）实验。

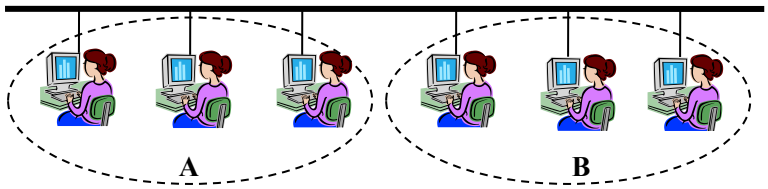
2、 网络路径跟踪（TRACE）实验。

3、 TCP 连接实验。

七、实验器材（设备、元器件）：

1、 实验人数 50～80 人，每人 1 台计算机；2 人一组配合完成本实验。

2、 拓扑：（A、B 范围中的主机分别简称为 A 主机和 B 主机）：



3、 设备：以太网交换机 2～4 台；计算机 50～80 台。

4、 软件：Sniffer 软件（捕获网络上传输的数据报文）。

八、实验步骤：

1、 地址解析协议（ARP）实验：

(1) 在 A、B 主机上运行 wireshark 软件，设置捕获条件：
Capture→Option→Capture Filter:arp

(2) 清空 A、B 主机上的 ARP 缓存表（命令：arp -d *）。

(3) 在 A、B 主机上启动 wireshark 的捕获过程。首先由 A 主机 PING B 主机。PING 结束以后，停止 A、B 主机的 wireshark 捕获过程，保存捕获数据。

(4) 查看 A、B 主机上的 ARP 缓存表（命令：arp -a）。

(5) 在 A、B 主机上再次启动 wireshark 的捕获过程，由 B 主机 PING A 主机。PING 结束以后，停止 A、B 主机的 wireshark 捕获过程，保存捕获数据。

(6) 查看 A、B 主机上的 ARP 缓存表（命令：arp -a）。

(7) 查看并比较步骤 3 和步骤 5 中 A、B 主机上 wiresharkr 软件所捕获的数据报文数量和类型。

2、 网络路径跟踪（TRACE）实验：

(1) 根据实验拓扑要求设置主机上的 TCP/IP 协议配置参数。运行 Sniffer 软件，设置捕获条件：
Capture→Option→Capture Filter:icmp

(2) 计算子网 A、B 的子网地址和子网广播地址。

(3) 路径跟踪——TRACE

1) 在主机的 cmd 窗口键入“tracert”命令，查看并分析选项-d、-h 的含义和作用。

2) 启动 wireshark 捕获过程，子网 A、B 中的主机 TRACE 对方子

网中的 1 个主机 IP 地址。TRACE 结束以后，停止 wireshark 的捕获过程，保存捕获数据，查看并解释本主机上显示的通信结果。

3) 启动 wireshark 捕获过程，使用-d 选项 TRACE 步骤 3-2 中的目的主机。TRACE 结束后，停止 Sniffer 的捕获过程，保存捕获数据，查看本主机上显示的通信结果，并与步骤 3-2 的结果相比较。

4) 启动 wireshark 捕获过程，使用-d 和-h 选项重新 TRACE 步骤 3-2 中的目的主机，-h 选项取值分别为 1、2、3。TRACE 结束以后，停止 wireshark 的捕获过程，保存捕获数据，查看本主机上显示的通信结果。

5) 启动 wireshark 捕获过程，使用-d 和-h 选项 TRACE 对方子网中 1 个不存在的主机 IP 地址，-h 选项取值为 6。TRACE 结束以后，停止 wireshark 的捕获过程，保存捕获数据，查看本主机上所显示的通信结果。

3、 TCP 连接实验：

(1) 在主机上运行 Sniffer 软件，设置捕获条件：

Capture→Option→Capture Filter:ftp

(2) 启动 wireshark 的捕获过程，并在主机的 cmd 窗口中以命令行的方式启动 FTP 客户进程，过程如下：

（黑斜体表示学生输入内容，其它为系统显示信息）

C:\>　ftp 192.168.3.254

Connected to 192.168.3.254.

220 Serv-U FTP Server v4.0 for WinSock ready…

User (192.168.3.254:(none)):　***ftp***

331 User name okay, please send complete E-mail address as password.

Password:　***ftp@***

230 User logged in, proceed.

ftp>　quit

221 Goodbye!

(3) 停止 wireshark 的捕获过程，保存捕获数据。

(4) 查看捕获报文中的本机 FTP 进程端口号、FTP 服务器进程端口号、本机 TCP 初始序号和服务器 TCP 初始序号。

(5) 重复步骤 2 和 3，查看捕获报文中的本机 FTP 进程端口号、FTP 服务器进程端口号、本机 TCP 初始序号和服务器 TCP 初始序号，并

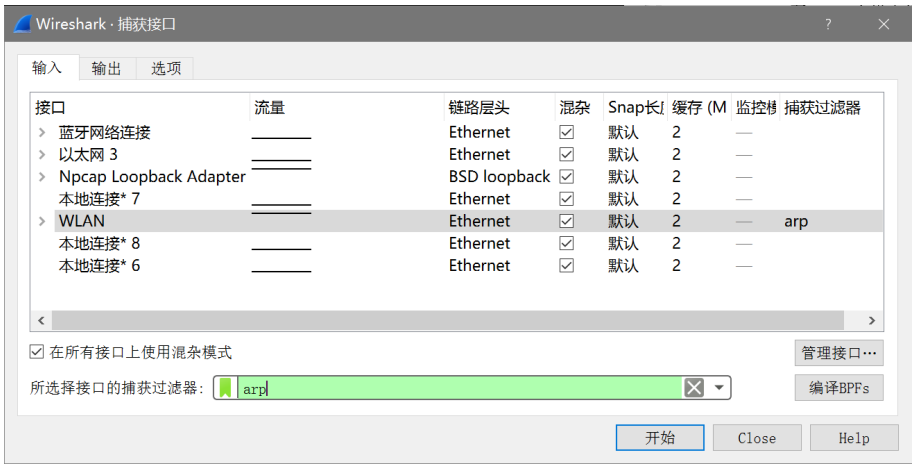
与步骤 4 的查看结果相比较。

九、实验数据及结果分析

1、 地址解析协议（ARP）实验：

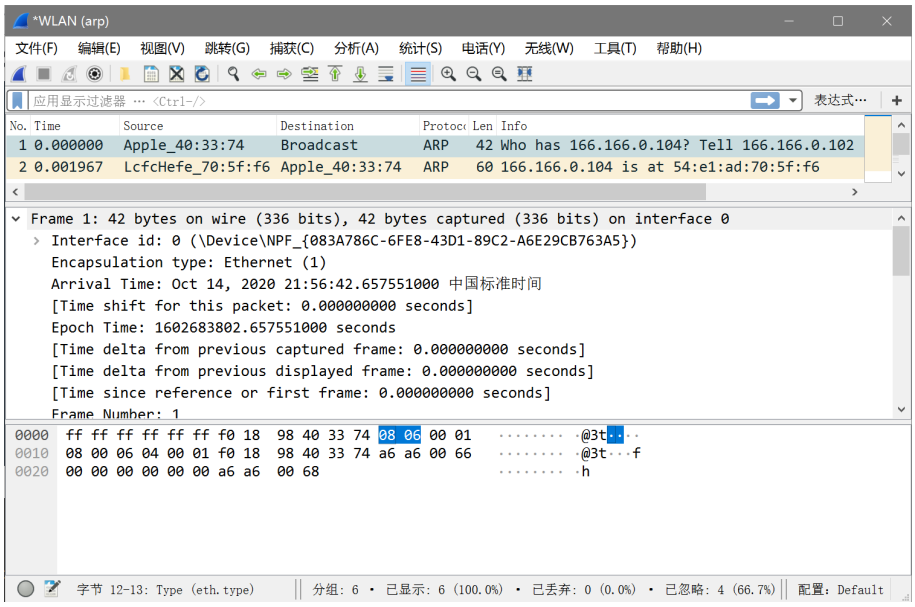
(1) 在 A、B 主机上运行 wireshark 软件，设置捕获条件：

Capture→Option→Capture Filter:arp



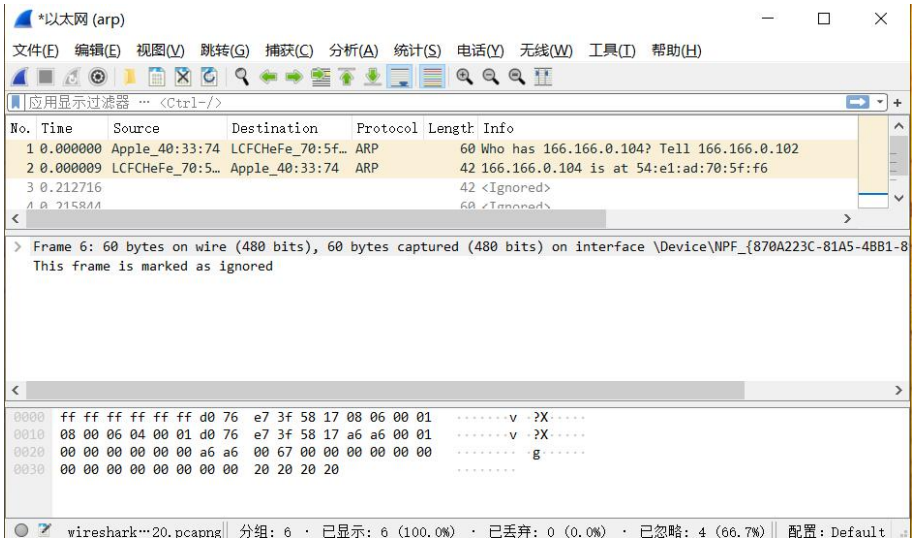
(2) 清空 A、B 主机上的 ARP 缓存表（命令：arp -d *）。

(3) 在 A、B 主机上启动 wireshark 的捕获过程。首先由 A 主机 PING B 主机。PING 结束以后，停止 A、B 主机的 wireshark 捕获过程，保存捕获数据。



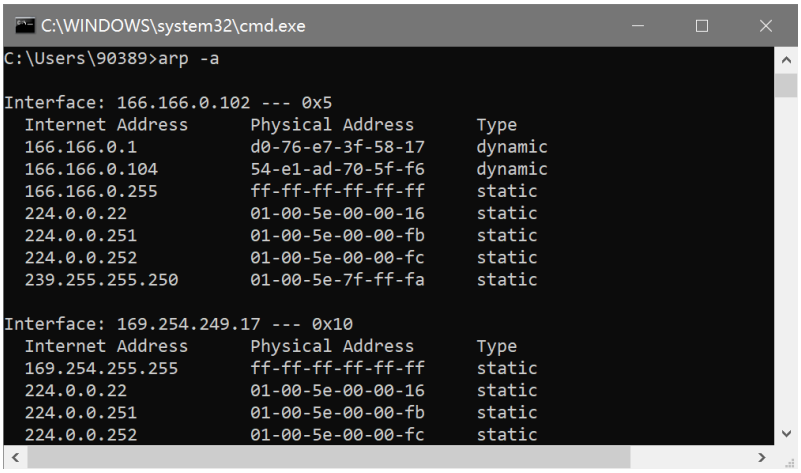
A 主机：166.166.0.102；B 主机：166.166.0.104；A 主机捕获的 ARP 报文显示，发起 ping 166.166.0.104 命令后，A 主机在局域网中发出 ARP 广播，询问“谁是 166.166.0.104？告诉 166.166.0.102”；其后，收到来自

B 主机的回应 ARP 报文，回答内容为“166.166.0.104 的 MAC 地址是 54:e1:ad:70:5f:f6”。

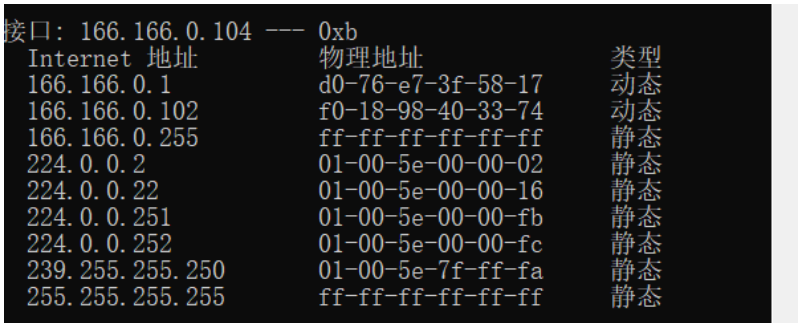


B 主机上使用 Wireshark 捕获的 ARP 报文显示，B 主机收到了来自 A 主机的询问报文，并向 A 主机发送回复报文。

(4) 查看 A、B 主机上的 ARP 缓存表（命令：arp -a）。



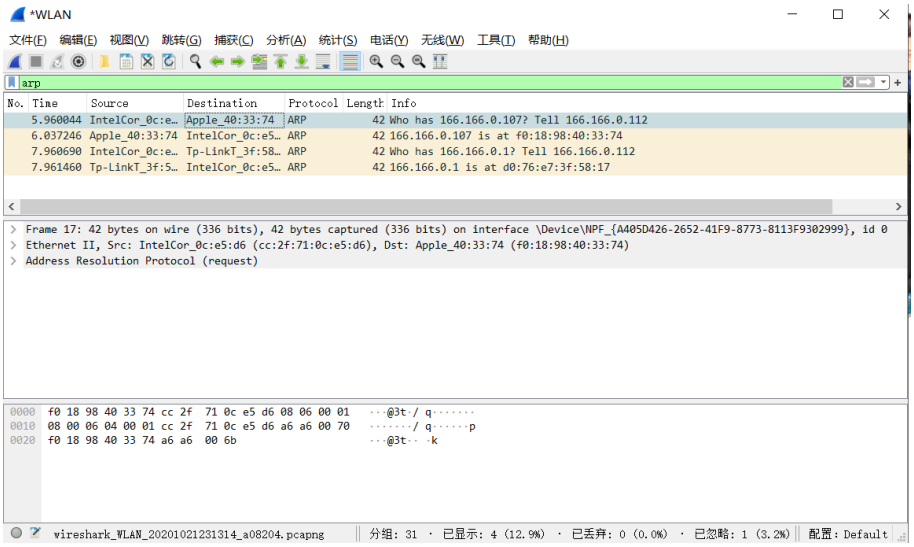
可以在 A 主机的缓存表中发现：已经记录 B 主机的<IP, MAC>地址对，记录类型为动态。



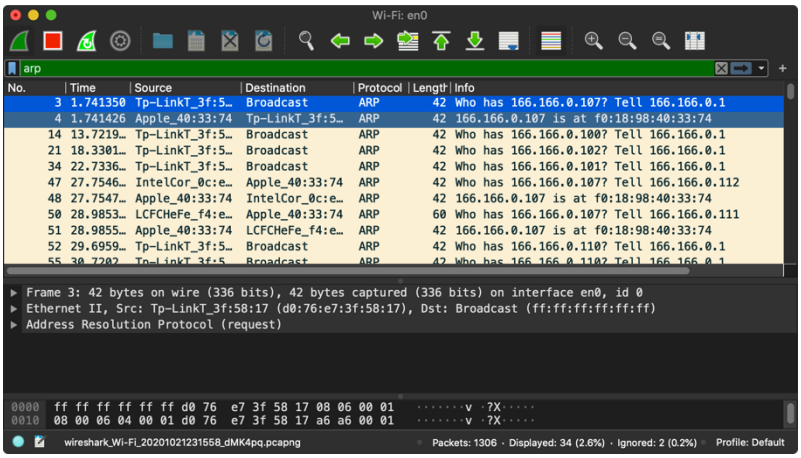
同样，可以在 B 主机的缓存表中发现：已经记录 A 主机的<IP, MAC>

地址对，记录类型为动态。

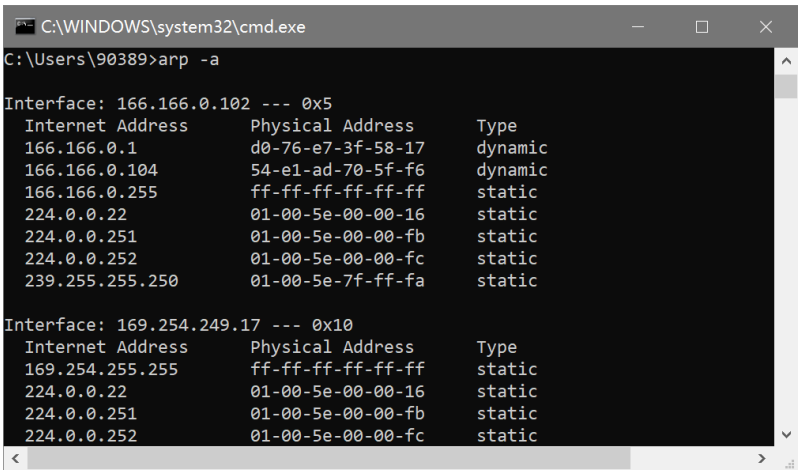
(5) 在 A、B 主机上再次启动 wireshark 的捕获过程，由 B 主机 PING A 主机。PING 结束以后，停止 A、B 主机的 wireshark 捕获过程，保存捕获数据。

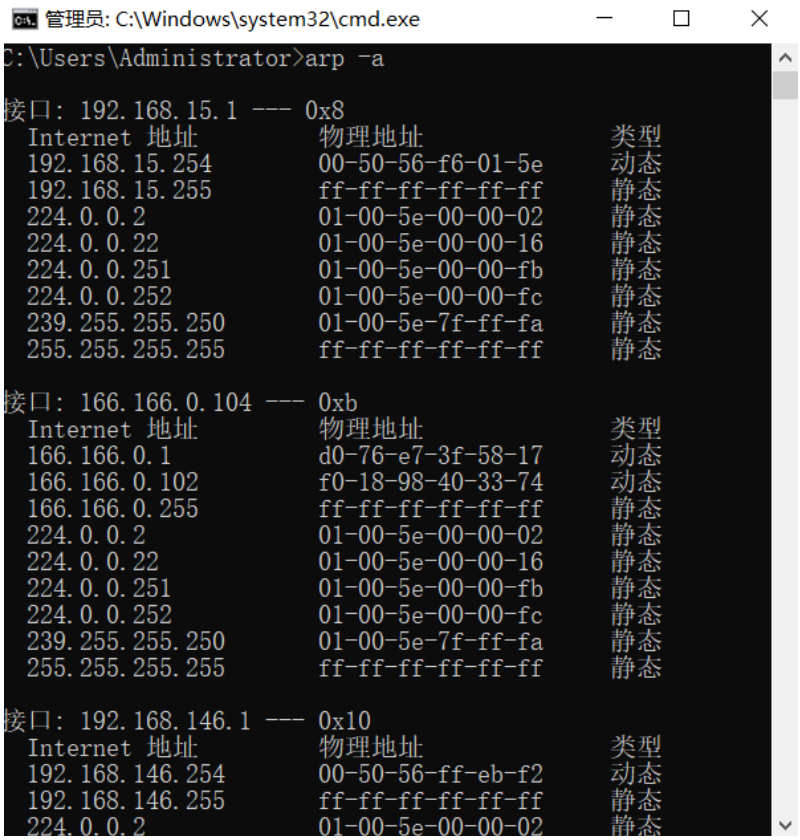


B 主机：166.166.0.112；A 主机：166.166.0.107；B 主机捕获的 ARP 报文显示，发起 ping 166.166.0.107 命令后，B 主机在局域网中发出 ARP 广播，询问“谁是 166.166.0.107？告诉 166.166.0.112”；其后，收到来自 A 主机的回应 ARP 报文，回答内容为“166.166.0.107 的 MAC 地址是 f0:18:98:40:33:74”。



(6) 查看 A、B 主机上的 ARP 缓存表（命令：arp -a）。





(7) 查看并比较步骤 3 和步骤 5 中 A、B 主机上 wiresharkr 软件所捕获的数据报文数量和类型。

2、 网络路径跟踪（TRACE）实验：

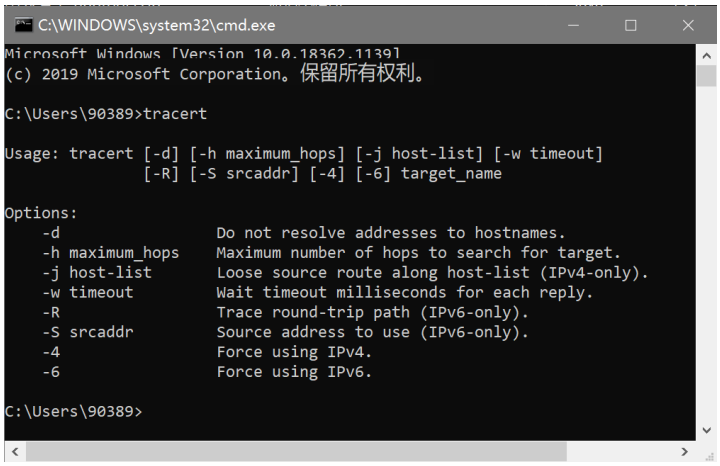
(1) 根据实验拓扑要求设置主机上的 TCP/IP 协议配置参数。运行 Sniffer 软件，设置捕获条件：

Capture→Option→Capture Filter:icmp

(2) 计算子网 A、B 的子网地址和子网广播地址。

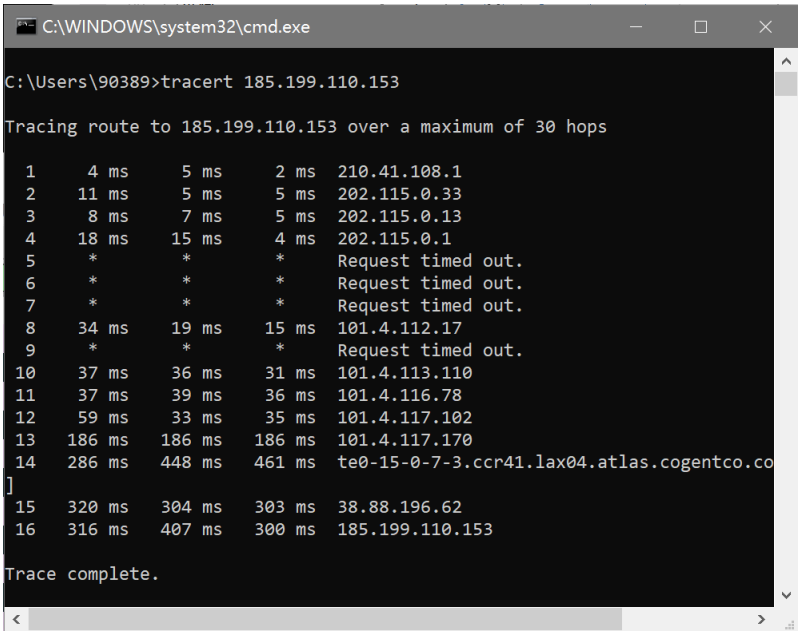
(3) 路径跟踪——TRACE

1) 在主机的 cmd 窗口键入“tracert”命令，查看并分析选项-d、-h 的含义和作用。

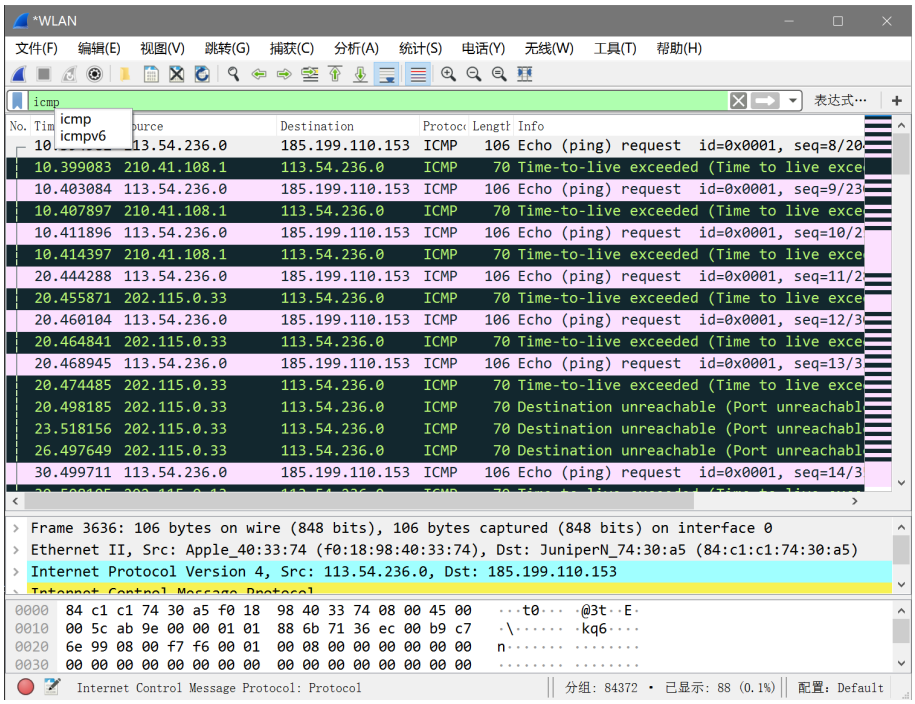


参数-d 的意思是指定不将 IP 地址解析到主机名称；参数-h 可以置顶最大跳数。

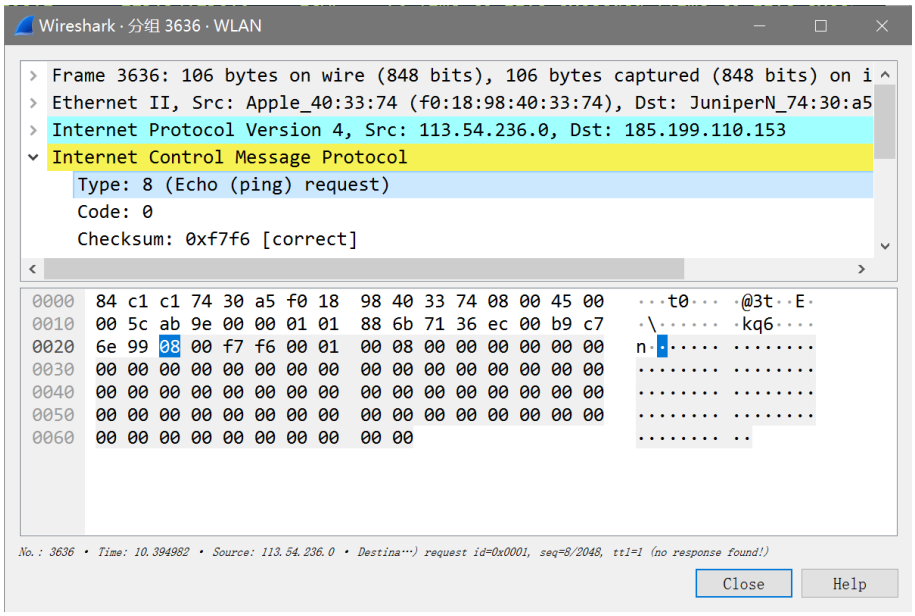
2) 启动 wireshark 捕获过程，子网 A、B 中的主机 TRACE 对方子网中的 1 个主机 IP 地址。TRACE 结束以后，停止 wireshark 的捕获过程，保存捕获数据，查看并解释本主机上显示的通信结果。



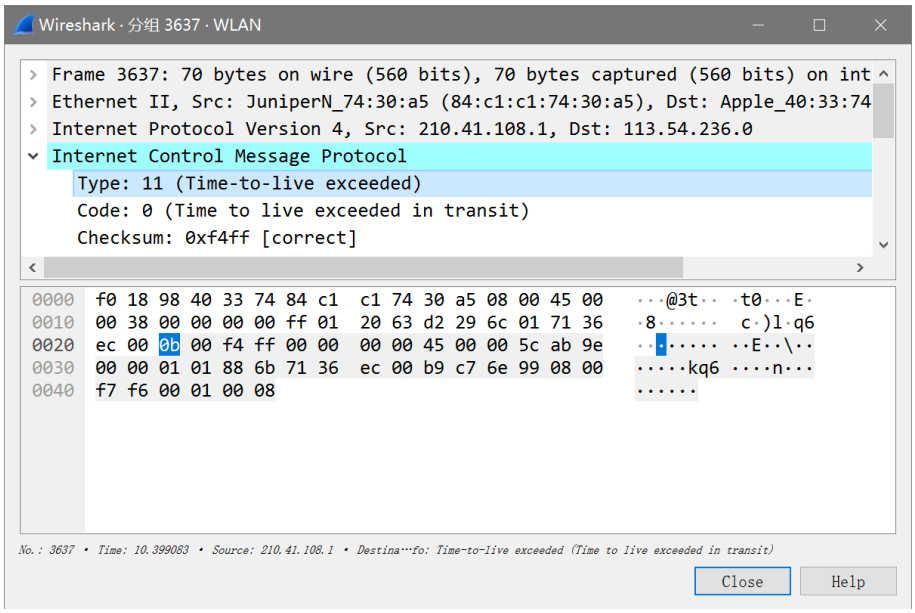
可以看出，从本机到目的主机，中间一共经历了 16 跳。



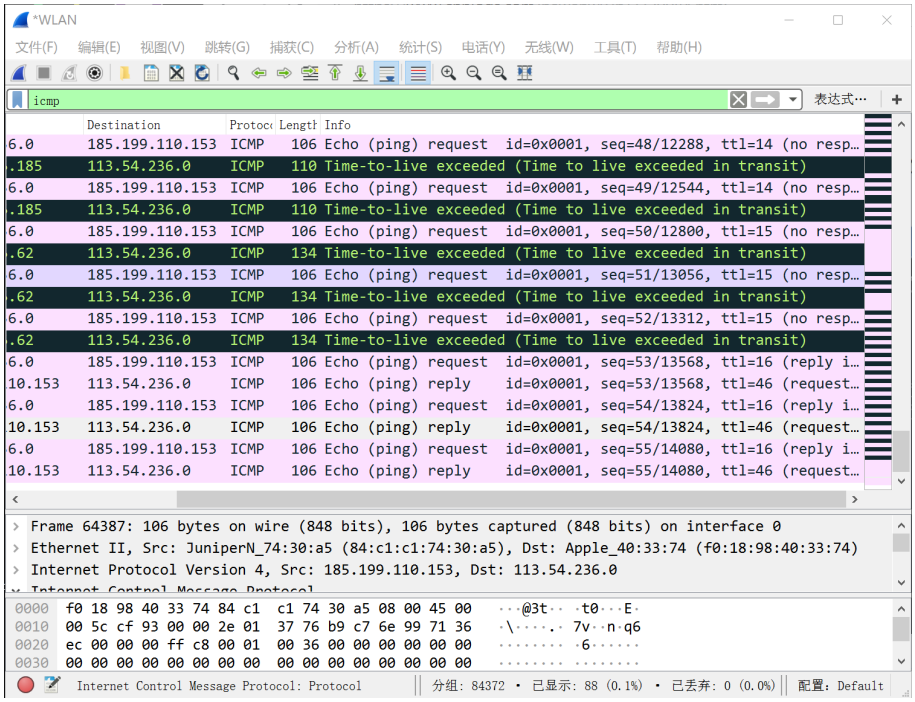
本机捕获到的 ICMP 报文。



第一个 ICMP 报文类型代码为 8，询问报文。

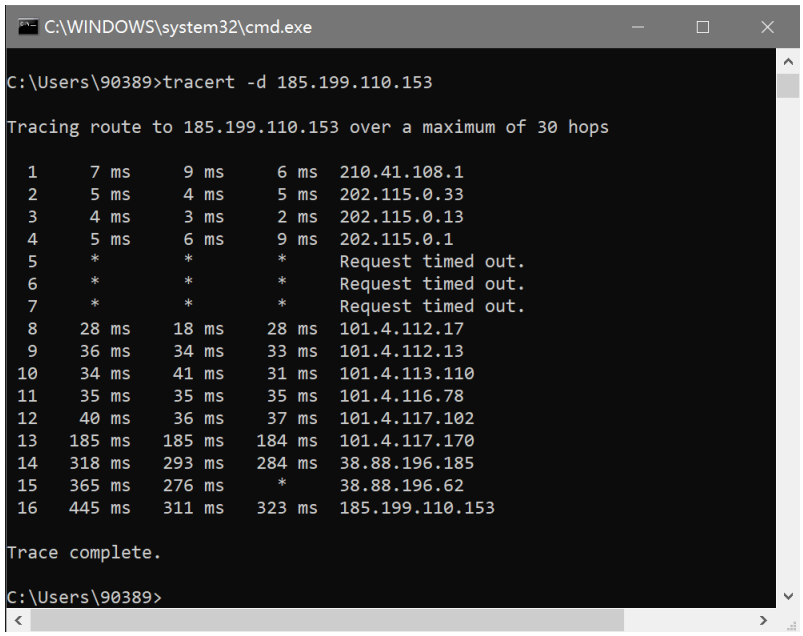


第二个 ICMP 报文类型代码为 11，超时报文。

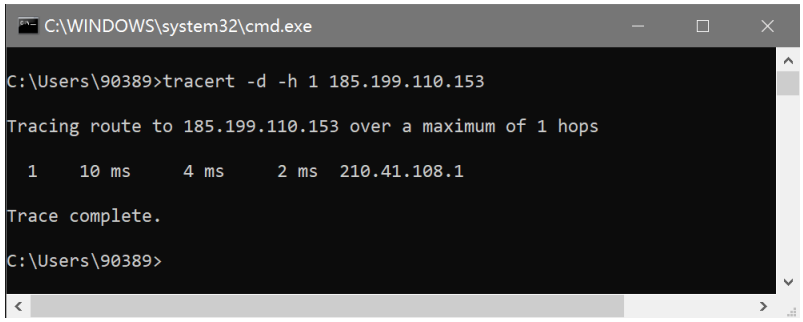


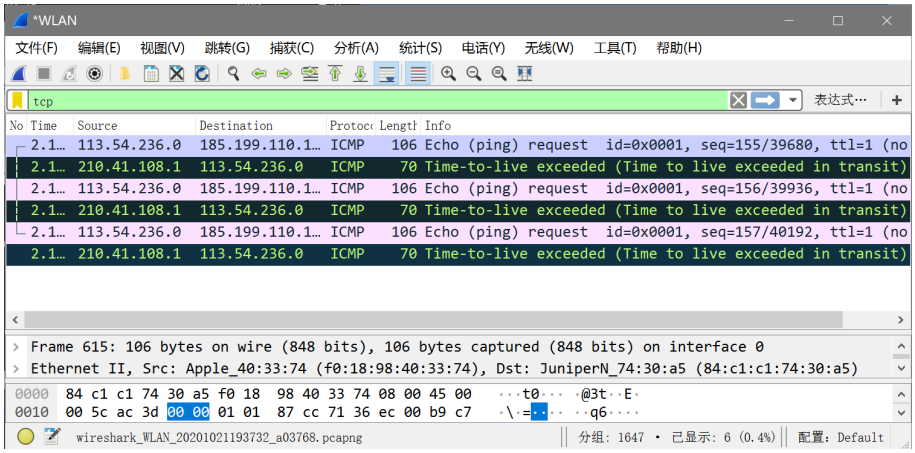
可以看出，在 TTL 增大到 16 之前，ICMP 报文一直处于“超时”状态。直到 TTL 增大到 16 后，得到了 ICMP 回复报文。说明，本机与目的主机之间有 16 跳。

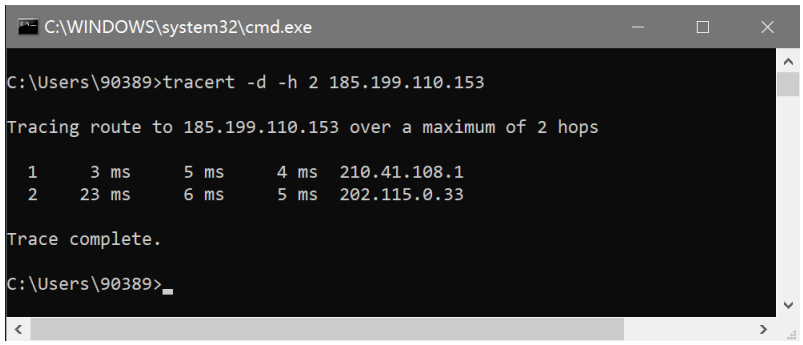
3) 启动 wireshark 捕获过程，使用-d 选项 TRACE 步骤 3-2 中的目的主机。TRACE 结束后，停止 Sniffer 的捕获过程，保存捕获数据，查看本主机上显示的通信结果，并与步骤 3-2 的结果相比较。

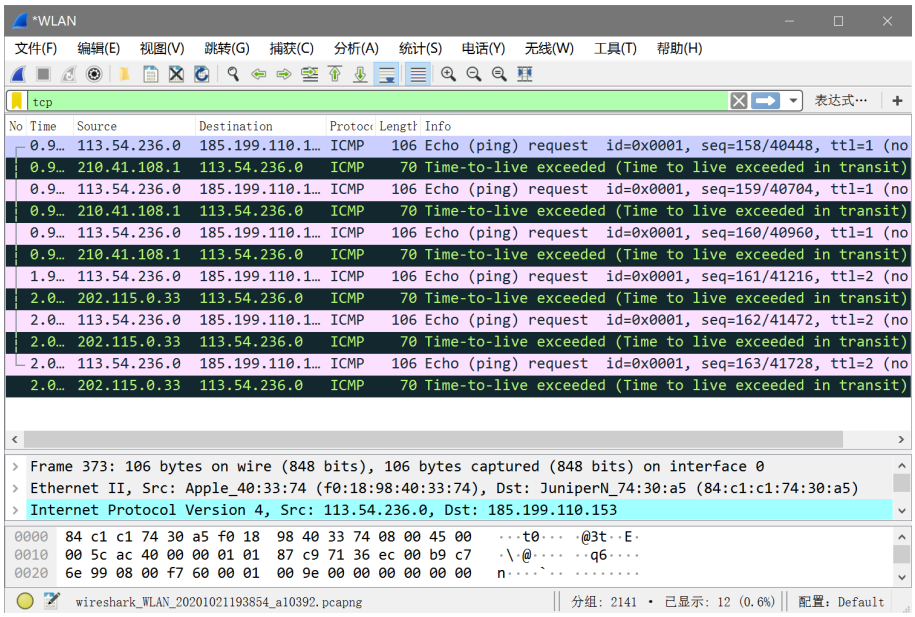


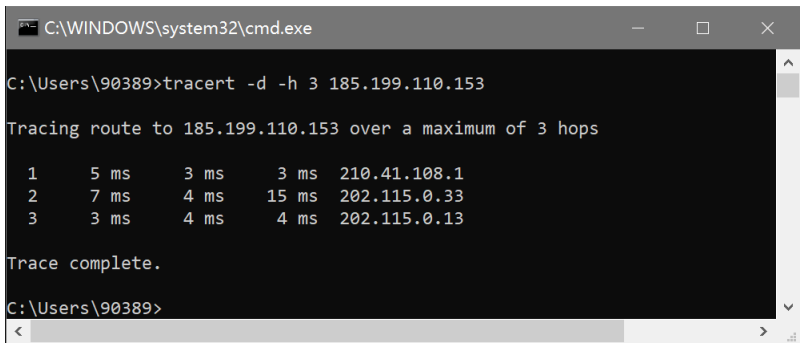
4) 启动 wireshark 捕获过程，使用-d 和-h 选项重新 TRACE 步骤 3-2 中的目的主机，-h 选项取值分别为 1、2、3。TRACE 结束以后，停止 wireshark 的捕获过程，保存捕获数据，查看本主机上显示的通信结果。

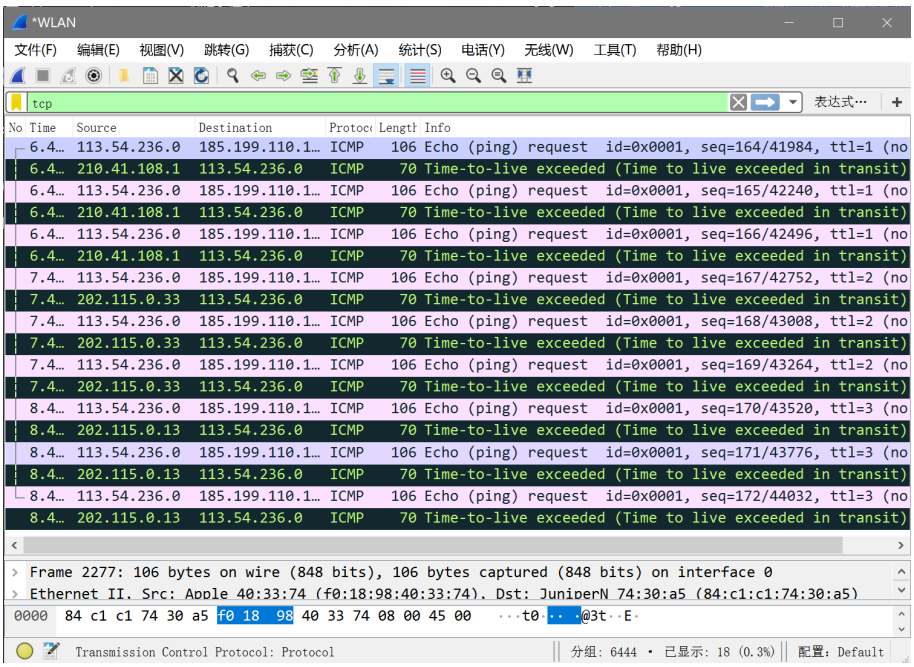




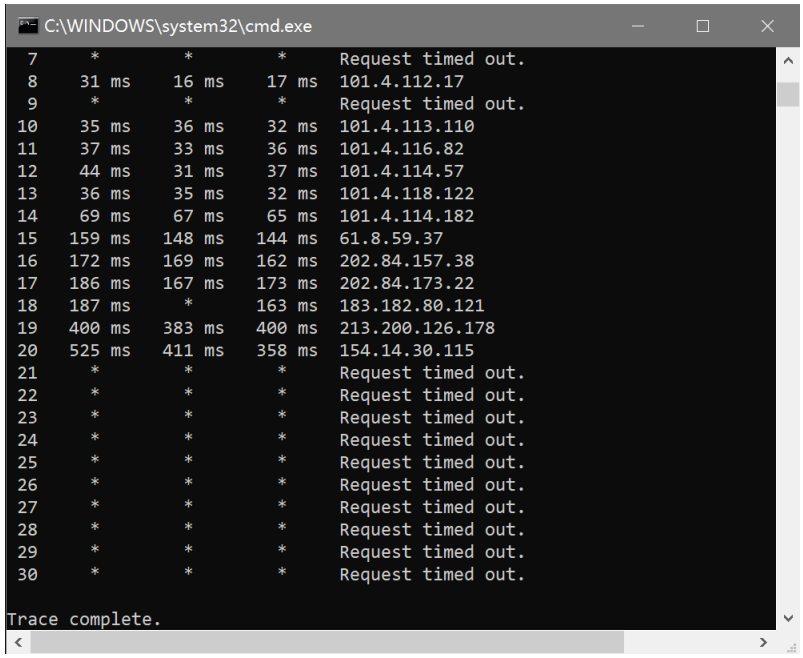

C:\WINDOWS\system32\cmd.exe
C:\Users\90389>tracert -d -h 2 185.199.110.153
Tracing route to 185.199.110.153 over a maximum of 2 hops
1 3 ms 5 ms 4 ms 210.41.108.1
2 23 ms 6 ms 5 ms 202.115.0.33
Trace complete.
C:\Users\90389>

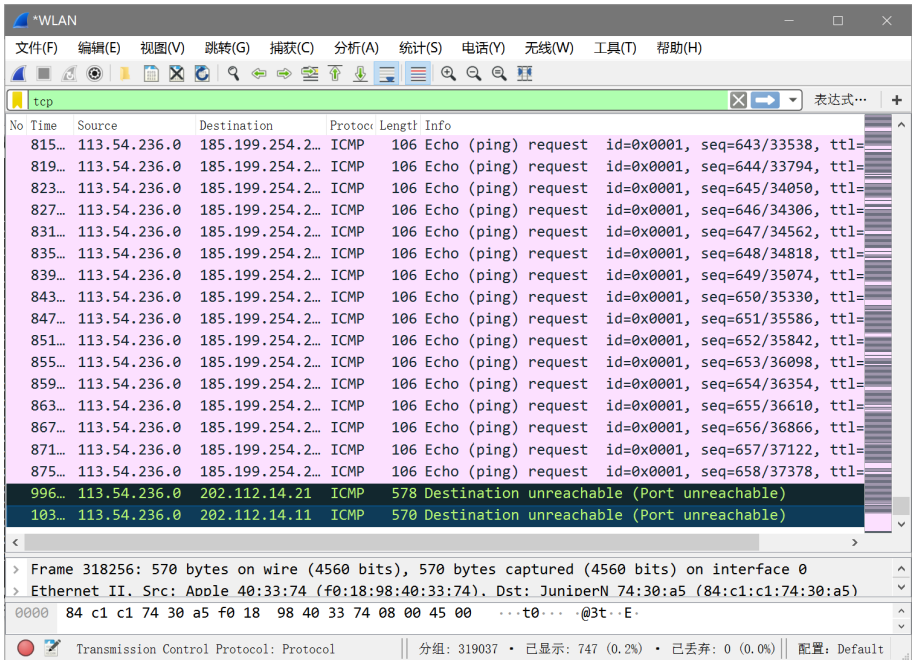

*WLAN
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
tcp
表达式…
No Time Source Destination Protoc Length Info
0.9… 113.54.236.0 185.199.110.1… ICMP 106 Echo (ping) request id=0x0001, seq=158/40448, ttl=1 (no
0.9… 210.41.108.1 113.54.236.0 ICMP 70 Time-to-live exceeded (Time to live exceeded in transit)
0.9… 113.54.236.0 185.199.110.1… ICMP 106 Echo (ping) request id=0x0001, seq=159/40704, ttl=1 (no
0.9… 210.41.108.1 113.54.236.0 ICMP 70 Time-to-live exceeded (Time to live exceeded in transit)
0.9… 113.54.236.0 185.199.110.1… ICMP 106 Echo (ping) request id=0x0001, seq=160/40960, ttl=1 (no
0.9… 210.41.108.1 113.54.236.0 ICMP 70 Time-to-live exceeded (Time to live exceeded in transit)
1.9… 113.54.236.0 185.199.110.1… ICMP 106 Echo (ping) request id=0x0001, seq=161/41216, ttl=2 (no
2.0… 202.115.0.33 113.54.236.0 ICMP 70 Time-to-live exceeded (Time to live exceeded in transit)
2.0… 113.54.236.0 185.199.110.1… ICMP 106 Echo (ping) request id=0x0001, seq=162/41472, ttl=2 (no
2.0… 202.115.0.33 113.54.236.0 ICMP 70 Time-to-live exceeded (Time to live exceeded in transit)
2.0… 113.54.236.0 185.199.110.1… ICMP 106 Echo (ping) request id=0x0001, seq=163/41728, ttl=2 (no
2.0… 202.115.0.33 113.54.236.0 ICMP 70 Time-to-live exceeded (Time to live exceeded in transit)
Frame 373: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 0
Ethernet II, Src: Apple_40:33:74 (f0:18:98:40:33:74), Dst: JuniperN_74:30:a5 (84:c1:c1:74:30:a5)
Internet Protocol Version 4, Src: 113.54.236.0, Dst: 185.199.110.153
0000 84 c1 c1 74 30 a5 f0 18 98 40 33 74 08 00 45 00
0010 00 5c ac 40 00 00 01 01 87 c9 71 36 ec 00 b9 c7
0020 6e 99 08 00 f7 60 00 01 00 9e 00 00 00 00 00 00
wireshark_WLAN_20201021193854_a10392.pcapng
分组: 2141 · 已显示: 12 (0.6%)
配置: Default


C:\WINDOWS\system32\cmd.exe
C:\Users\90389>tracert -d -h 3 185.199.110.153
Tracing route to 185.199.110.153 over a maximum of 3 hops
1 5 ms 3 ms 3 ms 210.41.108.1
2 7 ms 4 ms 15 ms 202.115.0.33
3 3 ms 4 ms 4 ms 202.115.0.13
Trace complete.
C:\Users\90389>



5) 启动 wireshark 捕获过程，使用-d 和-h 选项 TRACE 对方子网中 1 个不存在的主机 IP 地址，-h 选项取值为 6。TRACE 结束以后，停止 wireshark 的捕获过程，保存捕获数据，查看本主机上所显示的通信结果。





可以看出，tracert 一个不存在的主机时，会一直出现 ICMP 超时报文，无法得到目的主机的回应。

3、 TCP 连接实验：

(1) 在主机上运行 Sniffer 软件，设置捕获条件：

Capture→Option→Capture Filter:ftp

(2) 启动 wireshark 的捕获过程，并在主机的 cmd 窗口中以命令行的方式启动 FTP 客户进程，过程如下：

（黑斜体表示学生输入内容，其它为系统显示信息）

C:\>　ftp 192.168.3.254

Connected to 192.168.3.254.

220 Serv-U FTP Server v4.0 for WinSock ready…

User (192.168.3.254:(none)):　***ftp***

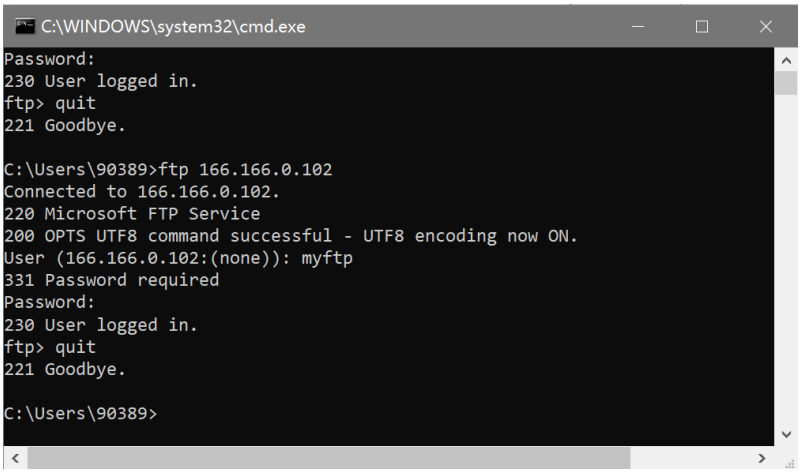
331 User name okay, please send complete E-mail address as password.

Password:　***ftp@***

230 User logged in, proceed.

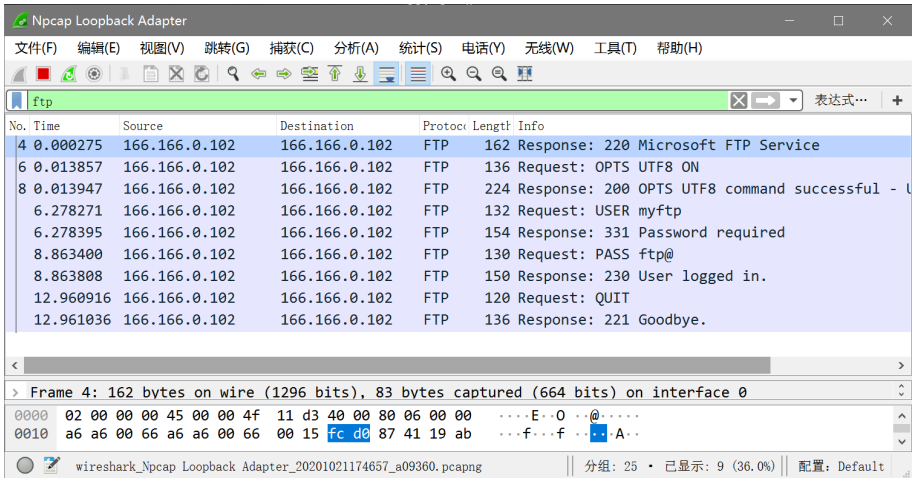
ftp>　quit

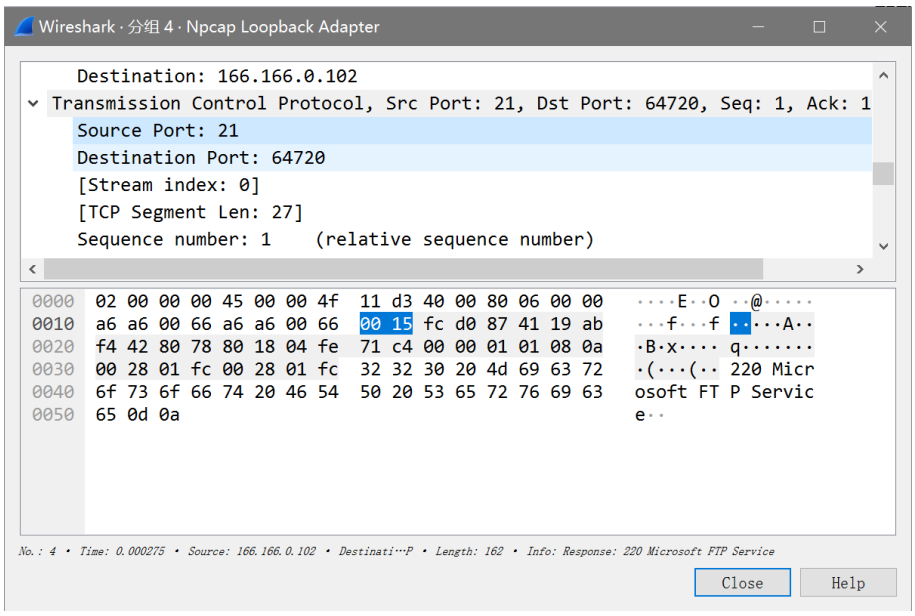
221 Goodbye!



(3) 停止 wireshark 的捕获过程，保存捕获数据。

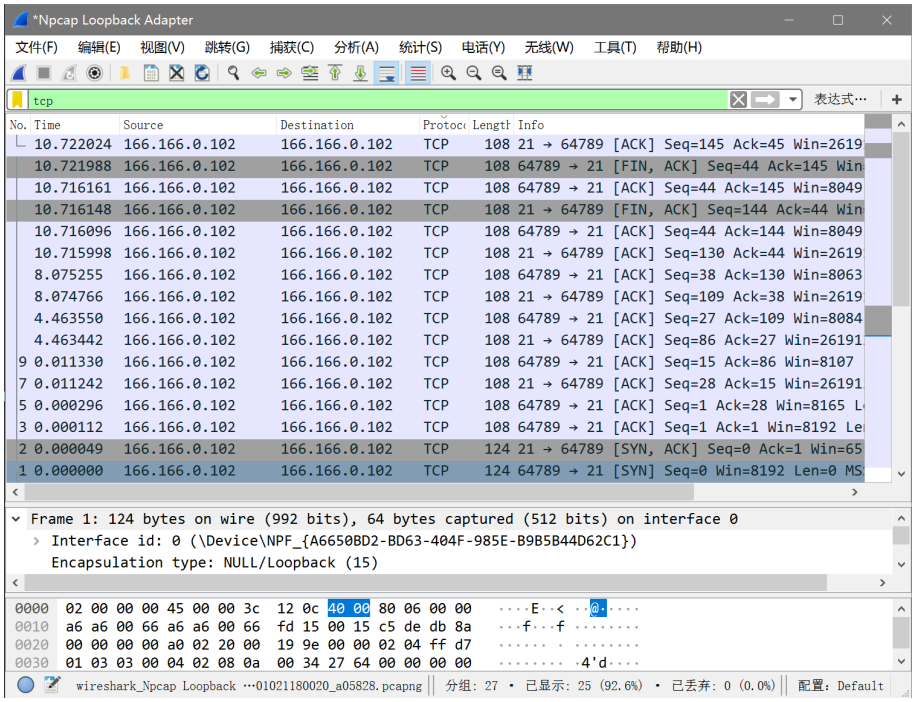
(4) 查看捕获报文中的本机 FTP 进程端口号、FTP 服务器进程端口号、本机 TCP 初始序号和服务器 TCP 初始序号。





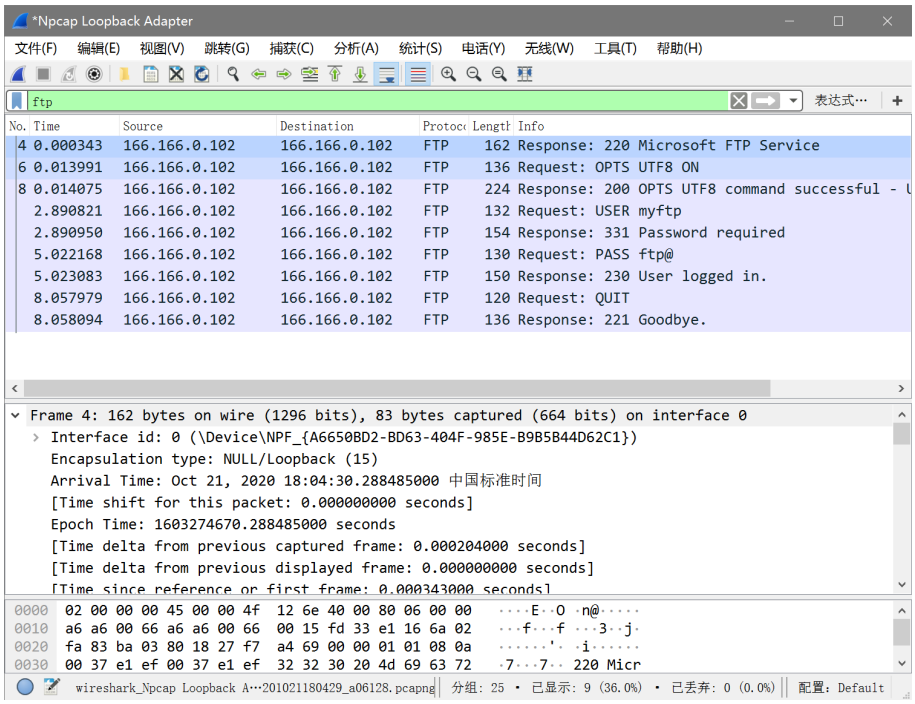
可以看出，本机 FTP 进程端口号为 21，FTP 服务器进程端口号为

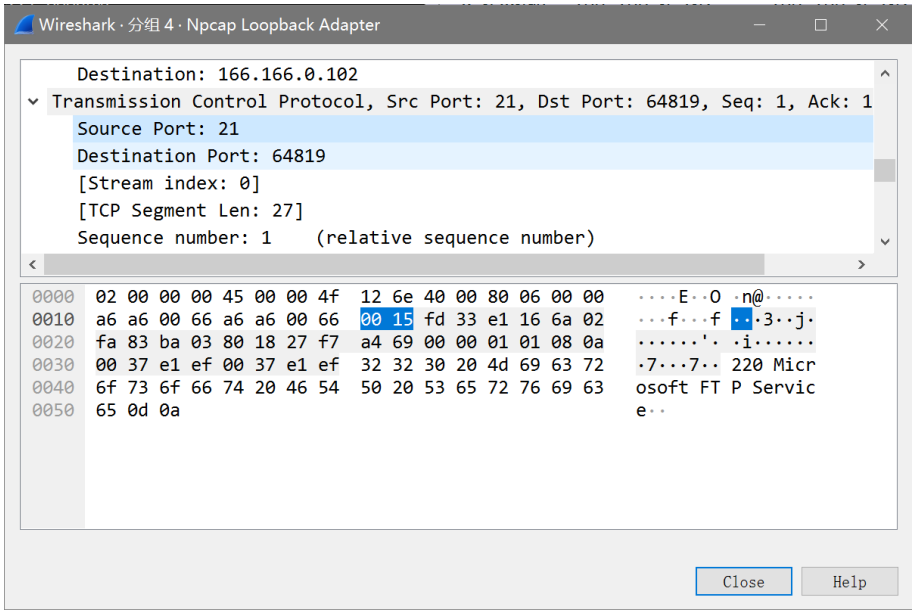
64720。



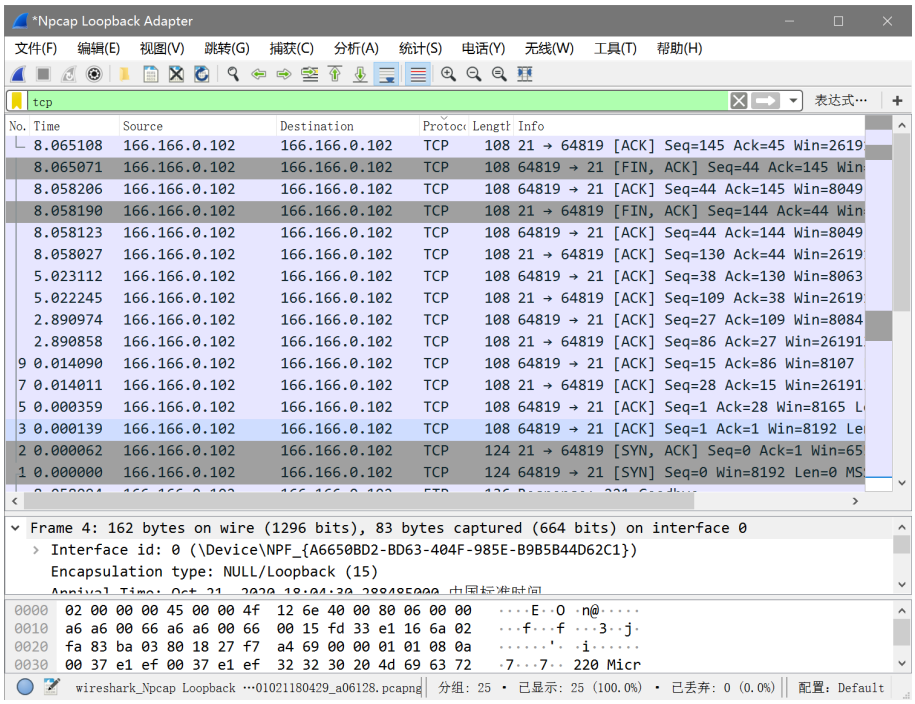
可以看出，本机 TCP 初始序列号为 1，服务器 TCP 初始序列号为 28。

(5) 重复步骤 2 和 3，查看捕获报文中的本机 FTP 进程端口号、FTP 服务器进程端口号、本机 TCP 初始序号和服务器 TCP 初始序号，并与步骤 4 的查看结果相比较。





可以看出，本机 FTP 进程端口号为 21，FTP 服务器进程端口号为 64819。与(4)相比，FTP 服务器进程端口号发生了变化。



可以看出，本机 TCP 初始序列号为 1，服务器 TCP 初始序列号为 28。与(4)相比，TCP 初始序列号未发生变化。

十、实验结论

按实验内容与步骤完成本实验，得到的实验结果与预期一致，使用 Wireshark 软件验证了 ARP 协议、tracert 命令、FTP 协议和 TCP 协议。

十一、总结及心得体会

通过对 ARP 协议、tracert 命令、FTP 协议和 TCP 协议的实践，了解、掌握了不同参数的作用，熟悉了相关命令操作。通过对 Wireshark 抓包工具的使用，实现捕捉 ARP、ICMP、FTP 等协议的数据包，以理解 TCP/IP 协议栈中多种协议的数据结构、会话连接建立和终止的过程、TCP 序列号、应答序号的变化规律。并且通过实验了解 FTP、HTTP 等协议明文传输的特性，以建立安全意识，防止 FTP、HTTP 等协议由于传输明文密码造成的泄密。

（一） 地址解析协议（ARP）实验

1、计算机在通信过程中，什么情况下要发送 ARP 请求分组？什么情况下不发送 ARP 请求分组？

答：在 ARP 高速缓存中不存在目标 IP 到其 MAC 地址的映射关系时，就需要再发送 ARP 请求分组了；反之则不需要发送。

2、如果步骤 4 或步骤 6 中显示 A 主机或 B 主机上有多余一条的 ARP 映射表项，请根据实验中的数据报文捕获结果，解释为什么会获得这些 ARP 映射表项？

答：这是系统自动搜索捕获的 ARP 映射表项，因为系统每隔一段时间就会自动搜索可用的 ARP，用 arp －d *命令可暂时清除缓存表，但一段时间后又会出现。

3、请分析本实验中关于 Sniffer 软件捕获条件的设置问题：

1）Address Type 捕获条件是否能设置成为 IP？为什么？

答：可以。捕获条件相当于是一个过滤器，可以设置捕获的协议类型或 IP 地址。

2）如果 Station2 的地址设置成为对方主机的地址，对实验的捕获操作会有什么影响？

答：捕获结束后，仅会在结果中显示报文目的地址为对方主机的报文。可以更准确的获取想要捕获的报文。

3）如果 Station1 和 Station2 的地址均设置成为 any，对实验的捕获操作会有什么影响？

答：经过这个网络端口的所有类型的报文都会被捕获到。使得捕获结果杂乱，不容易筛选出所需要的报文。

（二）网络路径跟踪（TRACE）实验

1、TRACE 的功能是什么？有哪些可能的响应？产生这些响应的原因是什么？

答：TRACE 路由跟踪实用程序，用于确定 IP 数据包访问目标所采取的

路径。Tracert 命令用 IP 生存时间 (TTL) 字段和 ICMP 错误消息来确定从一个主机到网络上其他主机的路由。可能产生的响应有：正确回显途径站点的 IP 地址与抵达时间、超时等。超时是由于路由不通畅或对方节点关闭了回显功能造成的。

2、分析步骤 3 中捕获的 TRACE 报文，阐述 TRACE 的工作原理。

答：TRACE 的原理是通过向目标发送不同 IP 生存时间 (TTL) 值的“Internet 控制消息协议 (ICMP)”回应数据包，Tracert 诊断程序确定到目标所采取的路由。要求路径上的每个路由器在转发数据包之前至少将数据包上的 TTL 递减 1。数据包上的 TTL 减为 0 时，路由器应该将“ICMP 已超时”的消息发回源系统。 Tracert 先发送 TTL 为 1 的回应数据包，并在随后的每次发送过程将 TTL 递增 1，直到目标响应或 TTL 达到最大值，从而确定路由。通过检查中间路由器发回的“ICMP 已超时”的消息确定路由。某些路由器不经询问直接丢弃 TTL 过期的数据包，这在 Tracert 实用程序中看不到。

（三）TCP 连接实验

1、一条 TCP 连接需要用哪些参数来标识？实验步骤 2 和实验步骤 5 中的 TCP 连接是否是同一条连接？请根据实验记录分别写出其连接标识。

答：一条 TCP 连接需要用源地址、源端口、目的地址、目的端口来标识。实验步骤 2 和实验步骤 5 中的 TCP 连接不是同一条连接。步骤 2 中，源地址、源端口、目的地址、目的端口分别是：166.166.0.102、21、166.166.0.102、64789;步骤 5 中，源地址、源端口、目的地址、目的端口分别是：166.166.0.102、21、166.166.0.102、64819。

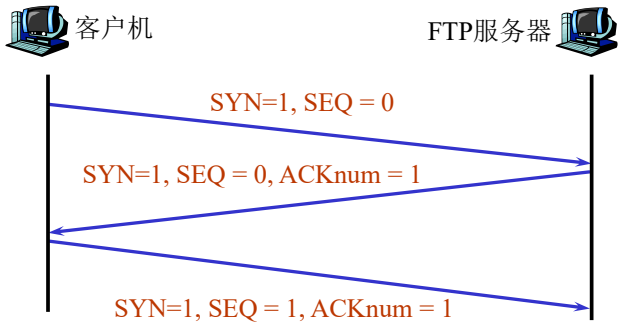
2、本实验中用来建立 TCP 连接的 3 个 TCP 报文段的详细作用分别是什么？每个报文段中包含了哪些用于连接建立的信息？

答：这个过程称为 TCP 连接的“三次握手”，其具体步骤如下。第一次握手：建立连接时，客户端发送 syn 包（seq=j）到服务器，并进入 SYN_SENT 状态，等待服务器确认；SYN：同步序列编号（Synchronize Sequence Numbers）。第二次握手：服务器收到 syn 包，必须确认客户的 SYN（ack=j+1），同时自己也发送一个 SYN 包（seq=k），即 SYN+ACK 包，此时服务器进入 SYN_RECV 状态。第三次握手：客户端收到服务器的 SYN+ACK 包，向服务器发送确认包 ACK(ack=k+1），此包发送完毕，客户端和服务器进入 ESTABLISHED（TCP 连接成功）状态，完成三次握手。三个 TCP 报文段起到了发起请求连接、相互确认请求的作用。

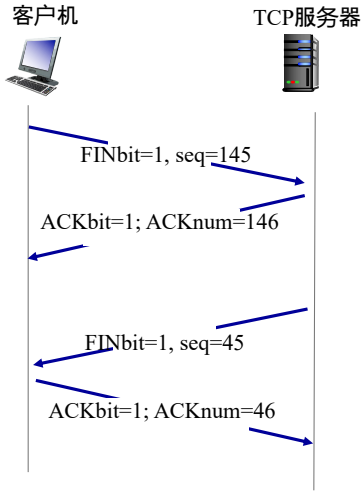
3、利用步骤 3 中保存的捕获数据，画出主机与 FTP 服务器之间的 TCP

连接建立过程和 TCP 连接终止过程的时序交互图，并在图中注明每个 TCP 报文段的类型、序号和确认号。

答：建立过程：



终止过程：



十二、对本实验过程及方法、手段的改进建议

本实验设计与教材结合紧密、较为简单，通过对 ARP 协议、tracert 命令、FTP 协议和 TCP 协议的实践，强化了学生对相关命令及其参数的理解与掌握。此外，通过使用 Wireshark 抓包工具，使学生了解相关协议的报文，对网络安全攻防技术的深入学习打下了坚实的基础。

报告评分：

指导教师签字：