实验三：数据包抓取与分析

1. 实验目的

学习安装、使用协议分析软件，掌握基本的数据报捕获、过滤和协议的分析技巧，能对抓取数据包进行分析。

2. 实验内容

协议分析软件的安装和使用、学会抓取数据包的方法并对对抓取数据包进行分析

3. 实验环境和要求

使用 Windows 操作系统；Internet 连接；抓包软件 Wireshark。

4. 注意事项

本章实验实施过程中应注意以下几个问题。

（1）涉及多台计算机的实验建议在虚拟机环境下完成，以降低对实验环境的要求。虚拟机环境可参考第 3 章的相关实验如果实验环境许可或不愿意使用虚拟机环境，这些实验也可以在真实环境下完成。

（2）实验中涉及的相关协议知识应在实验前向学生进行简单的介绍或回顾。

（3）在做协议分析实验的抓包操作之前应先设置过滤器(抓包过滤器或显示过滤器)。启动抓包后，要及时启动相关的网络应用软件进行操作(例如,要抓 HTTP 包，则应启动浏览器浏览网)，操作完成后再停止抓包，这样可以截获一个完整的会话过程，便于分析协议的交互过程(时序)。

（4）指导教师可预先抓取一定数量的样本数据包保存到文件中，供学生在实验中使用。或者从网上下载样本数据包：http://wiki.wireshark.org/samplecaptures

任务 3.1：网络协议分析软件 Wireshark 的使用

1. 实验目的：

（1）了解网络嗅探器软件（以 Wireshark 为例）的功能。

（2）了解网络嗅探器的过滤规则和设置方法

（3）掌握网络嗅探器软件的基本使用方法。

（4）会利用过滤规则设置抓取/显示特定的包。

2. 实验内容：

（1）安装配置网络嗅探器，了解嗅探器的命令菜单的功能。

（2）设置抓包过滤器和显示过滤器，抓取一个 HTTP 会话过程所传输的包观察包结构和格式，分析各字段的语义，以图形化方式显示该 HTTP 会话过程。

（3）仿照实验指导的步骤,利用过滤器抓取/显示一个完整的 POP3 会话(或显示样本文件中的一个完整的 POP3 会话)，将显示窗口进行截屏并粘贴到实验报告中，在实验报告中详细说明该会话中双方交换的各报文的含义。

（4）总结网络嗅探器都能做些什么。

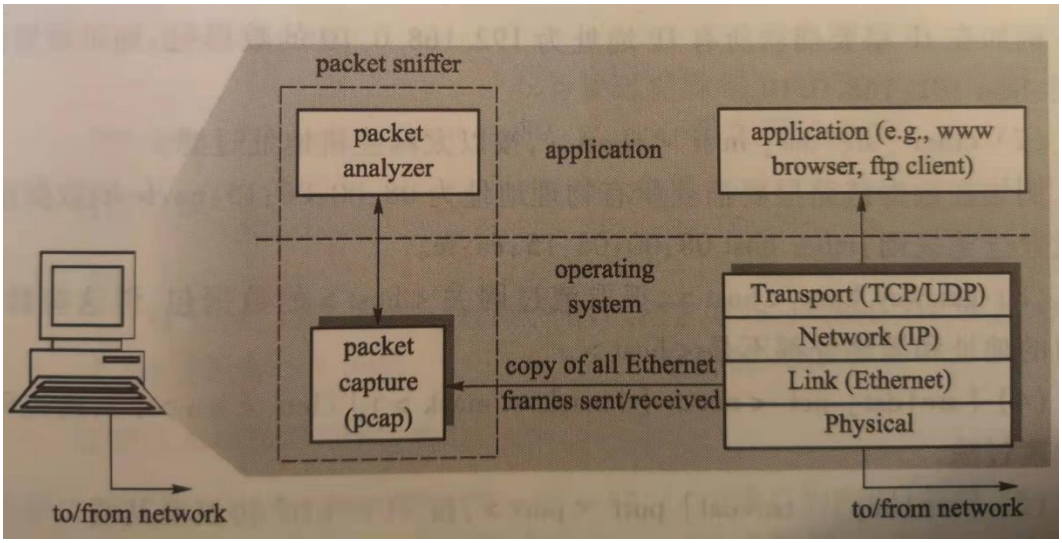
3. 实验条件：

（1）连接网络的计算机。

（2）Wirchark 软件

4. 实验分析：

网络嗅探器主要由包捕获器和包分析器两个部分组成，结构如图 2 所示。



图：网络嗅探器的结构

包捕获器用于接收每一个网络上(发送接收/经过)的链路层帧，并复制至内部缓冲区。而包分析器用于显示协议消息中的所有字段的内容。为了能做到这一点，包分析器必须“懂得”协议所交换的所有信息的结构。例如，如果要显示 HTTP 协议所交换的信息中的每一个字段的内容，包分析器首先解析以太网帧，抽取出其中的 IP 数据报。再对 IP 数据报进行解析，抽取出其中的 TCP 段。下一步再对 TCP 段进行解析，抽取出其中的 HTTP 报文。最后，将 HTTP 报文中的各个字段分解，如报文开始的 GET、POST、或 HEAD 字符串等。

网络中的协议数据包类型很多，为了只抓取那些感兴趣的数据包，就需要在

协议分析软件中设置抓包规则(包过滤器)。

Wireshark 的过滤规则有两种：**抓包过滤器**和**显示过滤器**。抓包过滤器用来设置抓包条件，即只抓取那些感兴趣的包。此过滤器用于抓包过程中。显示过滤器用来设置显示条件，即只显示那些感兴趣的包。此过滤器用于显示过程中。如果抓包时未设置抓包过滤器，将所有包都抓了回来，则显示时通常需要设置显示过滤器。否则抓回来的各种包混杂在一起，将会给协议包分析造成很大的困难。Wireshark 中的过滤器的语法规则格式为：

[not] 原语 [and | or [not] 原语…]

原语可以是以下形式中的一种。

（1）[src | dst] host <host>，按照 IP 地址或者名字过滤

例如：在 IP 层要捕获所有 IP 地址为 192.168.0.10 的数据包，则可设置过滤规则为：host 192.168.0.10。

（2）ether [src | dst] host <ehost>，按照以太网主机地址来过滤。

例如在数据链路层要捕获所有物理地址为 08:00:08:15:ca:fe 的数据包可设置过滤规则: ether host 088:00:08:15:ca:fe

（3）gateway host < host >，抓取流过网关<host>的数据包，但这些数据包的目的地址和源地址都不是<host>

（4）[src | dst] net <net> [{mask <mask> }|{len <len>}]，按掩码、掩码长度过滤。

（5）[tcp | udp] [src | dst] port <port>，按 TCP/UDP 协议及其端口号过滤。tcp | udp 必须出现在 src | dst 之前，如果省略 tcp | udp 则表示该地址所采用的协议既可以是 TCP，也可以是 UDP。

例如，要捕获所有端口号为 80 的 TCP 数据包，可设置过滤规则: tcp port 80。

（6）less | greater < length>，按数据包的长度过滤。

（7）ip | aether proto <protocol>，按协议过滤，既可以是以太协议，也可以是 IP。

（8）ether | ip broadcast | multicas，按以太网或者 IP 广播或多播过滤。

（9）<expr> relop <expr>，按数据包的字节或者字节范围创建复杂的过滤表达式。relop 为关系运算符，包括 eq、gt、lt、le、ge、ne 等。

例如，要捕获所有 IP 地址为 192.168.0.10 的所有非 HTTP 的数据包，可设置过滤规则：host192.168.0.10 and not tcp port 80。

以上规则中 src | dst 表示源地址“或者”目的地址，如果省略则表示所捕获的地址既可以是源地址，也可以是目的地址。

有关过滤器的详细介绍参见 http: //openmaniak.com//wireshark_filters.php 和 http: //www.tepdump.org/tcpdump_man.html.

下面是一些常用的过滤器实例。

1）过滤 IP 地址

ip.src eq 192.168.1.107or ip.dst eq192.168.1.107

ip.addr eq 192.168.1.107

2）过滤端口

tcp.port eq 80　//包括源和目的端口，过滤 UDP 协议时将 TCP 改为 UDP

tcp.port == 80 or udp.port eq 80

tcp.dstport == 80　//若要过滤源端口，则将 dstport 改为 srcport

tcp.port >= 1 and tcp.port <=80　　//指定端口范围

3）过滤协议

tcp

! arp（或者 not arp）　　// 排除 ARP

4）过滤 HTTP 模式

http.request.method == “POST”

http.request.url == “/image/logo-edu.gif”

http contains “GET”

http.request.method == “GET” && http contains “Host:”

5. 抓包的基本步骤如下：

（1） 启动 Wireshark 软件

（2） 选择网络接口

（3） 设置抓包过滤器

（4） 开始抓包

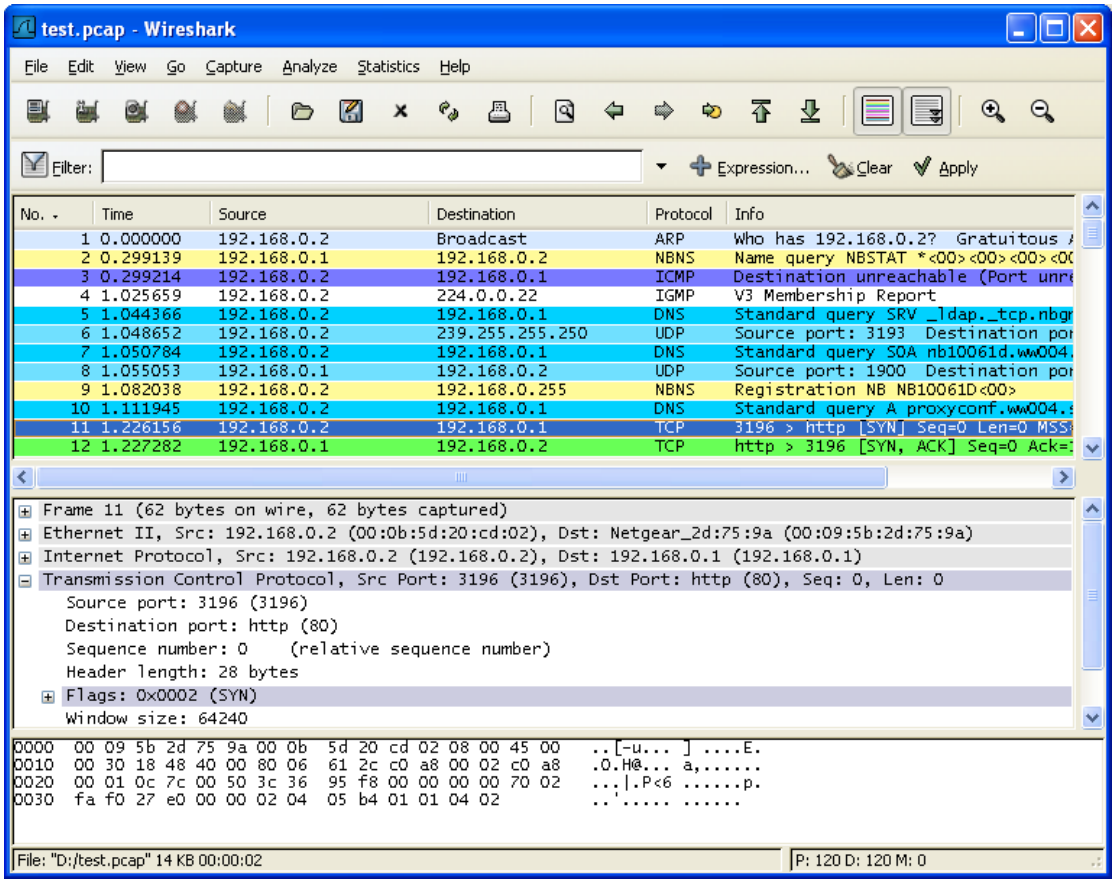
（5） 设置显示过滤器

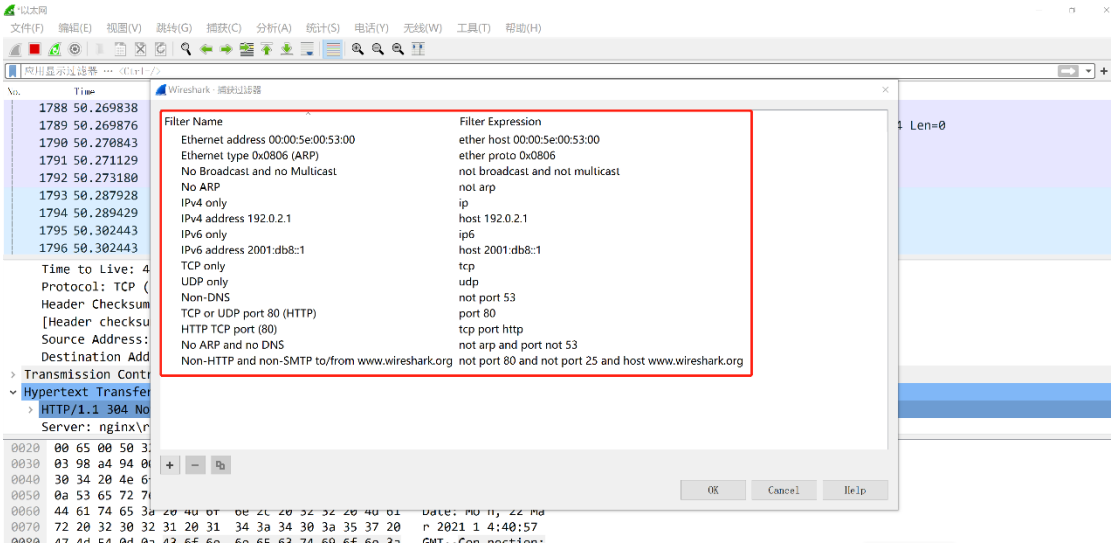
（6） 观察、分析所抓到的数据包

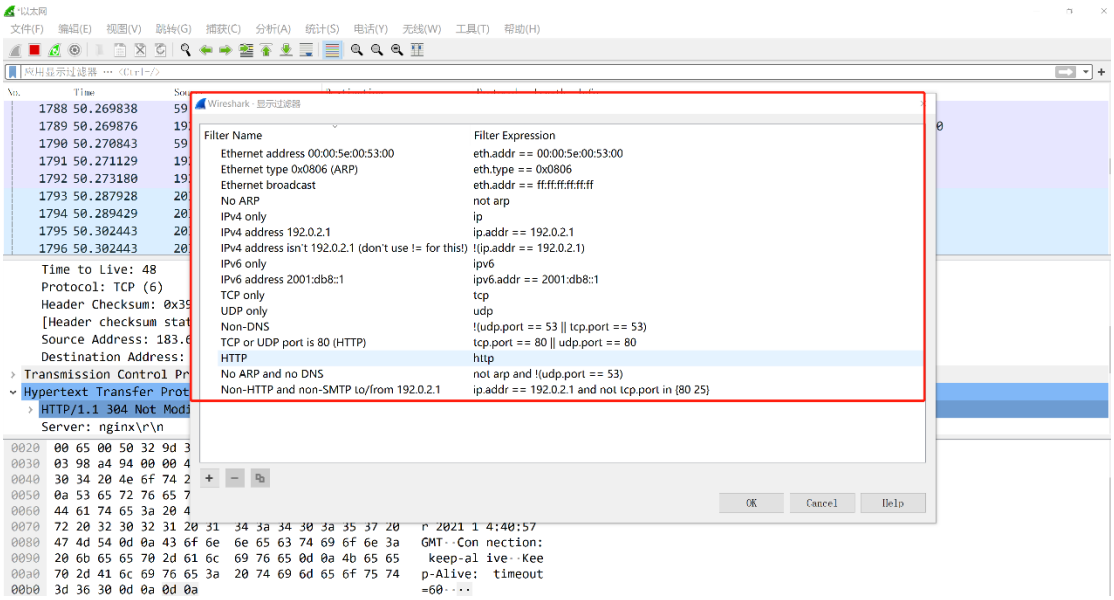
抓包结果将按时间顺序显示在上面的数据包列表窗口中，包括序号、发送时间、源地址、目的地址、协议等信息,其中源地址和目的地址是物理地址。单击表头中的标题，可以重新按照该标题排序。

在报文列表窗口中单击某个数据包时,在中间的协议窗口中将显示封包的各层协议：数据链路层帧及其首部、IP 数据包及其首部、UDP 数据包及其首部信息等。

在协议窗口中，单击指定协议或者协议的组成部分时，在下面的原始数据窗口中将高亮显示该协议或组成部分中所含数据的每个字节,窗口的左边显示的是十六进制数据,右边显示对应的 ASCI 码。图 2.17 为 Wireshark 的图形用户界面。图 2.18 为 Wireshark 的抓包过滤器设置窗口。图 2.19 为 Wireshark 的显示过滤器设置窗口。图 2.20 为在主界面中快速设置显示过滤器。图 2.21 为图形化协议分析一重建 HTTP 会话的结果显示。







任务 3.2：HTTP 协议的认知与分析

1. 实验目的

（1） 掌握浏览网页时 HTTP 协议的工作过程；

（2） 学会使用 Wireshark 分析 HTTP 协议。

2. 实验内容

（1）抓取一个 HTTP 会话所传输的报文，并分析报文的结构、格式及其内容

启动 Wireshark 抓包功能并定义过滤器，然后用浏览器打开一个网页（如：https://www.baidu.com/），然后再回到 Wireshark 抓包窗口，观察所抓到的 HTTP 报文，将显示的内容截屏，并粘贴到实验报告中，说明该报文中各字段的内容的

含义。

（2）观察一个完整的 HTTP 会话，并分析会话各个阶段的请求/响应操作。

（3）重建上述 HTTP 会话，并将截图粘贴到实验报告中，说明会话过程中各请求/响应报文的含义。

3. 根据上述实验内容回答以下问题

（1）你的浏览器运行的 HTTP 版本是 1.0 还是 1.1？服务器端的 HTTP 版本是什么？

（2）你的浏览器指出它所能够接受的语言是什么？

（3）你的计算机的 IP 地址是什么？你所浏览的网站服务器的 IP 地址是什么？

（4）服务器返回给浏览器的状态码（Status code）是什么？代表了什么含义？

（5）你所访问的 HTML 文件在服务器上最后的修改时间是什么？

（6）浏览器发出了多少个 HTTP GET 请求？

（7）观察浏览器中发出的第 1 个 HTTP GET 请求，其中是否含有“IF-MODIFIED-SINCE”这一行？

（8）观察服务器响应报文的内容，服务器是否显示地返回了所请求的文件内容？你如何解释这种现象？

（9）HTTP 客户端从服务器端获取网页对象使用的命令是什么？该命令每次能获取一个完整的网页吗？

（10）浏览一个网页需要进行几次 TCP 连接？连接次数与什么有关？

任务 3.3：TCP 协议的认知与分析

1. 实验目的

（1）掌握 TCP 协议的语法、语义和时序（操作流程）

（2）学会使用 Wireshark 分析 TCP 协议

2. 实验内容

1）抓取一个 TCP 会话所传输的报文，并分析报文的结构、格式及其内容

启动 Wireshark 抓包功能并定义过滤器，然后用浏览器打开一个网页（如：https://www.baidu.com/），然后再回到 Wireshark 抓包窗口，观察所抓到的 TCP 报

文，将显示的内容截屏，并粘贴到实验报告中，说明该报文中各字段的内容的含义。

2）捕获 TCP 协议通过三次握手建立连接过程完整数据包，截图粘贴到报告中，并分析该 TCP 连接的时序关系（注意：源 IP 地址与目的 IP 地址的匹配，特别是源 PORT 和目的 PORT 匹配）

3）在 TCP 的三次握手过程中，试分析 SYN、ACK、Sequence number 之间的关系

4）报文的序号以什么为单位？它在数据传输过程中的作用是什么？不同会话中报文的起始序号都是一样的吗？

5）请根据以上捕获到的 TCP 数据报，分析 TCP 报文组成格式（即包含哪些部分）