利用 Wireshark 观察网络报文

1.Wireshark 简介

Wireshark 是一个网络封包分析软件。网络封包分析软件的功能是抓取网络封包，并尽可能显示出最为详细的网络封包资料。

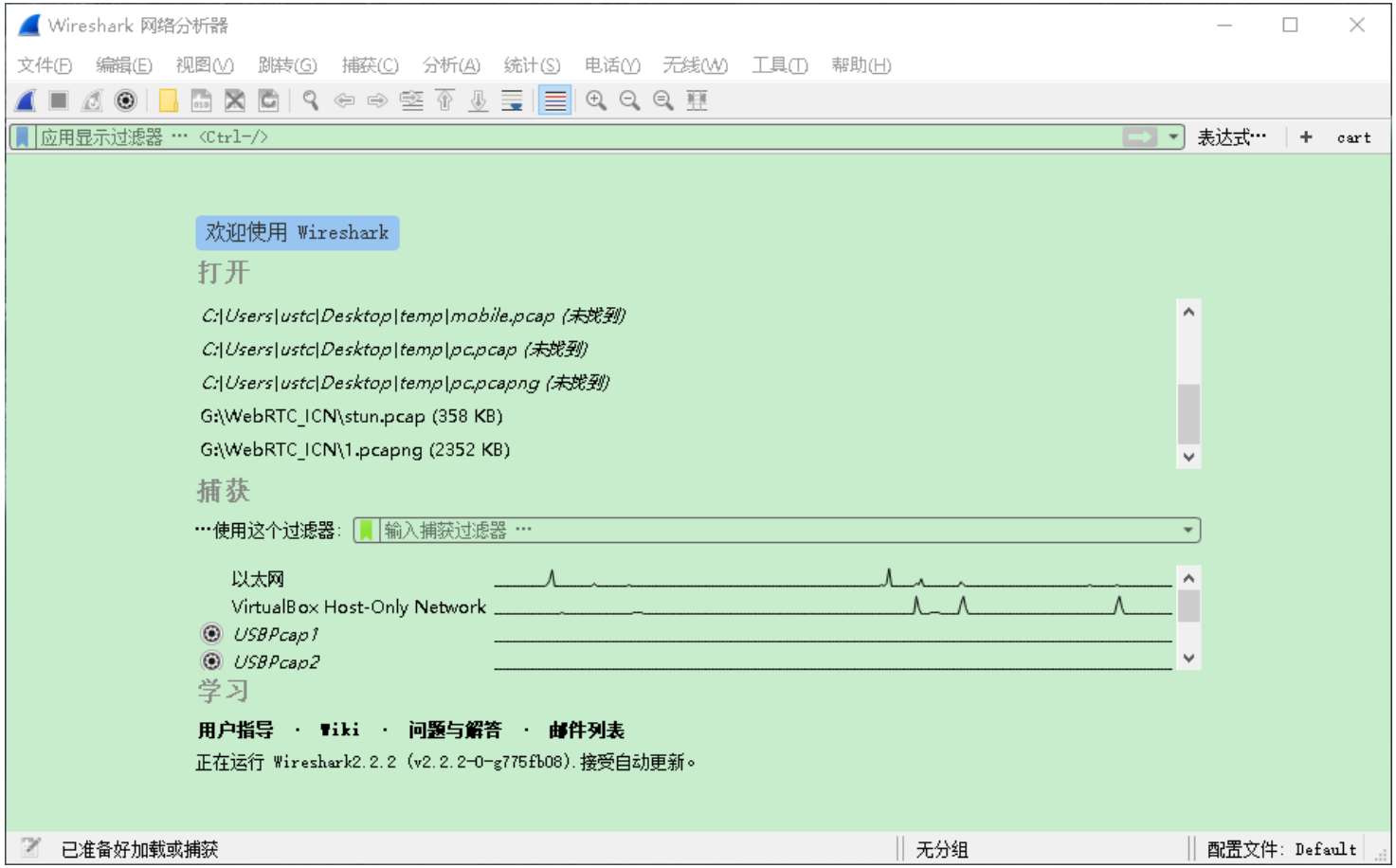
1.Wireshark 简介

抓包原理

- 网卡接收到一个报文后，会遍历系统中已经注册的 sniffer, 并调用其处理函数 .
- wireshark 通过注册一种虚拟的底层网络协议来获得报文副本 .

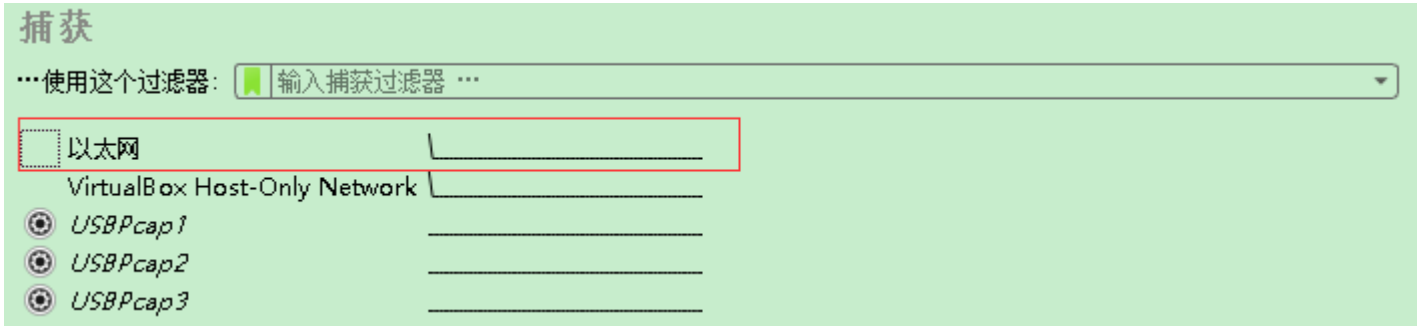
2. 打开 Wireshark

可在网上下载最新版本并安装。官网链接： https://www.wireshark.org/download.html



3. 选择捕获网卡

- wireshark 是捕获机器上的某一块网卡的网络包，当你的机器上有多块网卡的时候，你需要选择一个网卡。
- 在主界面上“捕获”一栏中即可选择，比如选择以太网。

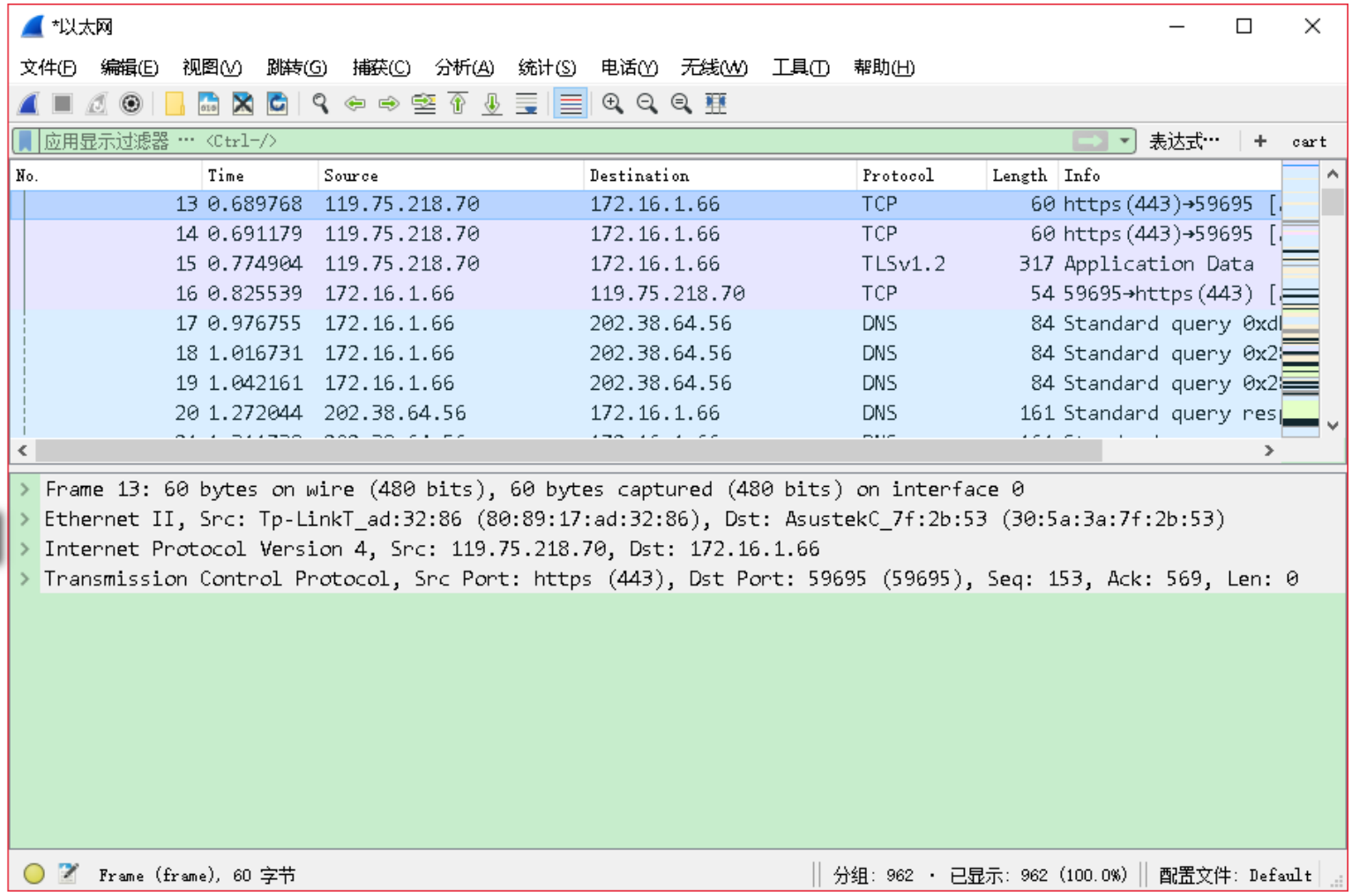


4.Wireshark 窗口介绍

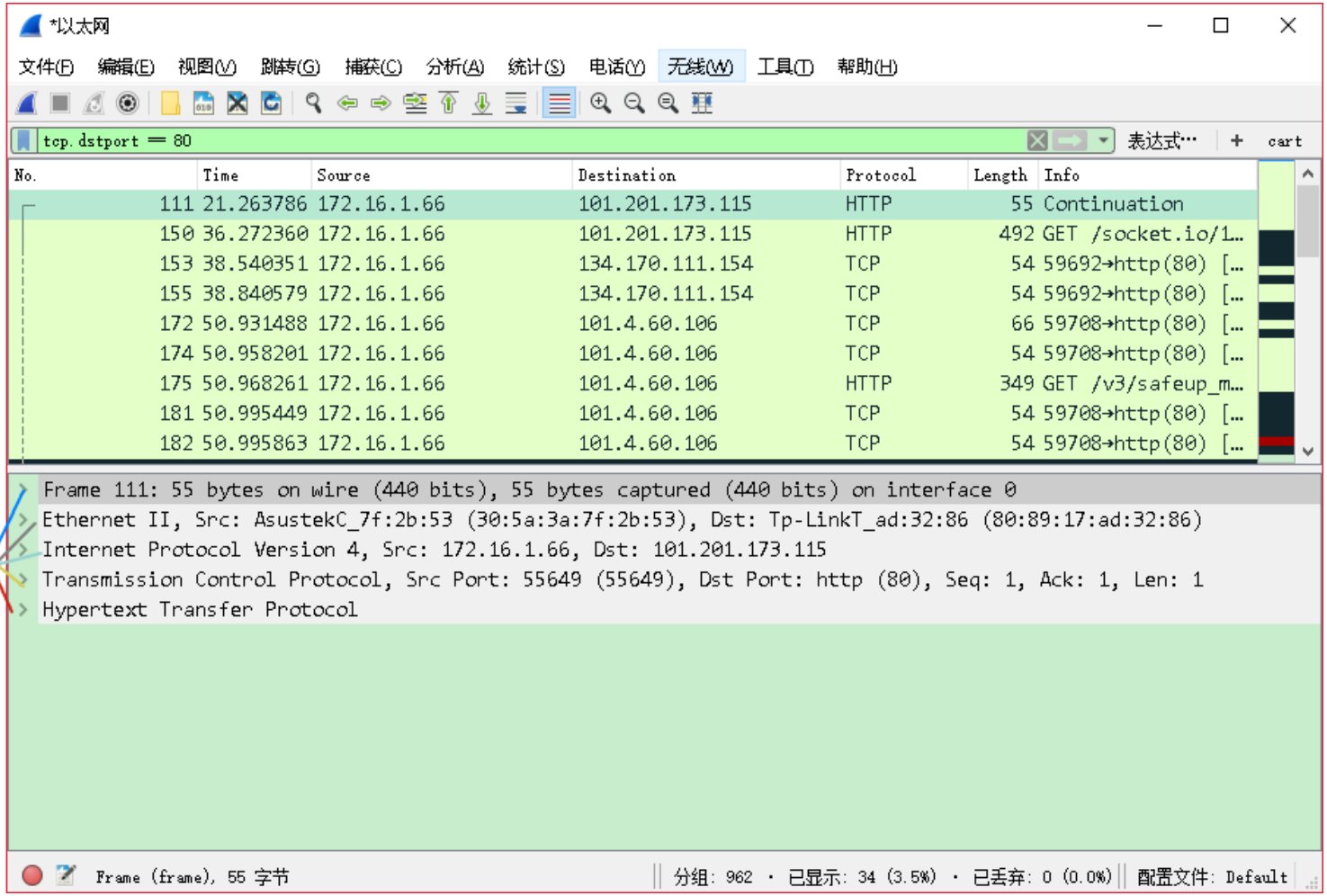
显示过滤器

封包列表

封包详细信息

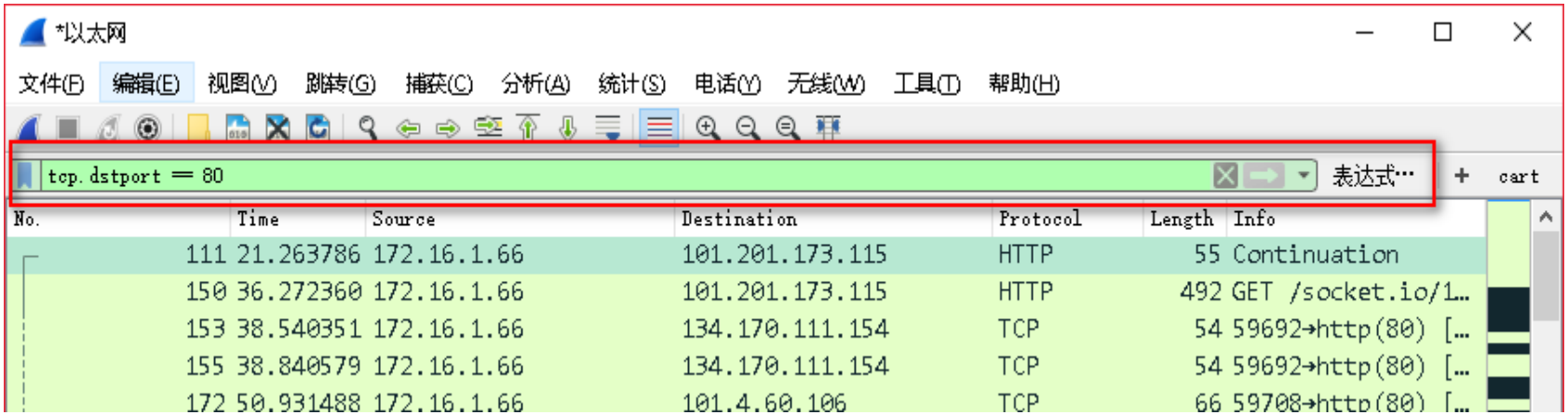


5.Wireshark 与对应的 TCP/IP 五层模型



6. 使用显示过滤器

- 有时候，我们会捕获到大量的冗余信息，有几千甚至几万条记录，以至于很难找到自己需要的部分。
- 所以就需要使用显示过滤器，来帮助我们过滤出我们需要的信息。
- 比如下图使用 tcp.dstport == 80 这个表达式来过滤出 TCP 目的端口是 80 的数据包。



6. 使用显示过滤器

比较操作符：
==　　eq
!=　ne
>　gt
<　lt
>=　　ge
contains
~　matches
&　bitwise_and

逻辑运算符
&&　and
||　or
^^　xor
!　not

注意 ip.addr != 172.16.1.111 的意义：
存在一个地址（源 ip 地址或目的 ip 地址）不等于 172.16.1.111.

6. 使用显示过滤器

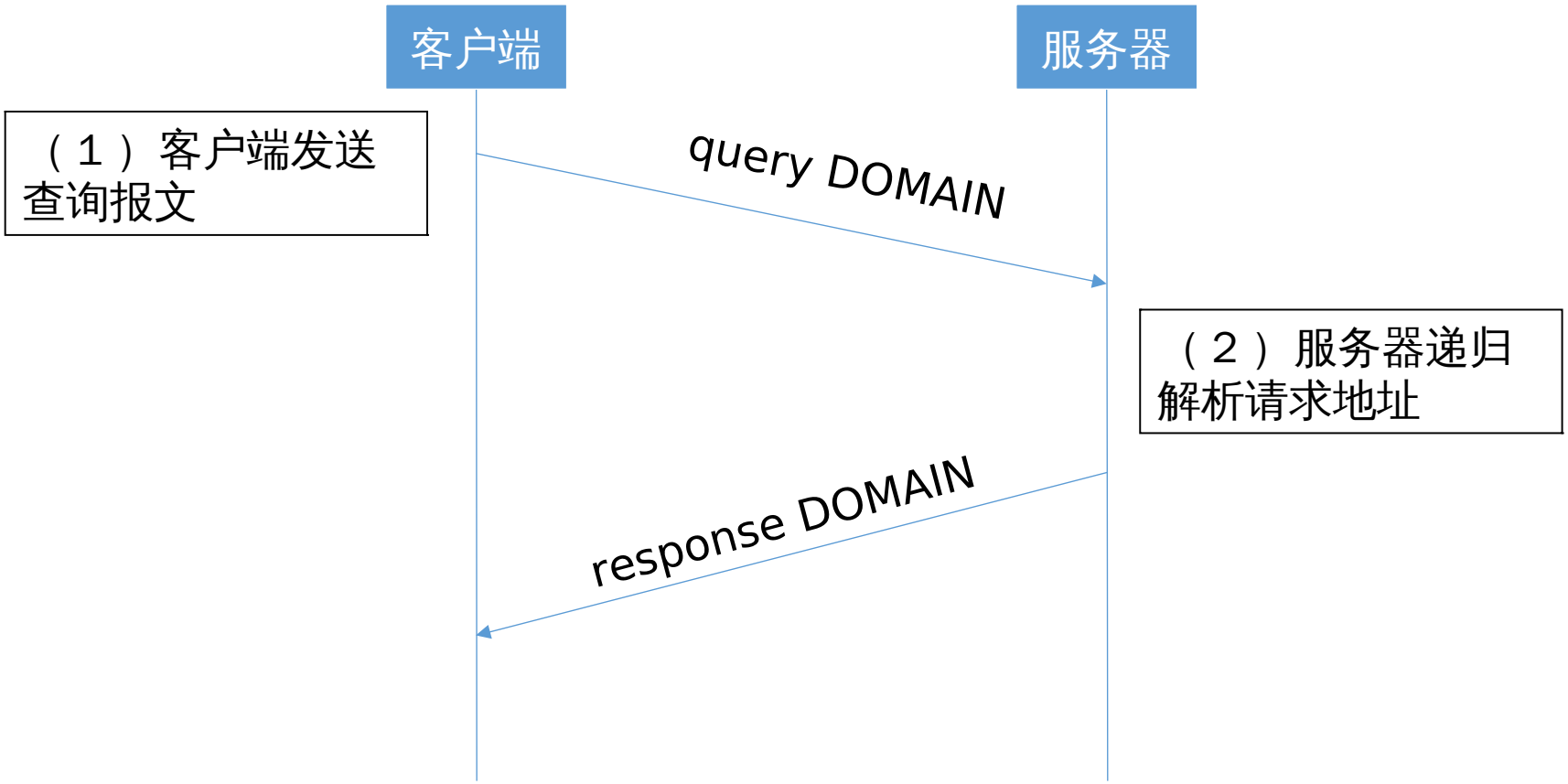
一些常用的过滤规则如下：

- 1. 协议过滤
  - 比如 tcp ，只显示应用 TCP 协议的数据包。
- 2.IP 过滤
  - 比如 ip.src == 192.168.1.102 ，显示源 IP 地址为 192.168.1.102 的数据包
  - ip.dst == 192.168.1.102，显示目标 IP 地址为 192.168.1.102 的数据包

6. 使用显示过滤器

- 3. 端口过滤
  - tcp.port == 80， 显示 TCP 协议端口为 80 的包
  - tcp.srcport == 80， 只显示 TCP 协议的源端口为 80 的数据包
- 4. HTTP 模式过滤
  - http.request.method==“GET”， 显示 HTTP GET 方法的包。
- 5. 逻辑运算符
  - ip.src ==192.168.1.102 or ip.dst==192.168.1.102 ，查看源 IP 地址或者目标 IP 地址是 192.168.1.102 的数据包
  - not 优先级最高，and 和 or 优先级相同，可以使用括号 .

7. 举例：观察 DNS 解析过程

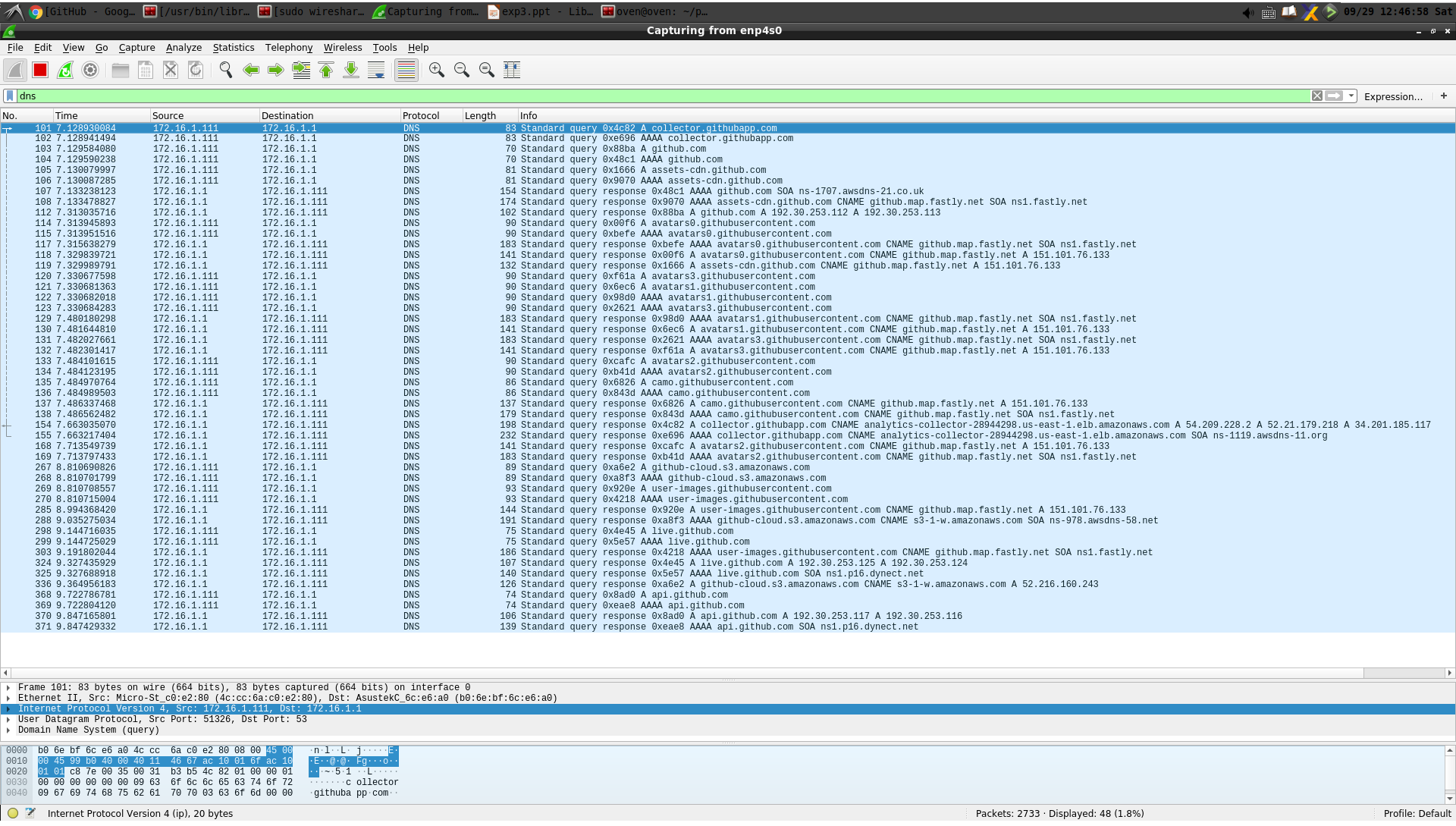


7. 举例：观察 DNS 解析过程

- 使用 nslookup 命令进行 DNS 查询或者直接在浏览器访问 .

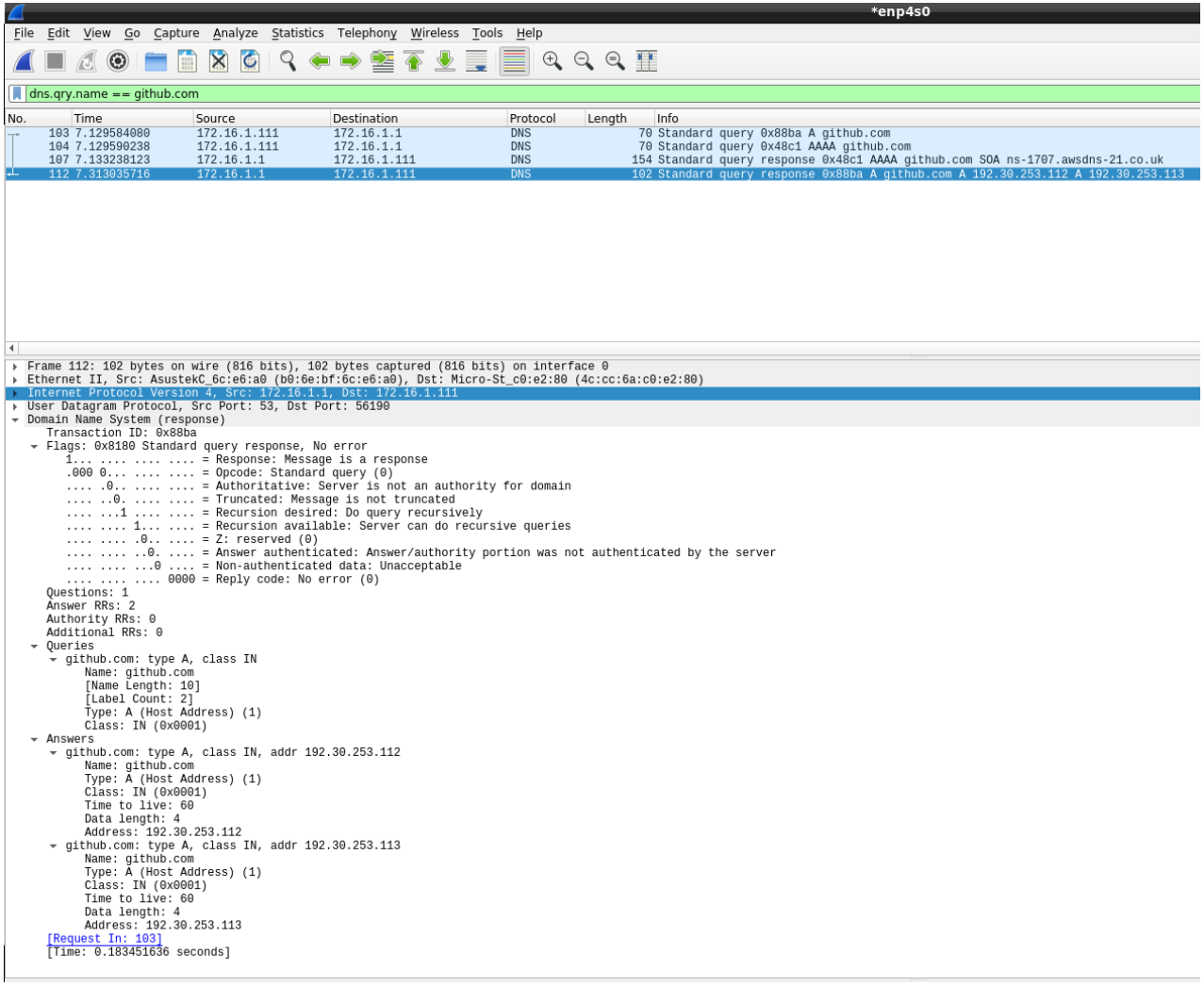
```
# oven @ oven in ~ [12:24:16]
$ nslookup github.com
Server:         172.16.1.1
Address:        172.16.1.1#53

Non-authoritative answer:
Name:   github.com
Address: 192.30.253.113
Name:   github.com
Address: 192.30.253.112


# oven @ oven in ~ [12:24:24]
$
```

7. 举例：观察 DNS 解析过程



7. 举例：观察 DNS 解析过程

dns.qry.name == github.com



8. 实验内容

利用 wireshark 观察在浏览器访问 https://www.github.com 时的交互过程 .

实验要求

(1) 保存抓包结果为文件，文件名为”学号 + 姓名 +wireshark+1+cap.pcap”. (5%)

(2) 分析对该网址的 DNS 解析过程 .

- 使用显示过滤器，使之只显示该过程中的报文 . 给出你定义的显示过滤器和过滤后的截图 (包含整个 wireshark 窗口). (10%)
- 对显示的每个数据包给出解释，格式为” No: 解释”. 解释为 : 查询”域名”或回复” No” 的查询，回复内容为” ip 列表” . (No 为 wireshak 中数据包的” No.” 列的值) (10%)
- 交互过程中可能出现多次 DNS query，给出其原因 . (30%)

8. 实验内容

(3) 分析 https 握手过程 .

- 使用显示过滤器，使之只显示与 https://www.github.com 交互的 https 报文 . 给出你定义的显示过滤器和过滤后的截图 (包含整个 wireshark 窗口). (15%)
- 参照第 12 页的交互图，画出握手过程的交互图 . 给出每次交互对应的数据包的 No. (30%)

(4) 注意

- 访问过程中，页面中有的元素需要从其他位置获取，因此需要访问其他域名，如从 https://avatars0.githubusercontent.com 获取用户头像 . 这种情况不需要考虑 .
- 截图时，如果报文过多，只需要显示最开始的若干报文 .
- 步骤 (2), (3) 的结果以一个 word 文档的形式提交，文件名为”学号 + 姓名 +wireshark+1+report.docx”.
- wireshark 抓包实验包含多个部分，本次为第一个部分 . (1)(2)(3) 中的百分比为该项占此部分实验分值的百分比 .
- 将抓包文件和 word 文档放在一个目录下，目录名为”学号 + 姓名 + 实验 4.1”，将此目录压缩为 zip 再提交 . 请在 2018 年 11 月 11 日之前 (不包含 11 月 11 日) 提交 .