華北水利水電大學

North China University of Water Resources and Electric Power

计算机网络

实验七：网络命令的使用 2

姓　名：高树林

学　号：202018526

专　业：人工智能

院　系：信息工程学院

一、实验目的

1、掌握常用的 TCP/IP 协议族的功能
2、掌握 WireShark 工具的使用
3、理解 TCP、IP、UDP、ICMP 协议数据包的首部结构

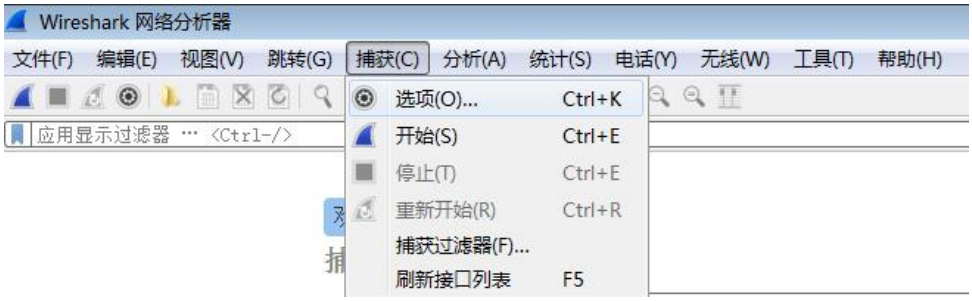
二、实验内容

1、IP 报文结构分析
2、TCP 数据段结构分析
3、UDP 数据段结构分析
4、ICMP 报文分析

三、实验过程

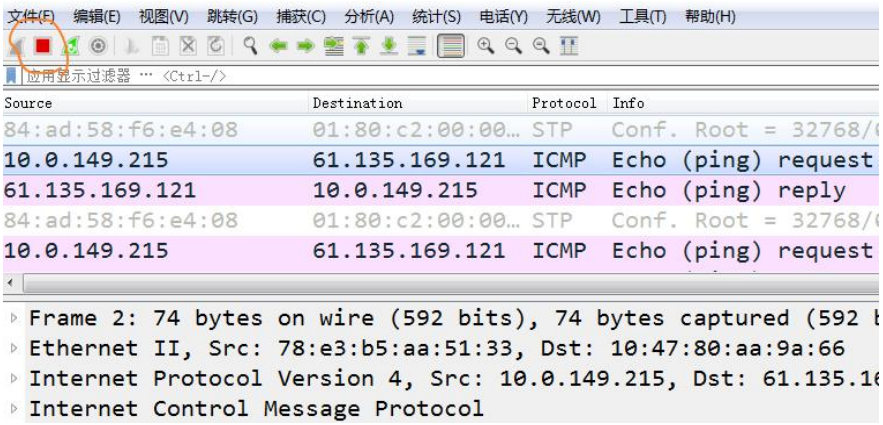
1、WireShark 工具的使用

第一步，打开 wireshark 后，从菜单栏选择“捕获”

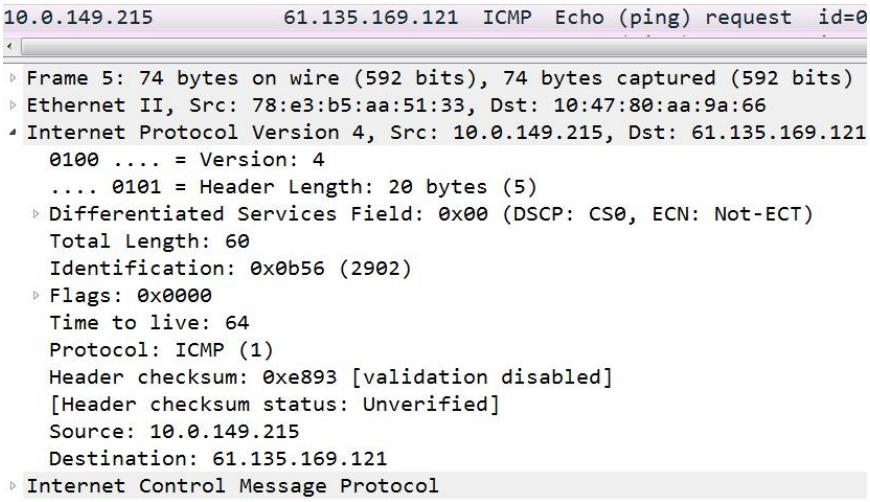


第二步，下拉菜单，选择“选项”，从而打开以下对话框，选中接口（网卡，列表中包括硬件/软件（虚拟）网卡），单击开始，就会捕获通过这个接口的数据帧。

第三步，开始捕获后，中部窗口中显示捕获到的数据帧。在工具栏有一个停止按钮，单击它停止捕获。下图中圈出了停止按钮。



第四步，选中感兴趣的数据帧，在下部窗口中查看数据帧的解析信息。下图选中了一个 ICMP 请求数据报。

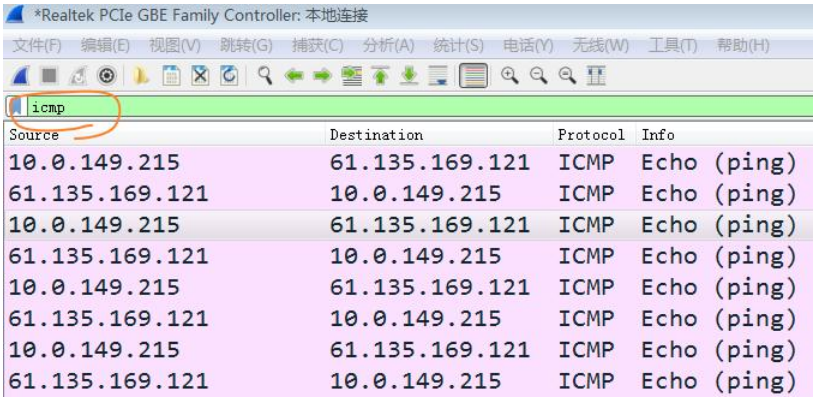


2、产生数据帧并捕获数

使用 ping 命令来产生 ICMP 请求及响应，比如“ping 114.114.114.114”；
使用 nslookup 命令来产生 UDP 数据段（DNS 请求及响应），比如“nslookup www.baidu.com”；
使用浏览器访问网页来产生 TCP 数据段（HTTP 请求及响应）。

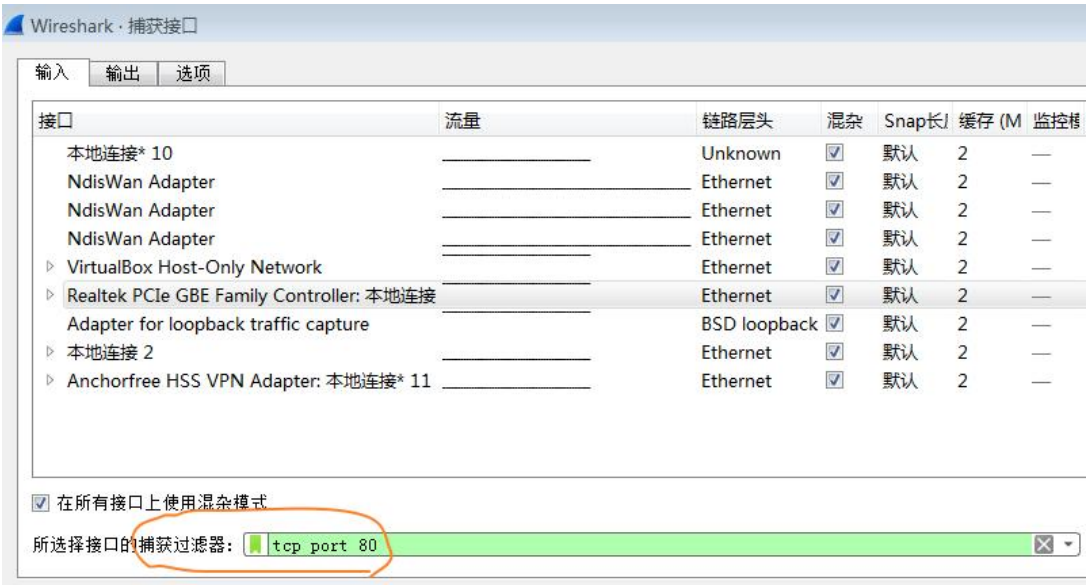
3、过滤捕获到的数据帧

如图所示，输入过滤规则“icmp”并回车，将只显示 ICMP 数据报。同理，清空过滤规则并回车，将显示所有的数据帧。



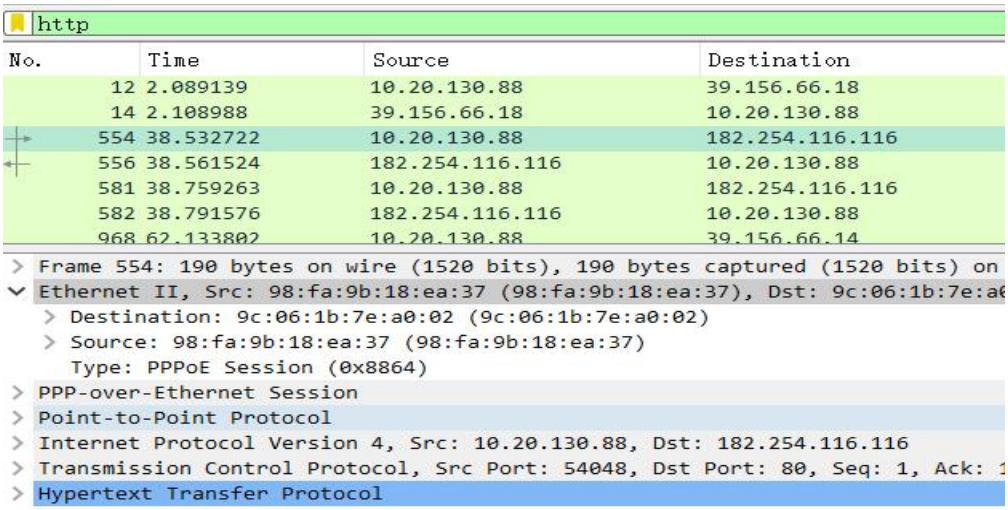
4、只捕获感兴趣的数据帧

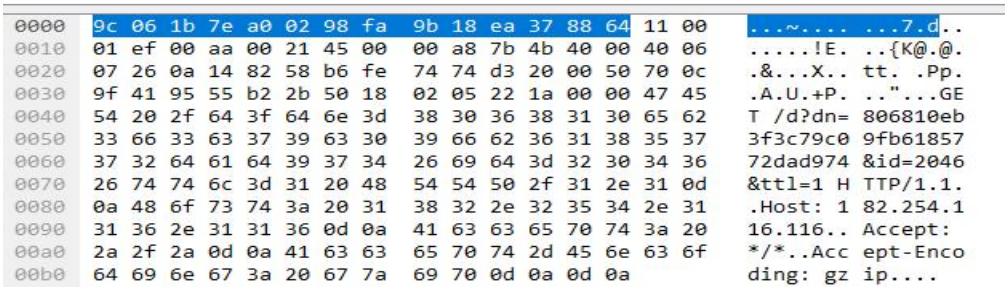
如图所示，选中捕获接口后，输入捕获过滤器规则“tcp port 80”，将只捕获源端口或目的端口为 80 的 TCP 数据段。适用于数据帧较较多，wireshark 逐一捕获拖慢系统运行速度的场合。



四、实验结果

1、以太网帧分析





在发送 HTTP 请求报文时，肯定是从主机发往远端 Web 服务器，该服务器并不在我主机的局域网上，因此在网络层，主机通过查路由表发现无法直接交付，所以将 IP 数据报发往默认路由，下一跳地址肯定为默认路由的 IP 地址，而该 IP 地址又通过 ARP 协议转为默认路由的 MAC 地址填入以太网帧的目的地址中。因此，在数据链路层，源 MAC 地址一定为主机的 MAC 地址，而目的 MAC 地址则为默认路由的 MAC 地址。

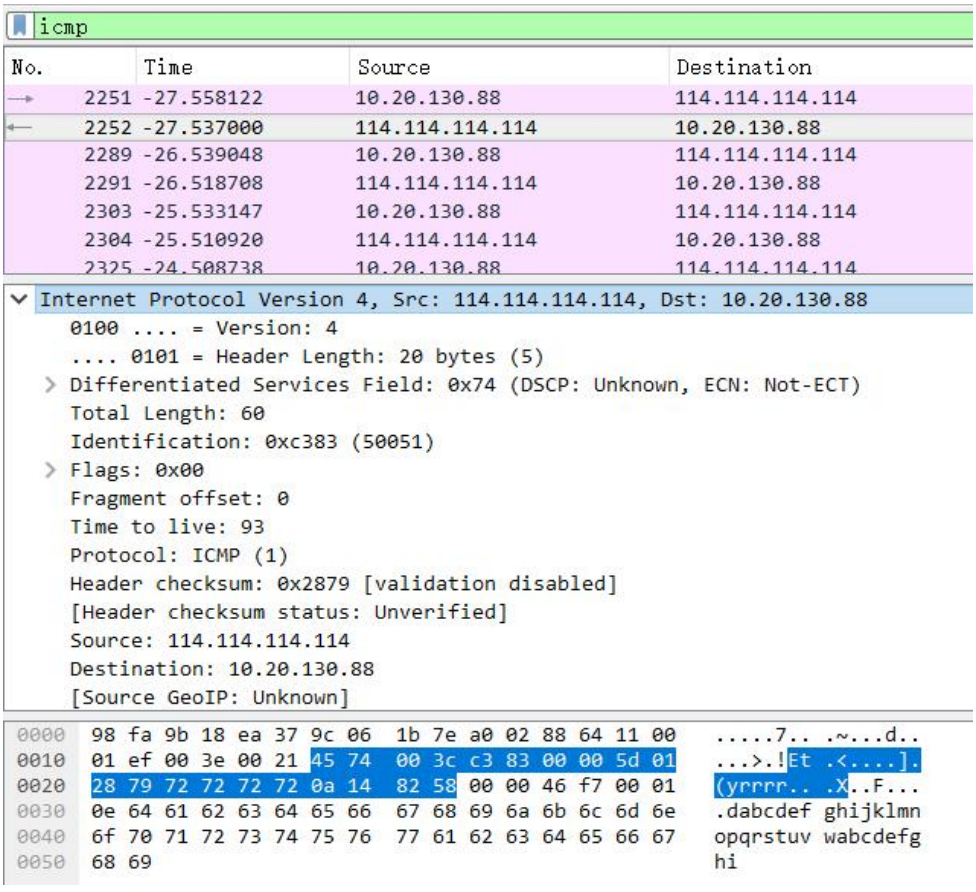
在命令行中输入 ipconfig /all，由下图可以看出，主机的 MAC 地址确实为 98-FA-9B-18-EA-37，与预期相符

```
  自动配置已启用. . . . . . . . . . . : 是

以太网适配器 以太网:

  连接特定的 DNS 后缀 . . . . . . . :
  描述. . . . . . . . . . . . . . . . : Realtek PCIe GbE Family Controller
  物理地址. . . . . . . . . . . . . . : 98-FA-9B-18-EA-37
  DHCP 已启用 . . . . . . . . . . . . : 是
  自动配置已启用. . . . . . . . . . . : 是
  本地链接 IPv6 地址. . . . . . . . . : fe80::fd94:969b:8601:b214%6(首选)
  自动配置 IPv4 地址  . . . . . . . . : 169.254.178.20(首选)
  子网掩码  . . . . . . . . . . . . . : 255.255.0.0
  默认网关. . . . . . . . . . . . . . :
```

2、ICMP 数据报分析



IP 报文版本号是 IPV4；
首部长度：20 bytes；
数据包总长度：60；
标示符：0xc383；

寿命：93；
上层协议：ICMP；
首部校验和：0x2879，并且是正确的；
源 IP 地址：114.114.114.114；
目的 IP 地址：10.20.130.88；

```
0000  98 fa 9b 18 ea 37 9c 06  1b 7e a0 02 88 64 11 00   .....7.. .~...
0010  01 ef 00 3e 00 21 45 74  00 3c c3 83 00 00 5d 01   ...>.!Et .<....
0020  28 79 72 72 72 72 0a 14  82 58 00 00 46 f7 00 01   (yrrrr.. .X..F
0030  0e 64 61 62 63 64 65 66  67 68 69 6a 6b 6c 6d 6e   .dabcdef ghijklmn
0040  6f 70 71 72 73 74 75 76  77 61 62 63 64 65 66 67   opqrstuv wabcdefg
0050  68 69                                             hi
```

类型：0（回显请求）　　代码/编码：0；
校验和：0x46f7（正确的校验和）；
标示符（大端顺序）：1（0x0001）；
标示符（小端顺序）：256（0x0100）；
序列号（大端顺序）：3684（0x0645）；
序列号（小端顺序）：25614(0x4506)；

3、IP 头部分析

ip.addr == 111.7.164.78

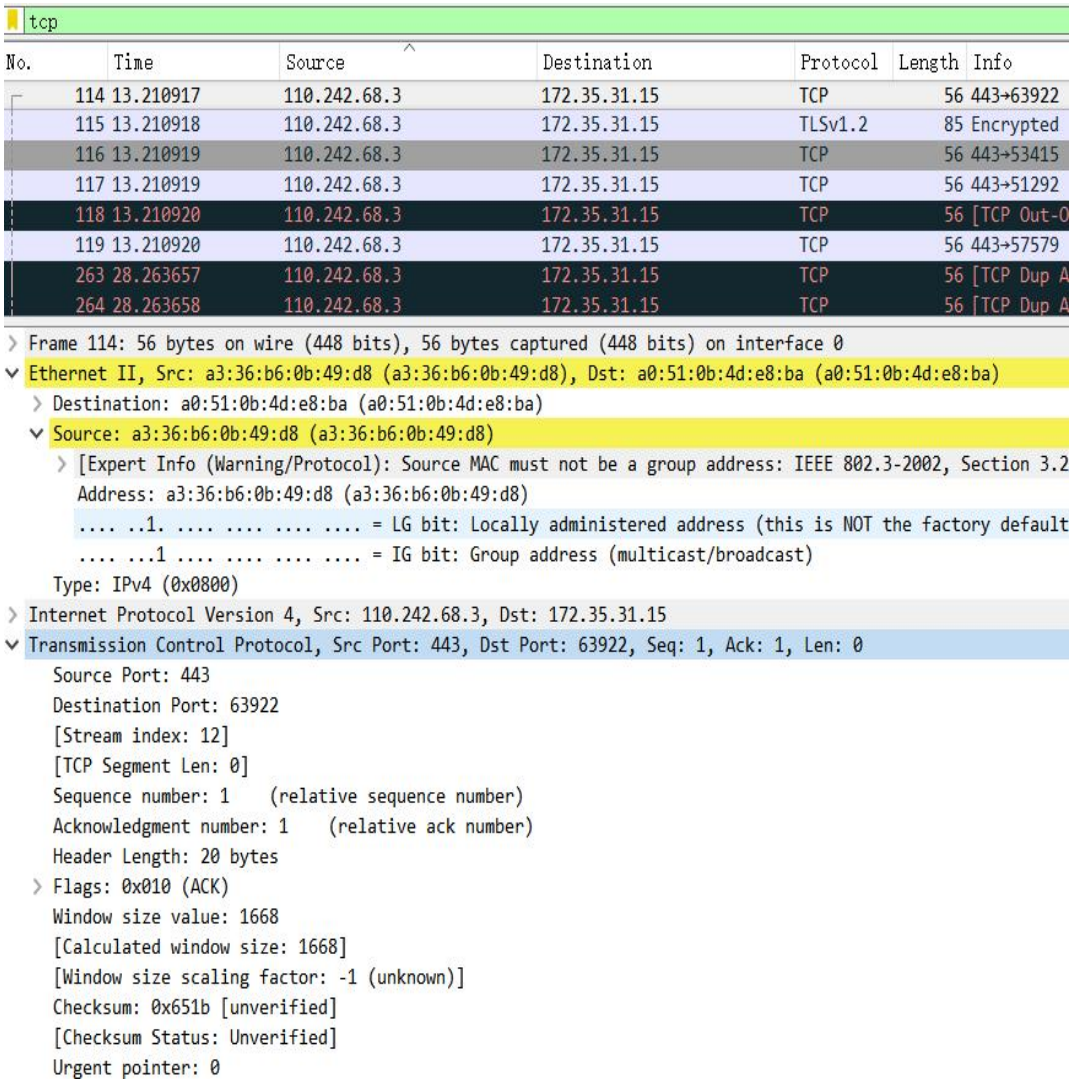
No.	Time	Source	Destination
1411	5.744365	10.20.130.88	111.7.164.78
1412	5.744406	10.20.130.88	111.7.164.78
1413	5.746730	111.7.164.78	10.20.130.88
1421	5.748098	111.7.164.78	10.20.130.88
1422	5.748495	111.7.164.78	10.20.130.88
1423	5.748531	10.20.130.88	111.7.164.78
1477	5.858781	10.20.130.88	111.7.164.78

```
> Point-to-Point Protocol
v Internet Protocol Version 4, Src: 111.7.164.78, Dst: 10.20.130.88
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x04 (DSCP: Unknown, ECN: Not-ECT)
    Total Length: 1480
    Identification: 0xdf78 (57208)
  > Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 56
    Protocol: TCP (6)
    Header checksum: 0xbdf1 [validation disabled]
    [Header checksum status: Unverified]
    Source: 111.7.164.78
    Destination: 10.20.130.88
```

```
0010  01 ef 05 ca 00 21 45 04  05 c8 df 78 40 00 38 06   .....!E. ...x@.8.
0020  bd f1 6f 07 a4 4e 0a 14  82 58 01 bb fe 88 ba 21   ..o..N.. .X.....!
0030  5a 19 79 00 f8 92 50 10  00 a4 f1 b9 00 00 17 03   Z.y...P. ........
0040  03 12 34 b5 17 c0 a5 9d  d0 f3 c7 c1 75 7a 12 94   ..4..... ....uz..
0050  d1 41 78 66 14 88 bb eb  54 7e b1 68 d1 f4 c8 f0   .Axf.... T~.h....
0060  73 c2 31 83 6a 1d be 4f  33 2b 3b b1 3e 0c fd 76   s.1.j..O 3+;.>..v
0070  76 db 25 bf ec e4 3c 39  6a 6a 09 ca ef 79 80 dd   v.%...<9 jj...y..
0080  07 a2 c9 52 10 c5 a4 72  fa a0 ac f6 42 db b9 3e   ...R...r ....B..>
0090  08 81 89 83 62 89 d5 9b  62 8b d6 87 7c e4 1d fe   ....b... b...|...
00a0  81 cd 34 2c fa fd 93 1b  a3 b3 e9 65 1f 97 7d 57   ..4,.... ...e..}W
```

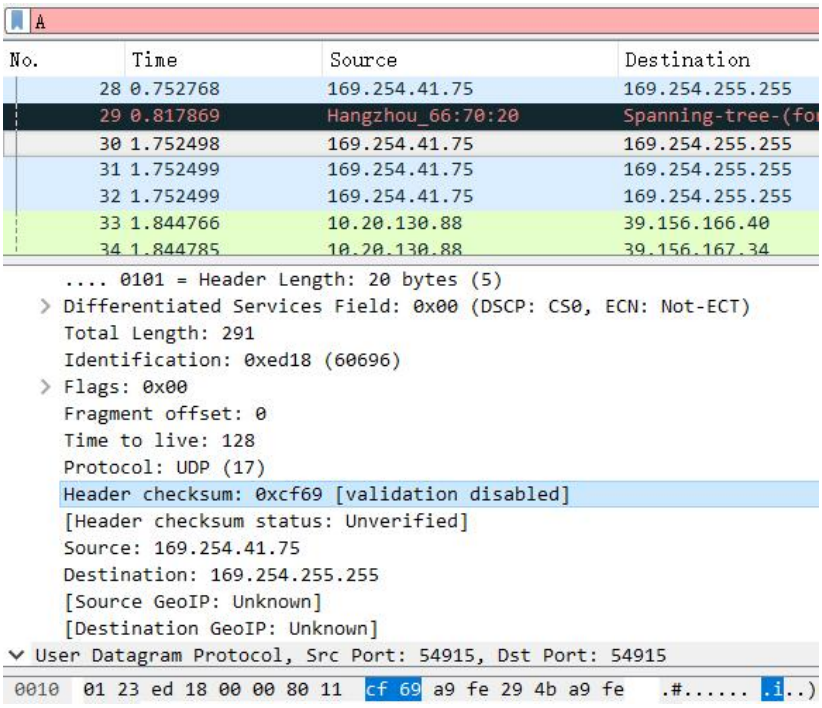
IP 报文版本号是 IPV4；
首部长度：20 bytes；
数据包总长度：1480；
标示符：0xdf78；
寿命：56；
上层协议：TCP；
首部校验和：0xbdf1，并且是正确的；
源 IP 地址：111.7.164.78；
目的 IP 地址：10.20.130.88；

4、TCP 头部分析



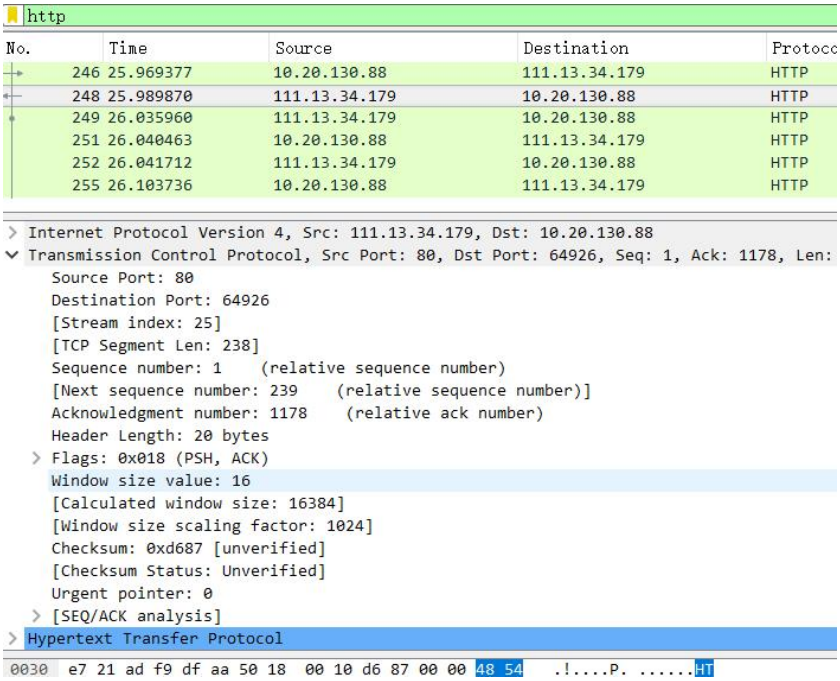
源端口号：443
目的端口号：63922
Sequence Number：发送序列号
Acknowledgment Number：确认序列号
Flags：SYN-同步序列号
Window size value：窗口大小
Checksum：检验和
Urgent potiner：紧急指针

5、UDP 头部分析



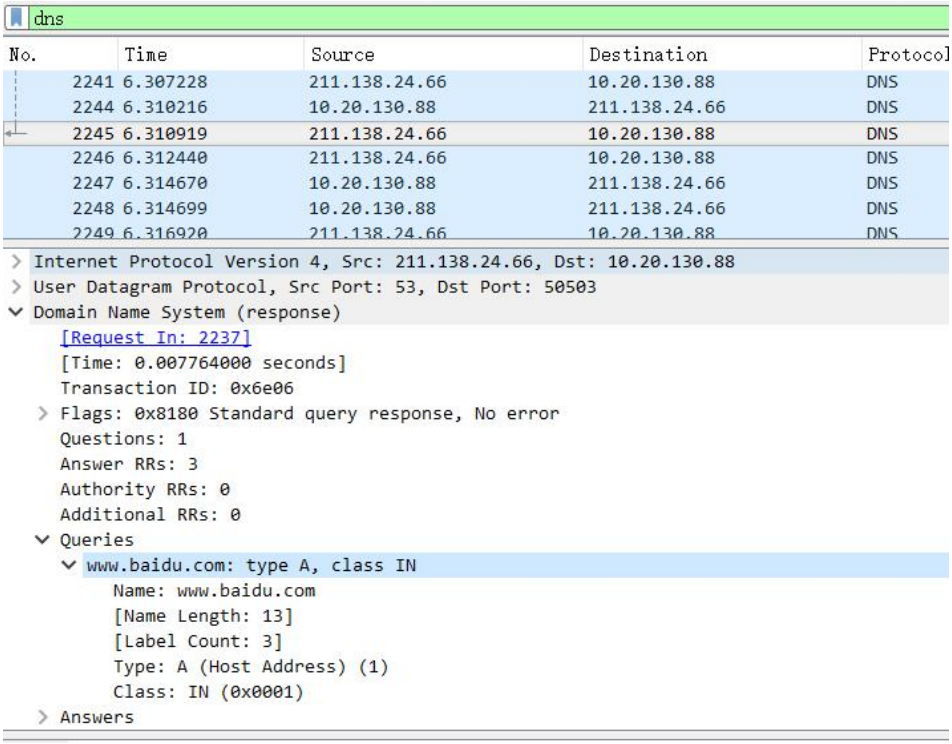
IP 报文版本号是 IPV4；
首部长度：20 bytes；
数据包总长度：291；
标示符：0xed18；
寿命：128；
上层协议：UDP；
首部校验和：0xcf69，并且是正确的；
源 IP 地址：169.254.41.75；
目的 IP 地址：169.254.255.255；

6、HTTP 分析

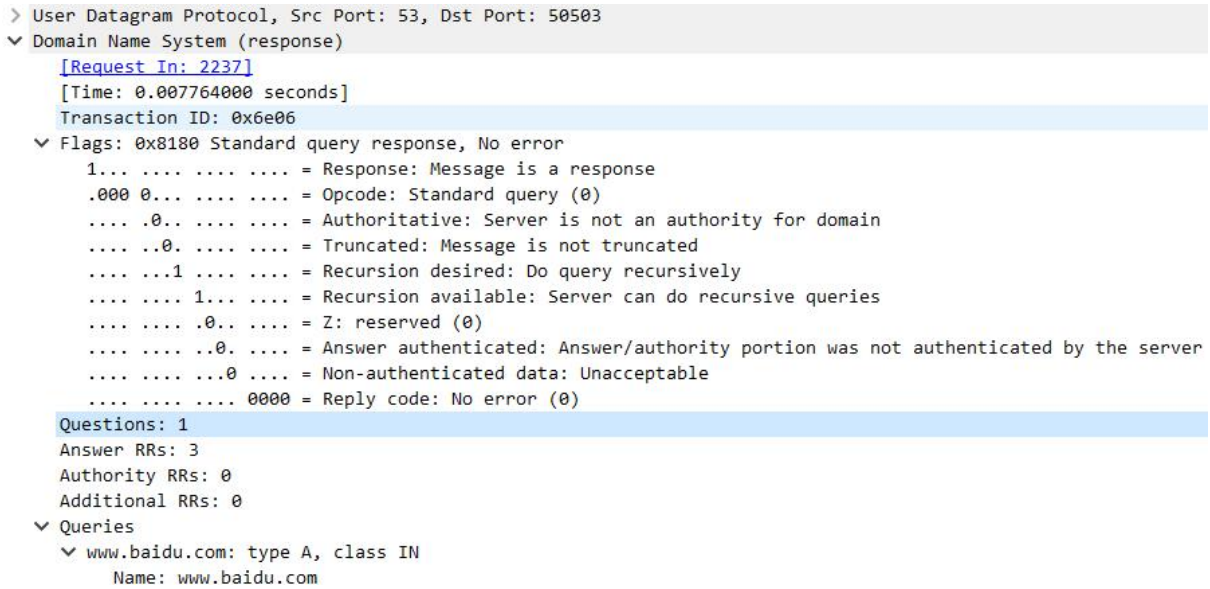


http1.1 20 请求版本
源端口：80；
目的端口：64926；
TCP 段长度：238；
确认号：1178；
紧急指针：0；

7、DNS 分析



请求的域名为：www.baidu.com
域名类型为 A（主机地址）
地址类型为 IN（互联网地址）



问题计数是 1 个，而回答计数 3 个，域名服务器计数 0 个，额外记录计数 0 个。

```
Additional RRs: 0
Queries
  www.baidu.com: type A, class IN
      Name: www.baidu.com
      [Name Length: 13]
      [Label Count: 3]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
Answers
  www.baidu.com: type CNAME, class IN, cname www.a.shifen.com
      Name: www.baidu.com
      Type: CNAME (Canonical NAME for an alias) (5)
      Class: IN (0x0001)
      Time to live: 930
      Data length: 15
      CNAME: www.a.shifen.com
  www.a.shifen.com: type A, class IN, addr 39.156.66.18
      Name: www.a.shifen.com
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 282
      Data length: 4
      Address: 39.156.66.18
  www.a.shifen.com: type A, class IN, addr 39.156.66.14
      Name: www.a.shifen.com
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 282
      Data length: 4
      Address: 39.156.66.14
```

请求区域域名为：www.baidu.com
回答 1：类型为 CNAME 代表别名，别名为：www.a.shifen.com
回答 2：类型为主机地址，IP 为：39.156.66.18
回答 3：类型为主机地址，IP 为：39.156.66.14

五、实验心得

学习利用 Wireshark 捕捉访问网页的全过程给我带来了许多收获。以下是我得到的一些主要收获：

深入了解网络通信：通过 Wireshark 捕捉和分析访问网页的全过程，我能够深入了解网络通信的细节和流程。我学到了不同协议之间的交互，例如 TCP/IP、HTTP、TLS/SSL 等。这使我对网络通信的工作原理和数据传输方式有了更清晰的认识。

网络故障排除能力提升：Wireshark 提供了一个全面的网络分析平台，我能够利用它来识别和解决访问网页中可能出现的故障。我学会了如何分析数据包，检查网络连接的问题，查找延迟和错误的原因。这对于快速诊断和解决网络故障非常有帮助。

性能优化的洞察力：Wireshark 捕捉的数据包可以让我了解网页访问过程中的性能问题和优化机会。通过分析数据包的时序和延迟，我可以评估网页加载速度，并找出可能导致延迟的原因，例如网络延迟、服务器响应时间、资源加载等。这

帮助我优化网页的性能，提供更好的用户体验。

安全问题的分析：通过 Wireshark 的分析功能，我可以识别和分析网页访问过程中的安全问题。我可以观察到 HTTP 请求和响应的头部信息，检查是否存在潜在的安全漏洞，例如明文传输敏感数据、缺乏加密等。这使我能够采取相应的安全措施来保护网页和用户数据的安全。

实际应用案例的学习：通过分析捕获的数据包，我可以学习实际应用案例，了解各种优秀网页设计和优化策略。我可以观察到网页中使用的各种技术和工具，如 CDN、缓存机制、压缩等，从中学习并应用于自己的项目中。

总的来说，学习利用 Wireshark 捕捉访问网页的全过程使我对网络通信有了更深入的了解，提高了我的故障排除能力和性能优化能力。它还帮助我更好地理解和应用网络安全措施，以保护网页和用户的数据安全。这种实验方法为我提供了宝贵的实践经验和洞察力，对我的个人成长和职业发展都有着积极的影响。具体收获如下：

提升了网络分析技能：通过使用 Wireshark 捕捉访问网页的全过程，我得到了更多实践经验，提升了我的网络分析技能。我学会了如何解读和分析数据包，识别网络问题，并能够进行深入的故障排除和性能优化。

加深对网络协议的理解：Wireshark 捕捉到的数据包展示了不同网络协议之间的交互和通信过程。通过分析这些数据包，我对 TCP/IP、HTTP、TLS/SSL 等协议的工作原理和细节有了更深入的理解。这对于理解网络架构和进行网络设计非常有帮助。

强化了安全意识：通过 Wireshark 捕捉访问网页的全过程，我能够观察到潜在的安全风险和漏洞。这使我更加关注网络安全，并提高了我的安全意识。我学会了识别和应对常见的安全问题，采取适当的安全措施来保护网络和用户数据的安全性。

增加了解决问题的能力：Wireshark 捕捉到的数据包可以帮助我分析和解决网络问题。通过观察和分析数据包，我能够快速定位问题的根源，并采取相应的措施进行故障排除。这培养了我的问题解决能力和分析思维。

学习了优化网络性能：通过分析捕捉到的数据包，我能够评估和优化访问网页的性能。我学会了识别和解决网络延迟、带宽限制、资源加载等问题，以提供更快

速和高效的用户体验。

提升了职业竞争力：具备网络分析技能和使用 Wireshark 的能力使我在职业领域中更具竞争力。这种技能和经验对于网络工程师、网络管理员、网络安全专家等职业非常有价值，能够为我在职业发展中带来更多机会。

总的来说，通过学习利用 Wireshark 捕捉访问网页的全过程，我不仅提升了自己的技术能力和知识水平，还增加了解决问题和优化网络性能的能力。这种实践经验对于个人的成长和职业发展都具有重要意义。