

计算机网络

实验三 数据包抓取与分析

谢瑞桃

xie@szu.edu.cn

[rtxie.github.io](https://github.com/rtxie)

计算机与软件学院

深圳大学





实验三 数据包抓取与分析

■ 实验目的

- 学习安装、使用协议分析软件，掌握基本的数据报抓取、过滤和分析方法，能分析HTTP、TCP、ICMP等协议。

■ 实验环境

- 使用具有Internet连接的Windows操作系统；
- 抓包软件Wireshark。



实验内容

1. 安装学习Wireshark软件
2. 抓包与分析HTTP协议
3. 分析TCP协议
4. 分析TCP三次握手
5. 分析ICMP协议



实验任务要求

- 请参考本讲义学习Wireshark软件的使用方法
- 安装Wireshark软件
- 理解TCP/IP协议分层模型
- 了解TCP报文格式
- 理解TCP三次握手
- 依照步骤完成实验内容1—5
- 对实验结果截图
- 撰写实验报告



实验报告撰写要求

- 使用教务处制作的实验报告模板
- 注意按进度填写实验时间和实验报告提交时间
- 填写模板中的每一部分
- 填写实验步骤时，做到条理清晰
- 注意截图清晰、美观
- 要求在演示操作步骤的截图上加标注，指出操作步骤和操作结果，没有会被扣分
- 实验报告只有截图，没有文字说明讲解会扣分
- 实验结果要有原理分析，否则会被扣分
- 出现一模一样的实验报告，均得零分



实验内容

1. 安装学习Wireshark软件
2. 抓包与分析HTTP协议
3. 分析TCP协议
4. 分析TCP三次握手
5. 分析ICMP协议



1. 安装学习Wireshark软件

- Wireshark是世界上最广泛使用的网络协议分析器。

- 官网地址：

<https://www.wireshark.org/>

- 软件使用手册：

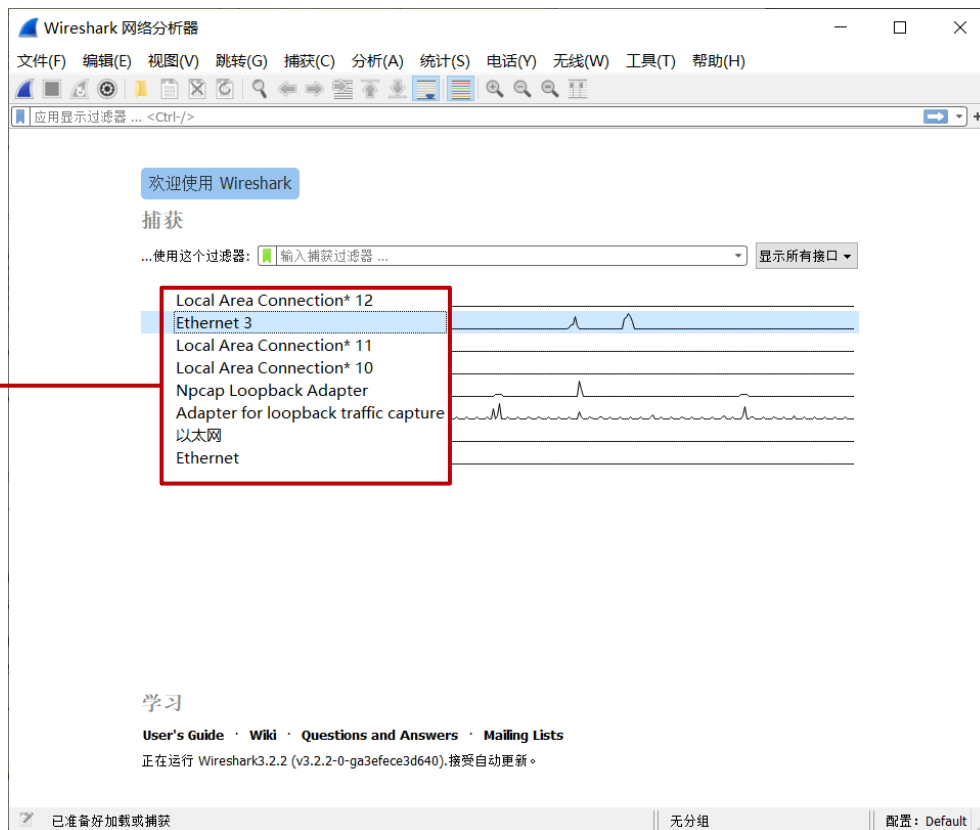
https://www.wireshark.org/docs/wsug_html_chunked

- 1) 请下载并安装该软件。(本讲义Wireshark版本号为3.2.2)

1. 安装学习Wireshark软件

- 2) 运行Wireshark，初始界面如下图。
- 3) 从接口列表中选择要捕获的接口，双击即可开始捕获。

接口列表



1. 安装学习Wireshark软件

4) Wireshark进入主界面，并开始捕获分组。

开始/停止捕获分组

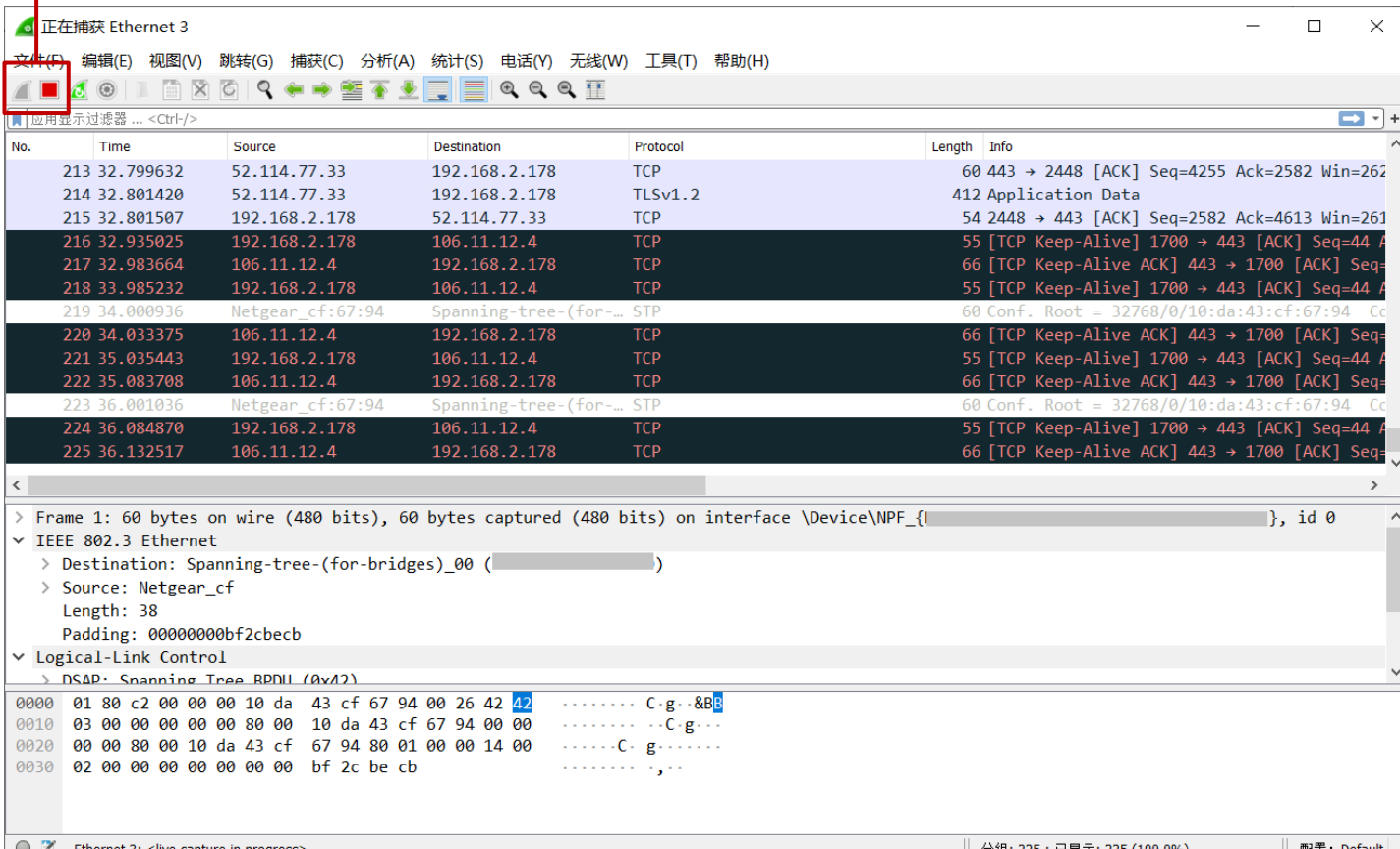
过滤器

分组列表栏

分组详情栏

分组字节栏

状态栏



No.	Time	Source	Destination	Protocol	Length	Info
213	32.799632	52.114.77.33	192.168.2.178	TCP	60	443 → 2448 [ACK] Seq=4255 Ack=2582 Win=262
214	32.801420	52.114.77.33	192.168.2.178	TLSv1.2	412	Application Data
215	32.801507	192.168.2.178	52.114.77.33	TCP	54	2448 → 443 [ACK] Seq=2582 Ack=4613 Win=261
216	32.935025	192.168.2.178	106.11.12.4	TCP	55	[TCP Keep-Alive] 1700 → 443 [ACK] Seq=44 A
217	32.983664	106.11.12.4	192.168.2.178	TCP	66	[TCP Keep-Alive ACK] 443 → 1700 [ACK] Seq=
218	33.985232	192.168.2.178	106.11.12.4	TCP	55	[TCP Keep-Alive] 1700 → 443 [ACK] Seq=44 A
219	34.000936	Netgear_cf:67:94	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/0/10:da:43:cf:67:94 Cc
220	34.033375	106.11.12.4	192.168.2.178	TCP	66	[TCP Keep-Alive ACK] 443 → 1700 [ACK] Seq=
221	35.035443	192.168.2.178	106.11.12.4	TCP	55	[TCP Keep-Alive] 1700 → 443 [ACK] Seq=44 A
222	35.083708	106.11.12.4	192.168.2.178	TCP	66	[TCP Keep-Alive ACK] 443 → 1700 [ACK] Seq=
223	36.001036	Netgear_cf:67:94	Spanning-tree-(for-...	STP	60	Conf. Root = 32768/0/10:da:43:cf:67:94 Cc
224	36.084870	192.168.2.178	106.11.12.4	TCP	55	[TCP Keep-Alive] 1700 → 443 [ACK] Seq=44 A
225	36.132517	106.11.12.4	192.168.2.178	TCP	66	[TCP Keep-Alive ACK] 443 → 1700 [ACK] Seq=

> Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{...}, id 0

IEEE 802.3 Ethernet

- > Destination: Spanning-tree-(for-bridges)_00 ()
- > Source: Netgear_cf
- Length: 38
- Padding: 00000000bf2cbeeb

Logical-Link Control

- > DSAP: Spanning Tree BDDU (0x42)

0000 01 80 c2 00 00 00 10 da 43 cf 67 94 00 26 42 42 C-g-&B

0010 03 00 00 00 00 00 80 00 10 da 43 cf 67 94 00 00 C-g...

0020 00 00 80 00 10 da 43 cf 67 94 80 01 00 00 14 00 C-g:.....

0030 02 00 00 00 00 00 00 00 bf 2c be eb ,..

Ethernet 3: <live capture in progress> | 分组: 225 · 已显示: 225 (100.0%) | 配置: Default



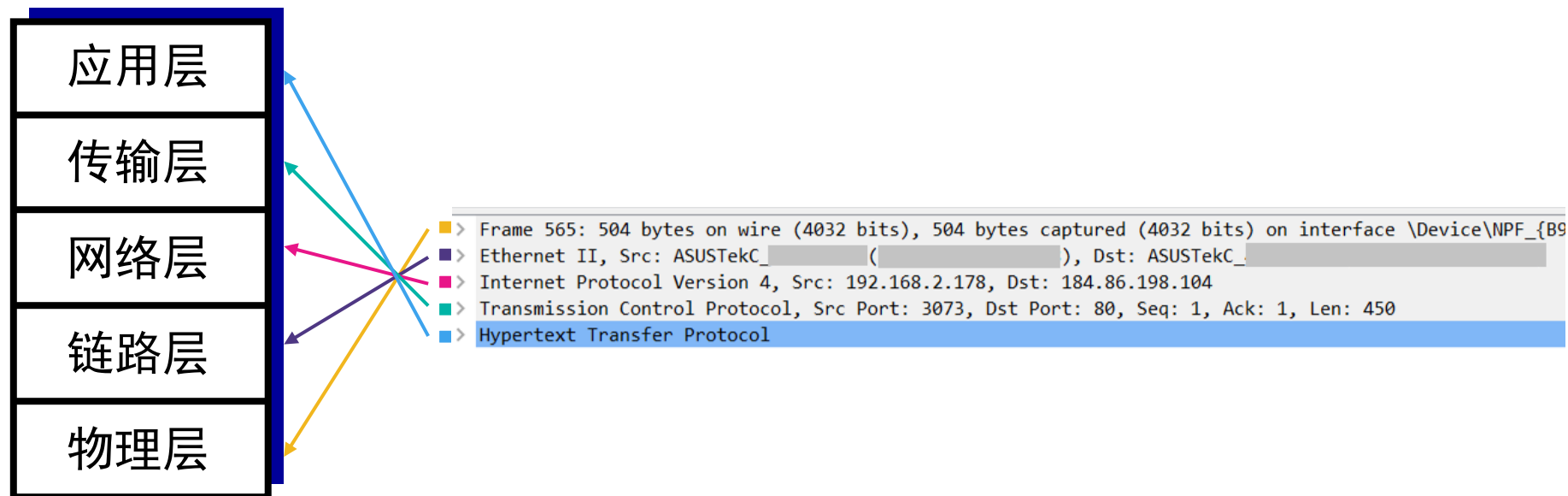
1. 安装学习Wireshark软件

5) 学习使用过滤器？

- 协议过滤
 - 举例：http
- IP地址过滤
 - 举例：ip.src == 192.168.2.178 and ip.dst == 184.86.198.104
 - 含义：过滤源地址是192.168.2.178并且目的地址是184.86.198.104的分组
- 模式过滤
 - 举例：http.request.method=="GET"
 - 含义：过滤http请求方法是GET的分组
- 端口过滤
 - 举例：tcp.port == 80
 - 含义：过滤tcp端口号是80的分组

1. 安装学习Wireshark软件

6) 分组详情栏





实验内容

1. 安装学习Wireshark软件
2. 抓包与分析HTTP协议
3. 分析TCP协议
4. 分析TCP三次握手
5. 分析ICMP协议

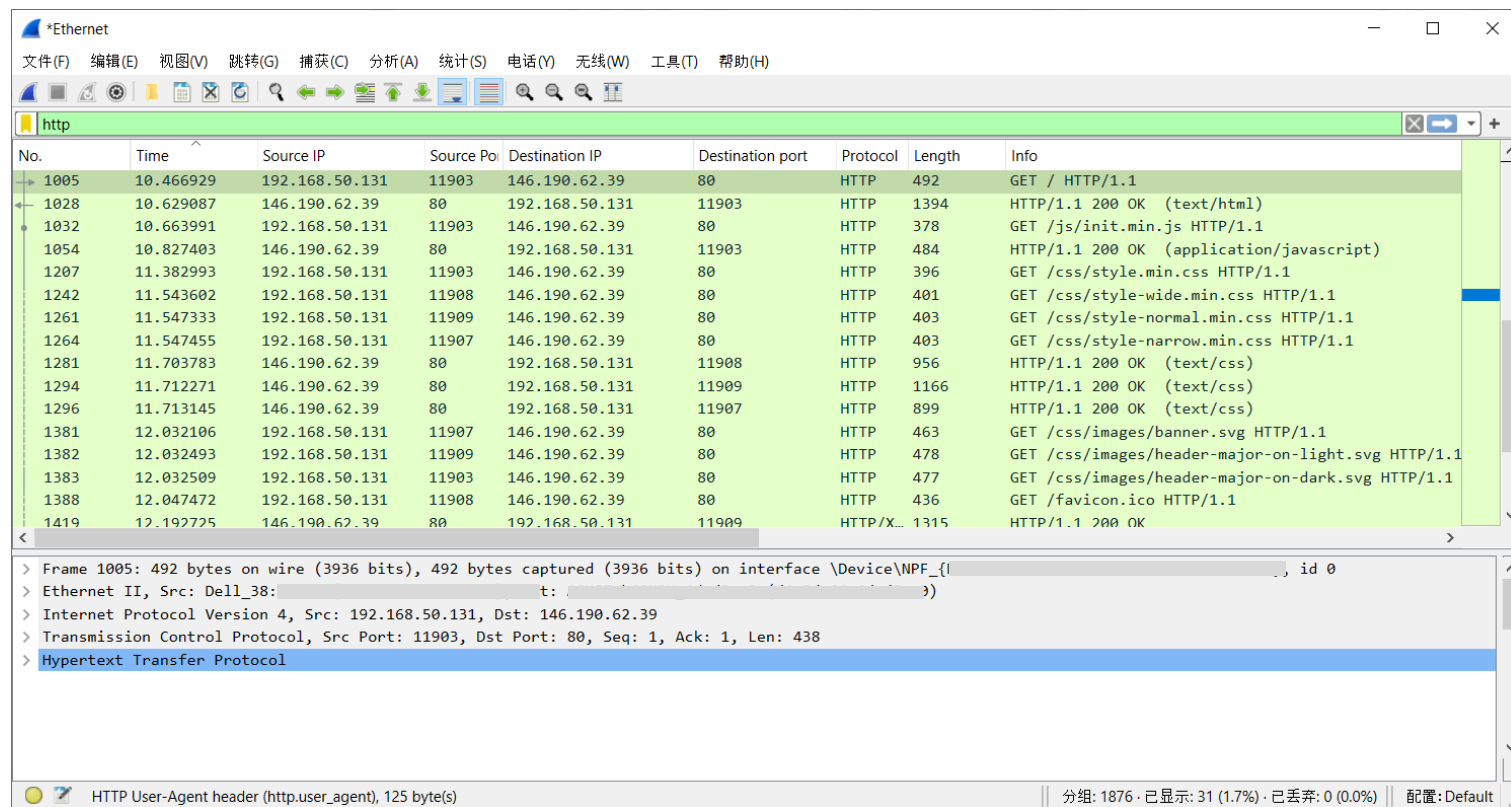


2. 抓包与分析HTTP协议

- 1) 开启Wireshark抓包，在过滤器中输入http，即过滤http协议的分组。
- 2) 打开浏览器，输入一个URL（例如http://httpforever.com/）。注意：为了避免浏览器缓存起作用，最好使用chrome浏览器的incognito隐身模式。

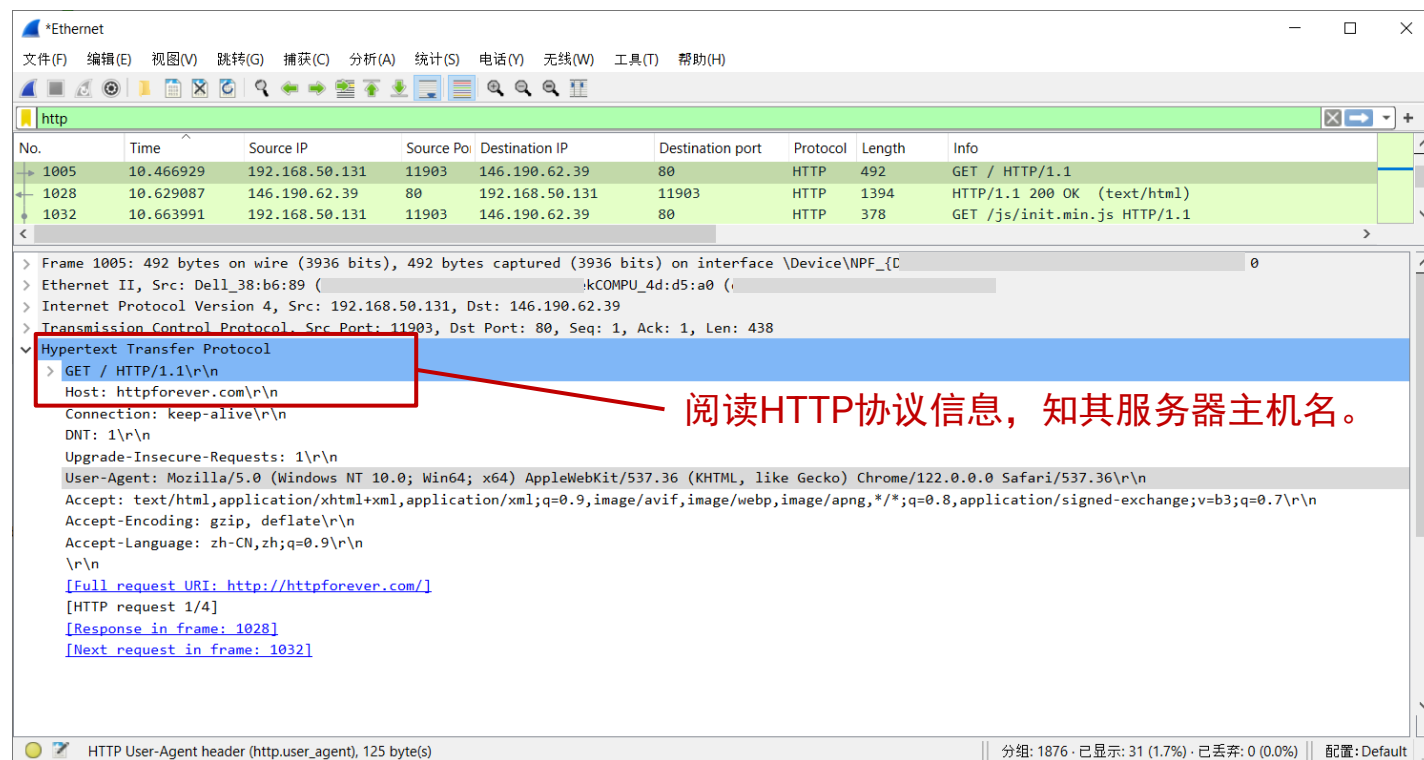
2. 抓包与分析HTTP协议

3) 观察到Wireshark分组列表栏中出现了HTTP协议分组。



2. 抓包与分析HTTP协议

- 4) 分析哪些分组是前一步浏览网页发生的。单击任意分组，分组详情栏显示其具体信息。



The screenshot shows the Wireshark interface with a packet capture on the 'http' filter. The packet list shows three packets: a GET request (No. 1005), a 200 OK response (No. 1028), and a GET for a JavaScript file (No. 1032). The packet details pane for packet 1005 is expanded, showing the Hypertext Transfer Protocol section. A red box highlights the 'GET / HTTP/1.1' line, and a red arrow points to it with the text '阅读HTTP协议信息，知其服务器主机名。' (Read HTTP protocol information, know its server host name).

No.	Time	Source IP	Source Port	Destination IP	Destination port	Protocol	Length	Info
1005	10.466929	192.168.50.131	11903	146.190.62.39	80	HTTP	492	GET / HTTP/1.1
1028	10.629087	146.190.62.39	80	192.168.50.131	11903	HTTP	1394	HTTP/1.1 200 OK (text/html)
1032	10.663991	192.168.50.131	11903	146.190.62.39	80	HTTP	378	GET /js/init.min.js HTTP/1.1

Frame 1005: 492 bytes on wire (3936 bits), 492 bytes captured (3936 bits) on interface \Device\NPF_{...} 0

Ethernet II, Src: Dell_38:b6:89 (...), Dst: {...} kCOMPU_4d:d5:a0 (...)

Internet Protocol Version 4, Src: 192.168.50.131, Dst: 146.190.62.39

Transmission Control Protocol, Src Port: 11903, Dst Port: 80, Seq: 1, Ack: 1, Len: 438

Hypertext Transfer Protocol

GET / HTTP/1.1\r\n

Host: httpforever.com\r\n

Connection: keep-alive\r\n

DNT: 1\r\n

Upgrade-Insecure-Requests: 1\r\n

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36\r\n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n

Accept-Encoding: gzip, deflate\r\n

Accept-Language: zh-CN,zh;q=0.9\r\n

\r\n

[Full request URI: http://httpforever.com/]

[HTTP request 1/4]

[Response in frame: 1028]

[Next request in frame: 1032]

HTTP User-Agent header (http.user_agent), 125 byte(s)

分组: 1876 · 已显示: 31 (1.7%) · 已丢弃: 0 (0.0%) | 配置: Default

2. 抓包与分析HTTP协议

- 5) 从步骤四所得的分组，获知此次通信的源IP地址和目的IP地址。(这里，192.168.50.131是私有IP地址，所以是用户的主机。)

The image shows a Wireshark packet capture window titled '*Ethernet'. The packet list pane shows three packets. The first packet (No. 1005) is selected, and its details pane is expanded to show the Hypertext Transfer Protocol section. A red box highlights the 'GET / HTTP/1.1' line in the details pane. Another red box highlights the 'Source IP' and 'Destination IP' columns in the packet list pane for the first packet. A red arrow points from the text '阅读HTTP协议信息，知其主机名。' to the highlighted 'GET / HTTP/1.1' line.

No.	Time	Source IP	Source Port	Destination IP	Destination port	Protocol	Length	Info
1005	10.466929	192.168.50.131	11903	146.190.62.39	80	HTTP	492	GET / HTTP/1.1
1028	10.629087	146.190.62.39	80	192.168.50.131	11903	HTTP	1394	HTTP/1.1 200 OK (text/html)
1032	10.663991	192.168.50.131	11903	146.190.62.39	80	HTTP	378	GET /js/init.min.js HTTP/1.1

Frame 1005: 492 bytes on wire (3936 bits), 492 bytes captured (3936 bits) on interface \Device\NPF... id 0

Ethernet II, Src: Dell_38:b6:89, Dst: TekCOMPU_4d:d5:a0

Internet Protocol Version 4, Src: 192.168.50.131, Dst: 146.190.62.39

Transmission Control Protocol, Src Port: 11903, Dst Port: 80, Seq: 1, Ack: 1, Len: 438

Hypertext Transfer Protocol

GET / HTTP/1.1

Host: httpforever.com

Connection: keep-alive

DNT: 1

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7

Accept-Encoding: gzip, deflate

Accept-Language: zh-CN,zh;q=0.9

[Full request URI: http://httpforever.com/]

[HTTP request 1/4]

[Response in frame: 1028]

[Next request in frame: 1032]



实验内容

1. 安装学习Wireshark软件
2. 抓包与分析HTTP协议
3. 分析TCP协议
4. 分析TCP三次握手
5. 分析ICMP协议

3.分析 TCP协议

1) 对2-4的分组，分析TCP协议信息。

http.pcapng

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)

http

No.	Time	Source	Destination	Protocol	Length	Info
565	32.589977	192.168.2.178	184.86.198.104	HTTP	504	GET /Xplore/home.jsp HTTP/1.1

> Frame 565: 504 bytes on wire (4032 bits), 504 bytes captured (4032 bits) on interface \Device\NPF_{...}

> Ethernet II, Src: ASUSTekC_..., Dst: ASUSTekC_...

> Internet Protocol Version 4, Src: 192.168.2.178, Dst: 184.86.198.104

> Transmission Control Protocol, Src Port: 3073, Dst Port: 80, Seq: 1, Ack: 1, Len: 450

- Source Port: 3073
- Destination Port: 80 **源端口号和目的端口号**
- [Stream index: 17]
- [TCP Segment Len: 450]
- Sequence number: 1 (relative sequence number)
- Sequence number (raw): 817913973 **序列号**
- [Next sequence number: 451 (relative sequence number)]
- Acknowledgment number: 1 (relative ack number)
- Acknowledgment number (raw): 1952848251 **确认号**
- 0101 = Header Length: 20 bytes (5) **报头长度**
- > Flags: 0x018 (PSH, ACK) **标志位**
- Window size value: 1026 **窗口大小**
- [Calculated window size: 262656]
- [Window size scaling factor: 256]
- Checksum: 0x43f6 [unverified] **校验和**
- [Checksum Status: Unverified]
- Urgent pointer: 0
- > [SEQ/ACK analysis]
- > [Timestamps]
- TCP payload (450 bytes) **数据**
- > Hypertext Transfer Protocol

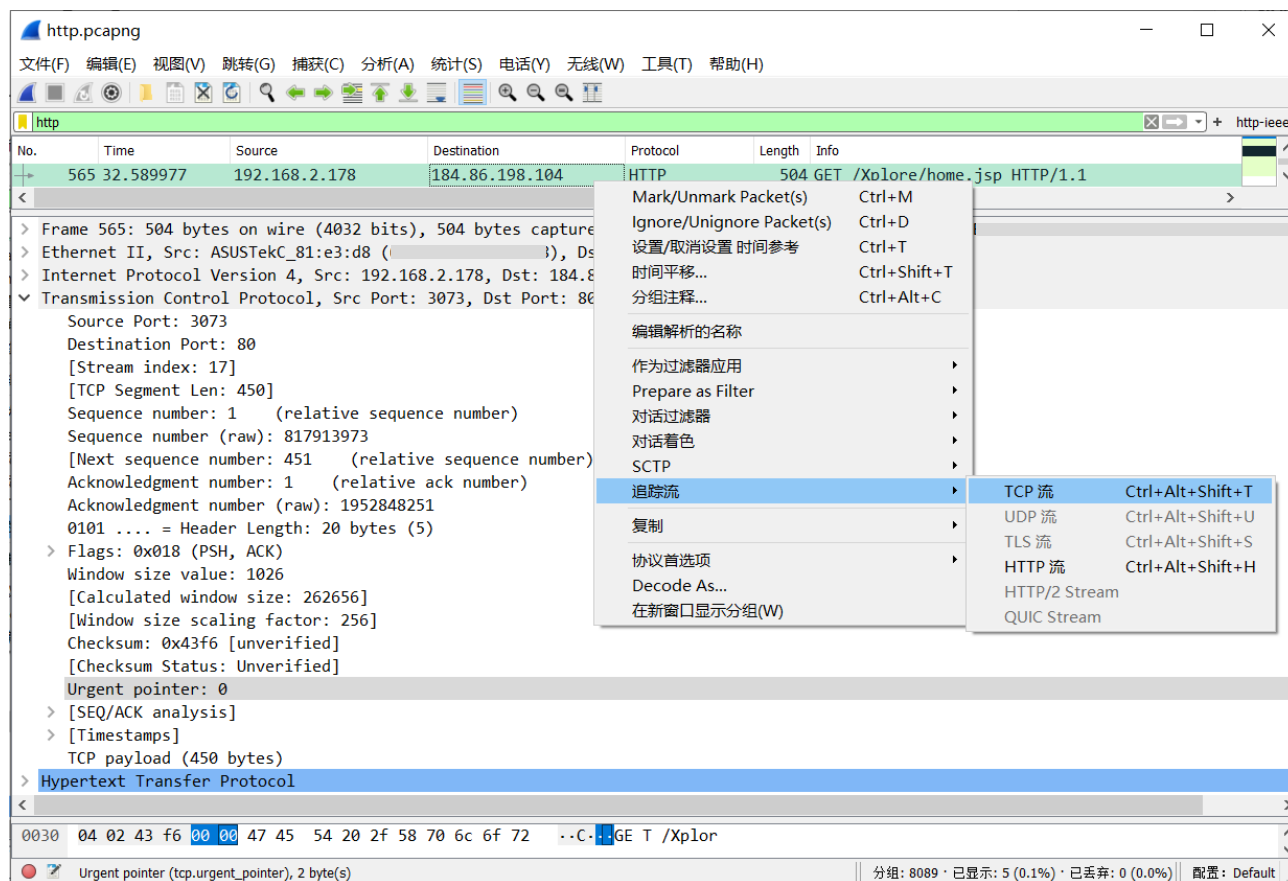
0030 04 02 43 f6 00 00 47 45 54 20 2f 58 70 6c 6f 72 ..C..GE T /Xplor

Urgent pointer (tcp.urgent_pointer), 2 byte(s)

分组: 8089 · 已显示: 5 (0.1%) · 已丢弃: 0 (0.0%) 配置: Default

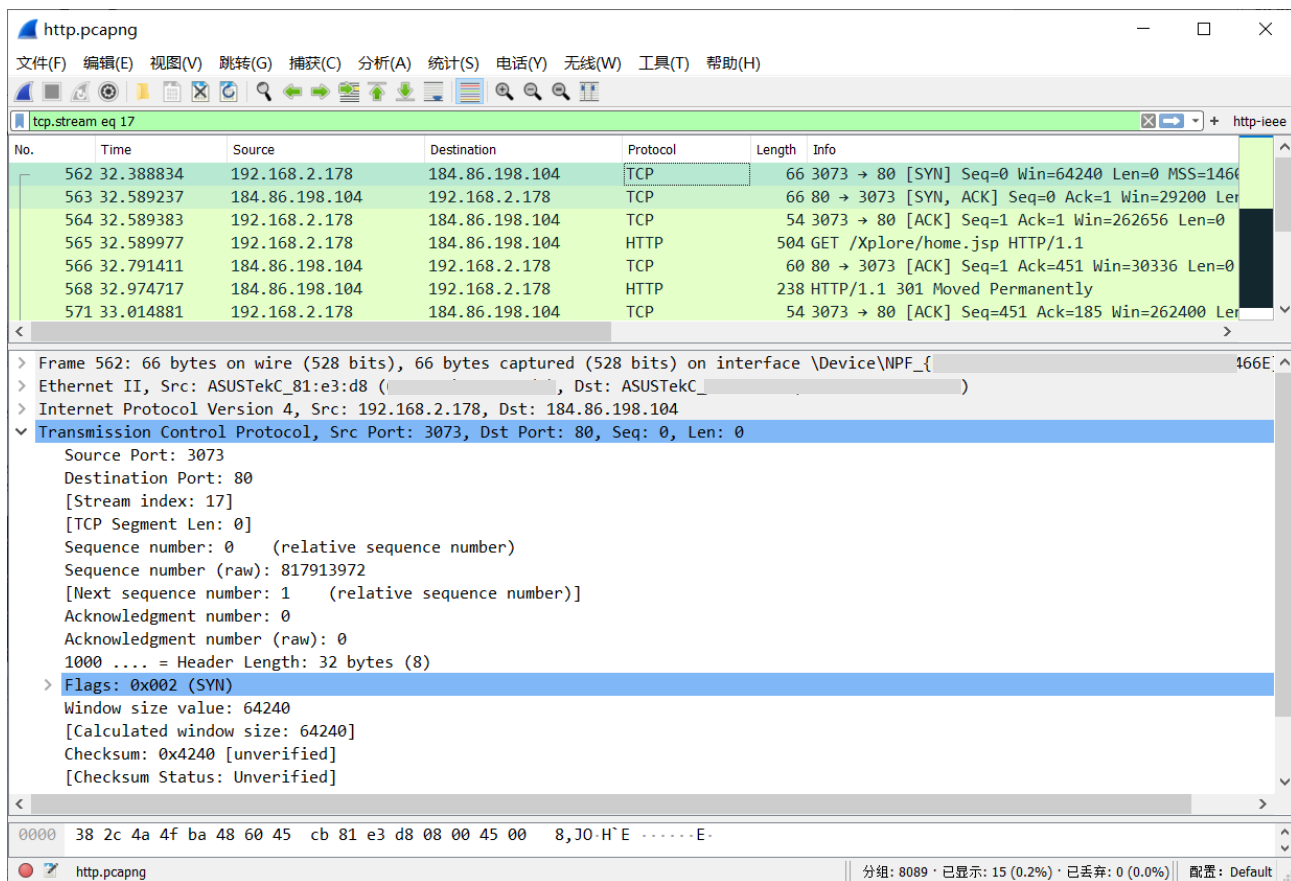
3.分析 TCP协议

2) 对2-4的分组，追踪其TCP流。点击右键，从下拉菜单中选择TCP流。



3.分析 TCP协议

3) 对2-4的分组，追踪其TCP流。点击右键，从下拉菜单中选择TCP流。



3.分析 TCP协议

- 4) 找到TCP建立连接的分组。原理：1) TCP连接建立应该在HTTP GET请求之前完成; 2) TCP 建立连接时会设置标志位SYN。

TCP 连接建立的分组

The image shows a Wireshark packet capture of a TCP connection establishment. The packet list shows several packets, with packet 562 highlighted. The details pane for packet 562 shows the TCP header information, including the SYN flag being set.

No.	Time	Source	Destination	Protocol	Length	Info
562	32.388834	192.168.2.178	184.86.198.104	TCP	66	3073 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=146
563	32.589237	184.86.198.104	192.168.2.178	TCP	66	80 → 3073 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0
564	32.589383	192.168.2.178	184.86.198.104	TCP	54	3073 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
565	32.589977	192.168.2.178	184.86.198.104	HTTP	504	GET /Xplore/home.jsp HTTP/1.1
566	32.791411	184.86.198.104	192.168.2.178	TCP	60	80 → 3073 [ACK] Seq=1 Ack=451 Win=30336 Len=0
568	32.974717	184.86.198.104	192.168.2.178	HTTP	238	HTTP/1.1 301 Moved Permanently
571	33.014881	192.168.2.178	184.86.198.104	TCP	54	3073 → 80 [ACK] Seq=451 Ack=185 Win=262400 Len=0

Frame 562: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{B9E6...}

Ethernet II, Src: ASUSTekC_81:e3:d8 (ASUSTekC_4f...), Dst: ASUSTekC_4f...

Internet Protocol Version 4, Src: 192.168.2.178, Dst: 184.86.198.104

Transmission Control Protocol, Src Port: 3073, Dst Port: 80, Seq: 0, Len: 0

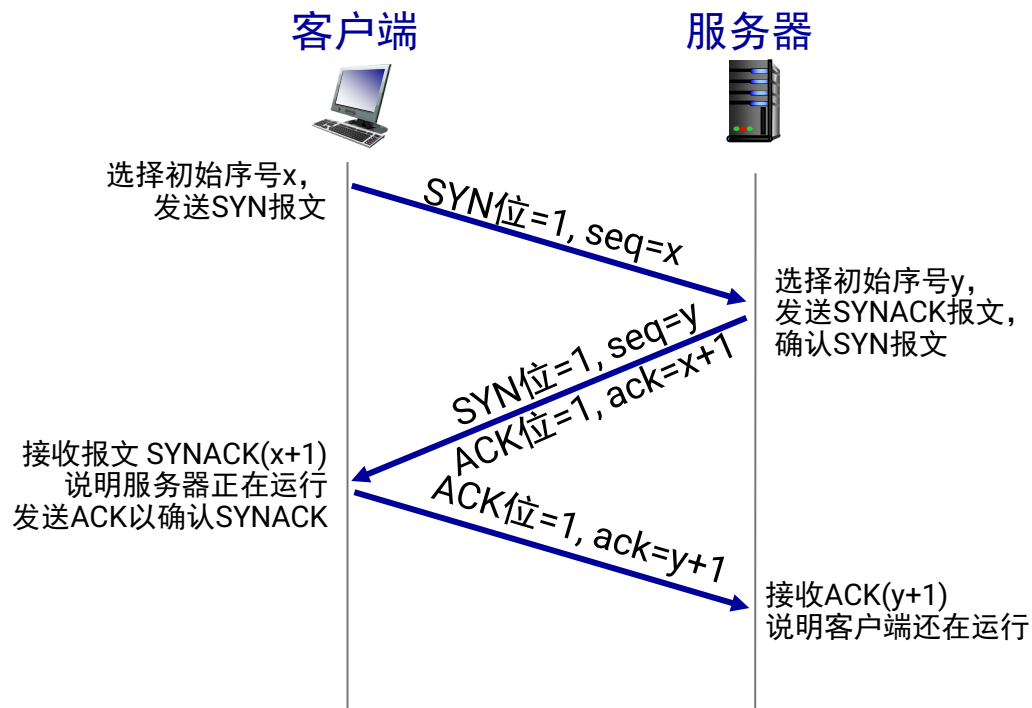
Source Port: 3073
Destination Port: 80
[Stream index: 17]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
Sequence number (raw): 817913972
[Next sequence number: 1 (relative sequence number)]
Acknowledgment number: 0
Acknowledgment number (raw): 0
1000 = Header Length: 32 bytes (8)
Flags: 0x002 (SYN)
Window size value: 64240
[Calculated window size: 64240]
Checksum: 0x4240 [unverified]
[Checksum Status: Unverified]



实验内容

1. 安装学习Wireshark软件
2. 抓包与分析HTTP协议
3. 分析TCP协议
4. 分析**TCP三次握手**
5. 分析ICMP协议

4. TCP三次握手



4. 分析TCP三次握手

1) 分析TCP三次握手，第一次握手(SYN)。

The image shows a Wireshark packet capture window titled 'http.pcapng'. The packet list on the left shows three packets: a SYN packet (No. 562), a SYN-ACK packet (No. 563), and an ACK packet (No. 564). The packet details pane on the right shows the selected packet (No. 562) expanded, displaying the following information:

- Transmission Control Protocol, Src Port: 3073, Dst Port: 80, Seq: 0, Len: 0
- Source Port: 3073
- Destination Port: 80
- [Stream index: 17]
- [TCP Segment Len: 0]
- Sequence number: 0 (relative sequence number)
- Sequence number (raw): 817913972
- [Next sequence number: 1 (relative sequence number)]
- Acknowledgment number: 0
- Acknowledgment number (raw): 0
- 1000 = Header Length: 32 bytes (8)
- Flags: 0x002 (SYN)
- 000. = Reserved: Not set
- ...0 = Nonce: Not set
- 0... = Congestion Window Reduced (CWR): Not set
-0.. = ECN-Echo: Not set
-0. = Urgent: Not set
-0 = Acknowledgment: Not set
-0... = Push: Not set
- 0 = Reset: Not set
- >1. = Syn: Set
-0 = Fin: Not set
- [TCP Flags:S.]

Red annotations highlight the 'Sequence number (raw): 817913972' and the 'Syn: Set' flag, with the text '序号是817913972' and 'SYN位置' respectively. A red '1' is placed below the 'SYN位置' text.

The packet bytes pane at the bottom shows the raw data of the packet, with the first 32 bytes (0000 38 2c 4a 4f ba 48 60 45 cb 81 e3 d8 08 00 45 00) highlighted in blue, corresponding to the TCP header.

4. 分析TCP三次握手

2) 分析TCP三次握手，第二次握手(SYNACK)。

The image shows a Wireshark packet capture window titled 'http.pcapng'. The packet list on the left shows three packets. The selected packet is packet 564, a TCP segment from 192.168.2.178 to 184.86.198.104, port 80 to 3073, with sequence number 3073 and acknowledgment number 80. The packet details pane on the right shows the following information:

- Transmission Control Protocol, Src Port: 80, Dst Port: 3073, Seq: 0, Ack: 1, Len: 0
- Source Port: 80
- Destination Port: 3073
- [Stream index: 17]
- [TCP Segment Len: 0]
- Sequence number: 0 (relative sequence number)
- Sequence number (raw): 1952848250
- [Next sequence number: 1 (relative sequence number)]
- Acknowledgment number: 1 (relative ack number)
- Acknowledgment number (raw): 817913973
- 1000 = Header Length: 32 bytes (8)
- Flags: 0x012 (SYN, ACK)
- 000. = Reserved: Not set
- ...0 = Nonce: Not set
- 0... = Congestion Window Reduced (CWR): Not set
-0.. = ECN-Echo: Not set
-0. = Urgent: Not set
-1 = Acknowledgment: Set
- 0... = Push: Not set
- 0 = Reset: Not set
- >1. = Syn: Set
-0 = Fin: Not set
- [TCP Flags:A..S.]

Red boxes highlight the following fields in the packet details pane:

- Sequence number (raw): 1952848250
- Acknowledgment number (raw): 817913973
- Acknowledgment: Set
- Syn: Set

Red text annotations on the right side of the image provide the following information:

- 序号是1952848250
- 确认号是817913973 = SYN序号加1
- ACK位置1
- SYN位置1

4. 分析TCP三次握手

3) 分析TCP三次握手，第三次握手(ACK)。

http.pcapng

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.stream eq 17

No.	Time	Source	Destination	Protocol	Length	Info
563	32.589237	184.86.198.104	192.168.2.178	TCP	66	80 → 3073 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0
564	32.589383	192.168.2.178	184.86.198.104	TCP	54	3073 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
565	32.589977	192.168.2.178	184.86.198.104	HTTP	504	GET /Xplore/home.jsp HTTP/1.1

Source Port: 3073
Destination Port: 80
[Stream index: 17]
[TCP Segment Len: 0]
Sequence number: 1 (relative sequence number)
Sequence number (raw): 817913973
[Next sequence number: 1 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
Acknowledgment number (raw): 1952848251
0101 = Header Length: 20 bytes (5)
✓ Flags: 0x010 (ACK)
000. = Reserved: Not set
...0 = Nonce: Not set
.... 0... = Congestion Window Reduced (CWR): Not set
.... .0.. = ECN-Echo: Not set
0 = Urgent: Not set
.... ..1 = Acknowledgment: Set
.... 0... = Push: Not set
....0.. = Reset: Not set
....0. = Syn: Not set
....0 = Fin: Not set
[TCP Flags:A....]
Window size value: 1026

0030 04 02 42 34 00 00 ..B4..

Urgent pointer (tcp.urgent_pointer), 2 byte(s)

分组: 8089 · 已显示: 15 (0.2%) · 已丢弃: 0 (0.0%) 配置: Default

确认号是1952848251 = SYNACK序号加1

ACK位置1



实验内容

1. 安装学习Wireshark软件
2. 抓包与分析HTTP协议
3. 分析TCP协议
4. 分析TCP三次握手
5. 分析ICMP协议

5. 分析ICMP协议

- 在Wireshark过滤栏里输入icmp（这是ping指令使用的协议）；
- 在PowerShell里使用ping指令；
- 请对比Wireshark分组列表栏与ping指令显示结果，并展开分析。

```
Windows PowerShell
PS C:\Users\rutiao> ping www.bing.com

正在 Ping cn-0001.cn-msedge.net [202.89.233.101] 具有 32 字节的数据:
来自 202.89.233.101 的回复: 字节=32 时间=40ms TTL=117
来自 202.89.233.101 的回复: 字节=32 时间=42ms TTL=117
来自 202.89.233.101 的回复: 字节=32 时间=40ms TTL=117
来自 202.89.233.101 的回复: 字节=32 时间=40ms TTL=117

202.89.233.101 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 40ms, 最长 = 42ms, 平均 = 40ms
PS C:\Users\rutiao>
```

正在捕获 Ethernet 3

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(V) 无线(W) 工具(T) 帮助(H)

icmp

No.	Time	Source	Destination	Protocol	Length	Info
68	15.046948	192.168.2.178	202.89.233.101	ICMP	74	Echo (ping) request id=0x0001, seq=7/1792, ttl=128 (reply in 70)
70	15.087288	202.89.233.101	192.168.2.178	ICMP	74	Echo (ping) reply id=0x0001, seq=7/1792, ttl=117 (request in 68)
87	16.050063	192.168.2.178	202.89.233.101	ICMP	74	Echo (ping) request id=0x0001, seq=8/2048, ttl=128 (reply in 88)
88	16.092901	202.89.233.101	192.168.2.178	ICMP	74	Echo (ping) reply id=0x0001, seq=8/2048, ttl=117 (request in 87)
92	17.053407	192.168.2.178	202.89.233.101	ICMP	74	Echo (ping) request id=0x0001, seq=9/2304, ttl=128 (reply in 94)
94	17.093807	202.89.233.101	192.168.2.178	ICMP	74	Echo (ping) reply id=0x0001, seq=9/2304, ttl=117 (request in 92)
97	18.060726	192.168.2.178	202.89.233.101	ICMP	74	Echo (ping) request id=0x0001, seq=10/2560, ttl=128 (reply in 98)
98	18.101549	202.89.233.101	192.168.2.178	ICMP	74	Echo (ping) reply id=0x0001, seq=10/2560, ttl=117 (request in 97)

< >

> Frame 68: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{...}

> Ethernet II, Src: ASUSTekC_81:e3:d8 (ASUSTekC_4f:ba:48), Dst: 08:00:00:00:00:00

> Internet Protocol Version 4, Src: 192.168.2.178, Dst: 202.89.233.101

> Internet Control Message Protocol

0000	38 2c 4a 4f ba 48 60 45	cb 81 e3 d8 08 00 45 00	8,30 H'EE-
0010	00 3c 7e 8c 00 00 80 01	00 00 c0 a8 02 b2 ca 59	-<.....Y
0020	e9 65 08 00 4d 54 00 01	00 07 61 62 63 64 65 66	-e-MT.....abcde
0030	67 68 69 6a 6b 6c 6d 6e	6f 70 71 72 73 74 75 76	ghijklmn opqrstuv
0040	77 61 62 63 64 65 66 67	68 69	wabcde fghij

Ethernet 3: <live capture in progress> 分组: 2094 · 已显示: 8 (0.4%) 配置: Default



恭喜你已完成实验