

1. What is the IP address of <http://www.faqs.org> website? KORHAN ERDOĞDU 30838

- We should open the pcap file in Wireshark.
- We apply the filter http to display only HTTP traffic.
- We then look for an HTTP GET request to <http://www.faqs.org>.
- In the packet details, we locate the "Destination" field to find the IP address of [www.faqs.org](http://www.faqs.org).

**Answer:** The IP address of <http://www.faqs.org> is 199.231.164.68

The screenshot shows the Wireshark interface with the following details:

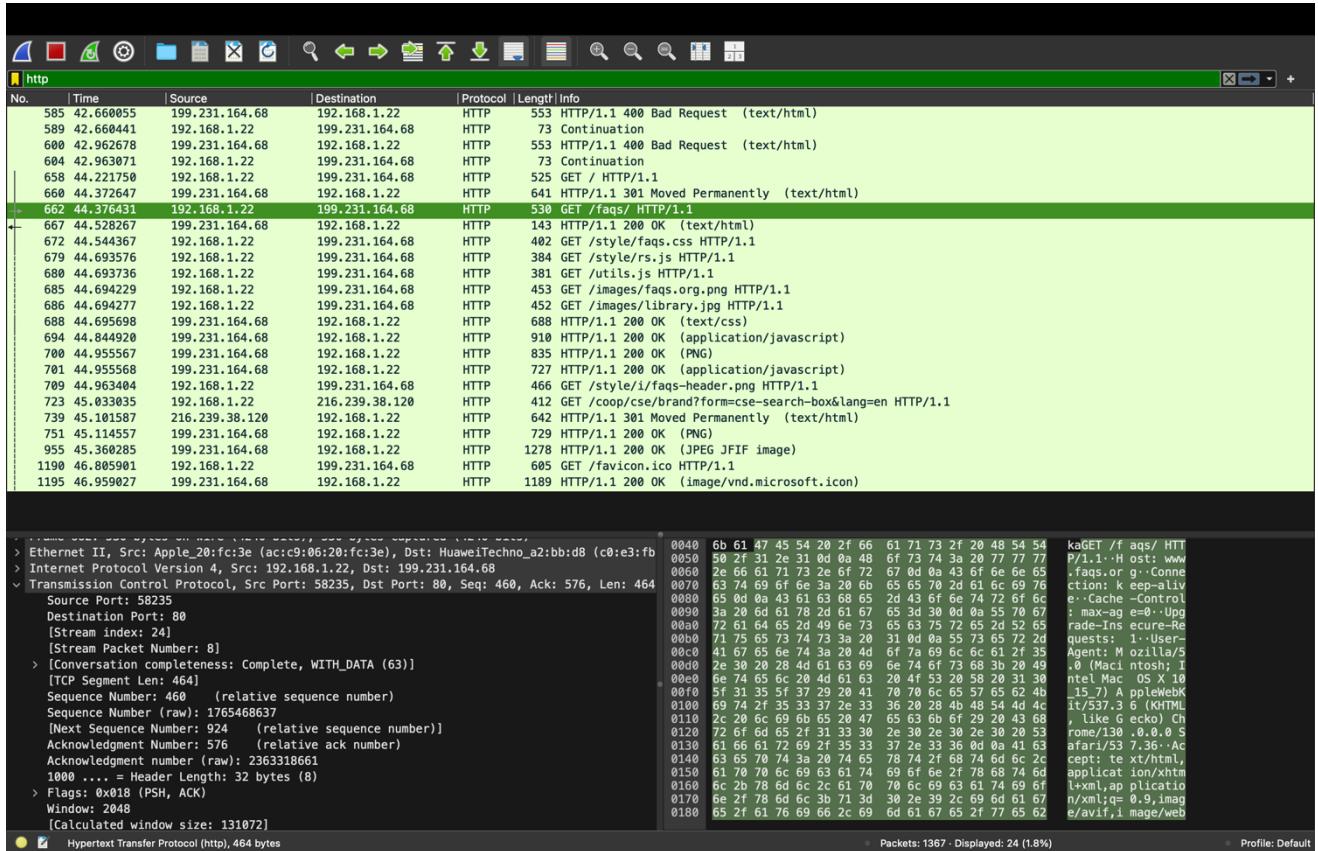
- Panels:** Top: Network, Time, Source, Destination, Protocol, Length, Info; Bottom: Bytes, Hex, ASCII.
- Packet List:** Shows 195 total packets. The 662nd packet is highlighted, which is a GET request to <http://www.faqs.org/faqs/>.
- Details Panel:** Shows the HTTP headers for the selected packet:
 

```
> GET /faqs/ HTTP/1.1\r\nHost: www.faqs.org\r\nConnection: keep-alive\r\nCache-Control: max-age=0\r\nUpgrade-Insecure-Requests: 1\r\nUser-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36
```
- Hex Panel:** Shows the raw hex and ASCII representation of the selected packet.
- Statistics Panel:** Shows 1367 total packets, 24 displayed (1.8%).
- Profiles Panel:** Shows the current profile is Default.

2. What are the source port and destination port numbers of the HTTP request used to get <http://www.faqs.org>?

- We apply the http filter to find the HTTP GET request to <http://www.faqs.org>.
- In the packet details, we look for the "Source Port" and "Destination Port" fields under the TCP header.
- The destination port for HTTP traffic should be 80, and the source port will be a temporary port assigned by our browser.

**Answer:** The source port is 58235 and the destination port is 80.



The screenshot shows a Wireshark capture window with the following details:

- Protocol Filter:** http
- Selected Row:** The 662nd packet, which is a GET request to <http://faqs/> (HTTP/1.1).
- Table Headers:** No., Time, Source, Destination, Protocol, Length, Info
- Table Data:** A list of 1195 packets. The selected row (662) shows the following details:
  - Time: 44.376431
  - Source: 192.168.1.22
  - Destination: 199.231.164.68
  - Protocol: HTTP
  - Length: 538
  - Info: GET /faqs/ HTTP/1.1
- Packet Details:** Shows the raw hex and ASCII data for the selected packet. The ASCII dump includes the URL and HTTP headers.
- Statistics:** Packets: 1367 - Displayed: 24 (1.8%)
- Profile:** Default

### 3. What is the IP address of github.com domain?

- We apply the icmp filter to locate the ICMP Echo Request sent to github.com when we pinged it.
- In the packet details, we find the "Destination" field in the IP header, which contains the IP address of github.com.

**Answer:** The IP address of github.com is: 140.82.121.3

No.	Time	Source	Destination	Protocol	Length	Info
143	44.911001	192.168.1.22	192.168.1.1	ICMP	70	Destination unreachable (Port unreachable)
755	45.117081	192.168.1.22	192.168.1.1	ICMP	70	Destination unreachable (Port unreachable)
852	45.250733	192.168.1.22	192.168.1.1	ICMP	70	Destination unreachable (Port unreachable)
881	45.284989	192.168.1.22	192.168.1.1	ICMP	70	Destination unreachable (Port unreachable)
1000	45.410806	192.168.1.22	192.168.1.1	ICMP	70	Destination unreachable (Port unreachable)
1284	48.737674	192.168.1.22	192.168.1.1	ICMP	70	Destination unreachable (Port unreachable)
1298	61.286606	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=0/0, ttl=64 (reply in 1299)
1299	61.327728	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=0/0, ttl=51 (request in 1298)
1300	62.287265	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=1/256, ttl=64 (reply in 1301)
1301	62.330994	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=1/256, ttl=51 (request in 1300)
1302	63.289116	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=2/512, ttl=64 (reply in 1303)
1303	63.333109	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=2/512, ttl=51 (request in 1302)
1304	64.294447	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=3/768, ttl=64 (reply in 1305)
1305	64.337072	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=3/768, ttl=51 (request in 1304)
1306	65.299799	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=4/1024, ttl=64 (reply in 1307)
1307	65.343700	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=4/1024, ttl=51 (request in 1306)
1310	66.303536	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=5/1280, ttl=64 (reply in 1311)
1311	66.347238	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=5/1280, ttl=51 (request in 1310)
1312	67.308833	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=6/1536, ttl=64 (reply in 1313)
1313	67.352696	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=6/1536, ttl=51 (request in 1312)
1314	68.314138	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=7/1792, ttl=64 (reply in 1315)
1315	68.358024	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=7/1792, ttl=51 (request in 1314)
1319	69.319368	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=8/2048, ttl=64 (reply in 1321)
1321	69.359749	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=8/2048, ttl=51 (request in 1319)
1322	70.321928	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=9/2304, ttl=64 (reply in 1323)
1323	70.366030	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=9/2304, ttl=51 (request in 1322)
1324	71.327264	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=10/2560, ttl=64 (reply in 1325)

```

> Frame 1300: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
> Ethernet II, Src: Apple_28:fc:3e (ac:9:06:20:fc:3e), Dst: HuaweiTechno_a2:bb:d8 (c:0:e3:fb)
> Internet Protocol Version 4, Src: 192.168.1.22, Dst: 140.82.121.3
    0100 .... = Version: 4
    ... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSSCP: CS0, ECN: Not-ECT)
        Total Length: 84
        Identification: 0x2331 (9009)
        ... 0000 .... = Flags: 0x0
        ... 0 0000 0000 0000 = Fragment Offset: 0
        Time to Live: 64
        Protocol: ICMP (1)
        Header Checksum: 0x9064 [validation disabled]
        [Header checksum status: Unverified]
        Source Address: 192.168.1.22
        Destination Address: 140.82.121.3
        [Stream index: 26]
    > Internet Control Message Protocol
    > Internet Control Message Protocol: Protocol

```

Packets: 1367 - Displayed: 38 (2.8%)      Profile: Default

4. What are the type numbers of the ICMP Echo request and ICMP Echo reply (used for ping)?

- We use the icmp filter to display all ICMP packets.
- We select an ICMP Echo Request packet and an ICMP Echo Reply packet.
- In each packet's details, we look for the "Type" field under the ICMP header. Typically, Type 8 represents Echo Request, and Type 0 represents Echo Reply.

Echo Request:

```
> Frame 1300: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
> Ethernet II, Src: Apple_20:fc:3e (ac:c9:06:20:fc:3e), Dst: HuaweiTechno_a2:bb:d8 (c0:e3:0
> Internet Protocol Version 4, Src: 192.168.1.22, Dst: 140.82.121.3
< Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x48f4 [correct]
        [Checksum Status: Good]
    Identifier (BE): 12366 (0x304e)
    Identifier (LE): 20016 (0x4e30)
    Sequence Number (BE): 1 (0x0001)
    Sequence Number (LE): 256 (0x0100)
        [Response frame: 1301]
    Timestamp from icmp data: Nov 11, 2024 20:50:08.649389000 +03
        [Timestamp from icmp data (relative): 0.000191000 seconds]
    > Data (48 bytes)
```

Echo Reply:

```
> Frame 1307: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
> Ethernet II, Src: HuaweiTechno_a2:bb:d8 (c0:e3:fb:a2:bb:d8), Dst: Apple_20:fc:3e (ac:c9:06
> Internet Protocol Version 4, Src: 140.82.121.3, Dst: 192.168.1.22
< Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x2014 [correct]
        [Checksum Status: Good]
    Identifier (BE): 12366 (0x304e)
    Identifier (LE): 20016 (0x4e30)
    Sequence Number (BE): 4 (0x0004)
    Sequence Number (LE): 1024 (0x0400)
        [Request frame: 1306]
    [Response time: 43.901 ms]
    Timestamp from icmp data: Nov 11, 2024 20:50:11.661895000 +03
        [Timestamp from icmp data (relative): 0.044120000 seconds]
    > Data (48 bytes)
```

5. What is the range of sequence numbers in the ICMP Echo request and ICMP Echo reply packets captured when github.com was pinged? The range should start with the first request sent and end with the last reply received.

- We apply the icmp filter.
- We locate the first ICMP Echo Request packet sent to github.com and note its sequence number.
- Then we find the last ICMP Echo Reply received from github.com and note its sequence number.
- This gives us the range, starting with the first request's sequence number and ending with the last reply's sequence number.

We observe that the first ICMP Echo request has a sequence number of 0 (in frame 1298), and the final ICMP Echo request shown has a sequence number of 3584 (in frame 1335).

This range includes all sequence numbers from the first ICMP Echo Request to the last ICMP Echo Reply shown in the capture.

**Answer:** 0 to 3584.

No.	Time	Source	Destination	Protocol	Length	Info
579	42.387854	192.168.1.22	192.168.1.1	ICMP	78	Destination unreachable (Port unreachable)
655	44.874247	192.168.1.22	192.168.1.1	ICMP	78	Destination unreachable (Port unreachable)
713	44.972817	192.168.1.22	192.168.1.1	ICMP	78	Destination unreachable (Port unreachable)
755	45.117081	192.168.1.22	192.168.1.1	ICMP	78	Destination unreachable (Port unreachable)
852	45.250733	192.168.1.22	192.168.1.1	ICMP	78	Destination unreachable (Port unreachable)
881	45.284989	192.168.1.22	192.168.1.1	ICMP	78	Destination unreachable (Port unreachable)
1080	45.418806	192.168.1.22	192.168.1.1	ICMP	78	Destination unreachable (Port unreachable)
1284	48.737674	192.168.1.22	192.168.1.1	ICMP	78	Destination unreachable (Port unreachable)
1298	61.298606	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=0/0, ttl=64 (reply in 1299)
1299	61.327728	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=0/0, ttl=51 (request in 1298)
1300	62.287265	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=1/256, ttl=64 (reply in 1301)
1301	62.330994	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=1/256, ttl=51 (request in 1300)
1302	62.389016	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=2/512, ttl=64 (reply in 1303)
1303	63.333109	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=2/512, ttl=51 (request in 1302)
1304	64.294447	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=3/768, ttl=64 (reply in 1305)
1305	64.337072	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=3/768, ttl=51 (request in 1304)
1306	65.299799	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=4/1024, ttl=64 (reply in 1307)
1307	65.343700	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=4/1024, ttl=51 (request in 1306)
1310	66.303536	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=5/1280, ttl=64 (reply in 1311)
1311	66.347238	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=5/1280, ttl=51 (request in 1310)
1312	67.308833	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=6/1536, ttl=64 (reply in 1313)
1313	67.352696	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=6/1536, ttl=51 (request in 1312)
1314	68.314138	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=7/1792, ttl=64 (reply in 1315)
1315	68.358024	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=7/1792, ttl=51 (request in 1314)
1316	69.319368	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=8/2048, ttl=64 (reply in 1321)
1321	70.359749	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=8/2048, ttl=51 (request in 1319)
1322	70.321928	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=9/2304, ttl=64 (reply in 1323)
1323	70.356630	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=9/2304, ttl=51 (request in 1322)
1324	71.327264	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=10/2560, ttl=64 (reply in 1325)
1325	71.371092	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=10/2560, ttl=51 (request in 1324)
1326	72.339085	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=11/2816, ttl=64 (reply in 1329)
1329	72.374614	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=11/2816, ttl=51 (request in 1328)
1330	73.334962	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=12/3072, ttl=64 (reply in 1331)
1331	73.378588	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=12/3072, ttl=51 (request in 1330)
1332	74.337959	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=13/3328, ttl=64 (reply in 1333)
1333	74.381555	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=13/3328, ttl=51 (request in 1332)
1334	75.339822	192.168.1.22	140.82.121.3	ICMP	98	Echo (ping) request id=0x304e, seq=14/3584, ttl=64 (reply in 1335)
1335	75.385788	140.82.121.3	192.168.1.22	ICMP	98	Echo (ping) reply id=0x304e, seq=14/3584, ttl=51 (request in 1334)

Frame 1298: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)  
Ethernet II, Src: Apple\_20:fc:3e (0:c:9:0:6:20), Dst: HuaweiTechno\_a2:bb:d8 (c:0:e3:fb)  
Internet Protocol Version 4, Src: 192.168.1.22, Dst: 140.82.121.3  
Internet Control Message Protocol  
Type: 8 (Echo (ping) request)

0000 c9 e3 fb a2 bb d8 ac c9 06 20 fc 3e 08 00 45 00 . . . . . > E-  
00 54 28 33 00 00 40 01 8d 62 c9 a8 01 16 8c 52 T(3-@ . b . . R  
0020 79 03 08 00 4b 36 30 4c 00 00 67 32 43 cf 00 09 y . . . K60N . g2C . . .  
0030 e6 6d 08 09 1e 1f 20 11 12 13 14 15 m . . . . .  
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 . . . . . %\$  
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 2g 2h 2i 2j 2k 2l & ()\*,- ./012345

Packets: 1367 - Displayed: 38 (2.8%)

Profile: Default

6. What is the value of the User-Agent header field of HTTP requests sent by your browser?

- We use the http filter to locate an HTTP GET request sent by our browser.
- In the packet details, we expand the HTTP layer and look for the "User-Agent" field under "Request Header."
- This field contains our browser's User-Agent string.

**Answer:** Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.0.0 Safari/537.36

```

No. | Time | Source | Destination | Protocol | Length | Info
---|---|---|---|---|---|---
585 42.660055 199.231.164.68 192.168.1.22 HTTP 553 HTTP/1.1 400 Bad Request (text/html)
589 42.660441 192.168.1.22 199.231.164.68 HTTP 73 Continuation
600 42.962678 199.231.164.68 192.168.1.22 HTTP 553 HTTP/1.1 400 Bad Request (text/html)
604 42.963071 192.168.1.22 199.231.164.68 HTTP 73 Continuation
658 44.221758 192.168.1.22 199.231.164.68 HTTP 525 GET / HTTP/1.1
668 44.372647 199.231.164.68 192.168.1.22 HTTP 641 HTTP/1.1 301 Moved Permanently (text/html)
662 44.376431 192.168.1.22 199.231.164.68 HTTP 530 GET /faqs/ HTTP/1.1
667 44.528267 199.231.164.68 192.168.1.22 HTTP 143 HTTP/1.1 200 OK (text/html)
672 44.544367 192.168.1.22 199.231.164.68 HTTP 402 GET /style/faqs.css HTTP/1.1
679 44.693576 192.168.1.22 199.231.164.68 HTTP 384 GET /style/rs.js HTTP/1.1
688 44.693736 192.168.1.22 199.231.164.68 HTTP 381 GET /utils.js HTTP/1.1
685 44.694229 192.168.1.22 199.231.164.68 HTTP 453 GET /images/faqs.org.png HTTP/1.1
686 44.694277 192.168.1.22 199.231.164.68 HTTP 452 GET /images/library.jpg HTTP/1.1
688 44.695698 199.231.164.68 192.168.1.22 HTTP 688 HTTP/1.1 200 OK (text/css)
694 44.844928 199.231.164.68 192.168.1.22 HTTP 910 HTTP/1.1 200 OK (application/javascript)
700 44.955567 199.231.164.68 192.168.1.22 HTTP 835 HTTP/1.1 200 OK (PNG)
701 44.955568 199.231.164.68 192.168.1.22 HTTP 727 HTTP/1.1 200 OK (application/javascript)
709 44.963404 192.168.1.22 199.231.164.68 HTTP 466 GET /style/i/faqs-header.png HTTP/1.1
723 45.033035 192.168.1.22 216.239.38.120 HTTP 412 GET /coop/cse/brand?form=cse-search-box&lang=en HTTP/1.1
739 45.101587 216.239.38.120 192.168.1.22 HTTP 642 HTTP/1.1 301 Moved Permanently (text/html)
751 45.114557 199.231.164.68 192.168.1.22 HTTP 729 HTTP/1.1 200 OK (PNG)
955 45.360285 199.231.164.68 192.168.1.22 HTTP 1278 HTTP/1.1 200 OK (JPEG/JFIF image)
1190 46.805901 192.168.1.22 199.231.164.68 HTTP 605 GET /favicon.ico HTTP/1.1
1195 46.959827 199.231.164.68 192.168.1.22 HTTP 1189 HTTP/1.1 200 OK (image/vnd.microsoft.icon)

> Internet II, Src: Apple_2e:TC:se (aci:cm:00:2e:TC:se), Dst: huaweiiecnno_82:00:00 (ce:ce:ce:TD:00:00)
> Internet Protocol Version 4, Src: 192.168.1.22, Dst: 199.231.164.68
> Transmission Control Protocol, Src Port: 58235, Dst Port: 80, Seq: 460, Ack: 576, Len: 460
  Hypertext Transfer Protocol
    > GET /faqs/ HTTP/1.1\r\n
      Host: www.faqs.org\r\n
      Connection: keep-alive\r\n
      Cache-Control: max-age=0\r\n
      Upgrade-Insecure-Requests: 1\r\n
      User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.0.0 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
      \r\n
    [Response in frame: 667]
    [Full request URI: http://www.faqs.org/faqs/]

```

Packets: 1367 - Displayed: 24 (1.8%) Profile: Default

7. How many .js files has/have been downloaded from http://www.faqs.org? What is/are the name(s) of these file(s)?

- We apply the http filter to locate all HTTP responses from http://www.faqs.org.
- We check the responses for entries where the "Content-Type" is "application/javascript" or the requested file has a .js extension.
- We count the number of .js files downloaded and note their names.

**Answer:** two JavaScript files, named rs.js and utils.js.

http && frame contains ".js"

No.	Time	Source	Destination	Protocol	Length	Info
679	44.693576	192.168.1.22	199.231.164.68	HTTP	384	GET /style/rs.js HTTP/1.1
680	44.693736	192.168.1.22	199.231.164.68	HTTP	381	GET /utils.js HTTP/1.1
739	45.181587	216.239.38.120	192.168.1.22	HTTP	642	HTTP/1.1 301 Moved Permanently (text/html)

```
> Frame 679: 384 bytes on wire (3072 bits), 384 bytes captured (3072 bits)
> Ethernet II, Src: Apple_20:fc:3e (ac:c9:06:20:fc:3e), Dst: HuaweiTechno_a2:bb:d8 (c0:e3:f0)
> Internet Protocol Version 4, Src: 192.168.1.22, Dst: 199.231.164.68
> Transmission Control Protocol, Src Port: 58236, Dst Port: 80, Seq: 1, Ack: 1, Len: 318
< Hypertext Transfer Protocol
> GET /style/rs.js HTTP/1.1\r\n
Host: www.faqs.org\r\n
Connection: keep-alive\r\n
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3990.72 Safari/537.36\r\n
Referer: http://www.faqs.org/faqs/\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: tr-TR,tr;q=0.9\r\n
\r\n
[Response in frame: 701]
[Full request URI: http://www.faqs.org/style/rs.js]
```

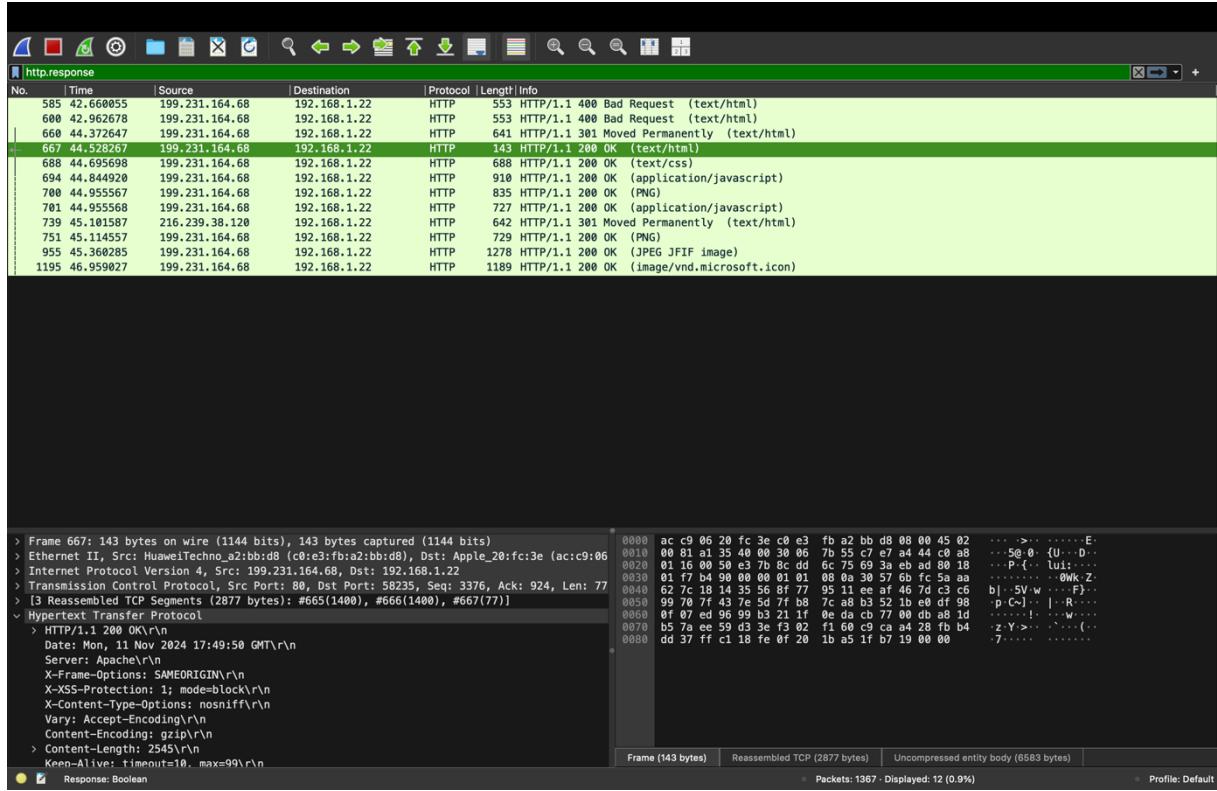
0000 c0 e3 fb a2 bb d8 ac c0 06 20 fc 3e 08 00 45 02 ..... .->..E-
0010 01 72 00 00 40 00 00 06 0b 9c c0 a8 01 16 c7 57 r-@. .....
0020 a4 44 e3 7c 00 50 52 f4 cf 53 f1 3a 63 0f 88 18 D-I PR- S:c...
0030 08 08 ab 4d 00 00 01 01 08 0a 9d 56 02 74 30 57 ..M... ..V:t0W
0040 6c a2 47 45 54 20 2f 73 74 79 6c 65 2f 72 73 2e l-GET /s tyle/rs.
0050 6a 73 28 48 54 54 58 2f 31 2e 31 0d 0a 48 6f 73 js HTTP/ 1.1-Hos
0060 74 3a 28 77 77 2e 66 61 71 73 2e 67 72 67 0a t: www.f aqs.org.
0070 0a 43 6f 6e 65 63 74 69 67 6e 3a 2a 6b 65 65 Connect ion: kee
0080 70 2d 61 6c 69 76 65 0d 0a 55 73 65 72 2d 41 67 p-alive ..User-Ag
0090 65 6c 28 4d 61 63 66 73 67 65 66 74 67 68 3b 20 49 66 74 amilox osx Int
00a0 20 28 4d 61 63 66 73 67 65 66 74 67 68 3b 20 49 66 74 (Macint osx Int
00b0 65 6c 28 4d 61 63 28 4f 53 20 58 20 31 30 5f 31 el Mac O S X 10\_1
00c0 35 5f 37 29 20 41 78 70 6c 65 57 65 62 4b 69 74 5.7) App leWebKit
00d0 2f 35 33 37 2e 33 36 28 28 48 54 4d 4c 2c 20 /537.36
00e0 6c 69 6b 65 20 47 65 63 6b 6f 29 20 43 68 72 6f (KHTML, like Gec ko) Chro
00f0 6d 65 2f 31 33 30 2e 30 2e 30 20 53 61 66 me/130.0 .0.0 Saf
0100 61 72 69 2f 35 33 37 2e 33 36 0d 0a 41 63 63 65 ar/537. 36..Acce
0110 70 74 3a 20 2a 2f 2a 0d 0a 52 65 66 65 72 65 72 pt: \*/\*..Referer

Packets: 1367 - Displayed: 3 (0.2%) Profile: Default

## 8. What is the Status Code of HTTP response for <http://www.faqs.org>?

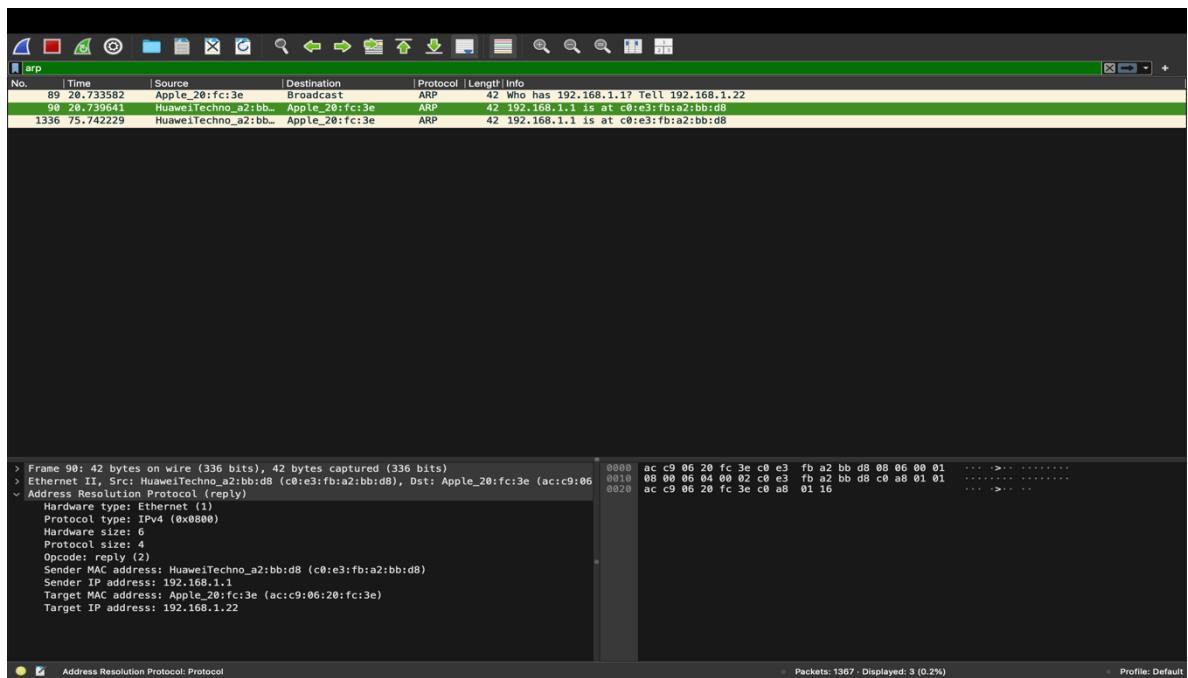
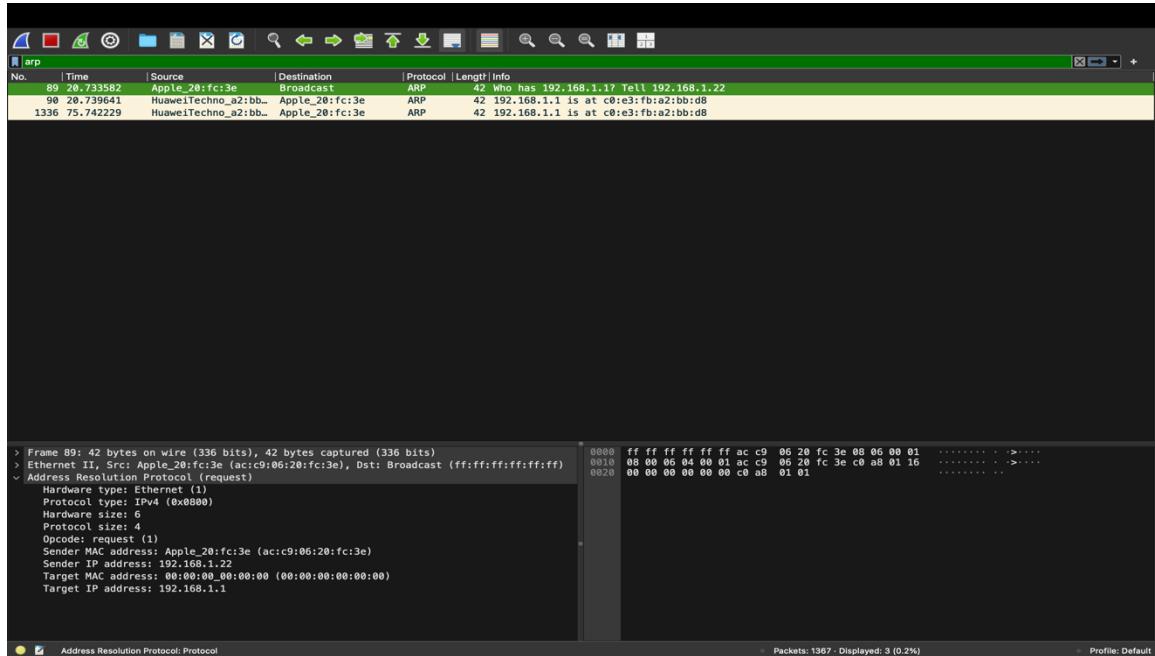
- We apply the http filter to find an HTTP response from www.faqs.org.
- In the packet details, we expand the HTTP header to locate the "Status Code" field, which shows the server's response code (e.g., 200 OK, 404 Not Found).

**Answer:** 200 OK



9. Locate an ARP request and reply pair. What are the Sender and Target MAC addresses, and Sender and Target IP addresses in the ARP request and reply packets?

- We use the arp filter to locate an ARP request packet.
- In the packet details, we note the Sender and Target MAC and IP addresses.
- Then we find the corresponding ARP reply packet, which contains the same addresses in reverse roles.



10. Write a Wireshark filter for showing packets where your IP address is the source and UDP is used. What is the application layer protocol that appears the most when you apply this filter?

- After applying the filter, we look at the Info column to observe the application layer protocol for each packet.
- We identify the most frequently appearing application layer protocol, which is likely to be DNS or another UDP-based protocol.

