

### Wireshark Lab: IP & ARP Assignment

1. Select the first UDP segment sent by your computer via the traceroute command to gaia.cs.umass.edu. (Hint: this is 44th packet in the trace file in the ipwireshark-trace1-1.pcapng file in footnote 2). Expand the Internet Protocol part of the packet in the packet details window. What is the IP address of your computer?

My computer IP address is 192.168.86.61 that is shown in the screenshot below.

```
Identification: 0xfda1 (64929)
> Flags: 0x00
...0 0000 0000 0000 = Fragment Offset: 0
> Time to Live: 1
Protocol: UDP (17)
Header Checksum: 0x2faa [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.86.61
Destination Address: 128.119.245.12
> User Datagram Protocol, Src Port: 64928, Dst Port: 33435
> Data (28 bytes)
```

2. What is the value in the time-to-live (TTL) field in this IPv4 datagram's header?

Time to Live: 1 that is shown in the screenshot below.

```
Identification: 0xfda1 (64929)
> Flags: 0x00
...0 0000 0000 0000 = Fragment Offset: 0
> Time to Live: 1
Protocol: UDP (17)
Header Checksum: 0x2faa [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.86.61
Destination Address: 128.119.245.12
> User Datagram Protocol, Src Port: 64928, Dst Port: 33435
> Data (28 bytes)
```

3. What is the value in the upper layer protocol field in this IPv4 datagram's header? [Note: the answers for Linux/macOS differ from Windows here].

UDP value is 17 that is shown in the screenshot below.

```
Identification: 0xfda1 (64929)
> Flags: 0x00
...0 0000 0000 0000 = Fragment Offset: 0
> Time to Live: 1
Protocol: UDP (17)
Header Checksum: 0x2faa [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.86.61
Destination Address: 128.119.245.12
> User Datagram Protocol, Src Port: 64928, Dst Port: 33435
> Data (28 bytes)
```

#### 4. How many bytes are in the IP header?

Header Length: 20 Bytes (5) that is shown in the screenshot below.

```
> Frame 44: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface en0, id 0
> Ethernet II, Src: Apple_98:d9:27 (78:4f:43:98:d9:27), Dst: Google_89:0e:c8 (3c:28:6d:89:0e:c8)
< Internet Protocol Version 4, Src: 192.168.86.61, Dst: 128.119.245.12
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 56
    Identification: 0xfda1 (64929)
  > Flags: 0x00
    ...0 0000 0000 0000 = Fragment Offset: 0
  > Time to Live: 1
```

#### 5. How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.

There are 20 bytes in the IP header which leaves 36 bytes for the payload of the IP datagram because we were sending a packet of length 56 bytes. It is shown in the screenshot below.

```
> Frame 44: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface en0, id 0
> Ethernet II, Src: Apple_98:d9:27 (78:4f:43:98:d9:27), Dst: Google_89:0e:c8 (3c:28:6d:89:0e:c8)
< Internet Protocol Version 4, Src: 192.168.86.61, Dst: 128.119.245.12
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 56
    Identification: 0xfda1 (64929)
  > Flags: 0x00
    ...0 0000 0000 0000 = Fragment Offset: 0
  > Time to Live: 1
```

#### 6. Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.

The fragment offset is set to 0, therefore, the packet has not been fragmented.?????

```
< Internet Protocol Version 4, Src: 192.168.86.61, Dst: 128.119.245.12
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 56
    Identification: 0xfda1 (64929)
  < Flags: 0x00
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment Offset: 0
```

## 7. Which fields in the IP datagram always change from one datagram to the next within this series of UDP segments sent by your computer destined to 128.119.245.12, via traceroute? Why?

Fields that must change:

The header checksum is always changing because header changes and identification is always changing to verify packets. Time to live is also changing but not always.

First datagram – identification: 0xfda1, Header Checksum: 0x2faa,

162	9.558934	40.78.128.150	192.168.86.61	TLSv1.2	108	Application Data
168	9.631180	192.168.86.61	40.78.128.150	TLSv1.2	111	Application Data
169	9.695860	40.78.128.150	192.168.86.61	TLSv1.2	411	Application Data
171	10.244346	128.119.240.65	192.168.86.61	TLSv1.2	102	Application Data
172	10.244350	128.119.240.65	192.168.86.61	TLSv1.2	1150	Application Data
175	10.262582	52.114.132.176	192.168.86.61	TLSv1.2	388	Application Data
177	10.289567	192.168.86.61	52.114.132.176	TLSv1.2	242	Application Data
315	15.771113	172.217.7.14	192.168.86.61	TLSv1.2	118	Application Data
44	1.865637	192.168.86.61	128.119.245.12	UDP	70	64928 → 33435 Len=28
48	1.874016	192.168.86.61	128.119.245.12	UDP	70	64928 → 33436 Len=28
50	1.875401	192.168.86.61	128.119.245.12	UDP	70	64928 → 33437 Len=28
52	1.876720	192.168.86.61	128.119.245.12	UDP	70	64928 → 33438 Len=28
56	1.885567	192.168.86.61	128.119.245.12	UDP	70	64928 → 33439 Len=28
58	1.889002	192.168.86.61	128.119.245.12	UDP	70	64928 → 33440 Len=28
60	1.892656	192.168.86.61	128.119.245.12	UDP	70	64928 → 33441 Len=28
62	1.907036	192.168.86.61	128.119.245.12	UDP	70	64928 → 33442 Len=28
64	1.928173	192.168.86.61	128.119.245.12	UDP	70	64928 → 33443 Len=28

  
`Internet Protocol Version 4, Src: 192.168.86.61, Dst: 128.119.245.12
 0100 .... = Version: 4
 .... 0101 = Header Length: 20 bytes (5)
 > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 56
 Identification: 0xfda1 (64929)
 < Flags: 0x00
 0... .... = Reserved bit: Not set
 .0.. .... = Don't fragment: Not set
 ..0. .... = More fragments: Not set
 ...0 0000 0000 0000 = Fragment Offset: 0` 

Secon datagram - identification: 0xfda1, Header Checksum: 0x2fa9

No.	Time	Source	Destination	Protocol	Length	Info
162	9.558934	40.78.128.150	192.168.86.61	TLSv1.2	108	Application Data
168	9.631180	192.168.86.61	40.78.128.150	TLSv1.2	111	Application Data
169	9.695860	40.78.128.150	192.168.86.61	TLSv1.2	411	Application Data
171	10.244346	128.119.240.65	192.168.86.61	TLSv1.2	102	Application Data
172	10.244350	128.119.240.65	192.168.86.61	TLSv1.2	1150	Application Data
175	10.262582	52.114.132.176	192.168.86.61	TLSv1.2	388	Application Data
177	10.289567	192.168.86.61	52.114.132.176	TLSv1.2	242	Application Data
315	15.771113	172.217.7.14	192.168.86.61	TLSv1.2	118	Application Data
44	1.865637	192.168.86.61	128.119.245.12	UDP	70	64928 → 33435 Len=28
48	1.874016	192.168.86.61	128.119.245.12	UDP	70	64928 → 33436 Len=28
50	1.875401	192.168.86.61	128.119.245.12	UDP	70	64928 → 33437 Len=28
52	1.876720	192.168.86.61	128.119.245.12	UDP	70	64928 → 33438 Len=28
56	1.885567	192.168.86.61	128.119.245.12	UDP	70	64928 → 33439 Len=28
58	1.889002	192.168.86.61	128.119.245.12	UDP	70	64928 → 33440 Len=28
60	1.892656	192.168.86.61	128.119.245.12	UDP	70	64928 → 33441 Len=28
62	1.907036	192.168.86.61	128.119.245.12	UDP	70	64928 → 33442 Len=28
64	1.928173	192.168.86.61	128.119.245.12	UDP	70	64928 → 33443 Len=28

  
`Internet Protocol Version 4, Src: 192.168.86.61, Dst: 128.119.245.12
 0100 .... = Version: 4
 .... 0101 = Header Length: 20 bytes (5)
 > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 56
 Identification: 0xfda2 (64930)
 < Flags: 0x00
 0... .... = Reserved bit: Not set
 .0.. .... = Don't fragment: Not set
 ..0. .... = More fragments: Not set
 ...0 0000 0000 0000 = Fragment Offset: 0`

## 8. Which fields in this sequence of IP datagrams (containing UDP segments) stay constant? Why?

The fields that stay constant are; version (since we are using IPv4), header length (since these are UDP packets), source IP (since all packets are sent from my computer), destination IP (since we are sending to the same host), Differentiated Services (since all packets are UDP), Upper Layer Protocol (since these are UDP packets). When we compare this value based on the screenshot below, we can receive that they stay constant.

### First datagram

162	9.558934	40.78.128.150	192.168.86.61	TLSv1.2	108	Application Data
168	9.631180	192.168.86.61	40.78.128.150	TLSv1.2	111	Application Data
169	9.695860	40.78.128.150	192.168.86.61	TLSv1.2	411	Application Data
171	10.244346	128.119.240.65	192.168.86.61	TLSv1.2	102	Application Data
172	10.244350	128.119.240.65	192.168.86.61	TLSv1.2	1150	Application Data
175	10.262582	52.114.132.176	192.168.86.61	TLSv1.2	388	Application Data
177	10.289567	192.168.86.61	52.114.132.176	TLSv1.2	242	Application Data
315	15.771113	172.217.7.14	192.168.86.61	TLSv1.2	118	Application Data
44	1.865637	192.168.86.61	128.119.245.12	UDP	70	64928 → 33435 Len=28
48	1.874016	192.168.86.61	128.119.245.12	UDP	70	64928 → 33436 Len=28
50	1.875401	192.168.86.61	128.119.245.12	UDP	70	64928 → 33437 Len=28
52	1.876720	192.168.86.61	128.119.245.12	UDP	70	64928 → 33438 Len=28
56	1.885567	192.168.86.61	128.119.245.12	UDP	70	64928 → 33439 Len=28
58	1.889002	192.168.86.61	128.119.245.12	UDP	70	64928 → 33440 Len=28
60	1.892656	192.168.86.61	128.119.245.12	UDP	70	64928 → 33441 Len=28
62	1.907036	192.168.86.61	128.119.245.12	UDP	70	64928 → 33442 Len=28
64	1.928173	192.168.86.61	128.119.245.12	UDP	70	64928 → 33443 Len=28

  
`Internet Protocol Version 4, Src: 192.168.86.61, Dst: 128.119.245.12
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 56
Identification: 0xfda1 (64929)
Flags: 0x00
0... .... = Reserved bit: Not set
.0... .... = Don't fragment: Not set
..0. .... = More fragments: Not set
...0 0000 0000 0000 = Fragment Offset: 0`

### Second datagram

No.	Time	Source	Destination	Protocol	Length	Info
162	9.558934	40.78.128.150	192.168.86.61	TLSv1.2	108	Application Data
168	9.631180	192.168.86.61	40.78.128.150	TLSv1.2	111	Application Data
169	9.695860	40.78.128.150	192.168.86.61	TLSv1.2	411	Application Data
171	10.244346	128.119.240.65	192.168.86.61	TLSv1.2	102	Application Data
172	10.244350	128.119.240.65	192.168.86.61	TLSv1.2	1150	Application Data
175	10.262582	52.114.132.176	192.168.86.61	TLSv1.2	388	Application Data
177	10.289567	192.168.86.61	52.114.132.176	TLSv1.2	242	Application Data
315	15.771113	172.217.7.14	192.168.86.61	TLSv1.2	118	Application Data
44	1.865637	192.168.86.61	128.119.245.12	UDP	70	64928 → 33435 Len=28
48	1.874016	192.168.86.61	128.119.245.12	UDP	70	64928 → 33436 Len=28
50	1.875401	192.168.86.61	128.119.245.12	UDP	70	64928 → 33437 Len=28
52	1.876720	192.168.86.61	128.119.245.12	UDP	70	64928 → 33438 Len=28
56	1.885567	192.168.86.61	128.119.245.12	UDP	70	64928 → 33439 Len=28
58	1.889002	192.168.86.61	128.119.245.12	UDP	70	64928 → 33440 Len=28
60	1.892656	192.168.86.61	128.119.245.12	UDP	70	64928 → 33441 Len=28
62	1.907036	192.168.86.61	128.119.245.12	UDP	70	64928 → 33442 Len=28
64	1.928173	192.168.86.61	128.119.245.12	UDP	70	64928 → 33443 Len=28

  
`Internet Protocol Version 4, Src: 192.168.86.61, Dst: 128.119.245.12
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 56
Identification: 0xfda2 (64930)
Flags: 0x00
0... .... = Reserved bit: Not set
.0... .... = Don't fragment: Not set
..0. .... = More fragments: Not set
...0 0000 0000 0000 = Fragment Offset: 0`

## 9. Describe the pattern you see in the values in the Identification field of the IP datagrams being sent by your computer.

According to below two screenshots, the pattern is the IP header Identification field increment with each UDP request and the field increases by one in each strand of echo requests.

First datagram - Identification: 0xfda1 (64929)

44	1.865637	192.168.86.61	128.119.245.12	UDP	70	64928	→	33435	Len=28
48	1.874016	192.168.86.61	128.119.245.12	UDP	70	64928	→	33436	Len=28
50	1.875401	192.168.86.61	128.119.245.12	UDP	70	64928	→	33437	Len=28
52	1.876720	192.168.86.61	128.119.245.12	UDP	70	64928	→	33438	Len=28
56	1.885567	192.168.86.61	128.119.245.12	UDP	70	64928	→	33439	Len=28
58	1.889002	192.168.86.61	128.119.245.12	UDP	70	64928	→	33440	Len=28
60	1.892656	192.168.86.61	128.119.245.12	UDP	70	64928	→	33441	Len=28
62	1.907036	192.168.86.61	128.119.245.12	UDP	70	64928	→	33442	Len=28
64	1.928173	192.168.86.61	128.119.245.12	UDP	70	64928	→	33443	Len=28

  
`Internet Protocol Version 4, Src: 192.168.86.61, Dst: 128.119.245.12
 0100 .... = Version: 4
 .... 0101 = Header Length: 20 bytes (5)
 > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 56
 Identification: 0xfda1 (64929)
 > Flags: 0x00
 0... .... = Reserved bit: Not set
 .0... .... = Don't fragment: Not set
 ..0. .... = More fragments: Not set
 ...0 0000 0000 0000 = Fragment Offset: 0` 

Second datagram - Identification: 0xfda2 (64930)

44	1.865637	192.168.86.61	128.119.245.12	UDP	70	64928	→	33435	Len=28
48	1.874016	192.168.86.61	128.119.245.12	UDP	70	64928	→	33436	Len=28
50	1.875401	192.168.86.61	128.119.245.12	UDP	70	64928	→	33437	Len=28
52	1.876720	192.168.86.61	128.119.245.12	UDP	70	64928	→	33438	Len=28
56	1.885567	192.168.86.61	128.119.245.12	UDP	70	64928	→	33439	Len=28
58	1.889002	192.168.86.61	128.119.245.12	UDP	70	64928	→	33440	Len=28
60	1.892656	192.168.86.61	128.119.245.12	UDP	70	64928	→	33441	Len=28
62	1.907036	192.168.86.61	128.119.245.12	UDP	70	64928	→	33442	Len=28
64	1.928173	192.168.86.61	128.119.245.12	UDP	70	64928	→	33443	Len=28

  
`Internet Protocol Version 4, Src: 192.168.86.61, Dst: 128.119.245.12
 0100 .... = Version: 4
 .... 0101 = Header Length: 20 bytes (5)
 > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 56
 Identification: 0xfda2 (64930)
 > Flags: 0x00
 0... .... = Reserved bit: Not set
 .0... .... = Don't fragment: Not set
 ..0. .... = More fragments: Not set
 ...0 0000 0000 0000 = Fragment Offset: 0` 

## 10. What is the upper layer protocol specified in the IP datagrams returned from the routers? [Note: the answers for Linux/MacOS differ from Windows here].

The upper layer protocol is ICMP (1) in the IP datagrams returned from the routers. It is shown in the screenshot below.

..0.. .... = Don't fragment: Not set
...0. .... = More fragments: Not set
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 64
Protocol: ICMP (1)
Header Checksum: 0xe3d0 [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.86.1
Destination Address: 192.168.86.61
> Internet Control Message Protocol
> Data (28 bytes)

## 11. Are the values in the Identification fields (across the sequence of all of ICMP packets from all of the routers) similar in behavior to your answer to question 9 above?

No, the values in the identification fields are not similar across sequence of all ICMP packets they are; - 1. 0x6889 2. 0x688a 3. 0x688b 4. 0xd5c3. In question 9, all of the identification fields were in order however ICMP packets' identification fields are not in order. Identification fields of first and 4<sup>th</sup> are shown in the screenshot below.

First - Identification: 0x6889 (26761)

```
> Frame 45: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0, id 0
> Ethernet II, Src: Google_89:0e:c8 (3c:28:6d:89:0e:c8), Dst: Apple_98:d9:27 (78:4f:43:98:d9:27)
> Internet Protocol Version 4, Src: 192.168.86.1, Dst: 192.168.86.61
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x6889 (26761)
  > Flags: 0x00
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
```

Fourth - Identification: 0xd5c3 (54723)

```
> Frame 53: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0, id 0
> Ethernet II, Src: Google_89:0e:c8 (3c:28:6d:89:0e:c8), Dst: Apple_98:d9:27 (78:4f:43:98:d9:27)
> Internet Protocol Version 4, Src: 10.0.0.1, Dst: 192.168.86.61
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
    Total Length: 84
    Identification: 0xd5c3 (54723)
  > Flags: 0x00
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 63
```

## 12. Are the values of the TTL fields similar, across all of ICMP packets from all of the routers?

No, the values of TTL fields are not similar across all of ICMP packets the first five values of TTL from ICMP packets are; 1. 64, 2. 64, 3. 64, 4. 63, 5. 61. We can receive top two datagrams' TTL.

First datagram - Time to Live: 64

45	1.868608	192.168.86.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
49	1.875315	192.168.86.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
51	1.876637	192.168.86.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
53	1.880429	10.0.0.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
57	1.888900	10.0.0.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
59	1.892580	10.0.0.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
61	1.906167	96.120.66.9	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
63	1.927998	96.120.66.9	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
66	1.940130	96.120.66.9	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
68	1.950559	68.87.181.105	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
70	1.965187	68.87.181.105	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
72	1.975638	68.87.181.105	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
74	1.990744	96.110.23.101	192.168.86.61	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
76	2.008708	96.110.23.101	192.168.86.61	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
78	2.024870	96.110.23.101	192.168.86.61	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
80	2.044952	162.151.52.226	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
82	2.062966	162.151.52.226	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)

```
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
  Total Length: 84
  Identification: 0x6889 (26761)
> Flags: 0x00
  0... .... = Reserved bit: Not set
  .0. .... = Don't fragment: Not set
  ..0. .... = More fragments: Not set
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 64
  Protocol: ICMP (1)
```

Second datagram - Time to Live: 64

45	1.868608	192.168.86.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
49	1.875315	192.168.86.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
51	1.876637	192.168.86.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
53	1.880429	10.0.0.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
57	1.888900	10.0.0.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
59	1.892580	10.0.0.1	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
61	1.906167	96.120.66.9	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
63	1.927998	96.120.66.9	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
66	1.940130	96.120.66.9	192.168.86.61	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
68	1.950559	68.87.181.105	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
70	1.965187	68.87.181.105	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
72	1.975638	68.87.181.105	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
74	1.990744	96.110.23.101	192.168.86.61	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
76	2.008708	96.110.23.101	192.168.86.61	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
78	2.024870	96.110.23.101	192.168.86.61	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
80	2.044952	162.151.52.226	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)
82	2.062966	162.151.52.226	192.168.86.61	ICMP	98	Time-to-live exceeded (Time to live exceeded in transit)

```

.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
Total Length: 84
Identification: 0x688a (26762)
▼ Flags: 0x00
  0... .... = Reserved bit: Not set
  .0... .... = Don't fragment: Not set
  ..0. .... = More fragments: Not set
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 64
Protocol: ICMP (1)

```

**13. Find the first IP datagram containing the first part of the segment sent to 128.119.245.12 sent by your computer via the traceroute command to gaia.cs.umass.edu, after you specified that the traceroute packet length should be 3000. (Hint: This is packet 179 in the ip-wireshark-trace1-1.pcapng trace file in footnote 2. Packets 179, 180, and 181 are three IP datagrams created by fragmenting the first single 3000-byte UDP segment sent to 128.119.145.12). Has that segment been fragmented across more than one IP datagram? (Hint: the answer is yes2!)**

Yes, that segment been fragmented across more than one IP datagram. The first fragment offset is 0 that means it is first fragment and more fragment is set which means more fragments will follow. It is shown in the screenshot below.

First datagram

▼	Internet Protocol Version 4, Src: 192.168.86.61, Dst: 128.119.245.12
0100	.... = Version: 4
....	0101 = Header Length: 20 bytes (5)
>	Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
	Total Length: 1500
	Identification: 0xfda2 (64930)
▼	Flags: 0x20, More fragments
0...	.... = Reserved bit: Not set
.0...	.... = Don't fragment: Not set
..1.	.... = More fragments: Set
...0	0000 0000 0000 = Fragment Offset: 0

Also, the screenshot below shows that this segment is fragmented.

179	12.788154	192.168.86.61	128.119.245.12	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=0, ID=fda2) [Reassembled in #181]
180	12.788155	192.168.86.61	128.119.245.12	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=1480, ID=fda2) [Reassembled in #181]
•	181	12.788155	192.168.86.61	128.119.245.12	UDP	54 64929 → 33435 Len=2972

**14. What information in the IP header indicates that this datagram been fragmented?**



For 179 More fragments: Set. If more fragments value is set then it means it is fragmented.

179	12.788154	192.168.86.61	128.119.245.12	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=0, ID=fda2) [Reassembled in #181]
180	12.788155	192.168.86.61	128.119.245.12	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=1480, ID=fda2) [Reassembled in #181]
181	12.788155	192.168.86.61	128.119.245.12	UDP	54	64929 → 33435 Len=2972
182	12.792190	192.168.86.61	192.168.86.61	ICMP	590	Time-to-live exceeded (Time to live exceeded in transit)
183	12.792881	192.168.86.61	128.119.245.12	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=0, ID=fda3) [Reassembled in #185]
184	12.792882	192.168.86.61	128.119.245.12	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=1480, ID=fda3) [Reassembled in #185]
185	12.792882	192.168.86.61	128.119.245.12	UDP	54	64929 → 33436 Len=2972
186	12.794526	192.168.86.61	192.168.86.61	ICMP	590	Time-to-live exceeded (Time to live exceeded in transit)
187	12.794636	192.168.86.61	128.119.245.12	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=0, ID=fda4) [Reassembled in #189]
188	12.794637	192.168.86.61	128.119.245.12	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=1480, ID=fda4) [Reassembled in #189]
189	12.794637	192.168.86.61	128.119.245.12	UDP	54	64929 → 33437 Len=2972
190	12.796638	192.168.86.61	192.168.86.61	ICMP	590	Time-to-live exceeded (Time to live exceeded in transit)
191	12.796749	192.168.86.61	128.119.245.12	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=0, ID=fda5) [Reassembled in #193]

```

Internet Protocol Version 4, Src: 192.168.86.61, Dst: 128.119.245.12
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1500
    Identification: 0xfda2 (64930)
  < Flags: 0x20, More fragments
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..1. .... = More fragments: Set
    ...0 0000 0000 0000 = Fragment Offset: 0

```

**15. What information in the IP header for this packet indicates whether this is the first fragment versus a latter fragment?**

When we look at the fragment offset value for 179 offset value is 0 so 179 is the first fragment. 0 0000 0000 0000 = Fragment Offset: 0.

```

0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 1500
  Identification: 0xfda2 (64930)
< Flags: 0x20, More fragments
  0... .... = Reserved bit: Not set
  .0.. .... = Don't fragment: Not set
  ..1. .... = More fragments: Set
  ...0 0000 0000 0000 = Fragment Offset: 0
> Time to Live: 1

```

**16. How many bytes are there in is this IP datagram (header plus payload)?**

There are 20 bytes in the IP header which leaves 1480 bytes for the payload of the IP datagram because we were sending a packet of length 1500 bytes

```

> Frame 179: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface en0, id 0
> Ethernet II, Src: Apple_98:d9:27 (78:4f:43:98:d9:27), Dst: Google_89:0e:c8 (3c:28:6d:89:0e:c8)
< Internet Protocol Version 4, Src: 192.168.86.61, Dst: 128.119.245.12
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1500
    Identification: 0xfda2 (64930)
  < Flags: 0x20, More fragments
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set

```



## 17. Now inspect the datagram containing the second fragment of the fragmented UDP segment. What information in the IP header indicates that this is not the first datagram fragment?

Its fragment offset is 1480 which is 20 bytes lower than 1500 bytes that total length of second fragment. It is shown in the screenshot below.

```

Internet Protocol Version 4, Src: 192.168.86.61, Dst: 128.119.245.12
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1500
    Identification: 0xfda2 (64930)
  < Flags: 0x20, More fragments
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..1. .... = More fragments: Set
  ...0 0101 1100 1000 = Fragment Offset: 1480

```

## 18. What fields change in the IP header between the first and second fragment?

Frame offset and header checksum values are changing. It is shown in the screenshot below.

First fragment – Header Checksum: 0x0a05

```

179 12.788154 192.168.86.61 128.119.245.12 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=fda2) [Reassembled in #181]
180 12.788155 192.168.86.61 128.119.245.12 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=fda2) [Reassembled in #181]
181 12.788155 192.168.86.61 128.119.245.12 UDP 54 64929 → 33435 Len=2972
182 12.792190 192.168.86.61 192.168.86.61 ICMP 590 Time-to-live exceeded (Time to live exceeded in transit)
183 12.792881 192.168.86.61 128.119.245.12 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=fda3) [Reassembled in #185]
184 12.792882 192.168.86.61 128.119.245.12 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=fda3) [Reassembled in #185]
185 12.792882 192.168.86.61 128.119.245.12 UDP 54 64929 → 33436 Len=2972
186 12.794526 192.168.86.61 192.168.86.61 ICMP 590 Time-to-live exceeded (Time to live exceeded in transit)
187 12.794636 192.168.86.61 128.119.245.12 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=fda4) [Reassembled in #189]
188 12.794637 192.168.86.61 128.119.245.12 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=fda4) [Reassembled in #189]
189 12.794637 192.168.86.61 128.119.245.12 UDP 54 64929 → 33437 Len=2972
190 12.796638 192.168.86.61 192.168.86.61 ICMP 590 Time-to-live exceeded (Time to live exceeded in transit)
191 12.796749 192.168.86.61 128.119.245.12 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=fda5) [Reassembled in #193]

  .0.. .... = Don't fragment: Not set
  ..1. .... = More fragments: Set
  ...0 0000 0000 0000 = Fragment Offset: 0
  > Time to Live: 1
    Protocol: UDP (17)
    Header Checksum: 0x0a05 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.86.61
    Destination Address: 128.119.245.12
    [Reassembled IPv4 in frame: 181]
  > Data (1480 bytes)

```

Second fragment – Header Checksum: 0x094c

```

179 12.788154 192.168.86.61 128.119.245.12 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=fda2) [Reassembled in #181]
180 12.788155 192.168.86.61 128.119.245.12 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=fda2) [Reassembled in #181]
181 12.788155 192.168.86.61 128.119.245.12 UDP 54 64929 → 33435 Len=2972
182 12.792190 192.168.86.61 192.168.86.61 ICMP 590 Time-to-live exceeded (Time to live exceeded in transit)
183 12.792881 192.168.86.61 128.119.245.12 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=fda3) [Reassembled in #185]
184 12.792882 192.168.86.61 128.119.245.12 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=fda3) [Reassembled in #185]
185 12.792882 192.168.86.61 128.119.245.12 UDP 54 64929 → 33436 Len=2972
186 12.794526 192.168.86.61 192.168.86.61 ICMP 590 Time-to-live exceeded (Time to live exceeded in transit)
187 12.794636 192.168.86.61 128.119.245.12 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=fda4) [Reassembled in #189]
188 12.794637 192.168.86.61 128.119.245.12 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=fda4) [Reassembled in #189]
189 12.794637 192.168.86.61 128.119.245.12 UDP 54 64929 → 33437 Len=2972
190 12.796638 192.168.86.61 192.168.86.61 ICMP 590 Time-to-live exceeded (Time to live exceeded in transit)
191 12.796749 192.168.86.61 128.119.245.12 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=fda5) [Reassembled in #193]

  .0.. .... = Don't fragment: Not set
  ..1. .... = More fragments: Set
  ...0 0101 1100 1000 = Fragment Offset: 1480
  > Time to Live: 1
    Protocol: UDP (17)
    Header Checksum: 0x094c [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.86.61
    Destination Address: 128.119.245.12
    [Reassembled IPv4 in frame: 181]
  > Data (1480 bytes)

```

19. Now find the IP datagram containing the third fragment of the original UDP segment. What information in the IP header indicates that this is the last fragment of that segment?

The fragment offset is 2960 which means it is not the first fragment and the more fragments flag is not set.

```

Internet Protocol Version 4, Src: 192.168.86.61, Dst: 128.119.245.12
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
    Identification: 0xfda2 (64930)
  < Flags: 0x01
    0... .... = Reserved bit: Not set
    .0.. .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
    ...0 1011 1001 0000 = Fragment Offset: 2960

```

20. How many entries are stored in your ARP cache?

As the below screenshot shows, 7 Entries – 6 Static and 1 Dynamic are stored.

```

C:\Users\msiip>arp -a

Interface: 10.185.242.61 --- 0x15
Internet Address      Physical Address      Type
10.185.242.1          74-4d-28-93-31-44    dynamic
10.185.243.255        ff-ff-ff-ff-ff-ff    static
224.0.0.22            01-00-5e-00-00-16    static
224.0.0.251           01-00-5e-00-00-fb    static
224.0.0.252           01-00-5e-00-00-fc    static
239.255.255.250       01-00-5e-7f-ff-fa    static
255.255.255.255       ff-ff-ff-ff-ff-ff    static
C:\Users\msiip>

```

21. What is contained in each displayed entry of the ARP cache?

As the below screenshot shows, ARP cache contained in each displayed entry IP addresses, types and MAC address.

```

C:\Users\msiip>arp -a

Interface: 10.185.242.61 --- 0x15
Internet Address      Physical Address      Type
10.185.242.1          74-4d-28-93-31-44    dynamic
10.185.243.255        ff-ff-ff-ff-ff-ff    static
224.0.0.22            01-00-5e-00-00-16    static
224.0.0.251           01-00-5e-00-00-fb    static
224.0.0.252           01-00-5e-00-00-fc    static
239.255.255.250       01-00-5e-7f-ff-fa    static
255.255.255.255       ff-ff-ff-ff-ff-ff    static
C:\Users\msiip>

```

## 22. What is the hexadecimal value of the source address in the Ethernet frame containing the ARP request message sent out by your computer?

The hexadecimal value of Source: BelkinIn\_75:b1:52 (c4:41:1e:75:b1:52). It is shown in the screenshot below.

```

v Ethernet II, Src: BelkinIn_75:b1:52 (c4:41:1e:75:b1:52), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  v Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    Address: Broadcast (ff:ff:ff:ff:ff:ff)
    ....1. .... = LG bit: Locally administered address (this is NOT the factory default)
    ....1. .... = IG bit: Group address (multicast/broadcast)
  v Source: BelkinIn_75:b1:52 (c4:41:1e:75:b1:52)
    Address: BelkinIn_75:b1:52 (c4:41:1e:75:b1:52)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
  Type: ARP (0x0806)
v Address Resolution Protocol (request)

0000  ff ff ff ff ff ff c4 41 1e 75 b1 52 08 06 00 01  ....A-u-R-...
0010  08 00 06 04 00 01 c4 41 1e 75 b1 52 80 77 f7 42  ....A-u-R-w-B
0020  00 00 00 00 00 00 80 77 f7 01  ....w-..

```

## 23. What is the hexadecimal value of the destination addresses in the Ethernet frame containing the ARP request message sent out by your computer? And what device (if any) corresponds to that address (e.g., client, server, router, switch or otherwise...)?

The hexadecimal value of Destination: Broadcast (ff:ff:ff:ff:ff:ff). It is shown in the screenshot below.

```

v Ethernet II, Src: BelkinIn_75:b1:52 (c4:41:1e:75:b1:52), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  v Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    Address: Broadcast (ff:ff:ff:ff:ff:ff)
    ....1. .... = LG bit: Locally administered address (this is NOT the factory default)
    ....1. .... = IG bit: Group address (multicast/broadcast)
  v Source: BelkinIn_75:b1:52 (c4:41:1e:75:b1:52)
    Address: BelkinIn_75:b1:52 (c4:41:1e:75:b1:52)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
  Type: ARP (0x0806)
v Address Resolution Protocol (request)

0000  ff ff ff ff ff ff c4 41 1e 75 b1 52 08 06 00 01  ....A-u-R-...
0010  08 00 06 04 00 01 c4 41 1e 75 b1 52 80 77 f7 42  ....A-u-R-w-B
0020  00 00 00 00 00 00 80 77 f7 01  ....w-..

```

## 24. What is the hexadecimal value for the two-byte Ethernet Frame type field. What upper layer protocol does this correspond to?

The hex value for the two-byte Ethernet frame is **ARP (0x0806)**, the corresponding upper layer protocol is ARP. It is shown in the screenshot below.

```

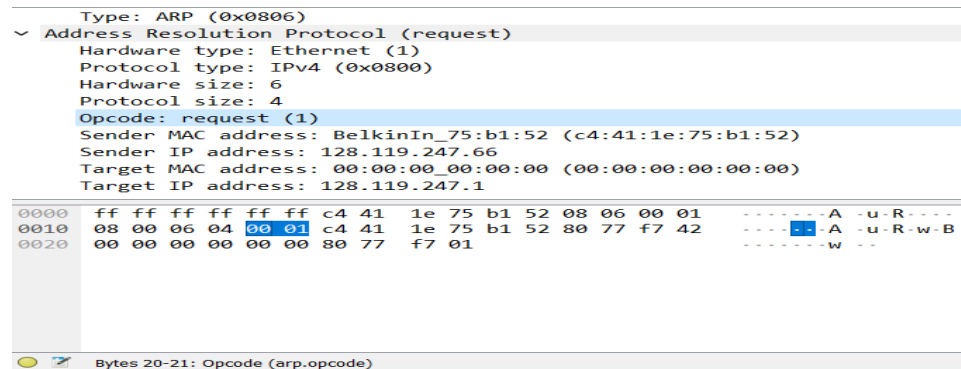
[Coloring Rule String: arp]
v Ethernet II, Src: BelkinIn_75:b1:52 (c4:41:1e:75:b1:52), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  v Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    Address: Broadcast (ff:ff:ff:ff:ff:ff)
    ....1. .... = LG bit: Locally administered address (this is NOT the factory default)
    ....1. .... = IG bit: Group address (multicast/broadcast)
  v Source: BelkinIn_75:b1:52 (c4:41:1e:75:b1:52)
    Address: BelkinIn_75:b1:52 (c4:41:1e:75:b1:52)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
  Type: ARP (0x0806)

0000  ff ff ff ff ff ff c4 41 1e 75 b1 52 08 06 00 01  ....A-u-R-...
0010  08 00 06 04 00 01 c4 41 1e 75 b1 52 80 77 f7 42  ....A-u-R-w-B
0020  00 00 00 00 00 00 80 77 f7 01  ....w-..

```

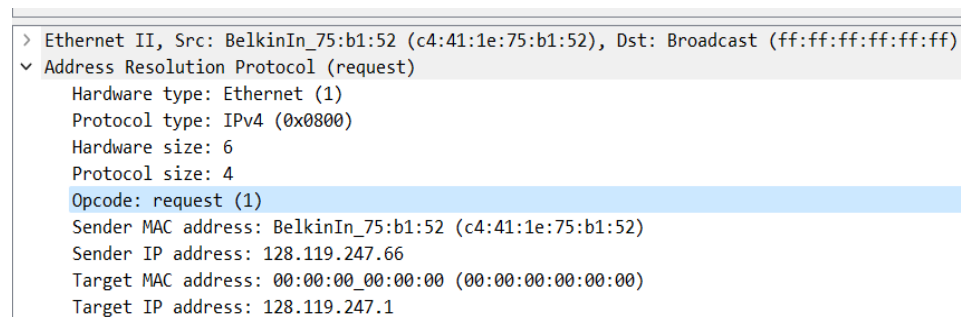
## 25. How many bytes from the very beginning of the Ethernet frame does the ARP opcode field begin?

There are 20 bytes from the very beginning of the Ethernet frame does the ARP opcode field begin. It is shown in the screenshot below.



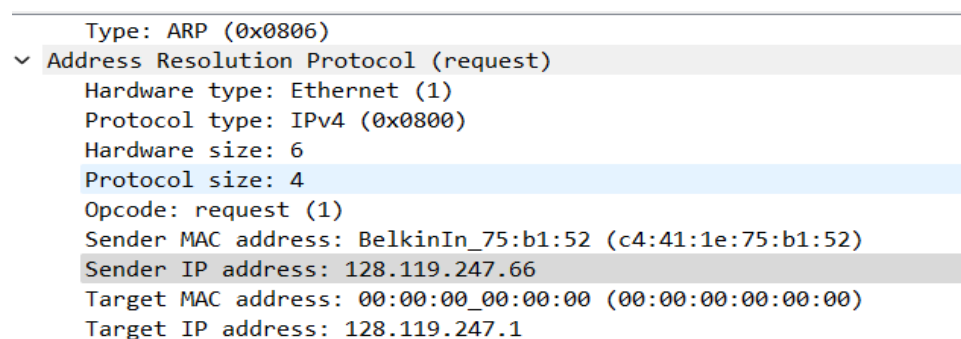
## 26. What is the value of the opcode field within the ARP request message sent by your computer?

Opcode request value is 1 which is shown in the screenshot below.



## 27. Does the ARP request message contain the IP address of the sender? If the answer is yes, what is that value?

Yes, ARP request message contains Sender IP address: 128.119.247.66. It is shown in the screenshot below.



**28. What is the IP address of the device whose corresponding Ethernet address is being requested in the ARP request message sent by your computer?**

It is Target IP address: 128.119.247.1 which is shown in the screenshot below.

```
Type: ARP (0x0806)
  Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: BelkinIn_75:b1:52 (c4:41:1e:75:b1:52)
    Sender IP address: 128.119.247.66
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 128.119.247.1
```

**29. What is the value of the opcode field within the ARP reply message received by your computer?**

Opcode reply value is 2 which is shown in the screenshot below.

```
Padding: 00000000000000000000000000000000
  Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 3ComEuro_7e:d9:01 (00:1e:c1:7e:d9:01)
    Sender IP address: 128.119.247.1
    Target MAC address: BelkinIn_75:b1:52 (c4:41:1e:75:b1:52)
    Target IP address: 128.119.247.66
```

**30. Finally (!), let's look at the answer to the ARP request message! What is the Ethernet address corresponding to the IP address that was specified in the ARP request message sent by your computer (see question 18)?**

Target MAC address: BelkinIn\_75:b1:52 (c4:41:1e:75:b1:52). It is shown in the screenshot below.

```
Padding: 00000000000000000000000000000000
  Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 3ComEuro_7e:d9:01 (00:1e:c1:7e:d9:01)
    Sender IP address: 128.119.247.1
    Target MAC address: BelkinIn_75:b1:52 (c4:41:1e:75:b1:52)
    Target IP address: 128.119.247.66
```