Step 1 & 2. Capture a trace and Inspect a trace

1) Use 'ifconfig' command to find out the my computer's ethernet address.

```
➜  ~ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
XHC20: flags=0<> mtu 0
XHC0: flags=0<> mtu 0
VHC128: flags=0<> mtu 0
en3: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether ac:de:48:00:11:22
        inet6 fe80::aede:48ff:fe00:1122%en3 prefixlen 64 scopeid 0x7
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect (100baseTX <full-duplex>)
        status: active
ap1: flags=8802<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
        ether a6:83:e7:05:d7:85
        media: autoselect
        status: inactive
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether a4:83:e7:05:d7:85
        inet6 fe80::c9a:da16:a250:5209%en0 prefixlen 64 secured scopeid 0x9
```

2) Use 'netstat -r' to check the gateway IP.

```
➜  ~ netstat -r
Routing tables

Internet:
Destination        Gateway            Flags        Refs      Use   Netif Expire
default            192.168.1.1        UGSc          115        0     en0
127                localhost          UCS             0        0     lo0
localhost          localhost          UH             21    15338     lo0
169.254            link#9             UCS             2        0     en0      !
192.168.1          link#9             UCS             1        0     en0      !
192.168.1.1/32     link#9             UCS             1        0     en0      !
192.168.1.1        fc:d7:33:c2:34:14  UHLWIir        53       74     en0   1195
192.168.1.100      e4:9a:dc:28:d:bf   UHLWIi          2       18     en0   1063
192.168.1.103/32   link#9             UCS             0        0     en0      !
224.0.0/4          link#9             UmCS            2        0     en0      !
224.0.0.251        1:0:5e:0:0:fb      UHmLWI          0        0     en0
239.255.255.250    1:0:5e:7f:ff:fa    UHmLWI          0      408     en0
255.255.255.255/32 link#9             UCS             0        0     en0      !
```

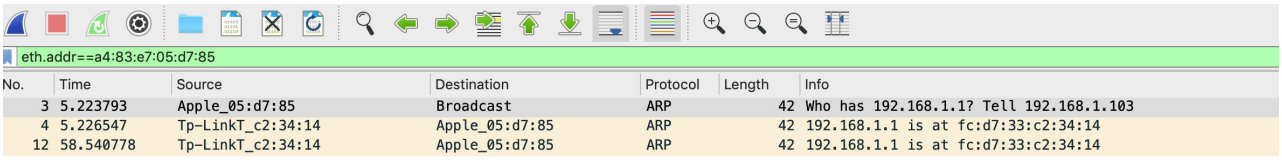
3) Use 'arp -a' to check the local ARP cache.

```
(base) Lius-MacBook-Pro:~ yueyuanj$ arp -a
? (192.168.1.1) at fc:d7:33:c2:34:14 on en0 ifscope [ethernet]
? (192.168.1.100) at e4:9a:dc:28:d:bf on en0 ifscope [ethernet]
? (192.168.1.109) at (incomplete) on en0 ifscope [ethernet]
? (224.0.0.251) at 1:0:5e:0:0:fb on en0 ifscope permanent [ethernet]
? (239.255.255.250) at 1:0:5e:7f:ff:fa on en0 ifscope permanent [etherne
```

4) Use ' arp -d 192.168.1.1' to clear the current arp cache for the IP.

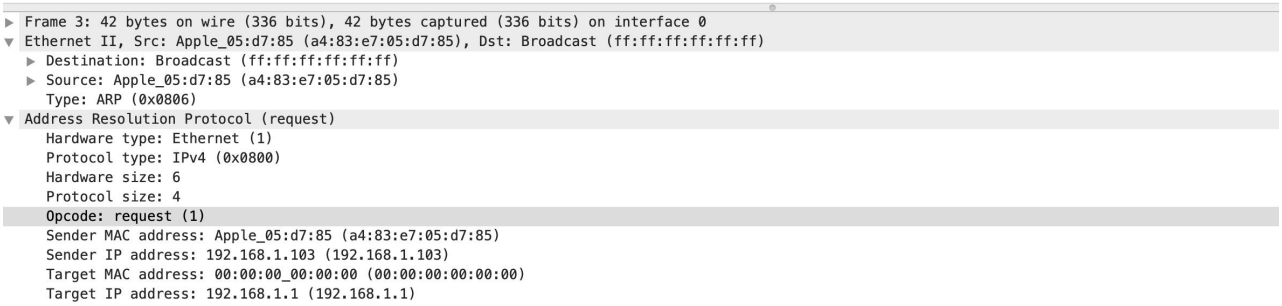
```
(base) Lius-MacBook-Pro:~ yueyuanj$ sudo arp -d 192.168.1.1
192.168.1.1 (192.168.1.1) deleted
```

5) Set up the filter to be ‘arp’ and ‘eth.addr==a4:83:e7:05:d7:85’, capture the following traces.



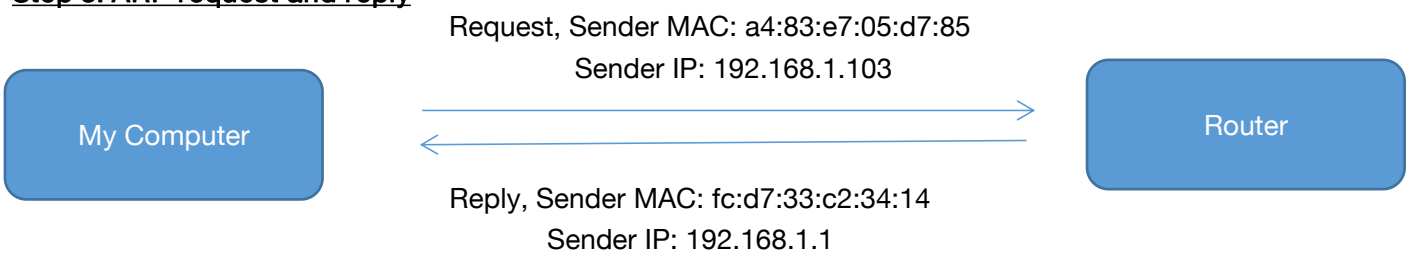
eth.addr==a4:83:e7:05:d7:85

No.	Time	Source	Destination	Protocol	Length	Info
3	5.223793	Apple_05:d7:85	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.103
4	5.226547	Tp-LinkT_c2:34:14	Apple_05:d7:85	ARP	42	192.168.1.1 is at fc:d7:33:c2:34:14
12	58.540778	Tp-LinkT_c2:34:14	Apple_05:d7:85	ARP	42	192.168.1.1 is at fc:d7:33:c2:34:14



```
▶ Frame 3: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▼ Ethernet II, Src: Apple_05:d7:85 (a4:83:e7:05:d7:85), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Source: Apple_05:d7:85 (a4:83:e7:05:d7:85)
    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: Apple_05:d7:85 (a4:83:e7:05:d7:85)
    Sender IP address: 192.168.1.103 (192.168.1.103)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.1.1 (192.168.1.1)
```

Step 3: ARP request and reply



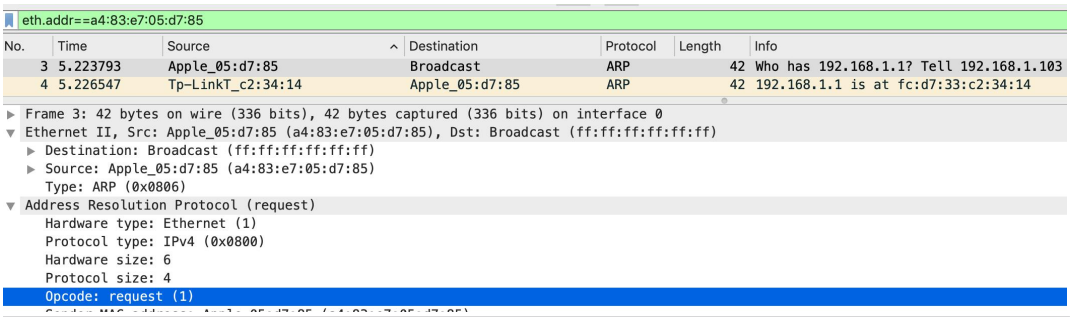
Step 4: Details of ARP over Ethernet

To look at further details of ARP, examine an ARP request and ARP reply to answer these questions:

1. What opcode is used to indicate a request? What about a reply?

--> Opcode 1 is for a request.

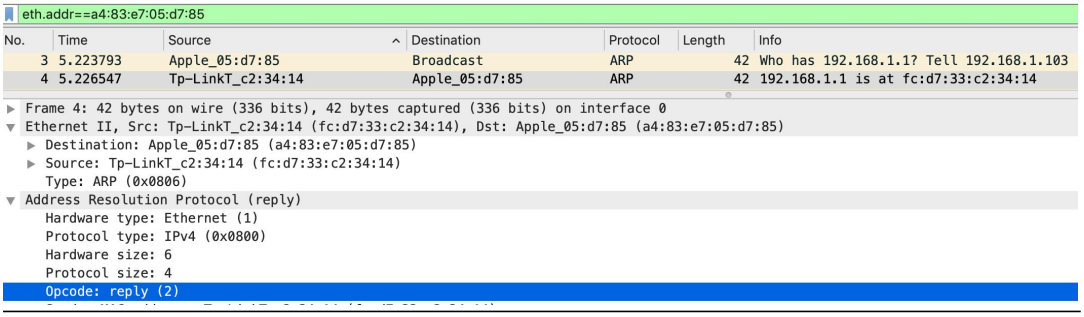
Opcode 2 is for a reply.



eth.addr==a4:83:e7:05:d7:85

No.	Time	Source	Destination	Protocol	Length	Info
3	5.223793	Apple_05:d7:85	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.103
4	5.226547	Tp-LinkT_c2:34:14	Apple_05:d7:85	ARP	42	192.168.1.1 is at fc:d7:33:c2:34:14

```
▶ Frame 3: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▼ Ethernet II, Src: Apple_05:d7:85 (a4:83:e7:05:d7:85), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Source: Apple_05:d7:85 (a4:83:e7:05:d7:85)
    Type: ARP (0x0806)
▼ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
```



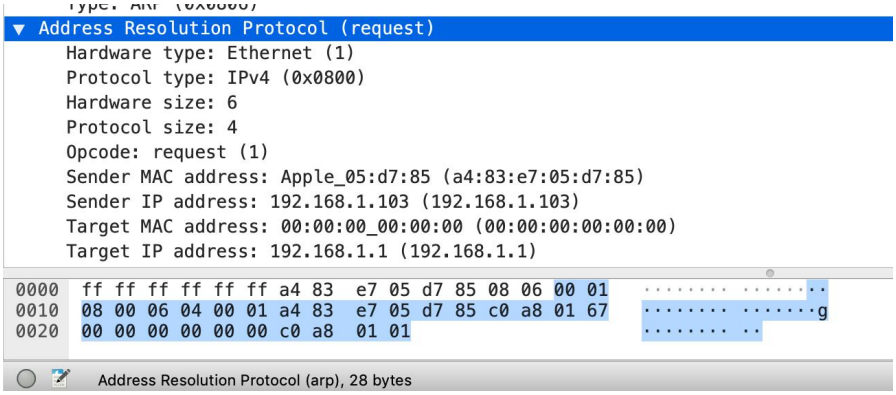
eth.addr==a4:83:e7:05:d7:85

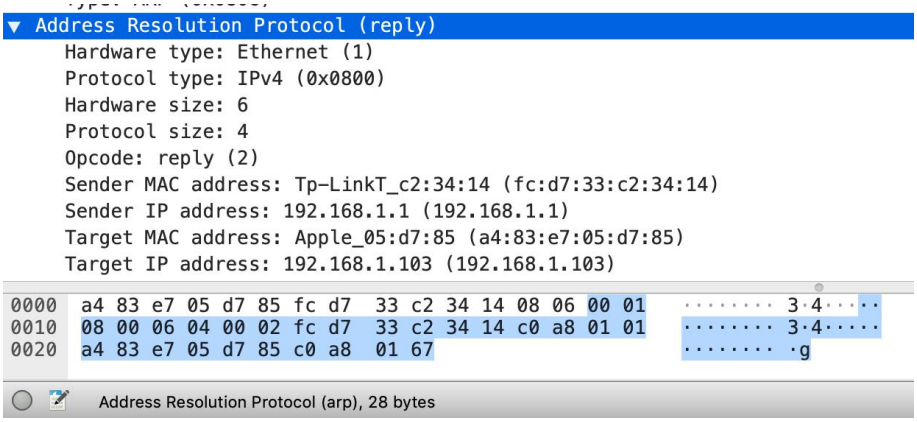
No.	Time	Source	Destination	Protocol	Length	Info
3	5.223793	Apple_05:d7:85	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.103
4	5.226547	Tp-LinkT_c2:34:14	Apple_05:d7:85	ARP	42	192.168.1.1 is at fc:d7:33:c2:34:14

```
▶ Frame 4: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▼ Ethernet II, Src: Tp-LinkT_c2:34:14 (fc:d7:33:c2:34:14), Dst: Apple_05:d7:85 (a4:83:e7:05:d7:85)
  ▶ Destination: Apple_05:d7:85 (a4:83:e7:05:d7:85)
  ▶ Source: Tp-LinkT_c2:34:14 (fc:d7:33:c2:34:14)
    Type: ARP (0x0806)
▼ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
```

2. How large is the ARP header for a request? What about for a reply?

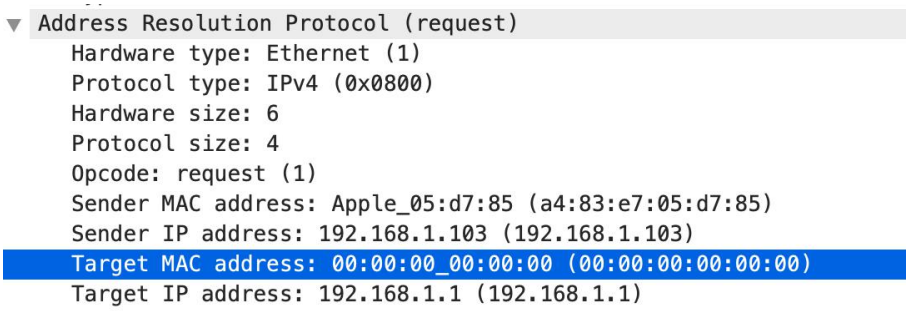
--> Both the request ARP header and the reply header are 28 bytes.





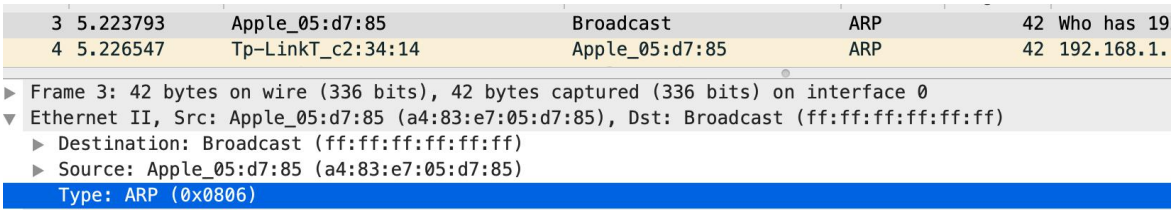
3. What value is carried on a request for the unknown target MAC address?

--> 00:00:00:00:00:00



4. What Ethernet Type value which indicates that ARP is the higher layer protocol?

--> 0x0806



5. Is the ARP reply broadcast (like the ARP request) or not?

--> No, it is not broadcast.