Undergraduate Lab Report of Jinan University

Course Title Computer Networks Lab Evaluation

Lab Name ARP: Address Resolution Protocol

Lab Address N117 Instructor Dr. CUI Lin (崔林)

Student Name 蒋云翔 Student No 2022102330

College International School

Department Major CST

Date 2024 / 11 / 17

1. Introduction

(1) Objective

- Know ARP packet format.
- Understand the operation of ARP cache table.
- Understand how ARP works (including both ARP request and reply messages).

1) Experiment Principle

1. Physical address and logical address:

For communication to occur effectively, devices need to know both the logical address (IP address) and the physical address (MAC address) of the intended recipient devices. Although every device on the Internet has one or more IP addresses, these are not enough to send data packets. Network interface cards (NICs) at the data link layer, like Ethernet cards, do not recognize Internet addresses. For instance, in Ethernet, each NIC comes with a unique 48-bit Ethernet address. These NICs work by sending and receiving frames using these 48-bit addresses, and they are completely unaware of the 32-bit IP addresses.

To make this concept clearer and more concrete, the following diagram shows the contents of a MAC frame, highlighting the detailed and subtle aspects of this crucial networking process:

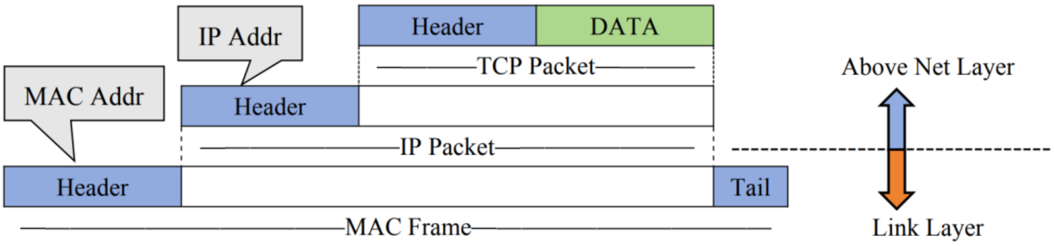


Figure 1: Encapsulation of a MAC Frame

2. ARP protocol introduction

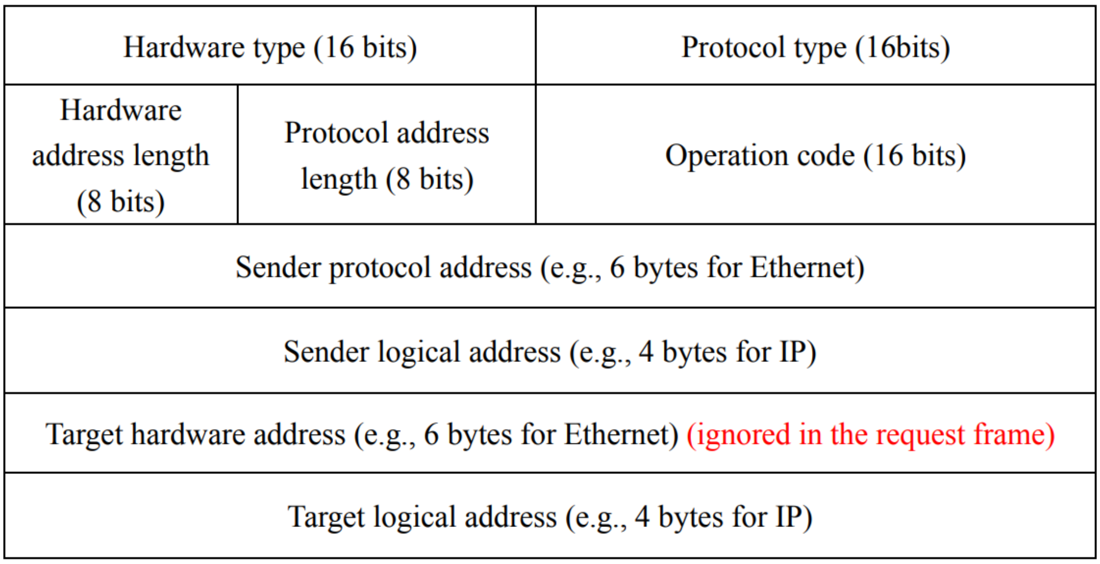
The Address Resolution Protocol (ARP) is a crucial component in networking, designed to help computers find the MAC address associated with a given IP address.

This protocol is widely used on all Internet-connected devices, highlighting its essential function in network communication. ARP is thoroughly described in RFC 826, which offers a detailed specification for its implementation and operation.

Using ARP instead of manual configuration files brings a substantial benefit: ease of use. This method significantly lightens the load for system administrators, who only need to assign an IP address to each device and determine the appropriate subnet masks. After these initial configurations, ARP efficiently manages the complexities of address resolution without further intervention.

3. ARP datagram format

The following figure shows the ARP datagram format:



Hardware type (16 bits)		Protocol type (16bits)
Hardware address length (8 bits)	Protocol address length (8 bits)	Operation code (16 bits)
Sender protocol address (e.g., 6 bytes for Ethernet)		
Sender logical address (e.g., 4 bytes for IP)		
Target hardware address (e.g., 6 bytes for Ethernet) (ignored in the request frame)		
Target logical address (e.g., 4 bytes for IP)		

Figure 2: ARP datagram format

Some fields of ARP datagram format:

(1) **Hardware Type:** 16 bits, this field delineates the specific type of network on which ARP is operational. Each distinct LAN type is assigned a unique integer identifier. For instance, the hardware type for Ethernet is designated as 1. ARP boasts versatility, as it is compatible with an array of network types.

(2) **Protocol Type:** This 16-bit field defines the type of protocol in use. For IPv4, the field holds the value

0x0800. The values permitted in this field are shared in a numbering space with EtherType values, demonstrating ARP's adaptability across various upper layer protocols.

(3) **Hardware Address Length:** Measured in bytes and encapsulated within 8 bits, this field represents the length of the hardware address. For example, the size of an Ethernet address is 6 bytes, translating to a corresponding hardware address length of 6.

(4) **Protocol Address Length:** This 8-bit field specifies the length of the packet's logical address, with byte as the unit of measurement. In the case of IPv4, the size is

4 bytes, resulting in a protocol address length also of 4.

(5) **Operation Code:** Occupying 16 bits, this field indicates the specific operation being conducted by the sender, with 1 signifying a request and 2 denoting a reply.

(6) **Sender Hardware Address:** This field holds the MAC address of the sender, providing a unique identifier for the originating device.

(7) **Sender Protocol Address:** This field specifies the logical address of the sender, such as a 4-byte IP address.

(8) **Target Hardware Address:** In this field, the MAC address of the intended recipient is specified. Notably, if the sender does not know the target's MAC address during an ARP request, this field's value is set to 0.

(9) **Target Protocol Address:** This field distinctly identifies the target's logical address.

4. ARP encapsulation

ARP datagram can be directly encapsulated in the data link frames. The value of Type filed, 0x0806, specifies the frame carries an ARP packet.

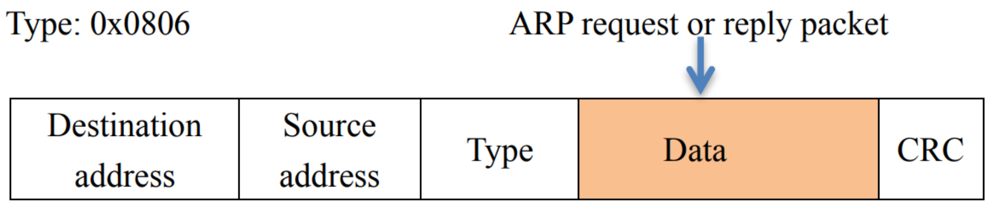


Figure 3: ARP encapsulation

5. ARP operational process

The operations of ARP (Address Resolution Protocol) encompass a comprehensive suite of functionalities, including ARP Request, Checking, ARP Reply, and Caching. ARP Request and Reply packets are transmit ted exclusively when the host lacks knowledge of the target machine's MAC address; once this address is ascertained, it is stored in the ARP Cache for future use, optimizing network efficiency.

Delving into the nuanced steps of the packet transmission process, we observe the steps of packet transmitting
process:

(1) The sender initiates the process by acquiring the IP address of the receiver.

(2) An ARP request incorporates the sender's MAC and IP addresses, alongside the receiver's IP address. And the receiver's MAC address field is set to 0.

(3) The packet then descends to the data link layer, where it is encapsulated within an Ethernet frame. Here, the sender's MAC address is utilized as the source address, while the physical broadcast address (FFFFFFFFFFFF) is designated as the destination address.

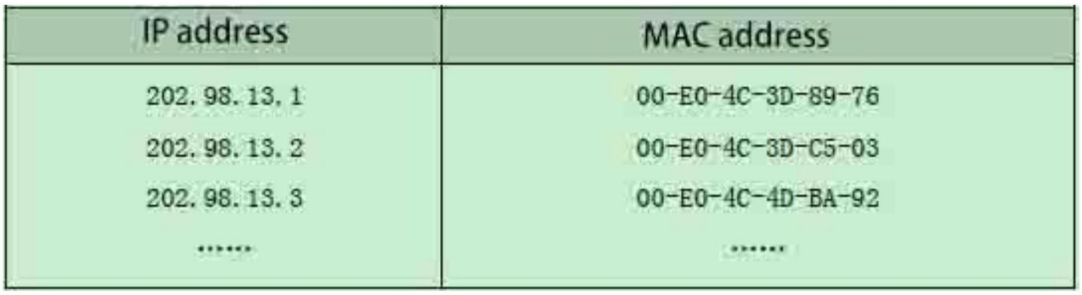
(4) Given that the frame bears a broadcast address, it captures the attention of every host and router within the current network domain. Each entity processes the ARP request using its ARP protocol. However, all parties, save for the intended destination, ultimately discard the request.

(5) In response, the destination host crafts an ARP reply packet, embedding its own MAC address, and sends it directly back to the sender in a unicast fashion.
(6) Upon receipt of the ARP reply, the sender extracts the anticipated MAC address of the destination host, completing the address resolution.

(6) To enhance future communications, the machine caches the newly acquired MAC address, creating an efficient IP to physical address mapping in its ARP table. This proactive measure ensures swift retrieval for subsequent interactions with the same machine, circumventing the need for repetitive broadcasts. Table entries are purged if they are not periodically refreshed, maintaining the table's relevance and accuracy

6. High speed ARP cache

In protocol realization, it's unnecessary to send ARP request packet everytime. Each host maintains a high-speed ARP cache, which is the mapping between IP and MAC address.



IP address	MAC address
202. 98. 13. 1	00-E0-4C-3D-89-76
202. 98. 13. 2	00-E0-4C-3D-C5-03
202. 98. 13. 3	00-E0-4C-4D-BA-92
......	......

Figure 4: ARP high speed cache

Upon initiating packet transmission, a host consults its ARP cache table to ascertain whether the destination MAC address is readily available. Should this be the case, the host utilizes this address as the destination MAC. Otherwise, the host proactively dispatches an ARP request with the aim of retrieving the requisite destination MAC address, subsequently recording this information in the ARP cache table for future reference. The ARP cache table, meticulous in its operation, adheres to a policy of temporal relevance. Entries that have not been utilized over a specified duration are purged from the table, ensuring both the efficiency and currency of the cache.

For users operating on Windows, the system provides a suite of commands to interact with and manage the ARP cache. Utilizing the command "*arp -a*" yields a comprehensive display of the entire ARP cache, while the "*arp -d*" command serves to clear the cache, resetting its contents. The forthcoming figure meticulously illustrates the dynamic process of utilizing and updating the ARP cache.

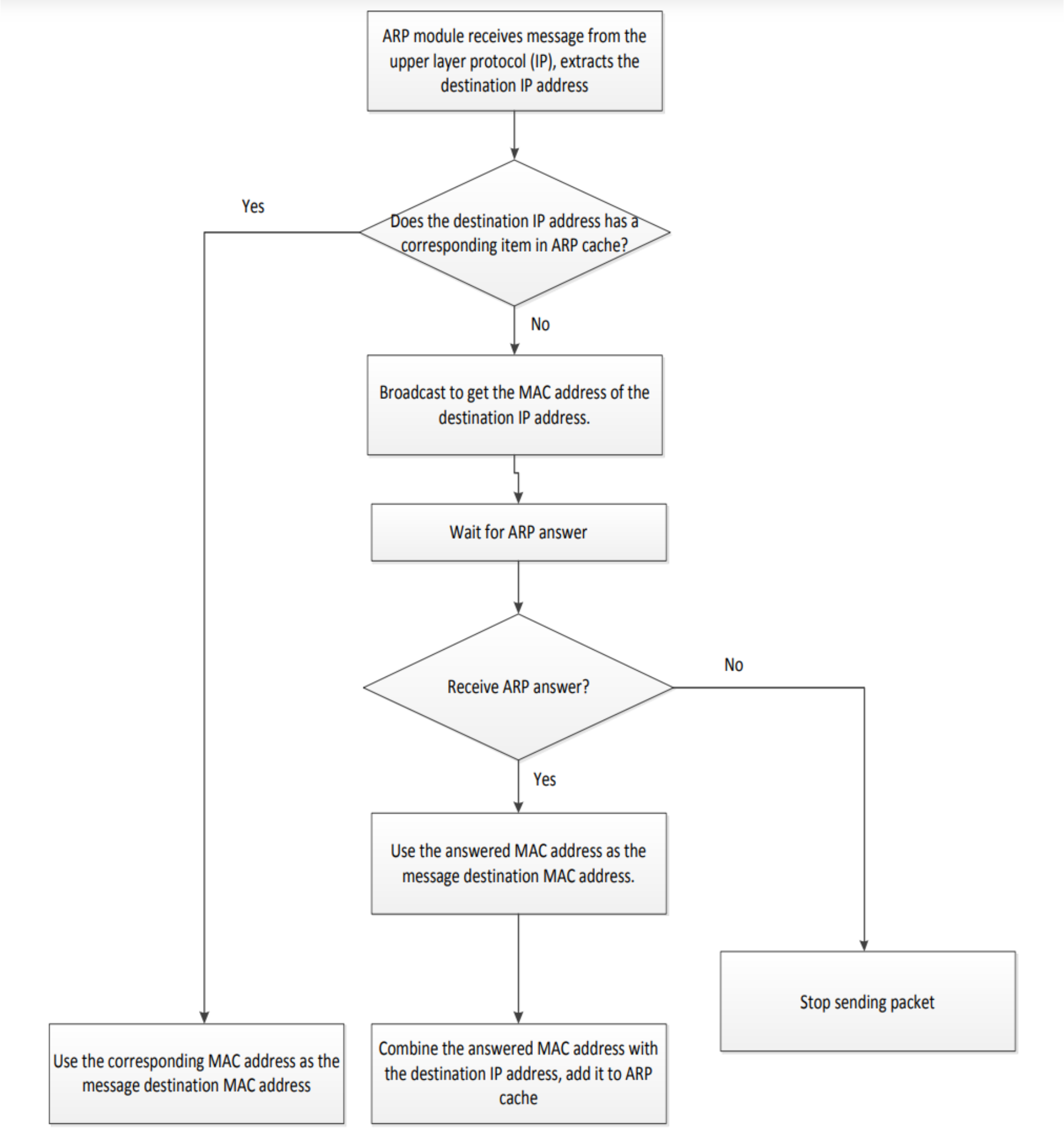

ARP module receives message from the upper layer protocol (IP), extracts the destination IP address
Does the destination IP address has a corresponding item in ARP cache?
Yes
No
Broadcast to get the MAC address of the destination IP address.
Wait for ARP answer
Receive ARP answer?
No
Yes
Use the answered MAC address as the message destination MAC address.
Stop sending packet
Use the corresponding MAC address as the message destination MAC address
Combine the answered MAC address with the destination IP address, add it to ARP cache

Figure 5: Using and updating process of ARP cache

7. ARP proxy

ARP proxy serves as a tool in subnetting, effectively bridging communication between different networks. When an ARP request is transmitted across networks, the intermediary router, serving as a nexus, possesses the capability to respond to this request. Upon receiving an IP packet, this router forwards the packet to the intended destination, be it a host or another router. The following figure provides a clear and illustrative example of this process in action, showcasing how the ARP proxy responds to an ARP request targeted at the destination IP address 141.23.56.23.

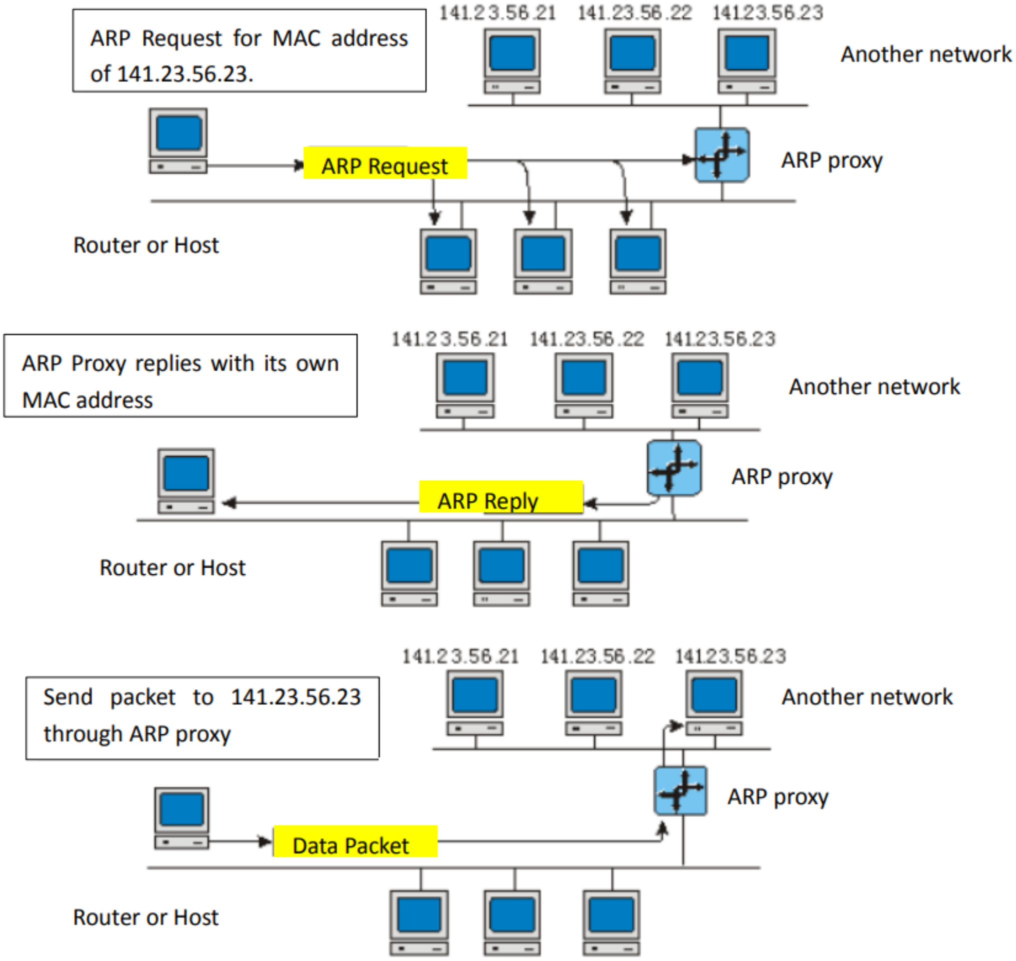
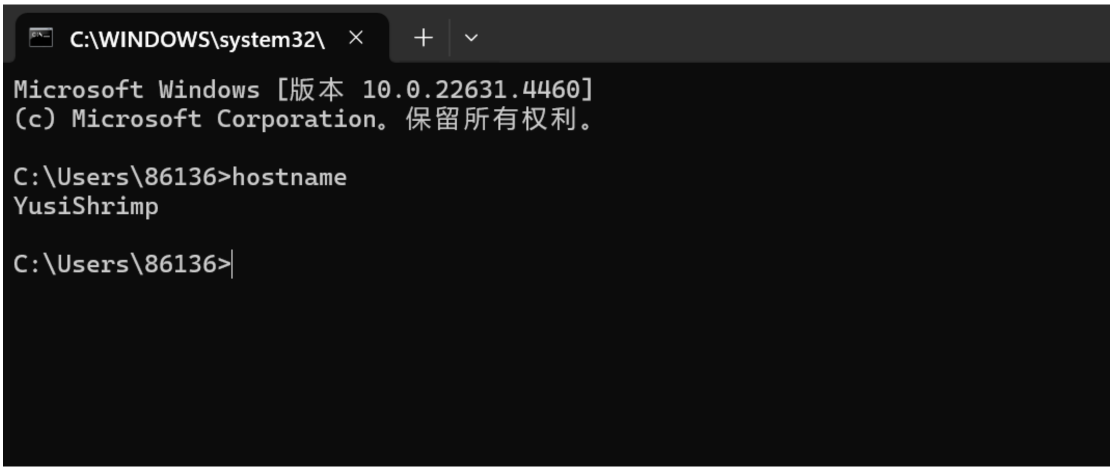


Figure 6: Proxy ARP

2. Lab Environment

I accomplish this LAB at home using a wireless network.

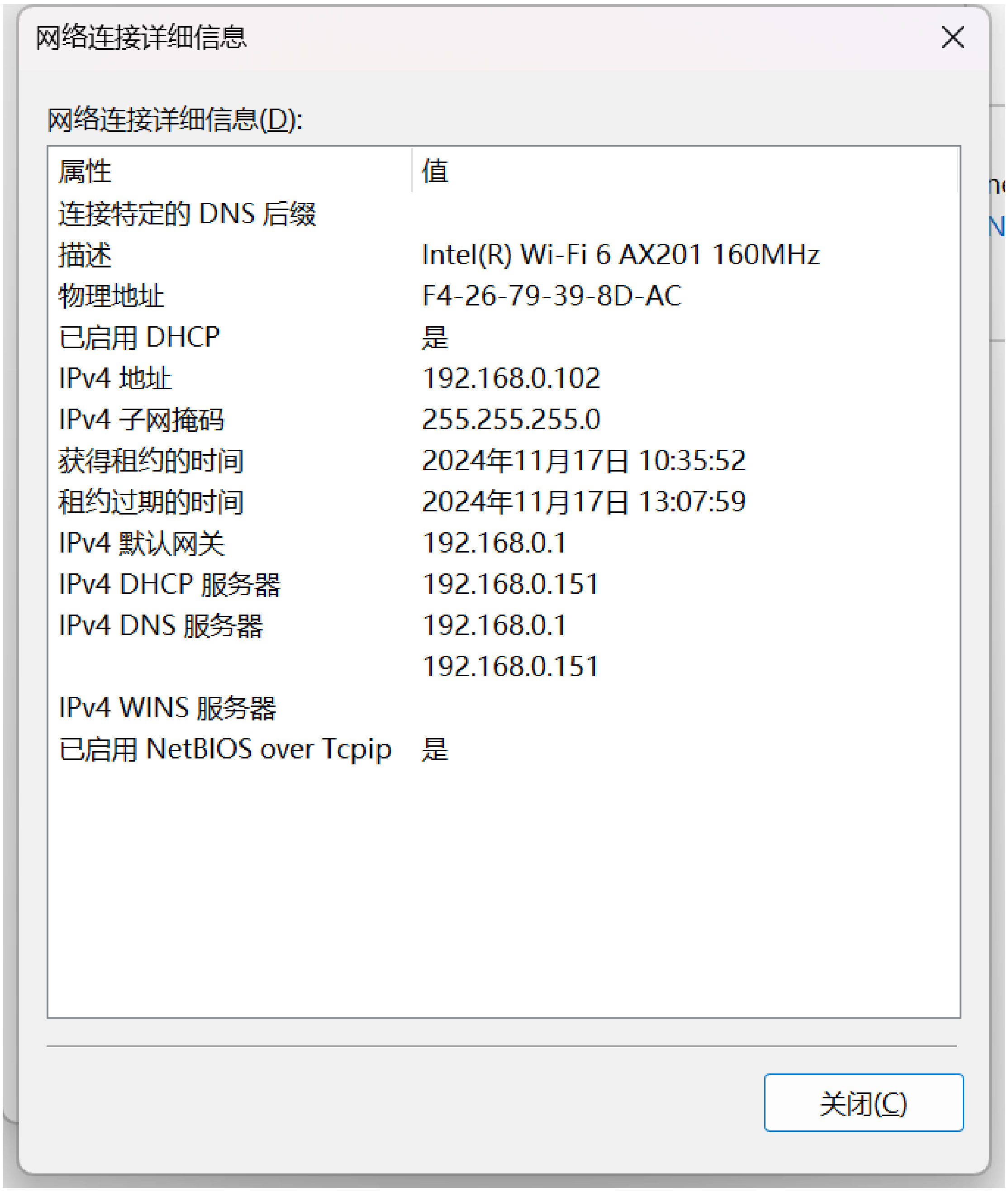
- Computer Name:



```
Microsoft Windows [版本 10.0.22631.4460]
(c) Microsoft Corporation。保留所有权利。

C:\Users\86136>hostname
YusiShrimp

C:\Users\86136>
```

Figure 7: Host name

- Detailed network information:



网络连接详细信息

网络连接详细信息(D):

属性	值
连接特定的 DNS 后缀	
描述	Intel(R) Wi-Fi 6 AX201 160MHz
物理地址	F4-26-79-39-8D-AC
已启用 DHCP	是
IPv4 地址	192.168.0.102
IPv4 子网掩码	255.255.255.0
获得租约的时间	2024年11月17日 10:35:52
租约过期的时间	2024年11月17日 13:07:59
IPv4 默认网关	192.168.0.1
IPv4 DHCP 服务器	192.168.0.151
IPv4 DNS 服务器	192.168.0.1 192.168.0.151
IPv4 WINS 服务器	
已启用 NetBIOS over Tcpip	是

Figure 8: Detailed network information

3. Experiment Steps and Results

1) Task 1: First try to capture a trace of ARP

1. Experiment steps:

[1] **Step 1:** Find the Ethernet address of the main network interface of computer with the ifconfig / ipconfig command. On Windows, execute command “ipconfig /all”. On Mac/Linux, execute command “ifconfig”.

- Among the output will be a section for the main interface of the computer (likely an Ethernet interface) and its Ethernet address. Common names for the interface are “eth0”, “en0”, or “Ethernet adapter”.

• On Windows, you can also obtain the MAC address through the graphic interface.

[2] **Step 2:** Find the IP address of the local router or default gateway that the computer uses to reach the rest of the Internet.

• You can locate the IP address of default gateway in results of last step.

• Or you can use the netstat / route command: You should be able to use the netstat command (“netstat -r” on Windows, Mac and Linux, may require Ctrl-C to stop). Alternatively, you can use the route command (“route print” on Windows, “route” on Linux, “route–n get default” on Mac). In either case you are looking for the gateway IP address that corresponds to the destination of “default” or “0.0.0.0”. See following figure.

[3] **Step 3:** Launch Wireshark and start a capture with a filter of “arp”.

• Remember to choose correct interface.

• We only want to record packets sent to/from your computer. So, the “promiscuous mode” of should be disabled (Normally, this is a default setting of Wireshark).

[4] **Step 4:** When the capture is started, use the “arp -d” command to clear the default gateway from the ARP cache. • Using the command “arp -a” will show you the contents of the ARP cache. You should see an entry for the IP address of the default gateway.

• To clear this entry, use the arp command with different arguments (“arp -d xx.xx.xx.xx”, where xx.xx.xx.xx is the IP address of the default gateway on Linux).

• This usage of arp command may need administrator privileges to run. If so, you may run as a privileged user on Windows or issue “sudo arp -d xx.xx.xx.xx” on Linux/Mac.

• Note that the command should run without error, but the ARP entry may not appear to be cleared if you check with “arp -a”. This is because your computer will send ARP packets to repopulate this entry as soon as you need to send a packet to a remote IP address, and that can happen very quickly due to background activity on the computer.

[5] **Step 5:** Now that the ARP cache have been cleared, fetch a remote page with Web browser. This will cause ARP to find the Ethernet address of the default gateway so that ARP packets can be sent. These ARP packets will be captured by Wireshark.

• You might clear the ARP cache and fetch a webpage a couple of times.

• Hopefully there will also be other ARP packets sent by other computers on the local network that will be captured. These packets are likely to be present if there are other computers on your local network. In fact, if you have a busy computer and extensive local network, you may capture many ARP packets.

• The ARP traffic of other computers will be captured when the ARP packets are sent to the broadcast address, since in this case they are destined for all computers including the one on which you are running Wireshark.

• Because ARP activity happens slowly, you may need to wait up to 30 seconds to observe some of this background ARP traffic

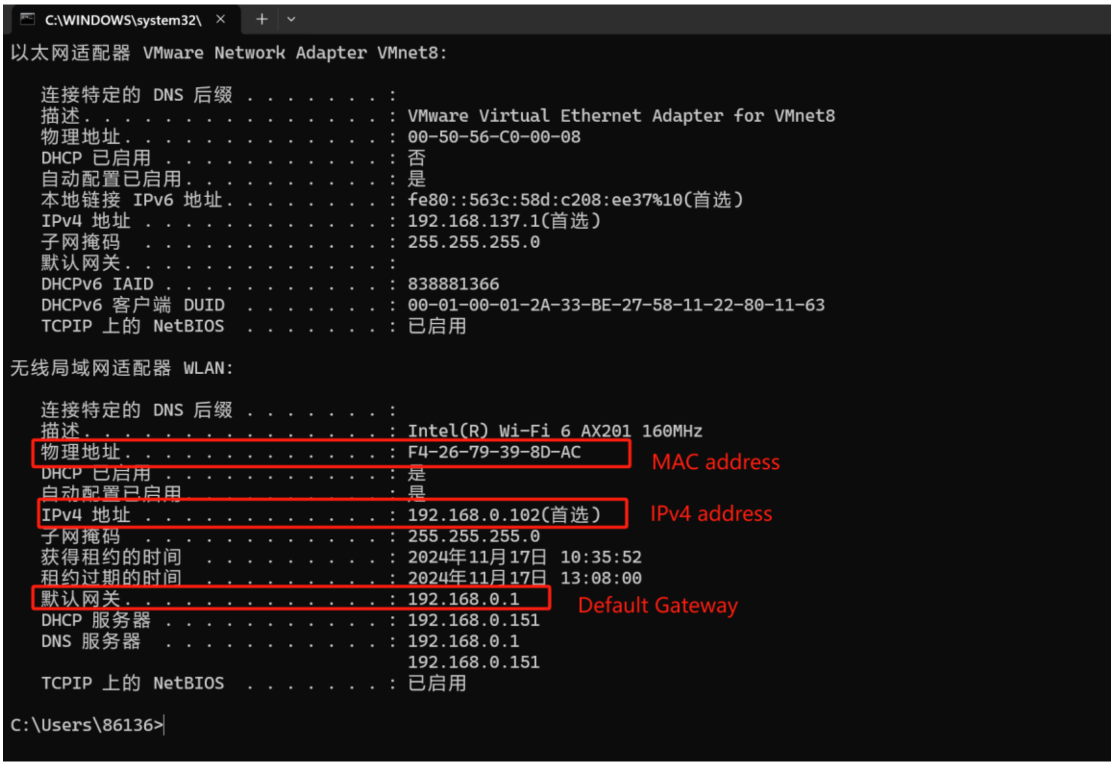
[6] **Step 6:** Once have captured some ARP traffic, stop the capture.

• If there are many ARP packets in your trace, you can narrow our view to only the ARP packets that are sent directly from or to your computer.

• You can set a filter for packets with the Ethernet address of your computer. For example, if your Eth ernet address is 01:02:03:04:05:06 then enter a filter expression of “eth.addr == 01:02:03:04:05:06”.

2. Results

Use the “ipconfig /all” command to obtain the network interface information of my computer. The information includes the computer’s IP address, physical address (MAC address), and the IP address of the default gateway:



```
以太网适配器 VMware Network Adapter VMnet8:

   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : VMware Virtual Ethernet Adapter for VMnet8
   物理地址. . . . . . . . . . . . . : 00-50-56-C0-00-08
   DHCP 已启用 . . . . . . . . . . . : 否
   自动配置已启用. . . . . . . . . . : 是
   本地链接 IPv6 地址. . . . . . . . : fe80::563c:58d:c208:ee37%10(首选)
   IPv4 地址 . . . . . . . . . . . . : 192.168.137.1(首选)
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 838881366
   DHCPv6 客户端 DUID  . . . . . . . : 00-01-00-01-2A-33-BE-27-58-11-22-80-11-63
   TCPIP 上的 NetBIOS  . . . . . . . : 已启用

无线局域网适配器 WLAN:

   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : Intel(R) Wi-Fi 6 AX201 160MHz
   物理地址. . . . . . . . . . . . . : F4-26-79-39-8D-AC
   DHCP 已启用 . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . : 是
   IPv4 地址 . . . . . . . . . . . . : 192.168.0.102(首选)
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   获得租约的时间  . . . . . . . . . : 2024年11月17日 10:35:52
   租约过期的时间  . . . . . . . . . : 2024年11月17日 13:08:00
   默认网关. . . . . . . . . . . . . : 192.168.0.1
   DHCP 服务器 . . . . . . . . . . . : 192.168.0.151
   DNS 服务器  . . . . . . . . . . . : 192.168.0.1
                                       192.168.0.151
   TCPIP 上的 NetBIOS  . . . . . . . : 已启用

C:\Users\86136>
```

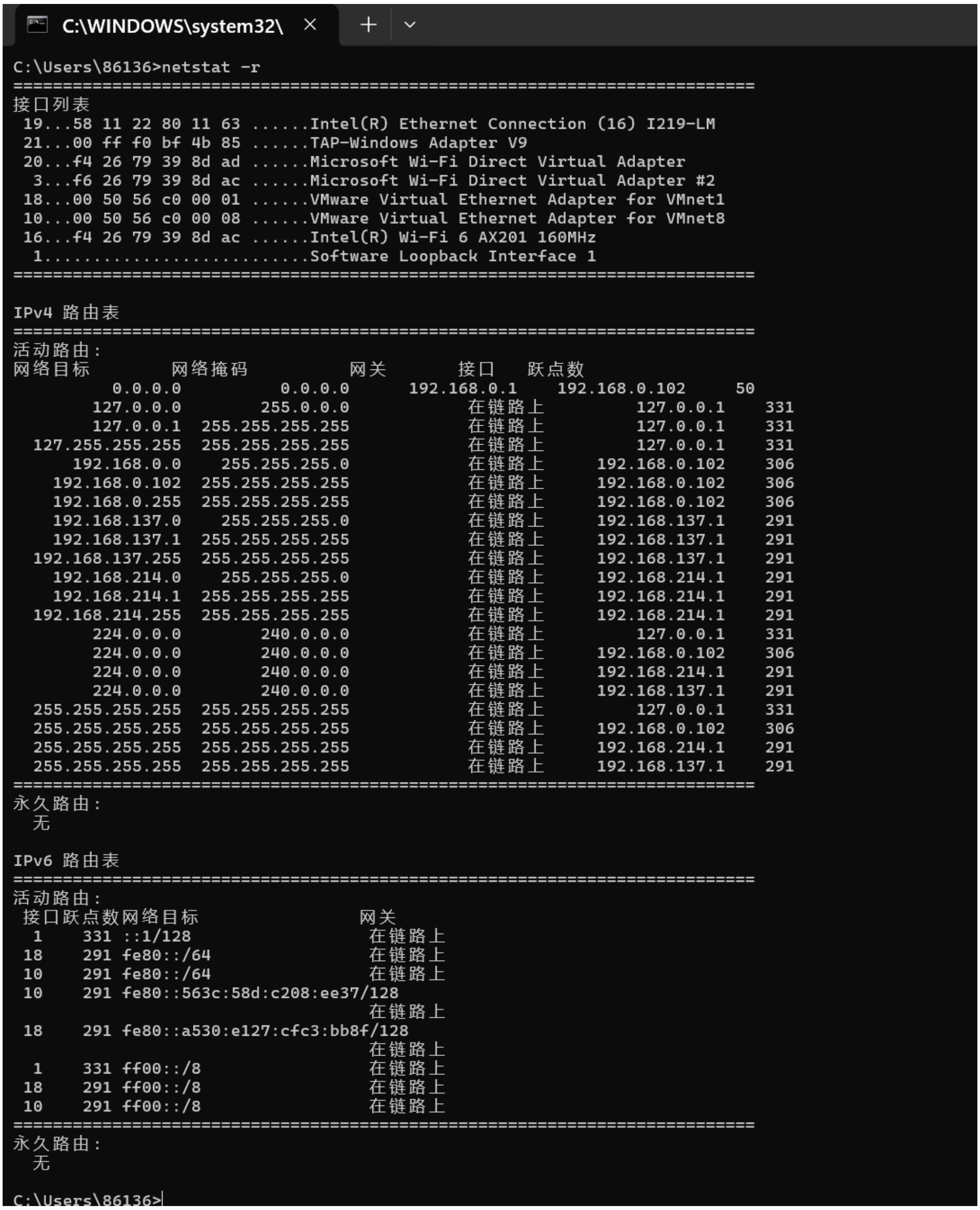
MAC address

IPv4 address

Default Gateway

Figure 9: Detailed network information

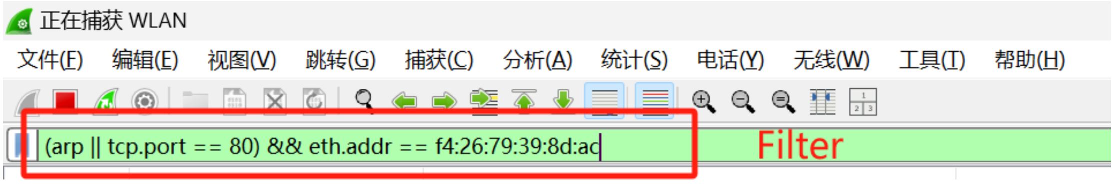
Use the “netstat -r” command to also get the default gateway IP information:



```
C:\Users\86136>netstat -r
===========================================================================
接口列表
 19...58 11 22 80 11 63 ......Intel(R) Ethernet Connection (16) I219-LM
 21...00 ff f0 bf 4b 85 ......TAP-Windows Adapter V9
 20...f4 26 79 39 8d ad ......Microsoft Wi-Fi Direct Virtual Adapter
  3...f6 26 79 39 8d ac ......Microsoft Wi-Fi Direct Virtual Adapter #2
 18...00 50 56 c0 00 01 ......VMware Virtual Ethernet Adapter for VMnet1
 10...00 50 56 c0 00 08 ......VMware Virtual Ethernet Adapter for VMnet8
 16...f4 26 79 39 8d ac ......Intel(R) Wi-Fi 6 AX201 160MHz
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 路由表
===========================================================================
活动路由:
网络目标        网络掩码          网关       接口   跃点数
          0.0.0.0          0.0.0.0      192.168.0.1    192.168.0.102     50
        127.0.0.0        255.0.0.0            在链路上         127.0.0.1    331
        127.0.0.1  255.255.255.255            在链路上         127.0.0.1    331
  127.255.255.255  255.255.255.255            在链路上         127.0.0.1    331
      192.168.0.0    255.255.255.0            在链路上     192.168.0.102    306
    192.168.0.102  255.255.255.255            在链路上     192.168.0.102    306
    192.168.0.255  255.255.255.255            在链路上     192.168.0.102    306
    192.168.137.0    255.255.255.0            在链路上     192.168.137.1    291
    192.168.137.1  255.255.255.255            在链路上     192.168.137.1    291
  192.168.137.255  255.255.255.255            在链路上     192.168.137.1    291
    192.168.214.0    255.255.255.0            在链路上     192.168.214.1    291
    192.168.214.1  255.255.255.255            在链路上     192.168.214.1    291
  192.168.214.255  255.255.255.255            在链路上     192.168.214.1    291
        224.0.0.0        240.0.0.0            在链路上         127.0.0.1    331
        224.0.0.0        240.0.0.0            在链路上     192.168.0.102    306
        224.0.0.0        240.0.0.0            在链路上     192.168.214.1    291
        224.0.0.0        240.0.0.0            在链路上     192.168.137.1    291
  255.255.255.255  255.255.255.255            在链路上         127.0.0.1    331
  255.255.255.255  255.255.255.255            在链路上     192.168.0.102    306
  255.255.255.255  255.255.255.255            在链路上     192.168.214.1    291
  255.255.255.255  255.255.255.255            在链路上     192.168.137.1    291
===========================================================================
永久路由:
  无

IPv6 路由表
===========================================================================
活动路由:
 接口跃点数网络目标                      网关
  1    331 ::1/128                  在链路上
 18    291 fe80::/64                在链路上
 10    291 fe80::/64                在链路上
 10    291 fe80::563c:58d:c208:ee37/128
                                    在链路上
 18    291 fe80::a530:e127:cfc3:bb8f/128
                                    在链路上
  1    331 ff00::/8                 在链路上
 18    291 ff00::/8                 在链路上
 10    291 ff00::/8                 在链路上
===========================================================================
永久路由:
  无

C:\Users\86136>
```

Figure 10: Using netstat -r command to get some information

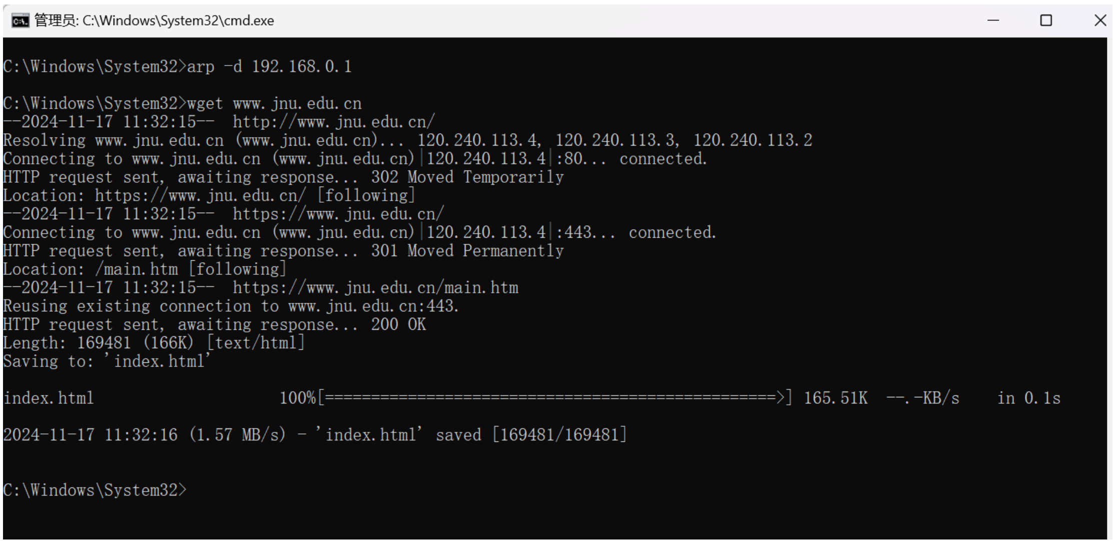
Next, start Wireshark, set the filter, and prepare for the ARP packet capture experiment:


正在捕获 WLAN
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)
Filter

(arp || tcp.port == 80) && eth.addr == f4:26:79:39:8d:ac

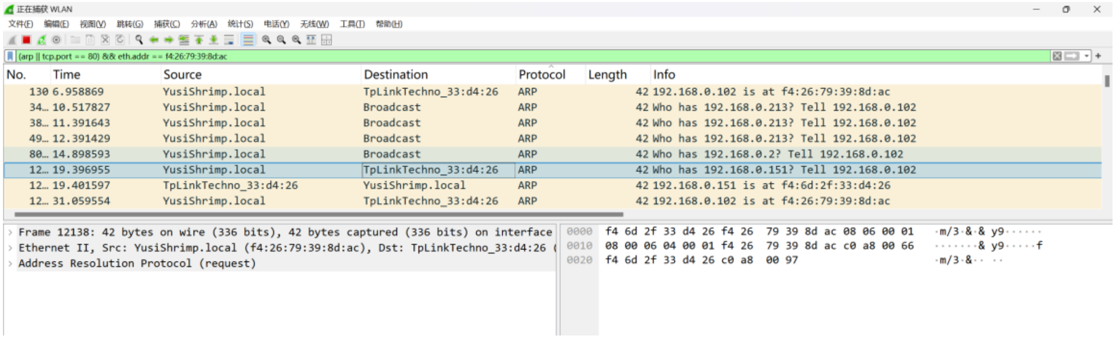
Figure 11: Set the Wireshark filter

Execute “arp -d default gateway IP address”, and then execute the “wget” command instead of using web browser to obtain the content of the “www.jnu.edu.cn” page:



```
C:\Windows\System32>arp -d 192.168.0.1

C:\Windows\System32>wget www. jnu. edu. cn
--2024-11-17 11:32:15--  http://www. jnu. edu. cn/
Resolving www. jnu. edu. cn (www. jnu. edu. cn)... 120.240.113.4, 120.240.113.3, 120.240.113.2
Connecting to www. jnu. edu. cn (www. jnu. edu. cn)|120.240.113.4|:80... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://www. jnu. edu. cn/ [following]
--2024-11-17 11:32:15--  https://www. jnu. edu. cn/
Connecting to www. jnu. edu. cn (www. jnu. edu. cn)|120.240.113.4|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: /main.htm [following]
--2024-11-17 11:32:15--  https://www. jnu. edu. cn/main.htm
Reusing existing connection to www. jnu. edu. cn:443.
HTTP request sent, awaiting response... 200 OK
Length: 169481 (166K) [text/html]
Saving to: 'index.html'

index.html                  100%[===============================================>] 165.51K  --.-KB/s    in 0.1s

2024-11-17 11:32:16 (1.57 MB/s) - 'index.html' saved [169481/169481]


C:\Windows\System32>
```

Figure 12: Clean ARP cache and fetch a remote webpage


正在捕获 WLAN
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)

```
(arp || tcp.port == 80) && eth.addr == f4:26:79:39:8d:ac
```

No.	Time	Source	Destination	Protocol	Length	Info
130	6.958869	YusiShrimp.local	TpLinkTechno_33:d4:26	ARP	42	192.168.0.102 is at f4:26:79:39:8d:ac
34…	10.517827	YusiShrimp.local	Broadcast	ARP	42	Who has 192.168.0.213? Tell 192.168.0.102
38…	11.391643	YusiShrimp.local	Broadcast	ARP	42	Who has 192.168.0.213? Tell 192.168.0.102
49…	12.391429	YusiShrimp.local	Broadcast	ARP	42	Who has 192.168.0.213? Tell 192.168.0.102
80…	14.898593	YusiShrimp.local	Broadcast	ARP	42	Who has 192.168.0.2? Tell 192.168.0.102
12…	19.396955	YusiShrimp.local	TpLinkTechno_33:d4:26	ARP	42	Who has 192.168.0.151? Tell 192.168.0.102
12…	19.401597	TpLinkTechno_33:d4:26	YusiShrimp.local	ARP	42	192.168.0.151 is at f4:6d:2f:33:d4:26
12…	31.059554	YusiShrimp.local	TpLinkTechno_33:d4:26	ARP	42	192.168.0.102 is at f4:26:79:39:8d:ac

```
> Frame 12138: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface
> Ethernet II, Src: YusiShrimp.local (f4:26:79:39:8d:ac), Dst: TpLinkTechno_33:d4:26 (
> Address Resolution Protocol (request)
```

```
0000  f4 6d 2f 33 d4 26 f4 26  79 39 8d ac 08 06 00 01   ·m/3·&·& y9······
0010  08 00 06 04 00 01 f4 26  79 39 8d ac c0 a8 00 66   ·······& y9·····f
0020  f4 6d 2f 33 d4 26 c0 a8  00 97                     ·m/3·&·· ··
```

Figure 13: Captured ARP protocol packages

2) Task 2: ARP for hosts within the same LAN

1. Experiment steps:

[1] **Step 1:** On both Host A and Host B, start Wireshark to capture both ARP and ICMP packets, and use arp -d to clear ARP cache.

[2] **Step 2:** Host A ping Host B.

[3] **Step 3:** Stop the capture in Wireshark, and inspect the captured ARP trace. There are two kinds of ARP packets, ARP request and reply (also called response). Inspect them in turn. (a) Find an ARP request for the default gateway and examine each field.

- The Info line for the request will start with “Who has …”.
- You can look for one of these packets that asks for the MAC address of the default gateway, e.g., “Who has xx.xx.xx.xx …” where xx.xx.xx.xx is your default gateway.
- You need to examine each fields of both Ethernet frame header and ARP packets.

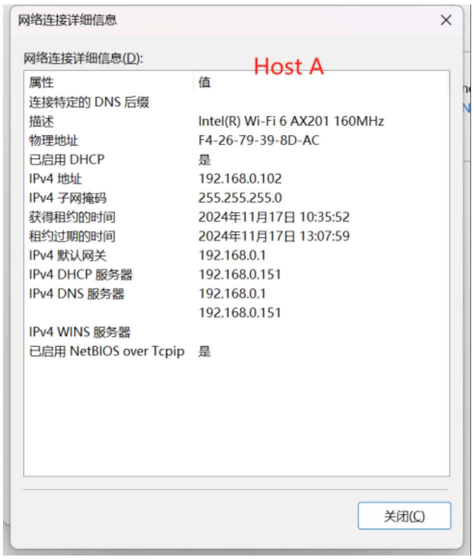
- Refer to the Reading material on course website for detailed format of ARP packets. (b) Select an ARP reply and examine its fields.
- The reply will answer a request and have an Info line of the form “xx.xx.xx.xx is at yy:yy:yy:yy:yy:yy”.
- You need to examine each fields of both Ethernet frame header and ARP packets.

[4] **Step 4:** Draw a figure that shows the ARP request/reply and ICMP reques t/response packets sent between Host A and Host B.

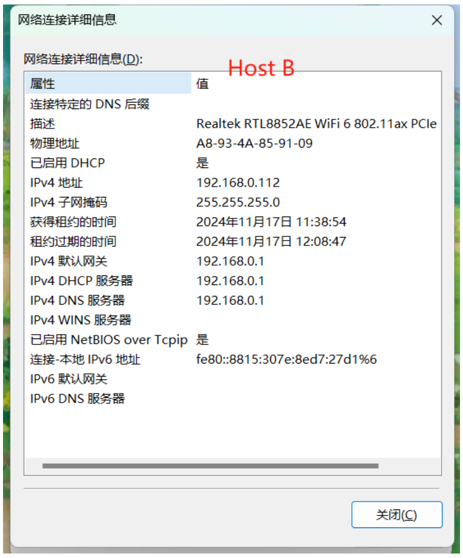
- Label one packet the request and the other the reply.
- Give the sender and target MAC & IP addresses for each ARP packet.
- Please indict the time sequence of these packets on the figure.

2. Results

The network information of the two Laptops under the same WLAN is as follows:

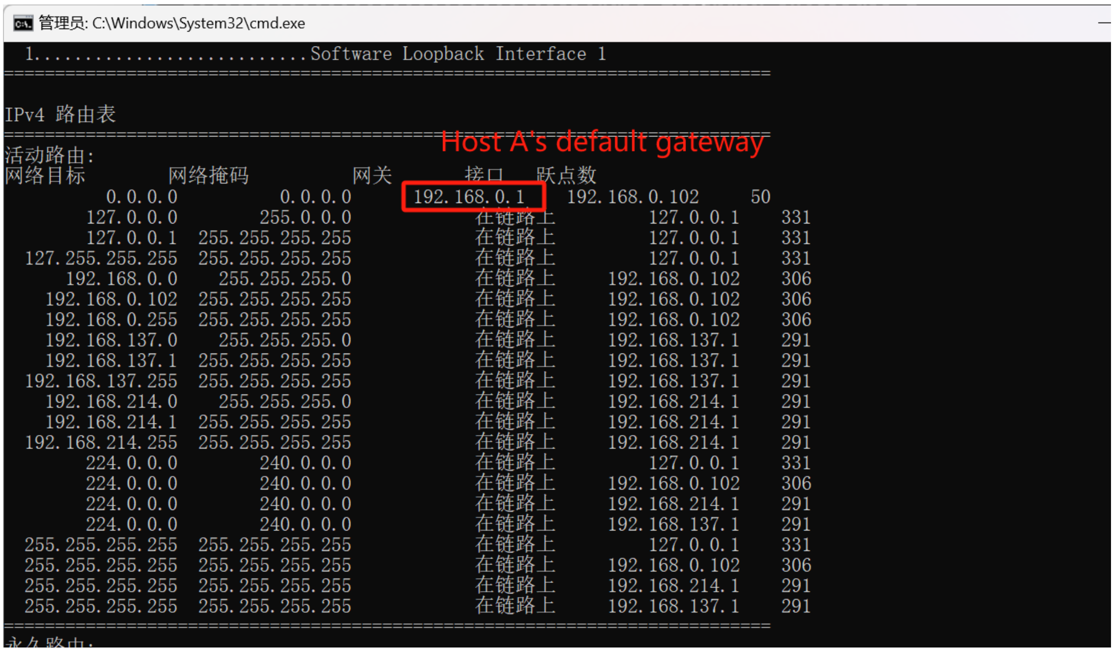


属性	值
连接特定的 DNS 后缀	
描述	Intel(R) Wi-Fi 6 AX201 160MHz
物理地址	F4-26-79-39-8D-AC
已启用 DHCP	是
IPv4 地址	192.168.0.102
IPv4 子网掩码	255.255.255.0
获得租约的时间	2024年11月17日 10:35:52
租约过期的时间	2024年11月17日 13:07:59
IPv4 默认网关	192.168.0.1
IPv4 DHCP 服务器	192.168.0.151
IPv4 DNS 服务器	192.168.0.1 192.168.0.151
IPv4 WINS 服务器	
已启用 NetBIOS over Tcpip	是



属性	值
连接特定的 DNS 后缀	
描述	Realtek RTL8852AE WiFi 6 802.11ax PCIe
物理地址	A8-93-4A-85-91-09
已启用 DHCP	是
IPv4 地址	192.168.0.112
IPv4 子网掩码	255.255.255.0
获得租约的时间	2024年11月17日 11:38:54
租约过期的时间	2024年11月17日 12:08:47
IPv4 默认网关	192.168.0.1
IPv4 DHCP 服务器	192.168.0.1
IPv4 DNS 服务器	192.168.0.1
IPv4 WINS 服务器	
已启用 NetBIOS over Tcpip	是
连接-本地 IPv6 地址	fe80::8815:307e:8ed7:27d1%6
IPv6 默认网关	
IPv6 DNS 服务器	

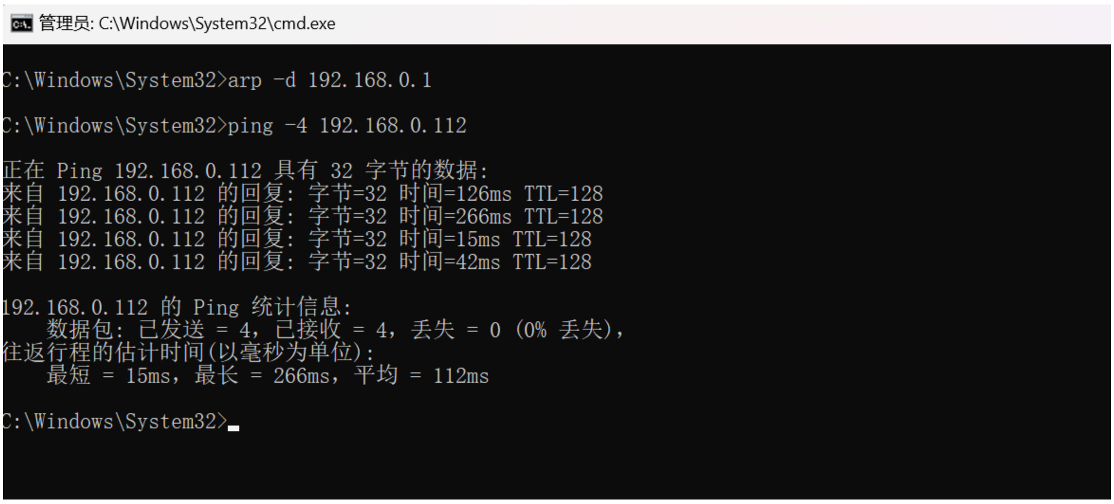
Figure 14: Network environment of Host A and B



Host A's default gateway

```
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 路由表
===========================================================================
活动路由:
网络目标        网络掩码          网关       接口   跃点数
          0.0.0.0          0.0.0.0      192.168.0.1    192.168.0.102     50
        127.0.0.0        255.0.0.0            在链路上         127.0.0.1    331
        127.0.0.1  255.255.255.255            在链路上         127.0.0.1    331
  127.255.255.255  255.255.255.255            在链路上         127.0.0.1    331
      192.168.0.0    255.255.255.0            在链路上     192.168.0.102    306
    192.168.0.102  255.255.255.255            在链路上     192.168.0.102    306
    192.168.0.255  255.255.255.255            在链路上     192.168.0.102    306
    192.168.137.0    255.255.255.0            在链路上     192.168.137.1    291
    192.168.137.1  255.255.255.255            在链路上     192.168.137.1    291
  192.168.137.255  255.255.255.255            在链路上     192.168.137.1    291
    192.168.214.0    255.255.255.0            在链路上     192.168.214.1    291
    192.168.214.1  255.255.255.255            在链路上     192.168.214.1    291
  192.168.214.255  255.255.255.255            在链路上     192.168.214.1    291
        224.0.0.0        240.0.0.0            在链路上         127.0.0.1    331
        224.0.0.0        240.0.0.0            在链路上     192.168.0.102    306
        224.0.0.0        240.0.0.0            在链路上     192.168.214.1    291
        224.0.0.0        240.0.0.0            在链路上     192.168.137.1    291
  255.255.255.255  255.255.255.255            在链路上         127.0.0.1    331
  255.255.255.255  255.255.255.255            在链路上     192.168.0.102    306
  255.255.255.255  255.255.255.255            在链路上     192.168.214.1    291
  255.255.255.255  255.255.255.255            在链路上     192.168.137.1    291
===========================================================================
永久路由:
```

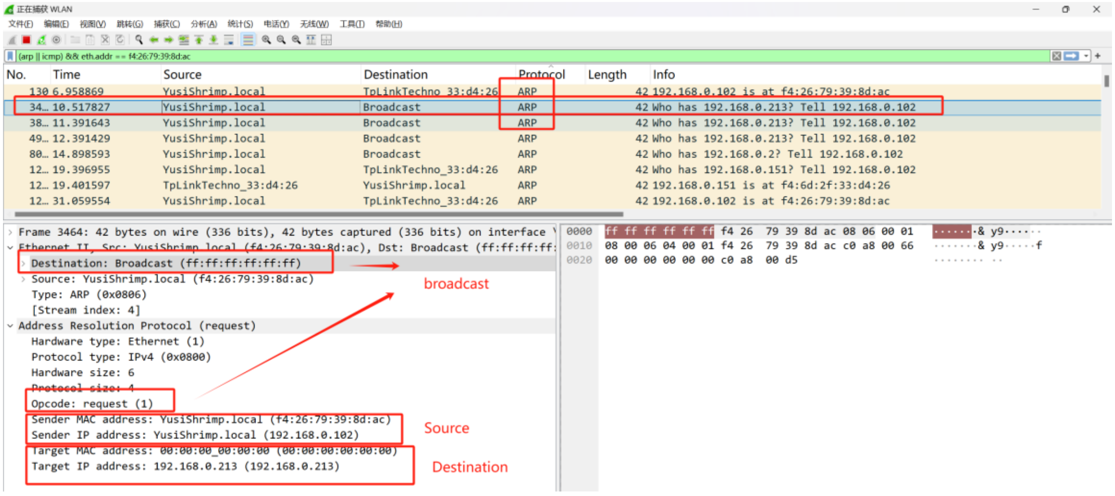
Figure 15: Host A's default gateway information



```
C:\Windows\System32>arp -d 192.168.0.1

C:\Windows\System32>ping -4 192.168.0.112

正在 Ping 192.168.0.112 具有 32 字节的数据:
来自 192.168.0.112 的回复: 字节=32 时间=126ms TTL=128
来自 192.168.0.112 的回复: 字节=32 时间=266ms TTL=128
来自 192.168.0.112 的回复: 字节=32 时间=15ms TTL=128
来自 192.168.0.112 的回复: 字节=32 时间=42ms TTL=128

192.168.0.112 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 = 15ms，最长 = 266ms，平均 = 112ms

C:\Windows\System32>_
```

Figure 16: Cleaning Host A's ARP cache and ping Host B

Next, I can observe the trace from the time of script execution through Wireshark started in Host A, which includes the ARP request issued by Host A and the ARP reply from Host B.


正在捕获 WLAN
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)

```
(arp || icmp) && eth.addr == f4:26:79:39:8d:ac

No.    Time        Source               Destination            Protocol   Length   Info
   130 6.958869    YusiShrimp.local     TpLinkTechno_33:d4:26  ARP        42 192.168.0.102 is at f4:26:79:39:8d:ac
   34… 10.517827   YusiShrimp.local     Broadcast              ARP        42 Who has 192.168.0.213? Tell 192.168.0.102
   38… 11.391643   YusiShrimp.local     Broadcast              ARP        42 Who has 192.168.0.213? Tell 192.168.0.102
   49… 12.391429   YusiShrimp.local     Broadcast              ARP        42 Who has 192.168.0.213? Tell 192.168.0.102
   80… 14.898593   YusiShrimp.local     Broadcast              ARP        42 Who has 192.168.0.2? Tell 192.168.0.102
   12… 19.396955   YusiShrimp.local     TpLinkTechno_33:d4:26  ARP        42 Who has 192.168.0.151? Tell 192.168.0.102
   12… 19.401597   TpLinkTechno_33:d4:26 YusiShrimp.local      ARP        42 192.168.0.151 is at f4:6d:2f:33:d4:26
   12… 31.059554   YusiShrimp.local     TpLinkTechno_33:d4:26  ARP        42 192.168.0.102 is at f4:26:79:39:8d:ac
```

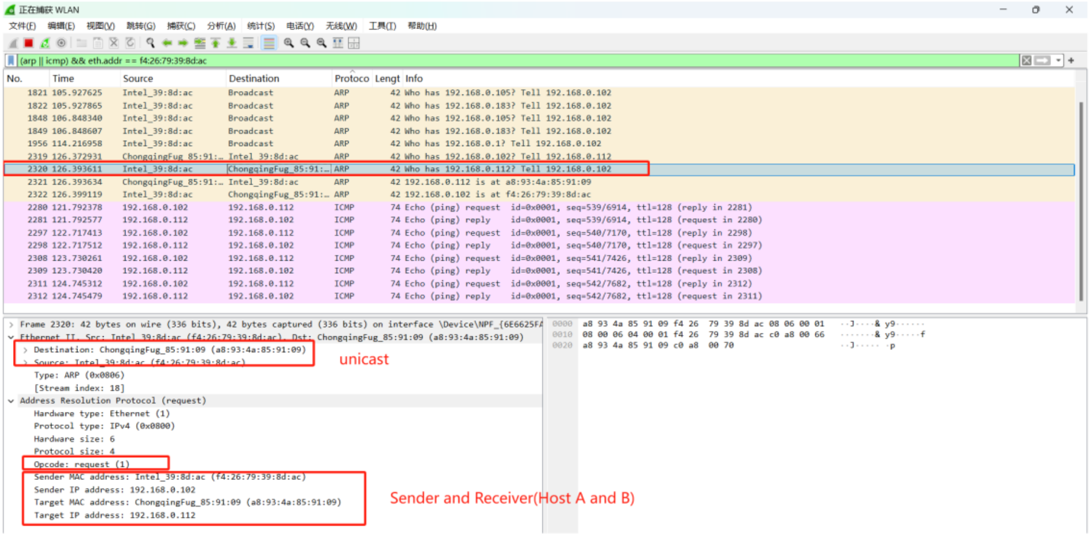
```
> Frame 3464: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \
v Ethernet II, Src: YusiShrimp.local (f4:26:79:39:8d:ac), Dst: Broadcast (ff:ff:ff:ff:
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: YusiShrimp.local (f4:26:79:39:8d:ac)
    Type: ARP (0x0806)
    [Stream index: 4]
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: YusiShrimp.local (f4:26:79:39:8d:ac)
    Sender IP address: YusiShrimp.local (192.168.0.102)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.0.213 (192.168.0.213)

0000  ff ff ff ff ff ff f4 26  79 39 8d ac 08 06 00 01   ······& y9······
0010  08 00 06 04 00 01 f4 26  79 39 8d ac c0 a8 00 66   ·······& y9·····f
0020  00 00 00 00 00 00 c0 a8  00 d5                     ········ ··
```

broadcast

Source

Destination

Figure 17: Captured ARP request from Host A


正在捕获 WLAN
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)

```
(arp || icmp) && eth.addr == f4:26:79:39:8d:ac

No.   Time        Source              Destination         Protoco Lengt Info
 1821 105.927625  Intel_39:8d:ac      Broadcast           ARP      42 Who has 192.168.0.105? Tell 192.168.0.102
 1822 105.927865  Intel_39:8d:ac      Broadcast           ARP      42 Who has 192.168.0.183? Tell 192.168.0.102
 1848 106.848340  Intel_39:8d:ac      Broadcast           ARP      42 Who has 192.168.0.105? Tell 192.168.0.102
 1849 106.848607  Intel_39:8d:ac      Broadcast           ARP      42 Who has 192.168.0.183? Tell 192.168.0.102
 1956 114.216958  Intel_39:8d:ac      Broadcast           ARP      42 Who has 192.168.0.1? Tell 192.168.0.102
 2319 126.372931  ChongqingFug_85:91:… Intel_39:8d:ac     ARP      42 Who has 192.168.0.102? Tell 192.168.0.112
 2320 126.393611  Intel_39:8d:ac      ChongqingFug_85:91:… ARP     42 Who has 192.168.0.112? Tell 192.168.0.102
 2321 126.393634  ChongqingFug_85:91:… Intel_39:8d:ac     ARP      42 192.168.0.112 is at a8:93:4a:85:91:09
 2322 126.399119  Intel_39:8d:ac      ChongqingFug_85:91:… ARP     42 192.168.0.102 is at f4:26:79:39:8d:ac
 2280 121.792378  192.168.0.102       192.168.0.112       ICMP     74 Echo (ping) request  id=0x0001, seq=539/6914, ttl=128 (reply in 2281)
 2281 121.792577  192.168.0.112       192.168.0.102       ICMP     74 Echo (ping) reply    id=0x0001, seq=539/6914, ttl=128 (request in 2280)
 2297 122.717413  192.168.0.102       192.168.0.112       ICMP     74 Echo (ping) request  id=0x0001, seq=540/7170, ttl=128 (reply in 2298)
 2298 122.717512  192.168.0.112       192.168.0.102       ICMP     74 Echo (ping) reply    id=0x0001, seq=540/7170, ttl=128 (request in 2297)
 2308 123.730261  192.168.0.102       192.168.0.112       ICMP     74 Echo (ping) request  id=0x0001, seq=541/7426, ttl=128 (reply in 2309)
 2309 123.730420  192.168.0.112       192.168.0.102       ICMP     74 Echo (ping) reply    id=0x0001, seq=541/7426, ttl=128 (request in 2308)
 2311 124.745312  192.168.0.102       192.168.0.112       ICMP     74 Echo (ping) request  id=0x0001, seq=542/7682, ttl=128 (reply in 2312)
 2312 124.745479  192.168.0.112       192.168.0.102       ICMP     74 Echo (ping) reply    id=0x0001, seq=542/7682, ttl=128 (request in 2311)
```

```
> Frame 2320: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{6E6625FA
v Ethernet II, Src: Intel_39:8d:ac (f4:26:79:39:8d:ac), Dst: ChongqingFug_85:91:09 (a8:93:4a:85:91:09)
  > Destination: ChongqingFug_85:91:09 (a8:93:4a:85:91:09)
  > Source: Intel_39:8d:ac (f4:26:79:39:8d:ac)
    Type: ARP (0x0806)
    [Stream index: 18]
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Intel_39:8d:ac (f4:26:79:39:8d:ac)
    Sender IP address: 192.168.0.102
    Target MAC address: ChongqingFug_85:91:09 (a8:93:4a:85:91:09)
    Target IP address: 192.168.0.112

0000  a8 93 4a 85 91 09 f4 26  79 39 8d ac 08 06 00 01   ··J····& y9······
0010  08 00 06 04 00 01 f4 26  79 39 8d ac c0 a8 00 66   ·······& y9·····f
0020  a8 93 4a 85 91 09 c0 a8  00 70                     ··J····· ·p
```

unicast

Sender and Receiver(Host A and B)

Figure 18: Captured ARP reply from Host B

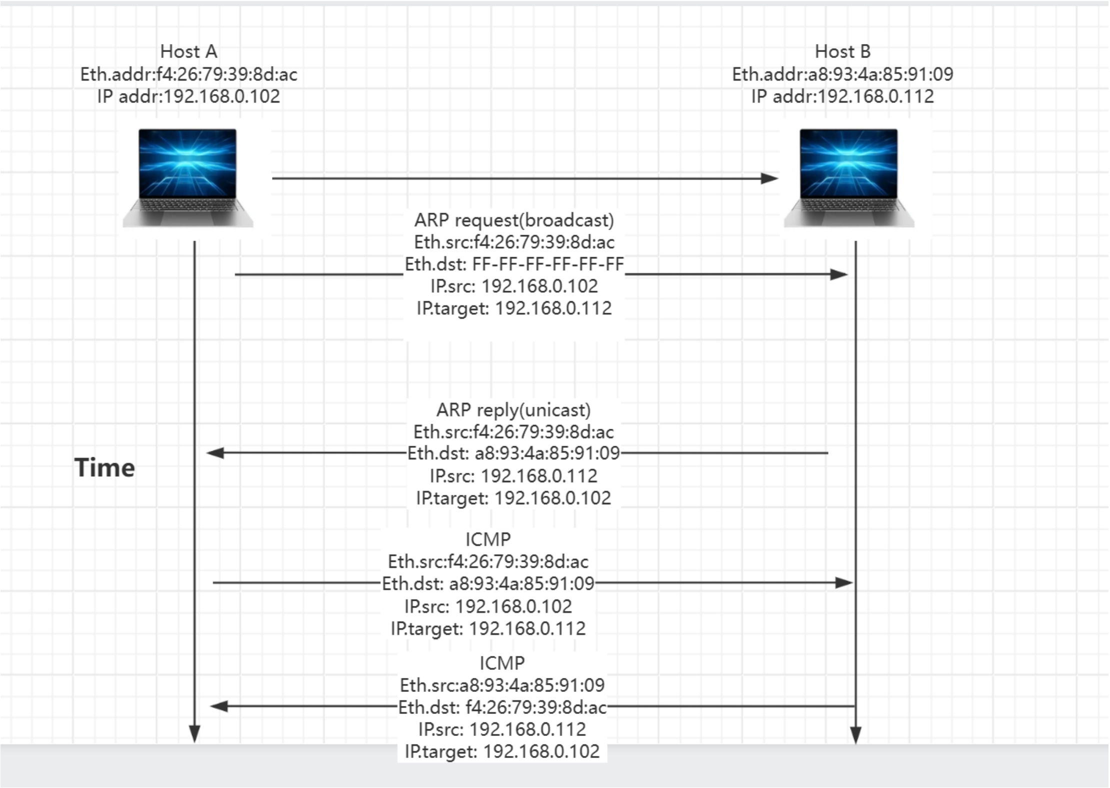


Figure 19: ARP request/reply and ICMP request/response packets sent between Host A and Host B

3. Answer the questions

Q1: What columns are included in the high-speed ARP cache table?

A1: The columns found in an ARP cache table include: the network interface through which the IP-MAC mapping was discovered or is valid. The IP address associated with a network device and MAC address corresponding to the IP address. And whether the entry is dynamic (learned through ARP requests) or static(manually configured).

Q2: What opcode is used to indicate a request? What about a reply?

A2: In the ARP, the opcode field is used to indicate the purpose of the ARP message. The opcode value for a request is 1, indicating an ARP request. The opcode value for a reply is 2, indicating an ARP reply.

Q3: How large is the ARP header for a request? What about for a reply?

A3: The size of the ARP header for a request and a reply is generally 28 bytes for IPv4.

Q4: What value is carried on a request for the unknown target MAC address?

A4: The target MAC address of the request is normally all zeros, or 00:00:00:00:00:00.

Q5: What Ethernet Type value indicates that ARP is the higher layer protocol?

A5: The Ethernet Type value for ARP is 0x806.

Q6: Is the ARP reply broadcast (like the ARP request) or not?

A6: The ARP reply is normally not broadcast. It is sent directly to the target using its Ethernet address.

3) Task 3: ARP for a remote server

1. Experiment steps:

Setup:

- A local computer within a LAN and a remote server on the Internet, e.g., www.baidu.com.
- Or, two computers in the JNU campus LAN but connected to different default gateway, e.g., one com puter in the lab room and the other one connects to the campus WiFi.

[1] Step 1: On the local computer, start Wireshark to capture both ARP and ICMP packets.

[2] Step 2: Execute the following commands in a batch file (Windows) or a shell script (Linux and MacOS):

arp -d ping www.baidu.com

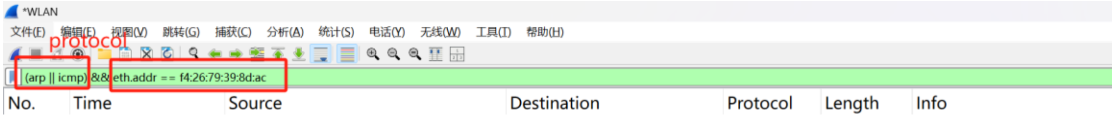
- Consider why we prefer to use a batch file. Are there any risks if not using batch processing?

[3] Step 3: Stop the capture in Wireshark, and inspect the captured ARP as in Task

[4] Step 4: Like Task 2, draw a figure that shows process of the ARP request/reply and ICMP probe/response packets, which are exchanged among the computer, the default gateway and the remote server.

2. Results

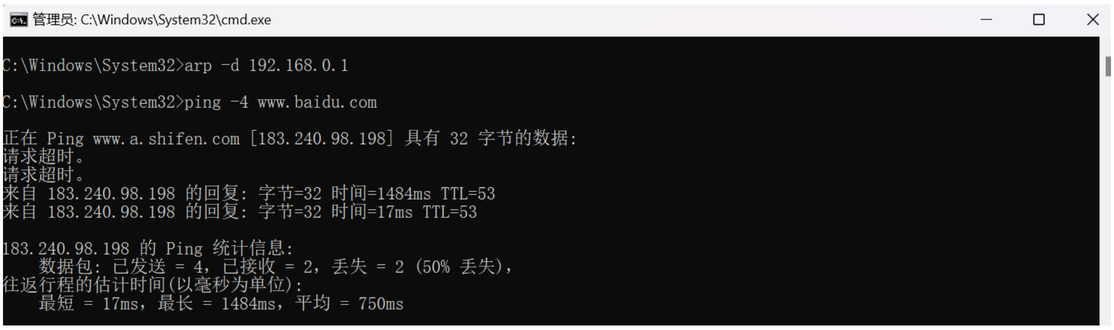
First start Wireshark and set the filter to “(arp || icmp) && eth.addr == local Ethernet address”:


*WLAN
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)
protocol
No.
Time
Source
Destination
Protocol
Length
Info

(arp || icmp) && eth.addr == f4:26:79:39:8d:ac

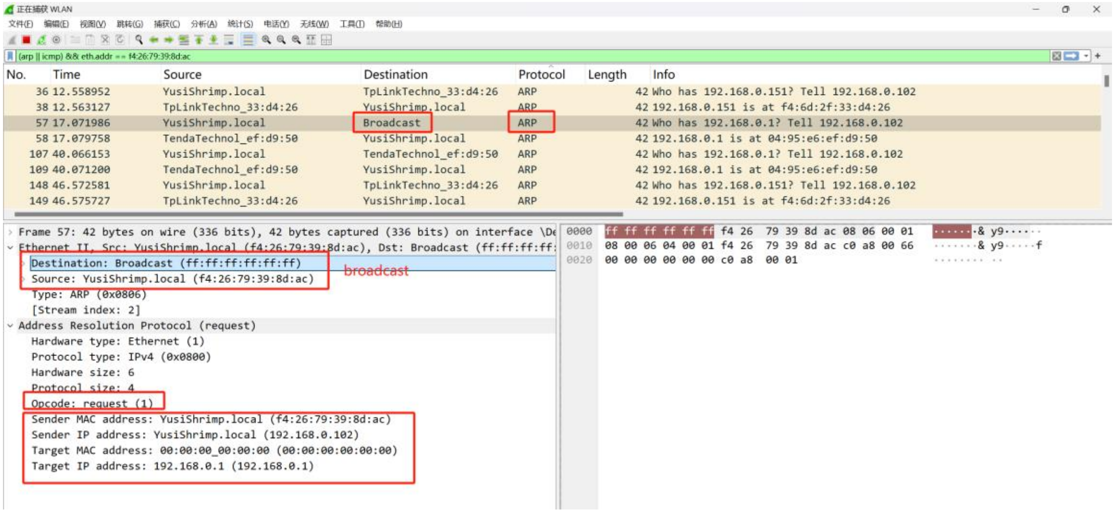
Figure 20: Set a filter

Use a simple script to implement the two commands “arp -d default gateway IP address” and “ping www.baidu.com” to be executed continuously. Finally, I find in Wireshark that the computer sent an ARP request to the default gateway. After obtaining the MAC address of the default gateway, computer can make an ICMP request to the remote server:



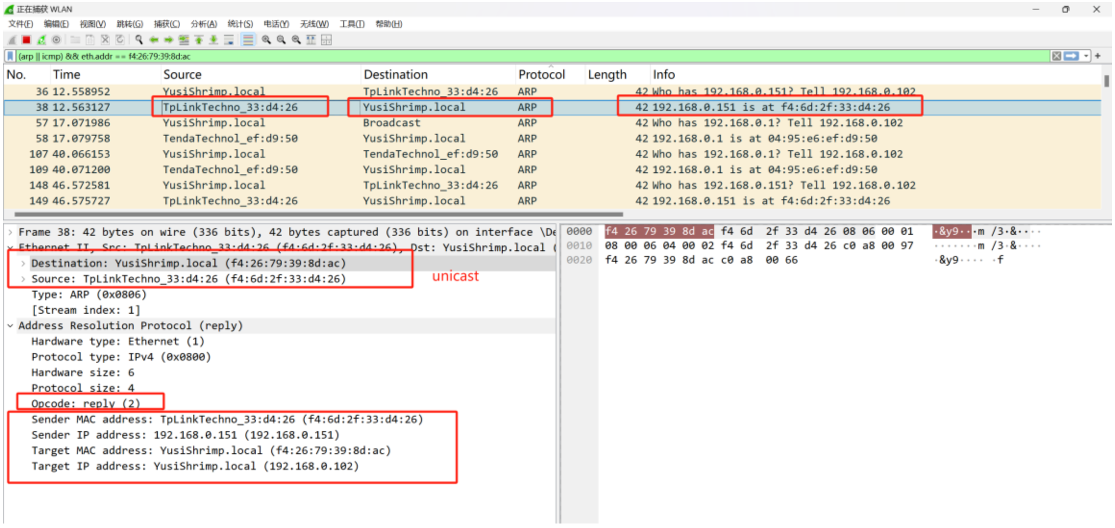
```
C:\Windows\System32>arp -d 192.168.0.1

C:\Windows\System32>ping -4 www.baidu.com

正在 Ping www.a.shifen.com [183.240.98.198] 具有 32 字节的数据:
请求超时。
请求超时。
来自 183.240.98.198 的回复: 字节=32 时间=1484ms TTL=53
来自 183.240.98.198 的回复: 字节=32 时间=17ms TTL=53

183.240.98.198 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 2, 丢失 = 2 (50% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 17ms, 最长 = 1484ms, 平均 = 750ms
```

Figure 21: perform the command


正在捕获 WLAN
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)
broadcast

```
(arp || icmp) && eth.addr == f4:26:79:39:8d:ac

No.  Time       Source                Destination           Protocol Length Info
  36 12.558952  YusiShrimp.local      TpLinkTechno_33:d4:26 ARP      42 Who has 192.168.0.151? Tell 192.168.0.102
  38 12.563127  TpLinkTechno_33:d4:26 YusiShrimp.local      ARP      42 192.168.0.151 is at f4:6d:2f:33:d4:26
  57 17.071986  YusiShrimp.local      Broadcast             ARP      42 Who has 192.168.0.1? Tell 192.168.0.102
  58 17.079758  TendaTechnol_ef:d9:50 YusiShrimp.local      ARP      42 192.168.0.1 is at 04:95:e6:ef:d9:50
 107 40.066153  YusiShrimp.local      TendaTechnol_ef:d9:50 ARP      42 Who has 192.168.0.1? Tell 192.168.0.102
 109 40.071200  TendaTechnol_ef:d9:50 YusiShrimp.local      ARP      42 192.168.0.1 is at 04:95:e6:ef:d9:50
 148 46.572581  YusiShrimp.local      TpLinkTechno_33:d4:26 ARP      42 Who has 192.168.0.151? Tell 192.168.0.102
 149 46.575727  TpLinkTechno_33:d4:26 YusiShrimp.local      ARP      42 192.168.0.151 is at f4:6d:2f:33:d4:26

> Frame 57: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \De
v Ethernet II, Src: YusiShrimp.local (f4:26:79:39:8d:ac), Dst: Broadcast (ff:ff:ff:ff:
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: YusiShrimp.local (f4:26:79:39:8d:ac)
    Type: ARP (0x0806)
    [Stream index: 2]
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: YusiShrimp.local (f4:26:79:39:8d:ac)
    Sender IP address: YusiShrimp.local (192.168.0.102)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.0.1 (192.168.0.1)

0000  ff ff ff ff ff ff f4 26  79 39 8d ac 08 06 00 01   ······&  y9······
0010  08 00 06 04 00 01 f4 26  79 39 8d ac c0 a8 00 66   ·······&  y9·····f
0020  00 00 00 00 00 00 c0 a8  00 01                     ········  ··
```

Figure 22: Capture ARP request


正在捕获 WLAN
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)
unicast

```
(arp || icmp) && eth.addr == f4:26:79:39:8d:ac

No.  Time       Source                Destination           Protocol Length Info
  36 12.558952  YusiShrimp.local      TpLinkTechno_33:d4:26 ARP      42 Who has 192.168.0.151? Tell 192.168.0.102
  38 12.563127  TpLinkTechno_33:d4:26 YusiShrimp.local      ARP      42 192.168.0.151 is at f4:6d:2f:33:d4:26
  57 17.071986  YusiShrimp.local      Broadcast             ARP      42 Who has 192.168.0.1? Tell 192.168.0.102
  58 17.079758  TendaTechnol_ef:d9:50 YusiShrimp.local      ARP      42 192.168.0.1 is at 04:95:e6:ef:d9:50
 107 40.066153  YusiShrimp.local      TendaTechnol_ef:d9:50 ARP      42 Who has 192.168.0.1? Tell 192.168.0.102
 109 40.071200  TendaTechnol_ef:d9:50 YusiShrimp.local      ARP      42 192.168.0.1 is at 04:95:e6:ef:d9:50
 148 46.572581  YusiShrimp.local      TpLinkTechno_33:d4:26 ARP      42 Who has 192.168.0.151? Tell 192.168.0.102
 149 46.575727  TpLinkTechno_33:d4:26 YusiShrimp.local      ARP      42 192.168.0.151 is at f4:6d:2f:33:d4:26

> Frame 38: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \De
v Ethernet II, Src: TpLinkTechno_33:d4:26 (f4:6d:2f:33:d4:26), Dst: YusiShrimp.local (
  > Destination: YusiShrimp.local (f4:26:79:39:8d:ac)
  > Source: TpLinkTechno_33:d4:26 (f4:6d:2f:33:d4:26)
    Type: ARP (0x0806)
    [Stream index: 1]
v Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: TpLinkTechno_33:d4:26 (f4:6d:2f:33:d4:26)
    Sender IP address: 192.168.0.151 (192.168.0.151)
    Target MAC address: YusiShrimp.local (f4:26:79:39:8d:ac)
    Target IP address: YusiShrimp.local (192.168.0.102)

0000  f4 26 79 39 8d ac f4 6d  2f 33 d4 26 08 06 00 01   ·&y9···m /3·&····
0010  08 00 06 04 00 02 f4 6d  2f 33 d4 26 c0 a8 00 97   ·······m /3·&····
0020  f4 26 79 39 8d ac c0 a8  00 66                     ·&y9····  ·f
```

Figure 22: Capture ARP reply

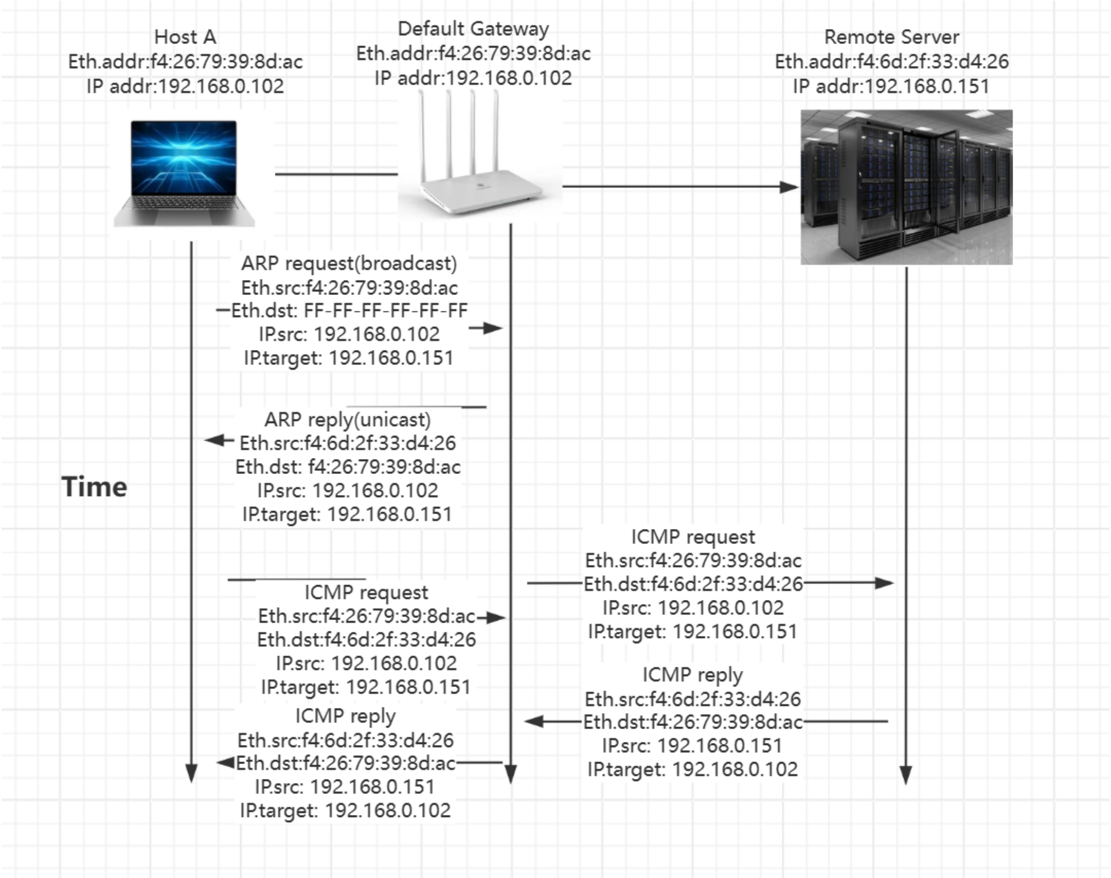


Figure 23: Process of the ARP request/reply and ICMP probe/response packets

4) Task 4: Explore on Your Own

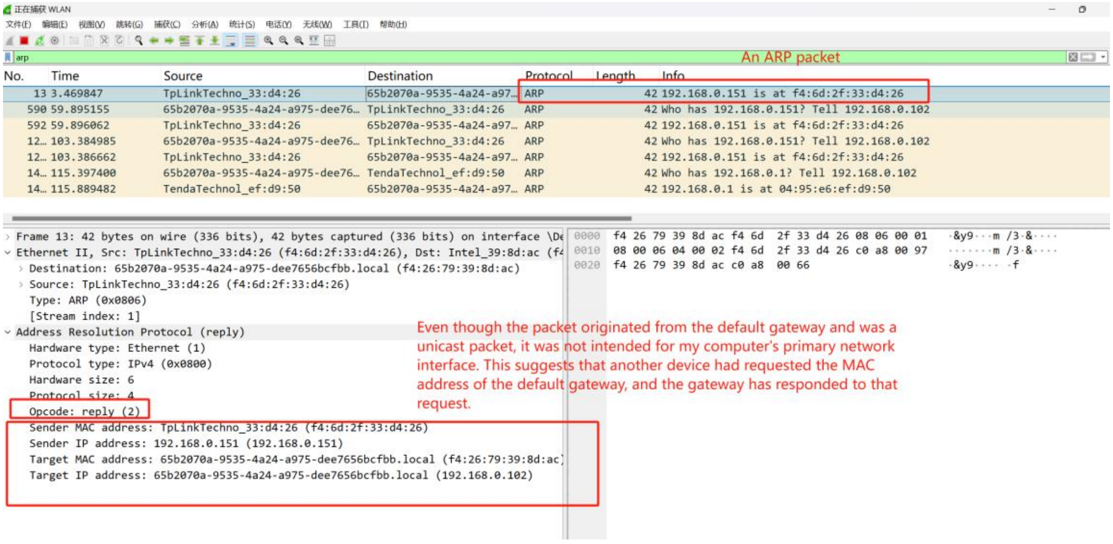
1. Experiment steps:

[1] Step 1: Start Wireshark, use the filter “arp”, and check these ARP protocol packets.

[2] Step 2: Use the “trace-arp.pcap” file provided in this experiment to check the ARP protocol packets in it.

2. Results

Using the “arp” filter, I captured other ARP packets:


正在捕获 WLAN
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)
arp

An ARP packet

```
No.   Time        Source                         Destination                  Protocol  Length  Info
   13 3.469847    TpLinkTechno_33:d4:26          65b2070a-9535-4a24-a97…      ARP       42 192.168.0.151 is at f4:6d:2f:33:d4:26
  590 59.895155   65b2070a-9535-4a24-a975-dee76… TpLinkTechno_33:d4:26        ARP       42 Who has 192.168.0.151? Tell 192.168.0.102
  592 59.896062   TpLinkTechno_33:d4:26          65b2070a-9535-4a24-a97…      ARP       42 192.168.0.151 is at f4:6d:2f:33:d4:26
  12… 103.384985  65b2070a-9535-4a24-a975-dee76… TpLinkTechno_33:d4:26        ARP       42 Who has 192.168.0.151? Tell 192.168.0.102
  12… 103.386662  TpLinkTechno_33:d4:26          65b2070a-9535-4a24-a97…      ARP       42 192.168.0.151 is at f4:6d:2f:33:d4:26
  14… 115.397400  65b2070a-9535-4a24-a975-dee76… TendaTechnol_ef:d9:50        ARP       42 Who has 192.168.0.1? Tell 192.168.0.102
  14… 115.889482  TendaTechnol_ef:d9:50          65b2070a-9535-4a24-a97…      ARP       42 192.168.0.1 is at 04:95:e6:ef:d9:50
```

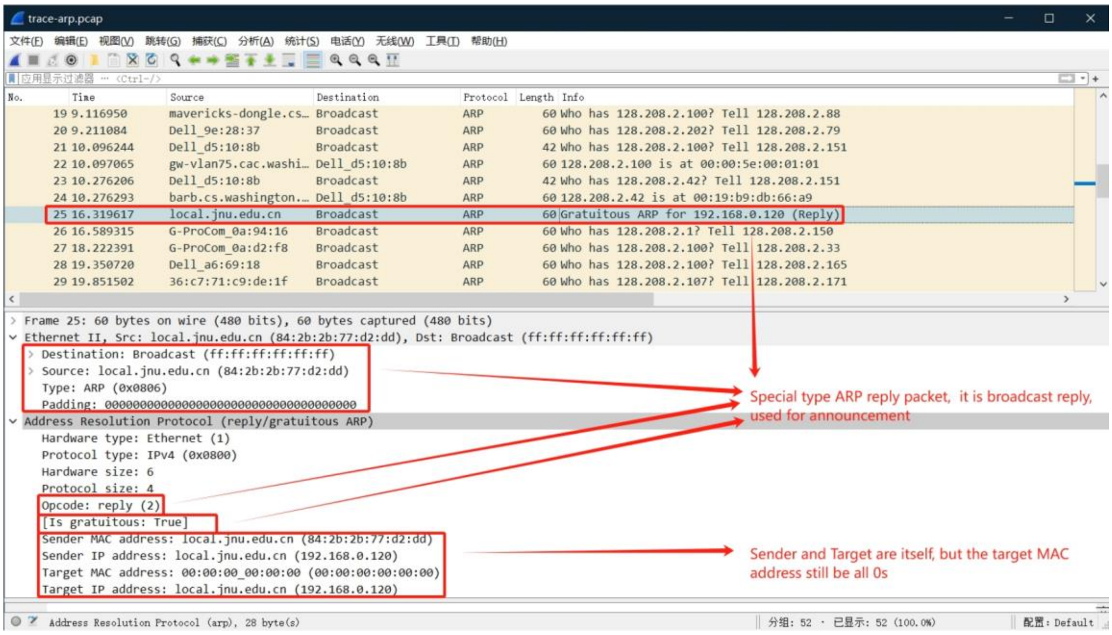
```
> Frame 13: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \De
v Ethernet II, Src: TpLinkTechno_33:d4:26 (f4:6d:2f:33:d4:26), Dst: Intel_39:8d:ac (f4
  > Destination: 65b2070a-9535-4a24-a975-dee7656bcfbb.local (f4:26:79:39:8d:ac)
  > Source: TpLinkTechno_33:d4:26 (f4:6d:2f:33:d4:26)
    Type: ARP (0x0806)
    [Stream index: 1]
v Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: TpLinkTechno_33:d4:26 (f4:6d:2f:33:d4:26)
    Sender IP address: 192.168.0.151 (192.168.0.151)
    Target MAC address: 65b2070a-9535-4a24-a975-dee7656bcfbb.local (f4:26:79:39:8d:ac)
    Target IP address: 65b2070a-9535-4a24-a975-dee7656bcfbb.local (192.168.0.102)
```

```
0000  f4 26 79 39 8d ac f4 6d  2f 33 d4 26 08 06 00 01   ·&y9···m /3·&····
0010  08 00 06 04 00 02 f4 6d  2f 33 d4 26 c0 a8 00 97   ·······m /3·&····
0020  f4 26 79 39 8d ac c0 a8  00 66                     ·&y9···· ·f
```

Even though the packet originated from the default gateway and was a unicast packet, it was not intended for my computer's primary network interface. This suggests that another device had requested the MAC address of the default gateway, and the gateway has responded to that request.

Figure 24: Other ARP reply packet

Observing the “trace-arp.pcap” file provided in this experiment, I found there are some gratuitous ARP packages in it:


trace-arp.pcap
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)
应用显示过滤器 … <Ctrl-/>

```
No.   Time       Source                 Destination   Protocol  Length  Info
   19 9.116950   mavericks-dongle.cs…   Broadcast     ARP       60 Who has 128.208.2.100? Tell 128.208.2.88
   20 9.211084   Dell_9e:28:37          Broadcast     ARP       60 Who has 128.208.2.202? Tell 128.208.2.79
   21 10.096244  Dell_d5:10:8b          Broadcast     ARP       42 Who has 128.208.2.100? Tell 128.208.2.151
   22 10.097065  gw-vlan75.cac.washi…   Dell_d5:10:8b ARP       60 128.208.2.100 is at 00:00:5e:00:01:01
   23 10.276206  Dell_d5:10:8b          Broadcast     ARP       42 Who has 128.208.2.42? Tell 128.208.2.151
   24 10.276293  barb.cs.washington.…   Dell_d5:10:8b ARP       60 128.208.2.42 is at 00:19:b9:db:66:a9
   25 16.319617  local.jnu.edu.cn       Broadcast     ARP       60 Gratuitous ARP for 192.168.0.120 (Reply)
   26 16.589315  G-ProCom_0a:94:16      Broadcast     ARP       60 Who has 128.208.2.1? Tell 128.208.2.150
   27 18.222391  G-ProCom_0a:d2:f8      Broadcast     ARP       60 Who has 128.208.2.100? Tell 128.208.2.33
   28 19.350720  Dell_a6:69:18          Broadcast     ARP       60 Who has 128.208.2.100? Tell 128.208.2.165
   29 19.851502  36:c7:71:c9:de:1f      Broadcast     ARP       60 Who has 128.208.2.107? Tell 128.208.2.171
```

```
> Frame 25: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
v Ethernet II, Src: local.jnu.edu.cn (84:2b:2b:77:d2:dd), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: local.jnu.edu.cn (84:2b:2b:77:d2:dd)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
v Address Resolution Protocol (reply/gratuitous ARP)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    [Is gratuitous: True]
    Sender MAC address: local.jnu.edu.cn (84:2b:2b:77:d2:dd)
    Sender IP address: local.jnu.edu.cn (192.168.0.120)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: local.jnu.edu.cn (192.168.0.120)
```

Special type ARP reply packet, it is broadcast reply, used for announcement

Sender and Target are itself, but the target MAC address still be all 0s

```
Address Resolution Protocol (arp), 28 byte(s)        分组: 52 · 已显示: 52 (100.0%)        配置: Default
```

Figure 25: Gratuitous ARP packet

4. Others

(1) Answer the questions

Q1: The command “arp -s InetAddr EtherAddr” allows you to manually add an entry to the ARP cache that resolves the IP address InetAddr to the physical address EtherAddr. What would happen if, when you manually added an entry, you entered the correct IP address, but the wrong Ethernet address for that remote interface? Tips: Check ARP Poison/Spoofing on the course website or Google.

A1: This situation could lead to a mismatch between the IP and MAC addresses stored in the ARP cache. When the system attempts to transmit a packet to the IP address you've added to the ARP cache, it would mistakenly use the incorrect Ethernet address linked to that IP address. Consequently, the packet would be directed to the wrong MAC address, which might be assigned to a different device or might not correspond to any existing device on the network.

If the erroneous MAC address is not associated with a valid device, there will be no response to the incoming packets. This scenario can result in communication failures or connectivity problems with the intended remote device.

However, the manually added ARP cache entry will remain until it either expires or is manually deleted. This persistence means that any subsequent attempts to communicate with the correct MAC address for the given IP address will continue to use the incorrect MAC address from the erroneous cache entry.

It also poses a potential risk for ARP spoofing, a technique employed by malicious actors to tamper with ARP in a network. ARP spoofing involves sending deceptive ARP messages to associate an attacker's MAC address with the IP address of a legitimate device on the network.

To resolve this issue, one must manually correct the ARP cache entry by updating the Ethernet address (EtherAddr) with the accurate MAC address for the remote interface. Alternatively, one can remove the incorrect entry from the ARP cache, allowing the system to automatically relearn the correct MAC address through the ARP protocol.

Q2: What is the default amount of time that an entry remains in your ARP cache before being removed?

A2: The duration an entry stays in the ARP cache before it is automatically purged can differ based on the specific operating system or network equipment in use. For Windows OS, the standard ARP cache timeout is set at 2 minutes, while macOS typically has a default ARP cache timeout of 20 minutes. These preset values are subject to change and can be adjusted according to the system's configuration or network-specific requirements. Moreover, various network devices or routers may operate with their own default ARP cache timeout settings.

Q3: Continue to Question 2, there is a timer for each entry in ARP cache. What happened if the timer is set too long or too short?

A3: The duration for which each entry is retained in the ARP cache is crucial for maintaining the accuracy and timely expiration of ARP mappings. Adjusting the timer settings can have significant effects:

- If the timer is too long:
    - ARP cache entries might linger beyond their relevance, even after the associated IP-MAC mappings have changed or are no longer valid. This can lead to the persistence of obsolete entries, which might trigger connectivity problems or communication delays.
    - Network devices could also be slow to acknowledge updated mappings

if the timer is overly extended, causing delays in network updates and potentially hindering regular network functions.

- If the timer is too short:
    - ARP cache entries might expire too rapidly, necessitating constant ARP requests to re-establish IP-MAC mappings. This can escalate network traffic and additional processing demands, affecting network performance and possibly leading to latency issues.

Therefore, striking the right balance with timer settings is essential for optimizing network efficiency and reliability.

Q4: For ARP protocol, what’s the difference between finding the MAC address of a host within the same subnet and a host within another subnet connected by router?

A4: To locate the MAC address of a device on the same subnet, a system first searches its ARP cache to see if it already has the MAC address linked to the destination IP address. If no such entry exists in the cache, the system initiates an ARP request, which is broadcasted across the local network, inquiring about the MAC address that corresponds to the destination IP address. Upon receiving this request, a device with the matching IP address within the subnet replies with an ARP response, which is a unicast message, containing its MAC address. The requesting device subsequently updates its ARP cache with this newly acquired MAC address and then continues the communication process directly using the MAC address for data transmission.

When a device needs to find the MAC address for a destination that lies in a different subnet, it first recognizes that the target IP address is not within its own subnet. The device then refers to its routing table to find the next-hop router that provides a path to the destination subnet. Alternatively, it may broadcast an ARP request on its local network to discover the MAC address associated with the IP address of the next-hop router. This ARP request is directed to the MAC address of the router's interface connected to the same local network.

Upon receiving the ARP request, the router, which serves as a gateway, responds with an ARP reply that is a unicast message, including its own MAC address. The requesting device subsequently updates its ARP cache with the MAC address of the router's interface. Armed with the router's interface MAC address, the device then encapsulates the IP packet within an Ethernet frame, using the router's MAC address as the destination address. The frame is transmitted onto the local network, and from there, the packet is routed onwards to reach the destination subnet through subsequent hops.

Q5: Is the length of ARP packet fixed? Explain why.

A5: Affirmative. The ARP packet adheres to a standardized format with a set length, which is generally 28 bytes.

The reasons for this fixed length in ARP packets are as follows:

1. **Efficient Processing**: A fixed length allows network devices to process ARP

packets more efficiently. Since the structure is consistent, devices can parse and handle ARP messages without additional checks or calculations for packet length.

2. **Compatibility and Interoperability**: The fixed length ensures that different network devices and operating systems can work together seamlessly. Adherence to the standard ARP packet structure is crucial for accurate interpretation and processing of ARP messages across the network.
3. **Simplified Implementation**: A fixed length makes it easier to implement ARP protocol handlers in network devices and operating systems. Developers can rely on a uniform packet length, which simplifies the efficient handling and processing of ARP packets.

Q6: Suppose a computer needs to send an IP packet. List at least two cases that the computer does not need to send ARP request.

A6: **Disseminating to Network Devices:**

There are instances where a computer is required to dispatch an IP packet to a unique destination address known as the broadcast address, which signifies all devices on the local network. In such cases, the computer can encapsulate the IP packet into an Ethernet frame designated for a broadcast MAC address (e.g., FF:FF:FF:FF:FF:FF) without the necessity of resolving the MAC address via an ARP request. This method ensures that the packet reaches every device on the network.

Intra-Network Interaction:

When the destination IP address of an IP packet is within the same local network or subnet as the source, the computer might already possess the MAC address of the destination device in its ARP cache. The ARP cache stores previously resolved mappings from IP to MAC addresses for a specified duration. Should the destination IP address be found in the ARP cache, the computer can directly encapsulate the IP packet into an Ethernet frame utilizing the existing MAC address, bypassing the need to initiate an ARP request.

(2) Some thoughts

This laboratory exercise is structured to carry out fundamental tasks related to the ARP (Address Resolution Protocol) protocol. It involves activities such as verifying the default gateway, displaying the ARP cache table, and clearing the ARP cache table. Concurrently, the lab utilizes Wireshark to capture ARP protocol data packets across various networks, focusing on both request and reply to packets. Through these activities, I have gained a deeper comprehension of the ARP protocol, expanded my understanding of computer networking, and acquired valuable knowledge.