四川大學

《计算机网络》实验报告（1）

Wireshark 初探

专　　业　软件工程

姓　　名　郭　政

学　　号　2023141461076

指导老师　程艳红

成绩分数

二零二五 年 四 月 三 日

计算机网络 Wireshark 初探

一、实验目的

1. 初步了解抓包过程以及抓包数据的分析过程
2. 理解网络体系结构分层通信过程

二、实验环境

硬件要求：支持 64 位操作系统，基于 x64 架构的处理器

操作系统：Windows 11（64 位）

软件要求：网络分析工具：Wireshark

三、实验要求

实验 1：分析数据包的封装，分析首部与协议层次的对应关系

实验 2：执行 tracert gaia.cs.umass.edu 根据观测到的结果进行分析

实验 3：使用 ipconfig/all 命令查询网卡的 MAC 地址，根据 MAC 地址的 OUI 查询生产厂家

四、实验过程

实验 1：Wireshark 抓包

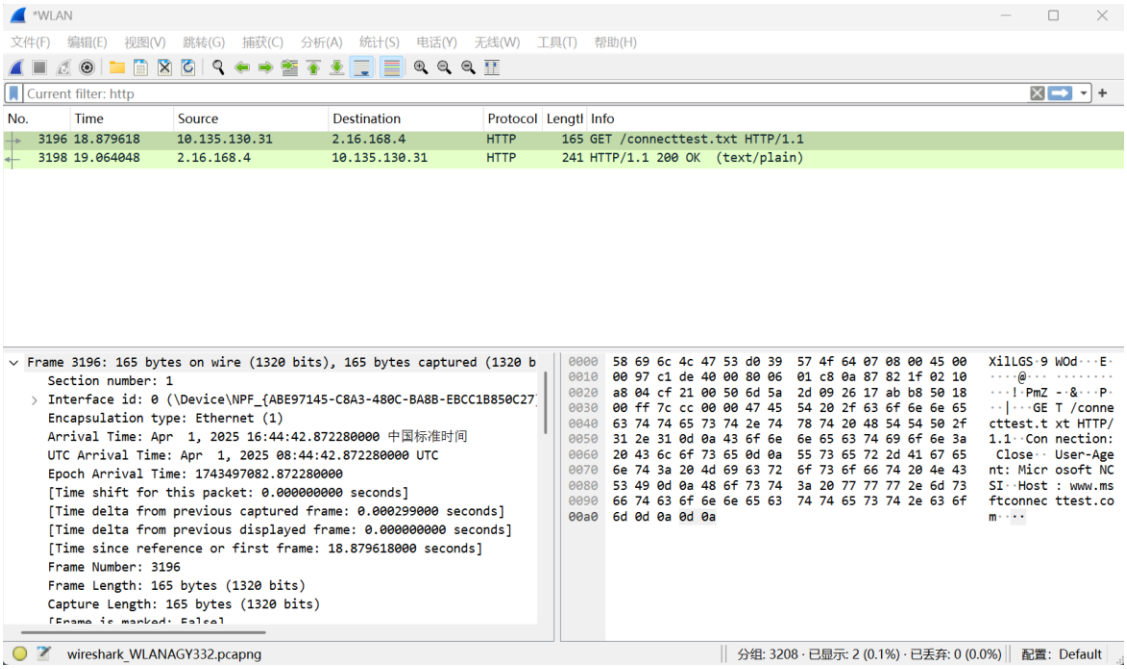


图 1 Wireshark 抓包实验结果

在这两个抓取的数据包中，第一个数据包是 HTTP 请求（GET），第二个数据包是 HTTP 响应（200 OK）。我对整体过程做如下分析：

第一个数据包 Frame 3196（HTTP GET 请求）

[1] 数据链路层（Ethernet II）

使用以太网协议，数据包发送源的 MAC 地址为 d0:39:57:4f:64:07，目标的 MAC 地址为 58:69:6c:4c:47:53。数据包从本地计算机发送到网络中的其他设备。

EtherType 是 0x0800，帧类型是 IPv4，传输的是 IPv4 数据。

图 2 数据链路层相关信息

[2] 网络层（IP）

使用 IPv4 协议，源的 IP 地址是 10.135.130.31，目标的 IP 地址是 2.16.168.4。数据包的标识符是 0xc1de，表明是一个独立的 IP 数据包 TTL（生存时间）为 128，说明数据包在路由时的最大跳数，防止在网络中无限循环。

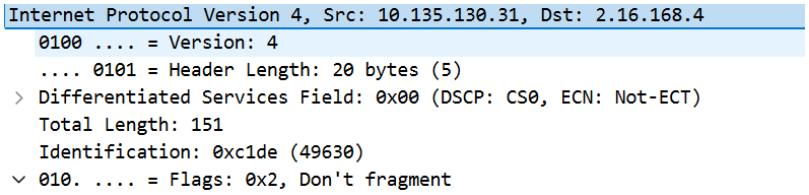


图 3 网络层相关信息

[3] 运输层（TCP）

源端口是 53025

目标端口是 80，目标端口是 HTTP 服务的端口

序列号 1 和确认号 1 确保数据包的顺序和确认机制

该数据包的标志位显示 PSH+ACK，表示数据传输完成并且需要确认

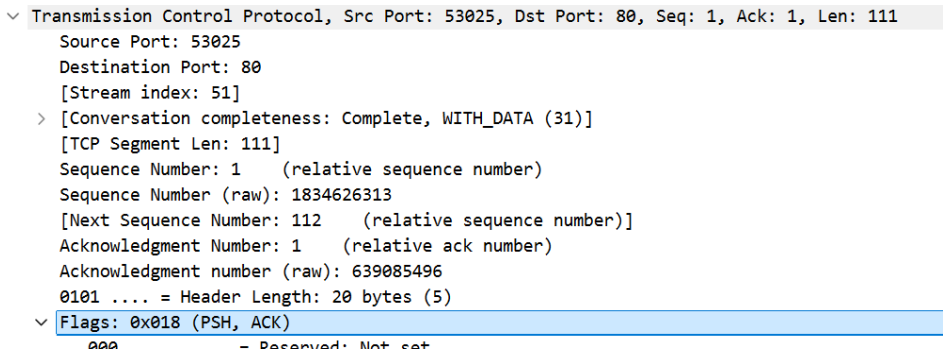


图 4 运输层相关信息

[4] 应用层（HTTP）

该数据包是一个 GET 请求，请求 URI 是/connecttest.txt，协议版本是 HTTP/1.1。

请求头包含三个部分：Connection: Close 表示客户端请求结束后关闭连接；User-Agent: Microsoft NCSI 表示使用 Microsoft 网络连接状态指示的客户端发起的请求；Host: www.msftconnecttest.com 表示目标服务器的主机名。

图 5 应用层相关信息

第二个数据包 Frame 3198（HTTP 200 响应）

[1] 数据链路层（Ethernet II）

源 MAC 地址是 58:69:6c:4c:47:53（RuijieNetwor_4c:47:53），目标地址是 d0:39:57:4f:64:07（LiteonTechno_4f:64:07），与发送时相反，表示响应从目标设备返回到源设备。EtherType 相同，帧类型仍为 IPv4

图 6 服务器传回原主机的数据链路层信息

[2] 网络层（IP）

源 IP 地址是 2.16.168.4，目标 IP 地址是 10.135.130.31，即响应从目标服务器返回到客户端计算机。

Time to Live 为 42，数据包在经过路由时经过了多个跳数

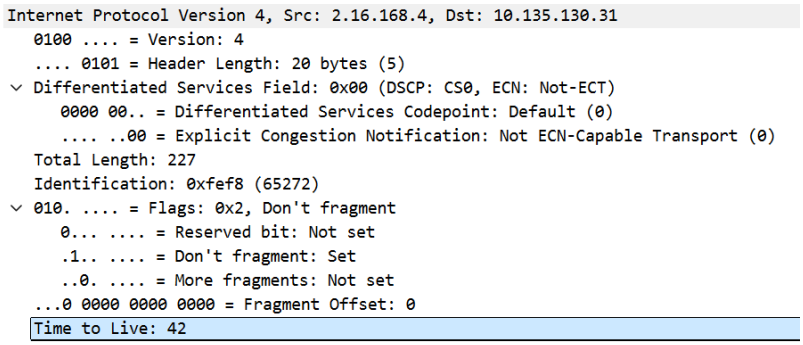


图 7 服务器传回原主机的网络层信息

[3] 运输层（TCP）

源端口为 80，目标端口为 53025，与数据包 1 传输时候正好相反，此时该响应是从 HTTP 服务器返回到客户端。

```
Transmission Control Protocol, Src Port: 80, Dst Port: 53025, Seq: 1, Ack: 112, Len: 187
  Source Port: 80
  Destination Port: 53025
  [Stream index: 51]
v [Conversation completeness: Complete, WITH_DATA (31)]
    ..0. .... = RST: Absent
    ...1 .... = FIN: Present
    .... 1... = Data: Present
    .... .1.. = ACK: Present
    .... ..1. = SYN-ACK: Present
    .... ...1 = SYN: Present
    [Completeness Flags: ·FDASS]
  [TCP Segment Len: 187]
  Sequence Number: 1    (relative sequence number)
  Sequence Number (raw): 639085496
  [Next Sequence Number: 188    (relative sequence number)]
  Acknowledgment Number: 112    (relative ack number)
  Acknowledgment number (raw): 1834626424
  0101 .... = Header Length: 20 bytes (5)
v Flags: 0x018 (PSH, ACK)
```

图 8 服务器传回原主机的运输层信息

该数据包的序列号和确认号与之前相同，标志位仍为 PSH+ACK，表示数据包已经推送且需要确认

[4] 应用层（HTTP）

该数据包是一个 HTTP 响应，状态码为 200 OK，表示请求成功。

```
Hypertext Transfer Protocol
v HTTP/1.1 200 OK\r\n
  v [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      [HTTP/1.1 200 OK\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
v Content-Length: 22\r\n
    [Content length: 22]
  Date: Tue, 01 Apr 2025 08:44:42 GMT\r\n
  Connection: close\r\n
  Content-Type: text/plain\r\n
  Cache-Control: max-age=30, must-revalidate\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.184430000 seconds]
  [Request in frame: 3196]
  [Request URI: http://www.msftconnecttest.com/connecttest.txt]
  File Data: 22 bytes
Line-based text data: text/plain (1 lines)
```

图 9 服务器传回原主机的运输层信息

Content-Length: 22：响应体的长度为 22 字节

Connection: close，表示服务器将在响应后关闭连接

Content-Type: text/plain，表示响应的内容是纯文本

Cache-Control: max-age=30, must-revalidate，表示缓存控制的策略

1. 数据包的封装的分析

通过对两个 HTTP 数据包的封装分析，我对数据封装过程总结如下：在第

一个数据包（HTTP GET 请求）中，数据从应用层开始，通过运输层和网络层，最终在数据链路层被封装成以太网帧，发送到目标服务器。在第二个数据包（HTTP 200 响应）中，响应数据同样经过数据链路层、网络层和运输层的封装后，从服务器返回给客户端。

表 1 Frame 3196（HTTP GET 请求）

协议层	封装内容	关键字段/作用
物理层	电/光信号传输	无体现
数据链路层	EthernetII 帧头	目标 MAC：58:69:6c:4c:47:53（网关） 源 MAC：d0:39:57:4f:64:07
网络层	IPv4 头部	源 IP：10.135.130.31（内网地址） 目标 IP：2.16.168.4（公网地址） TTL：128
传输层	TCP 头部	源端口：53025（客户端随机端口） 目标端口：80（HTTP） 确认号：1　　标志位：PSH+ACK
应用层	HTTP 请求	方法：GET/connecttest.txt Host：www.msftconnecttest.com

表 2 Frame 3198（HTTP 200 响应）

协议层	封装内容	关键字段/作用
物理层	电/光信号传输	无体现
数据链路层	EthernetII 帧头	目标 MAC：d0:39:57:4f:64:07 源 MAC：58:69:6c:4c:47:53（网关）
网络层	IPv4 头部	源 IP：2.16.168.4（公网地址） 目标 IP：10.135.130.31（内网地址） TTL：42
传输层	TCP 头部	源端口：80（HTTP） 目标端口：53025（客户端端口） 确认号：112　　标志位：PSH+ACK
应用层	HTTP 响应	状态码：200OK Content Type：text/plain

在数据链路层，源和目标的 MAC 地址指示了数据包的物理传输路径。在网络层，源和目标 IP 地址确保了数据能够在不同网络间传输。在运输层，TCP 协议提供了可靠的数据传输机制，确保数据包的顺序和完整性。在应用层，HTTP

协议定义了请求和响应的具体内容和格式，包括请求方法和 URI 等。

流程表示出来就是：HTTP 数据→添加 TCP 头部（端口、序列号）→添加 IP 头部（源/目标 IP）→添加 Ethernet 头部（MAC 地址）→物理层传输

多层次的协议封装机制使得数据能够在复杂的网络环境中可靠且高效地传输，确保应用程序和用户之间的通信正常进行。

2. 首部和协议层次的对应关系

对上述过程的首部和协议层之间的对应关系做如下整理：

表 3 数据链路层（EthernetII）

首部字段	示例值（Frame3196）	示例值（Frame3198）
目标 MAC 地址	58:69:6c:4c:47:53	d0:39:57:4f:64:07
源 MAC 地址	d0:39:57:4f:64:07	58:69:6c:4c:47:53
Ether Type	0x0800，IPv4	0x0800

表 4 网络层（IPv4）

首部字段	示例值（Frame3196）	示例值（Frame3198）
源 IP 地址	10.135.130.31	2.16.168.4
目标 IP 地址	2.16.168.4	10.135.130.31
TTL	128	42
协议号	6	6

表 5 传输层（TCP）

首部字段	示例值（Frame3196）	示例值（Frame3198）
源端口	53025	80
目标端口	80	53025
序列号	1	1
确认号	1	112
标志位	PSH+ACK	PSH+ACK

表 6 应用层（HTTP）

首部字段	示例值（Frame3196）	示例值（Frame3198）
请求方法	GET/connecttest.txt	不适用
Host	www.msftconnecttest.com	不适用
User-Agent	Microsoft NCSI	不适用
状态码	不适用（请求包）	200OK（请求成功）
Content-Type	不适用（请求包）	text/plain

文字分析如下：

[1] 数据链路层：以太网帧的局域网寻址

目标 MAC 地址指向网关

源 MAC 地址标识客户端的物理网卡

Ether Type 字段声明上层协议，确保接收设备正确解析后续内容

[2] 网络层：IP 地址的跨网络路由

源 IP 地址为客户端内网地址

目标 IP 地址为公网服务器的真实 IP

协议号声明上层为 TCP 协议，确保接收方将数据交给传输层处理

[3] 传输层：TCP 的可靠连接管理

源端口是客户端随机选择的临时端口

目标端口指向 HTTP 服务

序列号和确认号协同工作

标志位表示数据需立即推送至应用层，并确认此前通信的有效性

[4] 应用层：HTTP 的语义化交互

请求方法声明客户端希望获取指定资源

Host 指定目标域名

User-Agent 用于验证互联网访问能力

响应包的状态码表示请求成功

Content-Type 声明响应内容为纯文本

Content-Length 指明数据长度

说白了一句话总结：下层为上层提供服务，上层依赖下层实现功能

将复杂网络通信被分解为可管理的层次，实现从比特流到数据的转换

3. 实验手册上题目答案

对上述过程的首部和协议层之间的对应关系做如下整理：

（1）3 种不同的协议：HTTP，TCP，IPv4

（2）HTTP GET 请求和 HTTP OK 响应之间的时间间隔：

HTTP GET 请求时间是 18.87961 秒，HTTP OK 响应的时间是 19.06404 秒

时间间隔是：19.064048 - 18.879618 = 0.18443 秒

（3）gaia.cs.umass.edu 的 IP 地址是 2.16.168.4

我的电脑的 IP 地址是 10.135.130.31

（4）HTTP GET 请求：

GET /connecttest.txt HTTP/1.1

Connection: Close

User-Agent: Microsoft NCSI

Host: www.msftconnecttest.com

HTTP OK 响应：

HTTP/1.1 200 OK

Content-Length: 22

Date: Tue, 01 Apr 2025 08:44:42 GMT

Connection: close

Content-Type: text/plain

Cache-Control: max-age=30, must-revalidate

实验 2：tracert 命令及其分析

命令行中执行 tracert gaia.cs.umass.edu 命令，显示目标网站的 IP 地址为 128.119.245.12，运行记录如下。记住该地址，后续输入到 Wireshark 的过滤器中进行筛选

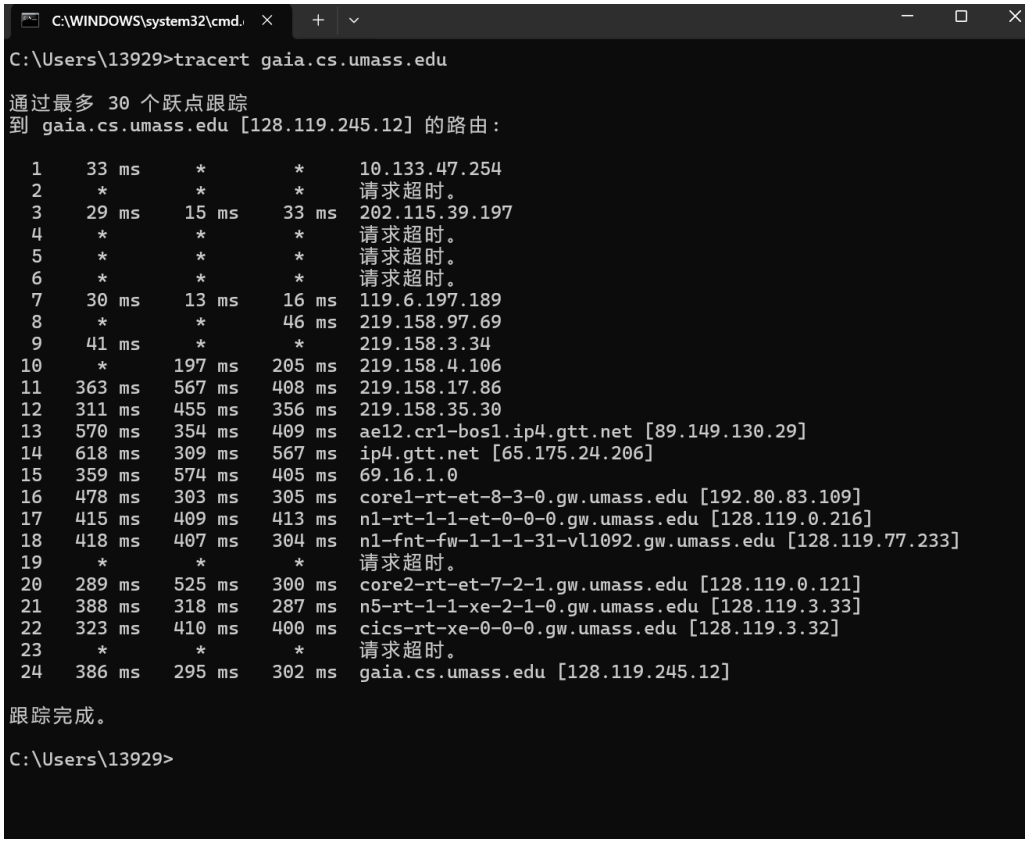


图 10 命令行中执行 tracert gaia.cs.umass.edu 命令

同时在 Wireshark 开始抓包，过滤器中输入 ip.addr==128.119.45.12

运行截图如下：

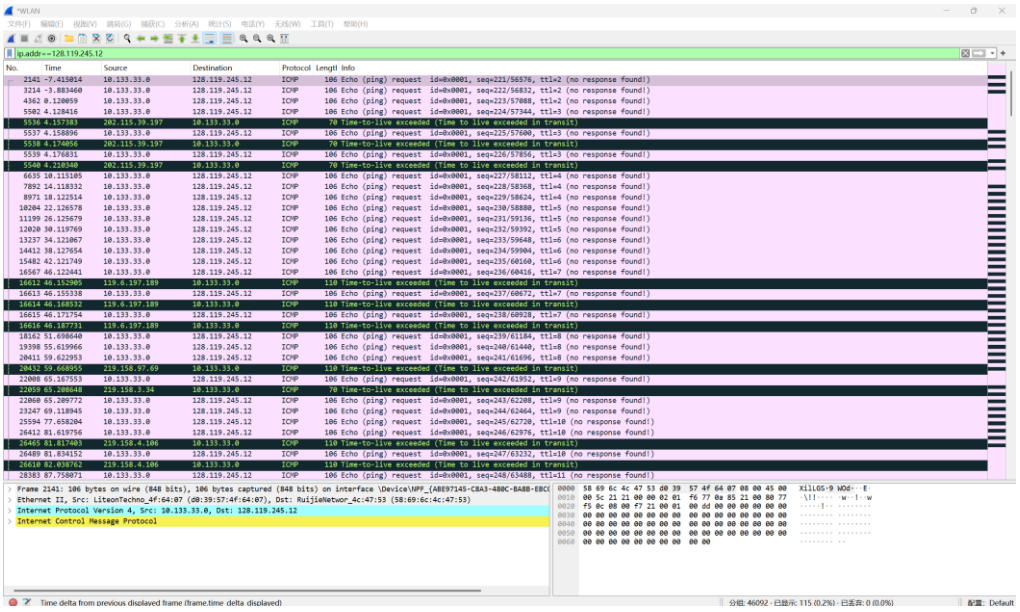


图 11 Wireshark 中抓包情况截图

我先对命令行的内容进行分析

第 1 跳：1 33 ms * * 10.133.47.254

数据包从源主机开始发送，并到达目标路由器 10.133.47.254。这里返回了有效的 RTT（33 毫秒），表示此跃点的响应正常。其他两列*表示没有进一步的 ICMP 响应。

第 2 跳：请求超时

请求超时说明路由器没有响应 ICMP 请求。可能是因为路由器配置了防火墙或策略，阻止了 ICMP 回应。

第 3 跳：3 29 ms 15 ms 33 ms 202.115.39.197

从源主机到 202.115.39.197 的延迟分别是 29ms、15ms 和 33ms。返回了有效的响应时间，表明该跃点响应正常。

第 4 到 6 跳：

```
4     *        *        *     请求超时。
5     *        *        *     请求超时。
6     *        *        *     请求超时。
```

连续的“请求超时”，数据包经过这些路由节点时没有响应。可能是这些路由器未响应 ICMP 请求或由于网络拥堵，数据包未能成功返回。

第 7 跳：7 30 ms 13 ms 16 ms 119.6.197.189

数据包成功到达 119.6.197.189，并返回了有效的响应时间（30 ms、13 ms、16 ms），这些值表明该跃点正常工作。

第 8 跳：8 * * 46 ms 219.158.97.69

显示部分请求成功（46 毫秒），这表明某些 ICMP 请求可能因网络不稳定或拥堵而丢失，其他两列 * 表示没有回应。

第 9 跳：9 41 ms * * 219.158.3.34

成功响应的 RTT 为 41ms，但有两列*表示某些 ICMP 请求没有收到回应

第 10 到 15 行：

```
10     *      197 ms   205 ms  219.158.4.106
11   363 ms   567 ms   408 ms  219.158.17.86
12   311 ms   455 ms   356 ms  219.158.35.30
13   570 ms   354 ms   409 ms  ae12.cr1-bos1.ip4.gtt.net [89.149.130.29]
14   618 ms   309 ms   567 ms  ip4.gtt.net [65.175.24.206]
15   359 ms   574 ms   405 ms  69.16.1.0
```

显示了不同的路由节点响应，其中 RTT（响应时间）逐渐增大，尤其是在从 219.158.4.106 到 ae12.cr1-bos1.ip4.gtt.net 等远程节点的延迟显著增加，表明该区域的网络较为拥堵或较远。

在第 14 跳和 15 跳 RTT 的差异较大，这也可能是由于网络的不同路径或网络设备负载变化导致的。

第 16 到 22 跳：

```
16   478 ms   303 ms   305 ms  core1-rt-et-8-3-0.gw.umass.edu [192.80.83.109]
17   415 ms   409 ms   413 ms  n1-rt-1-1-et-0-0-0.gw.umass.edu [128.119.0.216]
18   418 ms   407 ms   304 ms  n1-fnt-fw-1-1-1-31-vl1092.gw.umass.edu [128.119.77.233]
19     *        *        *     请求超时。
20   289 ms   525 ms   300 ms  core2-rt-et-7-2-1.gw.umass.edu [128.119.0.121]
21   388 ms   318 ms   287 ms  n5-rt-1-1-xe-2-1-0.gw.umass.edu [128.119.3.33]
22   323 ms   410 ms   400 ms  cics-rt-xe-0-0-0.gw.umass.edu [128.119.3.32]
```

显示通过 UMass 的路由节点。大多数跃点的响应正常，延迟逐步增大，表明数据包通过校园网络的多个路由器时，响应时间变得稍长。

第 23 跳：23 * * * 请求超时。

此行显示的请求超时表明该跃点没有响应，可能是由于防火墙限制、网络配置或设备故障。

第 24 跳：24 386 ms 295 ms 302 ms gaia.cs.umass.edu [128.119.245.12]

最终的目标 gaia.cs.umass.edu 的响应时间在 295 毫秒到 386 毫秒之间，成功响应。目标服务器到达，表明数据包通过所有中间跃点成功到达目标。

tracert 路由追踪的结果总共经过了 24 跳，其中有多个跳出现了请求超时的情况，意味着这些路由器没有响应或拒绝了 ICMP 请求。

最初，第一个路由器（10.133.47.254）响应时间为 33 毫秒，而接下来的部分则出现了请求超时的情况。第三跳 202.115.39.197 的响应时间仍然较短，表示从本地网络到外部网络的连接较为稳定，但之后的跳跃变得不太稳定，许多跳跃都显示了请求超时，表明数据包未能通过这些节点到达目标。

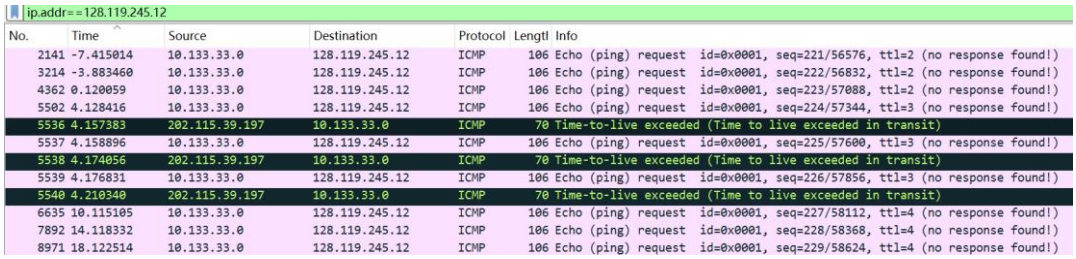
进入骨干网络后，延迟开始升高，尤其是从第 10 跳之后，延迟明显超过 300ms，甚至一度超过 500ms，代表数据已经跨洋传输（从中国到美国），同时可能在某些节点出现了拥塞或转发缓慢。

在第 13 跳时，数据包通过了 ae12.cr1-bos1.ip4.gtt.net，而响应时间显著

增加，显示出 570 毫秒、354 毫秒和 409 毫秒的延迟。随着路径逐渐接近目标，延迟也变得更大，这可能是由于跨国或跨区域的传输所致。

最终，到达目的地 gaia.cs.umass.edu 的第 24 跳显示了 386 毫秒、295 毫秒和 302 毫秒的延迟，表明网络连接较慢，但目标服务器最终成功响应。

Wireshark 中抓包太多，总计有 115 个包，我在这里不做过多分析，从中选取两个典型作为代表进行分析。


ip.addr==128.119.245.12

No.	Time	Source	Destination	Protocol	Lengtl	Info
2141	-7.415014	10.133.33.0	128.119.245.12	ICMP	106	Echo (ping) request id=0x0001, seq=221/56576, ttl=2 (no response found!)
3214	-3.883460	10.133.33.0	128.119.245.12	ICMP	106	Echo (ping) request id=0x0001, seq=222/56832, ttl=2 (no response found!)
4362	0.120059	10.133.33.0	128.119.245.12	ICMP	106	Echo (ping) request id=0x0001, seq=223/57088, ttl=2 (no response found!)
5502	4.128416	10.133.33.0	128.119.245.12	ICMP	106	Echo (ping) request id=0x0001, seq=224/57344, ttl=3 (no response found!)
5536	4.157383	202.115.39.197	10.133.33.0	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
5537	4.158896	10.133.33.0	128.119.245.12	ICMP	106	Echo (ping) request id=0x0001, seq=225/57600, ttl=3 (no response found!)
5538	4.174056	202.115.39.197	10.133.33.0	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
5539	4.176831	10.133.33.0	128.119.245.12	ICMP	106	Echo (ping) request id=0x0001, seq=226/57856, ttl=3 (no response found!)
5540	4.210340	202.115.39.197	10.133.33.0	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
6635	10.115105	10.133.33.0	128.119.245.12	ICMP	106	Echo (ping) request id=0x0001, seq=227/58112, ttl=4 (no response found!)
7892	14.118332	10.133.33.0	128.119.245.12	ICMP	106	Echo (ping) request id=0x0001, seq=228/58368, ttl=4 (no response found!)
8971	18.122514	10.133.33.0	128.119.245.12	ICMP	106	Echo (ping) request id=0x0001, seq=229/58624, ttl=4 (no response found!)

图 12 抓包情况截图

抓包成功分析 Frame 2141

包含一个 ping 请求，测试从源设备到目标服务器 128.119.245.12 的连通性

一、帧头信息

帧大小：106 字节

捕获接口：以太网接口，编号为\Device\NPF_{ABE97145-C8A3-480C-BA8B-EBCC1B850C27}

源 MAC 地址：LiteonTechno_4f:64:07（d0:39:57:4f:64:07），发送请求的主机网卡

目的 MAC 地址：RuijieNetwor_4c:47:53（58:69:6c:4c:47:53），本地网关或交换机

类型：IPv4（0x0800）

```
✓ Frame 2141: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface \Device\NPF_{ABE97145-C8A3-480C-BA8B-E
    Section number: 1
  ✓ Interface id: 0 (\Device\NPF_{ABE97145-C8A3-480C-BA8B-EBCC1B850C27})
      Interface name: \Device\NPF_{ABE97145-C8A3-480C-BA8B-EBCC1B850C27}
      Interface description: WLAN
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr  1, 2025 20:29:07.676709000 中国标准时间
    UTC Arrival Time: Apr  1, 2025 12:29:07.676709000 UTC
    Epoch Arrival Time: 1743510547.676709000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.006331000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: -7.415014000 seconds]
    Frame Number: 2141
    Frame Length: 106 bytes (848 bits)
    Capture Length: 106 bytes (848 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
```

图 13 Frame 2141 帧头信息

二、IP 头部信息

源 IP 地址：10.133.33.0，内部网络，说明这是从局域网内发出的请求。

目标 IP 地址：128.119.245.12

服务器地址 gaia.cs.umass.edu

TTL（Time to Live）：2，表示数据包最多还能经过多少个路由器跳点后才会被丢弃。TTL=2 意味着这是 traceroute 的第二跳发出的 ping 请求

协议：ICMP

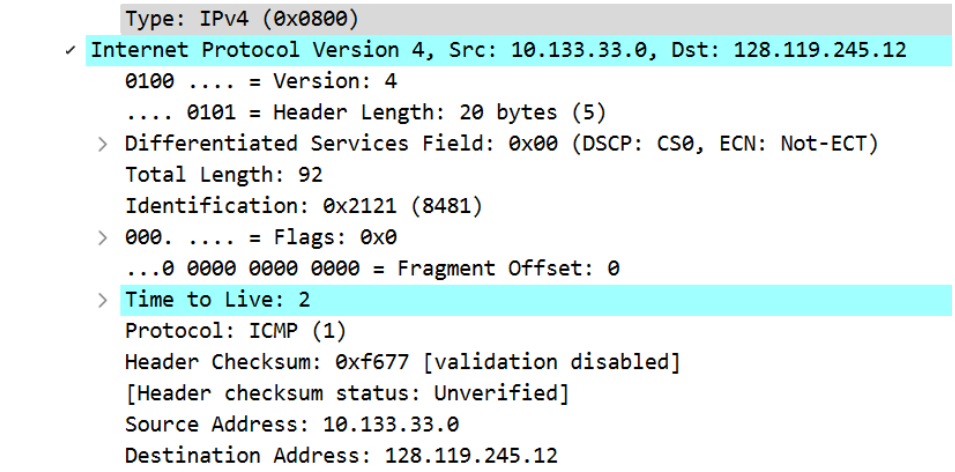


图 14 Frame 2141 IP 头部信息

三、ICMP 部分

类型：8（Echo Request），ping 请求，询问目标主机是否可达

代码（Code）：0

校验和（Checksum）：正确，说明数据未损坏

标识符（Identifier）：1（用于识别 ping 会话）

序列号（Sequence Number）：221（配合 Identifier 用于追踪多个 ping）

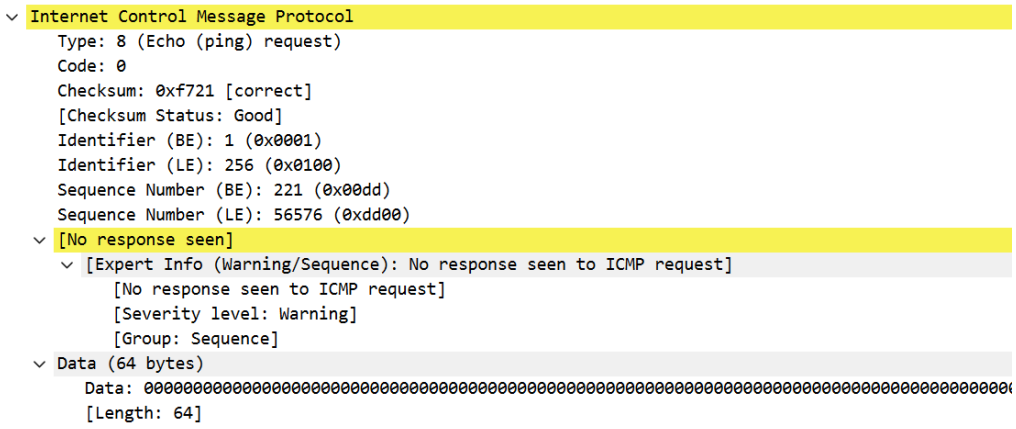


图 15 Frame 2141 ICMP 部分的信息

最后显示[Noresponseseen]，表明这帧之后没有看到对应的 ICMP Echo Reply，可能是目标地址未响应，或响应被防火墙或网络策略丢弃。

该帧的数据部分为 64 字节

抓包失败分析 Frame 5536

包含两个层次的 ICMP 数据包：一个是从目标设备发出的 ICMP 时间到生存

期超时错误消息，另一个是原始的 ping 请求。

一、帧头信息

帧大小：70 字节（560 位）

捕获接口：一个无线局域网接口，编号为\Device\NPF_{ABE97145-C8A3-480C-BA8B-EBCC1B850C27}

协议：以太网封装类型，帧类型为 IPv4

```
Frame 5536: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface \Device\NPF_{ABE97145-C8A3-480C-BA8B-E
    Section number: 1
  Interface id: 0 (\Device\NPF_{ABE97145-C8A3-480C-BA8B-EBCC1B850C27})
      Interface name: \Device\NPF_{ABE97145-C8A3-480C-BA8B-EBCC1B850C27}
      Interface description: WLAN
    Encapsulation type: Ethernet (1)
    Arrival Time: Apr  1, 2025 20:29:19.249106000 中国标准时间
    UTC Arrival Time: Apr  1, 2025 12:29:19.249106000 UTC
    Epoch Arrival Time: 1743510559.249106000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.000162000 seconds]
    [Time delta from previous displayed frame: 0.028967000 seconds]
    [Time since reference or first frame: 4.157383000 seconds]
    Frame Number: 5536
    Frame Length: 70 bytes (560 bits)
    Capture Length: 70 bytes (560 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:icmp:ip:icmp]
    [Coloring Rule Name: ICMP errors]
    [Coloring Rule String: icmp.type in { 3..5, 11 } || icmpv6.type in { 1..4 }]
```

图 16 Frame 5536 帧头信息

二、以太网头部

源 MAC 地址：RuijieNetwor_4c:47:53（58:69:6c:4c:47:53），发出此 ICMP 错误消息的设备

目标 MAC 地址：LiteonTechno_4f:64:07（d0:39:57:4f:64:07），接收 ICMP 错误消息的设备，发送了 ping 请求的主机

类型：IPv4（0x0800），都一样的

```
Ethernet II, Src: RuijieNetwor_4c:47:53 (58:69:6c:4c:47:53), Dst: LiteonTechno_4f:64:07 (d0:39:57:4f:64:07)
  Destination: LiteonTechno_4f:64:07 (d0:39:57:4f:64:07)
      Address: LiteonTechno_4f:64:07 (d0:39:57:4f:64:07)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Source: RuijieNetwor_4c:47:53 (58:69:6c:4c:47:53)
      Address: RuijieNetwor_4c:47:53 (58:69:6c:4c:47:53)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
```

图 17 Frame 5536 以太网头部信息

三、IPv4 头部

源 IP 地址：202.115.39.197：发出 ICMP 错误消息的设备的 IP 地址。

目标 IP 地址：10.133.33.0，接收 ICMP 错误消息的设备（发送 ping 请求的主机）

TTL 是 62，表明数据包还能经过 62 个路由器跳点，说明这个包已经在网络中传递了一段时间。

协议：ICMP（类型为 1），和之前的都一样，不过多分析

四、ICMP 部分的错误分析

先放实验截图：

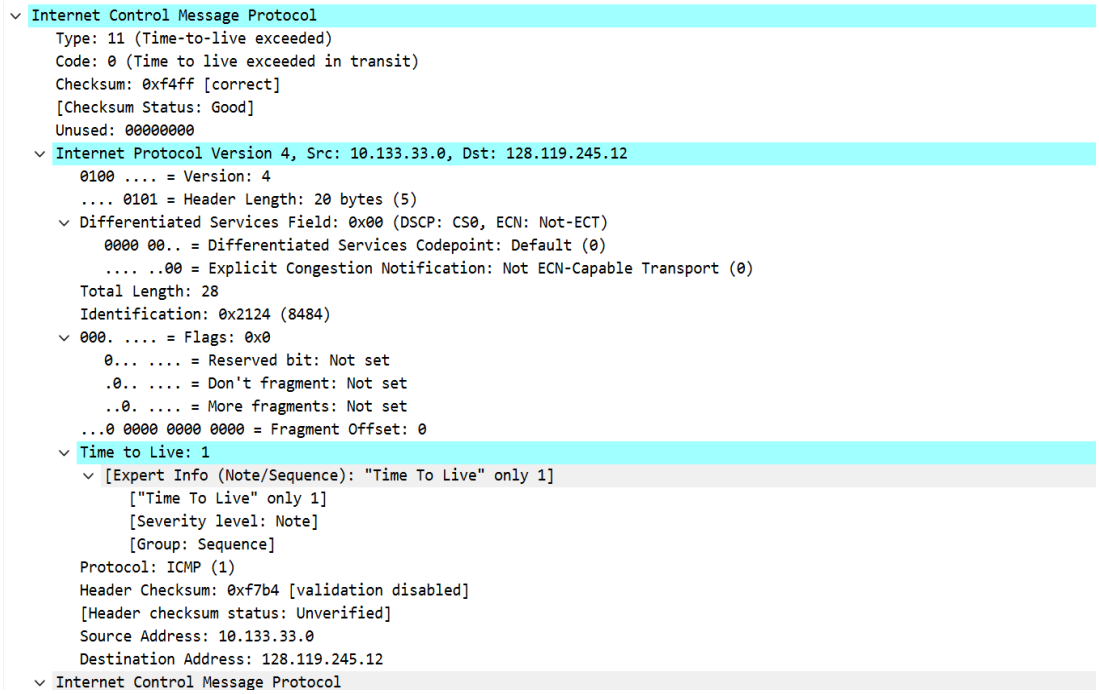


图 18 Frame 5536 ICMP 的错误信息

1. ICMP 错误消息：

类型（Type）：11（Time-to-live exceeded），一个超时错误消息，意味着数据包在传输过程中已经超过了它的 TTL，导致它在某个跳点被丢弃。

代码（Code）：0（Time to live exceeded in transit），数据包的 TTL 已到达 0，且被丢弃。

源地址：202.115.39.197 说明发出 ICMP 错误消息的是在 202.115.39.197 这个 IP 地址的设备。

目标地址：10.133.33.0，接收 ICMP 错误消息的设备，发送 ping 的主机

数据包：包含了原始的 ICMP 数据包作为内部数据（后面详细分析）

2.原始 ICMP Echo 请求：

类型：8，由源 IP 10.133.33.0 发出，目标为 128.119.245.12（目的地是网址 gaia.cs.umass.edu）

代码：0

标识符：1，用于标识该 ping 请求

序列号（Sequence Number）：224，是该请求的唯一标识符，用于区分不同的 ping 请求

TTL 值为 1，意味着这是一个刚刚发出的 ping 请求，而被丢弃的原因就是它的 TTL 已经过低，不能再跳跃到更多的路由器。

Frame 5536 是错误消息，在传输过程中，ping 请求的 TTL 达到 1 时就被丢弃了，且目标地址没有响应。由于 TTL 值设置为 1，使得数据包在第一跳就被丢弃，触发了 ICMP 的 TTL 超时错误。

最后，将超时的原因汇总如下：中间路由器不响应 ICMP 请求；丢包或网络不稳定（网络拥堵）；防火墙策略限制。

实验 2：ipconfig/all 命令及其分析

在命令行中执行 ipconfig/all 命令，显示出我的电脑配备了多个网卡，具体运行结果如下图所示：

```
C:\Users\13929>ipconfig/all

Windows IP 配置

   主机名  . . . . . . . . . . . . . : LAPTOP-196TL048
   主 DNS 后缀 . . . . . . . . . . . :
   节点类型  . . . . . . . . . . . . : 混合
   IP 路由已启用 . . . . . . . . . . : 否
   WINS 代理已启用 . . . . . . . . . : 否

无线局域网适配器 本地连接* 1:

   媒体状态  . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : Microsoft Wi-Fi Direct Virtual Adapter
   物理地址. . . . . . . . . . . . . : D2-39-57-4F-64-07
   DHCP 已启用 . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . : 是

无线局域网适配器 本地连接* 2:

   媒体状态  . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #2
   物理地址. . . . . . . . . . . . . : F2-39-57-4F-64-07
   DHCP 已启用 . . . . . . . . . . . : 否
   自动配置已启用. . . . . . . . . . : 是

以太网适配器 VMware Network Adapter VMnet1:

   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : VMware Virtual Ethernet Adapter for VMnet1
   物理地址. . . . . . . . . . . . . : 00-50-56-C0-00-01
   DHCP 已启用 . . . . . . . . . . . : 否
   自动配置已启用. . . . . . . . . . : 是
   本地链接 IPv6 地址. . . . . . . . : fe80::1b1d:3bdd:c1cf:8d62%4(首选)
   IPv4 地址 . . . . . . . . . . . . : 192.168.32.1(首选)
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 822104150
   DHCPv6 客户端 DUID  . . . . . . . : 00-01-00-01-2C-4A-02-B5-BC-EC-A0-07-3F-21
   TCPIP 上的 NetBIOS  . . . . . . . : 已启用
```

图 19 ipconfig/all 命令运行结果 1

```
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::b9f1:d7e:fe37:200a%2(首选)
   IPv4 地址 . . . . . . . . . . . . : 192.168.117.1(首选)
   子网掩码 . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 838881366
   DHCPv6 客户端 DUID . . . . . . . . : 00-01-00-01-2C-4A-02-B5-BC-EC-A0-07-3F-21
   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. . . . . . . . . . . . . . . : Realtek 8852CE WiFi 6E PCI-E NIC
   物理地址. . . . . . . . . . . . . : D0-39-57-4F-64-07
   DHCP 已启用 . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . : 是
   本地链接 IPv6 地址. . . . . . . . : fe80::4f17:5564:7d25:d769%18(首选)
   IPv4 地址 . . . . . . . . . . . . : 10.135.20.122(首选)
   子网掩码 . . . . . . . . . . . . : 255.255.240.0
   获得租约的时间 . . . . . . . . . : 2025年4月2日 13:46:08
   租约过期的时间 . . . . . . . . . : 2025年4月2日 15:16:08
   默认网关. . . . . . . . . . . . . : 10.135.31.254
   DHCP 服务器 . . . . . . . . . . . : 10.135.31.254
   DHCPv6 IAID . . . . . . . . . . . : 147863895
   DHCPv6 客户端 DUID . . . . . . . . : 00-01-00-01-2C-4A-02-B5-BC-EC-A0-07-3F-21
   DNS 服务器 . . . . . . . . . . . : 202.115.39.6
                                       202.115.39.9
   TCPIP 上的 NetBIOS . . . . . . . . : 已启用

以太网适配器 以太网:

   媒体状态 . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : Realtek PCIe GbE Family Controller
   物理地址. . . . . . . . . . . . . : BC-EC-A0-07-3F-21
   DHCP 已启用 . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . : 是
```

图 20 ipconfig/all 命令运行结果 2（接图 19）

无线适配器：WLAN，当前连接网络，IP 10.135.20.122，默认网关 10.135.31.254，我连接在四川大学的校园网上，网段为 10.135.16.0/20。

VMware 虚拟网卡：两个虚拟网卡 VMnet1 和 VMnet8，分别有 IP 地址 192.168.32.1 和 192.168.117.1，以前虚拟机搞的

Wi-Fi Direct 虚拟适配器：两个 Microsoft 虚拟 Wi-Fi 适配器，未连接

以太网适配器（有线网口）：未连接

在我做实验截图的时候，我的网络是通过无线适配器连接的（我连接了校园网），启用了 DHCP 自动获取 IP 和 DNS。

（1）我的网卡的 MAC 地址如下表所示：

表 7 我的电脑的 MAC 地址

适配器	MAC 地址
WLAN（无线网）	D0-39-57-4F-64-07
有线以太网	BC-EC-A0-07-3F-21
VMware VMnet1	00-50-56-C0-00-01
VMware VMnet8	00-50-56-C0-00-08
无线局域网适配器 1	D2-39-57-4F-64-07
无线局域网适配器 2	F2-39-57-4F-64-07

（2）根据 MAC 地址的 OUI 查询生产厂家，MAC 地址的前 12 个比特是 OUI（组织唯一标识符），用于标识生产厂商。

[1] 无线局域网适配器 WLAN

OUI：D0-39-57

厂家：Realtek Semiconductor Corp.

[2] 以太网适配器 以太网

OUI：BC-EC-A0

厂家：AzureWave Technologies, Inc.

[3] VMware Network Adapter VMnet1

OUI：00-50-56

厂家：VMware, Inc.

[4] VMware Network Adapter VMnet8

OUI：00-50-56

厂家：VMware, Inc.

[5] 无线局域网适配器 本地连接 1*

OUI：D2-39-57

厂家：Realtek Semiconductor Corp.

[6] 无线局域网适配器 本地连接 2*

OUI：F2-39-57

厂家：Realtek Semiconductor Corp.

----------END OF EXPERIMENT----------

Computer Network Wireshark Introduction
