Wireshark

Dr. Xiqun Lu
College of Computer Science and Technology,
Zhejiang University, Hangzhou

实验目的

- 通过分析各种不同网络协议，加深理解第一堂课中重点：“Protocol layering”
- Network architecture: a set of layers and protocols
- **Protocol stack**: a list of the protocols used by a certain system, one protocol per layer
  - A **protocol** defines the *format* and the *order* of messages exchanged between two or more communication entities, as well as the *actions* taken on the transmission and/or receipt of a message or other event. [2]

The TCP/IP Reference Model (IV)

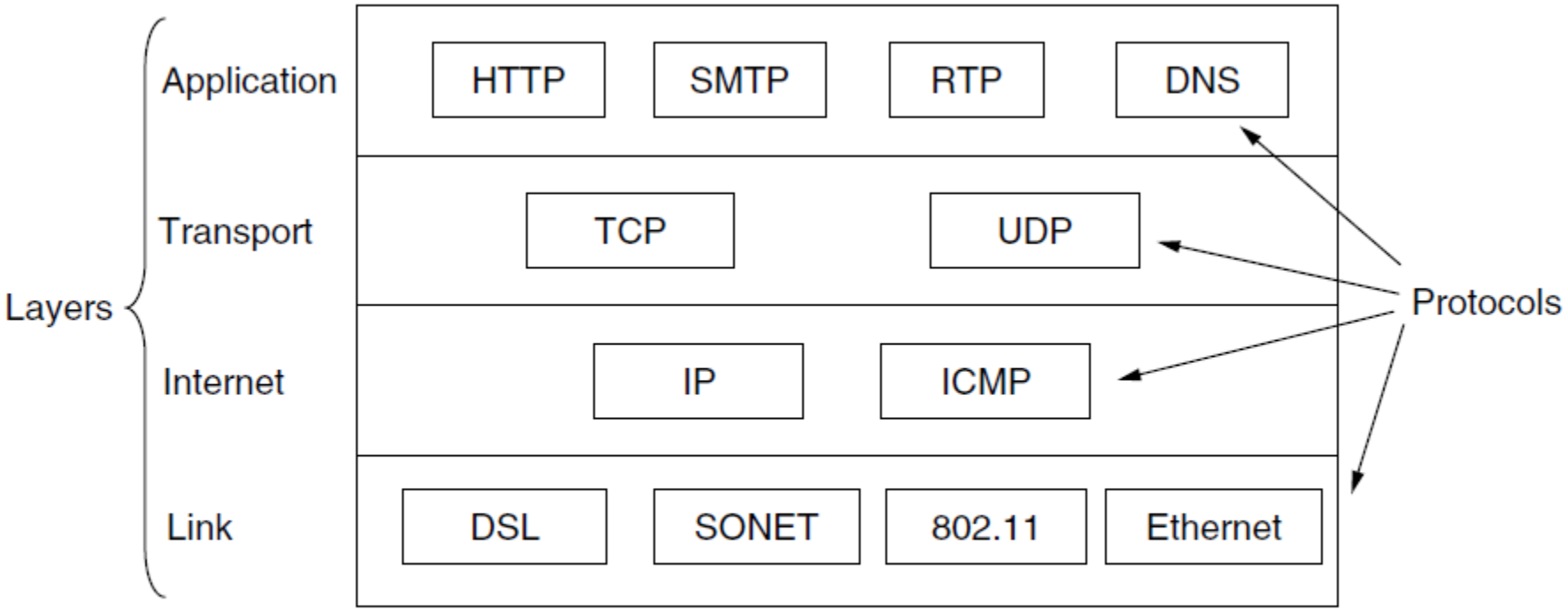
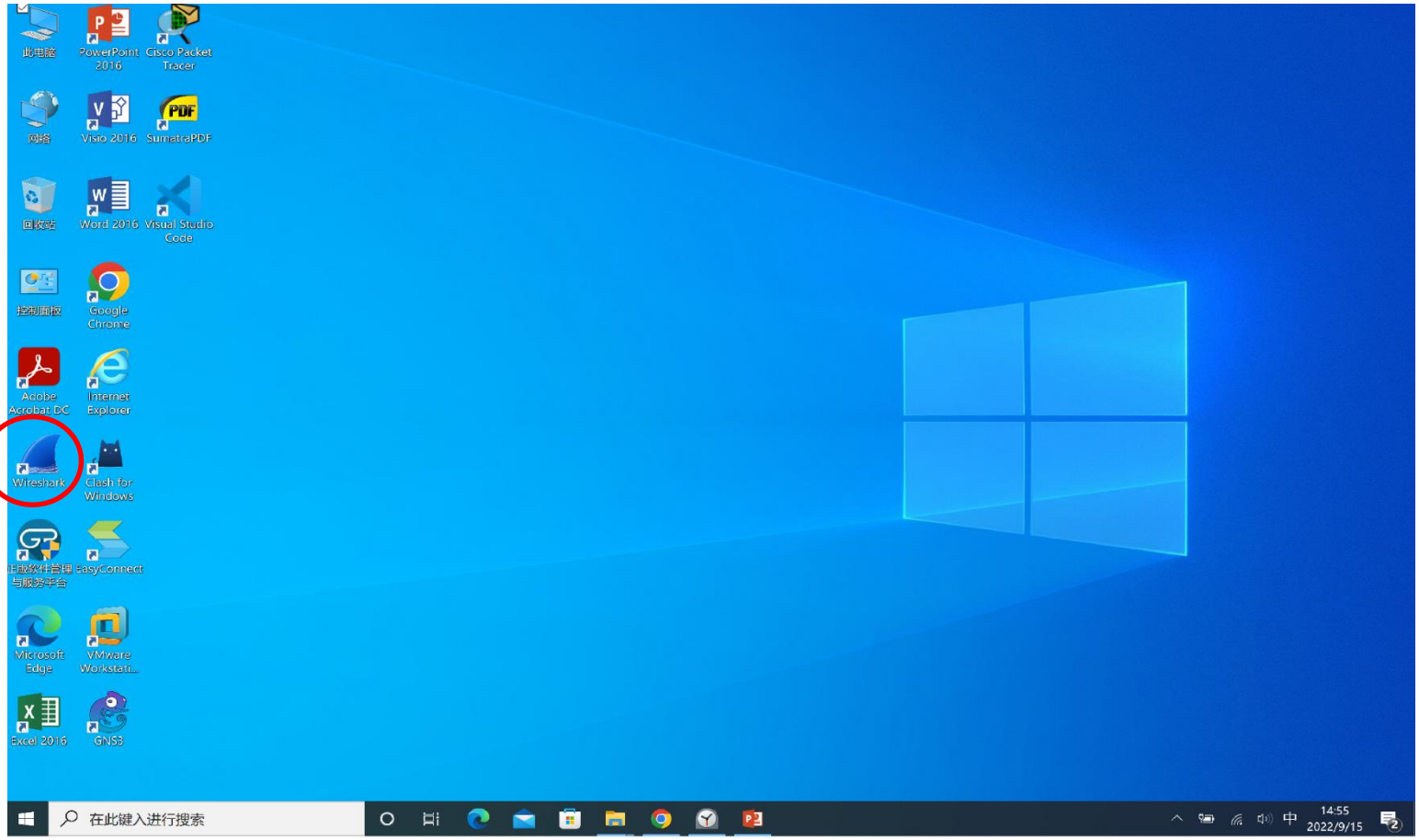


Figure 1-22. The TCP/IP model with some protocols we will study.

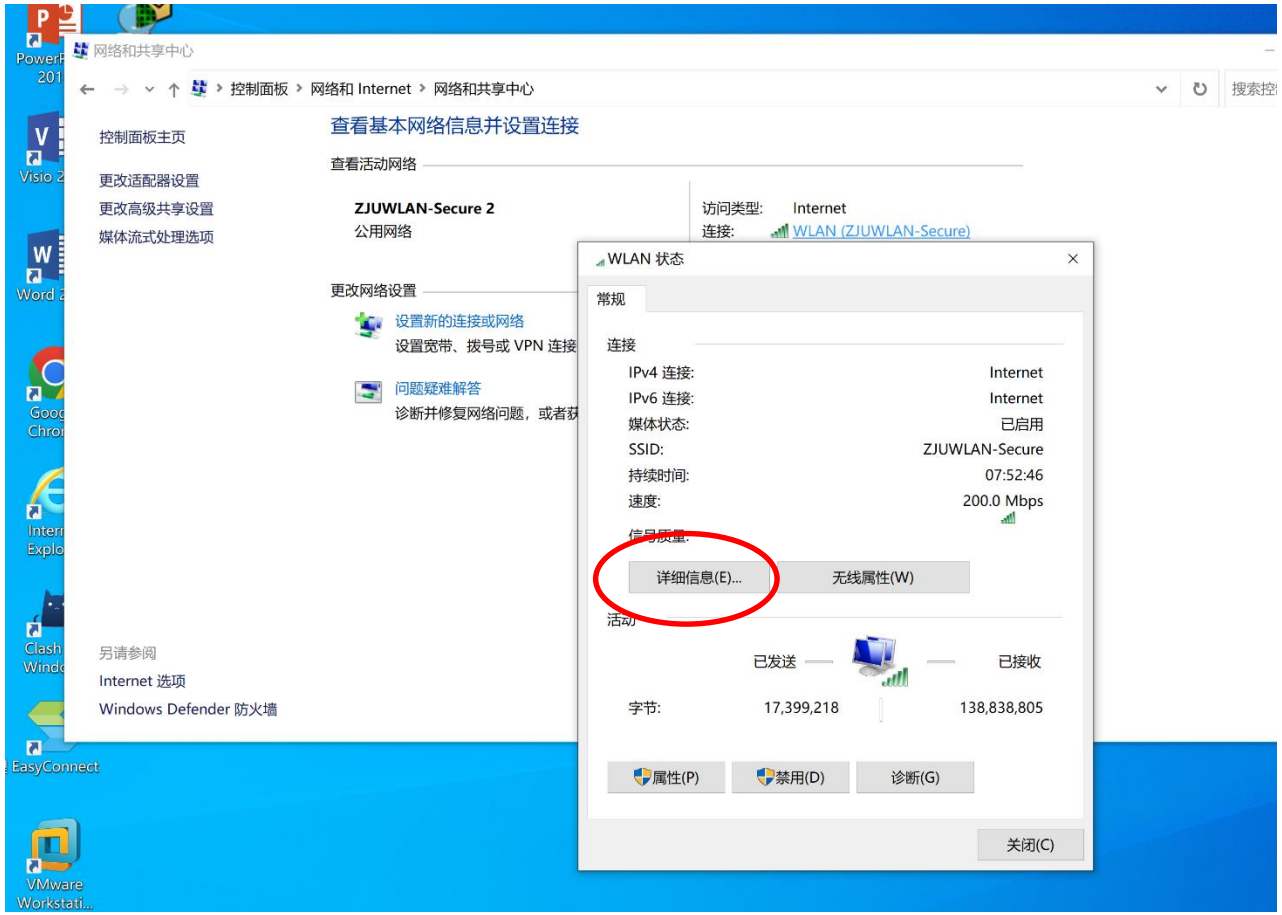
Step 1: WireShark

- 安装WireShark
  - 下载地址： https://www.wireshark.org/
  - 安装成功，桌面上会出现一个“鲨鱼鳍”的图标，如下图红圆圈中图标。



Step 2:了解自己电脑的一些网络一些设置

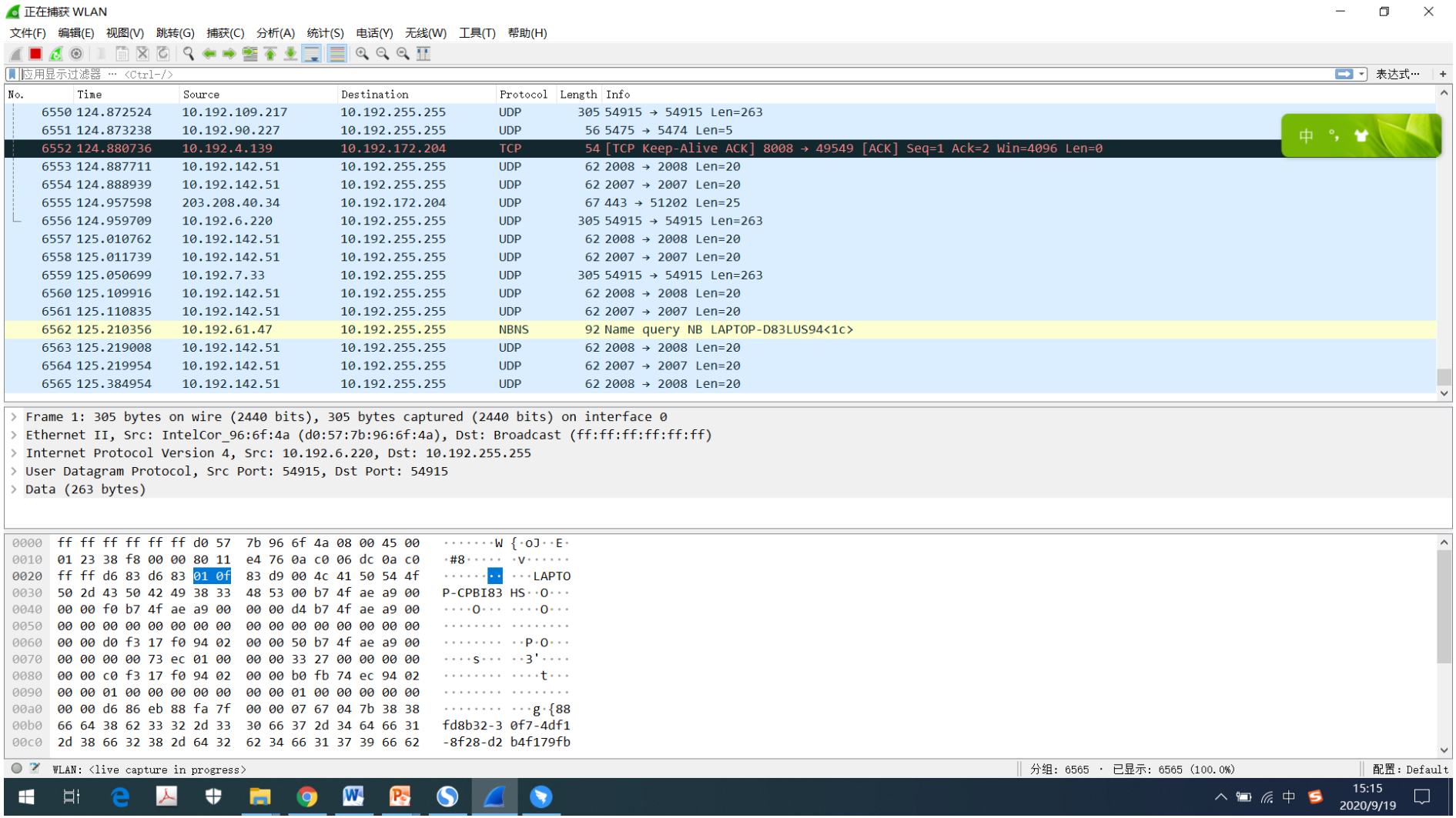
- 先在“控制面板”中打开网络中心，然后详细信息里有你电脑当前所使用的IP地址，你也可以看一下你电脑的物理地址（MAC地址）



Step 2:了解自己电脑的一些网络设置

- 一些协议名称
  - DNS
  - DHCP
- 一些地址信息与概念：
  - 子网掩码：255.255.0.0
  - IPv4地址(32bits)：10.162.54.132
  - IPv6地址(128bits)：240c:c781:7000:2d93:6133:b614:498b:82fc
    - IP地址随着使用环境变化而变化
  - 物理地址(MAC, 48bits): 34-2E-B7-DE-DD-DE
    - 如同人的身份证号

Interface of WireShark

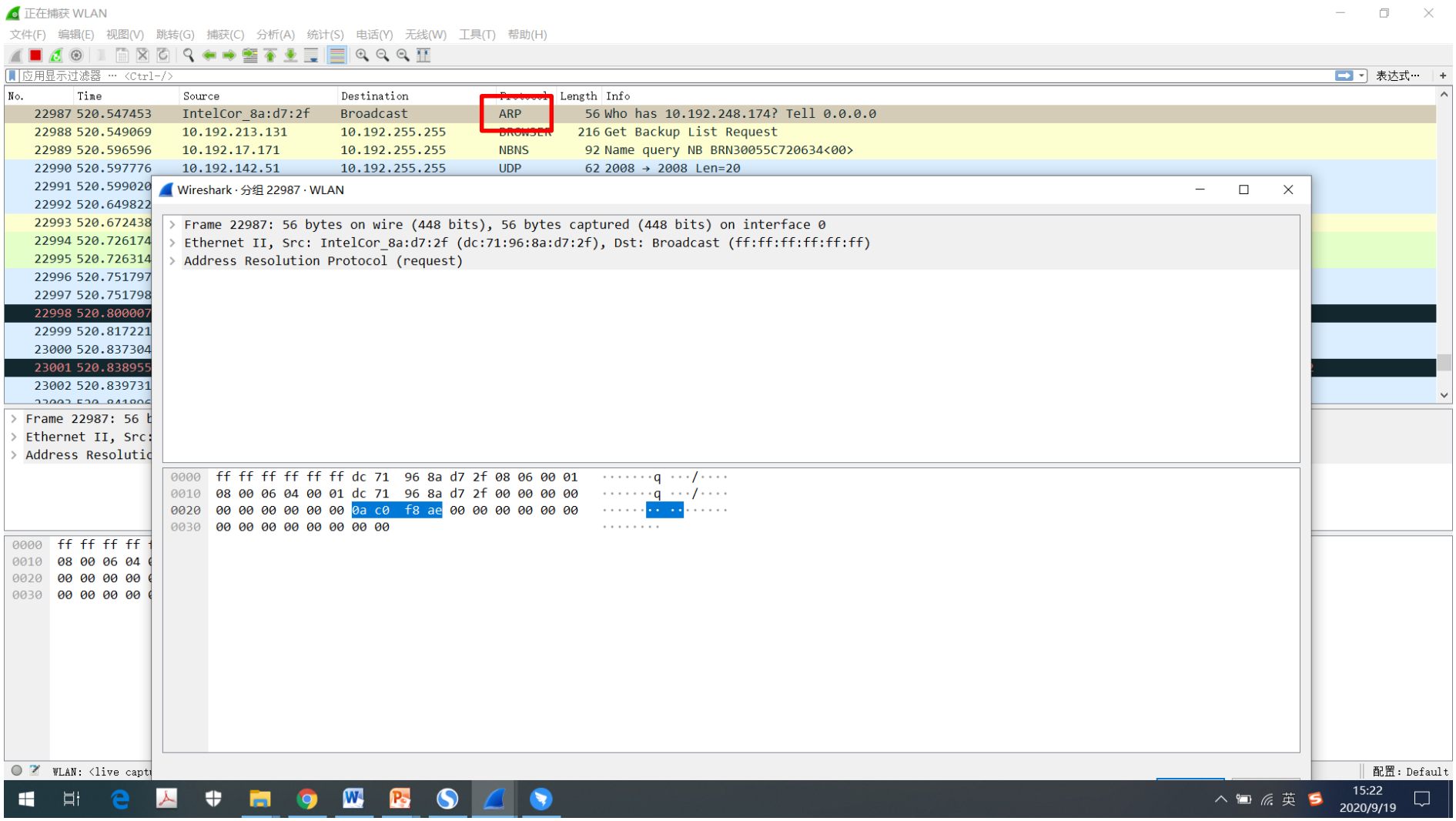


- 至上而下Wireshark三个面板：“Packet List”（分组列表），“Packet Detail”(分组详情)，“Packet Byte”(分组字节流)

Interface of Wireshark

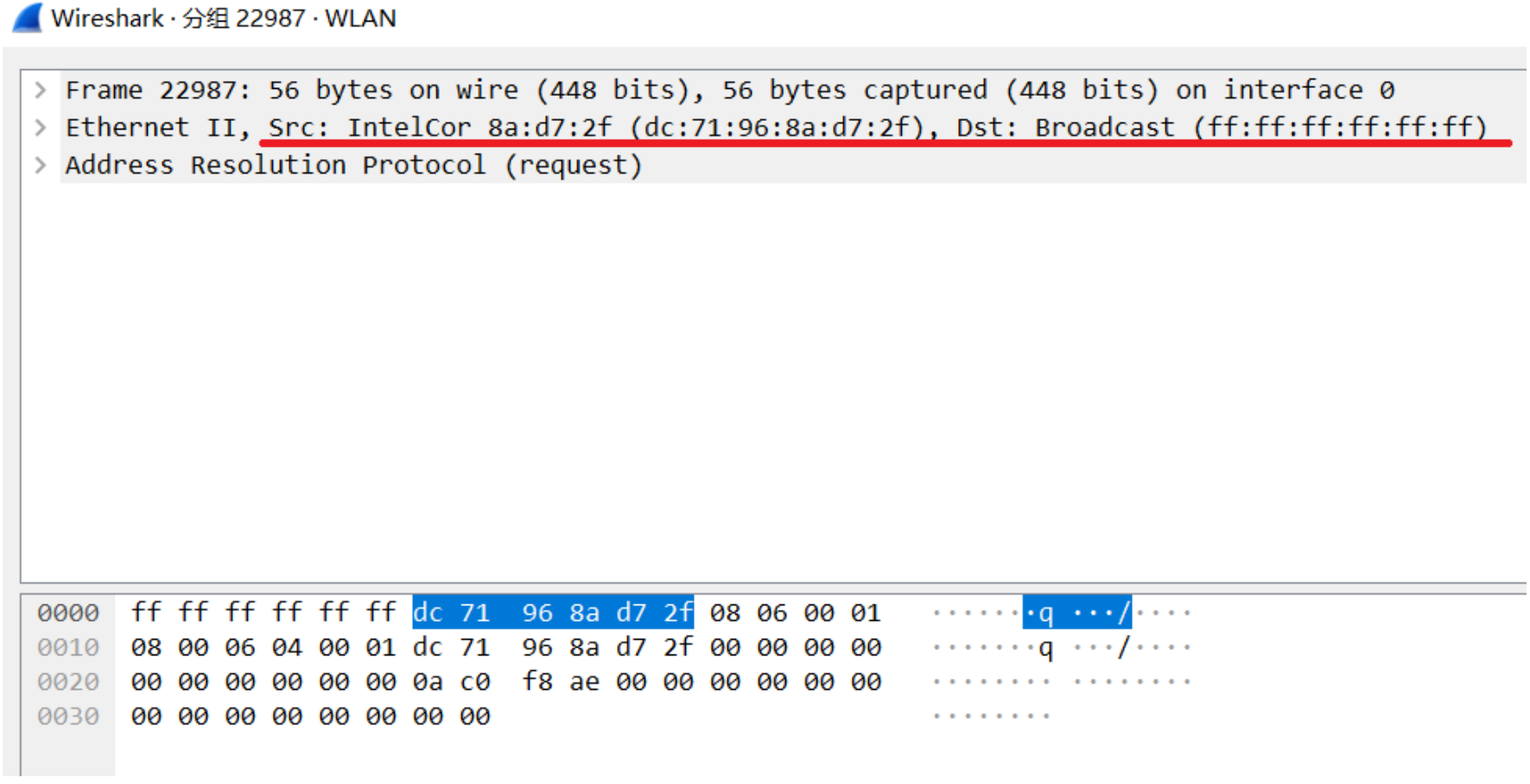
- Wireshark三个面板：
  - “Packet List”（分组列表）
  - “Packet Detail”(分组详情)
  - “Packet Byte”(分组字节流)
- 列表中的每行显示捕捉文件的一个包。如果你的鼠标移到其中一行上，该包的更多详细信息会显示在“Packet Detail/分组详情”和"Packet Byte/分组字节流"面板。
- 在分析(解剖)分组时，Wireshark会将协议信息放到各个列。因为高层协议通常会覆盖底层协议，您通常在分组列表面板看到的都是每个包的最高层协议描述。（在Wireshark中最高层是应用层，底层是数据链路层）

Example I: ARP



ARP: Address Resolution Protocol [2]

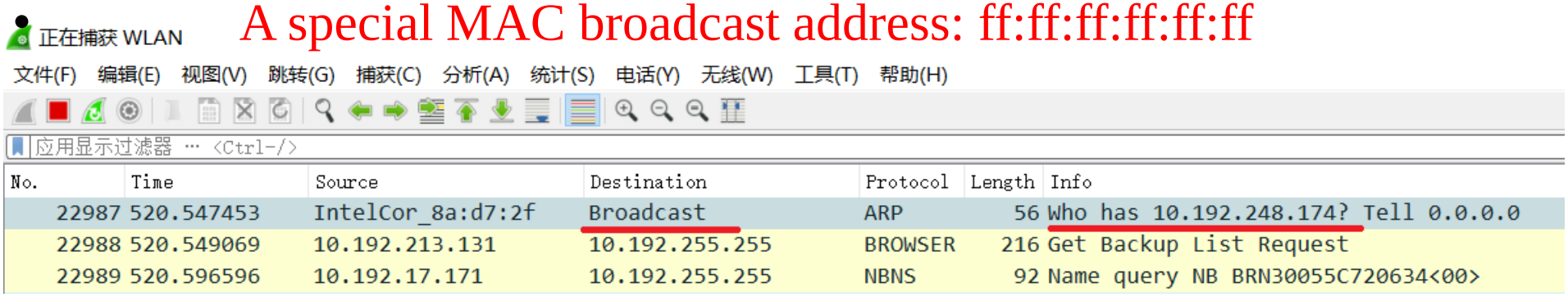
- Because there are both *network-layer addresses* (for example, Internet **IP addresses**) and *link-layer addresses* (that is, **MAC addresses**), there is a need to translate between them. For the Internet, this is the job of the **Address Resolution Protocol** (**ARP**) [RFC826]



ARP: Address Resolution Protocol [2]

- The purpose of the ARP query packet is to query all the other nodes on the subnet to determine the MAC address corresponding to the IP address that is being resolved.
- An ARP query packet (In this example: it give the IP address: 10.192.248.174, want to know the MAC address of the IP address. A IPv4 address has 32 bits and expressed in decimals (十进制) ×××.×××.×××.×××)
- The MAC addresses are 6 bytes long, giving 2^{48} possible MAC addresses, and are expressed in hexadecimal (十六进制). (In this example: the MAC address of the source is dc:71:96:8a:d7:2f)

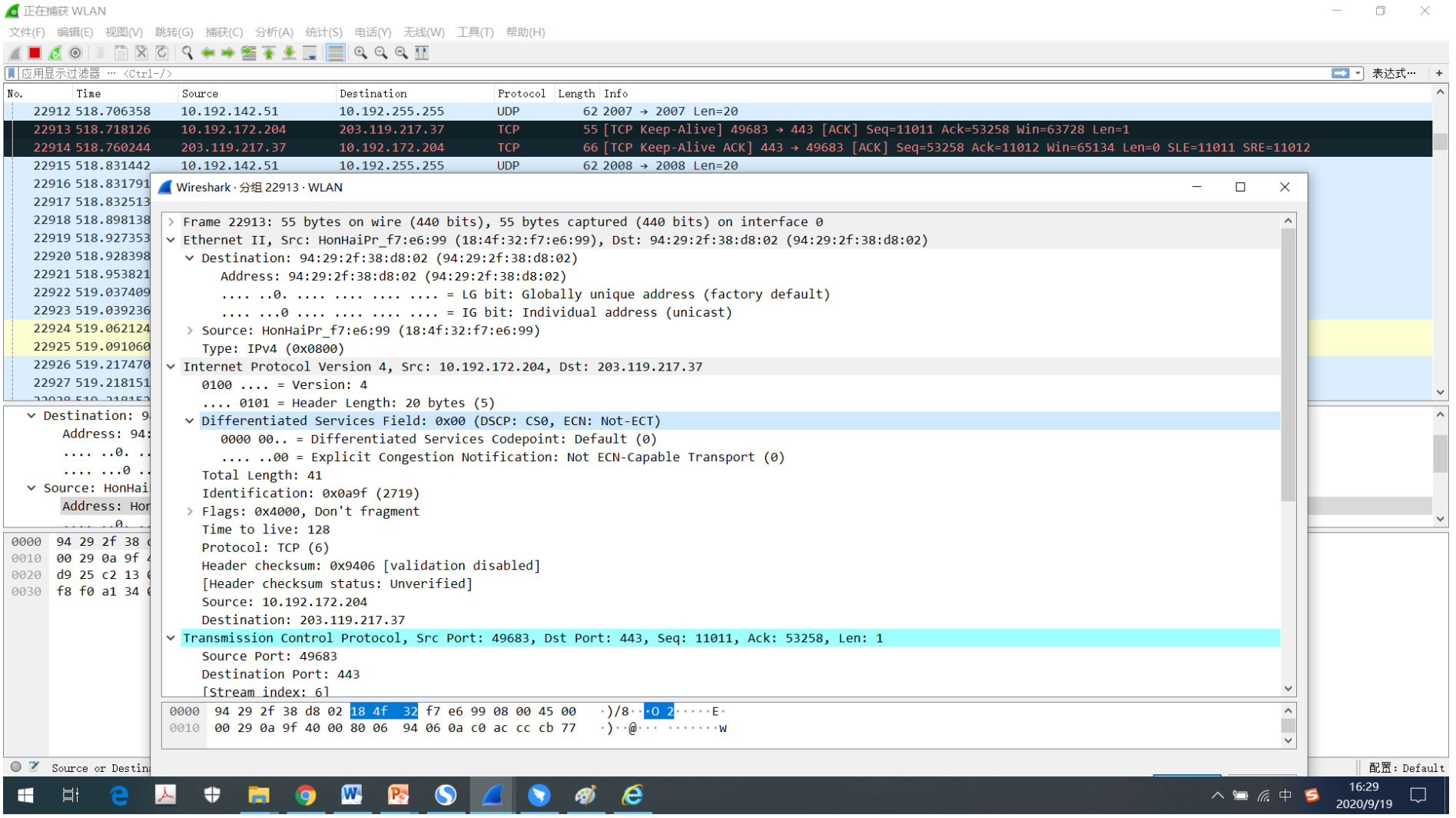
A special MAC broadcast address: ff:ff:ff:ff:ff:ff



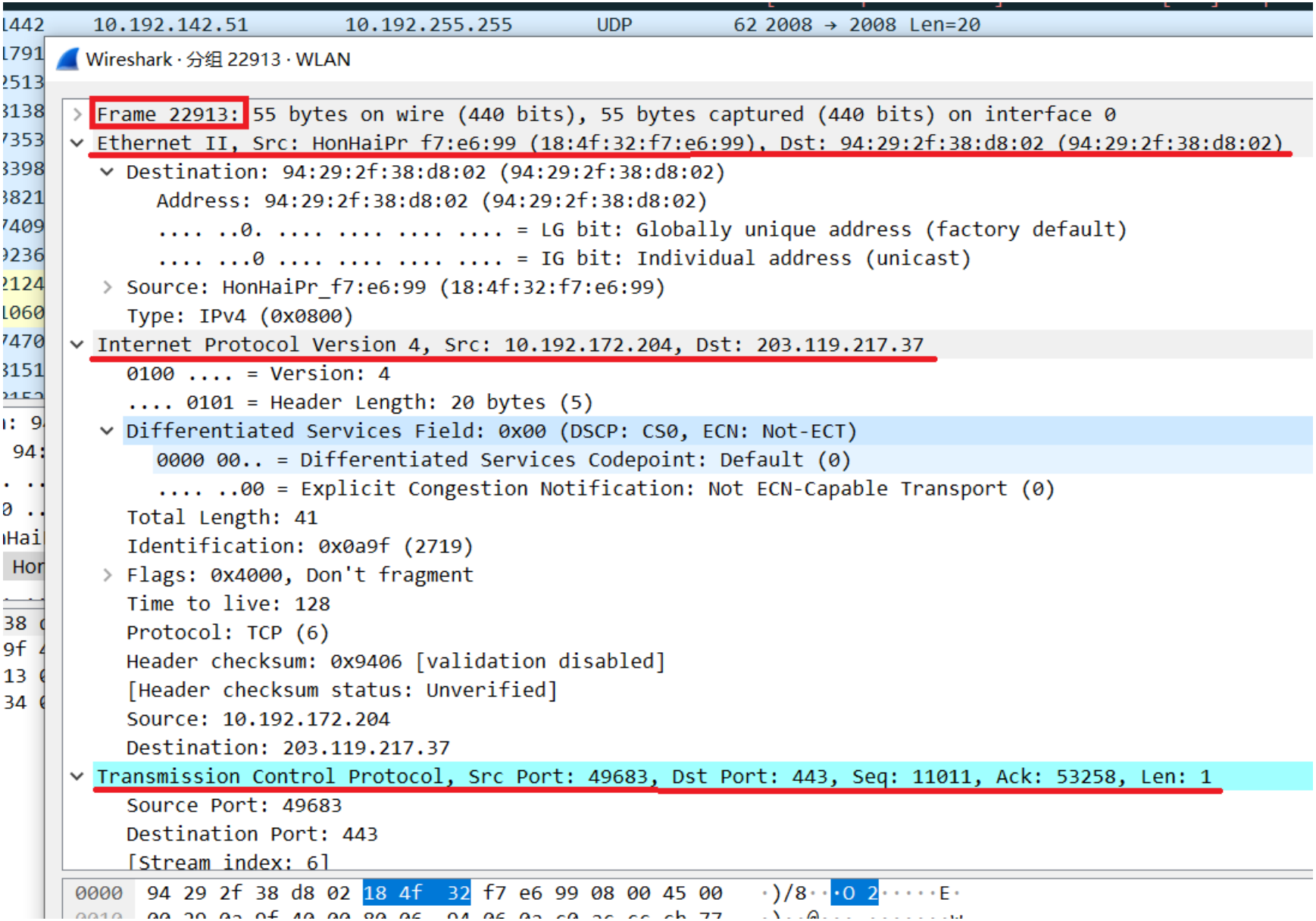
ARP: Address Resolution Protocol [2]

- Each node (host and router) has **an ARP table** in its memory, which contains mappings of IP addresses to MAC addresses.
- The ARP table contains a time-to-live (**TTL**) value, which indicates when each mapping will be deleted from the table.
  - A typical expiration time for an entry is 20 minutes from when an entry is placed in an ARP table.
- ARP vs. DNS
  - ARP resolves an IP address to a MAC address only for nodes on the same subnet.
  - DNS resolves host names to IP addresses for hosts anywhere in the Internet.
- ARP is probably best considered a protocol that straddles the boundary between the link and network layers.

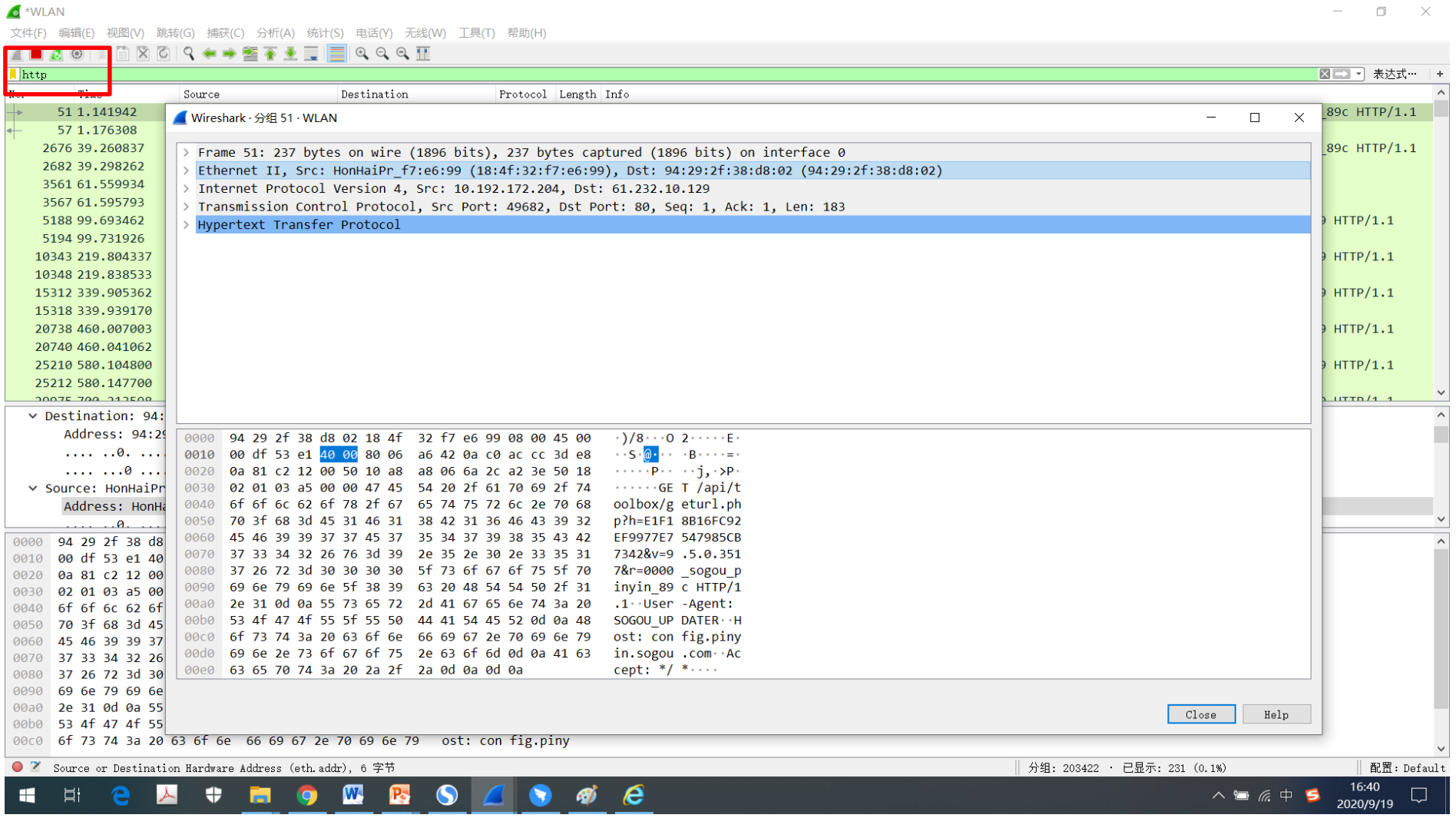
Example II: TCP



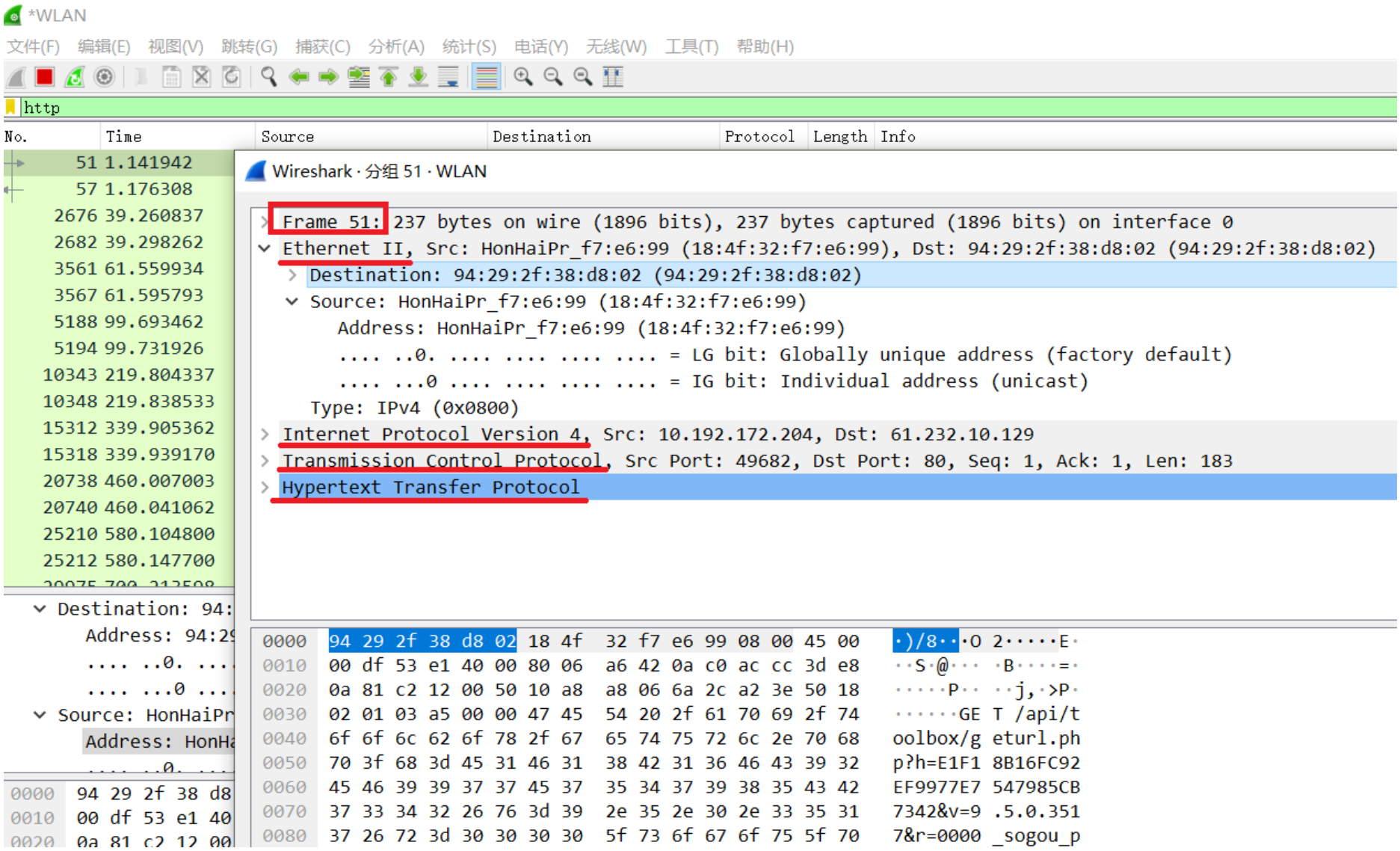
Example II: TCP



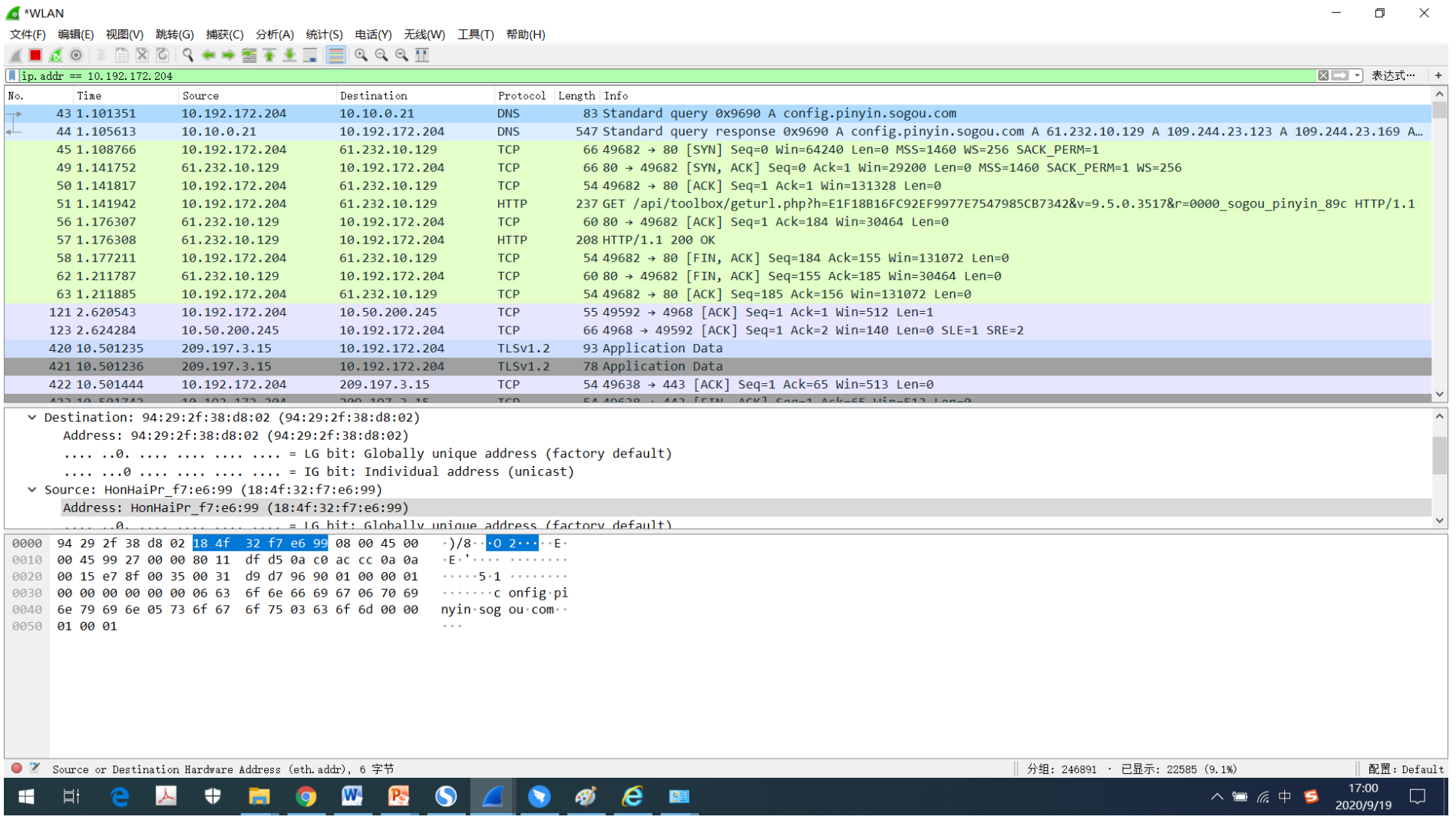
Example III: HTTP



Example III: HTTP



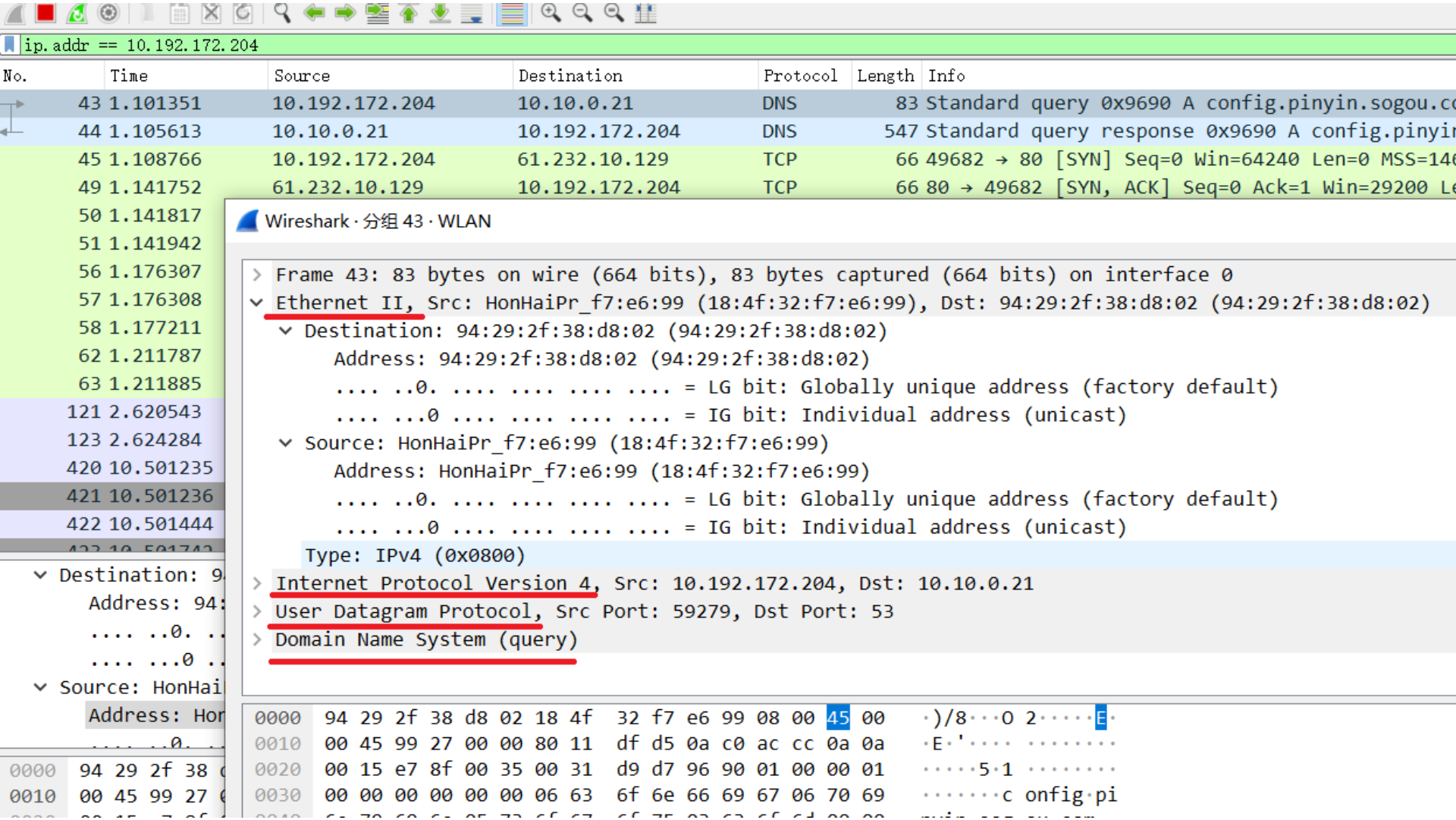
Example IV: ip.addr == 10.192.172.204



ip.addr == x.x.x.x vs. host.addr == x.x.x.x

- 实验中第4部分和第5部分相比，区别在于ip.addr == x.x.x.x是捕获所有数据包，但是只显示与ip地址为x.x.x.x有关的数据包，而host.addr == x.x.x.x只捕获ip地址为x.x.x.x的数据包。检查一下实验结果， host.addr = x.x.x.x命令下抓获数据包量要小很多。
- 注意命令“host.addr = x.x.x.x”已经停用，改为“ip.host = x.x.x.x”
- 或者用这两个命令：ip.src_host == x.x.x.x 只抓数据包中源地址为x.x.x.x的数据包；或 ip.dst_host == x.x.x.x只抓数据包中目标地址为 x.x.x.x的数据包。

Example V: DNS



注意：在DNS数据包中传输层用的协议是UDP，不是TCP协议！ Port number: 53

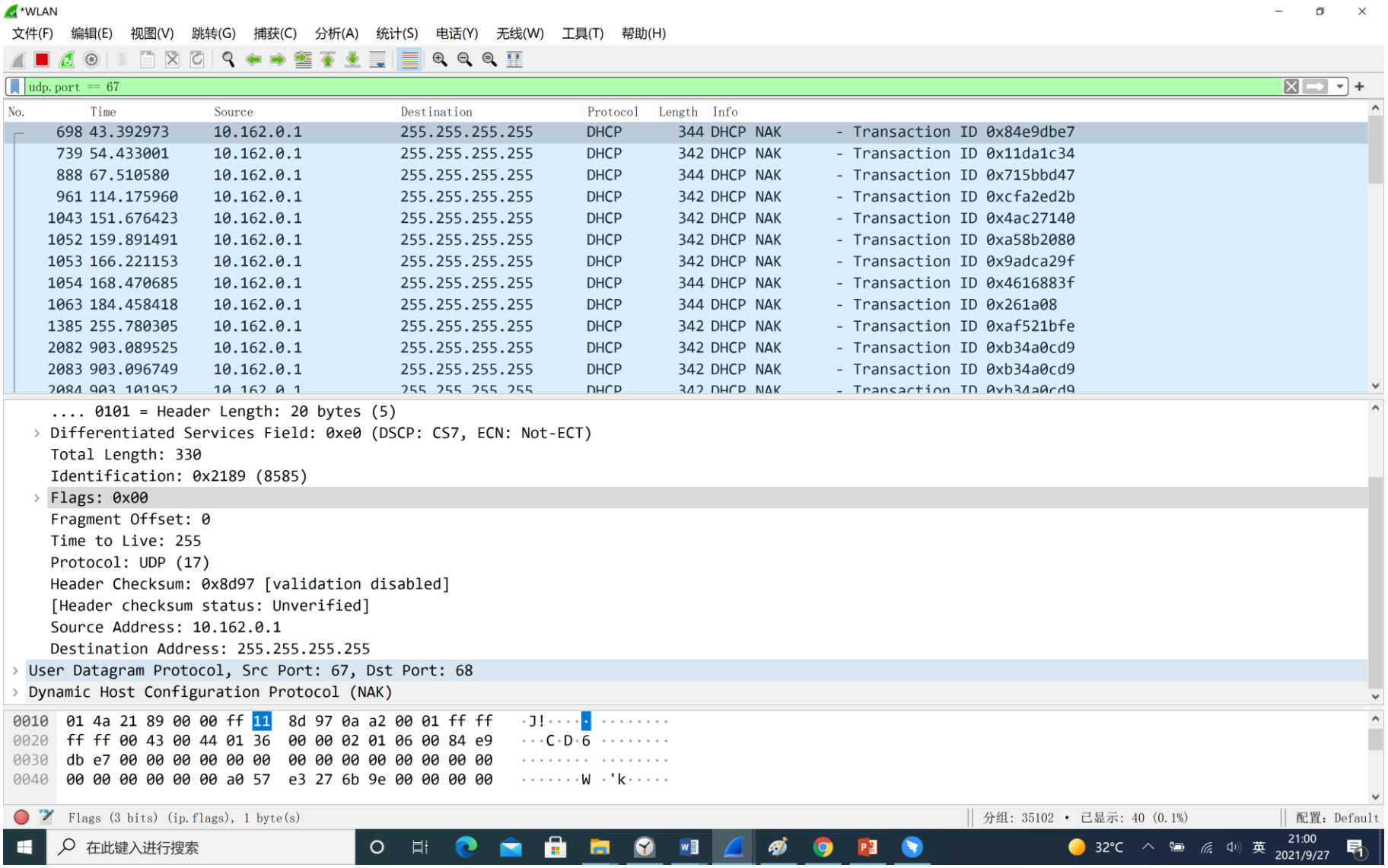
tcp.port == 443 (or 80, or 25)

- 实验1中Part 1第6题变为tcp.port == X 和udp.port == X

TCP 21 = 文件传输
TCP 22 = 远程登录协议
TCP 23 = 远程登录
TCP 25 = 电子邮件 (SMTP)
TCP 80 = http
TCP 110 = 电子邮件(Pop3)
TCP 179 = Border 网关协议 (BGP)
TCP 443 = 网页安全服务(HTTPS)
TCP 546 = DHCP Client
TCP 547 = DHCP Server

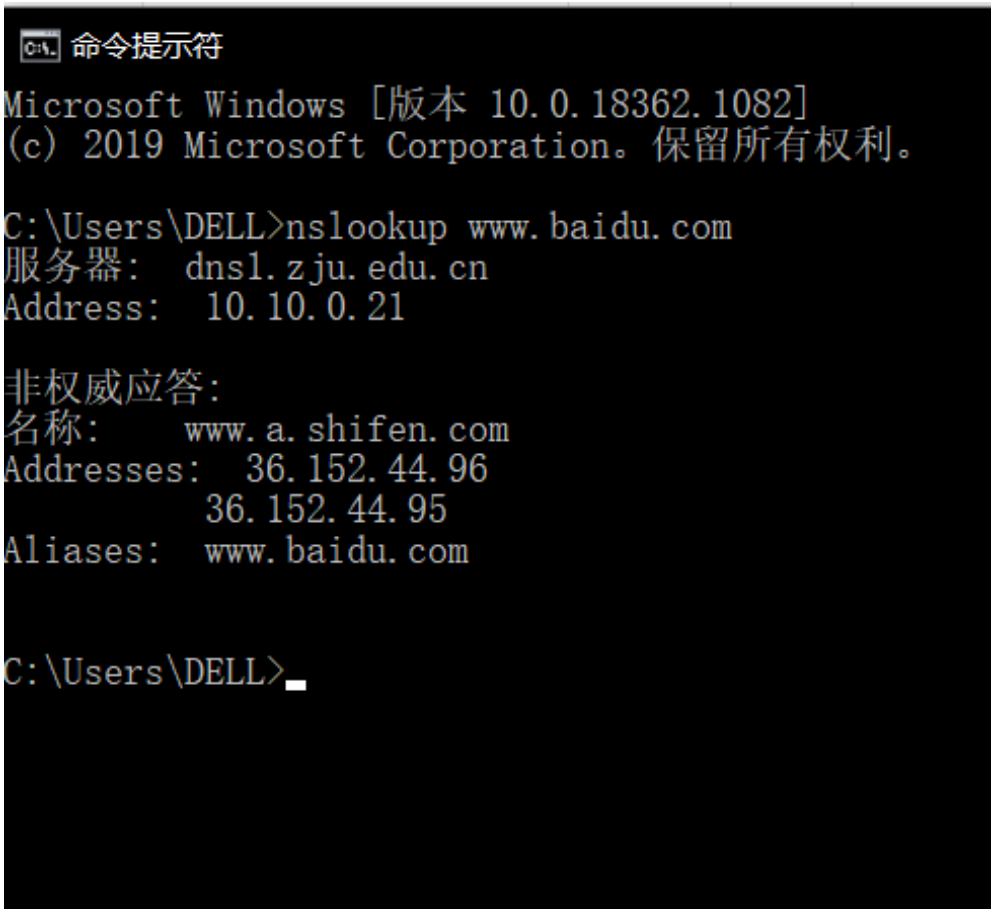
UDP 53 = 域名解析
UDP 67 = 动态IP服务 DHCP
UDP 68 = 客户端向68端口DHCP服务器广播请求地址配置， DHCP服务器向67端口广播回应请求。

“udp.port == 67” Example



Example: nslookup

- 技巧：先退出WireShark，然后重新打开，再运行“nslookup www.baidu.com”在命令行。



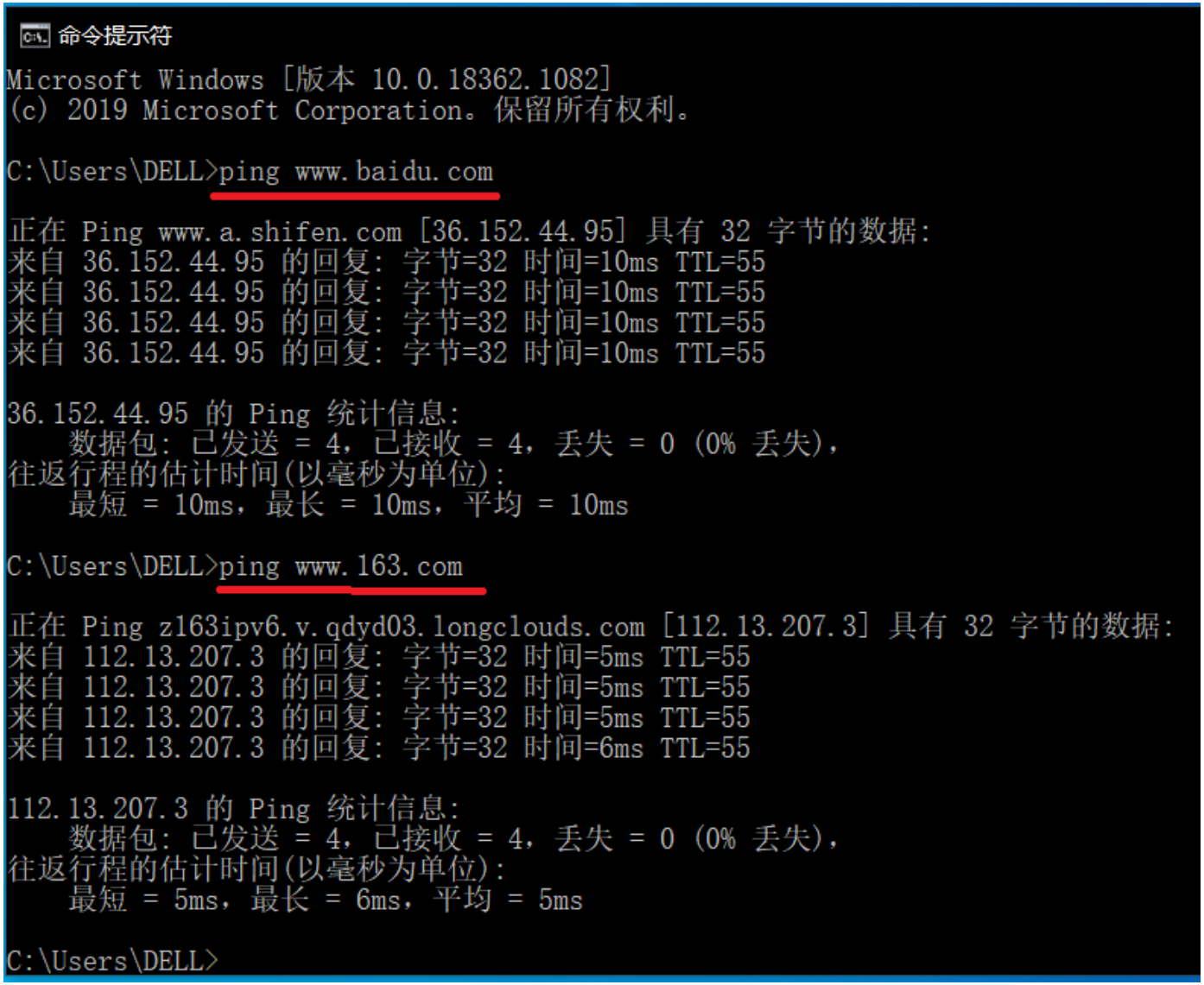
Example: nslookup [4]

- 正向解析：通过域名查找ip；
- 反向解析：通过ip查找域名；
  - IP反向解析主要应用到邮件服务器中来阻拦垃圾邮件，特别是在国外。多数垃圾邮件发送者使用动态分配或者没有注册域名的IP地址来发送垃圾邮件，以逃避追踪，使用了域名反向解析后，就可以大大降低垃圾邮件的数量。
    - 比如你用 xxx@name.com 这个邮箱给我的邮箱 123@163.com 发了一封信。163邮件服务器接到这封信会查看这封信的信头文件，这封信的信头文件会显示这封信是由哪个IP地址发出来的。然后根据这个IP地址进行反向解析，如果反向解析到这个IP所对应的域名是name.com 那么就接受这封邮件，如果反向解析发现这个IP没有对应到name.com，那么就拒绝这封邮件。

Example: nslookup [4]

```
命令提示符
服务器:  dns1.zju.edu.cn
Address:  10.10.0.21

非权威应答:
名称:    www.google.com
Addresses:  2001::1f0d:4808
          0.0.0.0
          127.0.0.1


C:\Users\DELL>nslookup -qt=ptr 36.152.44.96
服务器:  dns1.zju.edu.cn
Address:  10.10.0.21

*** dns1.zju.edu.cn 找不到 96.44.152.36.in-addr.arpa.: Non-existent domain

C:\Users\DELL>nslookup -qt=mx www.zju.edu.cn
服务器:  dns1.zju.edu.cn
Address:  10.10.0.21

zju.edu.cn
        primary name server = dns1.zju.edu.cn
        responsible mail addr = root.zju.edu.cn
        serial  = 2016112807
        refresh = 10800 (3 hours)
        retry   = 3600 (1 hour)
        expire  = 604800 (7 days)
        default TTL = 30 (30 secs)

C:\Users\DELL>
```

Example: ping (ICMP, Internet Control Message Protocol)



Internet Control Message Protocol (**ICMP**) [2]

- ICMP is specified in RFC 792.
- The most typical use of ICMP is for **error reporting**.
  - For example, when running a Telnet, FTP, or HTTP session, you may have encountered an error message such as "Destination network unreachable".
- ICMP is often considered part of IP but architecturally it lies just above IP, as ICMP messages are carried inside IP datagrams.
- ICMP messages have a type and a code field, and contain the header and the first 8 bytes of the IP datagram.

ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

Figure 4.23 ♦ ICMP message types

Example: Tracert (ICMP) (I)

- The **Tracert** program, which allows us to trace a route from a host to any other host in the world.
- Tracert is implemented with ICMP messages, to determine the names and addresses of the routers between source and destination,
  - 1) Tracert in the source sends *a series of ordinary IP datagrams* to the destination.
  - Each of these datagrams carries a **UDP** segment with an unlikely UDP port number.
  - The 1st of these datagrams has a TTL of 1, the 2nd of 2, the 3rd of 3, and so on. The source also starts timers for each of the datagrams.

Example: Tracert (ICMP) (II)

- Tracert is implemented with ICMP messages, to determine the names and addresses of the routers between source and destination,
  - 2) When the *n*th datagram arrives at the *n*th router, the *n*th router observes that *the TTL of the datagram has just expired*.
  - According to the rules of the IP protocol, the router discards the datagram and sends an ICMP warning message to the source (type 11 code 0)
  - This warning message includes the name of the router and its IP address.
  - 3) When this ICMP message arrives back at the source, the source obtains **the round-trip time** from the timer and the name and IP address of the *n*th router from the ICMP message

Example: Tracert (ICMP) (III)

- Tracert is implemented with ICMP messages, to determine the names and addresses of the routers between source and destination,
  - 4) How does a Tracert source know when to **stop** sending UDP segments?
  - Recall that the source increments the TTL field for each datagram it sends. Thus, one of the datagrams will eventually make it all the way to the destination host.
  - Because this datagram contains a UDP segment with an unlikely port number, the destination host sends **a port unreachable ICMP message** (type 3 code 3) back to the source.
  - When the source host receives this particular ICMP message, it knows it does not need to send additional probe packets.
  - The standard Tracert program actually sends sets of three packets with the same TTL; thus the Tracert output provides three results for each TTL.

References

- [1] https://www.wireshark.org/
- [2] J. F. Kurose and K.W. Ross, Computer Networking — A Top-down Approach, 5th Edition, Pearson Education Inc., 2010.
- [3] https://blog.csdn.net/gui951753/article/details/83070180 (这个博客中有解释多个站点对应一个IP地址的问题。)
- [4] https://www.cnblogs.com/machangwei-8/p/10353137.html