

# Undergraduate Lab Report of Jinan University

Course Title Computer Networks Lab Evaluation

Lab Name \_\_\_\_\_ IPv4 Address and Datagram

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 / 3

## 1. Introduction

### 1) Objective

- To learn about the details of IP (Internet Protocol).
- Know the format of IP packet.
- Know how to calculate IP checksum.

## 2) Experiment Principle

## 1. The Introduction of Internet Protocol

The Internet's backbone relies on a protocol called IP (Internet Protocol), which was designed with the primary goal of connecting different networks. I imagine the network layer's main job is to move data packets from one place to another, whether the devices are on the same network or across multiple networks.

Typically, the link layer encapsulates IP packets:

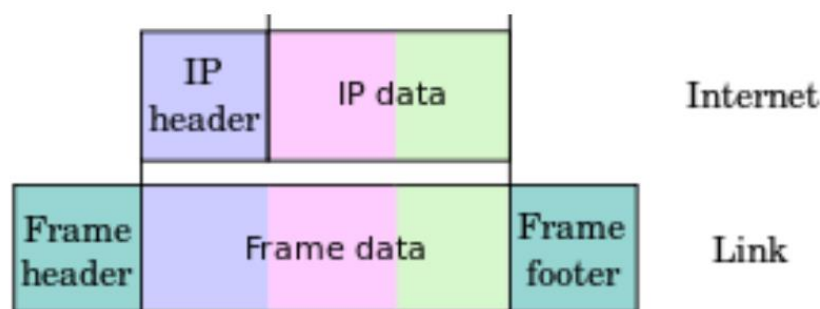


Figure1: Encapsulated IP packets

The transport layer breaks down data into manageable chunks that can be sent as IP packets. Although these packets can be as large as 64KB in theory, they usually stay under 1,500 bytes to fit within a single Ethernet frame. IP routers guide these packets through the Internet, passing them from one router to the next until they reach their destination. When the packets arrive, the network layer hands them off to the transport layer, which then forwards them to the intended recipient process. After all

the fragments have been gathered by the destination computer, the network layer is responsible for piecing them back together into the original datagram, which is then sent up to the transport layer.

## 2. The format of IP packet

The format of the IP packet header is crucial for ensuring that data is routed and handled accurately on the Internet. This header includes several fields that are vital for this process: it specifies the version of the protocol, the length of the header, the quality of service indicators, the maximum time a packet can exist in the network (its "time to live"), mechanisms for detecting errors, and the addresses of both the sender and the receiver. There is also room for additional options when needed for specific situations.

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP				ECN				Total Length															
4	32	Identification																Flags		Fragment Offset													
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															

Figure2: IP Header format

## 3. IP Address

A defining feature of IPv4 is its use of 32-bit address space. Each device and router on the Internet is assigned an IP address, which is included in the Source and Destination fields of IP packets. It's important to note that an IP address points to a network interface, not the host itself. This means that a host with multiple network interfaces will have multiple IP addresses. Routers, which often have multiple interfaces, typically have several IP addresses as a result.

The IPv4 address space is divided into five classes: A, B, C, D, and E. Classes A, B, and C are used for unicast addresses, which are addresses intended for a single recipient. Class D is set aside for multicast addresses, used for sending data to multiple recipients simultaneously. Class E is reserved for experimental or future use. This method of dividing the address space is known as classful addressing.

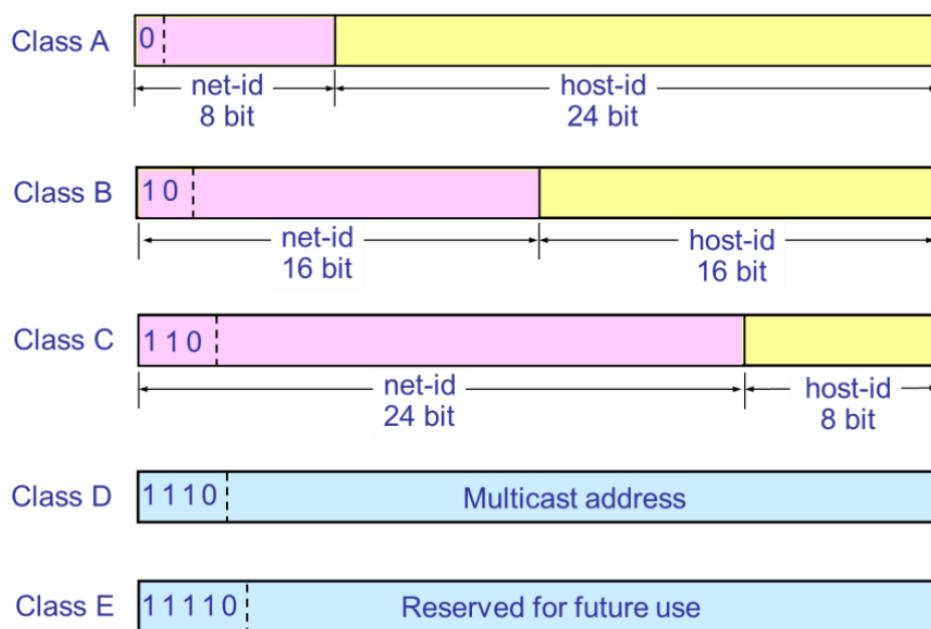


Figure 3: Five different classes in IP address space

Classes	Max. # of networks	First network ID	Last network ID	Max. # of hosts in each network
A	126 ( $2^7 - 2$ )	1	126	16,777,214
B	16,383 ( $2^{14}$ )	128.0	191.255	65,534
C	2,097,151 ( $2^{21}$ )	192.0.0	223.255.255	254

Figure4: IP address range of three common classes:

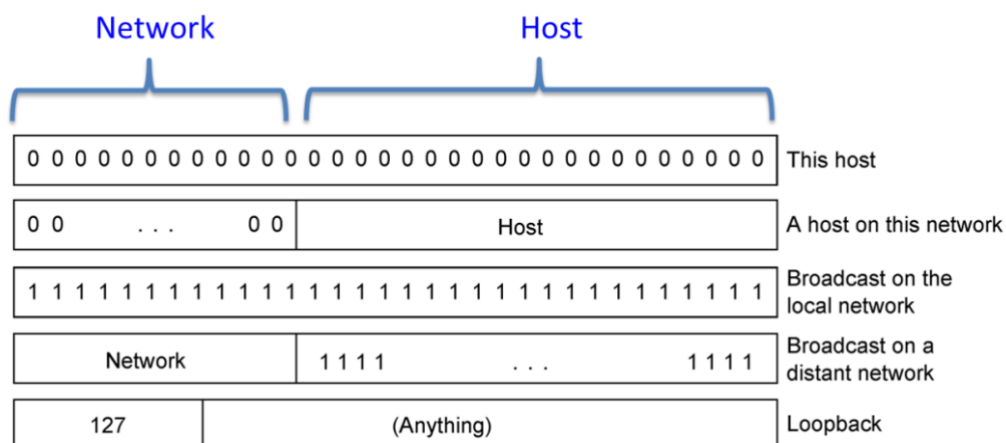


Figure5: Several Special IP Addresses

Brief explanation of these special IP addresses is as follows to help us understand more about the details:

- **Host Identifier:** An IP address consisting entirely of zeros (0.0.0.0) is used to denote the host itself.
- **Network Identifier:** IP addresses with zeros in the host portion are not assigned to any specific host; instead, they represent the network. For instance,

ce, the host '202.198.151.136' belongs to the subnet '202.198.151.0'.

- **Broadcast Address:** IP addresses with ones in the host portion are not assigned to hosts and serve as broadcast addresses. Sending a packet to a broadcast address will deliver it to all devices on that network, provided the network supports broadcasting. For example, the broadcast address for the network '202.198.151.136' is '202.198.151.255'.
- **Limited Broadcast:** The IP address with all ones (255.255.255.255) is typically used by diskless workstations to request an IP address from an IP address server during the boot process.
- **Loopback Address:** The address '127.0.0.1' is utilized for testing software or communicating with various network applications on the same machine.

#### Private IP addresses

Three ranges of IPv4 addresses for private networks are not routed on the Internet and thus their use need not be coordinated with an IP address registry.

	Start	End	No. of addresses
24-bit block (/8 prefix, $1 \times A$ )	10.0.0.0	10.255.255.255	16777216
20-bit block (/12 prefix, $16 \times B$ )	172.16.0.0	172.31.255.255	1048576
16-bit block (/16 prefix, $256 \times C$ )	192.168.0.0	192.168.255.255	65536

Figure 6: Private IP addresses

## 4. Lab Environment

I accomplish this lab at home using my own Hotspot, also I disabled IPv6 to avoid some possible effect.



Figure7: WLAN properties

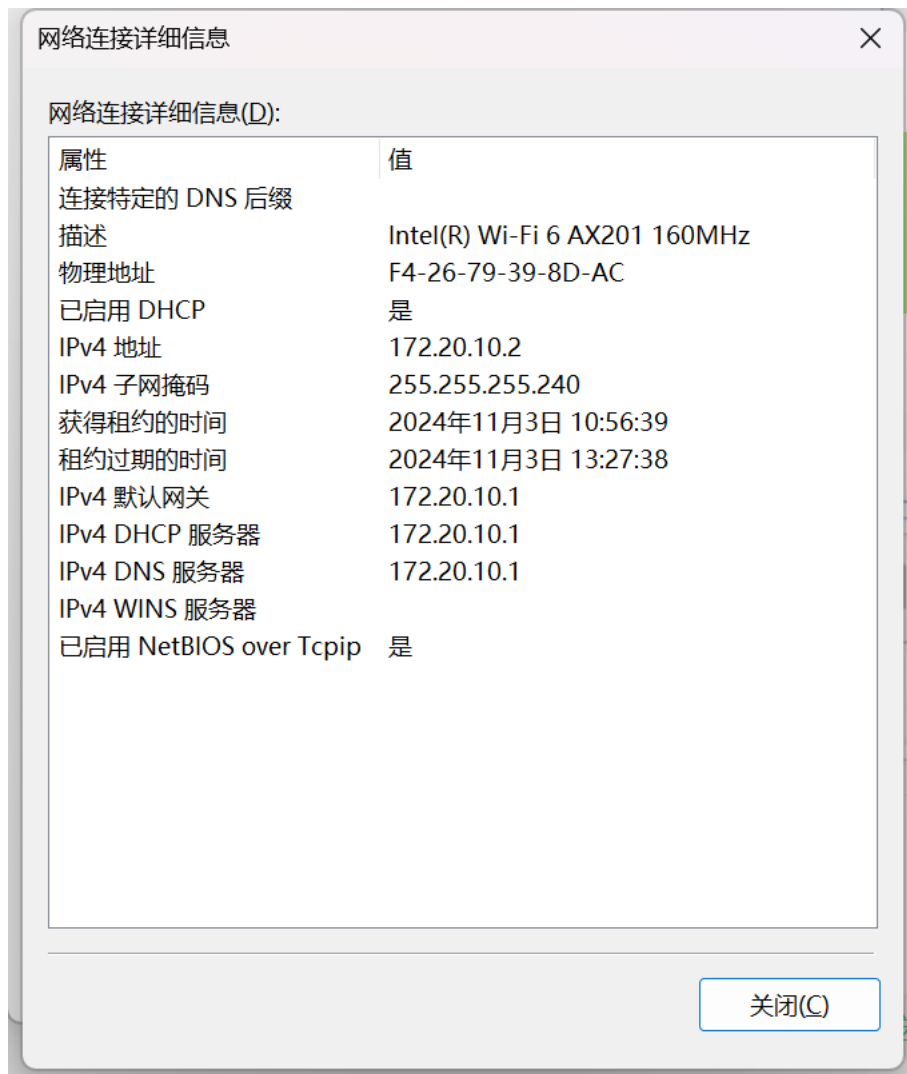


Figure8: Detailed information of Network

### Computer Name:

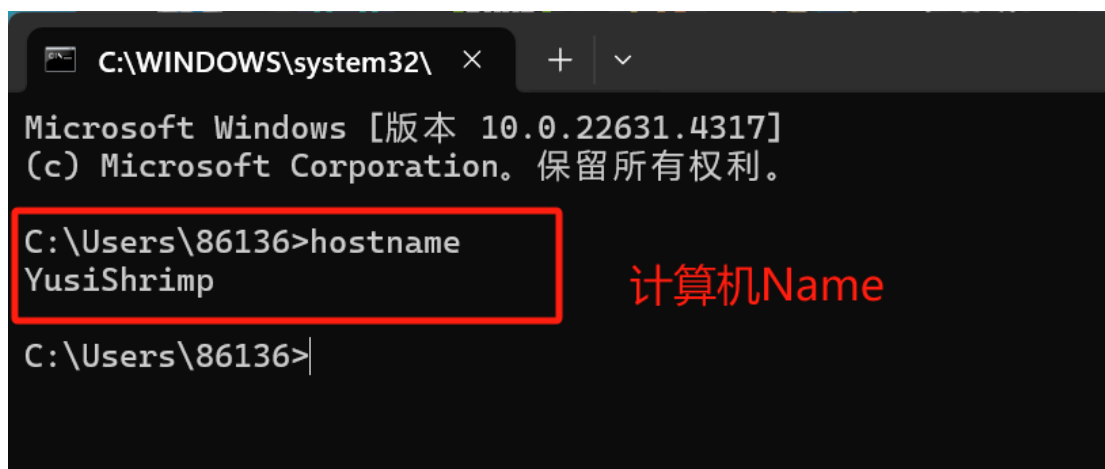


Figure9: Host name

## 5. Experiment Steps and Results

### 1) Task1: Capture a Trace

#### 1. Experiment Steps

- [1] Step 1: Pick a URL at a remote server, e.g., `http://www.jnu.edu.cn/` and fetch the contents with `wget` or `curl`, e.g., `"wget http://www.jnu.edu.cn/"` or `"curl http://www.jnu.edu.cn/"`.
- This will fetch the resource and either write it to a file (`wget`) or to the screen (`curl`). With `wget`, you expect a single response with status code "200 OK".
  - If the fetch does not work, try a different URL.
- [2] Step 2: Perform a traceroute to the same remote server to check that you can discover information about the network path. On Windows, type, e.g., `"tracert www.jnu.edu.cn"`. On Linux / Mac, type, e.g., `"traceroute www.jnu.edu.cn"`. A successful example is shown below; save the output as you will need it for later steps.
- On Linux / Mac and behind a NAT (as most home users or virtual machine users), we can use the `"-I"` option (that was a capital i) to traceroute, e.g., `"traceroute -I www.jnu.edu.cn"`. This will cause traceroute to send ICMP probes like `tracert` instead of its usual UDP probes.
  - traceroute may take up to a minute to run. Each line in the output shows information about the next IP hop from the computer running traceroute towards the target destination.
  - The lines with `"*"` indicate that there was no response from the network to identify that segment of the Internet path. Some unidentified segments are to be expected. However, if traceroute is not working correctly then nearly all the path will be `"*"`. In this case, try a different remote server, or use the supplied traces on the course website.
- [3] Step 3: Launch Wireshark and start a capture with a filter of `"tcp port 80"`. Make sure to check `"Resolve network names"`.

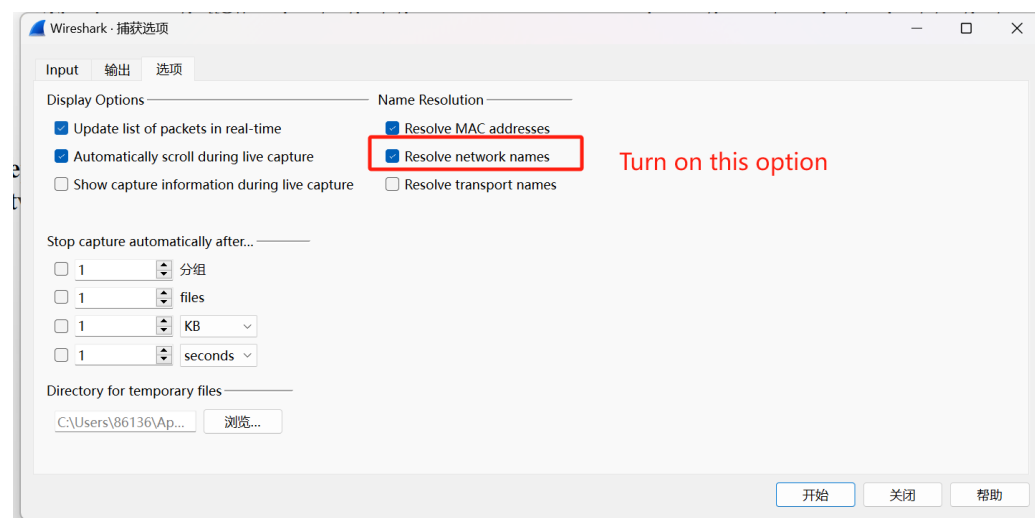


Figure10: Some related Capture options



Figure11: Set the corresponding filter

- We use the filter to record only standard web traffic.
  - Name resolution will translate the IP addresses of the computers sending and receiving packets into names. It will help you to recognize whether the packets are going to or from your computer.
  -
- [4] Step 4: After the capture is started, repeat the wget/curl command above. This time, the packets will also be recorded by Wireshark.
- [5] Step 5: After the command is complete, return to Wireshark and stop the trace. We now have a short trace like that shown in the figure below, along with the output of a traceroute we ran earlier to the corresponding server.

## 2. Results

```
C:\WINDOWS\system32\ x + v
Microsoft Windows [版本 10.0.22631.4317]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\86136>wget http://www.jnu.edu.cn/
--2024-11-03 17:31:35-- http://www.jnu.edu.cn/
Resolving www.jnu.edu.cn (www.jnu.edu.cn)... 61.164.126.124
Connecting to www.jnu.edu.cn (www.jnu.edu.cn)[61.164.126.124]:80... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://www.jnu.edu.cn/ [following]
--2024-11-03 17:31:36-- https://www.jnu.edu.cn/
Connecting to www.jnu.edu.cn (www.jnu.edu.cn)[61.164.126.124]:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: /main.htm [following]
--2024-11-03 17:31:36-- https://www.jnu.edu.cn/main.htm
Reusing existing connection to www.jnu.edu.cn:443.
HTTP request sent, awaiting response... 200 OK
Length: 169572 (166K) [text/html]
Saving to: 'index.html.2'

index.html.2          100%[=====>] 165.60K  310KB/s   in 0.5s

2024-11-03 17:31:37 (310 KB/s) - 'index.html.2' saved [169572/169572]

C:\Users\86136>
```

Figure12: Running wget command



```
C:\WINDOWS\system32\ > + v
Microsoft Windows [版本 10.0.22631.4317]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\86136>tracert -4 www.jnu.edu.cn

通过最多 30 个跃点跟踪
到 waftnc.jnu.edu.cn.ctdns.cn [61.164.126.124] 的路由:

 1    2 ms    2 ms    2 ms    172.20.10.1
 2    *      *      *      请求超时。
 3    *      19 ms   32 ms   192.168.188.33
 4    *      *      *      请求超时。
 5    *      *      *      请求超时。
 6    *      *      *      请求超时。
 7   346 ms   91 ms   37 ms   125.88.77.65
 8    *      *      *      请求超时。
 9    52 ms   74 ms   74 ms   220.186.222.74
10    *      *      *      请求超时。
11    *      *      *      请求超时。
12   68 ms   65 ms   56 ms   61.164.126.124

跟踪完成。

C:\Users\86136>
```

Figure13: Running traceroute(as tracert on Windows)

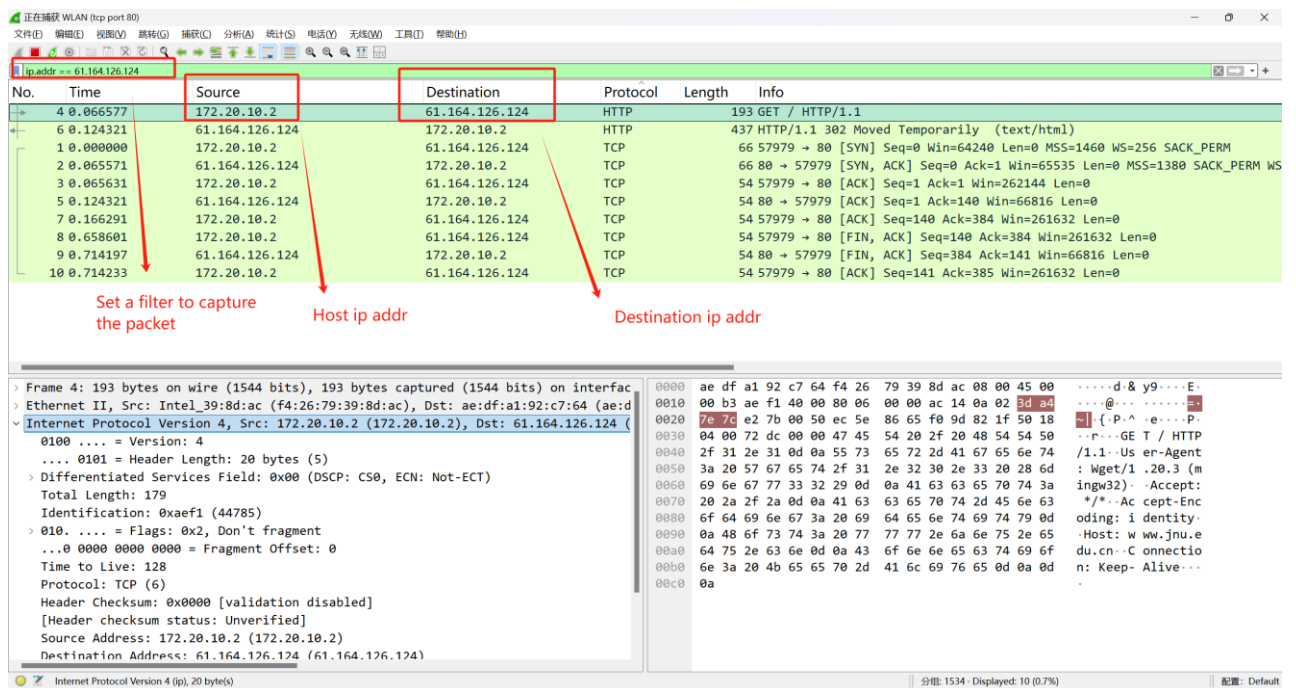


Figure14: Trace of wget/curl traffic showing the details of the IP header

## 2) Task2: Inspect the Trace and IP Packet Structure

### 1. Experiment Steps

- [1] Step 1: Select any packet in the trace (in the top panel) to see details of its structure (in the middle panel) and the bytes that make up the packet (in the bottom panel).
- [2] Step 2: In the middle panel, expand the IP header fields, and check each of these fields.
  - Our interest is the IP header, we can ignore the other higher and lower layer protocols (which are TCP and HTTP in this case).
  - We can click on the IP header to see the bytes that correspond to it in the bottom panel.

## 2. Results

Proceed to utilize the data packet trace obtained during the communication session with 'www.jnu.edu.cn' as mentioned in Task 1.

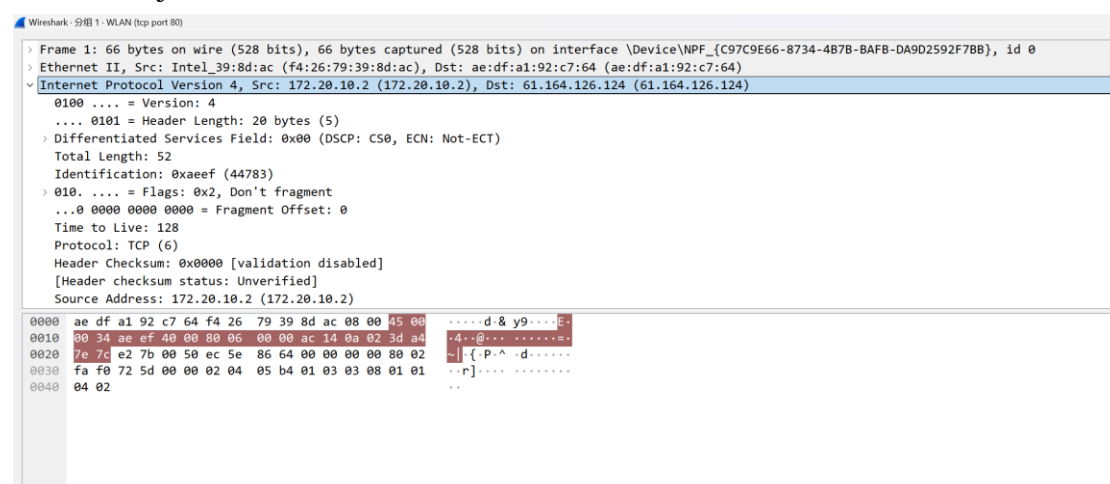


Figure15: Detailed information of IP packet

## 3. Answering the questions

Q1: What are the IP addresses of your computer and the remote server?

A1: My computer: 172.20.10.2

Remote server: 61.164.126.124

Q2: Does the Total Length field include the IP header plus IP payload, or just the IP payload?

A2: The Total Length field in an IP (Internet Protocol) header refers to the entire size of the IP datagram, which includes both the IP header and the IP payload.

Q3: How does the value of the Identification field change or stay the same for different packets? For instance, does it hold the same value for all packets in a TCP connection or does it differ for each packet? Is it the same in both directions? Can you see any pattern if the value does change?

A3: The Identification field in an IP header is used to identify the packets sent by a host, ensuring that packets are not confused with each other during transmission. The Identification field typically varies among different packets, each having a unique value to ensure proper identification and handling of packets. This variation applies not only to packets within the same TCP connection but also to both directions of the connection.

Q4: What is the initial value of the TTL field for packets sent from your computer? Is it the maximum possible value, or some lower value?

A4: The initial value of the Time to Live (TTL) field for packets sent from a computer is not a fixed value and can vary depending on the operating system and its configuration. However, it is typically set to a default value that is intended to ensure that the packet does not circulate indefinitely on the internet.

1. **Windows Systems:** For most Windows systems, the default TTL value is **128**.

2. **Linux Systems:** For Linux systems, the default TTL value is often **64**.

3. **Mac OS Systems:** For Mac OS systems, the default TTL value is typically **64**.

It's important to note that these are default values and can be changed by system administrators or through specific network configurations. The TTL field is decremented by 1 at each router or hop that the packet passes through. Once the TTL value reaches 0, the packet is discarded, and an Internet Control Message Protocol (ICMP) "Time Exceeded" message is sent back to the sender to indicate that the packet has expired.

So, while it's not the maximum possible value (which would be 255 for a one-byte field), the initial TTL is set to a value that is lower than the maximum but high enough to allow the packet to reach its destination without expiring prematurely under normal network conditions.

Q5: What is the length of the IP Header and how is this encoded in the header length field?

A5: The Internet Header Length (IHL) field specifies the length of the header in 32-bit words. Since each 32-bit word is equivalent to 4 bytes, the IHL field effectively indicates the number of 32-bit words multiplied by 4 to get the total header length in bytes. The IHL can range from 5 to 15, which means the header can be anywhere from 20 to 60 bytes long.

### 3) Task 3: Internet Paths

#### 1. Experiments Steps

- [1] Step 1: Label the IP addresses for all nodes on the path.
- [2] Step 2: Number each router by their distance on hops from the start of the path.
- [3] Step 3: If possible, try to label the routers along the path with the name of the real-world organization to which they belong.
  - To do this, we will need to interpret the domain names of the routers given by traceroute.
  - If unsure, label the routers with the domain name to be the organization. Ignore routers for which there is no domain name (or no IP address).
- [4] Step 4: Ensure the traceroute output is included in the report, e.g., a figure.

## 2. Results

```

C:\WINDOWS\system32\ x + v
Microsoft Windows [版本 10.0.22631.4317]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\86136>tracert -4 www.jnu.edu.cn

通过最多 30 个跃点跟踪
到 waftnc.jnu.edu.cn.ctdns.cn [61.164.126.124] 的路由:

 1    2 ms    2 ms    2 ms    172.20.10.1
 2    *      *      *      请求超时。
 3    *      19 ms   32 ms   192.168.188.33
 4    *      *      *      请求超时。
 5    *      *      *      请求超时。
 6    *      *      *      请求超时。
 7   346 ms   91 ms   37 ms   125.88.77.65
 8    *      *      *      请求超时。
 9    52 ms   74 ms   74 ms   220.186.222.74
10    *      *      *      请求超时。
11    *      *      *      请求超时。
12   68 ms   65 ms   56 ms   61.164.126.124

跟踪完成。

C:\Users\86136>

```

Figure16: Tracert command windows

The result obtained by tracert is that the network arrived from my computer through multiple routers the website I want to access, so the schematic diagram is as follows:

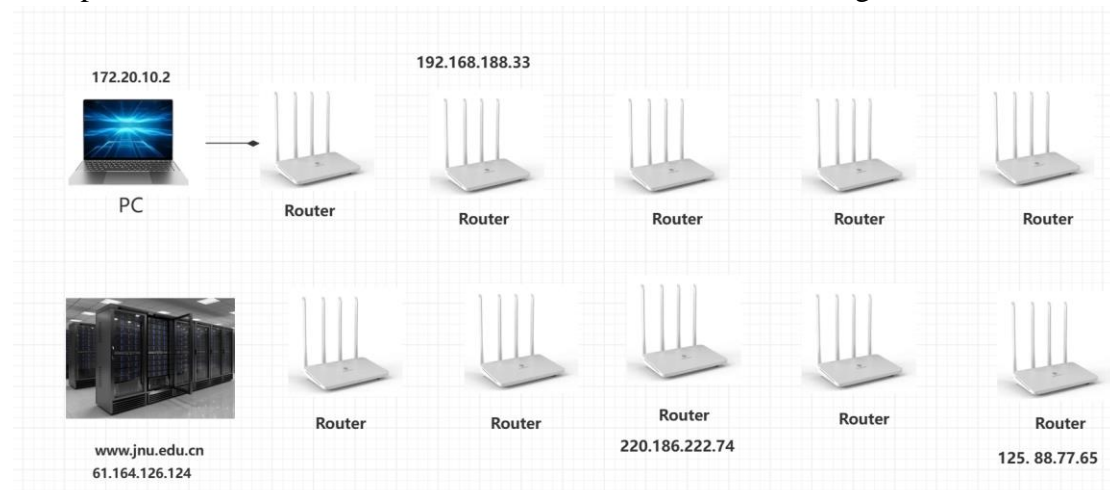


Figure17: Network path

#### 4) Task 4: IP Header Checksum

##### 1. Experiments Steps

- [1] Step 1: From the Wireshark trace, pick a packet sent from the remote server to computer, and ensure that it have a non-zero value in the checksum field.
- The checksum value sent over the network will be non-zero, so if the packet has a zero value it is because of the capture setup.
  - To make this exercise easier, try a packet that has an IP header of 20 bytes, which is the minimum header size when there are no options.
- [2] Step 2: Follow these sub-steps to check that the checksum value is correct:
- i. Break down the header into 10 sets of 16-bit segments, with each segment represented by 4 hexadecimal digits in the packet data panel at the bottom of the Wireshark window.
  - ii. Add the 10 words using regular addition.
  - iii. To compute the 1s complement sum from your addition so far, take any leading digits (beyond the 4 digits of the word size) and add them back to the remainder.
  - iv. The result should be 0xffff. This is zero in 1s complement form, or more precisely 0xffff is -0 (negative zero) while 0x0000 is +0 (positive zero).
  - v. In your calculation, next to each word, please add notes of the IPv4 fields to which it corresponds.

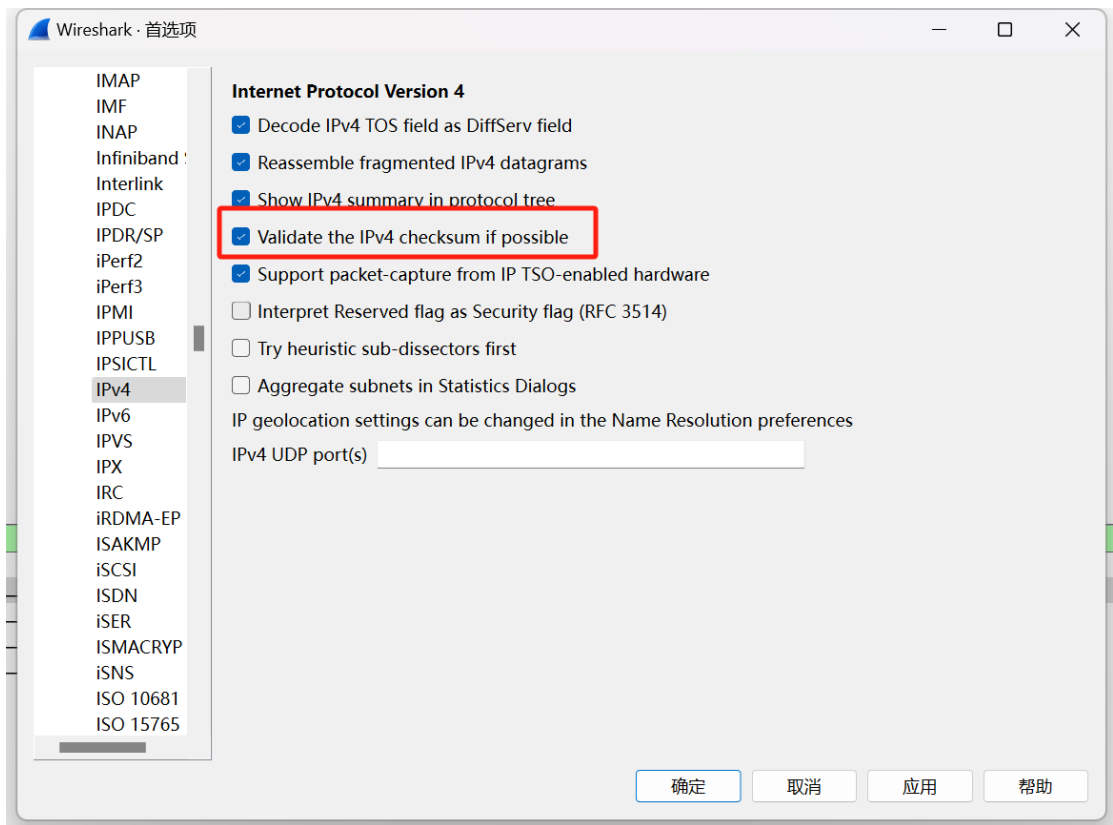


Figure18: Enabling IPv4 checksum computation

## 2. Results

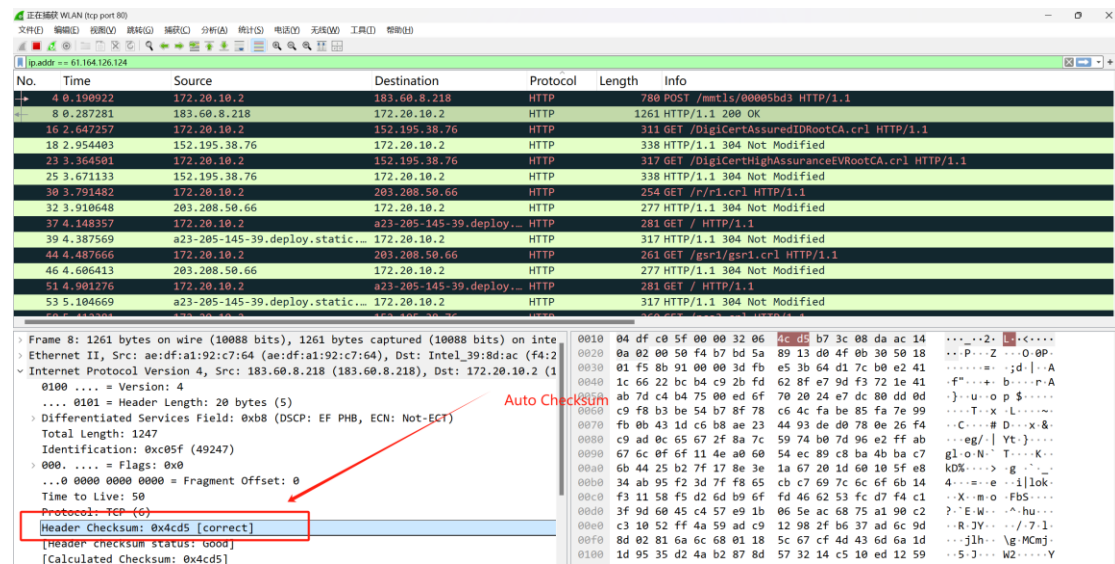
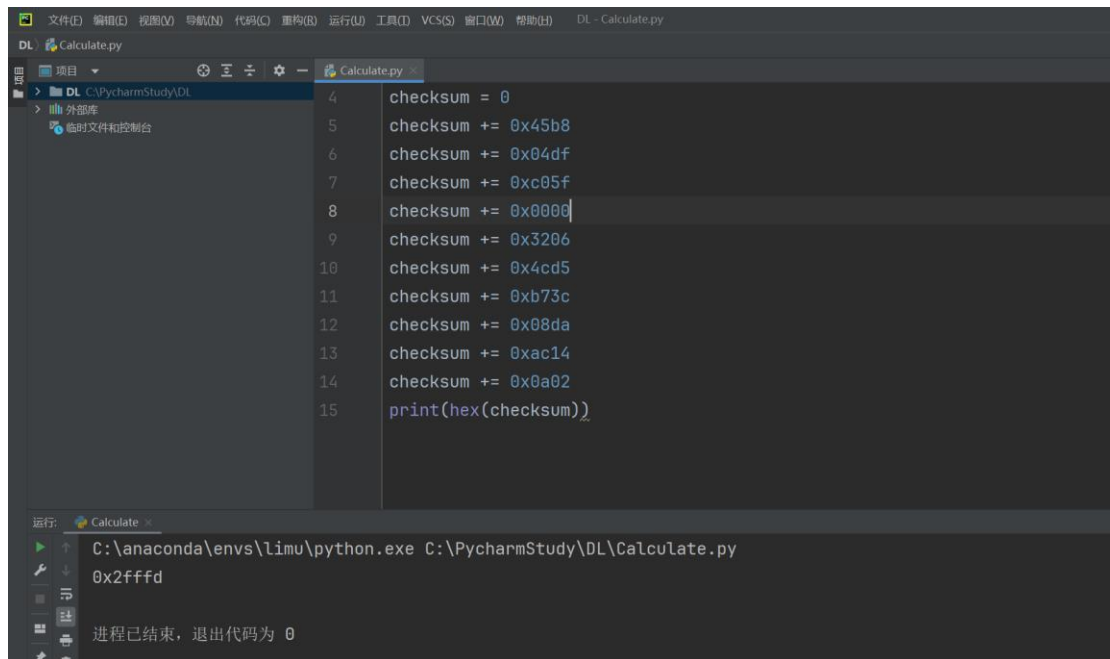


Figure19: Auto checksum



```
4 checksum = 0
5 checksum += 0x45b8
6 checksum += 0x04df
7 checksum += 0xc05f
8 checksum += 0x0000
9 checksum += 0x3206
10 checksum += 0x4cd5
11 checksum += 0xb73c
12 checksum += 0x08da
13 checksum += 0xac14
14 checksum += 0xa02
15 print(hex(checksum))
```

运行: Calculate

C:\anaconda\envs\limu\python.exe C:\PycharmStudy\DL\Calculate.py

0x2ffffd

进程已结束，退出代码为 0

Figure20. Checksum

### 3. Answering the questions

Q1: Which fields should be included when calculating the checksum of IP packet?

A1: It's important to note that the checksum only covers the IP header and not the data portion. This is because the header may change at each router (due to fields like TTL, Flags, and Fragment Offset potentially changing), and recalculating the checksum at each hop is necessary. Omitting the data portion from the checksum calculation reduces the computational workload.

### 5) Task 5: Explore on your own(Optional)



I have tried many ways to solve this problem, but there are always errors. I will do it again next time when I come to the lab and ask you to solve it

## 6. Others

### 1. Answering the questions:

Q1: Explain the difference between MAC and IP addresses in a network? Why should we use them?

A1: MAC addresses, which reside at the Data Link Layer, are inherently assigned to network interfaces and guarantee distinct identification within a local area network. They are essential for direct device interaction within a confined network environment.

Conversely, IP addresses operate at the Network Layer and serve as a logical identifier that may vary depending on the network or geographical location. They are crucial for directing data packets across various networks, enabling global communication between devices, including over the internet.

To encapsulate, MAC addresses are fundamental for local network interactions, whereas IP addresses provide extensive connectivity that spans across multiple networks.

Q2: What fields will be changed when an IP packet is forwarded by a router?

A2: As an IP packet journeys through a router, certain modifications are applied to its header to ensure proper routing and maintain packet integrity. Chief among these is the reduction of the TTL (Time to Live) value, which prevents the packet from circulating indefinitely within the network. Concurrently with this, the Header Checksum must be recalculated due to the change in the TTL value. Furthermore, in scenarios that involve Network Address Translation (NAT), there may be alterations to both the Source and Destination IP Addresses.

Q3: The checksum in IP header doesn't verify the data of IP payload. What are the advantages and disadvantages of such scheme?

A3: **Benefits:** A key benefit of this method is the improved speed it provides. By confining the verification process to the header, routers can more swiftly process and forward packets, which enhances network performance. This strategy also preserves the integrity of the layered network architecture. The IP layer concentrates on routing, while leaving the payload error-checking to the transport layer protocols like TCP or UDP, ensuring a clear division of duties. This separation of responsibilities also increases flexibility, as many transport layer protocols have their own error-checking mechanisms, avoiding unnecessary redundancy.

**Drawbacks:** However, not including the payload in the checksum calculation means that any data corruption might go unnoticed at the IP layer. This can result in wasted bandwidth, as corrupted packets might travel through several routers before being detected and addressed at their final destination.



Q4: What is the scope of limited broadcast address? Does a router forward limited broadcast packets?

A4: The specific broadcast address, denoted as 255.255.255.255 in IPv4, is designed to send a packet to every host on the sender's immediate local network segment. Its reach is strictly confined to the local level. Routers are programmed not to propagate these limited broadcast packets to prevent them from overwhelming the entire network, which could result in excessive traffic and potential network congestion. By keeping the limited broadcast address within the local subnet, it ensures that only the devices on that specific segment are the ones to receive and handle the broadcast message.

Q5: What is difference between limited broadcast address and direct broadcast address?

A5: The specific broadcast address, denoted as 255.255.255.255 in IPv4, is designed to send a packet to every host on the sender's immediate local network segment. Its reach is strictly confined to the local level. Routers are programmed not to propagate these limited broadcast packets to prevent them from overwhelming the entire network, which could result in excessive traffic and potential network congestion. By keeping the limited broadcast address within the local subnet, it ensures that only the devices on that specific segment are the ones to receive and handle the broadcast message.

Q6: Suppose there are two computers connected to the same LAN: A and B. Can computer A receive an IP packet which is sent by computer B with the destination of 127.0.0.1? Explain why?

A6: Computer A is incapable of receiving an IP packet from Computer B that is destined for 127.0.0.1. This address is recognized as the loopback address, which directs any packets sent to it for internal processing within the device that sent them, without ever being sent out through the physical network interface. Consequently, such packets do not propagate across the local area network. The loopback address is chiefly employed for internal testing and for facilitating communication between different software processes residing on the same computer. Therefore, when Computer B dispatches a packet to 127.0.0.1, the packet remains confined within Computer B for processing and is not disseminated or perceivable to any other networked devices, including Computer A.