# 华东师范大学软件工程学院实验报告

| | | |
|---|---|---|
| **实验课程**：计算机网络实践 | **年级**：2023 级 | **实验成绩**： |
| **实验名称**：IPV4 | **姓名**：顾翌炜 | |
| **实验编号**：Lab-3 | **学号**：10235101527 | **实验日期**：2024/12/06 |
| **指导教师**：王廷 | **组号**：01 | **实验时间**：2024/12/06 |

# 1 实验目的

1) 学会通过 `Wireshark` 分析 `IP` 协议
2) 了解 `IP` 数据报的组成
3) 了解 `IP` 各部分的含义

# 2 实验内容和实验步骤

## 2.1 实验内容

### 2.1.1 数据捕获

1. 启动 Wireshark 应用程序。
2. 通过菜单栏选择“捕获”→“选项”进行设置。
3. 选择已连接的以太网接口。
4. 设置捕获过滤器为 `tcp port 80` 以捕获通过 HTTP 端口的数据包。
5. 在命令行界面使用 `wget` 命令向 `http://www.baidu.com` 发送 HTTP 请求。
6. 打开 Wireshark 的捕获窗口，并在适当时候停止捕获过程。

### 2.1.2 分析 IPv4 包结构

根据对 IP 报文的理解，绘制出 IP 报文的结构图。在图中明确标注 IP 报头字段的位置及其大小（以字节为单位）。

### 2.1.3 数据包分析

对捕获到的 IP 数据包进行详细分析，关注以下字段的含义：

1. **版本号 (Version)**: 占用 4 位。用于指示当前使用的 IP 协议版本，常见值为 0100（代表 IPv4），0110（代表 IPv6）。
2. **首部长度 (Header Length)**: 占用 4 位。用于说明 IP 报头的总长度，考虑到 IP 报头中存在可变长度的选项字段。
3. **区分服务 (Differentiated Services)**: 占用 8 位。用于为不同的 IP 数据包指定不同的服务质量。
4. **总长度 (Total Length)**: 占用 16 位。表示 IP 包的总大小（包括头部和数据），单位为字节，最大值为 65535 字节。有效载荷的大小计算公式为：IP 包总长度 (Total Length) 减去 IP 报头长度 (Header Length)。
5. **标识 (Identifier)**: 长度 16 位。与 Flags 和 Fragment Offset 字段联合使用，用于对较大的数据报进行分段操作。所有拆分的分段将被标记为相同的值，以便于目的端设备能够识别各个分段属于同一数据报的一部分。
6. **标志 (Flags)**: 长度 3 位，第一位不使用。第二位是 DF (Don't Fragment) 位，当 DF=1 时，表明路由器不能对该数据报进行分段。第三位是 MF (More Fragments) 位，MF=1 表示后续还有更多的分片，MF=0 则表示当前分片是数据报的最后一个。
7. **片偏移 (Fragment Offset)**: 长度 13 位，以 8 个字节为偏移单位。此字段指示接收端该分片在原始数据报中的相对位置，从而确定分片在数据报数据部分的起始位置，便于重组还原原始 IP 包。
8. **生存时间 (TTL)**: 长度 8 位，以跳数为单位。此字段表示数据报在网络中传输过程中能经过的最大跳数。每经过一个三层设备（如路由器），TTL 值减 1。当 TTL 降至 0 时，数据报将被丢弃，防止因路由环路导致数据报在网络中无限循环。
9. **协议 (Protocol)**: 长度 8 位，用于标识上层协议。
10. **首部校验和 (Header Checksum)**: 长度 16 位，用于验证 IP 头部的正确性，但不包括数据部分。

### 2.1.4 回答问题

By looking at the IP packets in your trace, answer these questions:

1. What are the IP addresses of your computer and the remote server?
2. Does the Total Length field include the IP header plus IP payload, or just the IP payload?
3. 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?

4. 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?
5. How can you tell from looking at a packet that it has not been fragmented? Most often IP packets in normal operation are not fragmented. But the receiver must have a way to be sure. Hint: you may need to read your text to confirm a guess.
6. What is the length of the IP Header and how is this encoded in the header length field? Hint: notice that only 4 bits are used for this field, as the version takes up the other 4 bits of the byte. You may guess and check your text.

Hand in your drawing of an IP packet and the answers to the questions above.

### 2.1.5 Internet Path

在命令行下使用 `tracert` 命令，查看到达 `www.baidu.com` 的路由路径。根据输出画出网络路径。

### 2.1.6 计算 Checksum

IP 报头的校验和可以用来验证一个数据包是否正确。
选择一个 IP 报文，计算它的 checksum。计算对 IP 首部检验和的算法如下：

(1) 初始化 IP 数据包的校验和字段，将其值设为零。

(2) 将 IP 首部视为 16 位（即 2 字节）的数字序列，并对这些数字执行连续的二进制加法运算。在此过程中，必须保留从最高有效位（MSB）溢出的进位，因此应使用 32 位加法来确保准确性。

(3) 在加法运算中，如果产生进位，则将此进位加到结果的低 16 位上。在 32 位加法的情况下，这意味着在将高 16 位与低 16 位相加后，还需将此次加法中高 16 位产生的任何进位添加到低 16 位的总和中。

(4) 最后，将得到的总和进行位取反操作，以生成校验和。

### 2.1.7 问题讨论

We encourage you to explore IP on your own once you have completed this lab. Some ideas:

1. Read about and experiment with IPv6. Modern operating systems already include support for IPv6 so you may be able to capture IPv6 traffic on your network. You can also ”join the IPv6 backbone” by tunneling to an IPv6 provider.
2. Learn about tunnels, which wrap an IP packet within another IP header.
3. Read about IP geolocation, it is the process of assigning a geographical location to an IP address using measurements or clues from its name administrative databases. Try a geolocation service.
4. Learn about IPsec or IP security. It provides confidentiality and authentication for IP packets, and is often used as part of VPNs.

### 2.2 实验步骤

1. 启动 Wireshark，在菜单栏中选择“捕获”→“选项”进行设置。选择已连接的以太网接口，设置捕获过滤器为 `tcp port 80`，关闭混杂模式，并勾选 `enable network name resolution`。之后，开始捕获数据。
2. 切换回命令行界面，使用 `wget` 命令发起 HTTP 请求。

```
1      C:\User\GHOST> wget http://www.baidu.com
```

3. 返回到 Wireshark，停止数据捕获。
4. 对捕获的 IP 数据报进行分析，并绘制其结构图。
5. 对捕获的 IP 数据报进行详细数据分析，并回答相关的分析问题。
6. 在命令行中使用 `tracert` 命令，查看到达 `www.baidu.com` 的路由路径。

```
1      C:\User\GHOST> tracert www.baidu.com
```

7. 选择一个 IP 数据报，计算其校验和（checksum）。
8. 进行问题讨论。

## 3 实验环境

- 操作系统：`Windows 11 家庭中文版 23H2 22631.4460`
- 网络适配器：`Killer(R)Wi-Fi 6E AX1675i 160MHz Wireless Network Adapter(211NGW)`
- `Wireshark: Version 4.4.1`
- `wget: GNU Wget 1.21.4 built on mingw32`

## 4 实验过程与分析

### 4.1 捕获数据

首先，我们启动 `Wireshark`，在菜单栏选择捕获，并进行以下设置，连接已连接的以太网，捕获过滤器为 `tcp port 80`，捕获 IP 数据报。

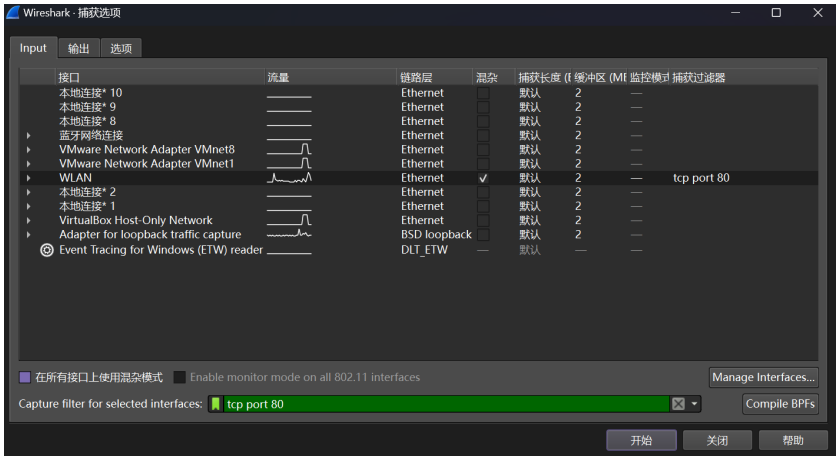


图 1: 设置捕获

然后在命令行使用 `wget` 指令，向 `www.baidu.com` 发送 `HTTP` 请求：


图 2: wget 发送 http 请求

打开 `Wireshark` 的捕获窗口，停止捕获，得到了以下结果：

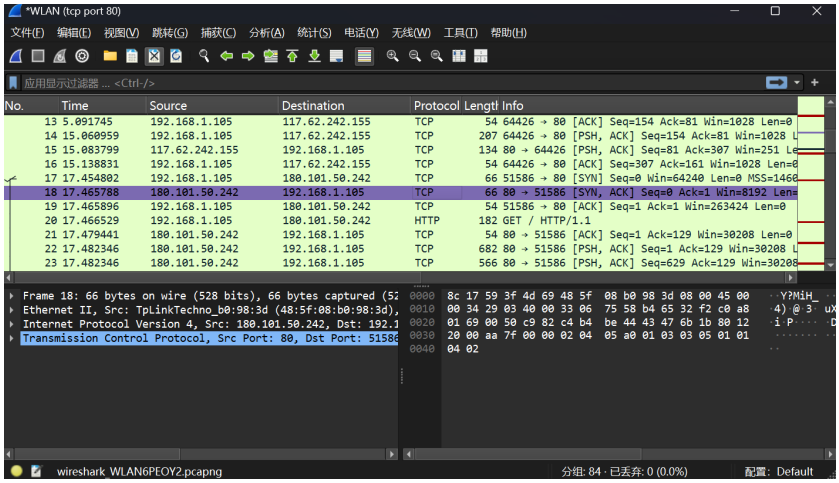


图 3: 捕获结果

## 4.2 分析 IPv4 包并绘制报文结构、数据分析

选择捕获结果中的一个 IP 数据报，分析它的结构，如下图所示：

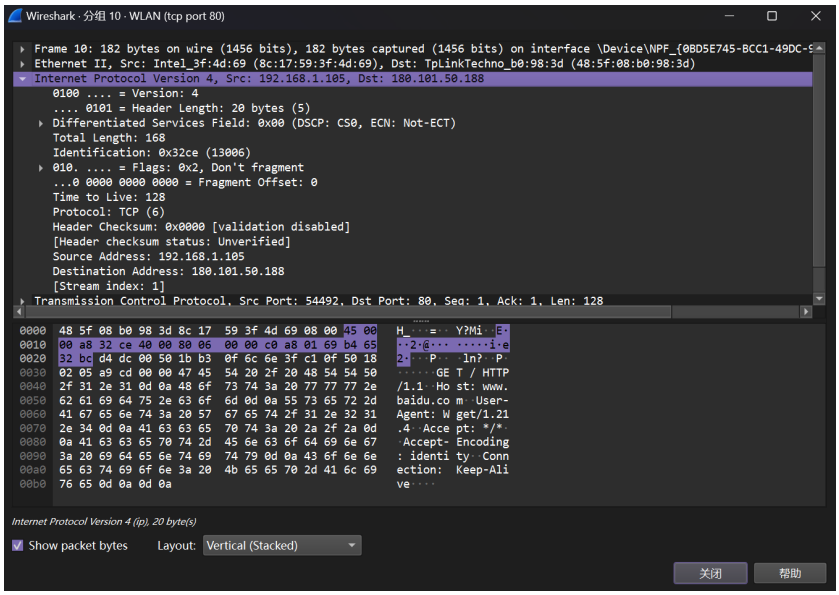


图 4: IPv4 数据报结构

可以看到，第一个字段表示版本号 (Version)，长度为 4bit，表示当前采用的 IP 协议版本号。一般是 0100(IPv4) 或者 0110(IPv6)。

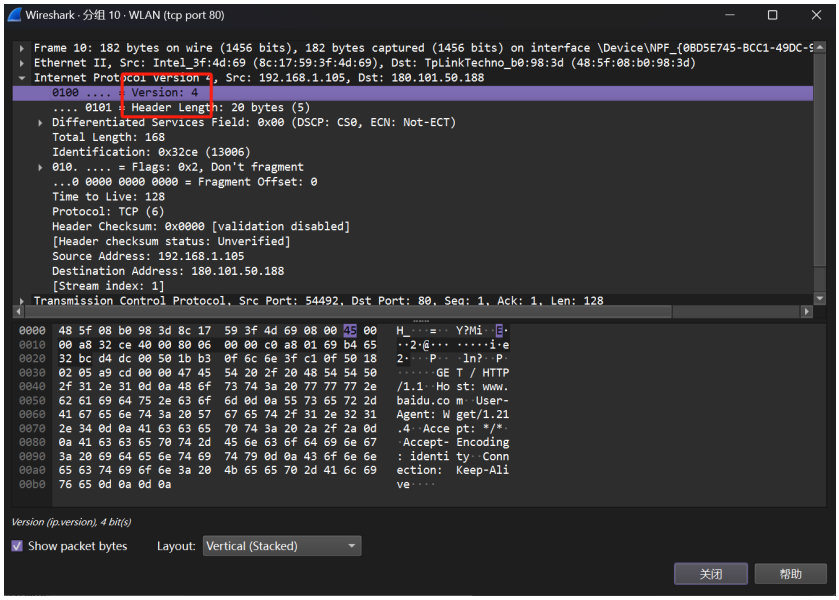


图 5: Version 字段

第二个字段表示首部长度，长度为 `4bit`，表示 IP 报头的长度。在这个数据报中，首部长度为 `20 bytes`。

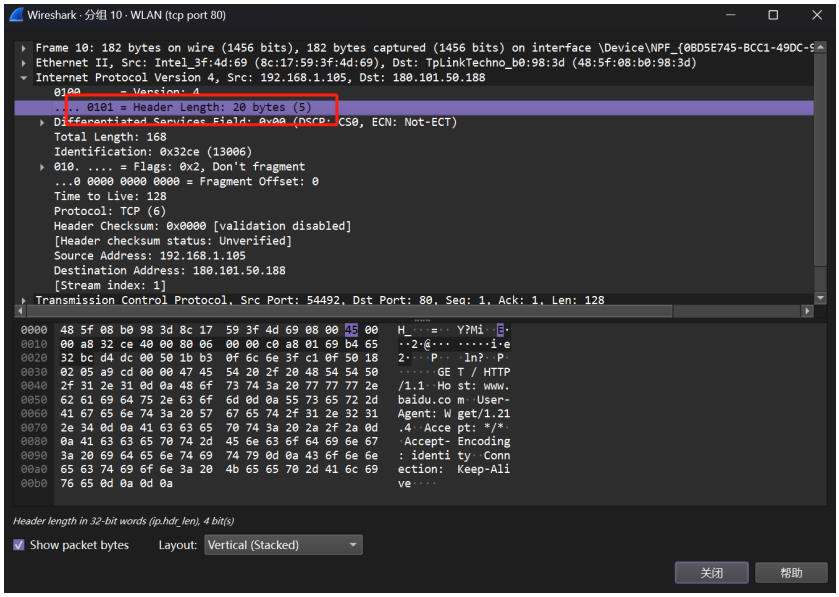


图 6: Header Length 字段

第三个字段表示区分服务，长度为 1 byte，用于为不同的 IP 数据报定义不同的服务质量。

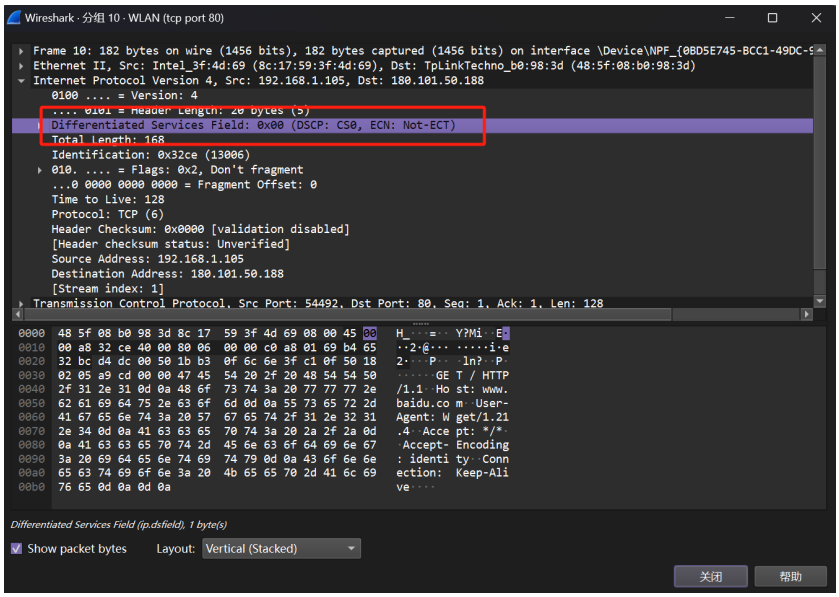


图 7: Differentiated Services 字段

在这个字段中，包括：

1) DSCP，长度为 6 bit，表示区分服务代码点，用于区分不同的服务质量;

2) ECN，长度为 2 bit，表示显式拥塞通知，用于指示网络拥塞。

在这个数据报中，DSCP 的值为 0x00，表示默认服务，ECN 的值为 0x00，表示没有拥塞。

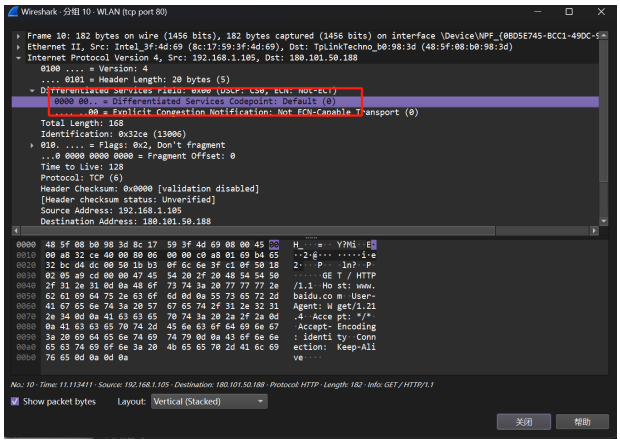

图 8: TCP 数据

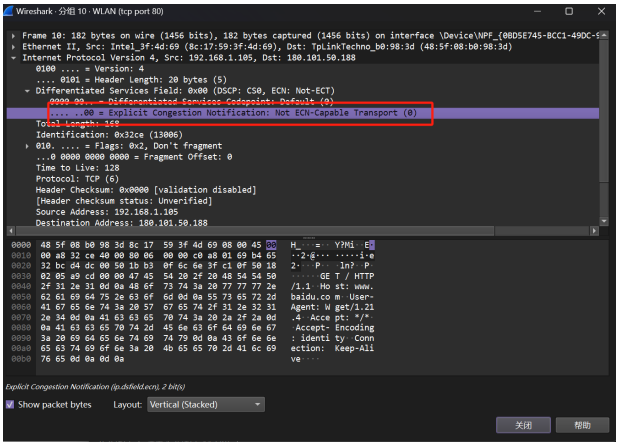

图 9: HTTP 数据

第四个字段指的是 IP 包的总长度，它占用 2 bytes。这个字段以字节为单位，表示 IP 包的总长度，包括头部和数据部分。因此，IP 包的最大长度被限制在 65,535 bytes。数据包的有效载荷大小可以通过从 IP 包的总长度（Total Length）中减去 IP 报头的长度（Header Length）来计算。在当前的数据报中，记录的总长度为 168 bytes。

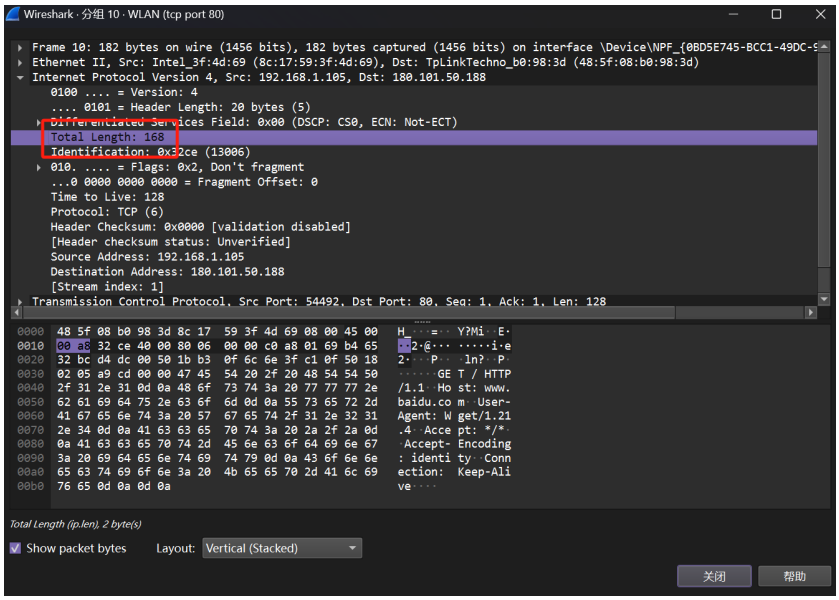


图 10: Total Length 字段

第五个字段是标识字段，它的长度为 2 bytes。此标识字段与 Flags 字段和 Fragment Offset 字段协同工作，用于对较大的数据报执行分段操作。当路由器需要将一个较大的数据报分割成多个较小的片段以便传输时，所有这些片段都会分配相同的标识值。这样，接收端设备就可以识别出哪些片段属于同一个原始数据报。在本例中，数据报的标识字段值为 0x32ce。

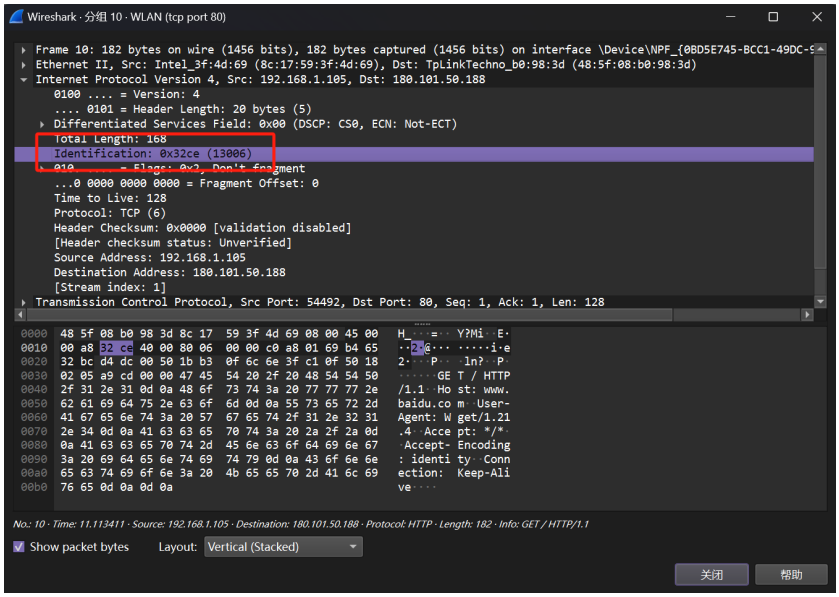


图 11: Identification 字段

第六个字段被称为标志字段，其长度为 3 bits。在这三个比特中，第一个比特是保留的，不用于任何特定的功能。第二个比特被称为 DF（Don't Fragment）位。如果 DF 设置为 1，这指示路由器不得对该数据报进行分段处理。在这种情况下，如果数据报的大小超过了网络的最大传输单元（MTU），路由器将丢弃该数据报，并发送一个错误信息回原始发送者。第三个比特是 MF（More Fragments）位，当 MF 设置为 1 时，它表明在当前分段之后，还有更多的分段存在；而当 MF 设置为 0 时，这表示当前分段是数据报的最后一个分段。在所讨论的数据报中，标志字段的值为 0x02，这表明该数据报未被分段，并且没有后续的分段数据报。

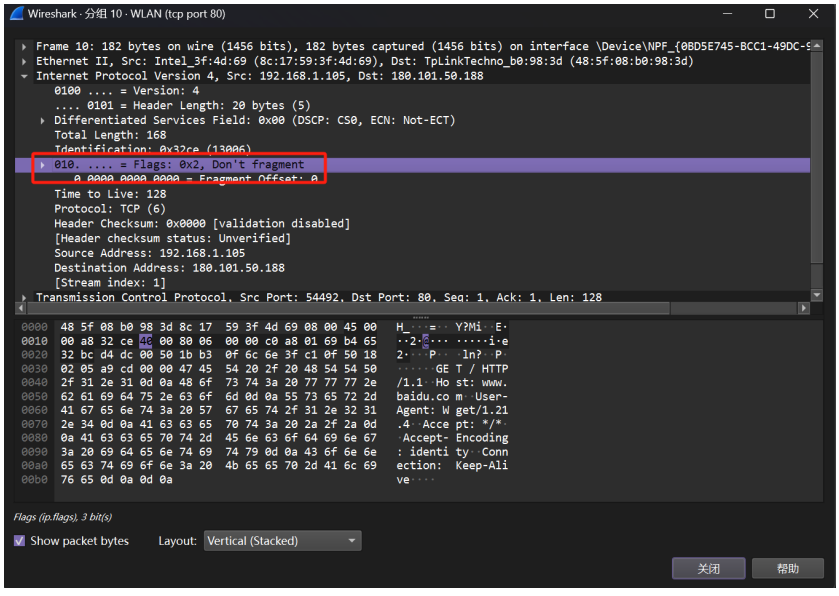


图 12: Flag 字段

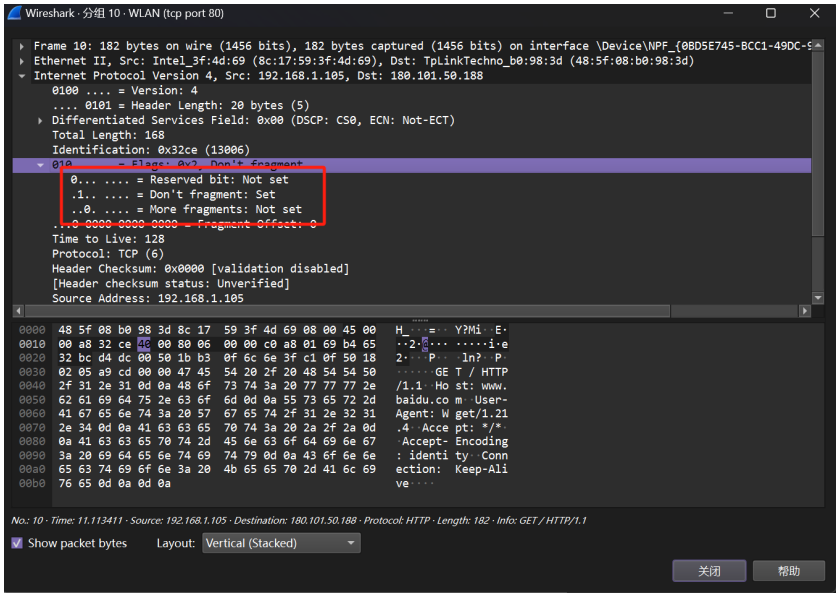


图 13: Flag 字段内容

第七个字段定义为分片偏移，它的长度为 `13 bits`，并且以 `8` 字节为单位进行度量。分片偏移字段的功能是向接收端指明，当前分片在原始数据报中的位置。具体来说，它指示了分片数据部分相对于原始数据报数据起始部分的偏移量。这一信息对于接收端正确地重新组装原始 IP 数据包至关重要。在本数据报实例中，分片偏移量被设置为 `0`，意味着该分片是原始数据报的第一个分片，或者该数据报未被分片。

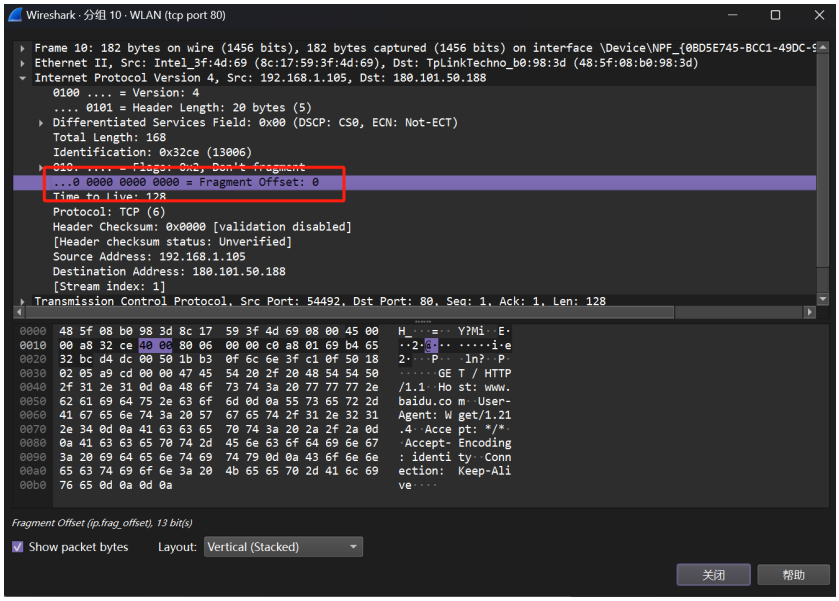


图 14: 14. Fragment Offset 字段.jpg

第八个字段是生存时间（Time to Live, TTL），其长度为 `8 bits`，并且以跳数（hops）为度量单位。此字

段的设计初衷是为了限定数据报在网络中传输时所能经过的最大路由器数量。每当数据报通过一个三层设备，其 TTL 值将减少 1。若 TTL 值降至 0，该数据报将被网络设备丢弃，从而避免数据报在存在路由环路的网络中无限循环。在本数据报中，生存时间被设置为 `128`，这表明它在被丢弃前可以经过 128 个网络节点。

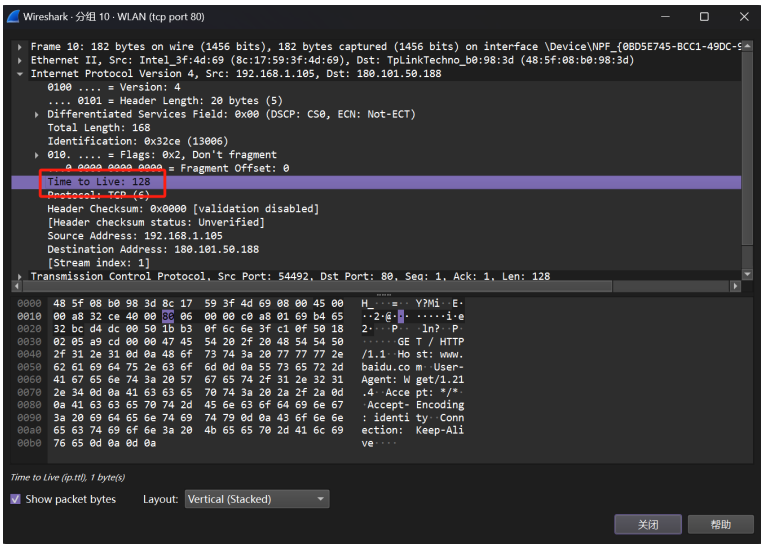


图 15: TTL 字段

第九个字段被称为协议字段，占用 `8 bits`。此字段用于指明数据报携带的上层协议类型。它允许接收端识别应该使用哪种协议来处理数据报的有效载荷。在本例中，数据报指定的协议为 `TCP`，其对应的值为 `0x06`，这表明数据报的上层协议是传输控制协议，通常用于提供可靠的、有序的和错误检测的数据传输。

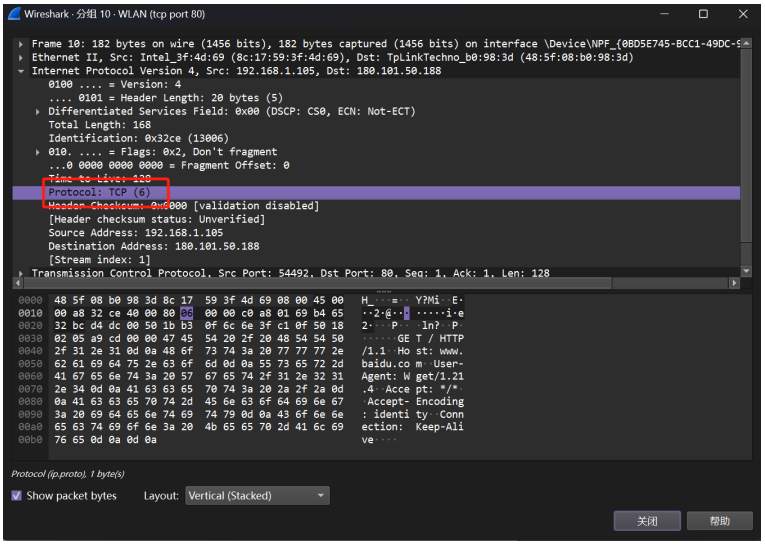


图 16: Protocal 字段

第十个字段是首部校验和，它的长度为 `16 bits`。此字段专用于验证 IP 头部的完整性。校验和的计算不包

括数据部分，仅涵盖 IP 包的头部信息。这是一种错误检测机制，用于确保在传输过程中头部信息未被损坏。在本数据报的案例中，记录的首部校验和值为 `0x03d9`。


图 17: Header Checksum 字段

第十一个字段定义为源 IP 地址，占据 `32 bits` 的空间。这个字段用于记录发送方的网络地址，从而允许接收端识别数据包的起始点。在本次讨论的数据报中，源 IP 地址被指定为 `192.168.1.105`，这为网络通信提供了发送方的位置信息。

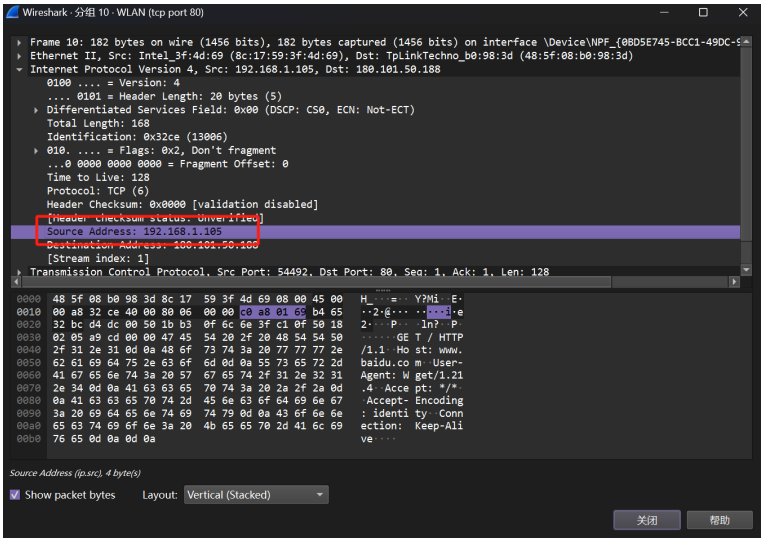


图 18: Source Address 字段

第十二个字段是目的 IP 地址，它的长度为 `32 bits`。此字段用于标识数据包的接收方的网络地址，确保数据能够被正确地送达预定目的地。在本数据报实例中，目的 IP 地址被设置为 `180.101.50.188`，这明确了数据包的最终接收者。

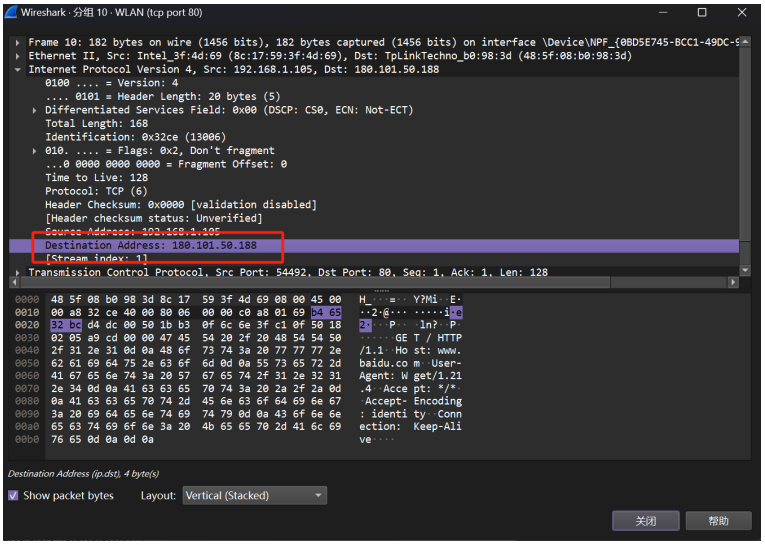


图 19: Destination Address 字段

可以做出如下示意图来表示 IP 数据报的结构：

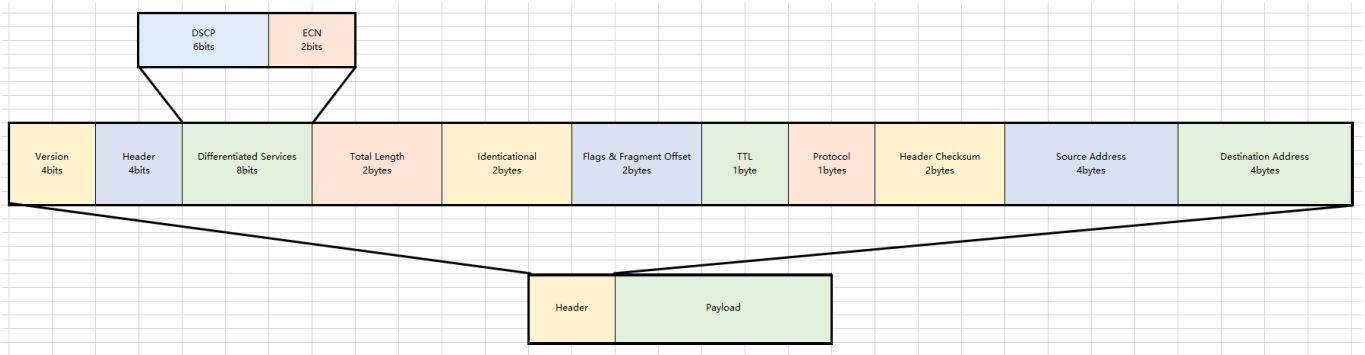


图 20: IP 数据报结构示意图

## 4.3 回答问题

1. What are the IP addresses of your computer and the remote server?
   我的电脑的 IP 地址为 `192.168.1.105`，远程服务器的 IP 地址为 `180.101.50.188`。
2. Does the Total Length field include the IP header plus IP payload, or just the IP payload?
   `Total Length` 字段包括 `IP` 报头和 `IP` 数据的总长度。
3. 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?
   标识字段的值在不同的数据包中不同。在同一个 `TCP` 连接中，标识字段的值不同。在同一个方向上，标识字段的值不同。在不同的方向上，标识字段的值不同。在这个数据报中，标识字段的值为 `0x32ce`。

4. 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?
   我的电脑发送的数据包的生存时间字段的初始值为 128，不是最大值。
5. How can you tell from looking at a packet that it has not been fragmented? Most often IP packets in normal operation are not fragmented. But the receiver must have a way to be sure. Hint: you may need to read your text to confirm a guess.
   如果一个数据包没有被分段，那么它的标志字段的 DF 位为 1，且标志字段的 MF 位为 0。
6. What is the length of the IP Header and how is this encoded in the header length field? Hint: notice that only 4 bits are used for this field, as the version takes up the other 4 bits of the byte. You may guess and check your text.
   IP 报头的长度为 20 bytes，版本号和首部长度字段共占 8 bits，其中版本号占 4 bits，首部长度占 4 bits。

## 4.4 Internet Path

在命令行下使用 tracert 命令，查看到达 www.baidu.com 的路由路径。根据输出画出网络路径。

```
C:\Users\GHOST>tracert www.baidu.com

通过最多 30 个跃点跟踪
到 www.a.shifen.com [180.101.50.242] 的路由：

  1     1 ms     1 ms     1 ms  192.168.1.1
  2     4 ms     5 ms     3 ms  61.152.15.98
  3     6 ms     5 ms     6 ms  61.152.14.17
  4     7 ms     4 ms     6 ms  124.74.254.21
  5     *        *        *     请求超时。
  6     *       11 ms    10 ms  202.97.54.93
  7    10 ms     9 ms     9 ms  58.213.94.178
  8    10 ms     9 ms     9 ms  58.213.94.206
  9    11 ms    12 ms    11 ms  58.213.96.50
 10     *        *        *     请求超时。
 11     *        *        *     请求超时。
 12     *        *        *     请求超时。
 13    10 ms    10 ms     9 ms  180.101.50.242

跟踪完成。
```

图 21: tracert 命令输出

画出网络路径图如下：

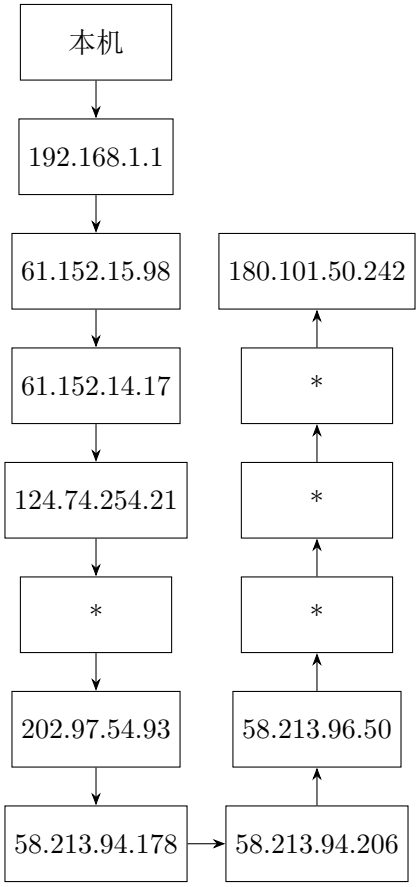


图 22: 网络路径图

## 4.5 计算 checksum

IP 报头的校验和可以用来验证一个数据包是否正确。我们再选择一个新的 IP 报文，计算它的 checksum。

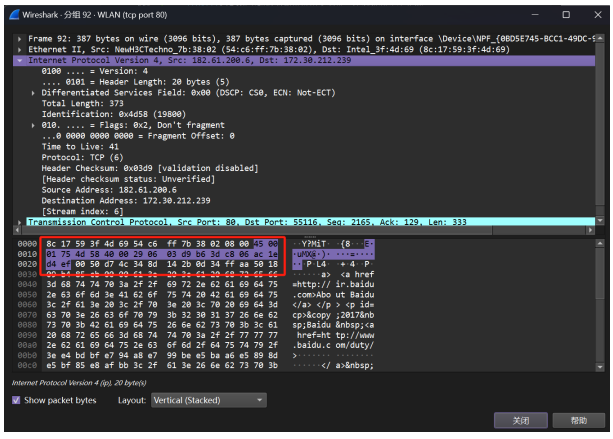


图 23: 选择一个 IPv4 进行计算

将其校验和字段置为 0 后，数据如下：45 00 01 75 4d 58 40 00 29 06 00 00 b6 3d c8 06 ac 1e d4 ef
将其分为两个字节一组，进行二进制求和，再将最高位的进位加到低 16 位，得到的结果如下：

```
      45 00
+     01 75
+     4d 58
+     40 00
+     29 06
+     b6 3d
+     c8 06
+     ac 1e
+     d4 ef
--------------
   3 fc 23
+       3
--------------
      fc 26
```

取反后，得到 checksum 为 0x03d9，与原数据报中的 checksum 字段相同。

```
  ff ff
- fc 26
-----------
  03 d9
```

## 4.6 问题讨论

1. Read about and experiment with IPv6. Modern operating systems already include support for IPv6so you may be able to capture IPv6 traffic on your network. You can also “join the IPv6’backbone by tunneling to an IPv6 provider.

   IPv6 是 IP 协议的下一代协议，它的主要特点是地址空间更大，报头更简单，安全性更好，支持多播和组播，支持流量标签，支持流量优先级，支持更多的选项和扩展，支持更多的协议。现代操作系统已经支持 IPv6，可以在网络上捕获 IPv6 流量。也可以通过隧道连接到 IPv6 提供者。

   我们使用 Wireshark 捕获 IPv6 数据包.

   在 Wireshark 的筛选器中输入 ipv6，可以看到捕获到的 IPv6 数据包。

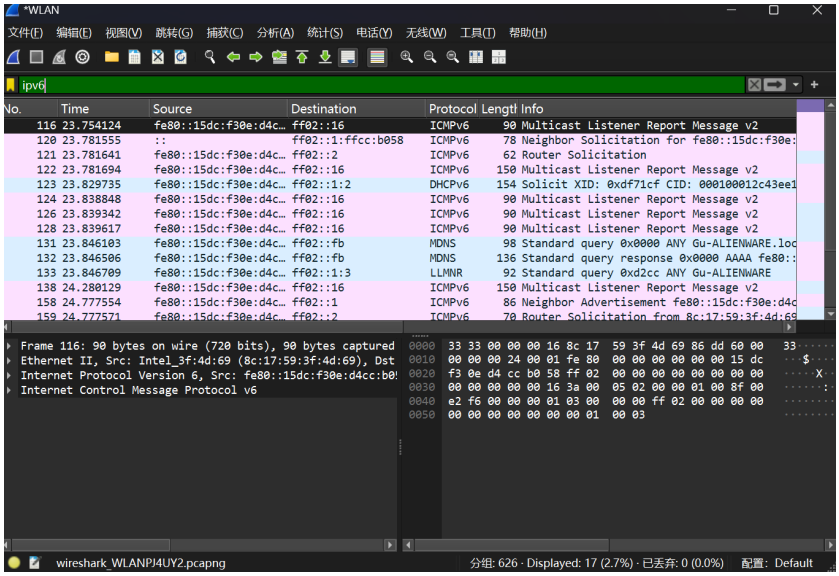


图 24: Wireshark 捕获 IPv6 数据包

选择其中的一个数据包，可以看到其结构如下：

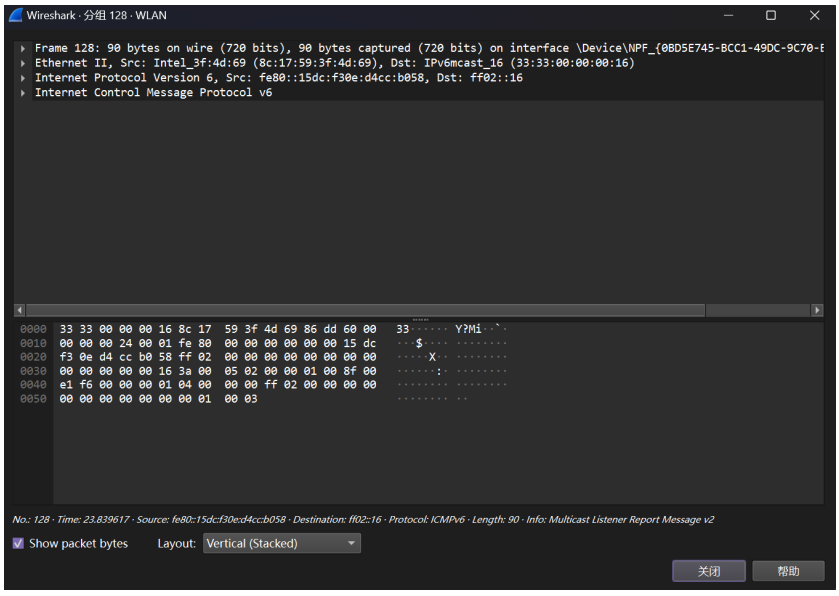


图 25: IPv6 数据包结构

观察可知，IPv6 数据包的构造与 IPv4 数据包存在明显差异。IPv6 数据包的头部固定为 40 字节长，且其源地址和目的地址各占据 16 字节。与 IPv4 不同的是，IPv6 数据包中不再包含校验和字段。此外，IPv6 引入了 Hop Limit 字段，这一字段接替了 IPv4 中的 TTL（Time to Live）字段的功能，用于控制数据包在网络中的传输跳数。

2. Learn about tunnels, which wrap an IP packet within another IP header.

隧道技术是一种网络通信方法，它允许将一个 IP 数据包嵌入到另一个 IP 数据包的头部之中。这种方法特别适用于不同 IP 版本之间的数据传输。具体来说，隧道技术能够实现 IPv6 数据包在 IPv4 网络中的传输，通过将 IPv6 数据包封装在 IPv4 数据包内来完成。反之，它也能够支持 IPv4 数据包在 IPv6 网络中的传输，即通过将 IPv4 数据包封装进 IPv6 数据包来实现。这种灵活性使得隧道技术成为 IP 版本转换和共存策略中的关键组成部分。

3. Read about IP geolocation, It is the process of assigning a geographical location to an IP addressusing measurements or clues from its name administrative databases. Try a geolocation service.
   IP 地理定位是一个技术过程，它将具体的地理位置与 IP 地址相关联。这一过程可以通过直接测量或通过查询 IP 地址的域名管理系统数据库来实现。为了进一步了解或应用 IP 地理定位，可以使用在线服务。例如，网站 `https://www.iplocation.net/` 提供了一项功能，允许用户通过输入 IP 地址来获取相关的地理信息。
4. Learn about IPsec or IP security. It provides confidentiality and authentication for IP packets, and isoften used as part of VPNs.
   IPsec，即网际协议安全，是一种旨在增强 IP 数据包传输安全性的协议。它主要用于提供数据的保密性与发送者身份的验证，经常集成于虚拟私人网络（VPN）的架构中。IPsec 通过在网络层对 IP 数据包实施加密和认证机制，保障信息在网络传输过程中的安全性，确保数据的机密性、完整性，并核实通信双方的身份，从而为网络通信提供了一道坚固的安全屏障。

# 5 实验结果总结

在本次实验中，我们通过 Wireshark 工具捕获了 IP 数据包，并对其进行了深入分析。具体步骤和发现如下：

我们首先使用 Wireshark 捕获了网络中的 IP 数据包。接着，我们对 IP 数据包的结构进行了详细分析，并解答了与这些数据包相关的一系列问题。利用 `tracert` 命令，我们追踪了到达`www.baidu.com` 的路由路径，并且绘制了相应的网络路径图。我们挑选了一个特定的 IP 数据包，并计算了其校验和（checksum）。

在实验的最后阶段，我们对实验过程中遇到的问题进行了深入讨论。此外，我们还捕获并分析了 IPv6 数据包，以扩展我们对 IP 协议家族的理解。

# 6 附录

无