



哈尔滨工业大学  
Harbin Institute of Technology

# 计算机网络 课程实验报告

实验名称	IPv4 分组收发转发					
姓名	张明远		院系	计算机科学与技术		
班级	2003104		学号	120L030501		
任课教师	聂兰顺		指导教师	聂兰顺		
实验地点	格物 207		实验时间	2022.10.31		
实验课表现	出勤、表现得分(10)		实验报告 得分(40)		实验总分	
	操作结果得分(50)					
教师评语						

**实验目的：****IPv4分组收发实验目的：**

IPv4协议是互联网的核心协议，它保证了网络节点（包括网络设备和主机）在网络层能够按照标准协议互相通信。IPv4地址唯一标识了网络节点和网络的连接关系。在我们日常使用的计算机的主机协议栈中，IPv4协议必不可少，它能够接收网络中传送给本机的分组，同时也能根据上层协议的要求将报文封装为IPv4分组发送出去。

本实验通过设计实现主机协议栈中的IPv4协议，让学生深入了解网络层协议的基本原理，学习IPv4协议基本的分组接收和发送流程。

另外，通过本实验，学生可以初步接触互联网协议栈的结构和计算机网络实验系统，为后面进行更为深入复杂的实验奠定良好的基础。

**IPv4 分组转发实验目的：**

通过前面的实验，我们已经深入了解了IPv4协议的分组接收和发送处理流程。本实验需要将实验模块的角色定位从通信两端的主机转移到作为中间节点的路由器上，在IPv4分组收发处理的基础上，实现分组的路由转发功能。

网络层协议最为关注的是如何将IPv4分组从源主机通过网络送达目的主机，这个任务就是由路由器中的IPv4协议模块所承担。路由器根据自身所获得的路由信息，将收到的IPv4分组转发给正确的下一跳路由器。如此逐跳地对分组进行转发，直至该分组抵达目的主机。IPv4分组转发是路由器最为重要的功能。

本实验设计模拟实现路由器中的IPv4协议，可以在原有IPv4分组收发实验的基础上，增加IPv4分组的转发功能。对网络的观察视角由主机转移到路由器中，了解路由器是如何为分组选择路由，并逐跳地将分组发送到目的主机。本实验中也会初步接触路由表这一重要的数据结构，认识路由器是如何根据路由表对分组进行转发的。

**实验内容：****IPv4分组收发实验内容：****1) 实现IPv4分组的基本接收处理功能**

对于接收到的IPv4分组，检查目的地址是否为本地地址，并检查IPv4分组头部中其它字段的合法性。提交正确的分组给上层协议继续处理，丢弃错误的分组并说明错误类型。

**2) 实现IPv4分组的封装发送**

根据上层协议所提供的参数，封装IPv4分组，调用系统提供的发送接口函数将分组发送出去。

**IPv4分组转发实验功能：****1) 设计路由表数据结构。**

设计路由表所采用的数据结构。要求能够根据目的IPv4地址来确定分组处理行为（转发情况下需获得下一跳的IPv4地址）。路由表的数据结构和查找算法会极大的影响路由器的转发性能，有兴趣的同学可以深入思考和探索。

**2) IPv4分组的接收和发送。**

对前面实验（IP实验）中所完成的代码进行修改，在路由器协议栈的IPv4模块中能够正确完成分组的接收和发送处理。具体要求不做改变，参见“IP实验”。

**3) IPv4分组的转发。**

对于需要转发的分组进行处理，获得下一跳的IP地址，然后调用发送接口函数做进一步处理。

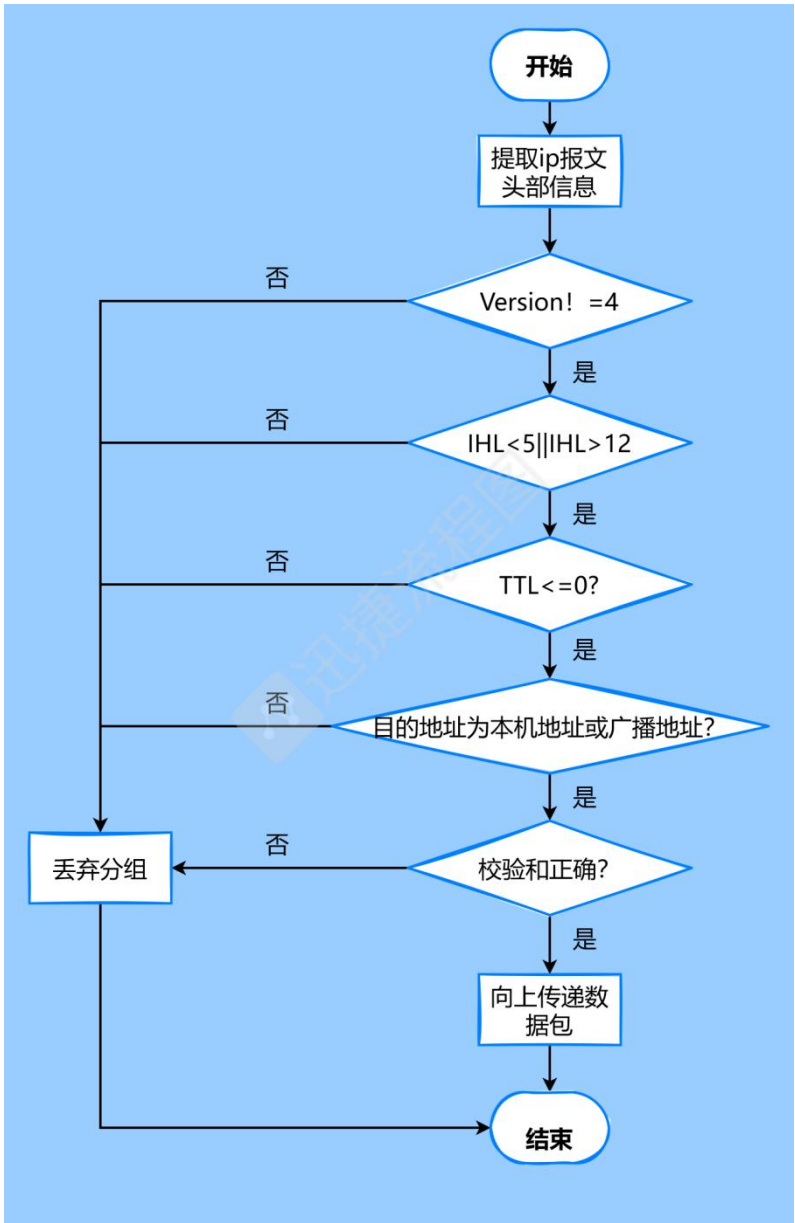
**实验过程：**

### (一) IP分组收发实验

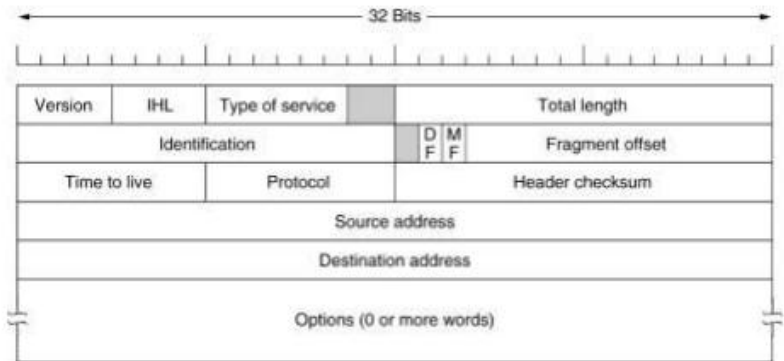
#### 1. 发送函数流程图



#### 2. 接收函数流程图



3. 直接进行信息分析，无新建的数据结构，不再进行说明。
3. IPv4数据包头部信息如下图所示：



## 1) 版本号字段错误检验原理

版本号Version在第0字节的高4位，ipv4时值为4。只需要提取Version并进行比较即可。

## 2) 头部长度（IP Head Length）字段检验原理

头部长度信息是以4字节为单位，存储在IP数据包头部的第0字节的低4位，所以仅需利用位运算提取出低4位的信息；并且根据最少的IP数据包的头部信息为20字节，所以低于5的头部长度字段是错误的。

## 3) 生存时间（Time to live, TTL）字段检验原理

生存时间字段在IP数据包的头部第8字节，因此仅仅需要将对应的字节提取出来，与0进行比较。如果TTL小于等于0说明其已经过期，丢弃即可。

## 4) 头部校验和（Header Checksum）字段检验原理

头部校验和字段在IP数据包的头部第10个字节，并且根据之前进行计算时取反的性质，将所有的头部信息进行与checksum生成时相同的计算步骤，得到的结果应该为全1；否则说明头部校验和错误。

前面部分的具体实现如下：

```
int stud_ip_recv(char *pBuffer, unsigned short length)
{
    int version = pBuffer[0] >> 4; //提取version
    int ihl = pBuffer[0] & 0xf; //提取ihl
    short ttl = (unsigned short)pBuffer[8]; //提取ttl
    short checksum = ntohs(*(unsigned short *)(pBuffer + 10)); //提取checksum
    int destination = ntohl(*(unsigned int *)(pBuffer + 16)); //提取destination

    //version错误
    if (version != 4)
    {
        ip_DiscardPkt(pBuffer, STUD_IP_TEST_VERSION_ERROR);
        return 1;
    }

    //ihl错误
    if (ihl < 5 || ihl > 12)
    {
        ip_DiscardPkt(pBuffer, STUD_IP_TEST_HEADLEN_ERROR);
        return 1;
    }

    //ttl错误
    if (ttl <= 0)
    {
        ip_DiscardPkt(pBuffer, STUD_IP_TEST_TTL_ERROR);
        return 1;
    }

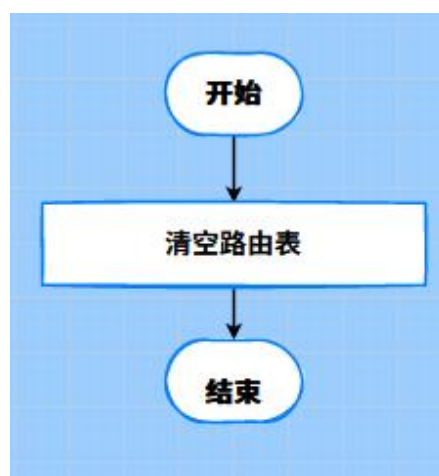
    //destination错误
    if (destination != getIpv4Address() && destination != 0xffff)
    {

```

```
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_DESTINATION_ERROR);
    return 1;
}
//计算检验和
unsigned long sum = 0;
unsigned long temp = 0;
int i;
for (i = 0; i < ihl * 2; i++)
{
    temp += (unsigned char)pBuffer[i * 2] << 8;
    temp += (unsigned char)pBuffer[i * 2 + 1];
    sum += temp;
    temp = 0;
}
unsigned short l_word = sum & 0xffff;
unsigned short h_word = sum >> 16;
//检验和错误
if (l_word + h_word != 0xffff)
{
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_CHECKSUM_ERROR);
    return 1;
}
//全部正确
ip_SendtoUp(pBuffer, length);
return 0;
}
```

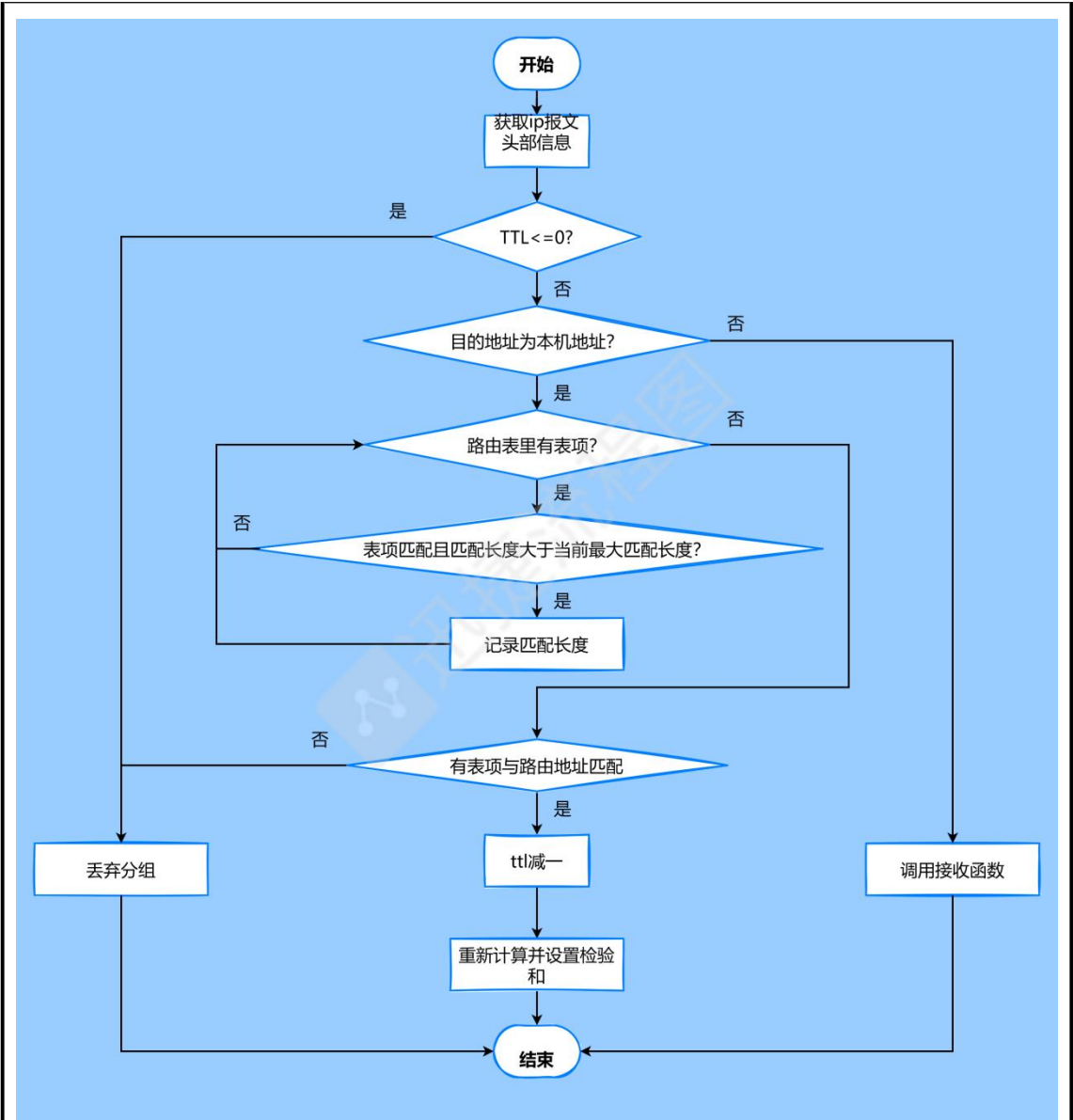
## （二）IP分组转发实验

1. 路由表采用全局变量，路由表初始化函数stud\_Route\_Init()在初始化时只执行清空操作，展示流程图：



下面是路由增加和路由转发的实现流程图





上图为系统处理收到的IP分组的stud\_fwd\_deal()函数的流程图。

2. 路由表使用的数据结构为c++内置的STL中的vector。查找方式为顺序查找。

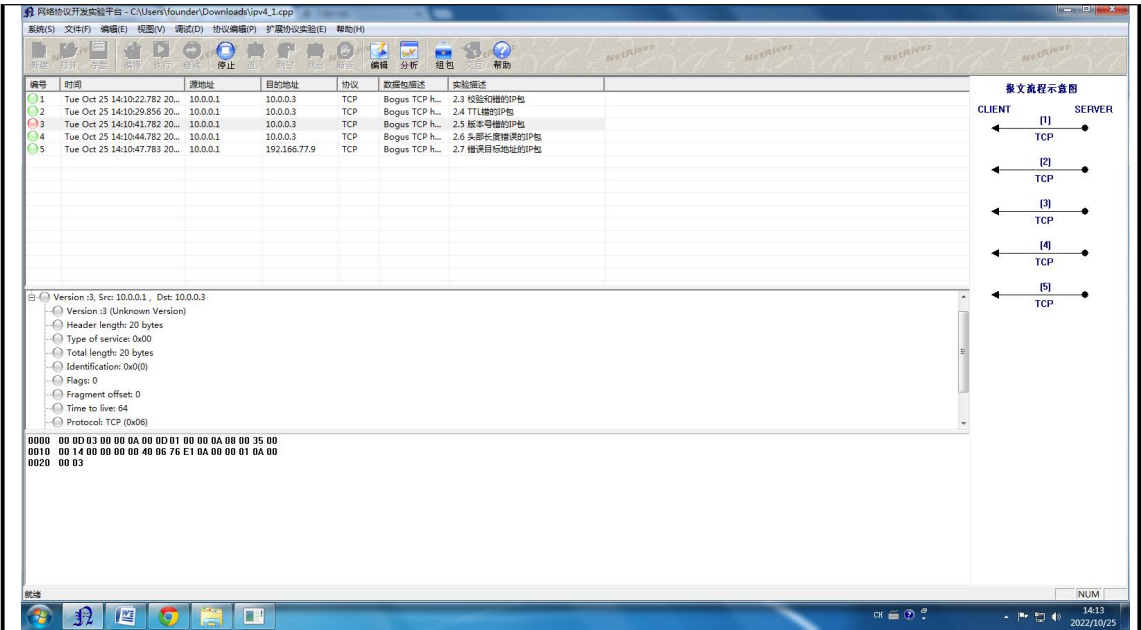
实验结果：

采用演示截图、文字说明等方式，给出本次实验的实验结果。

(一) IP分组收发实验

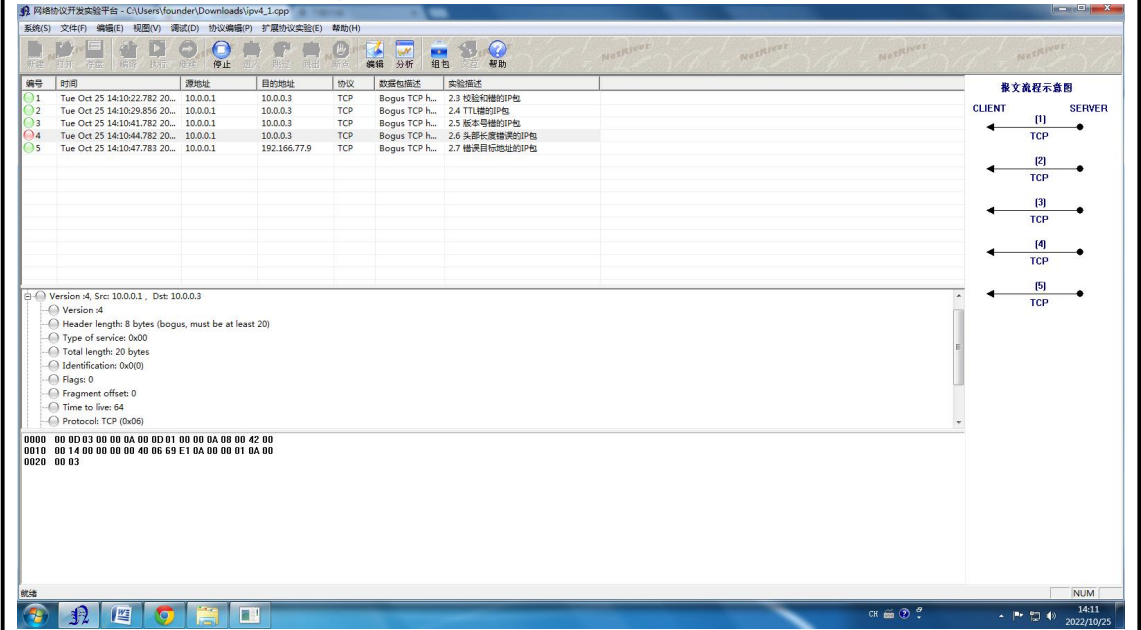
1. version字段错误





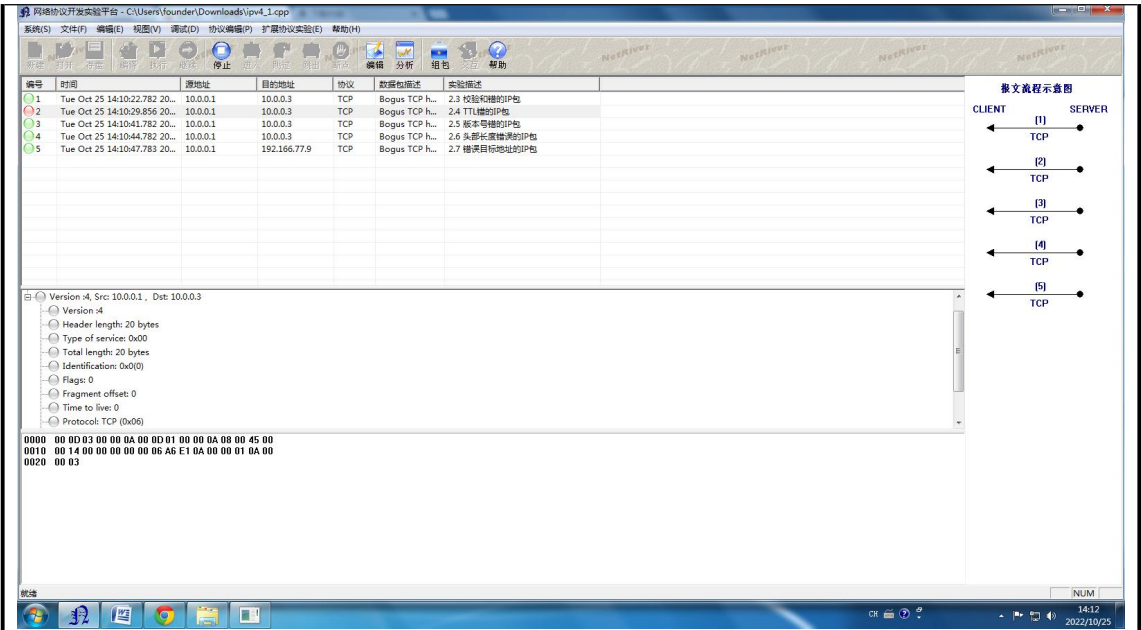
可以看到在版本号错误的实验样例中，给出的IP数据包的version段为3，与IPv4的数据包的头部version为4不符。

## 2. 头部长度（IP Head Length）错误



可以看到在实验测试中给出的样例为头部长度为8，这与最少的IP头部长度字段为20不符。

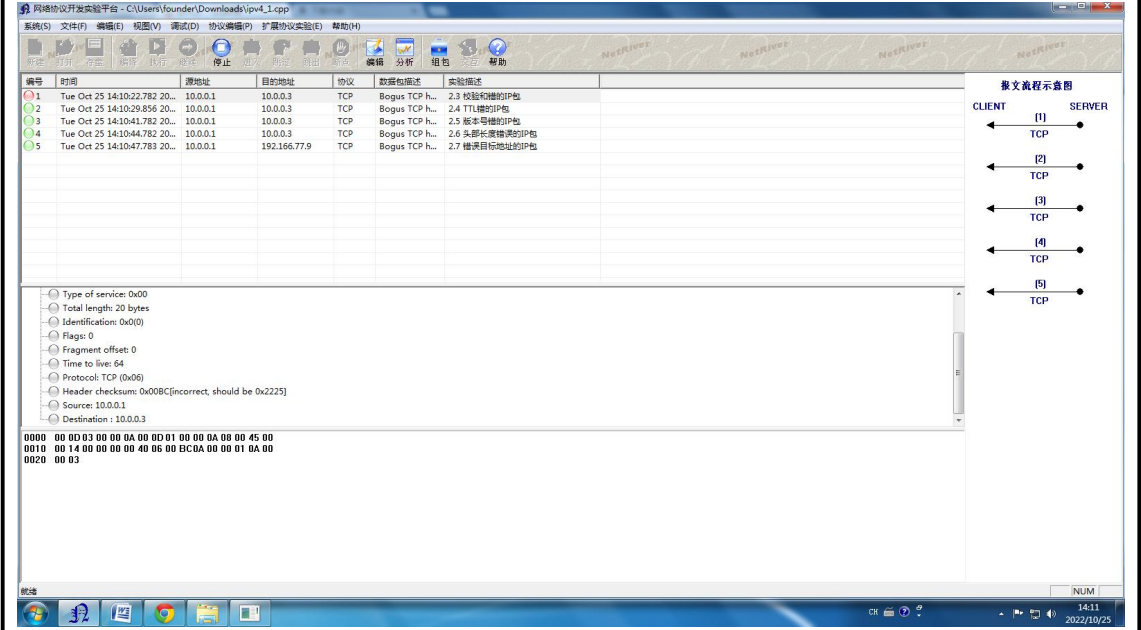
## 3. 生存时间（TTL）错误



可以看到在实验中给出的带有错误TTL数据段的数据包，在TTL字段的长度为0，意味着该数据包应该进行丢弃。

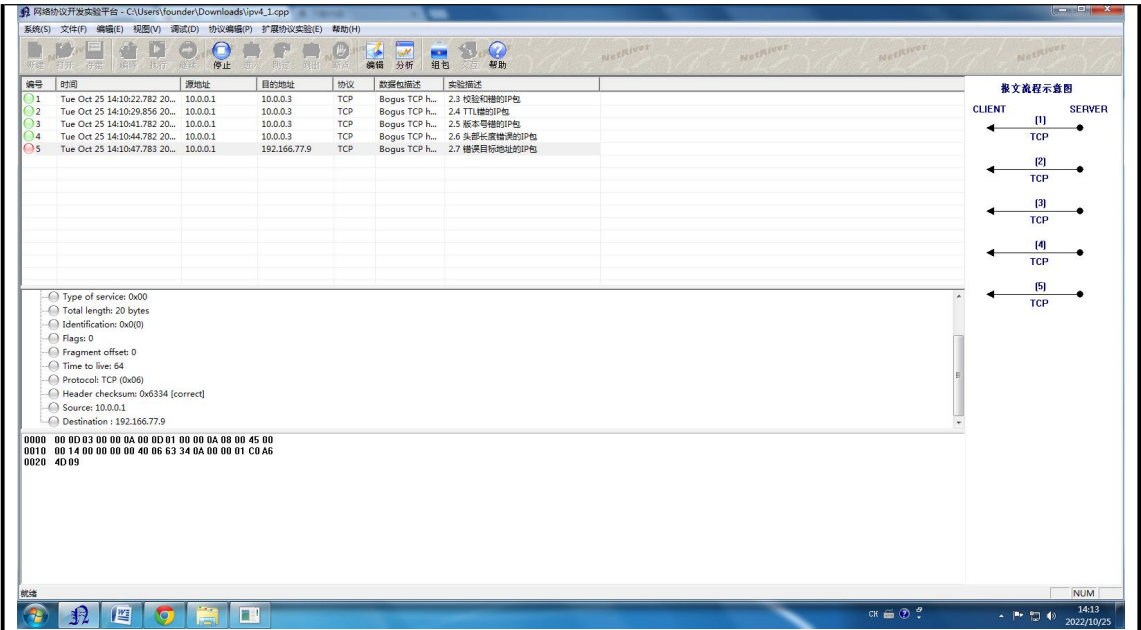
4. 校验和（Header Checksum）字段错误

根据发送方checksum的计算公式可知，在接收方重新计算checksum的结果应该为全1（即16个1），在实验给出的有关校验和错误的样例中，如下



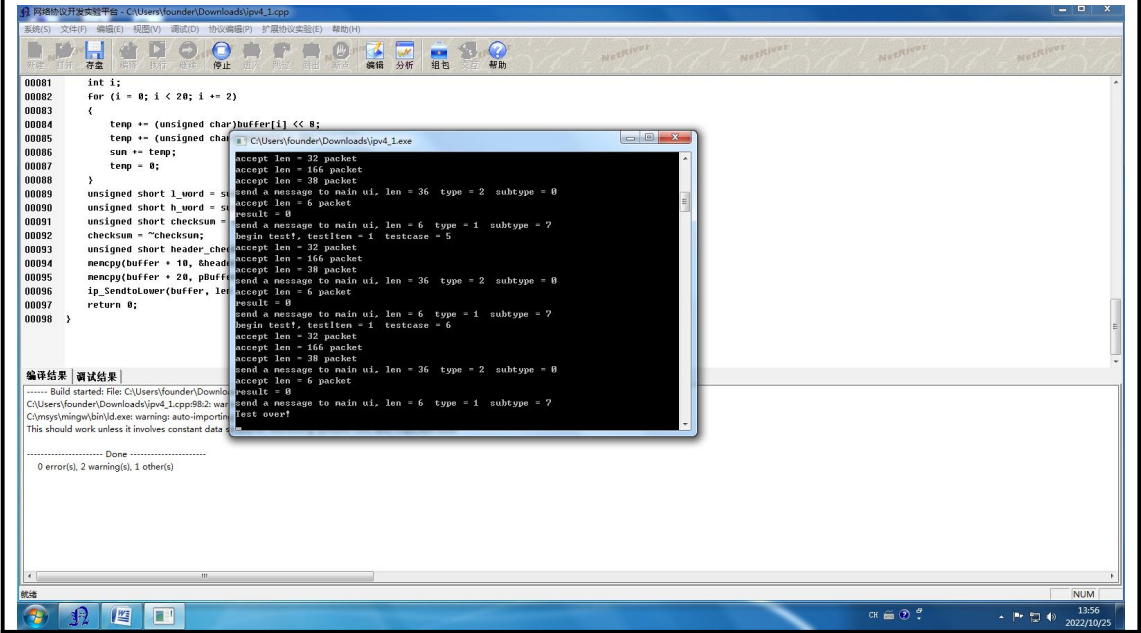
可以看到，正确的校验和为0x2225，实际为0x00BC，所以发生了校验和错误。

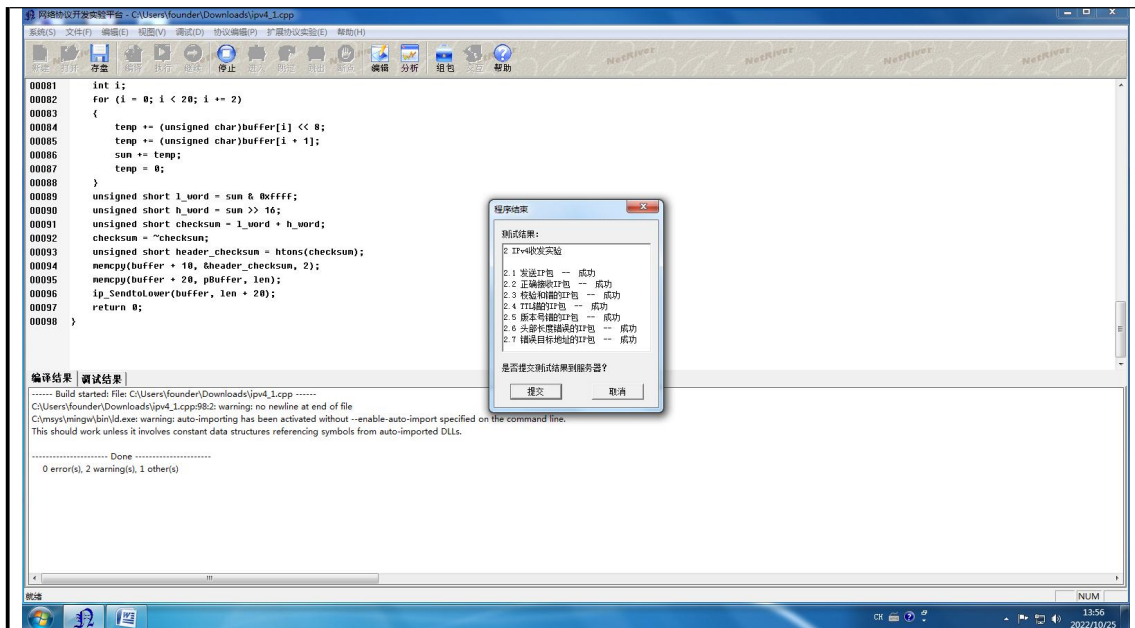
5. 目的地址（Destination）字段错误



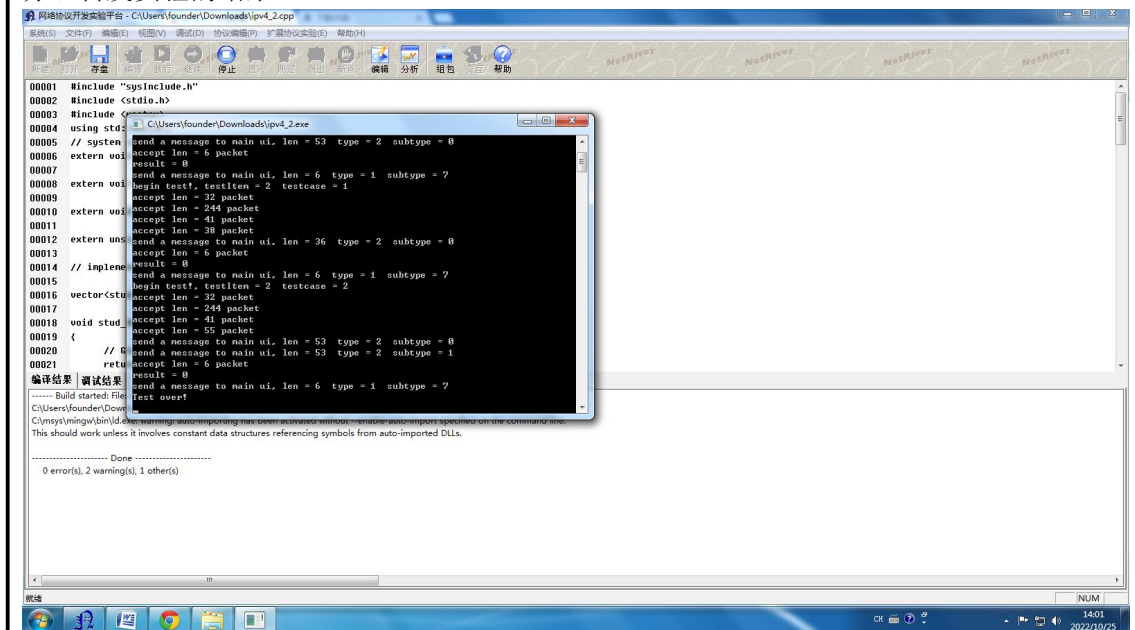
可以看到目的地址由10.0.3变为了192.166.77.9，应该丢弃。

6. 分组收发实验的结果





## 分组转发实验的结果



学期	序号	学号	姓名	院系	班级	实验名称	实验日期	实验结果	总成绩	程序	报告
2022年秋季	1	1201030501	张明远	计算机科学与技术学院	2003104	IPv4收发实验	2022-11-07	<div><div style="width: 80%;"></div></div>	0	<div><div></div><div></div><div></div><div></div></div>	
2022年秋季	2	1201030501	张明远	计算机科学与技术学院	2003104	IPv4转发实验	2022-11-07	<div><div style="width: 60%;"></div></div>	0	<div><div></div><div></div><div></div><div></div></div>	

对实验过程中的思考问题进行讨论或回答。

1. 如果继续使用本次实验中所使用的vector进行效率的提高，最简单的方式就是将路由表进行有序的存储（如按照IP目的地址的大小存储），在vector进行二分查找，将每次转发搜索路由表的时间由 $O(n)$ 变为 $O(\log n)$ 。
2. 如果改进路由表的数据结构，可以使用平衡树之类数据结构，将查询的时间变为稳定 $O(\log n)$ ，并且缩小新增表项重新整理成有序的时间。

本次实验是在由清华大学在2006年所构建的实验平台上进行。提供了查看数据包的构建以及断点调试的功能。这对于实现原本的功能有着很大的帮助。

通过在实验中，模拟实现分组收发和分组转发，对于路由器的功能实现有了更深入的认识。