哈爾濱工業大學
Harbin Institute of Technology

计算机网络
课程实验报告

实验名称	IPv4 分组收发实验，Ipv4 分组转发实验				
姓名	姚舜宇		院系	计算学部	
班级	1903602		学号	1190202107	
任课教师	李全龙		指导教师	李全龙	
实验地点	格物 207		实验时间	2021.11.13	
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					

计算机科学与技术学院 SINCE 1956...
School of Computer Science and Technology

实验目的：

本次实验的主要目的。

IPv4 分组收发实验：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：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：

概述本次实验的主要内容，包含的实验项等。

IPv4分组收发实验：

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. 实验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：

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。
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。
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。

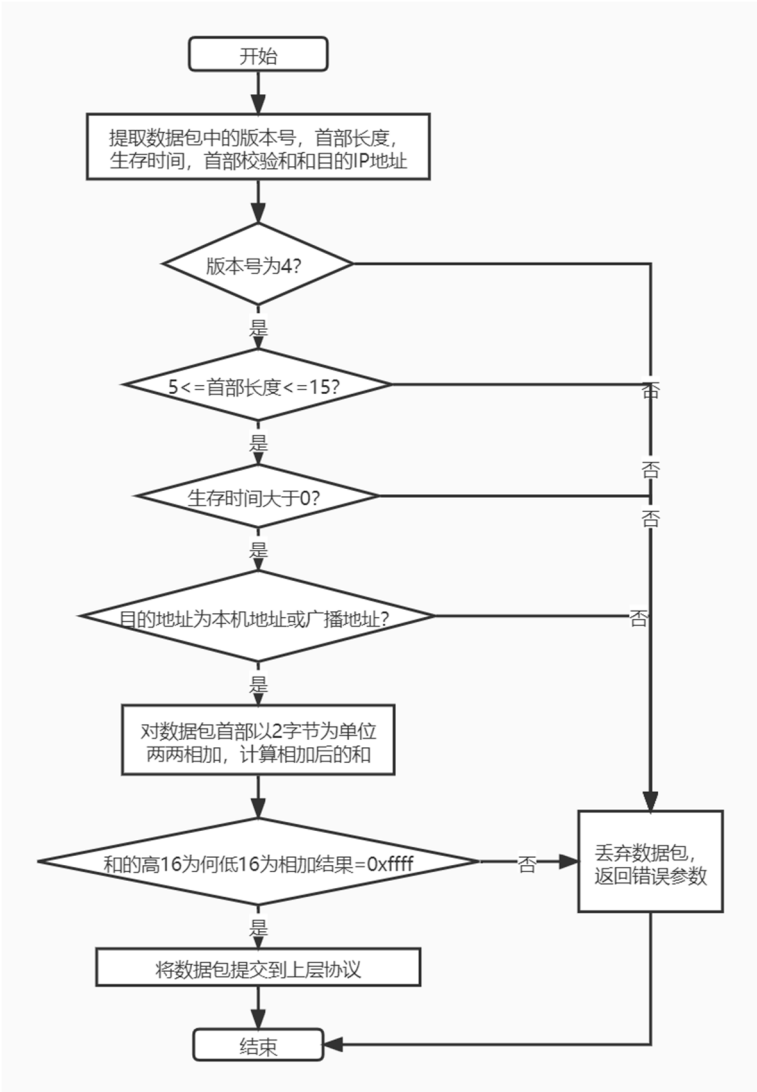
实验过程：

以文字描述、实验结果截图等形式阐述实验过程，必要时可附相应的代码截图或以附件形式提交。

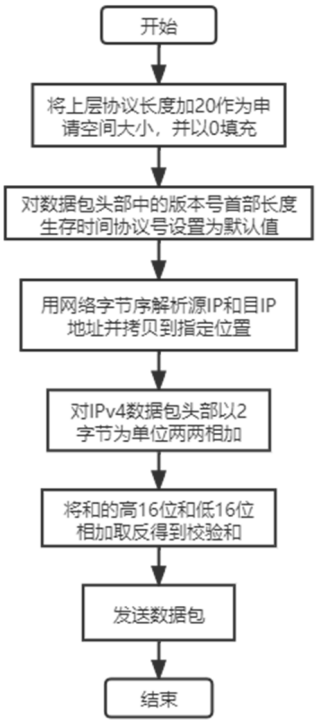
IPv4分组收发实验：

IPv4分组示意图：

接收函数stud_ip_recv()的流程图：



发送函数stud_ip_Upsend()的流程图：



各个字段的读取与检验：

根据IPv4分组格式，可以读取出数据包的各个字段：

```
int version = pBuffer[0] >> 4;
int head_length = pBuffer[0] & 0xf;
short ttl = (unsigned short)pBuffer[8];
short checksum = ntohs(*(unsigned short*)(pBuffer + 10));
int destination = ntohl(*(unsigned int*)(pBuffer + 16));
```

版本号检验：

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

对于IPv4分组，版本号必须为4，否则调用ip_DiscardPkt()函数并输入错误类型STUD_IP_TEST_VERSION_ERROR。

首部长度检验：

```
if (head_length < 5 || head_length > 15)
{
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_HEADLEN_ERROR);
    return 1;
}
```

首部长度由4位二进制表示，首部字段最小为20字节，以4字节为单位，故最小是5。4为二进制最大表示15，故头部长度最大为15。若错误则调用函数并输入错误类型

STUD_IP_TEST_HEADLEN_ERROR。
生存时间检验：

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

生存时间比如大于0，否则调用函数并输入错误类型STUD_IP_TEST_TTL_ERROR。
首部校验和检验：

```
unsigned long sum = 0;
unsigned long t = 0;
int i;
for (i = 0; i < head_length * 2; i++)
{
    t += (unsigned char)pBuffer[i * 2] << 8;
    t += (unsigned char)pBuffer[i * 2 + 1];
    sum += t;
    t = 0;
}
unsigned short low_of_sum = sum & 0xffff;
unsigned short high_of_sum = sum >> 16;
if (low_of_sum + high_of_sum != 0xffff)
{
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_CHECKSUM_ERROR);
    return 1;
}
```

计算校验和，对数据包首部以两个字节为单位，两两相加，计算相加之后的sum，然后取出sum的高16位和低16位相加与0xffff进行比较，若不相等，则调用函数并输入错误类型STUD_IP_TEST_CHECKSUM_ERROR。
目的地址字段的检测：

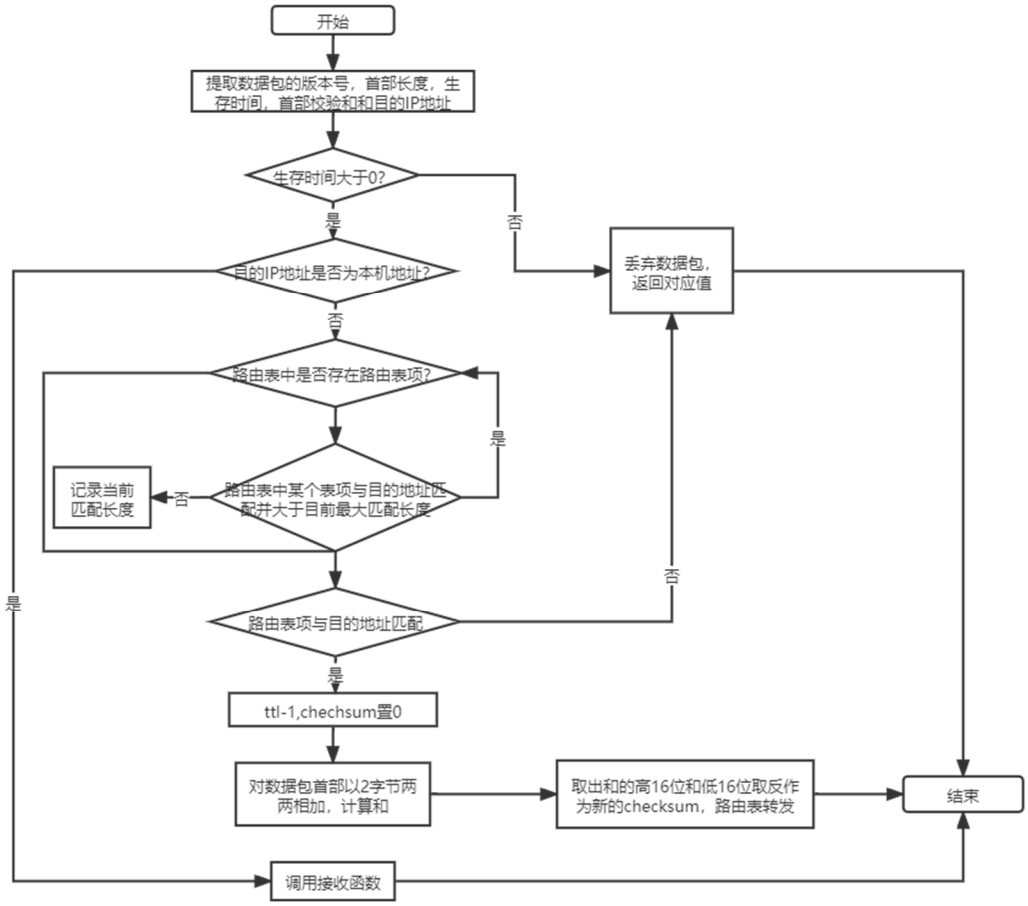
```
if (destination != getIpv4Address() && destination != 0xffff)
{
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_DESTINATION_ERROR);
    return 1;
}
```

若不是目的地址或广播地址，则调用函数并输入错误类型STUD_IP_TEST_DISTINATION_ERROR。

IPv4分组转发实验：
路由表初始化函数：
在实验中使用了一个全局向量(vector)来作为存储路由表，在创建时就完成了初始化，所以初始化函数不需要执行任何指令。

```
void stud_Route_Init()
{
    return;
}
```

路由转发函数的流程图：

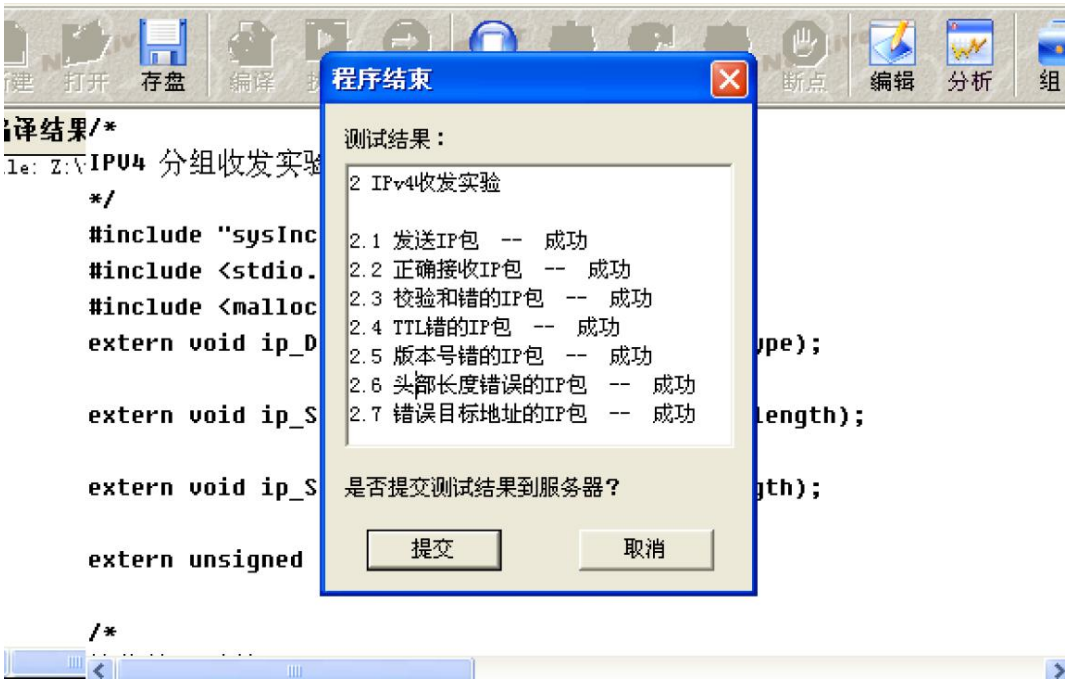


程序中使用向量vector来作为路由表，里面存储了所有的路由表项。当需要增加路由表项时，直接在向量末尾增加新的路由表项即可。当需要查找路由表项时，需要从头部进行遍历，以获得下一跳的IP地址。

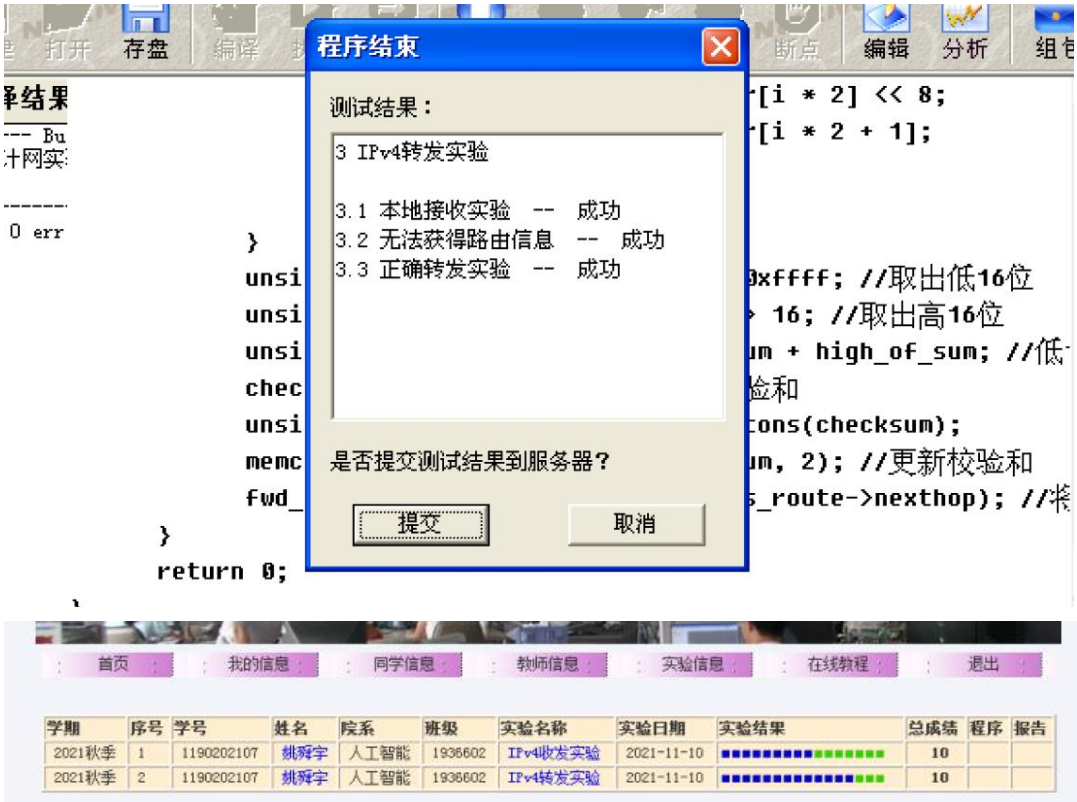
实验结果：

采用演示截图、文字说明等方式，给出本次实验的实验结果。

IPv4分组收发实验：



IPv4分组转发实验：



问题讨论：

使用线性结构来存储路由表项，检索时候效率会较低，所以可以使用其他方法来进行存储，如树结构，按照目的IP地址的大小对路由表项进行排序，检索时只需要对数时间复杂度即可。

心得体会：

结合实验过程和结果给出实验的体会和收获。

通过此次实验，我熟悉了IP分组的结构，路由表的作用以及数据包的收发和分组转发的过程。了解了路由器路由的原理，并逐跳将数据包发送到目的主机的过程。