



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 分组收发实验 IPv4 分组转发实验					
姓名	张瑞		院系	计算学部		
班级	1903104		学号	1190201421		
任课教师	刘亚维		指导教师	刘亚维		
实验地点	格物 207		实验时间	2021 年 11 月 14 日		
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。

IPv4分组收发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。

IPv4分组转发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：****(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：****(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：**

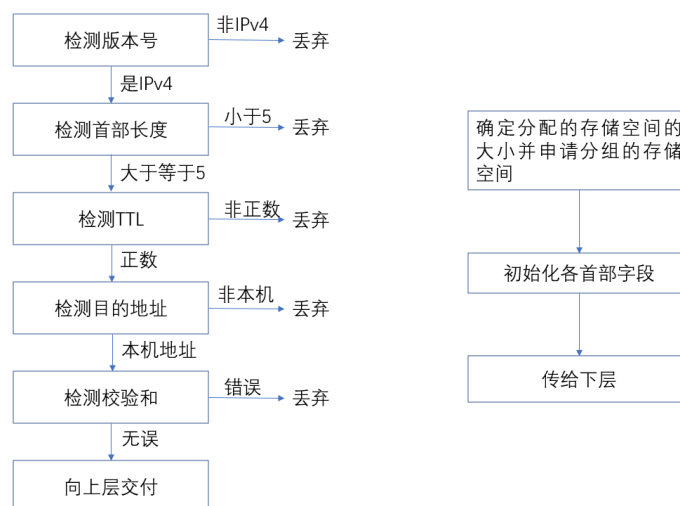
对于需要转发的分组进行处理，获得下一跳的IP地址，然后调用发送接口函数做进一步处理。

**实验过程：**

### IPv4分组收发实验：

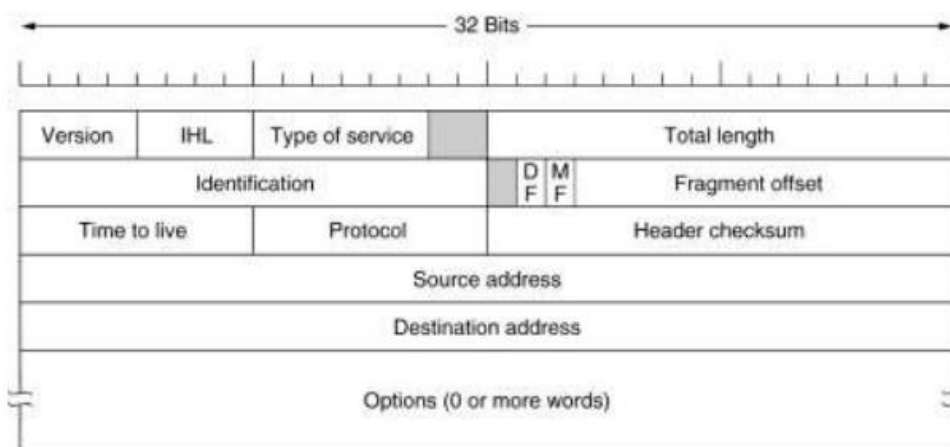
IPv4分组收发实验中涉及到多个函数，其中extern void ip\_DiscardPkt(char\* pBuffer,int type)、extern void ip\_SendtoLower(char\*pBuffer,int length)、extern void ip\_SendtoUp(char\*pBuffer,int length)和extern unsigned int getIpv4Address()函数都已经由系统实现，我们只需要实现数据接收的int stud\_ip\_rcv(char \*pBuffer,unsigned short length)和数据发送的stud\_ip\_Upsend(char \*pBuffer,unsigned short len,unsigned int srcAddr, unsigned int dstAddr,byte protocol,byte ttl)。

#### (1)接收和发送函数的程序流程图：



#### (2)各字段错误检测原理：

IPv4分组头部格式如下：



##### (a)版本号：

版本号在第一个字节的前4位，pBuffer[0]是第1个字节，因此可以用 pBuffer[0] >> 4 来获取版本号，检测是否等于4，如果等于4，则版本号正确，否则错误，需要丢弃。代码如下：

```

int version = pBuffer[0] >> 4; //版本号
if (version != 4){
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_VERSION_ERROR);
    return 1;
}
    
```

## (b)首部长度的:

首部长度的长度是紧跟在版本号后的 4 位, pBuffer[0]是第一个字节, 因此可以用 pBuffer[0] & 0xf 来获取首部长度的长度。如果小于 5, 则代表出错, 需要丢弃, 因为首部本身就会达到长度 5。代码如下:

```
int headLength = pBuffer[0] & 0xf;
if (headLength < 5){
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_HEADLEN_ERROR);
    return 1;
}
```

## (c)生存时间:

生存时间就是 TTL, 在第 9 个字节里, 同上很容易就可以获取该值, TTL = (unsigned short)pBuffer[8]。如果 TTL 的值小于等于 0, 则这个包不能再被转发了, 需要丢弃。代码如下:

```
int TTL = (unsigned short)pBuffer[8];
if (TTL <= 0){
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_TTL_ERROR);
    return 1;
}
```

## (d)首部校验和:

首部校验和在第 11 和 12 个字节, 校验和计算方法: 将首部每 16 位化为一个数, 反码运算求和, 若最高位进位, 则加回末位。代码如下:

```
unsigned short sum = 0;
unsigned short tempSum = 0;
//将首部每 16 位化为一个数, 反码运算求和, 最高位进位加回末位
for (int i = 0; i < headLength * 2; i++){
    tempSum = ((unsigned char)pBuffer[i * 2] << 8) + (unsigned char)pBuffer[i * 2 + 1];
    if (0xffff - sum < tempSum)
        sum = sum + tempSum + 1;
    else
        sum = sum + tempSum;
}
//反码和不为 0xffff 则出错
if (sum != 0xffff){
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_CHECKSUM_ERROR);
    return 1;
}
```

## (3)分组发送初始化原理:

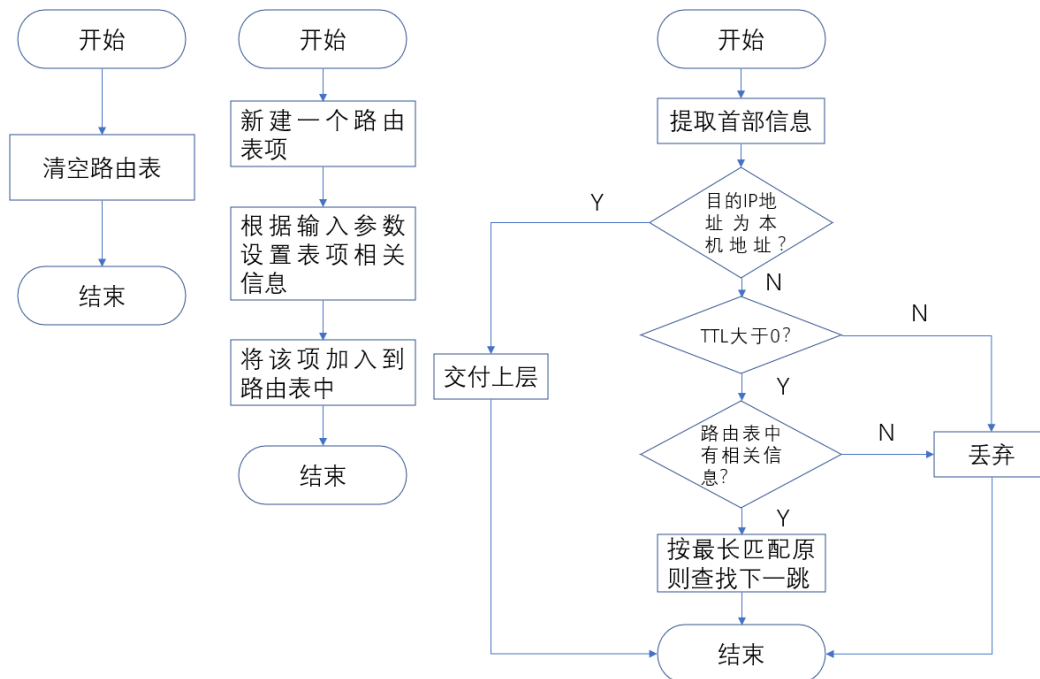
同(2)中提到的获取值的方法一样, 将从上层传来的信息按 IPv4 分组头部格式赋给指定位置, 最后对首部计算校验和 (计算时将校验和部位的值视为全 0) 即可。

## IPv4分组转发实验:

IPv4分组转发实验中涉及到多个函数, 其中extern void fwd\_LocalRcv(char \*pBuffer, int length)、extern void fwd\_SendtoLower(char \*pBuffer, int length, unsigned int nextHop)、extern void fwd\_DiscardPkt(char \*pBuffer, int type)和extern unsigned int getIpv4Address()函数都已经

由系统实现，我们需要设计路由表格式，再实现路由表维护函数void stud\_Route\_Init()和void stud\_route\_add(stud\_route\_msg \*proute)，以及转发处理流程函数int stud\_fwd\_deal(char \*pBuffer, int length)。

(1)路由表初始化、路由增加、路由转发三个函数的实现流程图：



(2)新建数据结构：

新建的数据结构用于实现路由表表项，存储路由信息，主要有3项：目的IP地址、下一跳IP地址和掩码长度，用vector存储各个表项。

每个路由表表项的数据结构如下：

```

struct route_table
{
    int dest; //目的IP地址
    int nexthop; //下一跳地址
    int masklen; //掩码长度
};
    
```

(3)存在大量分组的情况下提高转发效率的方法：

(a)改进数据结构和查找算法，通过提高查找效率来提高转发效率。

(b)引入多线程，同时处理多个转发。

实验结果：



对实验过程中的思考问题进行讨论或回答。

(2)更新checksum操作一定要放在所有有关首部的修改之后,若先计算checksum后再修改其余部分会导致逻辑出错。

通过本次实验，我解了更多的IPv4收发和转发的流程和其中的细节，并且复习了各种数据类型的字节数量，还对大端序、小端序和网络序的关系进行了一次复习，并实践了其转换过程。同时，通过对报文首部信息的读取识别，增强了对于IPv4报文格式的记忆和理解。