



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	林燕燕		院系	人工智能		
班级	1903601		学号	1190200501		
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						

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。

实验内容：**IPv4 分组收发实验：**

根据计算机网络实验系统所提供的上下层接口函数和协议中分组收发的主要流程，独立设计实现一个简单的 IPv4 分组收发模块。要求实现的主要功能包括：

1) IPv4 分组的基本接收处理，能够检测出接收到的 IP 分组是否存在如下错误：校验和错、TTL 错、版本号错、头部长度错、错误目标地址；

2) IPv4 分组的封装发送；

注：不要求实现 IPv4 协议中的选项和分片处理功能

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。

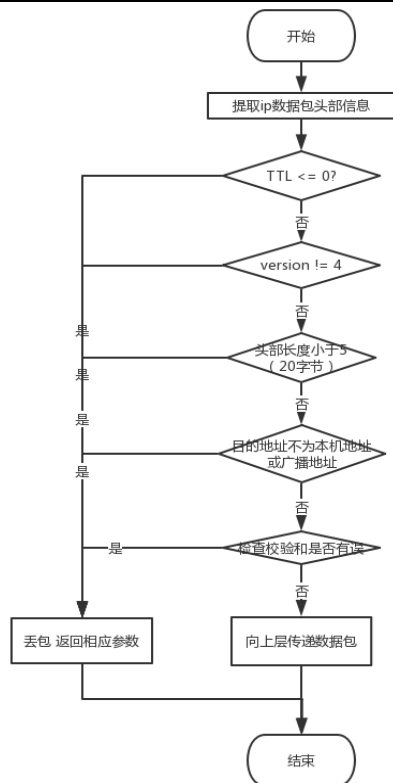
3) IPv4 分组的转发。

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。

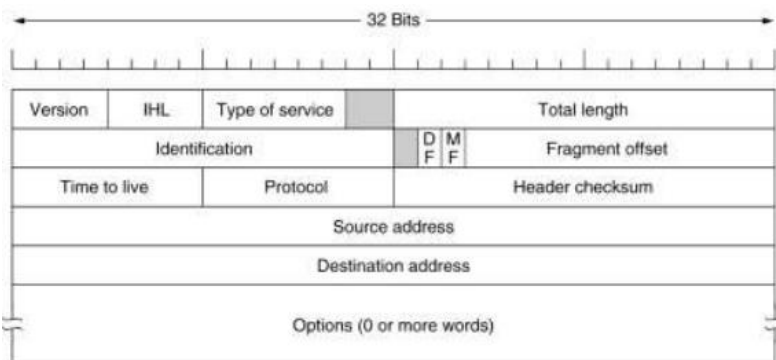
实验过程：**IPv4 分组收发实验：**

发送和接收函数的实现程序流程图：:





头部信息:



版本号 (Version) 错误检测原理: 将第0字节的高4位提取出来, 并且与4进行比较

```

00033     if (version != 4)
00034     {
00035         // 版本号不为4
00036         ip_DiscardPkt(pBuffer, STUD_IP_TEST_VERSION_ERROR);
00037         return 1;
00038     }
    
```

头部长度 (IP Head length) 错误检测原理: 将第0字节的低4位提取出来, 并且与5进行比较

```

00039     if (head_length < 5)
00040     {
00041         // 头部长度低于20字节
00042         ip_DiscardPkt(pBuffer, STUD_IP_TEST_HEADLEN_ERROR);
00043         return 1;
00044     }
    
```

生存时间（Time to live）错误检测原理：提取第8字节，与0进行比较

```

00027     if (ttl == 0)
00028     {
00029         // 生存时间为0，已过期
00030         ip_DiscardPkt(pBuffer, STUD_IP_TEST_TTL_ERROR);
00031         return 1;
00032     }
    
```

头部校验和（Header checksum）字段错误检测原理：

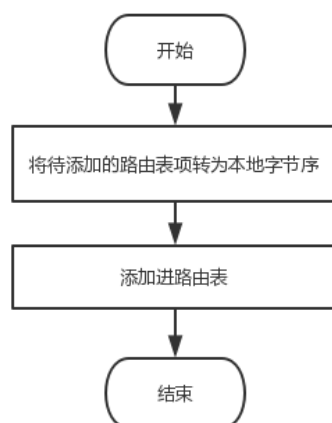
计算头部校验和，若不为一，则错误

```

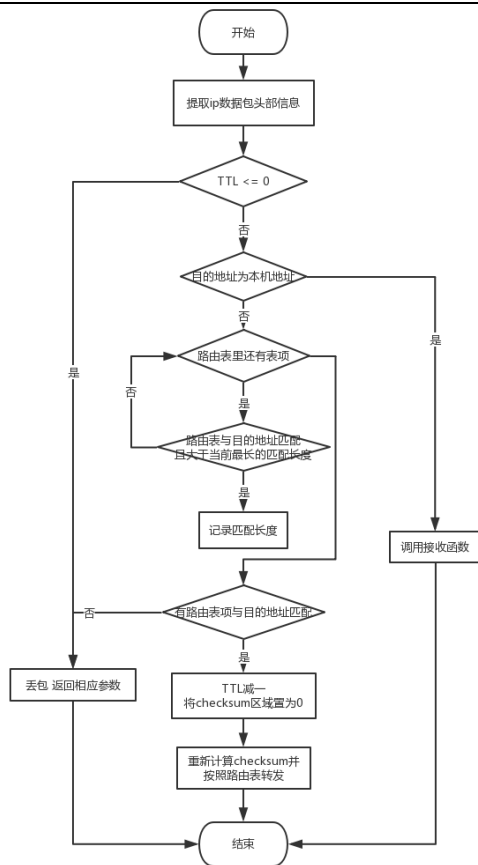
00055     for (i = 0; i < head_length * 2; i++)
00056     {
00057         // 取16位 相加
00058         temp = ((unsigned char)pBuffer[i * 2] << 8) + (unsigned char)pBuffer[i * 2 + 1];
00059         sum += temp;
00060         temp = 0;
00061     }
00062     unsigned short low = sum & 0xffff;
00063     unsigned short high = sum >> 16;
00064     while(high != 0)
00065     {
00066         // 若高16位不为0，将高16位与低16位相加
00067         low += high;
00068         high = low >> 16;
00069     }
00070     sum = low;
00071     if (sum != 0xffff)
00072     {
00073         ip_DiscardPkt(pBuffer, STUD_IP_TEST_CHECKSUM_ERROR);
00074         return 1;
00075     }
    
```

IPv4 分组转发实验：

路由增加实现流程图：



路由转发实现流程图：



路由表初始化函数stud_Route_Init():

```

00020 vector<stud_route_msg> route;
00021
00022 void stud_Route_Init()
00023 { // 初始化路由表
00024     route.clear();
00025     return;
00026 }
  
```

路由增加函数stud_route_add():

```

00028 void stud_route_add(stud_route_msg *proute)
00029 { // 路由增加
00030     stud_route_msg temp;
00031     unsigned int dest = ntohl(proute->dest);
00032     unsigned int masklen = ntohl(proute->masklen);
00033     unsigned int nexthop = ntohl(proute->nexthop);
00034     temp.dest = dest;
00035     temp.masklen = masklen;
00036     temp.nexthop = nexthop;
00037     route.push_back(temp);
00038     return;
00039 }
  
```

处理收到的IP分组的函数stud_fwd_deal():

判断生存时间:

```
00050         if (ttl <= 0)
00051         {
00052             // 生存时间小于0 不能转发 丢弃 IP 分组
00053             fwd_DiscardPkt(pBuffer, STUD_FORWARD_TEST_TTLERROR);
00054             return 1;
00055         }
```

判断目的地址:

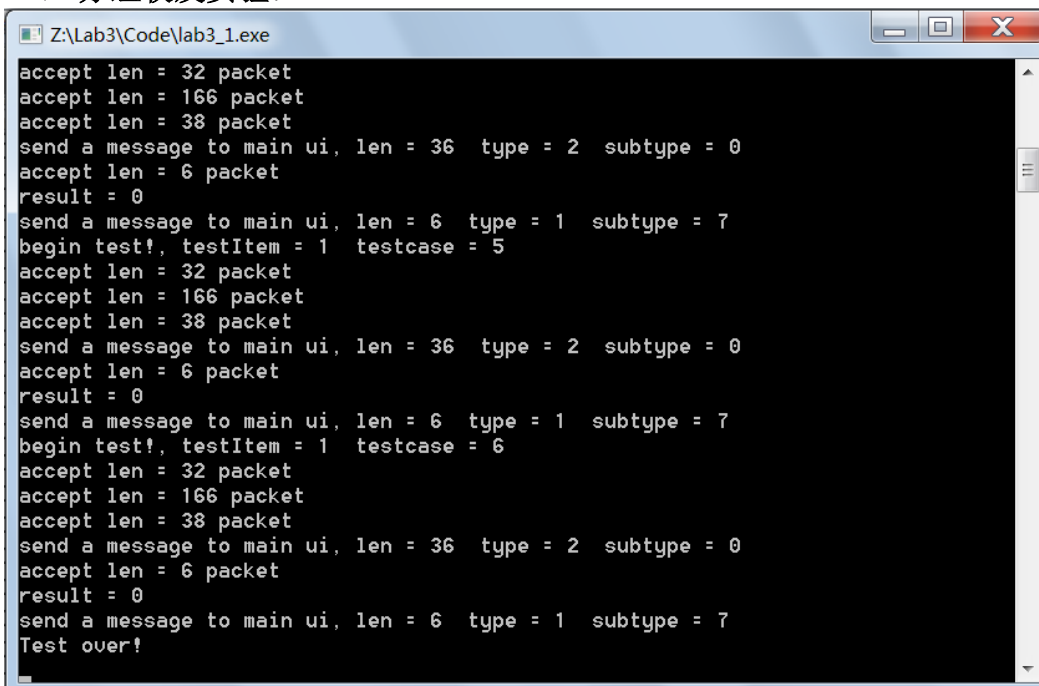
```
00057         if (destination == getIpv4Address())
00058         { // 判断分组地址与本机地址是否相同,上交ip分组
00059             fwd_LocalRcv(pBuffer, length);
00060             return 0;
00061         }
```

判断掩码:

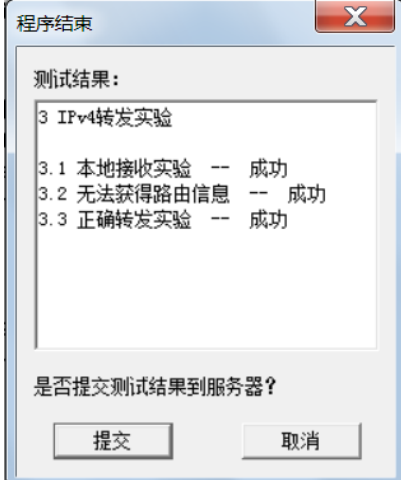
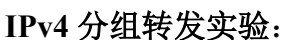
```
00063         stud_route_msg *ans_route = NULL;
00064         int temp_dest = destination;
00065         // 判断掩码是否匹配
00066         for (int i = 0; i < route.size(); i++)
00067         {
00068             unsigned int temp_sub_net = route[i].dest & ((1 << 31) >> (route[i].masklen - 1));
00069
00070             if (temp_sub_net == temp_dest)
00071             {
00072                 ans_route = &route[i];
00073                 break;
00074             }
00075         }
```

实验结果:

IPv4 分组收发实验:



```
Z:\Lab3\Code\lab3_1.exe
accept len = 32 packet
accept len = 166 packet
accept len = 38 packet
send a message to main ui, len = 36 type = 2 subtype = 0
accept len = 6 packet
result = 0
send a message to main ui, len = 6 type = 1 subtype = 7
begin test!, testItem = 1 testcase = 5
accept len = 32 packet
accept len = 166 packet
accept len = 38 packet
send a message to main ui, len = 36 type = 2 subtype = 0
accept len = 6 packet
result = 0
send a message to main ui, len = 6 type = 1 subtype = 7
begin test!, testItem = 1 testcase = 6
accept len = 32 packet
accept len = 166 packet
accept len = 38 packet
send a message to main ui, len = 36 type = 2 subtype = 0
accept len = 6 packet
result = 0
send a message to main ui, len = 6 type = 1 subtype = 7
Test over!
```



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
2021秋季	1	1190200501	林燕燕	人工智能	1903601	IPv4收发实验	2021-11-11	■■■■■■■■■■■■■	10		
2021秋季	2	1190200501	林燕燕	人工智能	1903601	IPv4转发实验	2021-11-12	■■■■■■■■■■■■■	10		

问题讨论：

在IP分组转发实验中，如果存在大量的分组的情况下，如何提高转发效率：

1. 继续使用本次实验中所使用的vector进行效率的提高，可以将路由表进行有序的存储，在vector进行二分查找。
2. 如果改进路由表的数据结构，可以使用平衡树等数据结构，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。

心得体会：

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的认识。