

# 计算机网络LAB2——IPV4收发实验

1400012849 李岩昊

## 1. 实验概述

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；
- 2) IPv4 分组的封装发送；

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。

## 2. 实验实现思路

这次实验的数据结构非常简单，只需要简单的操作将IP头中信息的对应字段拿出来进行操作即可。无论是检查还是封装具体的工作相差不大，重点在于字段的理解和校验和算法的实现。

### 1. 检验Version、TTL、Head Length、Destination Address

将对应字段从Head中取出。Version必须为4，TTL不得为0，Head Length至少为5字节，目标地址要么和本机地址一样要么是广播。如果不符合要求则丢弃Packet并传递对应的错误信息，否则接受Packet。

具体实现细节如下——

```
int version = ((int)pBuffer[0]) >> 4;
if (version != 4){
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_VERSION_ERROR);
    return 1;
}
int ihl = ((int)pBuffer[0]) & 0xf;
if (ihl < 5){
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_HEADLEN_ERROR);
    return 1;
}
int ttl = (int)pBuffer[8];
if (ttl == 0){
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_TTL_ERROR);
    return 1;
}
int dest = ntohl(*((unsigned int*)(pBuffer + 16)));
if (dest != getIpv4Address() && dest != 0xffffffff){
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_DESTINATION_ERROR);
    return 1;
}
```

## 2. 求校验和算法

参考RFC1071这篇Memo的快速求校验和算法求得，阅读这篇英文Memo很好的提高了我对校验和算法的理解。

```
// Fast algo from RFC1071
unsigned int checksum = 0;
for (int i = 0; i < 20; i += 2){
    checksum += (pBuffer[i] & 0xff) << 8;
    checksum += (pBuffer[i + 1] & 0xff);
}
while (checksum >> 16)
    checksum = (checksum >> 16) + checksum;
checksum = (~checksum) & 0xffff;
if (checksum){
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_CHECKSUM_ERROR);
    return 1;
}
```

## 3. 实验中遇到的困难

1. 在计算校验和时错误理解了RFC1071上算法的求校验和部分的实现，经过手动推导得出了正确理解下的算法实现。因此代码中第一部分和第二部分的求校验和算法实现略有不同。
2. 没了。

## 4. 附录代码

见提交tar包中附的代码以及在线测试系统上的代码