



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、转发实验					
姓名	熊峰		院系	计算机科学与技术		
班级	1903104		学号	1190200708		
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	2021.11.12		
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：

本次实验的主要目的：

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。

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****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。

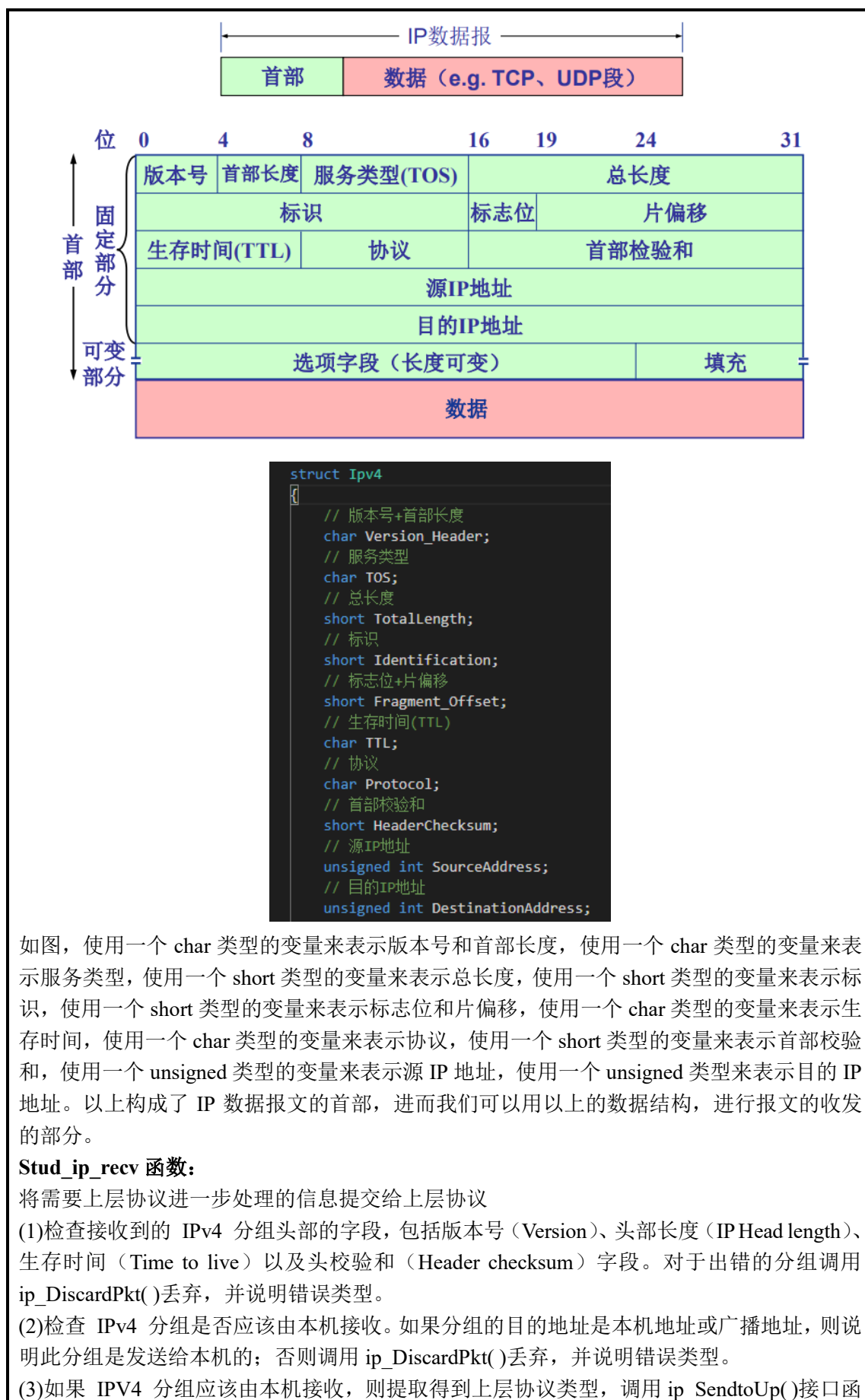
对前面实验（IP 实验）中所完成的代码进行修改，在路由器协议栈的IPv4模块中能够正确完成分组的接收和发送处理。具体要求不做改变，参见“IP 实验”。

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：**

- 1) 根据系统所提供和的上下接口函数和协议中分组收发的主要流程，设计一个简单的 IPv4 报文的结构体，如下所示：



数，交给系统进行后续接收处理。

首先对 pBuffer 处理，强制转换为 IPv4 数据报类型。进而可以判断是否为 ipv4 报文，若不为 ipv4 报文，则调用 ip_DiscardPkt 函数，并抛出 STUD_IP_TEST_VERSION_ERROR 错误；接下来判断首部长度是否小于 5，由于首部需要 $5 \times 4 = 20$ 个字节，因此长度至少大于等于 5，若小于 5，则调用 ip_DiscardPkt 函数，并抛出 STUD_IP_TEST_HEADLEN_ERROR 错误，接下来判断生命周期是否为 0，若生命周期为 0，则调用 ip_DiscardPkt 函数，并抛出 STUD_IP_TEST_TTL_ERROR 错误。接下来判断目的地址是否为本机，若不为本机，则调用 ip_DiscardPkt 函数，并抛出 STUD_IP_TEST_DESTINATION_ERROR。最后判断校验和是否正确，若不正确则调用 ip_DiscardPkt 函数，并抛出 STUD_IP_TEST_CHECKSUM_ERROR 的错误。若全部检查完毕且没有错误，则发送报文。

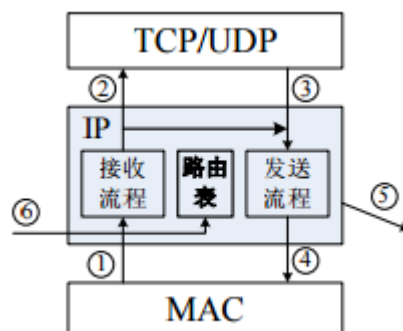
stud_ip_Upsend 函数：

发送接口函数

- (1) 根据所传参数（如数据大小），来确定分配的存储空间的大小并申请分组的存储空间。
- (2) 按照 IPv4 协议标准填写 IPv4 分组头部各字段，标识符（Identification）字段可以使用一个随机数来填写。（注意：部分字段内容需要转换成网络字节序）
- (3) 完成 IPv4 分组的封装后，调用 ip_SendtoLower() 接口函数完成后续的发送处理工作，最终将分组发送到网络中。

根据所给报文长度，申请 len+20 个字节的长度，便于存放 IP 报文的首部，并调用 ipv4 结构体的初始化函数，待初始化后即可将其发送。

IPv4 分组转发实验：



定义路由表结构体：

```
struct route
{
    unsigned low, high;
    unsigned masklen;
    unsigned nextIP;
    route(unsigned low, unsigned high, unsigned masklen, unsigned nextIP)
    {
        this->low = low;
        this->high = high;
        this->masklen = masklen;
        this->nextIP = nextIP;
    }
};
```

stud_Route_Init()函数：

清空路由表。

stud_route_add(stud_route_msg *proute)函数:

完成路由的增加，本函数在系统启动的时候调用，可以初始化路由表。

stud_fwd_deal(char *pBuffer, int length)函数:

1) 查找路由表。

根据相应路由表项的类型来确定下一步操作：错误分组调用函数 fwd_DiscardPkt ()进行丢弃上交分组调用接口函数 fwd_LocalRcv ()提交给上层协议继续处理，转发分组进行转发处理。转发分组还要从路由表项中获取下一跳的 IPv4 地址。

2) 转发处理流程。

对 IPv4 头部中的 TTL 字段减 1, 重新计算校验和, 然后调用下层接口 fwd_SendtoLower() 进行发送处理。

pBuffer: 指向接收到的 IPv4 分组头部

length: IPv4 分组的长度

返回值: 0 为成功, 1 为失败;

本函数是 IPv4 协议接收流程的下层接口函数, 实验系统从网络中接收到分组后会调用本函数。调用该函数之前已完成 IP 报文的合法性检查,

因此本函数中应该实现如下功能:

a.判定是否为本机接收的分组, 如果是则调用 fwd_LocalRcv ();

b.按照最长匹配查找路由表获取下一跳, 查找失败则调用 fwd_DiscardPkt ();

c.调用 fwd_SendtoLower()完成报文发送;

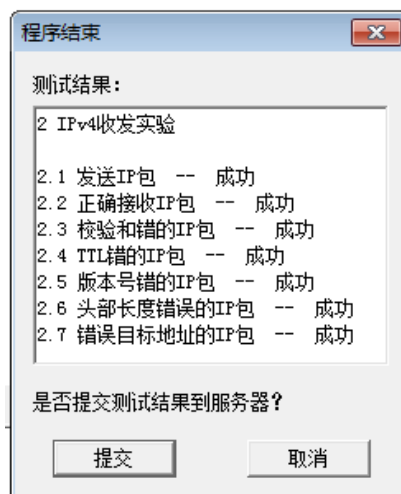
d.转发过程中注意 TTL 的处理及校验和的变化。

首先从 pBuffer 中获取 destIP, 并获取 localIP, 若 destIP 为 0xFFFFFFFF 或者 destIP 等于 localIP, 则判断为本机接收的分组, 调用 fwd_LocalRcv()函数。若不为本机接收的分组, 则此时获取 nextIP, 若 TTL 为 0, 则此时调用 fwd_DiscardPkt() 函数, 并抛出 STUD_FORWARD_TEST_TTLERROR 错误, 表明生命周期已经结束, 否则重新写入 TTL, 接下来检验校验和是否正确, 若无误, 则调用 fwd_SendtoLower()函数, 完成报文发送, 否则调用 fwd_DiscardPkt()函数, 并抛出 STUD_FORWARD_TEST_NOROUTE 错误。

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: 全部成功。

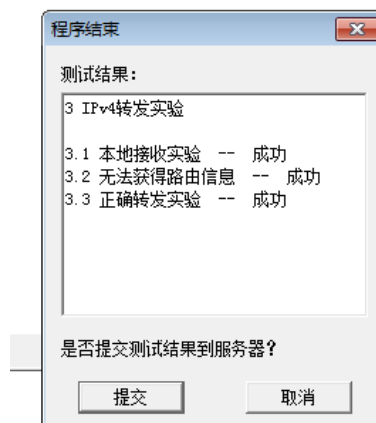


```

C:\Users\Alienware\Desktop\Experiment\Experiment2\Experiment2.exe
result = 0
send a message to main ui, len = 6 type = 1 subtype = 7
begin test!, testItem = 1 testcase = 1
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 = 2
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 = 3
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 = 4
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!
$

```

IPv4 分组转发实验：全部成功



```

C:\Users\Alienware\Desktop\Experiment\Experiment3\Experiment3.exe
send a message to main ui, len = 53  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 = 2  testcase = 1
accept len = 32 packet
accept len = 244 packet
accept len = 41 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 = 2  testcase = 2
accept len = 32 packet
accept len = 244 packet
accept len = 41 packet
accept len = 55 packet
send a message to main ui, len = 53  type = 2  subtype = 0
send a message to main ui, len = 53  type = 2  subtype = 1
accept len = 6 packet
result = 0
send a message to main ui, len = 6  type = 1  subtype = 7
Test over!

```

问题讨论:

对于部分系统处理的部分，了解不够深刻。

心得体会:

对获取IPv4报文，判断丢弃或正确接收，以及如何进行报文转发有了更深刻的体会和了解。