



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			院系	计算机科学与技术		
班级			学号			
任课教师			指导教师			
实验地点			实验时间			
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：

（注：实验报告模板中的各项内容仅供参考，可依照实际实验情况进行修改。）

本次实验的主要目的。

1. 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，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。

2. 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。

1. 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。

2. 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：

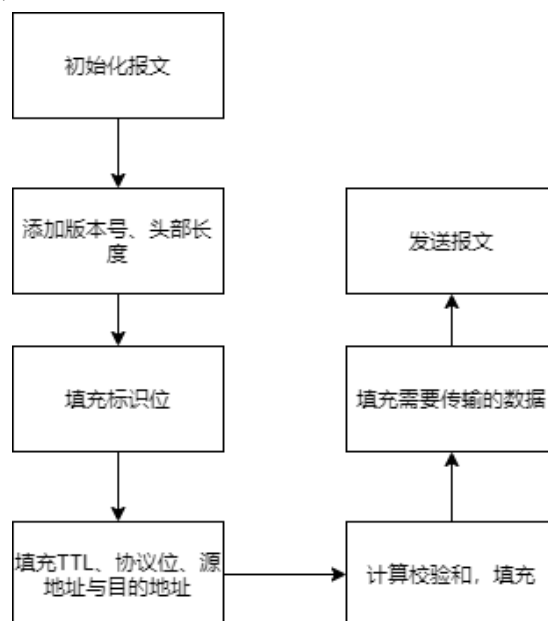
以文字描述、实验结果截图等形式阐述实验过程，必要时可附相应的代码截图或以附件形式提交。

IPv4分组收发实验

发送函数：在接口函数stud_is_Upsend()中，需要完成如下处理：

- 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。

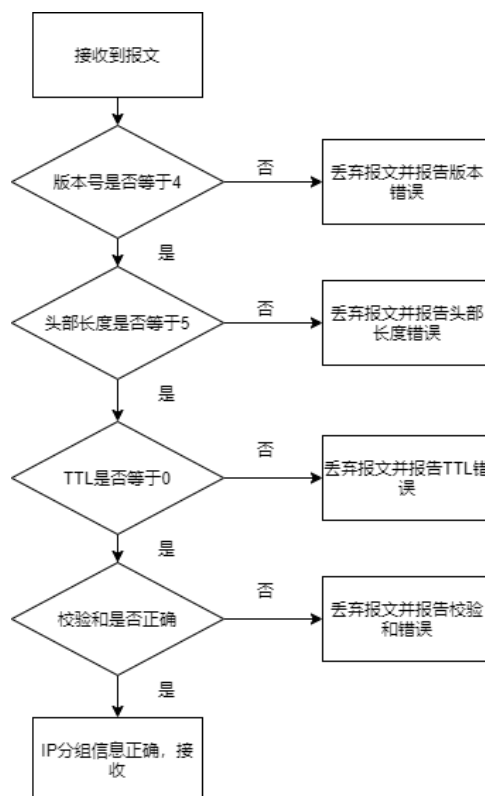
发送函数流程图如下：



接收函数：在接收函数stud_ip_recv()中，需要完成如下处理

- ① 检查接收到的 IPv4 分组头部的字段，包括版本号（Version）、头部长度（IP Head length）、生存时间（Time to live）以及头校验和（Header checksum）字段。对于出错的分组调用 ip_DiscardPkt() 丢弃，并说明错误类型。
- ② 检查 IPv4 分组是否应该由本机接收。如果分组的目的地址是本机地址或广播地址，则说明此分组是发送给本机的；否则调用ip_DiscardPkt()丢弃，并说明错误类型。
- ② 如果IPV4分组应该由本机接收，则提取得到上层协议类型，调用 ip_SendtoUp()接口函数，交给系统进行后续接收处理。

流程图如下：



接收函数错误检测原理:

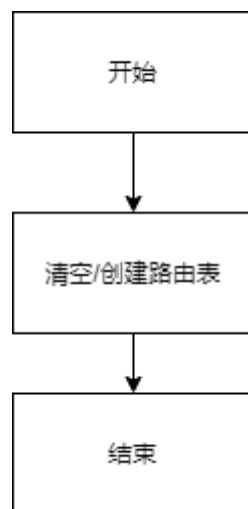
- ① 版本号检测原理: 由于本实验使用的是IPV4, 因此需要检测版本号是否等于4。
- ② 头部长度检测原理: 提取头部长度, 如果小于5则可以判断头部长度错误。
- ③ TTL检测原理: TTL应该是一个正数, 如果提取出的 $TTL \leq 0$, 可以判断TTL错误
- ④ 头部校验和检测原理: 将IP报文头部信息使用十六进制表示, 并两两组合相加, 对于相加的结果如果大于0xffff, 则将大于0xffff的部分和小于0xffff的部分相加作为最后结果, 如果最后的结果等于0xffff, 则正确, 否则错误。

各种错误信息对应的实验中出现的真实数据可见实验结果部分。

IPv4分组转发实验

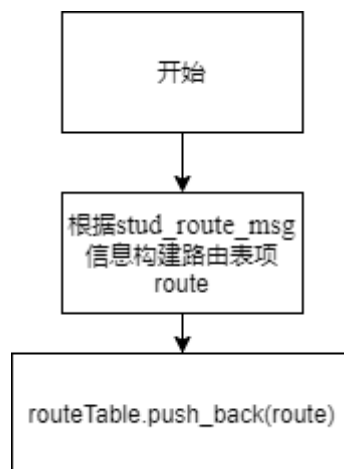
路由表初始化、路由增加、路由转发三个函数的实现流程图

路由表初始化函数:

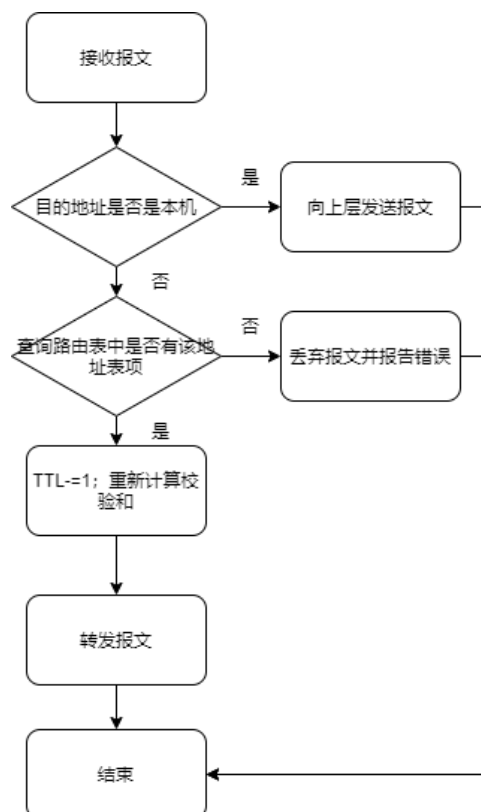


路由增加函数

从stud_route_msg结构中取得dest, masklen, nexthop, 转为网络字节序之后经过处理, 构建结构体route, 并添加到vector中。



路由转发函数:



所新建数据结构的说明

结构体中共有四项, 第一项是掩码值, 是根据掩码长度计算出的, 这也是唯一一个在创建的时候需要计算的项; 第二项是目的地址, 创建的时候直接从stud_route_msg获取并转换即可; 第三项是掩码长度, 用于计算掩码; 第四项是下一跳地址, 在创建路由表项的时候直接获取转换即可。详细数据结构代码如下:

1. `struct Route`//路由表表项
2. {

```
3.     unsigned int mask;//掩码
4.     unsigned int dest;//目的地址
5.     unsigned int masklen;//掩码长度
6.     unsigned int nexthop;//下一跳地址
7.
8.     Route(unsigned int dest,unsigned int masklen,unsigned int nexthop)
9.     {
10.        this->dest=ntohl(dest);
11.        this->masklen=ntohl(masklen);
12.        this->nexthop=ntohl(nexthop);
13.        this->mask=0;
14.        if(this->masklen)
15.        {
16.            this->mask = (int)0x80000000 >> (this->masklen - 1);
17.        }
18.    }
19.};
```

请分析在存在大量分组的情况下如何提高转发效率，如果代码中有相关功能实现，请给出具体原理说明。

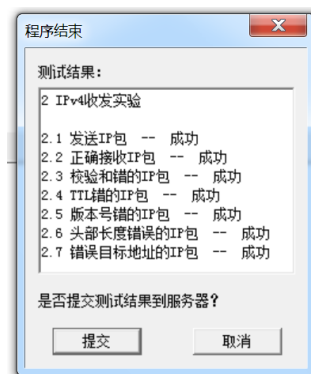
- 1.存储结构由线性结构转换为树形结构，检索过程中利用树形结构性质，提高匹配效率。
- 2.由于转发分组过程中对于每个分组的操作都类似，因此可以使用硬件来创建并行检查与转发过程。

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

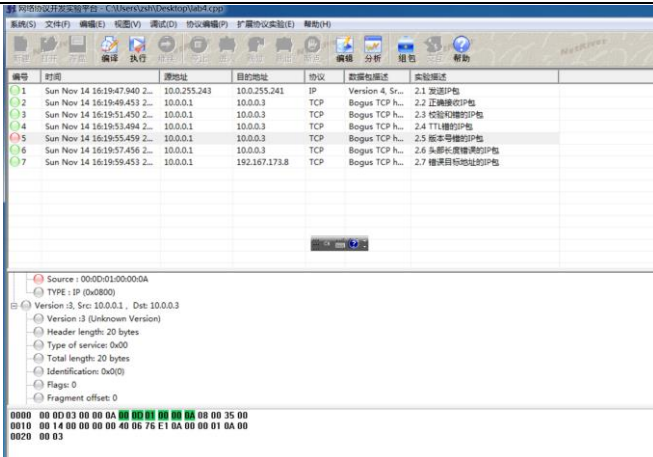
实验结果展示：



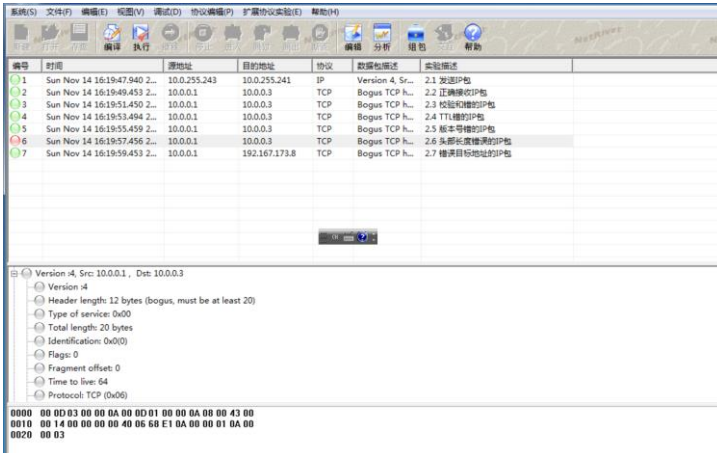
错误信息展示：

版本号错误：

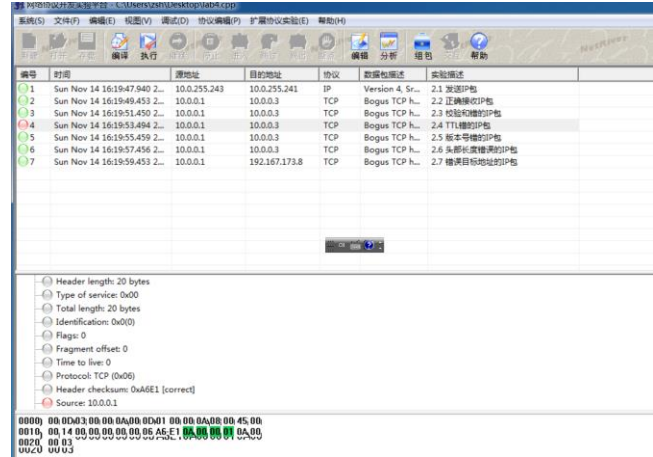
版本号应该是4，但是发送的信息中版本号是3



头部长度的错误：
最小的头部的长度是20字节，但是发送的信息只有12字节，报错



TTL的错误：
TTL=0，错误



头部校验和的错误：
头部校验和应该是0x4AFC，但是发送的信息中是0x03E8

编号	时间	源地址	目的地址	协议	数据包描述	实验描述
1	Sun Nov 14 16:19:47.940 Z...	10.0.255.243	10.0.255.241	IP	Version 4, Sr...	2.1 发送IP包
2	Sun Nov 14 16:19:48.453 Z...	10.0.0.1	10.0.0.3	TCP	Bogus TCP h...	2.2 正确接收IP包
3	Sun Nov 14 16:19:51.450 Z...	10.0.0.1	10.0.0.3	TCP	Bogus TCP h...	2.3 校验和错误的IP包
4	Sun Nov 14 16:19:53.494 Z...	10.0.0.1	10.0.0.3	TCP	Bogus TCP h...	2.4 TTL错误的IP包
5	Sun Nov 14 16:19:55.459 Z...	10.0.0.1	10.0.0.3	TCP	Bogus TCP h...	2.5 版本号错误的IP包
6	Sun Nov 14 16:19:57.456 Z...	10.0.0.1	10.0.0.3	TCP	Bogus TCP h...	2.6 头部长度错误的IP包
7	Sun Nov 14 16:19:59.453 Z...	10.0.0.1	192.167.173.8	TCP	Bogus TCP h...	2.7 错误目标地址的IP包

Type of service: 0x00
Total length: 20 bytes
Identification: 0x0(0)
Flags: 0
Fragment offset: 0
Time to live: 64
Protocol: TCP (0x06)
Header checksum: 0x03E8[incorrect, should be 0xA4FC]
Source: 10.0.0.1
Destination: 10.0.0.3

0000 00 0D 03 00 00 0A 00 0D 01 00 00 0A 08 00 45 00
0010 00 14 00 00 00 00 40 06 03 E8 00 00 00 0A 00
0020 00 03

IPv4分组转发实验

实验结果展示：

程序结束

测试结果：

3 IPv4转发实验

3.1 本地接收实验 -- 成功

3.2 无法获得路由信息 -- 成功

3.3 正确转发实验 -- 成功

是否提交测试结果到服务器？

提交

取消

问题讨论：

对实验过程中的思考问题进行讨论或回答。

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：

这一点在实验过程部分就简单讨论过，在此展开讨论。

1.首先最直接的一点就是改进存储的结构，在本实验中使用的是线性结构，因此查询的时间复杂度就是O(n)。如果想把查询时间优化到O(logn)，可以根据IP目的地址进行有序存储，在查询的时候使用二分查找，这样查询时间就可以优化到O(logn)，如果希望查询时间复杂度为O(1)，可以将路由表设置为一个哈希表，使用空间换取时间。

2.如果不考虑从查询时间入手，那么由于存在大量分组，因此可以并行转发，所有的检测、匹配等过程都可以并行处理，这可以从硬件层面入手处理。

心得体会：

结合实验过程和结果给出实验的体会和收获。

经过本实验，对于IPV4的报文结构与检测、转发等功能有了更深入的认识，也对于路由表的建立、维护和工作过程有了较为深入的认识。