

# 哈尔滨工业大学

## <<计算机网络>>

### 实验报告

(2018 年度春季学期)

姓名:	许家乐
学号:	1150310329
学院:	计算机学院
教师:	李全龙

## 实验四 IPv4 分组转发实验

### 一、实验目的

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。

### 二、实验内容

#### (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”。

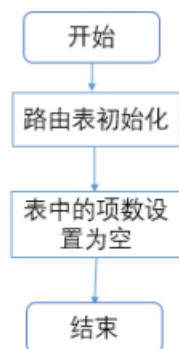
#### (3) IPv4 分组的转发。

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。

### 三、实验过程及结果

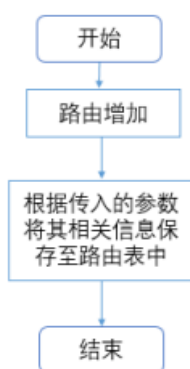
#### 1. 主要函数流程图

##### (1) 路由表初始化



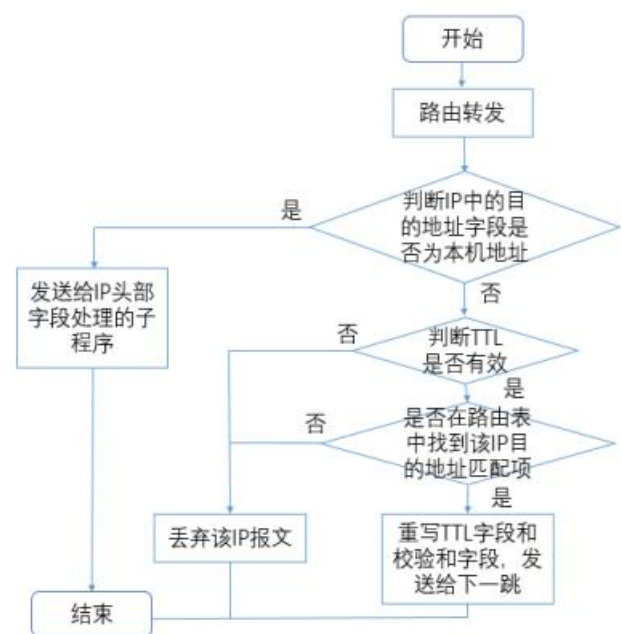
路由表的初始化设置相对简单，只是将路由表清空即可。

(2) 路由表添加项



根据系统已经规定的参数进行传入，将其相关的信息保存到路由表中，例如目的地址、子网掩码的长度、子网掩码的值以及下一跳的位置。其中子网掩码的值可以通过移位运算和子网掩码的长度计算得到。

(3) 路由转发



本地获取到 IP 报文后，首先对头部字段提取出 TTL 和目的 IP 地址，如果 TTL

小于 0，那么直接调用 fwd\_DiscardPkt() 丢弃该 IP 报文，同时判断目的 IP 地址是否为本机地址，如果是本机地址，则调用本地 IP 的处理判断子程序 fwd\_LocalRcv() 进行其头部的深度判断，如果目的地址不是本机地址，则根据此目的地址寻找路由表看是否能够找到匹配项，如果找到匹配项，则修改 IP 头部字段中 TTL 的值并且重写校验和，再将修改过的 IP 报文发送给数据链路层，用于在网络中传输。如果没有在路由表中没有找到匹配项，则丢弃该 IP 报文。

## 2. 新建数据结构的说明

新建的数据结构主要就是路由表这个数据结构，主要用于存贮其中每一项的信息，包括目的 IP 地址、子网掩码、下一跳地址。

```
struct RNode
{
    int dest;           //目标地址
    int masklen;        //掩码长度
    int nexthop;        //下一跳地址
};
```

## 3. 大量分组的情况下如何提高转发效率

①路由聚合，节省路由表空间的同时，缩短寻找下一跳的时间，否则不仅存在大量“前缀”相同且下一跳也相同的路由表项，而且在搜索下一跳时还需要进行更多步的匹配，造成时间和空间的浪费。在代码中实现，路由表中记录目标 IP 地址的网络地址。

②多线程工作同时查找路由表。分组到来后分配到不同线程，各线程同时搜索路由表，显然会提高时间效率。

③路由表存储结构由线性结构改为树形结构，可采用 B+树，提高匹配效率。

## 4. 代码

```
#include "sysInclude.h"
#include <vector>
#include <algorithm>

using std::vector;

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);
extern unsigned int getIpv4Address();

struct RNode
{
    int dest;
    int masklen;
    int nexthop;
    RNode(int d = 0, int m = 0, int n = 0) :
        dest(d), masklen(m), nexthop(n)
    {}
};
```

```
vector<RNode> routeTable;

void stud_Route_Init()
{
    routeTable.clear();
    return;
}

bool cmp(const RNode & a, const RNode & b)
{
    if (htonl(a.dest) > htonl(b.dest))
    {
        return true;
    }
    else if (htonl(a.dest) == htonl(b.dest))
    {
        return htonl(a.masklen) > htonl(b.masklen);
    }
    else
    {
        return false;
    }
}

void stud_route_add(stud_route_msg *proute)
{
    int dest;
    routeTable.push_back(RNode(ntohl(proute->dest), ntohl(proute->masklen),
    ntohl(proute->nexthop)));
    sort(routeTable.begin(), routeTable.end(), cmp);
    return;
}

int stud_fwd_deal(char *pBuffer, int length)
{
    int version = pBuffer[0] >> 4;
    int ihl = pBuffer[0] & 0xf;
    int ttl = (int)pBuffer[8];

    int dstIP = ntohl(*(unsigned int*)(pBuffer + 16));

    if (dstIP == getIpv4Address())
```

```
{
    fwd_LocalRcv(pBuffer, length);
    return 0;
}

if (ttl <= 0)
{
    fwd_DiscardPkt(pBuffer, STUD_FORWARD_TEST_TTLERROR);
    return 1;
}

for (vector<RNode>::iterator ii = routeTable.begin(); ii != routeTable.end(); ii++)
{
    if (ii->dest == dstIP)
    {
        char *buffer = new char[length];
        memcpy(buffer, pBuffer, length);
        buffer[8]--;
        int sum = 0;
        unsigned short int localChecksum = 0;
        for (int i = 0; i < 2 * ihl; i++)
        {
            if (i == 5)
                continue;
            sum = sum + (buffer[i * 2] << 8) + (buffer[i * 2 + 1]);
            sum %= 65535;
        }
        localChecksum = htons(0xffff - (unsigned short int)sum);
        memcpy(buffer + 10, &localChecksum, sizeof(short unsigned int));
        fwd_SendtoLower(buffer, length, ii->nextHop);
        return 0;
    }
}

fwd_DiscardPkt(pBuffer, STUD_FORWARD_TEST_NOROUTE);
return 1;
}
```

## 四、实验心得

通过本次实验，我有以下几点收获：

- ① 在课堂知识的基础上进一步掌握了有关路由转发表的知识，对路由转发有了更深刻的理解。

- ② 了解了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的过程。
- ③ 初步接触了路由表这一重要的数据结构，理解了路由器如何根据路由表对分组进行转发。