



南开大学
Nankai University

南 开 大 学

网 络 空 间 安 全 学 院

网络技术与应用实验报告

路由器实验

李潇逸 2111454

年级：2021 级

专业：信息安全、法学

指导教师：张建忠

2023 年 12 月 21 日

目录

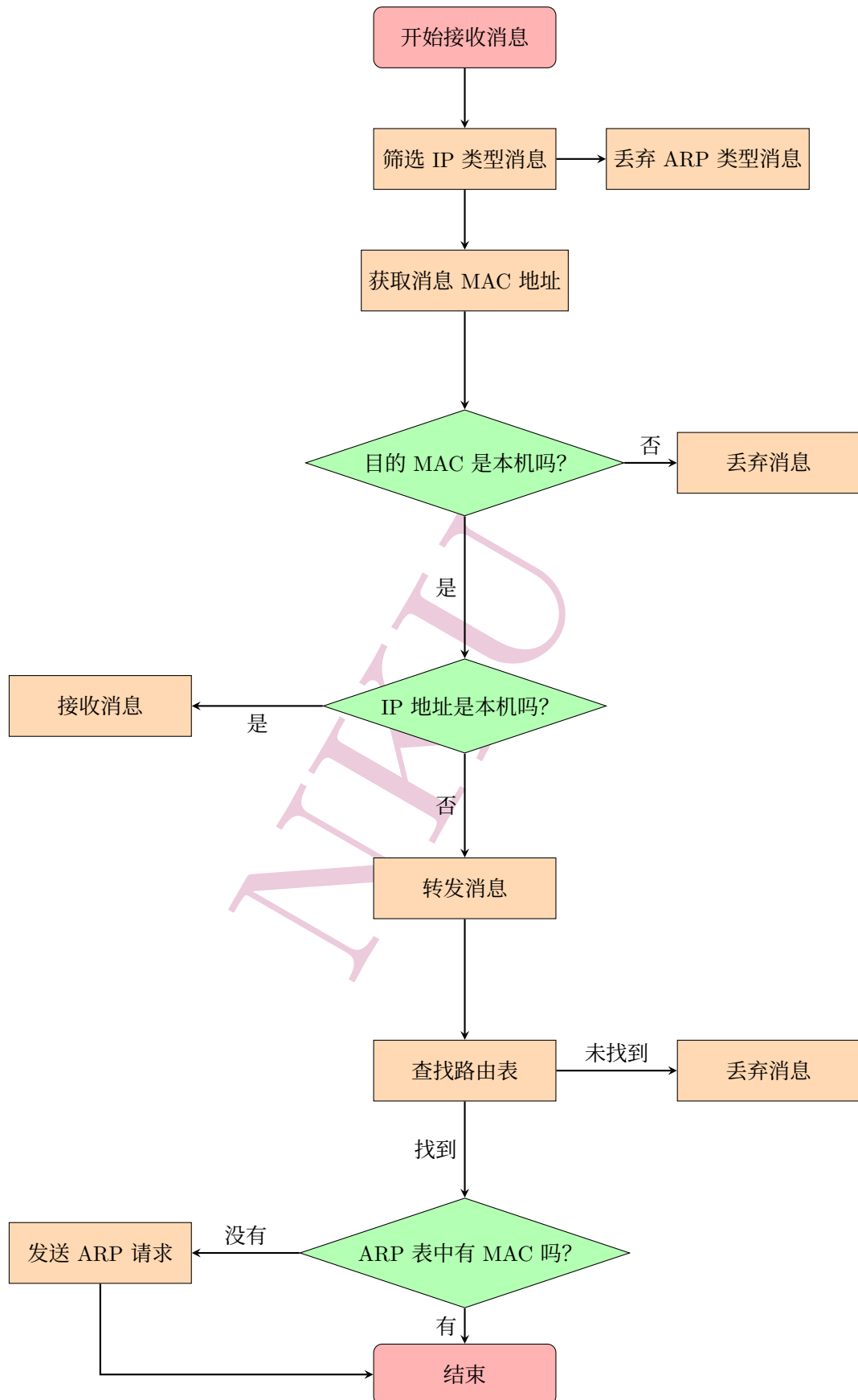
一、 实验要求	1
二、 实验步骤	1
三、 代码分析	3
(一) 路由表	3
(二) ARP 表	5
四、 结果展示	6
五、 总结	6

一、 实验要求

- (1) 设计和实现一个路由器程序，要求完成的路由器程序能和现有的路由器产品（如思科路由器、华为路由器、微软的路由器等）进行协同工作。
- (2) 程序可以仅实现 IP 数据报的获取、选路、投递等路由器要求的基本功能。可以忽略分片处理、选项处理、动态路由表生成等功能。
- (3) 需要给出路由表的手工插入、删除方法。
- (4) 需要给出路由器的工作日志，显示数据报获取和转发过程。
- (5) 完成的程序须通过现场测试，并在班（或小组）中展示和报告自己的设计思路、开发和实现过程、测试方法和过程。

二、 实验步骤

1. 获取本机的网卡列表及相应的信息，打开网卡并进行监听
2. 在接收消息中，筛选 IP 类型的消息，丢弃 ARP 类型。获取消息的 MAC 地址，如果消息的目的 MAC 地址不是本机，则丢弃，如果目的 MAC 地址以及目的 IP 地址全部指向自己，则接收，如果目的 MAC 地址是本机，但 IP 地址不是本机，则转发。之后查找路由表的对应的下一步 IP，若没找到，则丢弃；若找到，则进行下一步。之后查看 ARP 表，看是否有下一跳 IP 地址的 MAC 地址，若没有，则发送 ARP 请求获取 MAC 地址，若有，直接返回。



3. 组装报文，将 SrcMAC 变为本机的 MAC，将 DesMAC 变为下一跳的 MAC。若 TTL=0，则返回报文，不转发。若 TTL>0，则-1，重新计算校验和。

4. 发送消息, 进行循环

三、 代码分析

(一) 路由表

路由表

```
1 class RouterTable//路由表
2 {
3 public:
4     RouterItem* head;
5     RouterItem* tail;
6     int num;//条数
7     RouterTable()//初始化, 添加直接相连的网络
8     {
9         head = new RouterItem;
10        tail = new RouterItem;
11        head->nextitem = tail;
12        num = 0;
13        for (int i = 0; i < 2; i++)
14        {
15            RouterItem* temp = new RouterItem;
16            temp->net = (inet_addr(ip[i])) & (inet_addr(mask[i]))
17                ;//本机网卡的ip和掩码进行按位与即为所在网络
18            temp->mask = inet_addr(mask[i]);
19            temp->type = 0;//0表示直接连接, 不可删除
20            this->RouterAdd(temp);
21        }
22    void RouterAdd(RouterItem* a)//路由表的添加
23    {
24        RouterItem* pointer;
25        if (!a->type)
26        {
27            a->nextitem = head->nextitem;
28            head->nextitem = a;
29            a->type = 0;
30        }
31        else//按照掩码由长至短找到合适的位置
32        {
33            for (pointer = head->nextitem; pointer != tail &&
34                pointer->nextitem != tail; pointer = pointer->
35                nextitem)
36            {
37                if (a->mask < pointer->mask && a->mask >=
38                    pointer->nextitem->mask || pointer->
39                    nextitem == tail)
```

```
36         {
37             break;
38         }
39     }
40     a->nextitem = pointer->nextitem;
41     pointer->nextitem = a;
42 }
43 RouterItem* p = head->nextitem;
44 for (int i = 0; p != tail; p = p->nextitem, i++)
45 {
46     p->index = i;
47 }
48 num++;
49 }
50 void RouterRemove(int index)//路由表的删除
51 {
52     for (RouterItem* t = head; t->nextitem != tail; t = t->
53         nextitem)
54     {
55         if (t->nextitem->index == index)
56         {
57             if (t->nextitem->type == 0)
58             {
59                 //SetColor(12, 0);
60                 printf("该项不可删除\n");
61                 return;
62             }
63             else
64             {
65                 t->nextitem = t->nextitem->nextitem;
66                 return;
67             }
68         }
69     }
70     SetColor(12, 0);
71     printf("无该表项\n");
72 }
73 void print()
74 {
75     for (RouterItem* p = head->nextitem; p != tail; p = p->
76         nextitem)
77     {
78         p->PrintItem();
79     }
80 }
81 DWORD RouterFind(DWORD ip)//查找最长前缀, 返回下一跳的ip
82 {
83     for (RouterItem* t = head->nextitem; t != tail; t = t->
```

```

82         nextitem)
83     {
84         if ((t->mask & ip) == t->net)
85         {
86             return t->nextip;
87         }
88     }
89     return -1;
90 };

```

在这里我使用链表对路由表项进行存储,在对路由表项进行插入时首先对其掩码长度进行判断,按照由长到短的顺序依次排列,在查询时就可以按照此逻辑进行查找。

(二) ARP 表

ARP 表

```

1  class ArpTable//ARP表
2  {
3  public:
4      DWORD ip;
5      BYTE mac[6];
6      static int num;
7      static void InsertArp(DWORD ip, BYTE mac[6])
8      {
9          arptable[num].ip = ip;
10         getMac(ip, arptable[num].mac);
11         memcpy(mac, arptable[num].mac, 6);
12         num++;
13     }
14     static int FindArp(DWORD ip, BYTE mac[6])
15     {
16         memset(mac, 0, 6);
17         for (int i = 0; i < num; i++)
18         {
19             if (ip == arptable[i].ip)
20             {
21                 memcpy(mac, arptable[i].mac, 6);
22                 return 1;
23             }
24         }
25         return 0;
26     }
27 }arptable[50];

```

ARP 表中存储 IP 地址和其对应的 MAC 地址,在这里由于 ARP 数据量并不多,因此直接采用数组的形式进行存储,在查找时也是依次查找。然而,这种做法忽视了数组的越界问题,当 ARP 数量增大时很容易造成数据泄露,因此可以尝试换成链表。

四、 结果展示

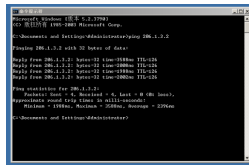


图 1: 结果展示

五、 总结

通过本次实验，我对网络技术与应用这门课程有了更为深刻、系统的理解与认识，对于网络编程有了更深的学习。同时，通过路由器编程，对于路由器相关知识点、IP 数据、ARP 数据等知识点都有了更深的学习。此外，本实验包含了对之前几次实验的复习，能够系统性的复习本学期的课程知识。

参考文献

- [1] 李潇逸. github 作业链接. https://github.com/SesameZMT/NKU_NetworkTechnology.git.

NKU