

网络技术路由器程序

朱子轩

计算机科学与技术专业

2023 年 12 月 7 日



① 设计思路及实现方法

② 测试方法

① 设计思路及实现方法

② 测试方法

路由表

路由表中包含以下内容：

- 网段
- 子网掩码
- 下一跳
- 是否为直接投递

```
typedef struct RouterTable_t {  
    DWORD IP;  
    DWORD Mask;  
    DWORD Next;  
    WORD flag;  
} RouterTable_t;
```

ARP 缓存

ARP 缓存中包含以下内容:

- 建立时间
- 保存时长
- IP 地址
- MAC 地址

```
typedef struct ARPTable_t {  
    clock_t time;  
    int keep;  
    DWORD IP;  
    BYTE Mac[6];  
} ARPTable_t;
```

路由表查找

查询路由表的步骤如下：

- 遍历 vector，查找网络和掩码的与是否和网络号一致。
- 查询到一致的数据后，记录掩码长度，判断和上一次的掩码的大小。如果掩码更长，则保留当前查询项。
- 遍历结束，返回匹配项。

ARP 缓存查找

查询 ARP 缓存的步骤如下：

- 遍历 ARP 缓存，查找 IP 地址。
- 如果存在，则判断时间是否过期，，如果没过期则返回 true，保存 MAC 地址。
- 其余情况返回 false。

ARP 请求

本次实验主要包括对 ARP 请求和 IP 数据报的处理。对于 ARP 请求，需要经过以下步骤：

- 判断是否为 ARP 响应。
- 如果是，将 IP 和 MAC 地址加入 ARP 缓存中，记录加入缓存的时间，设置有效时间。

ARP 请求

```
ARPFrame_t* ARPFrame = (ARPFrame_t*)packet;
int op = ARPFrame->Operation;
if (op == ntohs(2))
{
    ARPTable_t t;
    t.IP = ARPFrame->SendIP;
    memcpy(t.Mac, ARPFrame->SendHa, 6);
    t.time = clock();
    t.keep = 300000;
    ARPTable.push_back(t);
    char output[INET_ADDRSTRLEN];
    inet_ntop(AF_INET, &t.IP, output, INET_ADDRSTRLEN);
    string MACstr = format("{:02X}:{:02X}:{:02X}:{:02X}:{:02X}:{:02X}", ARPFrame->SendHa[0],
    cout << "收到ARP报文响应, 添加ARP表项: " << output << " " << MACstr << endl;
    continue;
}
```

IP 数据报

对于 IP 数据报，需要经过以下步骤：

- 校验并检查以太帧头部信息中的目的 MAC 地址是否为本机 MAC 地址，如果不是，则丢弃。
- 查询路由表项，如果是直接投递，则应将 MAC 地址设置为目的 MAC 地址；如果不是，则应将 MAC 地址设置为下一跳的 MAC 地址。
- 查询 MAC 地址，如果存在，则填入，将 TTL 减一，重新计算头部检验和，转发数据报；如果不存在，发送 ARP 请求。

IP 数据报

```
if (choose.flag == RO_DIRECT)
{
    bool ret = getMAC(IPPacket->IPHeader.DstIP, IPPacket->FrameHeader.DesMAC);
    if(ret)
        pcap_sendpacket(adhandle, (u_char*)packet, sizeof(FrameHeader_t) + len);
    else
    {
        MACWait.push_back(MACWait_t(IPPacket->IPHeader.DstIP, packet));
    }
    if (ret) cout << "获取MAC地址成功 ";
    cout << "直接投递 ";
}
else if(compMAC(IPPacket->FrameHeader.DesMAC, deviceMAC))
{
    bool ret = getMAC(choose.Next, IPPacket->FrameHeader.DesMAC);
    if(ret)
        pcap_sendpacket(adhandle, (u_char*)packet, sizeof(FrameHeader_t) + len);
    else
    {
        MACWait.push_back(MACWait_t(choose.Next, packet));
    }
    if (ret) cout << "获取MAC地址成功 ";
    cout << "路由转发 ";
}
```

IP 数据报传输分析

对于主机 A ping 主机 B，经过以下步骤：

- 添加路由表项，两个直连网段，一个路由转发。
- 第一次发送，路由器 A 收到主机 A 的消息，查找路由表，得到下一跳。查询 ARP 缓存，不存在该项，则发送 ARP 广播，请求路由器 B 的 MAC 地址。
- 第二次发送，路由器 A 收到主机 A 的消息，转发给路由器 B。主机 B 回应后，查找路由表，发现是直接发送给主机 A，查询 ARP 缓存，不存在该项，则发送 ARP 广播，请求主机 A 的 MAC 地址。
- 第三四次发送，正常转发。

① 设计思路及实现方法

② 测试方法

测试方法

添加直接投递和转发的路由表项，在虚拟机上运行，在需要 ping 的主机上运行 wireshark。由于在虚拟机中，四台虚拟机采取桥接网络，在 vmware 中使用的是相同的网卡，所以可以直接看到所有的转发过程。

查看 wireshark 是否能识别转发过程中发送的数据报，并且各主机是否能正确响应。

Thanks!