**题目：网络层ARP协议设计与实现**

**姓名：王怡恒**

**学号：2022300385**

**班号：10012201**

**时间：2024/11/17**

**计算机学院**

**目 录**

[1、实验目的 1](file:///D:\\qq\\3229804499\\filerecv\\西北工业大学试点班实验报告模板.doc" \l "_Toc341302685)

[2、实验要求 1](file:///D:\\qq\\3229804499\\filerecv\\西北工业大学试点班实验报告模板.doc" \l "_Toc341302686)

[3、实现原理及流程图 1](file:///D:\\qq\\3229804499\\filerecv\\西北工业大学试点班实验报告模板.doc" \l "_Toc341302688)

[4、算法设计 4](file:///D:\\qq\\3229804499\\filerecv\\西北工业大学试点班实验报告模板.doc" \l "_Toc341302689)

[5、运行结果与分析 5](file:///D:\\qq\\3229804499\\filerecv\\西北工业大学试点班实验报告模板.doc" \l "_Toc341302690)

6、实验过程发现的新技术问题及建议................................8

[7、参考文献 9](file:///D:\\qq\\3229804499\\filerecv\\西北工业大学试点班实验报告模板.doc" \l "_Toc341302691)

## 实验目的

1. 理解网络协议栈结构：了解网络协议栈的基本构成及各层功能。
2. 掌握ARP协议设计：学习ARP协议如何实现IP地址到MAC地址的映射，以及其在不同情况下的应用。
3. 实现IP分组传输：通过ARP协议实现两台计算机之间的IP分组传输，理解不同网络环境下的传输机制。

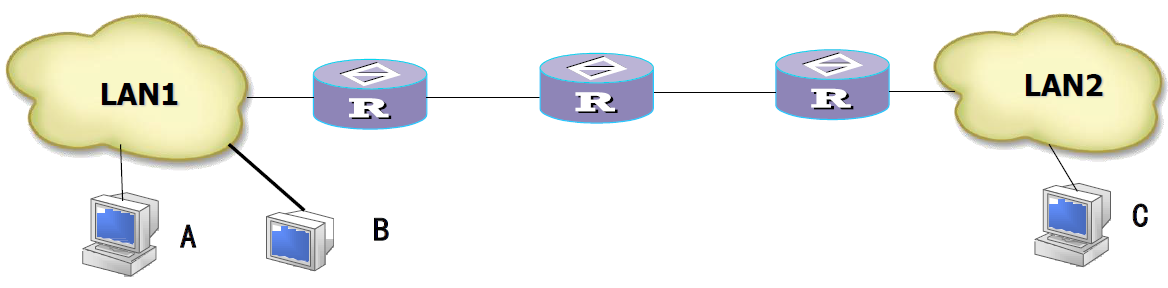
## 实验要求

1. 源代码：编写实现ARP协议的源代码。
2. 可执行代码：确保编写的代码能够正确编译和运行。
3. 实验报告：撰写实验报告，详细记录实验过程、遇到的问题及解决方法等。
4. 提交格式：所有材料需压缩成一个文件，命名规则为“学号-姓名-第3次实验”。

## 实验原理

1. ARP协议设计任务

基本任务：实现IP地址到MAC地址的映射。



情况1：如果源IP与目的IP在同一网段，则实现目的IP到目的MAC地址的映射。

情况2：如果源IP与目的IP不在同一网段，则获得网关IP地址到MAC地址的映射。

1. A发送IP分组给B计算机

确定网络地址：A计算机利用自己的子网掩码与A和B的IP地址进行“与”运算，获得两个网络地址。

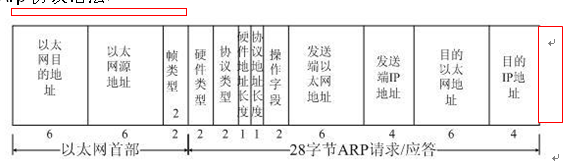
判断是否同网段：

同网段：检查ARP缓存中是否存在目的B的IP地址对应的MAC地址，如果存在则直接使用；否则，发送ARP请求获得目的MAC地址。

不同网段：查询路由表，找到下一跳IP地址（通常是网关IP地址），检查ARP缓存中是否存在网关IP地址对应的MAC地址，如果存在则直接使用；否则，发送ARP请求获得网关MAC地址。

1. ARP协议设计

ARP协议语法：



硬件类型：0x0001，表示以太网，2字节。

协议类型：0x0800，表示上层协议类型，2字节。

硬件地址长度：6，1字节。

协议地址长度：4，1字节。

操作码 (OP)：1表示ARP请求，2表示ARP应答，3表示RARP请求，4表示RARP应答。

发送方MAC地址：6字节。

发送方IP地址：4字节。

目的方MAC地址：6字节。

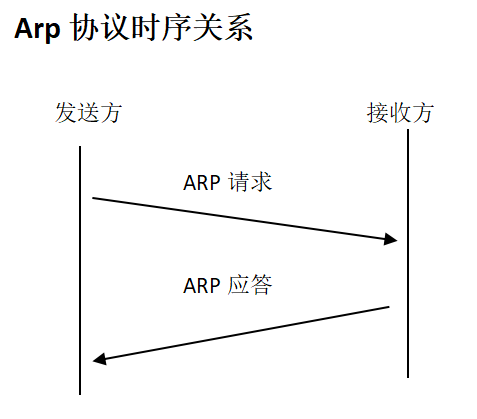
目的方IP地址：4字节。

ARP协议语义：

OP = 1：表示ARP请求，以广播帧形式发送，目的MAC地址字段为48个0，表示请求目的IP对应的MAC地址。

OP = 2：表示ARP应答，以单播形式发送，告诉源发送方自己IP地址对应的MAC地址。

#### 基本流程：



* 发送方实现：

定义网络层常量（本地IP、子网掩码、网关IP、DNS服务器IP、DHCP标志）。

定义ARP协议数据单元数据结构变量。

建立和初始化ARP缓存队列。

检查ARP缓存中是否有目的MAC地址，如果有则直接获取；否则，调用ARP协议发送ARP请求。

等待接收ARP应答，如果超时则重发ARP请求，最多发送三次。

如果成功接收到ARP应答，更新ARP缓存并发送IP分组。

* 接收方实现：

定义网络层常量。

定义ARP协议数据单元数据结构变量。

建立和初始化ARP缓存队列。

调用ARP \_recv函数处理接收到的ARP请求报文。

如果请求中的源IP和源MAC地址映射不在ARP缓存队列中，则增加该映射并发送ARP应答。

## 算法设计

### **1.**ARP\_Cache\_Table.cpp

重要算法：ARP节点管理

* struct arp\_node\* make\_arp\_node（）：创建一个新的ARP节点，分配内存并初始化IP地址、MAC地址和状态。
* void insert\_arp\_node（）：将新的ARP节点插入到ARP表中，如果IP地址不存在，则将节点添加到表尾。
* int delete\_arp\_node（）：从ARP表中删除指定的ARP节点，通过比较IP地址和MAC地址找到匹配的节点并删除。

### 2. Network\_ARP\_send.cpp

重要算法：ARP数据包发送

* void load\_arp\_packet（）：加载ARP数据包，将本地MAC地址和IP地址填入源地址字段，将目标IP地址填入目标地址字段。
* void network\_arp\_send（）：发送ARP数据包，首先加载ARP数据包，然后调用以太网发送函数发送数据包。

### 3. Network\_ARP\_recv.cpp

重要算法：ARP数据包接收和处理

* int is\_accept\_arp\_packet（）：检查是否接受传入的ARP数据包。它通过检查硬件类型、协议类型、目标IP地址和操作码来确定数据包是否有效。如果数据包有效且源IP地址不存在于ARP表中，则创建并插入新的ARP节点。
* u\_int8\_t\* network\_arp\_recv（）：处理接收到的ARP数据包。首先调用 is\_accept\_arp\_packet 函数检查数据包是否有效。如果有效，则输出ARP数据包信息和ARP表。如果是ARP请求，则发送ARP回复；如果是ARP回复，则返回源MAC地址。

### 4. Network\_IPV4\_send.cpp

重要算法：IP数据包发送

* int network\_ipv4\_send（）：发送IP数据包，首先计算文件大小和需要的片段数，然后逐个片段加载IP头部和数据，检查目标MAC地址是否在ARP表中，如果不在则发送ARP请求获取目标MAC地址，最后发送IP数据包。

### 5. Network\_IPV4\_recv.cpp

重要算法：IP数据包接收

* int network\_ipv4\_recv（）：接收并处理IP数据包，首先检查数据包是否有效，然后处理数据包分片，将数据加载到缓冲区或文件中，最后输出IP数据包的详细信息并根据上层协议类型调用相应的处理函数。

### 6. Ethernet\_send.cpp

重要算法：以太网数据包发送

* int ethernet\_send\_packet（）：发送以太网数据包，首先加载以太网头部和数据，然后调用 pcap\_sendpacket 函数发送数据包，如果发送失败则返回错误码，否则返回成功码。

### 7. Ethernet\_recv.cpp

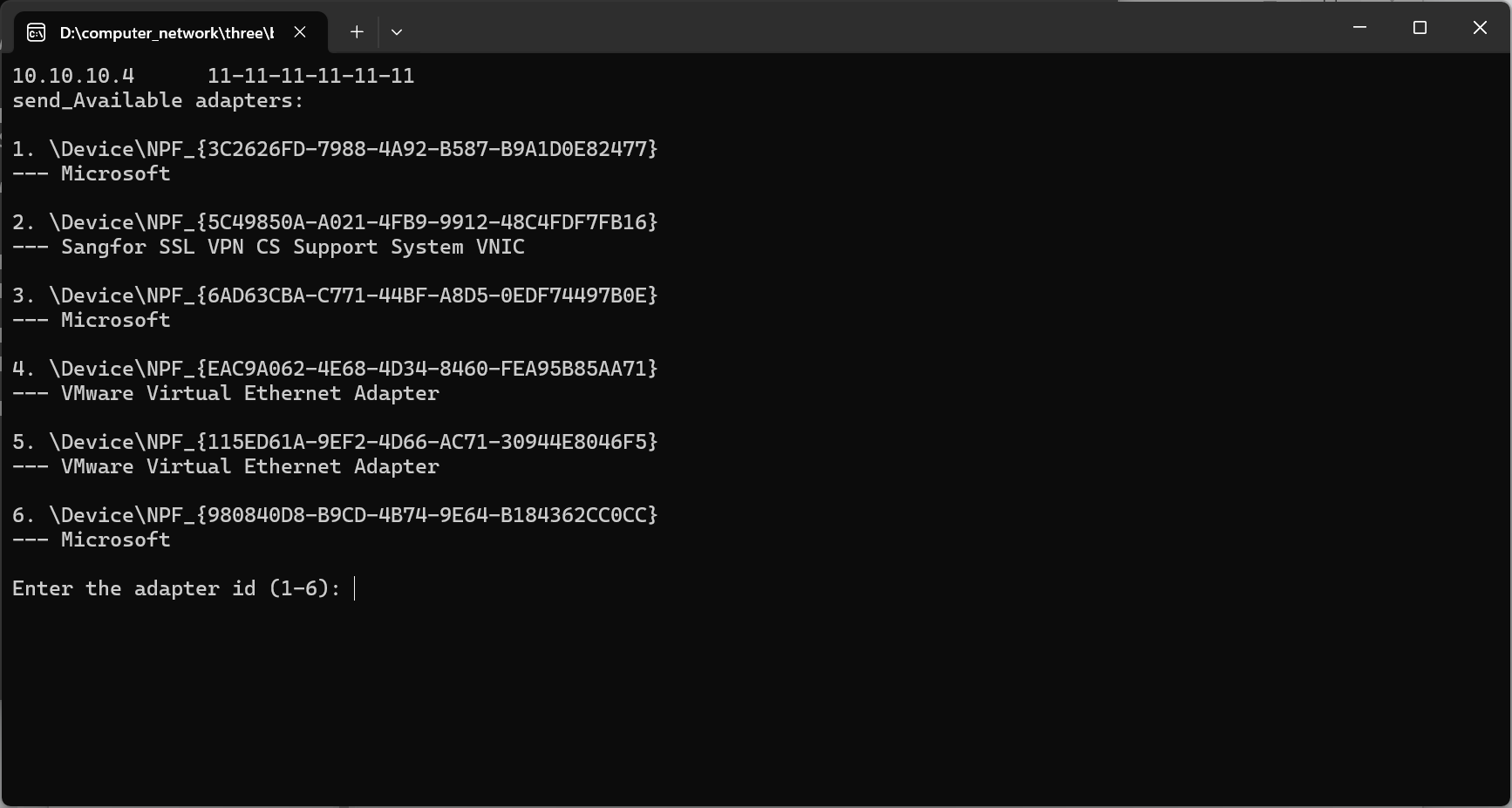
重要算法：以太网数据包接收和处理

* void ethernet\_recv\_packet（）：接收以太网数据包。首先将缓冲区转换为以太网头部结构体指针，然后根据以太网类型调用相应的处理函数。如果是ARP数据包，则调用 network\_arp\_recv 函数处理；如果是IP数据包，则调用 network\_ipv4\_recv 函数处理；否则，打印未知以太网类型的消息。

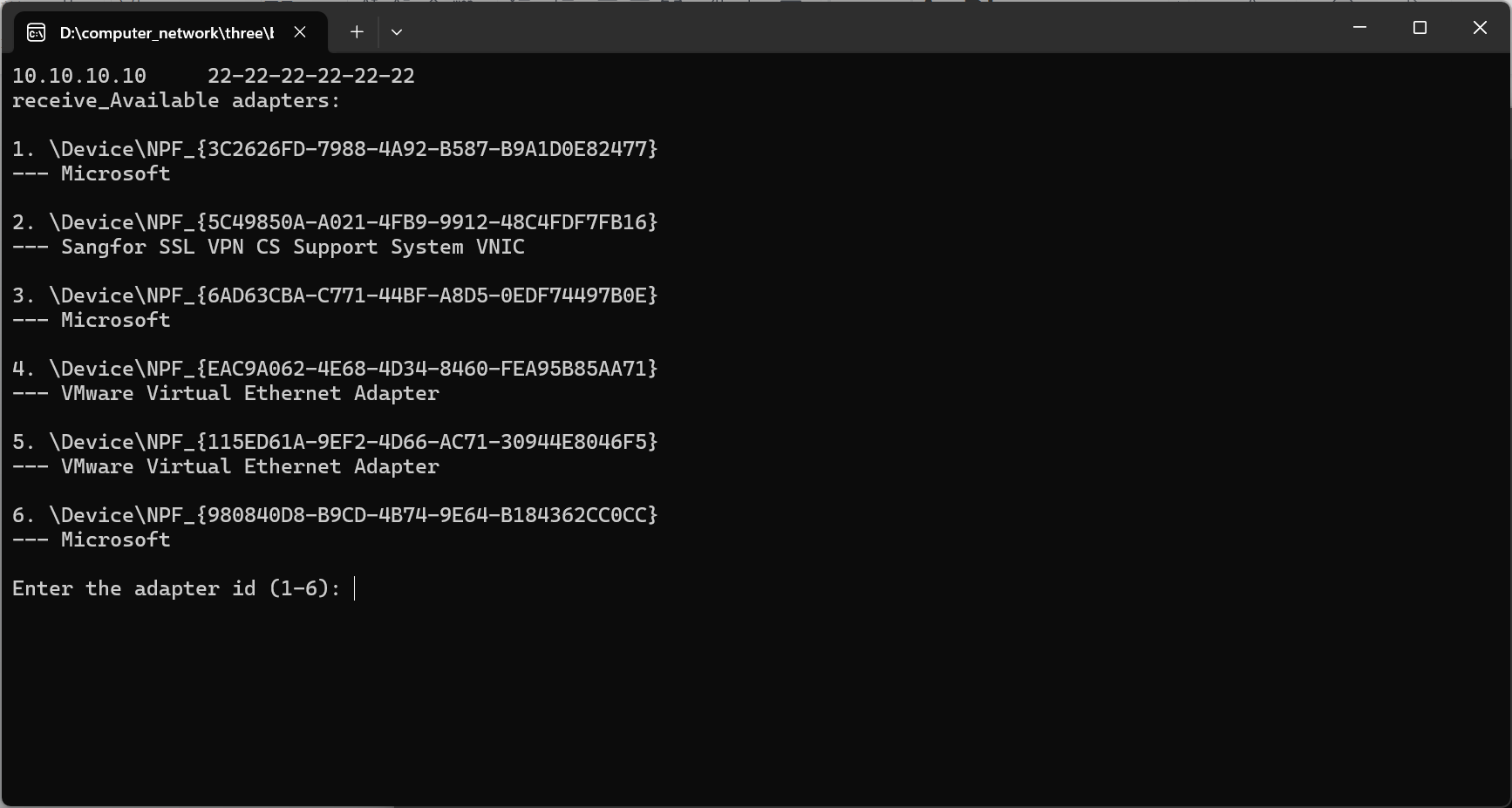
## 实验结果

**发送方**

首先，发送方的缓存表中只有自己的IP与MAC地址，



**接收方**也是：



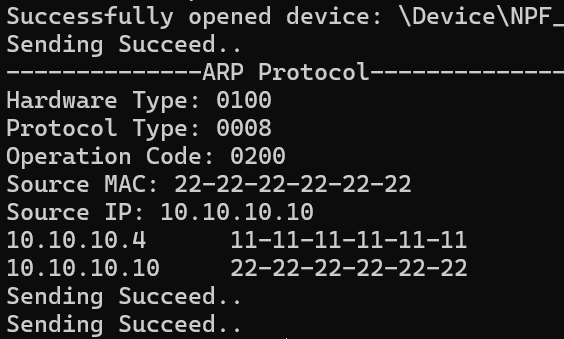
然后，接收方收到ARP请求：



就将收到的ARP报文的发送方的IP与MAC地址加入缓存表中，并发送ARP应答

**发送方**

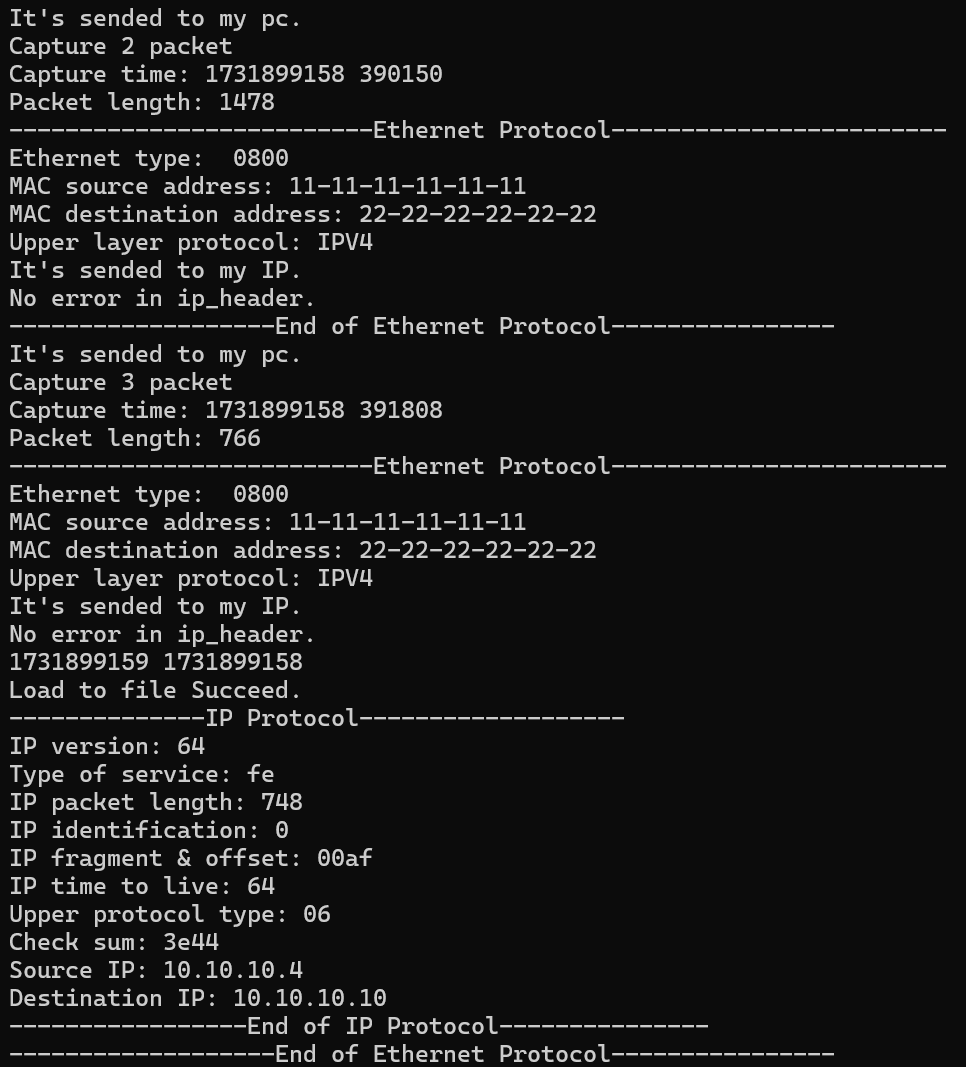
发送方接收到ARP应答，并将发送ARP应答的接收方的IP与MAC加入缓存表：



在得知接收方的MAC后，就能向接收方发送IP分组，即data\_send.txt的内容。

**接收方**

接收方接收到发送方发送的IP分组：



由于我的数据给的大于1500，可以看到有分片。

可以看到接收到了：IP数据包的长度为748，位偏移为00af，检验和为3e44，源IP地址和目的IP地址，均正确。

## 实验过程发现的问题

1. 一定要加上选择网卡的功能，要是直接选择默认网卡，会出现收不到的情况，加的时候要注意handle变量不要重复定义局部变量，因为我把它定义为了全局的外部变量，所以传参的时候handle也不需要额外传，直接共享外部变量的值就行。
2. 发送端发送完成IP数据报之后终端会自动关闭，导致我无法查看目的端传回来的ARP报文，所以要在Main.cpp中加上system("pause");保持终端。
3. 代码逻辑要搞清楚，比如说为什么不管是send端还是receive端都有send和receive文件，这是因为ARP报文的传送使得双方既是发送端也是接收端，搞清楚这个逻辑就可以。

## 参考文献

《第三次课程：ARP协议设计与实现》

《GitHub Copilot》