华东师范大学计算机科学技术系上机实践报告

课程名称：计算机网络	**年级**：2022级	**上机实践成绩**：
指导教师：洪道诚	**姓名**：朱宇笑	**创新实践成绩**：
实验名称：IEEE 802标准和以太网	**学号**：10225001410	**上机实践日期**：2023.11.03
座位编号：F	**组号**：6	**上机实践时间**：2学时

1 实验目的

1. 掌握以太网的报文格式
2. 掌握MAC 地址的作用
3. 掌握MAC广播地址的作用
4. 掌握LLC 帧报文格式
5. 掌握协议编辑器和协议分析器的使用方法
6. 掌握协议栈发送和接收以太网数据帧的过程

2 实验环境

采用网络拓扑结构一

3 实验原理

3.1 OSI 模型和 TCP/IP 协议族

3.1.1 OSI简介

国际标准化组织（ISO）成立于 1947 年，它是个多国团体，专门就一些国际标准达成世界范围的一致。网络方面的 ISO 标准就是 OSI （开放系统互连）模型。它是在 20 世纪 70 年代后期问世的。

在不需要改变底层硬件或软件逻辑的情况下，OSI 模型使两个不同的系统能够较容易地通信。OSI 模型并不是协议，它是个灵活的、稳健的和可互操作的模型，用来设计网络体系结构，它使得所有类型的计算机系统可以通信。OSI 模型包括 7 个层次，每一层都定义了通过网络传送信息的一些过程，如下图所示。掌握了 OSI 模型的基本概念后，就有了学习数据通信较牢固的基础。

图 1: 协议栈

3.1.2 OSI 模型中的层次

1. 物理层

物理层协调在物理媒体中传送比特流所需的各种功能。物理层涉及到接口和传输媒体的机械的和电气的规约。它还定义了这些物理设备和接口在传输过程中所必须完成的任务。

2. 数据链路层

数据链路层把物理层（即原始的传输设施）转换为可靠的链路。

3. 网络层

网络层负责把数据包从源点交付到终点，这可能要跨越多个网络。数据链路层是监督在同一个链路上的两个相邻节点之间数据包的交付，而网络层则确保每一个数据包能够从它的源点到达终点。

如果两个节点连接到同一条链路上，那么通常就不需要网络层。但是，如果两个节点连接在不同的网络上，而这些网络是由一些连接的设备连接起来的，那么通常是需要网络层来完成从源点到终点的交付。

4. 传输层

传输层负责把报文进行端到端的交付。网络层虽然负责单个数据包的端到端交付，但它并不考虑这些数据包之间的关系。传输层要确保整个报文原封不动地按序到达，负责从源点到终点这一级的差错控制和流量控制。

5. 会话层

会话层是网络的对话控制器。它建立并维持通信系统之间的交换，并使这些通信系统同步。会话层完成以下任务:

● 对话控制：会话层允许两个节点进行对话状态控制。它允许两个进程之间的通信按半双工或全双工的方式进行。

● 同步：会话层允许进程将若干个同步点插入到数据流中，以完成传输的同步。

6. 表示层

表示层负责两个系统所交换的信息的语法和语义。表示层完成以下任务:

● 转换：在两个系统中的进程所交换的信息的形式通常都是字符串，数字等等。这些信息在传输之前必须变为比特流。由于不同的计算机使用不同的编码系统，所以表示层负责在这些不同的编码方法之间提供互操作性。

● 加密：为了携带敏感信息，一个系统必须确保能够进行保密。加密就是发送端把原始信息转换为另一种形式，然后再把这种形式的报文发送出去。

●压缩：数据压缩减少了信息中所包含的比特数。在传输多媒体信息时，数据压缩是特别重要的。

7. 应用层

应用层使用户（不管是人还是软件）接入网络，给用户提供了接口，也提供了对多种服务的支持。

3.1.3 TCP/IP 协议族

TCP/IP 协议族是在 OSI 模型出现之前出现的。因此 TCP/IP 协议族的层次无法准确地和OSI 模型对应起来。TCP/IP 协议族由 5 层组成：物理层、数据链路层、网络层、传输层和应用层。前四层与 OSI 的前四层相对应，提供物理标准、网络接口、网际互连、以及传输功能。然而OSI 的高三层在 TCP/IP 中则叫做应用层。如下图所示：

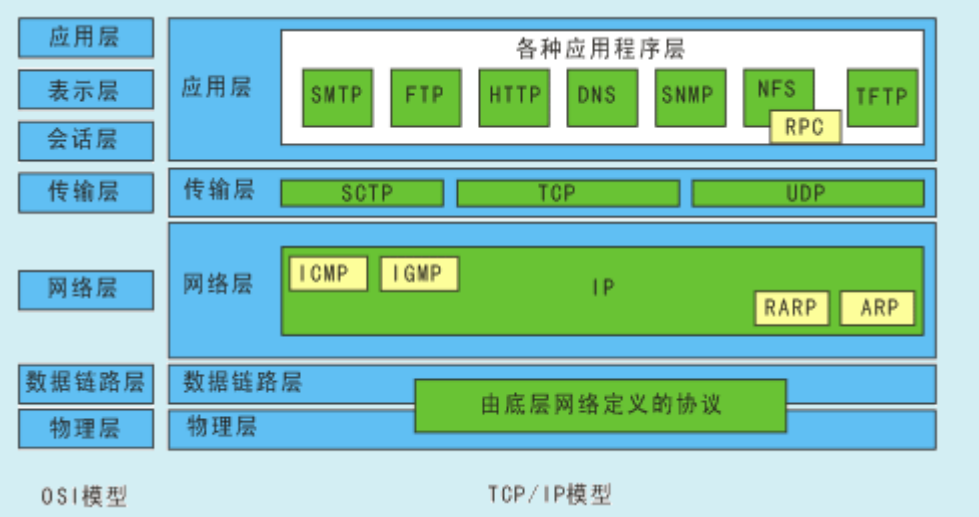

图 2: 协议族

TCP/IP 是由一些交互的模块组成的分层次的协议，每个模块提供特定的功能。TCP/IP 协议族中的各层包含了一些相对独立的协议，可以根据系统的需要把这些协议混合和配套使用。

在传输层中，TCP/IP 定义了 3 个协议：传输控制协议（TCP）、用户数据报协议（UDP）和流控制协议（SCTP）。在网络层中，TCP/IP 定义的主要协议是网际协议（IP）。

3.2 IEEE802 参考模型

1980 年 2 月 IEEE（电气和电子工程师协会）成立了 802 局域网标准委员会，开始了有关局域网标准化的工作。IEEE 802 局域网参考模型中只定义了物理层和数据链路层，在模型中较高层次，IEEE 802 参照 OSI 模型，尽可能与其相符合。IEEE 标准模型与 OSI 模型的比较如下图所示：

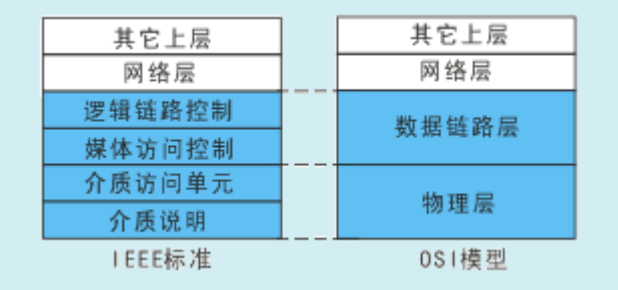

图 3: 模型比较

在 OSI 模型中，物理层的任务是为上一层提供一个物理连接，以透明的方式传送比特流。而在 IEEE802 局域网参考模型中，物理层被分为上下两个子层，分别为：

- 下面的子层是对电缆介质的说明；
- 上面的子层是介质访问单元（MAU）。

电缆可以是各种介质，如双绞线，同轴电缆等。MAU 的主要作用是信息编码、信号发送和介质处理等。在 OSI 模型中，数据链路层的任务是把物理层转化成可靠的链路，使物理层对上层（网络层）看起来好像是不产生差错的。而 IEEE 802 标准的数据链路层被分为两个子层：

- 下面的子层是媒体访问控制子层（MAC）；
- 上面的子层是逻辑链路控制子层（LLC）。

LLC 子层的功能是实现有效的数据传输，负责数据链路层的流量控制和差错控制。MAC 子层的功能是保证物理功能和逻辑功能的连续性，还把从 LLC 子层收到的数据组装成帧，并把帧交给物理层进行编码。

3.3 以太网简介

IEEE 802.3 所支持的局域网标准最早是由 Xerox 开发的，后来通过 Digital 公司、Intel 公司和 Xerox 公司联合扩展为以太网标准，符合以太网标准的局域网络称为以太网。

3.3.1 以太网的分类

数据速率为 10Mbps 的以太网称为标准以太网，数据速率为 100Mbps 的以太网称为快速以太网，数据速率为 1000Mbps 的以太网称为千兆以太网。目前 10G 以太网的标准也已正式制定。

3.3.2 以太网的物理地址

以太网上的每一个主机都有自己的网络接口卡（NIC ）。网络接口卡通常安装在主机内部，并为主机提供一个 6 字节的物理地址，如：44-45-53-54-00-00。在遵循 IEEE802 标准的以太网络中，将这个物理地址称作“MAC 地址”。MAC 地址是惟一的，任意两个不同的网络接口卡都具有不同的 MAC 地址。MAC 地址中的某些位具有特殊的意义，如下图所示：

图 4: MAC 地址位

I/G 比特位表示 Individual/Group，当 I/G 位为 0 时，地址字段表示单个站地址（单播），为 1 时表示组地址，用来进行多播。

G/L 比特位表示 Global/Local，当 G/L 位为 1 时是全球管理，保证在全球没有相同的地址，当为 0 时是本地管理，这时用户可任意分配网络上的地址。

以太网 MAC 地址可分为三类：单播地址、广播地址和多播地址。单播地址（unicast）是一对一的，该地址是特定主机的 MAC 地址；广播地址（broadcast）：广播地址是一对全体的，该地址为全 1，指明数据帧是发送给所有主机的。多播地址（multicast）：多播地址是一对多的，指明数据帧是发送给一部分主机的。

3.4 以太网访问模式

当多个节点被连接到一条链路上时，叫做多点链路或广播链路。这时就需要一个协议来协调链路的访问，使得同一时刻只有一个节点访问链路。如果发生同一时刻多个节点使用链路的情况，则称为链路发生了冲突。带有冲突检测的载波侦听多路访问（CSMA/CD）是这样一种方案。发送主机在传输过程中仍继续监听信道，以检测是否存在冲突。如果发生冲突，信道上可以检测到超过发送主机本身发送的载波信号的幅度，由此判断出冲突的存在。一旦检测到冲突，就立即停止发送，并向总线上发一串阻塞信号，用以通知总线上其它各有关主机。这样，通道容量就不致因白白传送已受损的帧而浪费，可以提高总线的利用率。以太网使用 CSMA/CD 作为其访问模式。

3.5 以太网的帧格式

3.5.1 以太网的 MAC 帧格式

以太网的 MAC 帧格式有两种标准，一种是 DIX Ethernet V2 标准，另一种是 IEEE 的 802.3 标准。但两种帧格式可以在同一以太网络共存。两种帧格式都具有 7 个域：前导码、帧首定界符、目的 MAC 地址、源 MAC 地址、协议类型或数据长度、数据、帧校验序列。如图片所示：

图 5: MAC 帧格式

两种格式的帧可以依据协议类型或数据长度字段的值进行区分。如果此帧是 DIX Ethernet V2 标准格式帧，则协议类型或数据长度字段的值大于 1536；如果此帧是 IEEE 820.3 标准格式的帧，则协议类型或数据长度字段的值小于 1518。对 DIX Ethernet V2 帧来说，此字段的值代表了高层协议的类型；对 IEEE 802.3 帧来说，它的高层协议一定是 LLC，此字段的值代表了数据的长度。

在以太网的 MAC 帧格式中，各个字段的含义如下：

- 前导码：这是以太网 MAC 帧的第一个域，包含了 7 个字节的二进制“1”和“0”间隔的代码，即“10101010.....10”共 56 位，提示接收方一个数据帧即将到来，同时使接收系统建立起同步时钟。
- 帧首定界符：帧首定界符标记了帧的开始。它是一个字节的“10101011”二进制序列，帧首定界符通知接收方后面所有的内容都是数据，以便接收方对数据帧进行定位。
- 目的 MAC 地址：目的 MAC 地址为 6 个字节，标记了数据帧下一个主机的物理地址。如果数据包的目的地址必须从一个网络穿越到另一个网络，那么目的 MAC 地址所包含的是连接当前网络和下一个网络的路由器地址。当数据包到达目标网络后，目的 MAC 地址域换成目的主机的地址。
- 源 MAC 地址：源 MAC 地址也是 6 个字节。它包含了最后一个转发此帧的设备的物理地址。该设备可以是发送此数据帧的主机，也可以是最近接收和转发此数据帧的路由器。
- 协议类型或数据长度：如果该字段的值小于 1518，它用于定义后面数据字段的长度；如果字段的值大于 1536，它定义一个封装在帧中的数据包的类型。
- 数据：它的长度范围是从 46 到 1500 字节之间。46 是以太网 MAC 帧所封装的高层协议数据的最小长度。如果高层协议的数据包小于 46 字节，则填充到 46 字节。
- 帧校验序列：最后一个域是帧校验序列，以太网采用 32 位冗余校验（CRC）。校

验范围是除了前导码、帧首定界符和帧校验序列外的所有内容。

3.5.2 LLC 帧格式

LLC 的帧格式如下图所示：

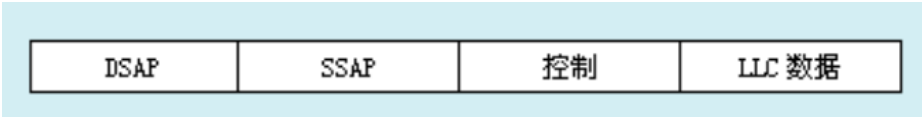

图 6: LLC 帧格式

其中，DSAP（目的服务访问点）和 SSAP（源服务访问点）是 LLC 所使用的地址，用来标识接收和发送数据的计算机上的用户实体。DSAP 的第一个比特是用来指明帧是为单地址还是组地址，0 表示单地址，1 表示组地址。SSAP 的第一个比特用来指明帧是命令帧还是响应帧。0 表示命令帧，1 表示响应帧。

LLC 定义了三种帧：信息帧（I-帧）、监控帧（S-帧）和无编号帧（U-帧）。帧的类型可从控制字段识别。对于信息帧和监控帧，控制字段为 2 字节长，而对于无编号帧，控制字段为 1 字节长。

下图表示了 LLC 三类帧的控制字段的比较。

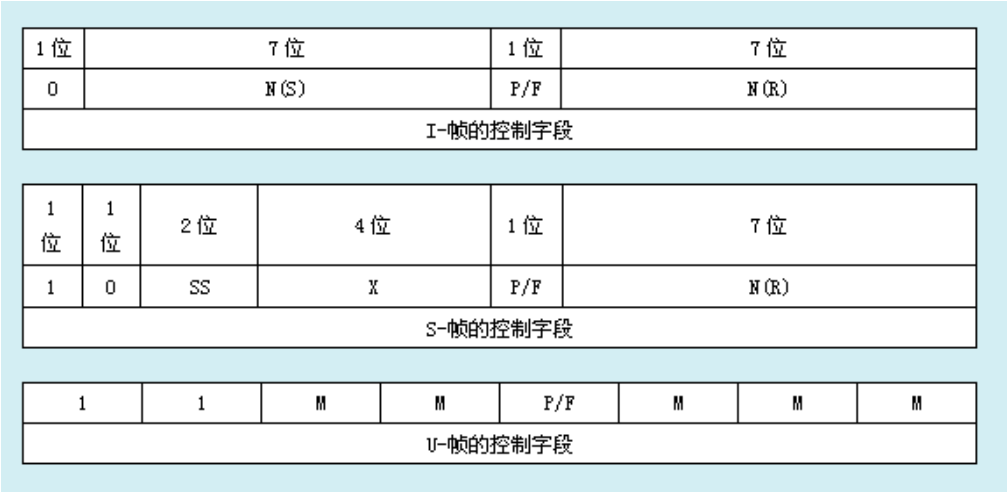

图 7: LLC 控制字段比较

- N(S)：发送序号。
- N(R)：接收序号。
- SS：监控功能位，00 表示准备接收（RR）；10 表示未准备接收（RNR）；01 表示拒绝（REJ）。
- M：修正功能位。
- X：保留，设置为 0。

- P/F：Poll/final 位。命令 LLC PDU 传输/响应 LLC PDU 传输。

3.5.3 LLC-PDU 与相邻层的 PDU 之间的关系

IEEE 802 标准为 LLC 和 MAC 子层的帧格式作了详细规定。下图描述了网络层 PDU、LLC子层 PDU 和 MAC 子层 PDU 的关系。

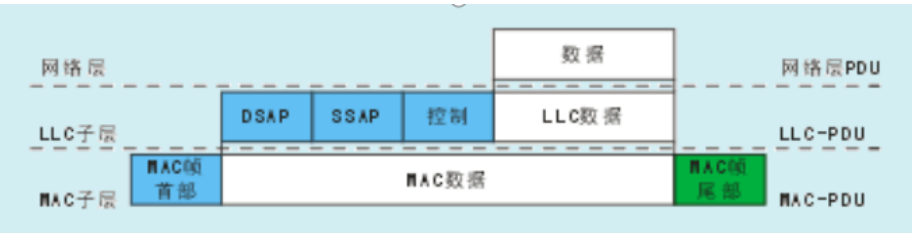

图 8: LLC-PDU 与相邻层 PDU 关系

LLC 帧（即 LLC-PDU）与媒体无关，而 MAC（即 MAC-PDU）则与局域网的媒体访问方式有很大关系，不同的局域网有不同的 MAC 帧格式。

3.5.4 LLC 地址与 MAC 地址

在 MAC 帧的帧首中，有目的 MAC 地址和源 MAC 地址，它们都是 6 字节长。在 LLC 帧的帧首中，则设有 DSAP 和 SSAP，该地址是逻辑地址，表示的是数据链路层的不同访问服务点。

LLC 地址与 MAC 地址是两个不同的概念，在局域网中，一个主机上的多个服务访问点可以利用同一条数据链路。从这一点可以看出，LLC 子层带有 OSI 网络层的某些功能。

3.6 协议栈实现代码解析

本实验将通过对 netproto_eth_student.h 和 netproto_eth_student.c 两个文件进行编码，完成协议栈中以太网数据帧接收和发送的实现。

netproto_eth_student.h 文件中定义了以太网数据帧中“协议类型与数据长度”字段值以及以太网数据帧的负载内容、负载长度，关键代码如下所示：

```
#define PAYLOAD_DATA "Hello,World!"
#define PAYLOAD_LEN sizeof(PAYLOAD_DATA)
#define MAX_PAYLOAD_LEN 1024
#define TYPE_LENGTH 5893
```

这段代码定义了 3 个宏，他们代表的含义如下所示：

宏	值	描述
PAYLOAD_DATA	”Hello, World!”	定义以太网负载数据
PAYLOAD_LEN	sizeof(PAYLOAD_DATA)	定义以太网数据长度
TYPE_LENGTH	5893	定义以太网帧中的“协议类型或数据长度”字段的值

在实验的编码过程中，应该使用这些宏对相应的变量进行赋值。学生也可以根据自己的需求修改这些宏定义的值。

netproto_eth_student.c 文件是协议栈中以太网数据帧的实现部分，其中定义了 2 个函数。下面分别介绍这些协议栈的实现部分。

函数 netp_eth_output_student 的功能是编辑并发送一个 Ethernet V2 数据帧。这个函数的编码工作需要由学生完成。

当有数据到达本机网络接口时，函数 netp_eth_input_student 将被调用，并传递给这个函数原始数据。该函数的返回值为 push_to_lwip 的枚举类型值，push_to_lwip 的定义如下：

```
enum push_to_lwip{
 NETP_PUSH_TO_LWIP,//数据处理完成后，交给 lwIP 继续处理
 NETP_NO_PUSH_LIWP//数据处理完成后，不交给 lwIP 继续处理，本层处理完毕以后数据包被丢弃
};
```

返回 NETP_PUSH_TO_LWIP 表示这个数据帧应该提交给协议栈上层继续处理，而返回NETP_NO_PUSH_LIWP 则表示不需要提交给协议栈上层处理，本层处理完毕后，这个数据帧将被丢弃。需要根据正确的逻辑关系返回适当的值，使协议栈正常工作。在编码过程中可能会遇到一些结构体、宏和函数，下表是对他们进行介绍：

结构体/函数	声明或定义	描述

struct netp_eth_header	struct netp_eth_header {
	  struct netp_eth_addr dest_address;
	  struct netp_eth_addr sour_address;
	  u16_t type;
	};
		以太网数据帧头结构

struct netp_eth_addr	struct netp_eth_addr {
	  u8_t addr[ETH_ADDRESS_LEN];
	};
		以太网地址

ETH_HEADER_LEN	#define ETH_HEADER_LEN 14	以太网帧头长度
PAYLOAD_DATA	#define PAYLOAD_DATA "Hello, EXPcns!"	以太网负载内容
PAYLOAD_LEN	#define PAYLOAD_LEN sizeof(PAYLOAD_DATA)	以太网负载长度
MAX_PAYLOAD_LEN	#define MAX_PAYLOAD_LEN 1024	最大负载长度
TYPE_LENGTH	#define TYPE_LENGTH 5893	协议类型与数据长度字段值

netp_set_eth_addr	void netp_set_eth_addr(
	  struct netp_eth_addr *p_addr,
	  u8_t v1, u8_t v2, u8_t v3,
	  u8_t v4, u8_t v5, u8_t v6
	);
		设置以太网数据帧 MAC 地址

netp_current_hw_addr	int netp_current_hw_addr(u8_t* hardware_address);	获取正在使用的网络适配器的物理地址
netp_packet_send	Int netp_packet_send(void* buffer, int len);	发送一个数据帧
htons	u16_t htons(u16_t n);	将 16 位数值由主机字节序转换为网络字节序
netp_is_eth_addr_broadcast	int netp_is_eth_addr_broadcast(struct netp_eth_addr *mac_address);	判断一个 MAC 地址是否为广播地址

netp_eth_addr_cmp	int netp_eth_addr_cmp(
	  struct netp_eth_addr *mac_address1,
	  struct netp_eth_addr *mac_address2
	);
		判断两个 MAC 地址是否相同

图 9: 结构体函数描述

3.7 各模块推荐流程

3.7.1 Ethernet V2 数据帧发送流程

编码实现 Ethernet V2 数据帧发送推荐使用如下流程：

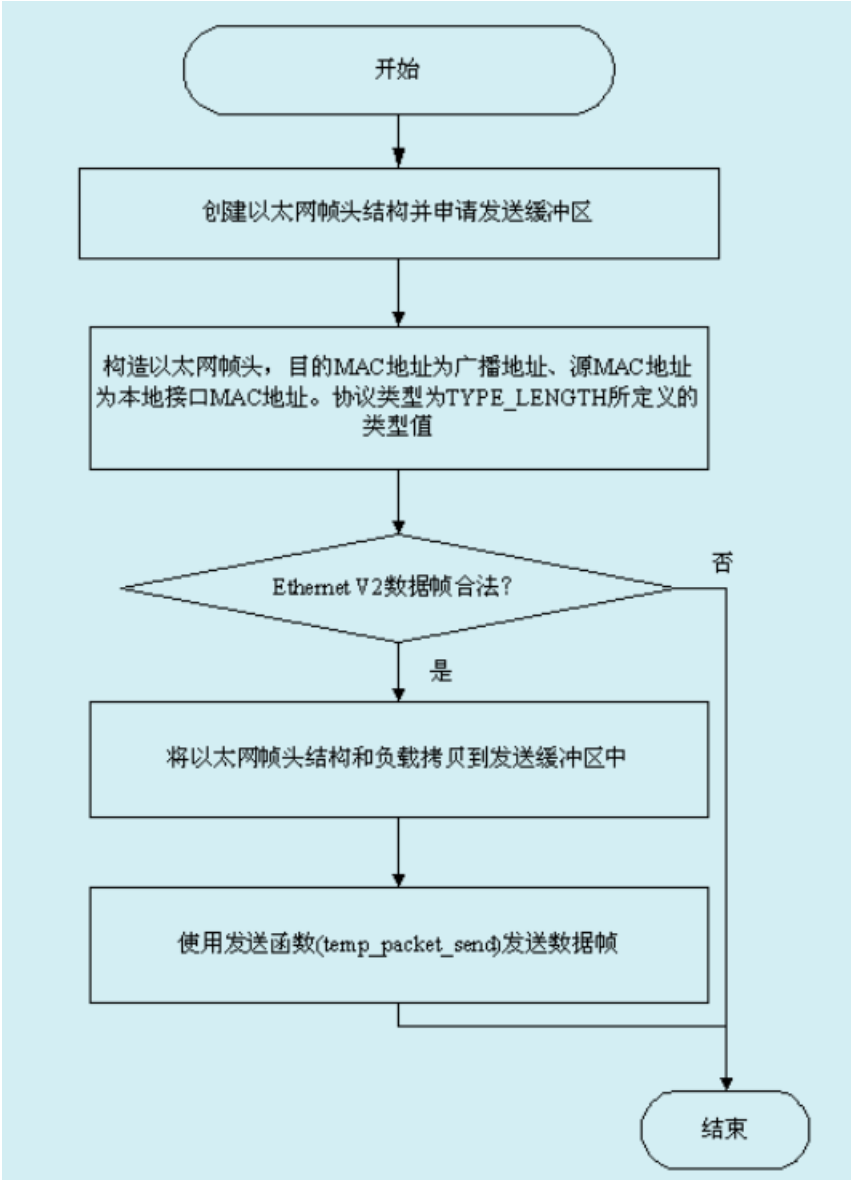

图 10: 流程图

3.7.2 Ethernet V2 数据帧处理流程

编码实现处理 Ethernet V2 输入数据帧推荐使用如下流程：

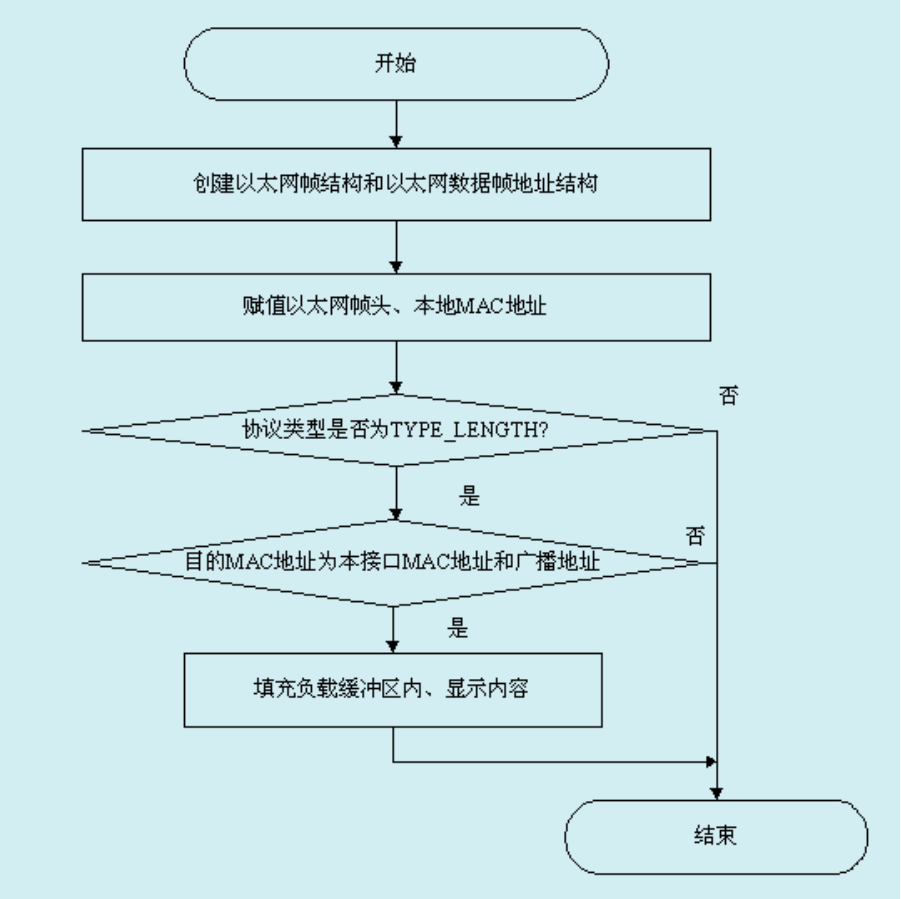

图 11: 流程图

4 实验内容

4.1 领略真实的 MAC 帧

1. 主机B 启动协议分析器，新建捕获窗口进行数据捕获并设置过滤条件（提取ICMP 协议）。
2. 主机A ping 主机B，察看主机B 协议分析器捕获的数据包，分析MAC 帧格式。
3. 将主机B 的过滤器恢复为默认状态。

4.2 理解 MAC 地址的作用

本练习将主机A 和B 作为一组，主机C 和D 作为一组，主机E 和F 作为一组。现仅以主机A、B 为例，其它组的操作参考主机A、B 的操作。

1. 主机B 启动协议分析器，打开捕获窗口进行数据捕获并设置过滤条件（源MAC 地址为主机A 的MAC 地址）。

2. 主机A ping 主机B。

3. 主机B 停止捕获数据，在捕获的数据中查找主机A 所发送的ICMP 数据帧，并分析该帧内容,记录实验结果

4.3 编辑并发送 MAC 广播帧

本练习将主机A、B、C、D、E、F作为一组进行实验。

1. 主机E启动协议编辑器。
2. 主机E编辑一个MAC帧：
   目的MAC地址：FFFFFF-FFFFFF
   源MAC地址：主机E的MAC地址
   协议类型或数据长度：大于0x0600
   数据字段：编辑长度在46—1500字节之间的数据
3. 主机A、B、C、D、F启动协议分析器，打开捕获窗口进行数据捕获并设置过滤条件（源MAC地址为主机E的MAC地址）。
4. 主机E发送已编辑好的数据帧。
5. 主机A、B、C、D、F停止捕获数据，察看捕获到的数据中是否含有主机E所发送的数据帧。

- 结合练习三的实验结果，简述FFFFFF-FFFFFF作为目的MAC地址的作用。

4.4 编辑并发送LLC帧

本练习将主机A和B作为一组，主机C和D作为一组，主机E和F作为一组。现仅以主机A、B所在组为例，其它组的操作参考主机A、B所在组的操作。

1. 主机A启动协议编辑器，并编写一个LLC帧。目的MAC地址：主机B的MAC地址源MAC地址：主机A的MAC地址协议类型和数据长度：001F控制字段：填写02（注：回车后变成0200，该帧变为信息帧，控制字段的长度变为2字节）用户定义数据/数据字段：AAAAAAABBBBBB-BCCCCCCCDDDDDD（注：长度为27个字节）

2. 主机B启动协议分析器并开始捕获数据。

3. 主机A发送编辑好的LLC帧。

4. 主机B停止捕获数据，在捕获到的数据中查找主机A所发送的LLC帧，分析该帧内容。

- 记录实验结果
- 简述“协议类型和数据长度”字段的两种含义。

5. 将第1步中主机A已编辑好的数据帧修改为“无编号帧”(前两个比特位为1)，用户定义数据/数据字段修改为AAAAAAABBBBBBBCCCCCCCDDDDDDD（注：长度为28个字节），重做第2、3、4步。

4.5 发送Ethernet V2数据帧功能的实现

本练习将主机A、B、C、D、E、F作为一组进行实验。实验开始前，先单击“初始环境”。在实验中，主机A将新接口的IP地址设置为172.16.0.11、主机B使用处于连接状态的物理接口，将新接口的IP设置为172.16.0.12、主机C将新接口的IP地址设置为172.16.0.13、主机D使用处于连接状态的物理接口，将新接口的IP地址设置为172.16.0.14、主机E使用处于连接状态的物理接口，将新接口的IP地址设置为172.16.0.15、主机F将新接口的IP地址设置为172.16.0.16。所有主机使用子网掩码

255.255.255.0，默认网关设置为 0.0.0.0。

1. 所有主机编码实现发送 Ehternet V2 数据帧

● 各主机打开安装目录 ExpCNS/work/EXPcns_student/netproto_eth_student /netpro- toþ_student 下的 netproto_eth_student.c 文件，在函数 netp_eth_output_student 内编写实现代码。

● 参考实验原理 Ethernet V2 数据帧发送推荐流程图给出的流程，分析已经存在的代码。已经存在的代码定义了一个以太网数据帧头部结构和一个能容纳以太网帧头和负载的发送缓冲区send_buff，另外还实现了将以太网帧头结构和负载拷贝到发送缓冲区的过程。

● 构造、填充以太网数据帧头构造并填充一个以太网数据帧头。目的 MAC 地址设置为广播地址即FF-FF-FF- FF-FF-FF，可以使用 netp_set_eth_addr 函数设置 MAC 地址。源 MAC 地址设 置为本接口的 MAC 地址，可以使用 netp_current_hw_addr 函数获取本接口的 MAC 地址。协议类型或数据长度字段值应设置为 0x0806，表示上层协议为 arp 协议，可以使用 MAC_PROTO_ARP宏。

● 判断是否为合法的 Ethernet V2 数据帧根据实验原理关于 MAC 帧格式的介绍，判断要发送的数据帧是不是合法的 Ethernet V2 数据帧，即“协议类型或数据长度”字段值是否大于1536。

● 使用发送函数 netp_packet_send 发送数据帧。

2. 当完成代码编写后，所有主机打开协议分析器，开始捕获数据。
3. 所有主机调试并运行程序。
4. 各主机停止数据捕获，观察实验现象。

● 捕获到的数据帧，“协议类型或数据长度”字段值是什么?

4.6 处理 Ethernet V2 输入数据帧功能的实现

本练习将主机 A、B、C、D、E、F 作为一组进行实验。实验开始前，先单击“初始环境”。该练习需要在前一个练习的基础上进行。在实验中，主机 A 将新接口的 IP 地址设置为 172.16.0.11、主机 B 使用处于连接状态的物理接口，将新接口的 IP 设置为 172.16.0.12、主机 C 将新接口的IP 地址设置为 172.16.0.13、主机 D 使用处于连接状态的物理接口，将新接口的 IP 地址设置为172.16.0.14、主机 E 使用处于连接状态的物理接口，将新接口的 IP 地址设置为 172.16.0.15、主机 F 将新接口的 IP 地址设置为 172.16.0.16。所有主机使用子网掩码 255.255.255.0，默认网关设置为 0.0.0.0。

1. 所有主机编码实现处理 Ethernet V2 输入数据帧功能

● 各主机打开安装目录 ExpCNS/work/EXPcns_student/netproto_eth_student/ netproto_eth_student 下的 netproto_eth_student.c 文件，在函数

netp_eth_input_student 内编写实现代码，参考实验原理处理 Ethernet V2 输入数据帧推荐流程图给出的流程，思考代码编写方案。首先分析 net-proto_eth_student.c 文件中已经给出的代码。已经存在的代码定义了一个以太网数据帧头部结构和以太网数据帧地址结构，实现了从输入缓冲区中拷贝以太网帧头结构，获取本接口 MAC 地址的过程。最后的几行代码填充了负载缓冲区，显示其内容。

- 提取“协议类型或数据长度”字段值为 TYPE_LENGTH 的数据帧通过判断以太网帧中的“协议类型或数据长度”字段值是否为 TYPE_LENGTH （自定义上层协议）来过滤以太网数据帧。如果接收到的数据包不满足条件，则应该返回 NETP_PUSH_TO_LWIP 交给协议栈处理。

- 提取“目的 MAC 地址”字段值为本接口地址或广播地址的数据帧.

通过判断以太网帧中的“目的 MAC 地址”字段值是否为本接口 MAC 地址或广播地址（目的 MAC 地址为全 1）来过滤不是发送给本接口的数据帧。如果接收到的数据包不满足条件，则应该返回 NETP_PUSH_TO_LWIP 交给协议栈处理。

2. 所有主机调试并运行程序将在安装目录 ExpCNS/work/EXPcns_student/netproto_eth_student/ netproto_eth_student 下的 netproto_eth_student.h 文件中 PAYLOAD_DATA 定义的内容修改为自己要发送的负载信息，例如“I am host A”。

- 你收到的负载内容是什么?

5 实验步骤与结果

5.1 领略真实的 MAC 帧（练习一）

启动协议分析器，新建捕获窗口进行数据捕获并设置过滤条件（提取 ICMP 协议）。

首先在命令行输入 ipconfig /all，查看本机的 MAC 地址和 IP 地址。

A 的 MAC 地址为F4-8E-38-AD-0D-51，IP 地址为 172.16.0.61。

图 12: B 的 MAC 地址和 IP 地址

B 的MAC 地址为F4-8E-38-AD-0D-68，IP 地址为 172.16.0.62。子网掩码自动生成，为 255.255.0.0。首先主机 B 先提取协议分析器，设置其地址，协议如下：

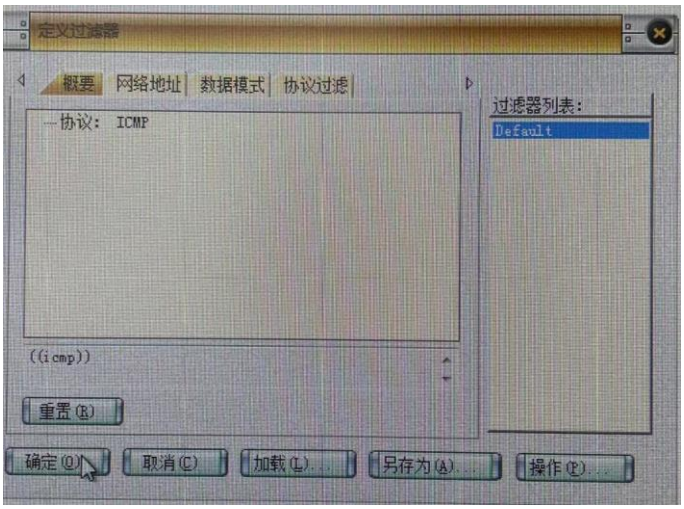

图 13: B 提取协议分析器

然后主机 A ping B

B 的会话分析如下：

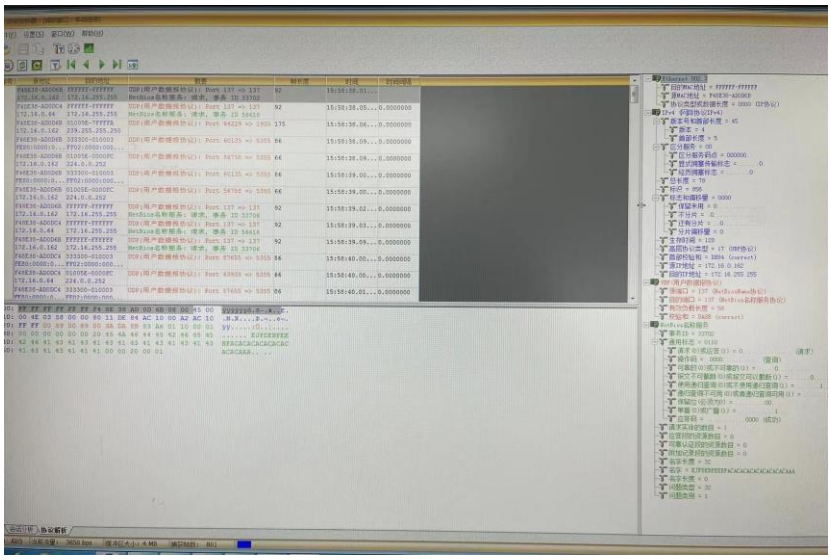

图 14: B 的协议解析

这整个过程反应了完整的请求回复。

5.2 理解 MAC 地址的作用

在前面的实验中已经知道了主机 A 地址为F4-8E-38-AD-0D-51，主机 B 的地址为 F4-8E-38-AD-0D-68。

首先主机 B 先提取协议分析器，设置其地址、协议如下，然后新建捕获窗口开始捕获、

主机 A ping 主机 B

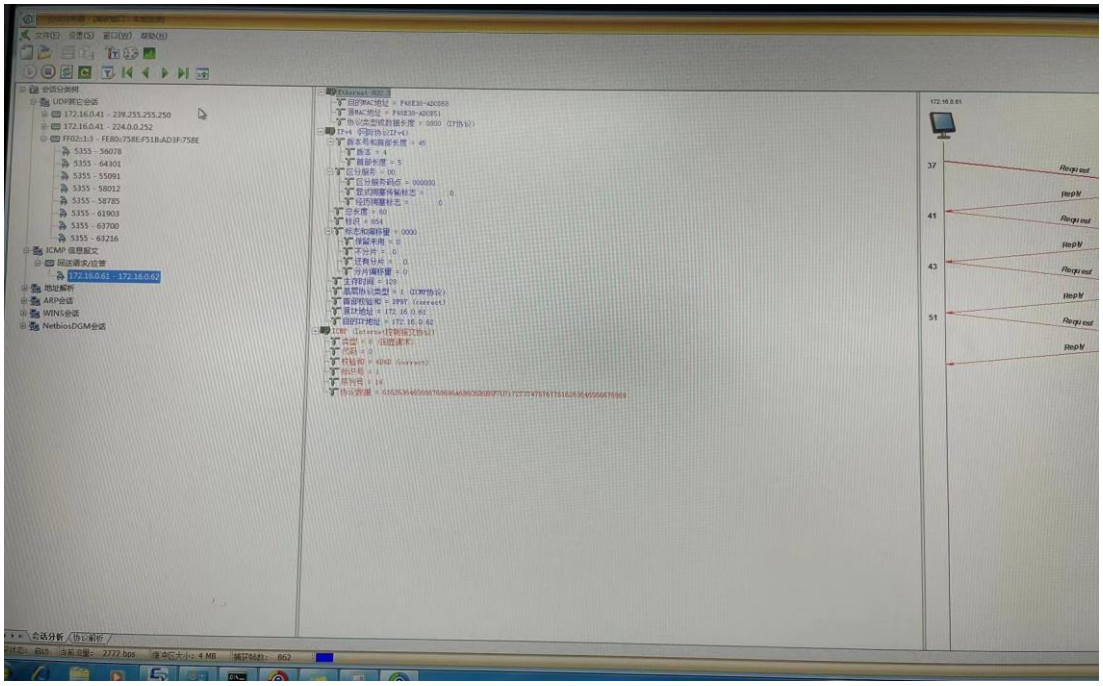

图 15: B 的会话分析

	本机MAC地址	源MAC地址	目的MAC地址
主机B	F48E38-AD0D68	F48E38-AD0B51	F48E38-AD0D68

思考问题：

1.MAC协议应用于TCP/IP协议模型的哪一层?

数据链路层

2.如何区分以太网的两种标准格式?

查看以太网帧的帧类型字段或长度字段。如果字段值在0x0600（十进制为1536）以上，表示使用Ethernet II标准，而如果字段值小于等于1500，则表示使用802.3 Ethernet标准。

5.3 编辑并发送 MAC 广播帧

首先主机 E 启动仿真编辑器，编辑一个 MAC 帧，源地址填写主机 E 的地址，目的地址填写 FFFF-FFFF。

主机 B设置过滤器，其中 MAC 地址填写主机 E 的MAC 地址，另一端则填写 ANY 方便接收数据。可以看到发送后捕捉到的结果:

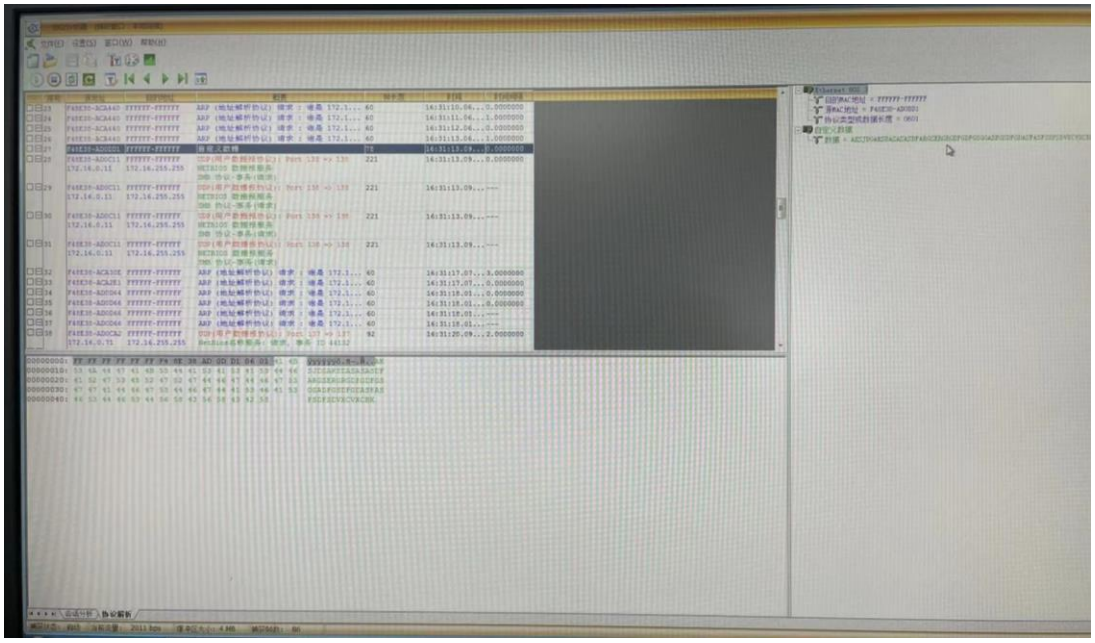

图 16: B 的抓取结果

主机 E 发出的数据在主机 A、C、D、E、F 上已经接收到。

经过思考，FFFFFF-FFFFFF 作为目的 MAC 地址，，就是意味着它是一个广播帧，使得它可以向 6 台主机发送该报文。同时为了保证 6 台主机都可以接收到该报文，需要修改其他几个接受端 A、B、C、D、F主机过滤器的另一端为 Any。

思考问题

1. 主机 A、B、C、D、F 是否可以收到主机 E 的广播帧?

可以收到。

2. 说明 MAC 广播帧的范围?

同一网段内所有主机都可以收到MAC广播帧。

5.4 编辑并发送 LLC 帧

对于主机 A，编写一个 LLC 帧，数据字段:

AAAAAAABBBBBBBCCCCCCCDDDDDD，并发送。目的 MAC 地址填写主机 B 的 MAC 地址，源 MAC 地址填写主机 A 的 MAC 地址，这里填写的是协议类型和数据长度填写 001F。控制字段填写 02。回车后变成 0200，该帧变为信息帧，控制字段的长度变为 2 字节。

B 启动相应的协议分析器，进行接收:

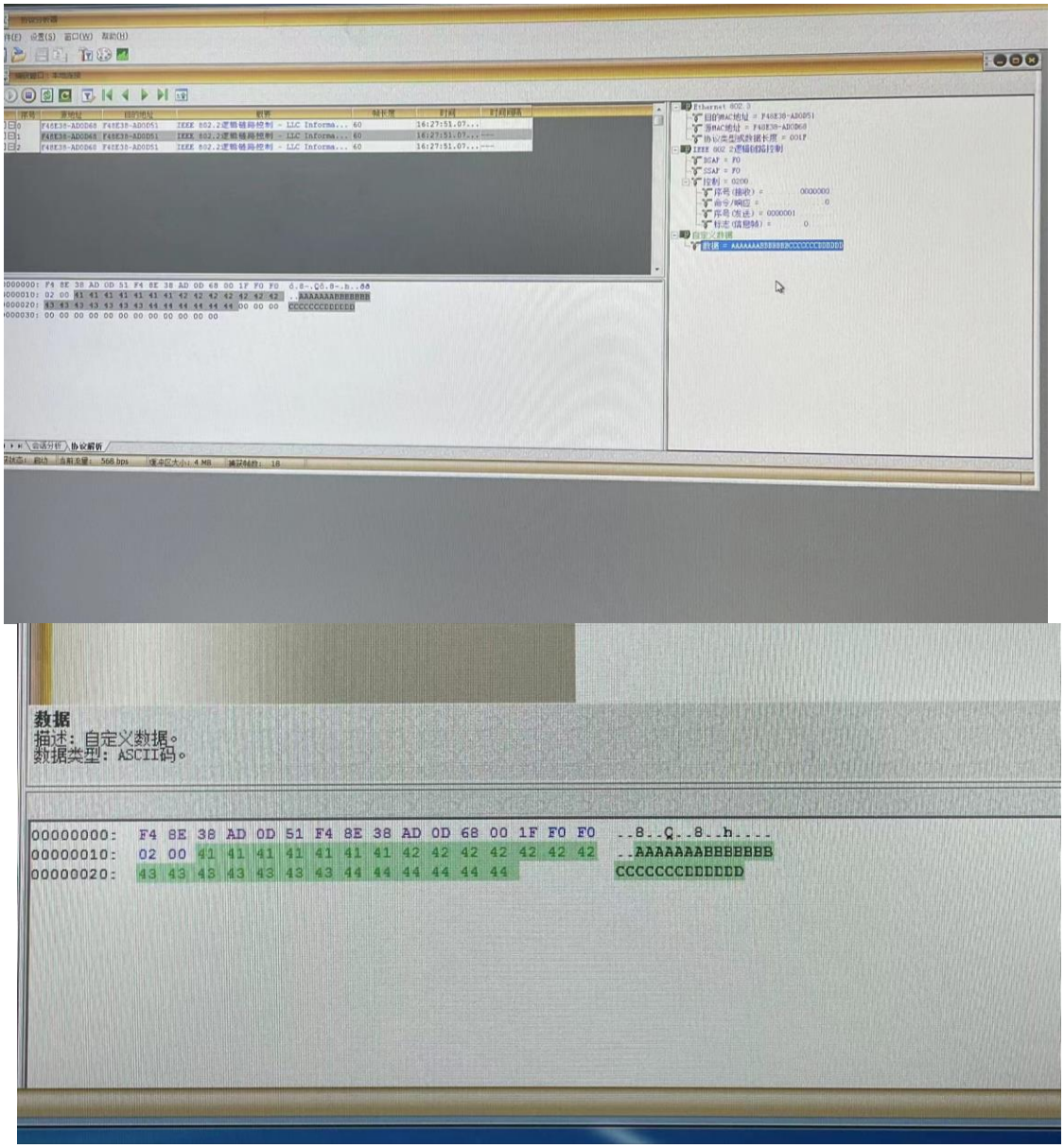

图 17 ：B 的协议分析

可以发现 B 收到了相应的数据，接收成功。

帧类型	发送序号 N(S)	接受序号 N(R)
信息帧	0000001	0000000

简述“协议类型和数据长度”字段的两种含义。

协议类型是网络适配器向网络中发送数据时所使用格式，数据长度是 CPU 一次可以处理的数据长度。

思考问题

1. 如何编辑 LLC 无编号帧和 LLC 数据帧。

当 LLC 帧控制字段长度为 1 字节时表示无编号帧，为 2 字节时表示数据帧。

2. 在协议分析端捕获到该帧，帧的长度是多少？由此理解以太网的最短帧长度。

64 字节，因为如果数据长度未到 46，则会填充到 46 字节。实验中抓到的包是 60 可能是因为把最后 4 个字节的 FCS 丢掉的结果。

3. 为什么 IEEE 802 标准将数据链路层分割为 MAC 子层和 LLC 子层?

为了实现链路层和物理层的完全独立。MAC 子层与物理层相关联，而 LLC 子层则完全独立出来，为高层提供服务，这样就实现了物理层和数据链路层的完全独立，解决了 OSI 模型中局域网物理层和数据链路层不能完全独立的问题。

4. **为什么以太网有最短帧长度的要求**?

为了检测以太网中帧碰撞而设置的，最短帧长度为 64 字节，就可能出现网络上有两个帧在传播，当发生碰撞时，会产生碰撞而造成网络无法发送数据。由此可以判断，当帧长度大于等于64 字节时才是有效的帧，最小帧长必须大于整个网络的最大时延位，即最大时延时间内可以传输的数据位。如果数据帧小于 64 字节，则说明它不符合规格，可能是碰撞后的数据碎片。