



哈爾濱工業大學(深圳)
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

计算机网络

第二次实验

CONTENTS

目录

「01」

实验目的

「02」

实验原理

「03」

实验步骤

「04」

作业提交



实验目的



➤ Lab2 协议栈之Eth协议实现

- ① 熟悉网络协议栈的分层结构，掌握协议栈的数据结构定义；
- ② 了解以太网数据包捕获机制，熟悉数据包的收发过程；
- ③ 熟悉数据链路层数据帧格式，掌握以太网数据帧的发送和接收处理过程。

➤ Lab3 协议栈之ARP协议实现

- ① 熟悉ARP数据包格式，掌握ARP映射表的建立与更新过程；
- ② 掌握ARP数据包的发送和接收处理过程。



只有敲代码才能
感受到温暖



实验任务



➤ Lab2 协议栈之Eth协议实现

在给定的**协议栈代码框架**上，编写**Eth数据帧的发送和接收函数**，使其能够发送和接收数据帧，并且能通过实验评测系统的测试。

➤ Lab3 协议栈之ARP协议实现

- 在完成Eth的基础上，编写**ARP报文的发送、接收和请求报文函数**，使其能够发送和接收ARP数据报文，并且能通过实验评测系统的测试。

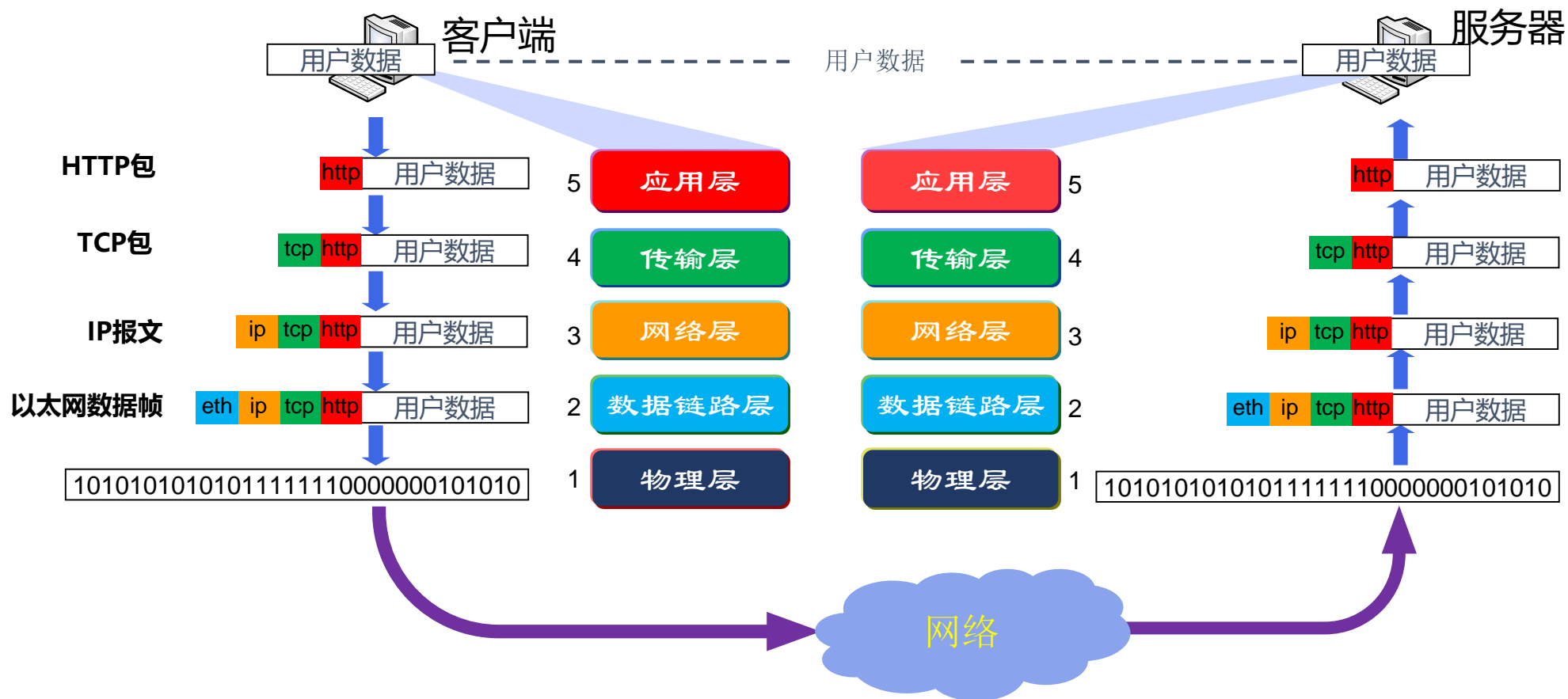


只有敲代码才能
感受到温暖



TCP/IP分层结构

数据包封装与解封装过程





1 如何捕获底层数据包？

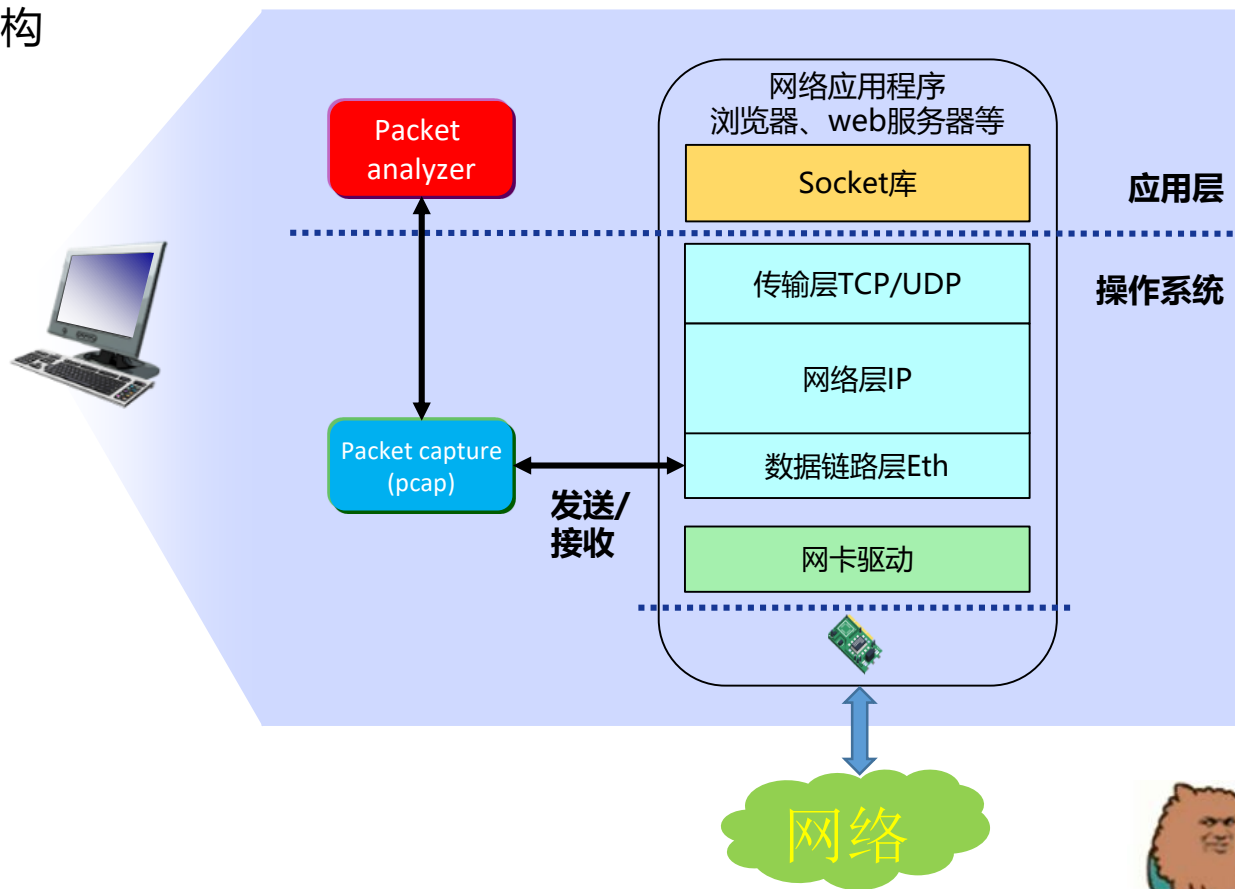
- **要求：** 不可改变操作系统的协议栈结构

➤ 分组捕获库（npcap库）

- 拷贝/过滤网卡接收到的链路层数据包

➤ 已封装好的驱动层

```
int driver_open();  
int driver_recv(buf_t *buf);  
int driver_send(buf_t *buf);  
void driver_close();
```



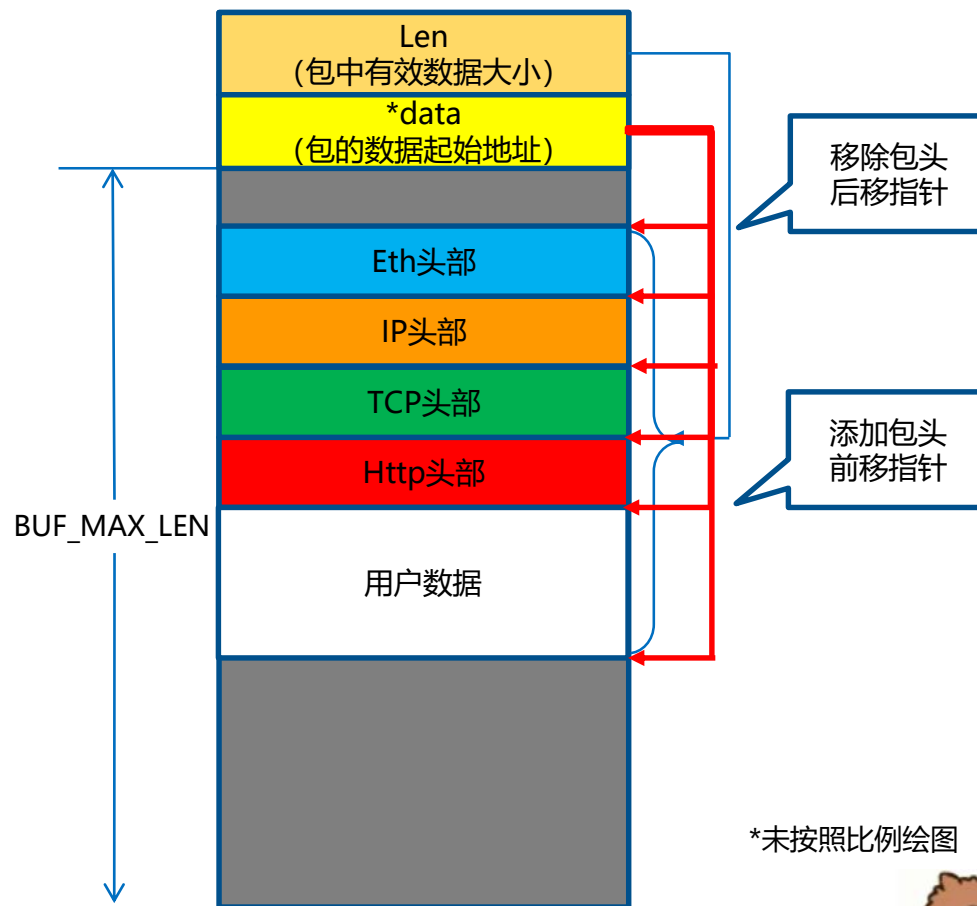


2 如何定义数据包结构?

➤ 要求:

- 数据包长度不固定，可长可短
- 支持不同协议层的头部添加和移除

```
8 //协议栈的通用数据包buffer, 可以在头部装卸数据, 以供协议头的添加和去除
9 typedef struct buf
10 {
11     size_t len;           // 包中有效数据大小
12     uint8_t *data;        // 包的数据起始地址
13     uint8_t payload[BUF_MAX_LEN]; // 最大负载数据量
14 } buf_t;
15
16 int buf_init(buf_t *buf, size_t len);
17 int buf_add_header(buf_t *buf, size_t len);
18 int buf_remove_header(buf_t *buf, size_t len);
19 int buf_add_padding(buf_t *buf, size_t len);
20 int buf_remove_padding(buf_t *buf, size_t len);
21 void buf_copy(void *pdst, const void *psrc, size_t len);
```



*未按照比例绘图



只有敲代码才能
感受到温暖



3

如何转换大小端存储?

➤ TCP/IP 协议规定:

网络字节序是大端字节序;

X86平台是以小端字节序存储

- 发送: 小端字节序转换成大端字节序
- 接收: 大端字节序转换成小端字节序

```
19 #pragma pack(1) // 编译器将按照 1 个字节对齐
20 typedef struct ether_hdr
21 {
22     uint8_t dest[NET_MAC_LEN]; // 目标 mac 地址
23     uint8_t src[NET_MAC_LEN]; // 源 mac 地址
24     uint16_t protocol; // 协议/长度
25 } ether_hdr_t;
26 #pragma pack() // 取消自定义字节对齐方式
```

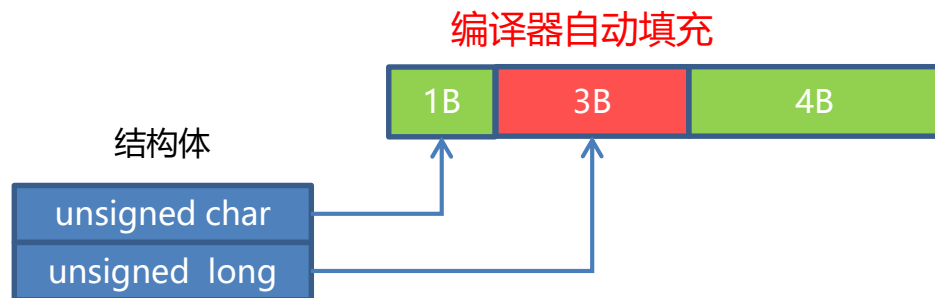
#define swap16(x) (((x)&0xFF) << 8) | (((x) >> 8) & 0xFF) // 为16位数据交换大小端

4

如何解决结构体对齐?

➤ 要求:

禁用结构体对齐



请思考

MAC地址需不需要大小端转换?



只有敲代码才能
感受到温暖

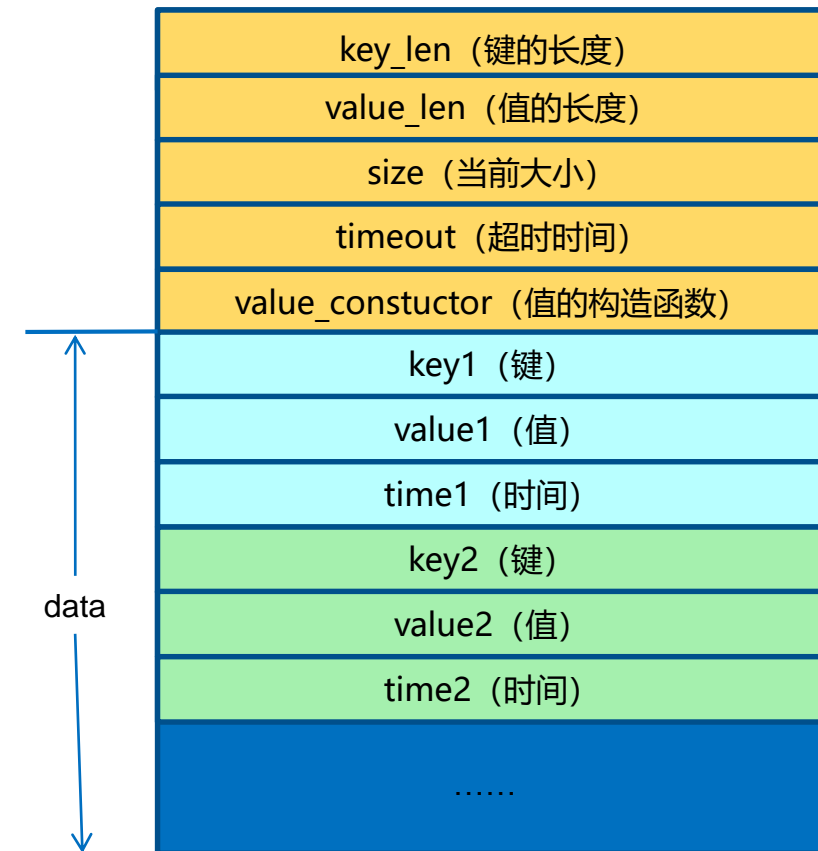


5 Map的使用

- 使用指针操作编写了键值对容器
 - 支持get、set、delete、foreach操作
 - 实现协议动态注册机制

```
typedef void (*map_constructor_t)(void *dst, const void *src, size_t len);
typedef void (*map_entry_handler_t)(void *key, void *value, time_t *timestamp);

typedef struct map // 协议栈的通用泛型map, 即键值对的容器, 支持超时时间与非平凡值类型
{
    size_t key_len;           // 键的长度
    size_t value_len;        // 值的长度
    size_t size;              // 当前大小
    size_t max_size;         // 最大容量
    time_t timeout;          // 超时时间, 0为永不超时
    map_constructor_t value_constructor; // 形如memcpy的值构造函数, 用于拷贝非平凡数据结构到容器中,
    uint8_t data[MAP_MAX_LEN]; // 数据
} map_t;
```

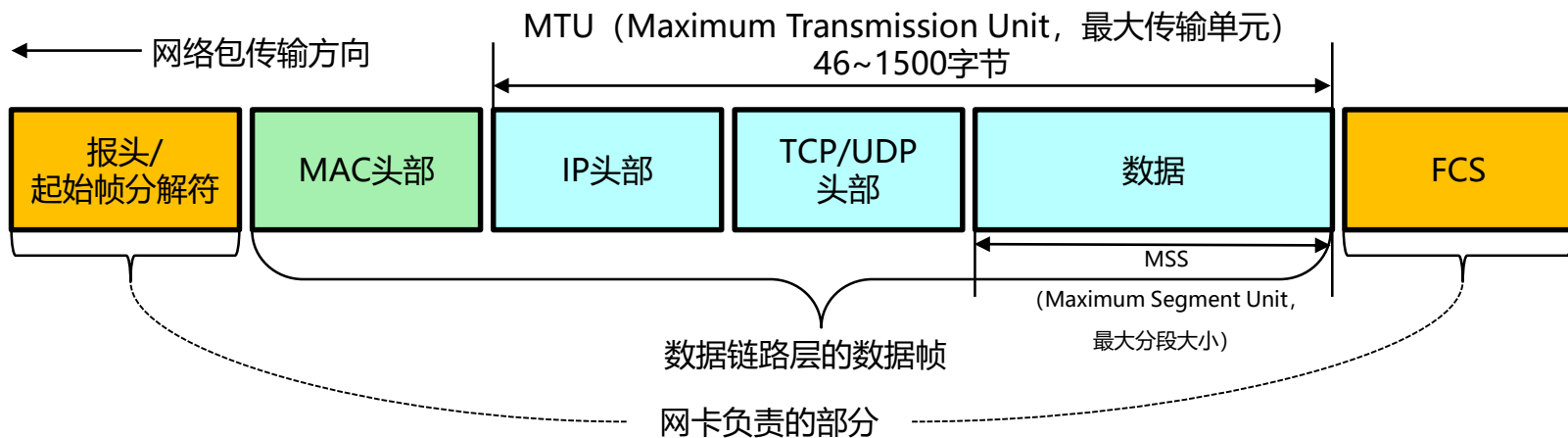




实验原理 – 以太网数据帧格式



➤ 在网线上传输的数据包格式：



MAC头部			
MAC目标地址	MAC源地址	长度/类型	负载
6 字节	6字节	2字节	46~1500字节
		0800	IP数据
		0806	ARP请求/应答
			PAD
			28字节 18字节



只有敲代码才能
感受到温暖



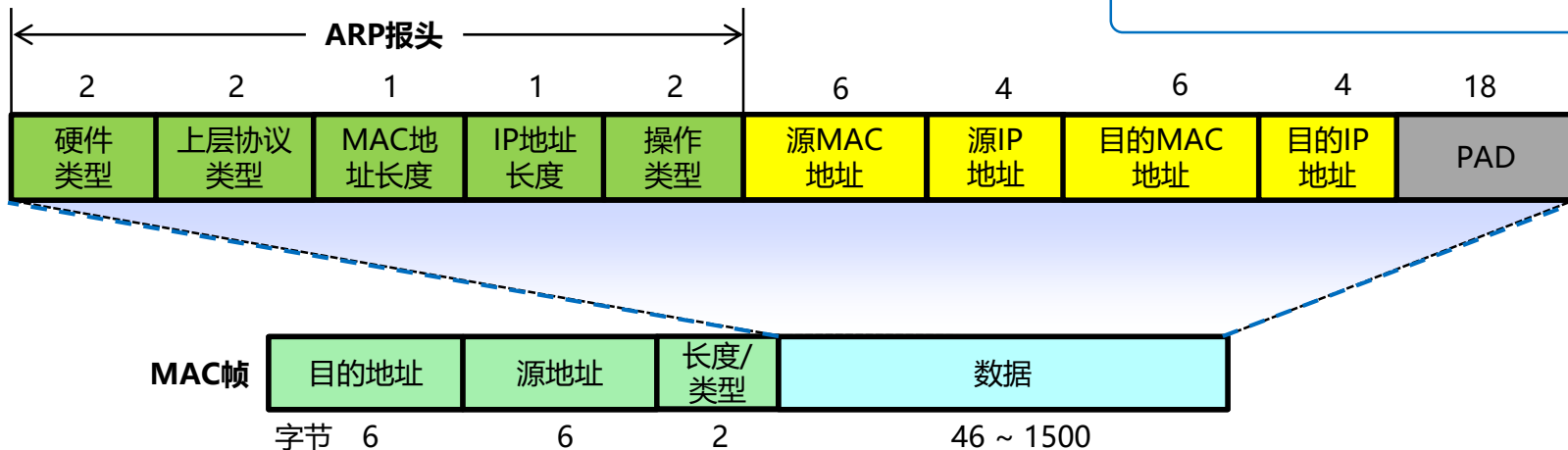
实验原理 – ARP报文格式



➤ **ARP**: 将目的IP地址解析成MAC地址

请思考

当网速大于10Mbps时,
能发送小于46字节的报文吗?



PAD原因

最初传统以太网速率是10Mbps，为了确保帧的发送时间大于51.2us（争用期），CSMA/CD协议规定最小帧长是64字节（MAC头部14B + 46B + FCS 4B）



只有敲代码才能
感受到温暖



本次实验的目标



- ✓ Eth数据帧的发送和接收，通过测试
- ✓ ARP报文的发送、接收和请求报文函数，通过测试

src/ethernet.c

```
void ethernet_out(buf_t *buf, const uint8_t *mac, net_protocol_t protocol)
{
    // TO-DO
}

11 void ethernet_in(buf_t *buf)
12 {
13     // TO-DO
14 }
```

src/arp.c

```
93 void arp_out(buf_t *buf, uint8_t *ip)
94 {
95     // TO-DO
96 }

59 void arp_req(uint8_t *target_ip)
60 {
61     // TO-DO
62 }

81 void arp_in(buf_t *buf, uint8_t *src_mac)
82 {
83     // TO-DO
84 }

70 void arp_resp(uint8_t *target_ip, uint8_t *target_mac)
71 {
72     // TO-DO
73 }
```



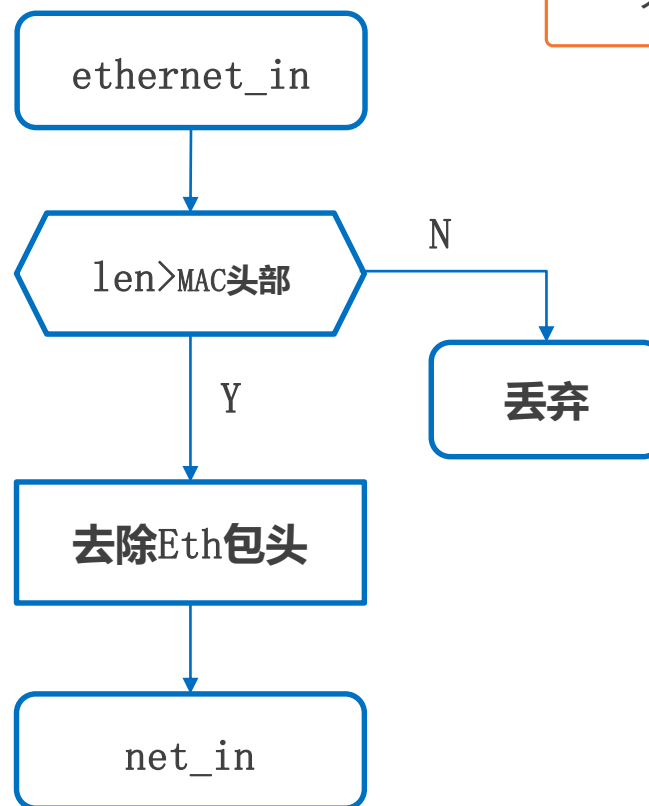
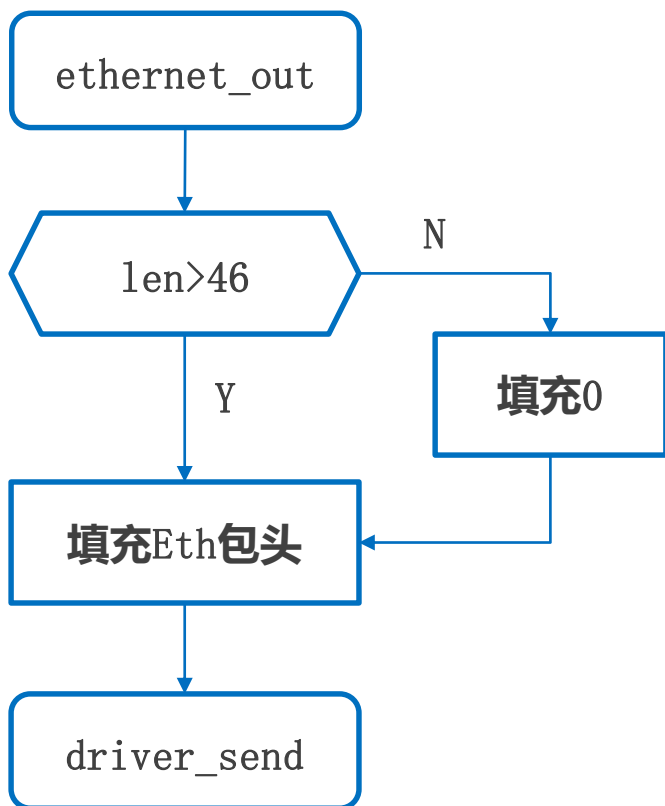
只有敲代码才能
感受到温暖



✓ Eth数据帧的发送和接收

注意

大小端转换



```
buf_add_header(buf, sizeof(ether_hdr_t));  
ether_hdr_t *hdr = (ether_hdr_t *)buf->data;
```

*详细步骤请参考指导书



只有敲代码才能
感受到温暖



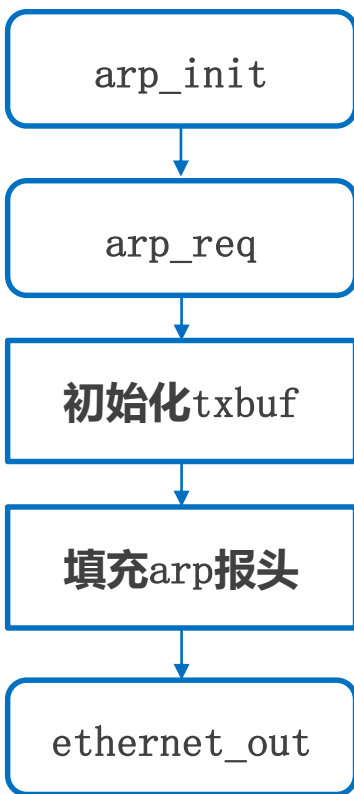
实验步骤



✓ ARP报文的发送、接收和请求报文函数

无回报ARP包

启用网卡时告诉所有人自己的IP地址和MAC地址



No.	Time	Source	Destination	Protocol	Length	Info
500	11.543911	11:22:33:44:55:66	Broadcast	ARP	60	ARP Announcement for 192.168.163.103
834	24.737103	VMware_c0:00:08	Broadcast	ARP	42	Who has 192.168.163.2? Tell 192.168.163.1
1016	37.988491	VMware_c0:00:08	Broadcast	ARP	42	Who has 192.168.163.2? Tell 192.168.163.1
1032	38.736544	VMware_c0:00:08	Broadcast	ARP	42	Who has 192.168.163.2? Tell 192.168.163.1
1042	39.736488	VMware_c0:00:08	Broadcast	ARP	42	Who has 192.168.163.2? Tell 192.168.163.1

> Frame 500: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{59662C36-F150-4042-A5DB-0DFF}
> Ethernet II, Src: 11:22:33:44:55:66 (11:22:33:44:55:66), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Destination: Broadcast (ff:ff:ff:ff:ff:ff) 目标MAC是广播地址
Address: Broadcast (ff:ff:ff:ff:ff:ff)
.... ..1. = LG bit: Locally administered address (this is NOT the factory default)
.... ..1. = IG bit: Group address (multicast/broadcast)
> Source: 11:22:33:44:55:66 (11:22:33:44:55:66)
> [Expert Info (Warning/Protocol): Source MAC must not be a group address: IEEE 802.3-2002, Section 3.2.3(b)]
Address: 11:22:33:44:55:66 (11:22:33:44:55:66)
.... ..0. = LG bit: Globally unique address (factory default)
.... ..1. = IG bit: Group address (multicast/broadcast)
Type: ARP (0x0806)
Padding: 00000000000000000000000000000000
> Address Resolution Protocol (ARP Announcement)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (1)
[Is gratuitous: True]
[Is announcement: True]
Sender MAC address: 11:22:33:44:55:66 (11:22:33:44:55:66)
Sender IP address: 192.168.163.103 发送方地址: 填自己的
Target MAC address: 00:00:00 00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.163.103 接收方地址: 填自己的

0000	ff ff ff ff ff ff	11 22 33 44 55 66	08 06 00 01" 3DUf ..
0010	08 00 06 04 00 01	11 22 33 44 55 66	c0 a8 a3 67" 3DUf ..g
0020	00 00 00 00 00 00	c0 a8 a3 67 00 00	00 00 00 00g.....
0030	00 00 00 00 00 00	00 00 00 00	

*详细步骤请参考指导书



只有敲代码才能感受到温暖



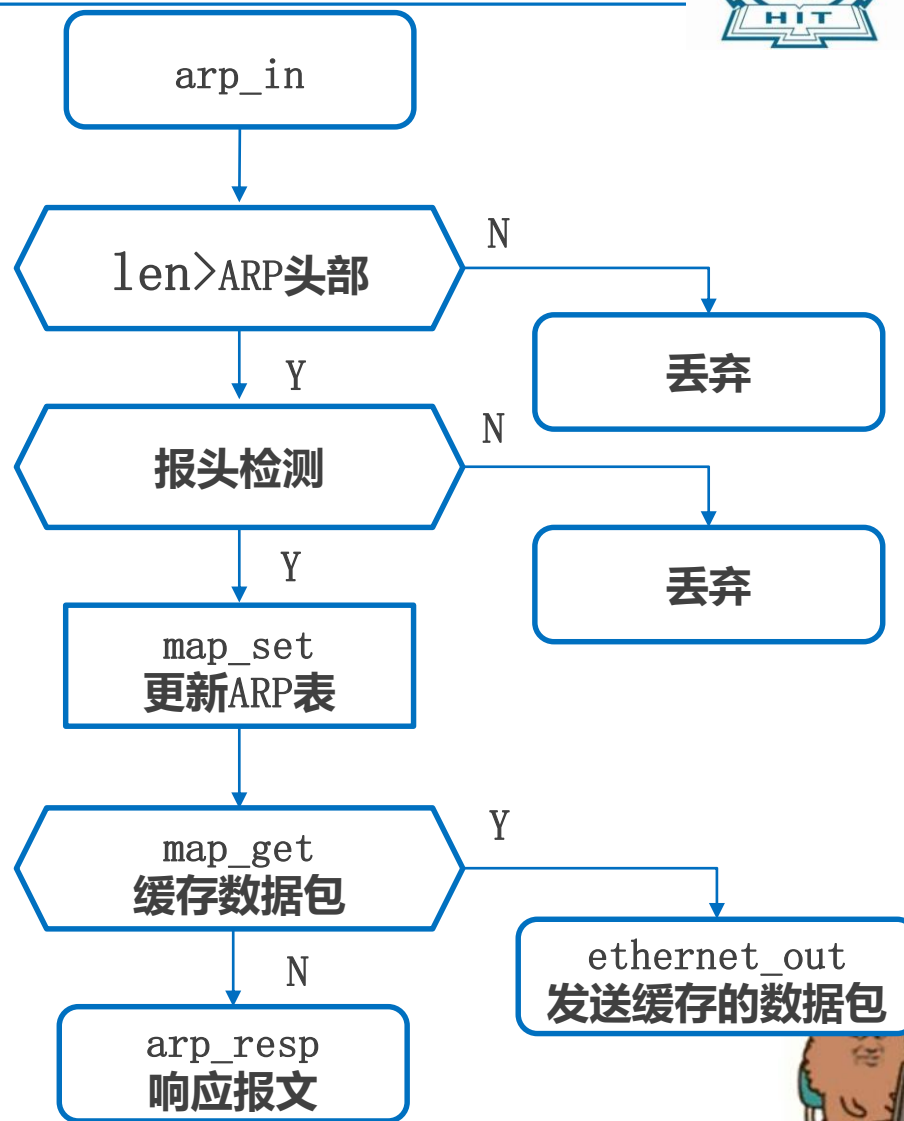
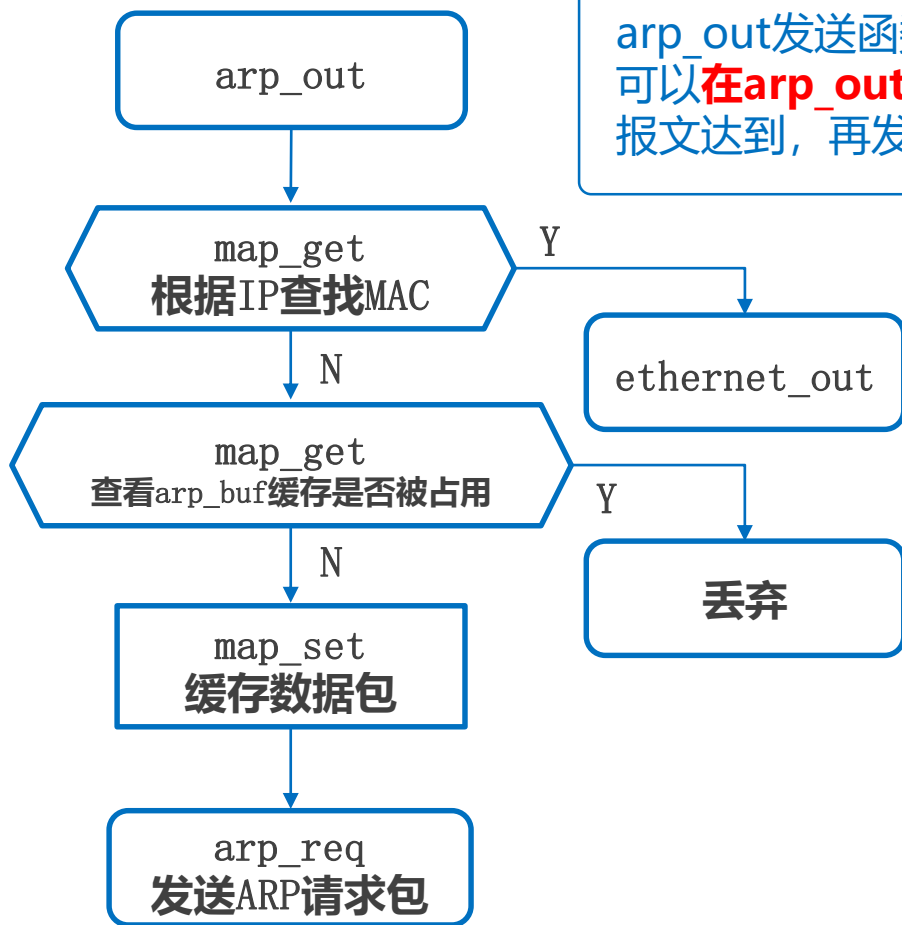
实验步骤



✓ ARP报文的发送、接收和请求报文函数

请思考

arp_out发送函数中有发送ARP请求包，可以在**arp_out函数中等待**该请求的回应报文达到，再发送缓存的数据包吗？



*详细步骤请参考指导书



只有敲代码才能
感受到温暖



实验自测

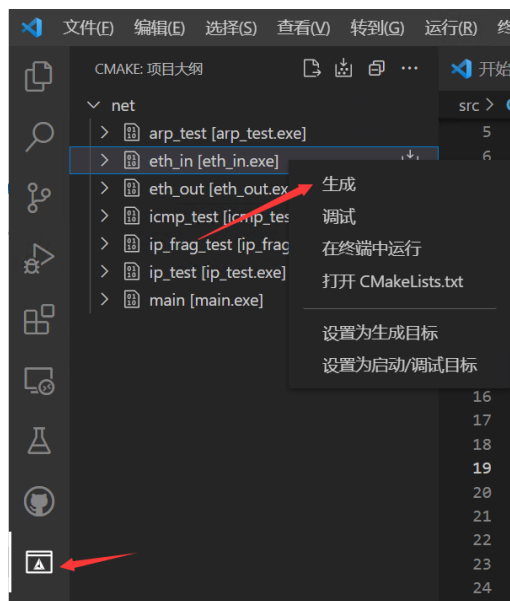


✓ 请参考附录B 网络编程实验环境搭建Windows或Linux的开发环境

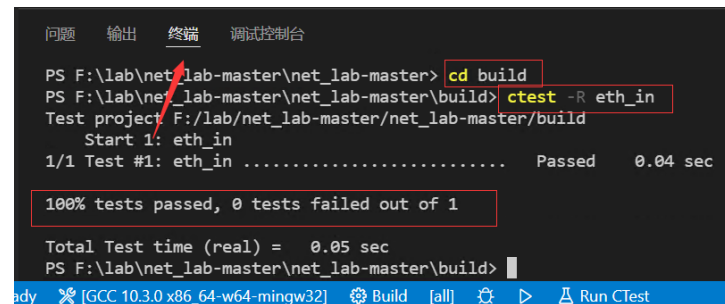
演示视频

【网络协议栈编译调试】

https://www.bilibili.com/video/BV1kT411q7aB/?share_source=copy_web



测试通过



只有敲代码才能
感受到温暖



实验自测



利用VSCode的比较工具来查找bug

测试失败

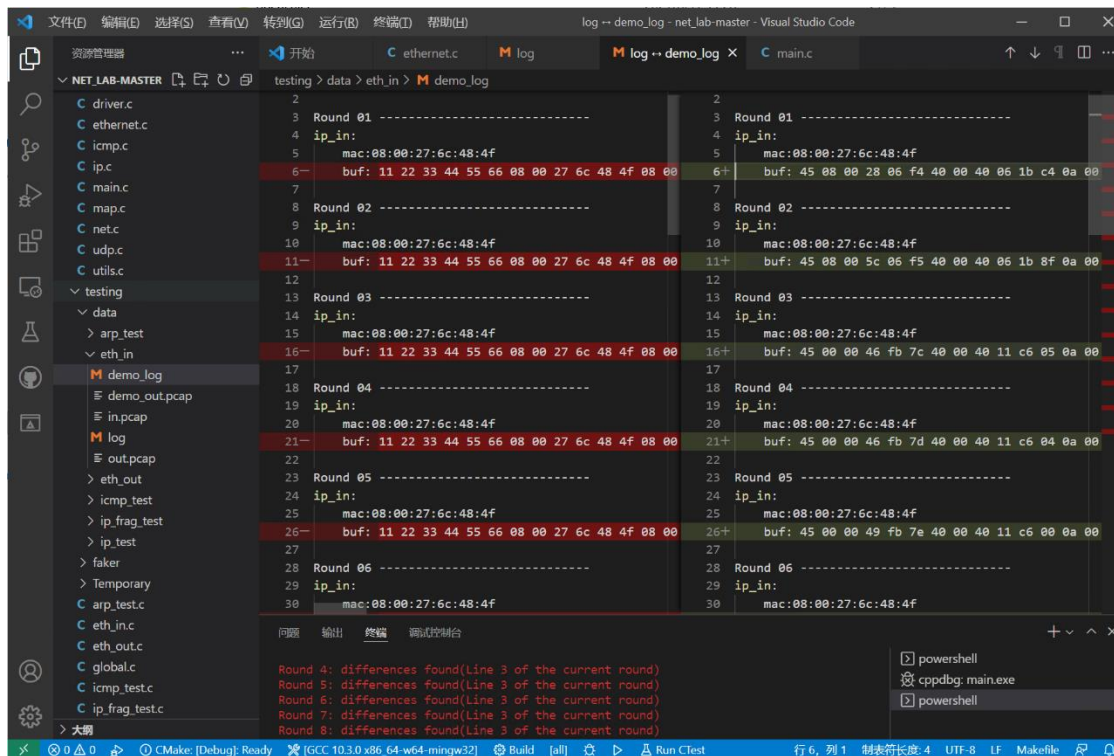
```
问题 输出 终端 调试控制台

PS F:\lab\net_lab-master\net_lab-master> cd build
PS F:\lab\net_lab-master\net_lab-master\build> ctest -R eth_in
Test project F:/lab/net_lab-master/net_lab-master/build
Start 1: eth_in
1/1 Test #1: eth_in .....***Failed    0.05 sec

0% tests passed, 1 tests failed out of 1

Total Test time (real) = 0.05 sec

The following tests FAILED:
  1 - eth_in (Failed)
Errors while running CTest
Output from these tests are in: F:/lab/net_lab-master/net_lab-master/build/Testing/Temporary/LastTest.log
Use "--rerun-failed --output-on-failure" to re-run the failed cases verbosely.
PS F:\lab\net_lab-master\net_lab-master\build>
```



只有敲代码才能
感受到温暖



提交内容： 你所修改过的代码 + 实验报告（有模板）

截止时间：

实验课后两周内提交至HITsz Grader 作业提交平台，具体截止日期参考平台发布。

- 登录网址：：<http://grader.tery.top:8000/#/login>
- 推荐浏览器： Chrome
- 初始用户名、密码均为学号，登录后请修改

注意

上传后可自行下载以确认是否正确提交



只有敲代码才能
感受到温暖



**同学们
请开始实验吧！**