计算机网络实验

第四次实验

CONTENTS
目录

实验任务

Lab5 协议栈之IP协议实现

- 在完成ARP的基础上，编写**IP报文的发送、接收、IP分片以及计算校验和函数**，使其能够发送和接收IP数据报文，并且能通过实验评测系统的测试。

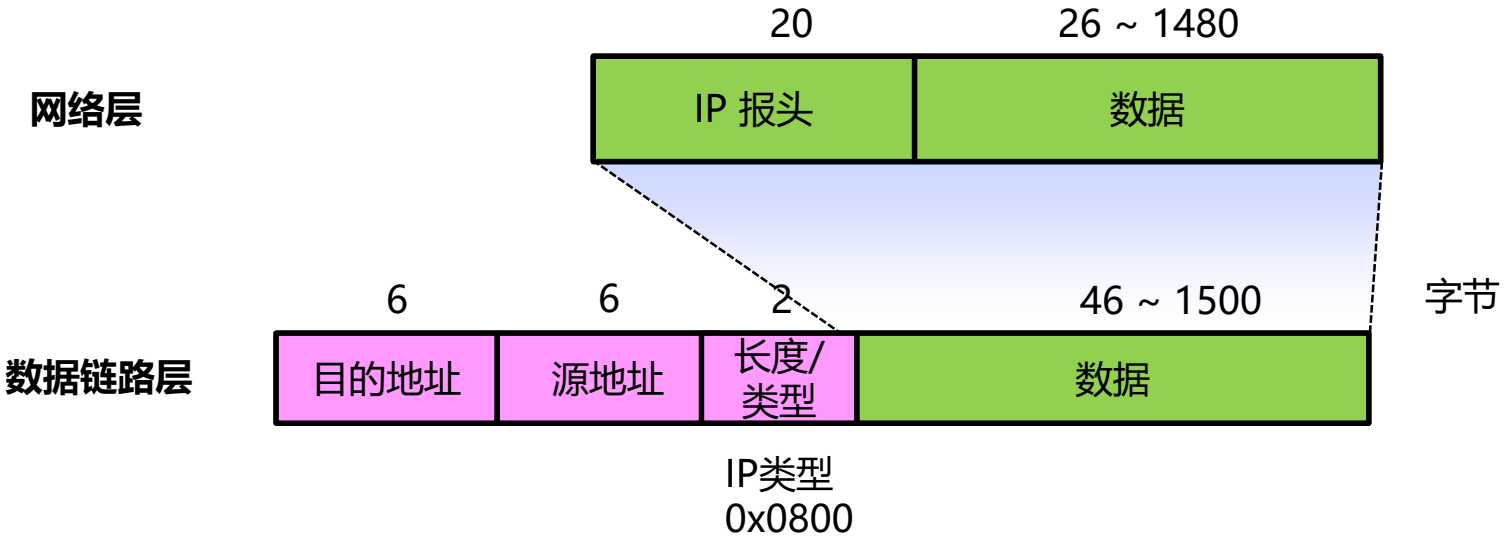
Lab6 协议栈之ICMP协议实现

- 在完成IP协议的基础上，编写ICMP报文的**接收、响应报文和差错报文**，使其能够发送和接收ICMP数据报文，并且能通过实验评测系统的测试。

Lab7 协议栈之UDP协议实现

- 在完成ICMP协议的基础上，编写UDP报文的**发送、接收和UDP校验和**函数，使其能够发送和接收UDP数据报文，并且**能与UDP测试工具进行自收自发**。

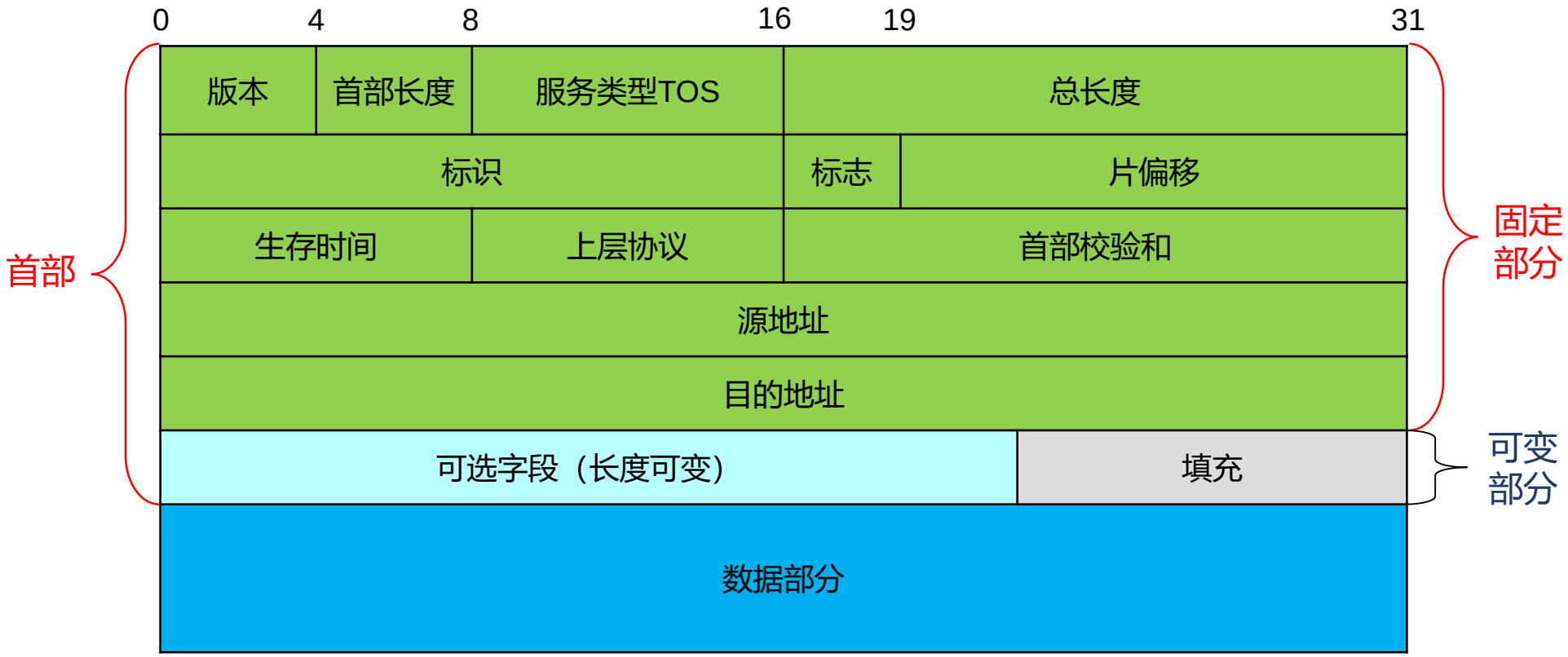
- **IP提供了一种尽力而为、无连接的数据交付服务**
  - **尽力而为**：不保证IP数据报能成功到达目的地
  - **无连接**：不维护任何链接状态，数据报互相独立，可不按顺序交付



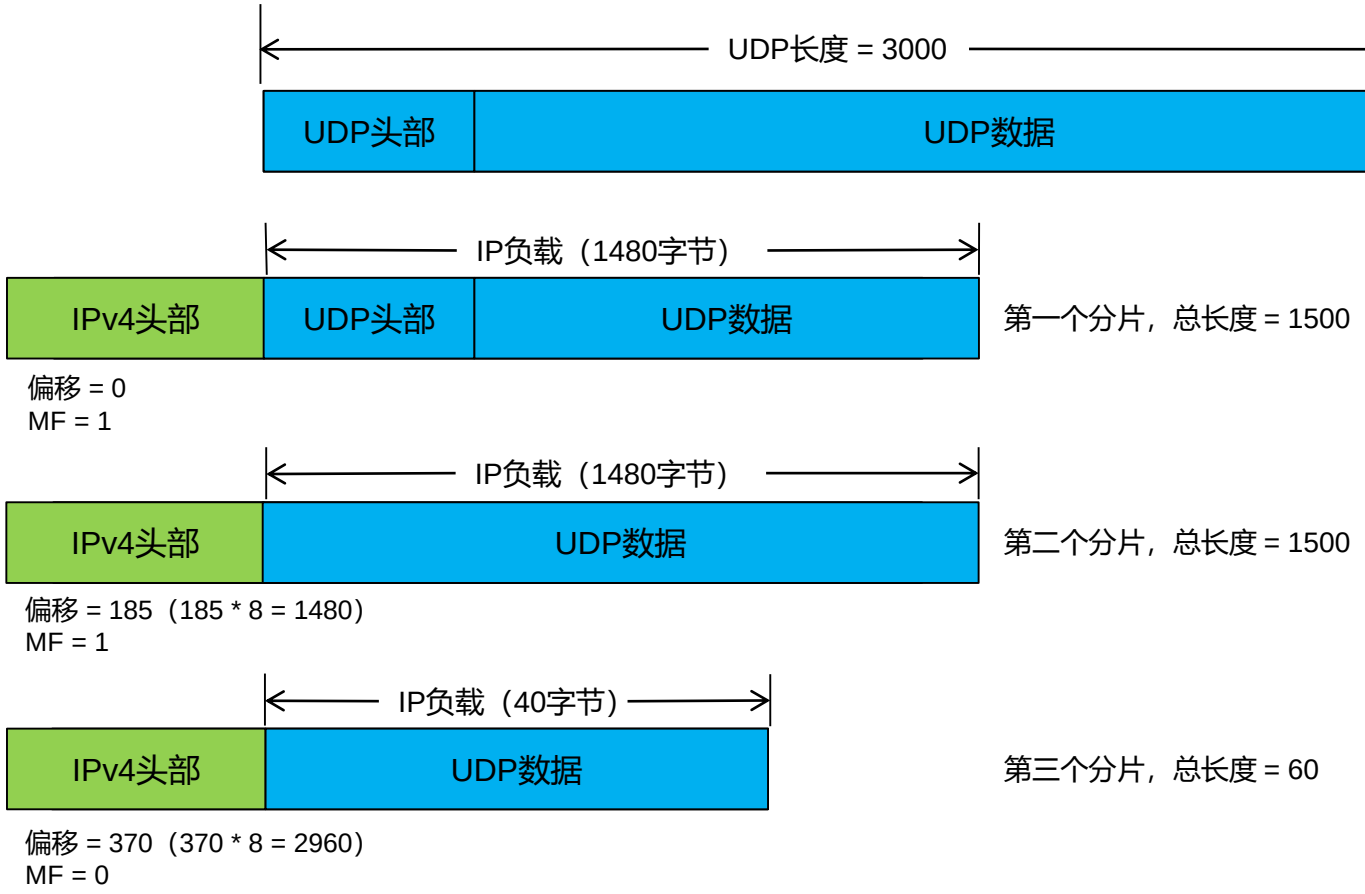
- **IPv4头部格式：**
  - 只考虑固定20字节的IP首部

请思考

总长度为16bit（65535字节），能发这么大的数据包吗?



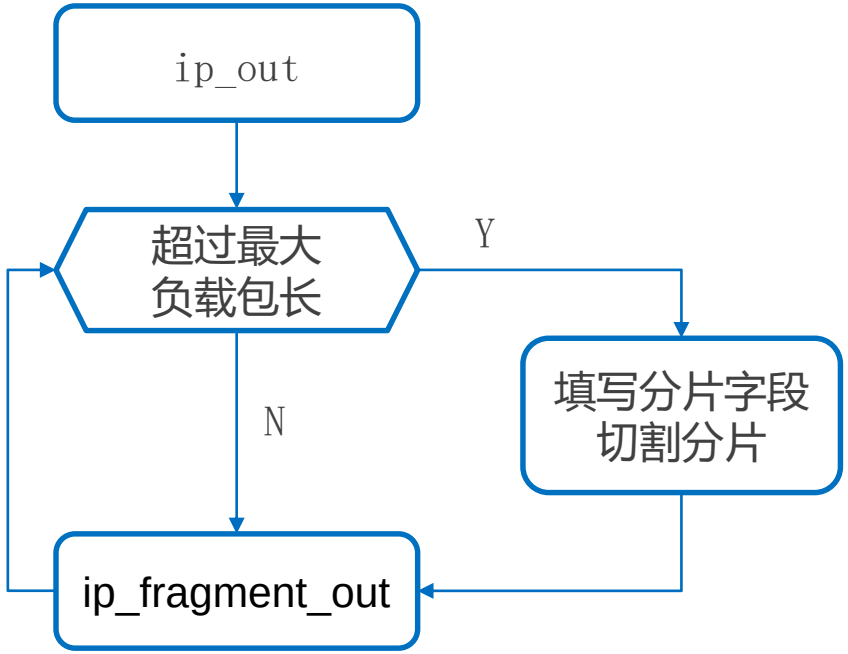
- **IP分片：**通过标识、标志、位偏移完成数据报的分片与重组操作

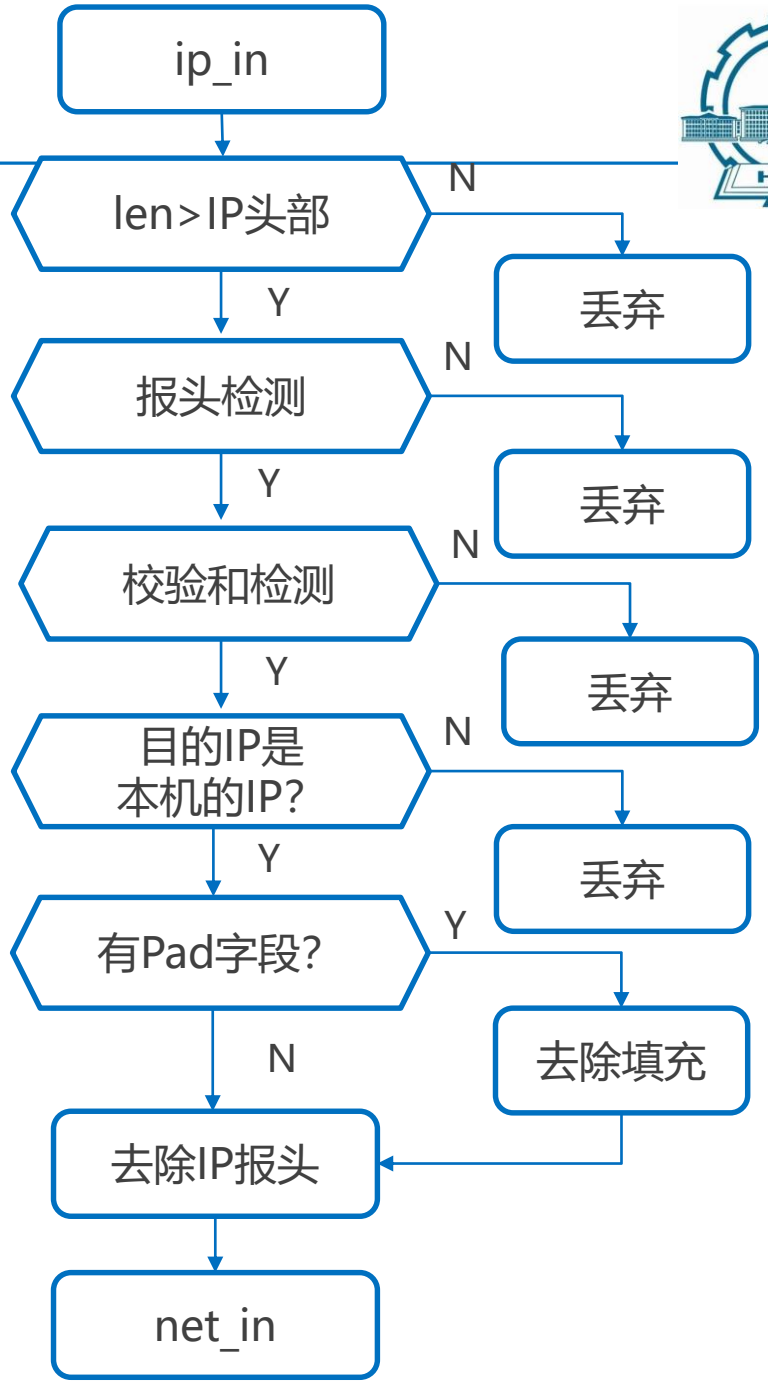


实验步骤

1 IP协议

✓ IP报文的发送、接收函数





IP头部校验和 — 反码求和

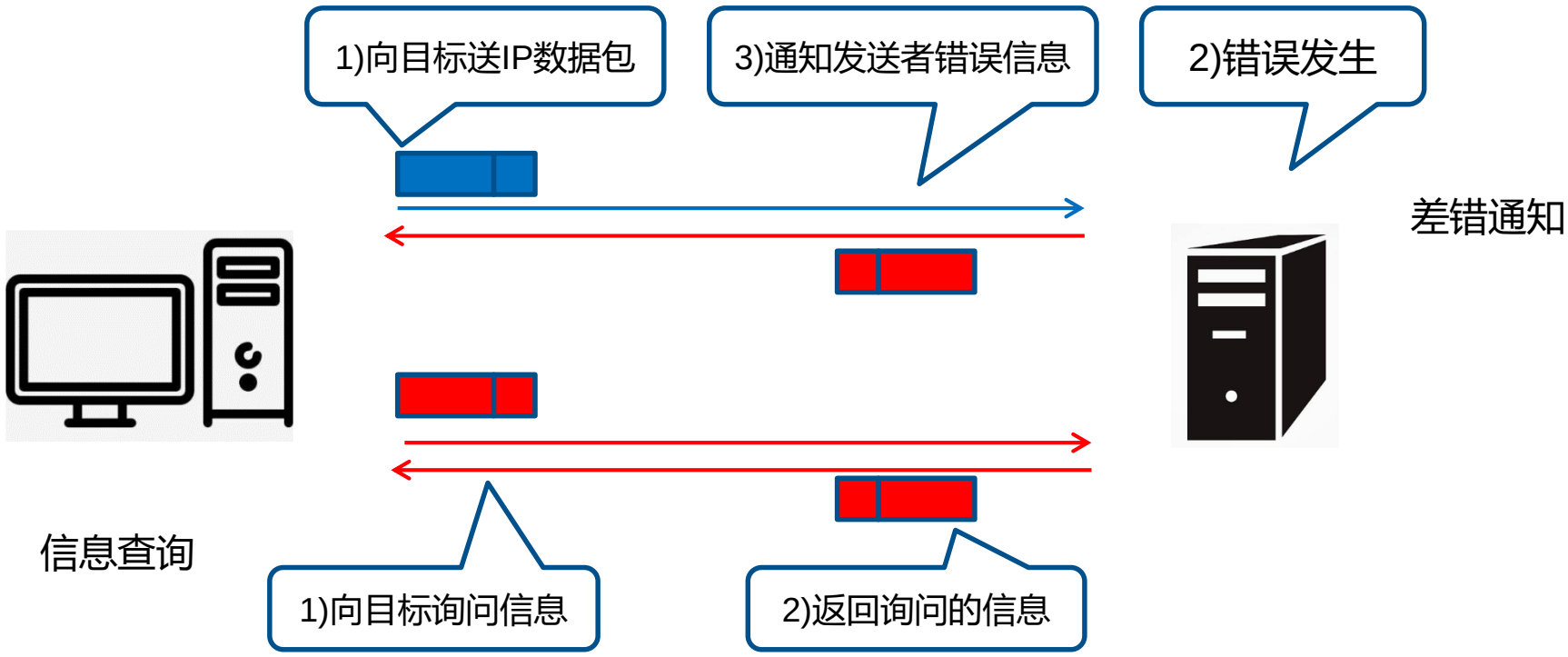
- 把数据看成由一串16bit的半字组成；
- 对两个二进制数直接进行加法运算；
- 如果最后还剩8个bit值，也要相加这个8bit值；
- 判断相加后32bit结果值的高16位是否为0，如果不为0，则将高16位和低16位相加，依次循环，直至高16位为0为止；
- 将上述的和（低16位）取反，即得到校验和。

请思考

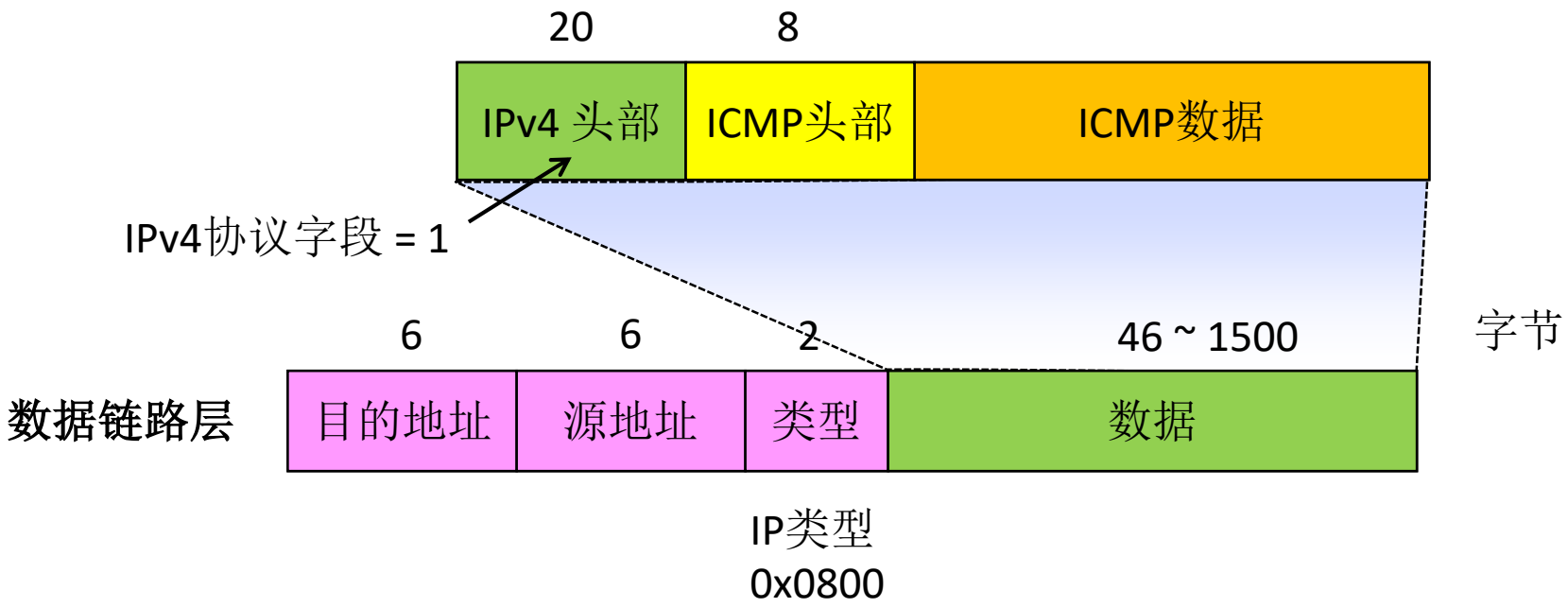
校验和16bit要做大小端转换吗?

2 ICMP协议

- **ICMP：Internet Control Message Protocol，网际报文控制协议**
  - **差错报文**：通知出错原因的错误消息
  - **查询报文**：用于诊断的查询消息

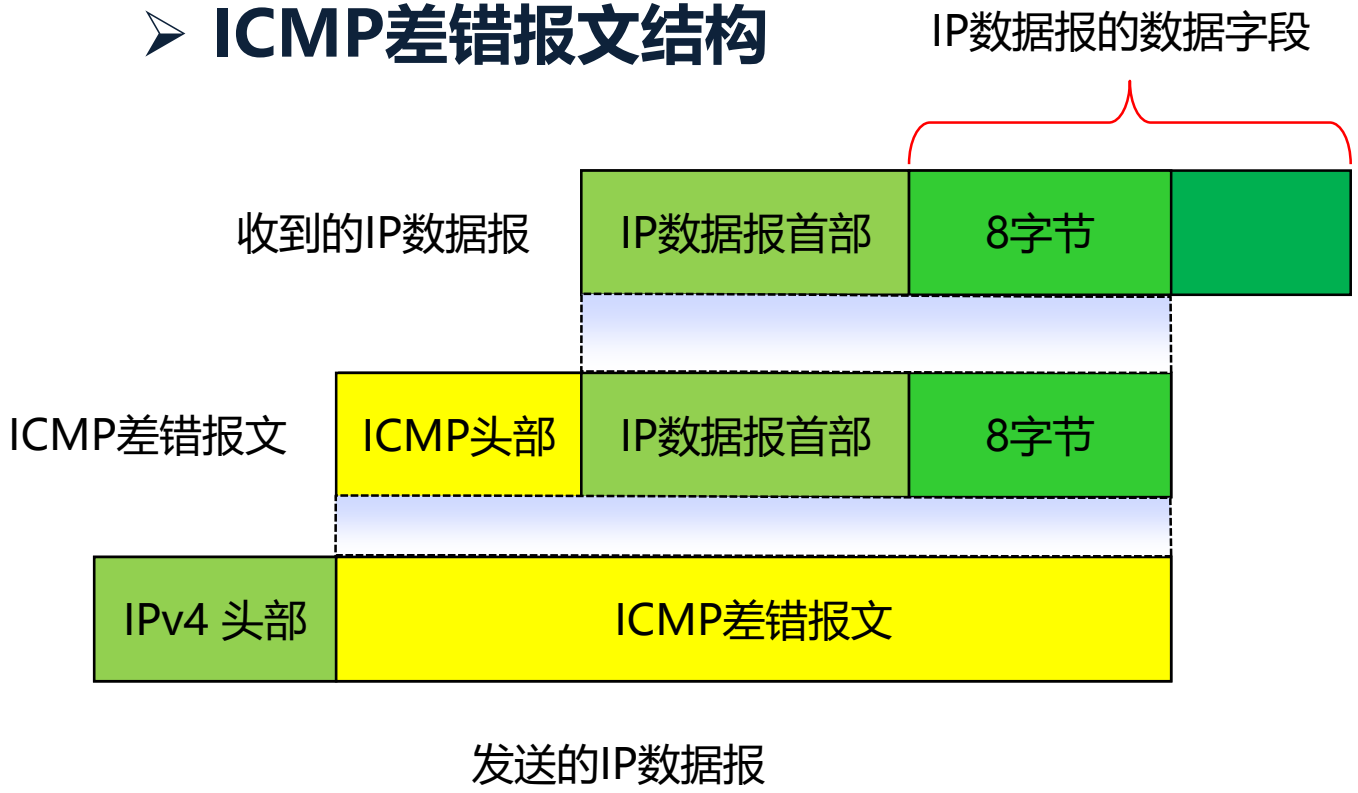


ICMP报文格式


20
8
IPv4 头部
ICMP头部
ICMP数据
IPv4协议字段 = 1
6
6
2
46 ~ 1500
字节
数据链路层
目的地址
源地址
类型
数据
IP类型
0x0800


只有敲代码才能
感受到温暖

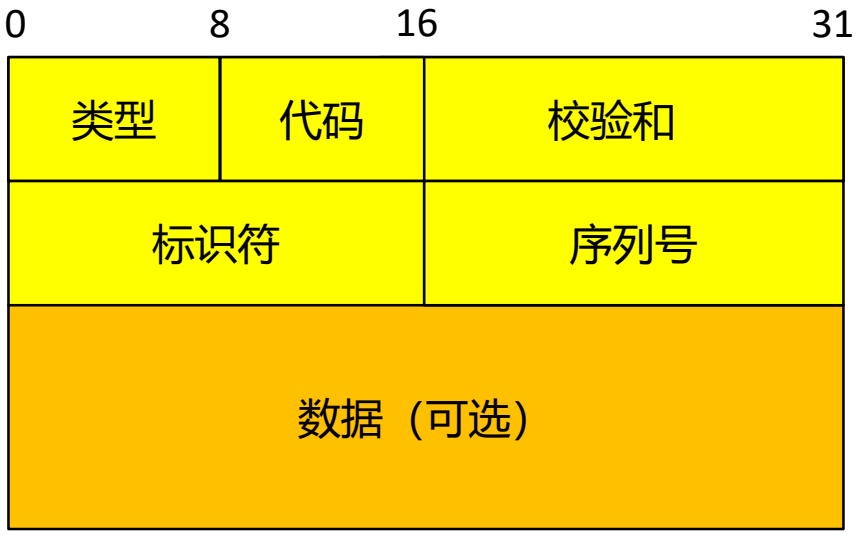
ICMP差错报文结构



请思考

为何要返回IP数据包中的前8个字节?

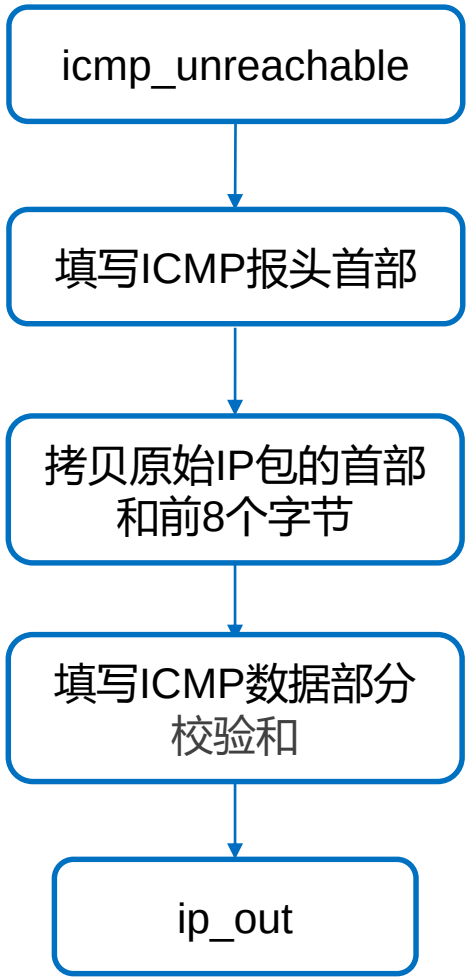
ICMP报查询报文结构

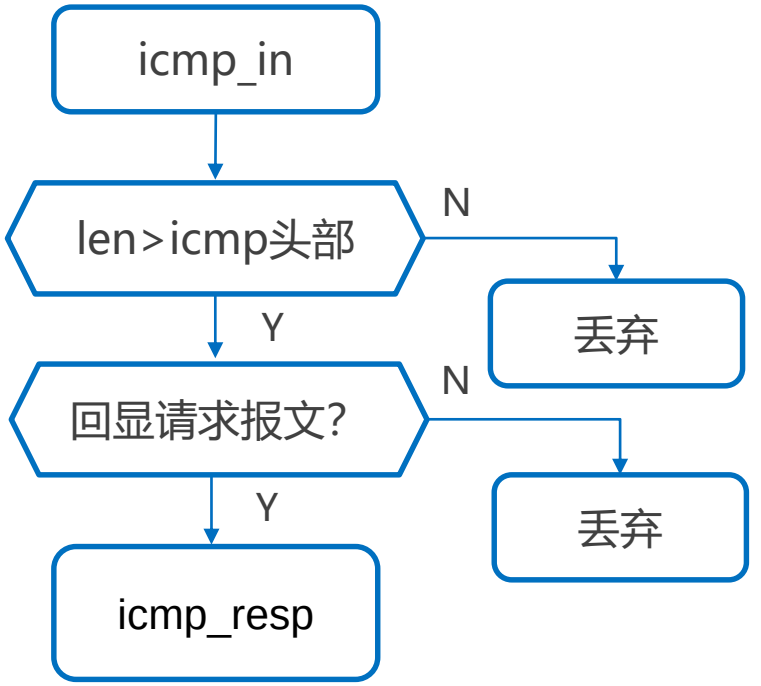


TYPE	CODE	Description
0	0	查询报文：Echo Reply——回显应答（Ping应答）
3	0	差错报文：Network Unreachable——网络不可达
3	1	差错报文：Host Unreachable——主机不可达
3	2	差错报文：Protocol Unreachable——协议不可达
3	3	差错报文：Port Unreachable——端口不可达
8	0	查询报文：Echo request——回显请求（Ping请求）
11	0	差错报文：TTL equals 0 during transit——传输期间生存时间为0
12	0	差错报文：IP header bad (catchall error)——坏的IP首部（包括各种差错）

2 ICMP协议

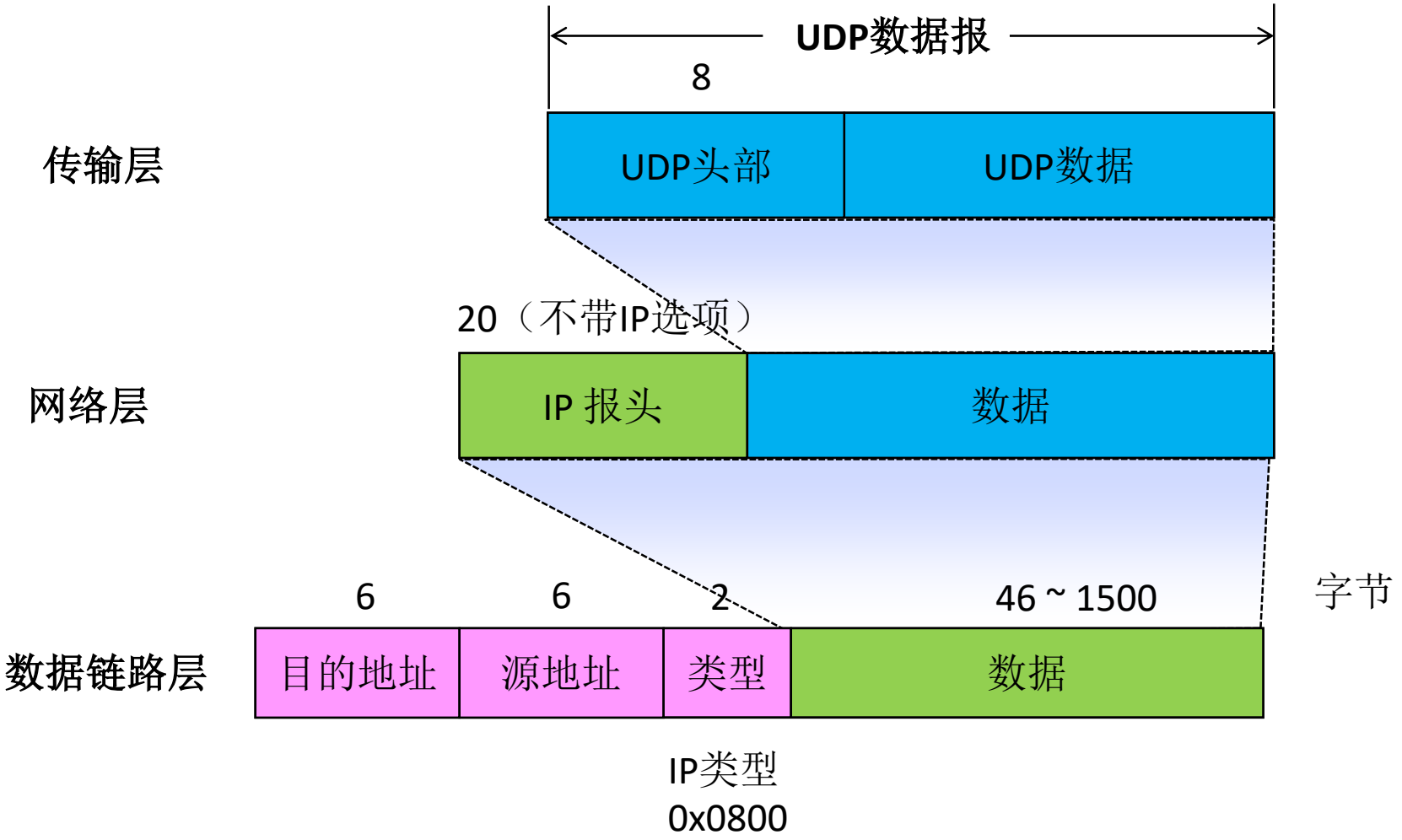
✓ ICMP报文的差错报文和接收响应报文






```
typedef enum icmp_code
{
    ICMP_CODE_PROTOCOL_UNREACH = 2, // 协议不可达
    ICMP_CODE_PORT_UNREACH = 3      // 端口不可达
} icmp_code_t;
```

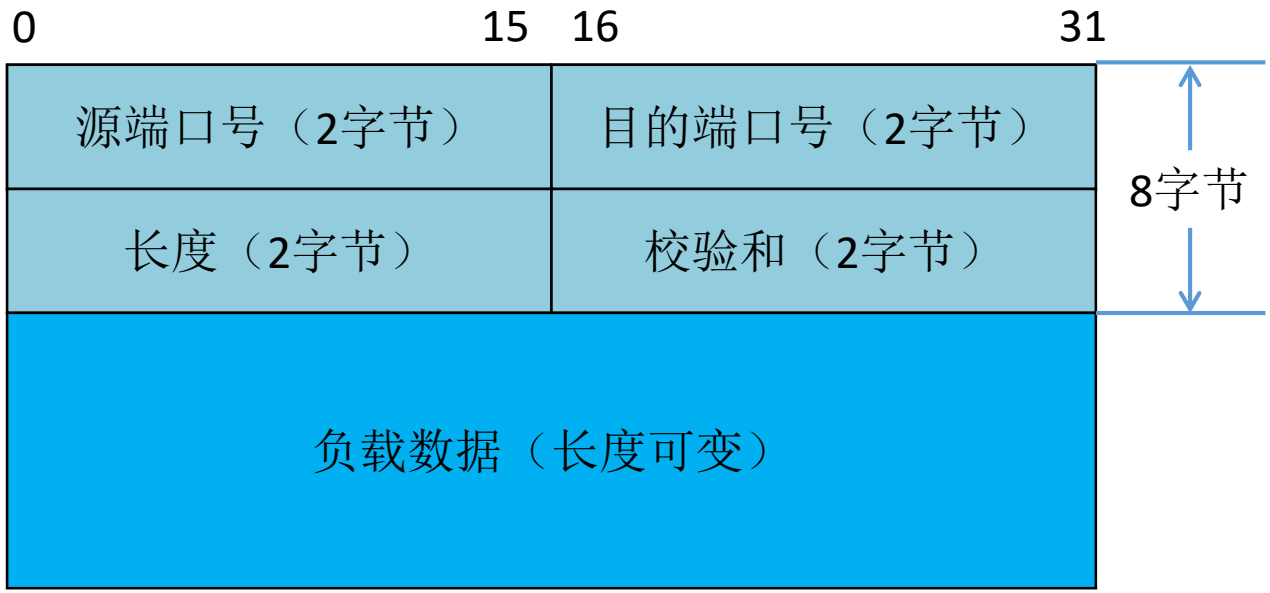
- **UDP**：用户数据报协议，是一种无连接的、不可靠的传输协议。



➢ UDP报文格式

注意

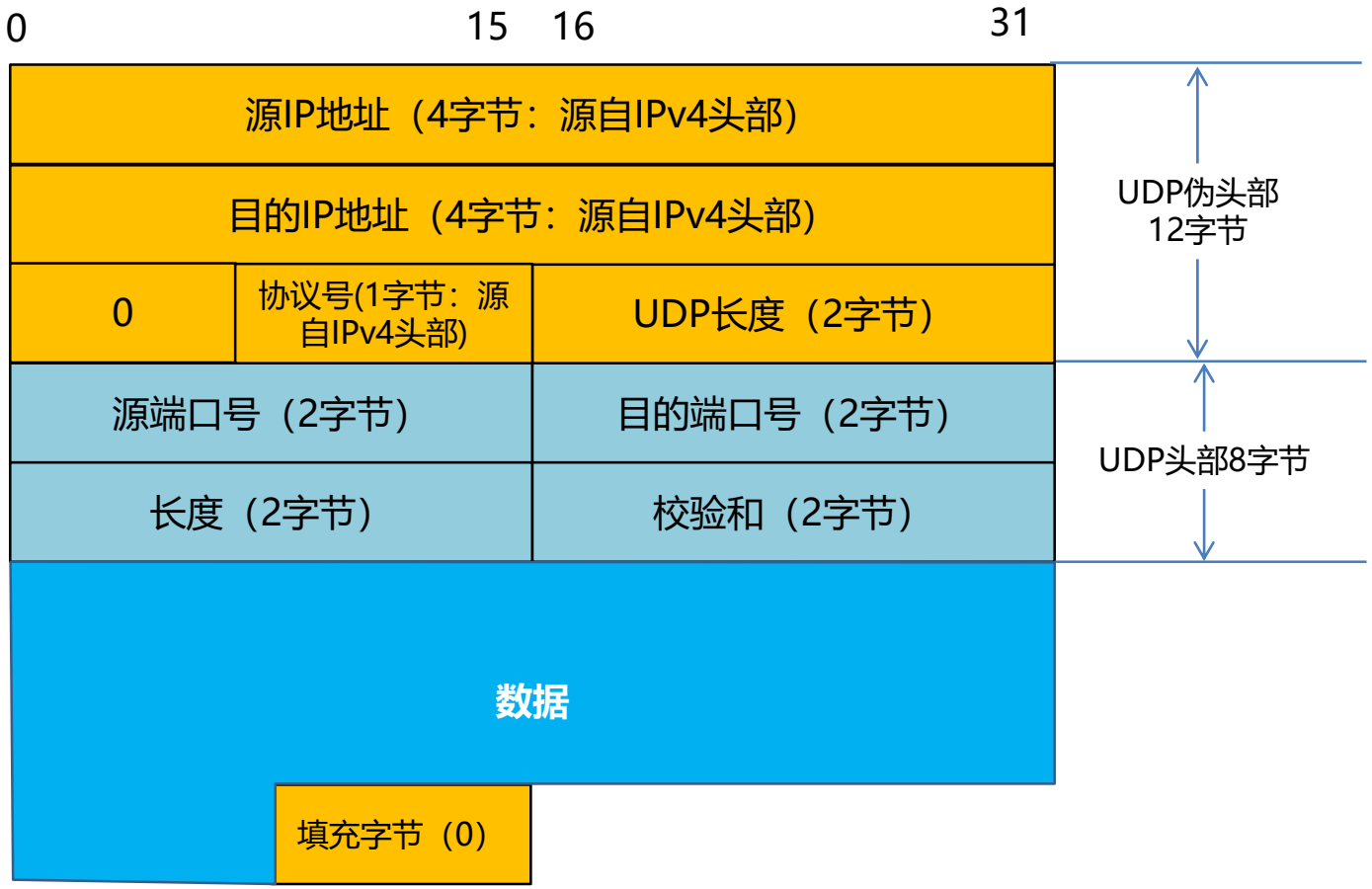
UDP最大能传输65507字节（65535 – 8 - 20）



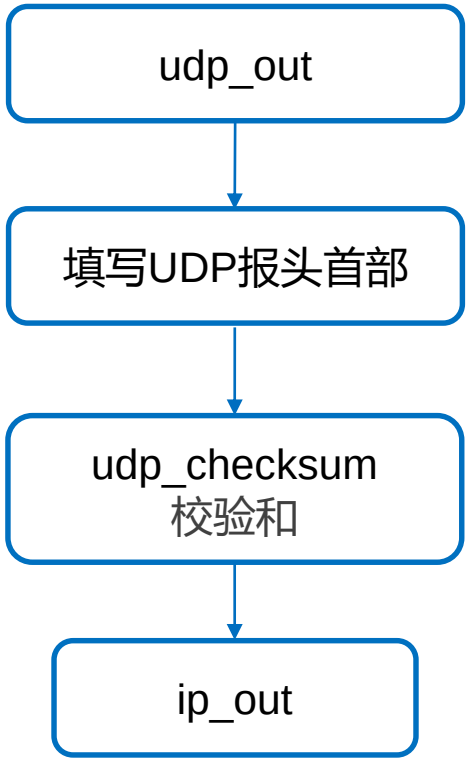
➢ UDP校验和：UDP头部、UDP数据和UDP伪头部

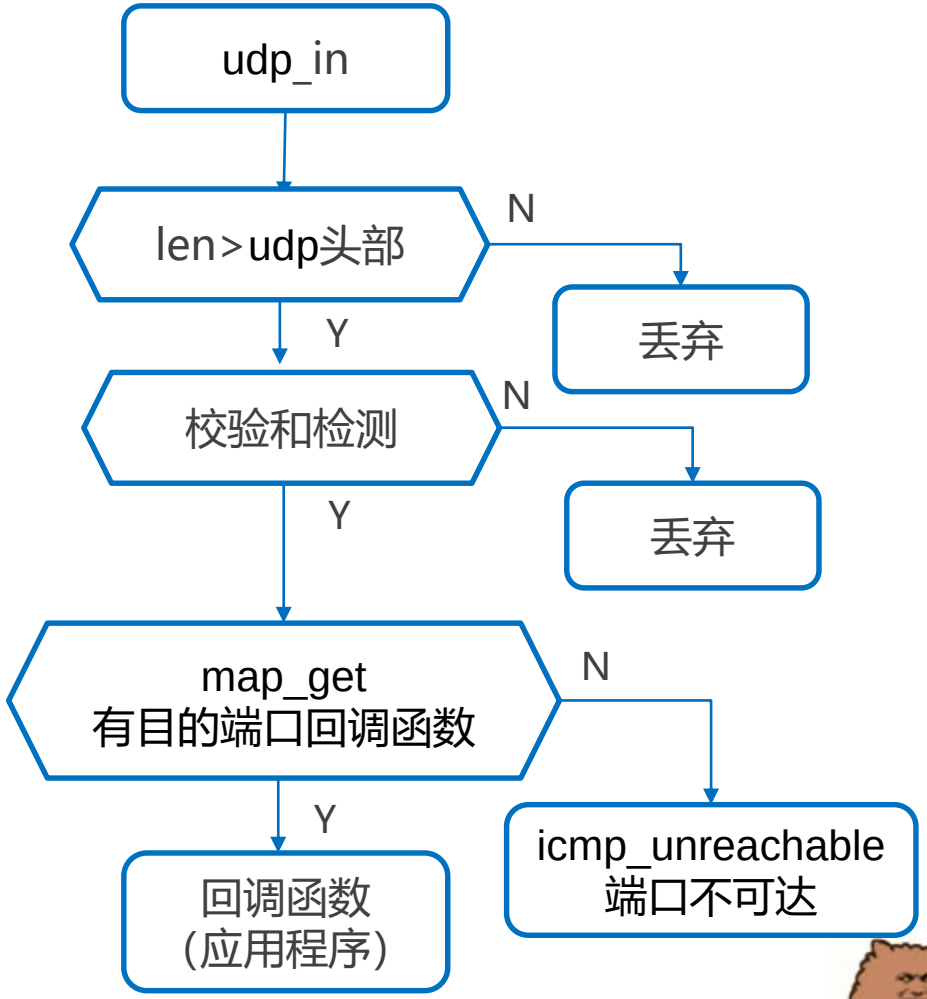
通信五元组

源IP地址
目标IP地址
协议号
源端口号
目标端口号协议

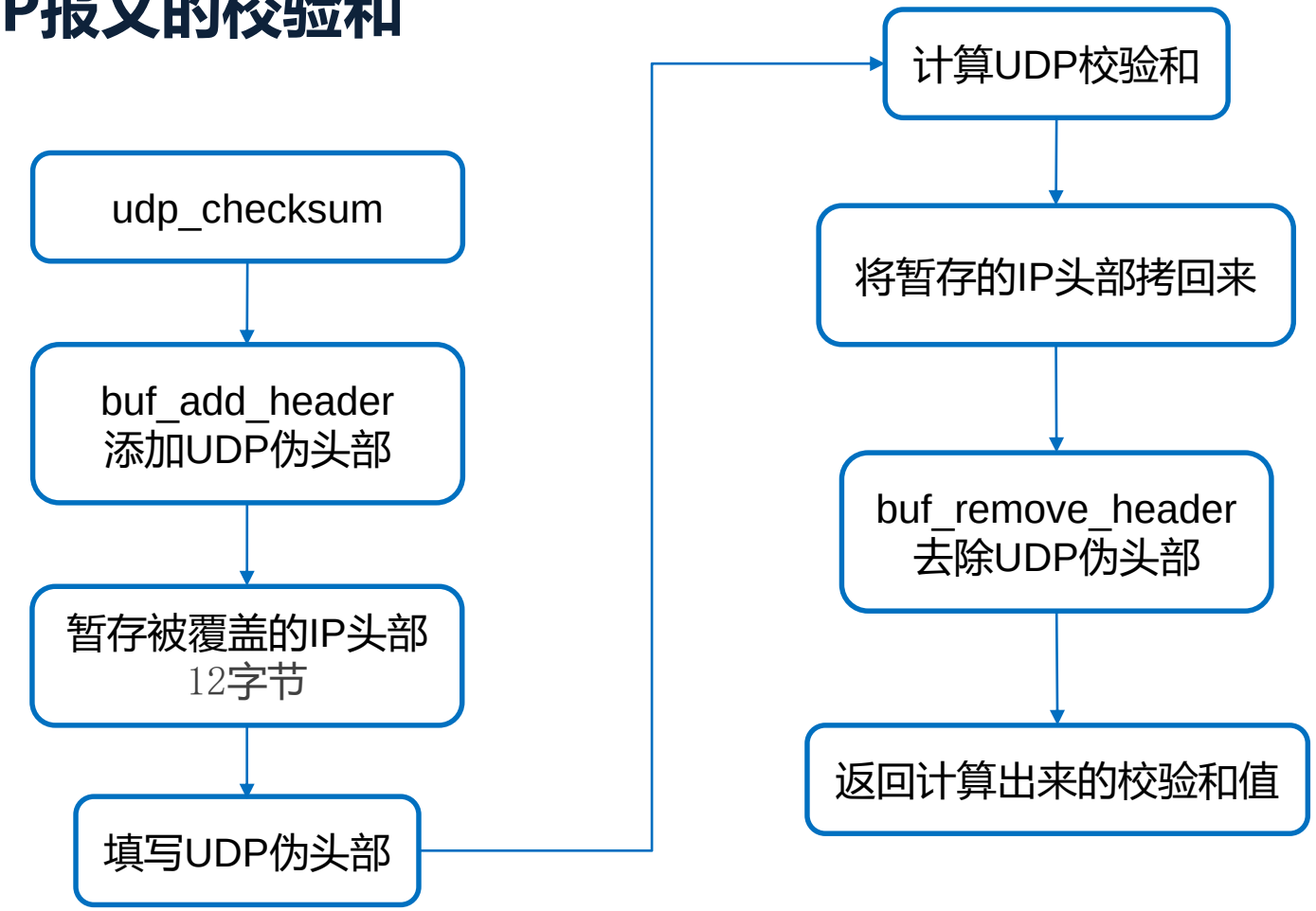


✓ UDP报文的接收和响应





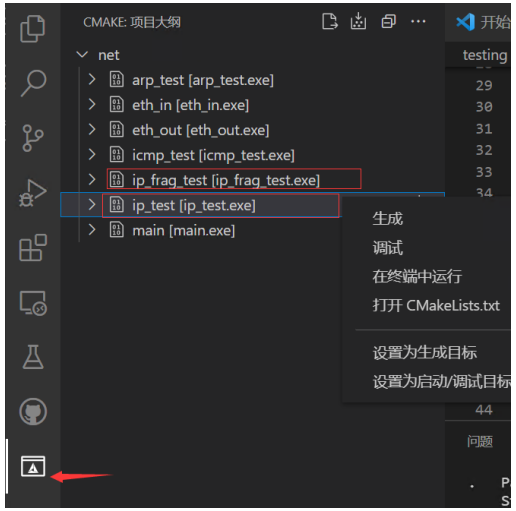
✓ UDP报文的校验和

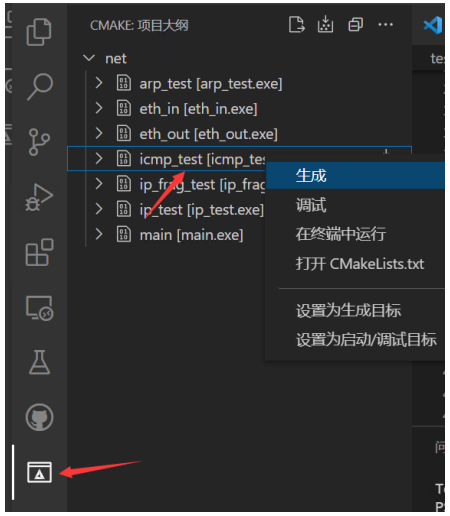

udp_checksum
buf_add_header
添加UDP伪头部
暂存被覆盖的IP头部
12字节
填写UDP伪头部
计算UDP校验和
将暂存的IP头部拷回来
buf_remove_header
去除UDP伪头部
返回计算出来的校验和值


只有敲代码才能
感受到温暖

实验自测

✓ IP/ICMP实验自测





测试通过

```
PS F:\lab\net_lab-master\net_lab-master\build> ctest -R ip_test
Test project F:/lab/net_lab-master/net_lab-master/build
    Start 4: ip_test
1/1 Test #4: ip_test ..........................   Passed    0.04 sec

100% tests passed, 0 tests failed out of 1

Total Test time (real) =   0.05 sec
PS F:\lab\net_lab-master\net_lab-master\build> ctest -R ip_frag_test
Test project F:/lab/net_lab-master/net_lab-master/build
    Start 5: ip_frag_test
1/1 Test #5: ip_frag_test .....................   Passed    0.02 sec

100% tests passed, 0 tests failed out of 1

Total Test time (real) =   0.03 sec
PS F:\lab\net_lab-master\net_lab-master\build>
```

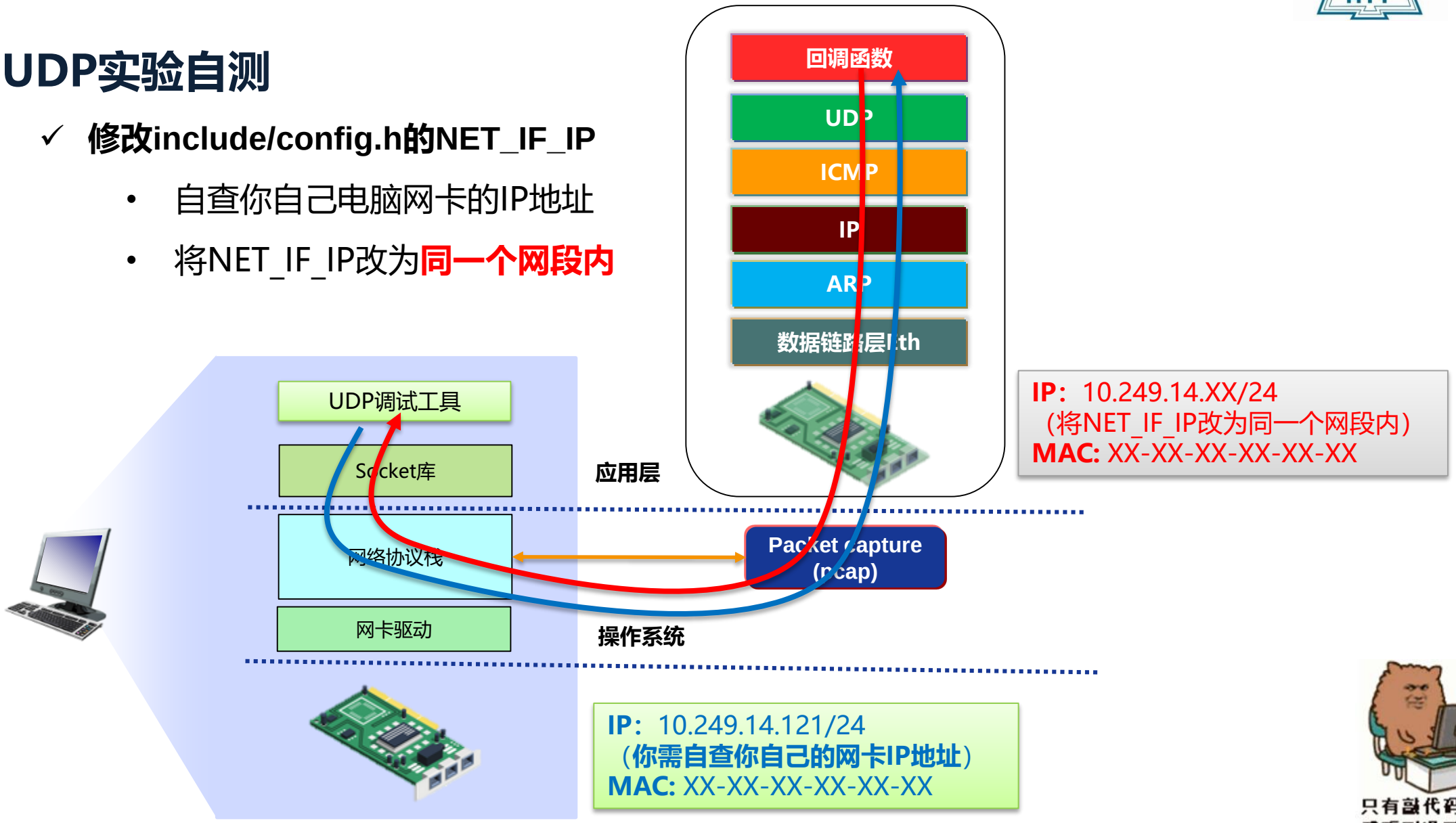
```
PS F:\lab\net_lab-master\net_lab-master\build> ctest -R icmp_test
Test project F:/lab/net_lab-master/net_lab-master/build
    Start 6: icmp_test
1/1 Test #6: icmp_test ........................   Passed    0.04 sec

100% tests passed, 0 tests failed out of 1

Total Test time (real) =   0.05 sec
PS F:\lab\net_lab-master\net_lab-master\build>
```

实验自测

✓ UDP实验自测

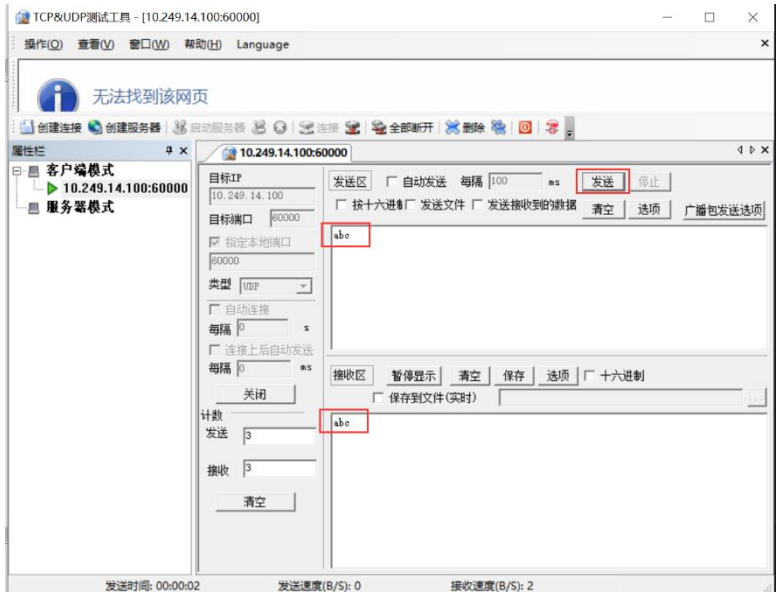
✓ 修改include/config.h的NET_IF_IP

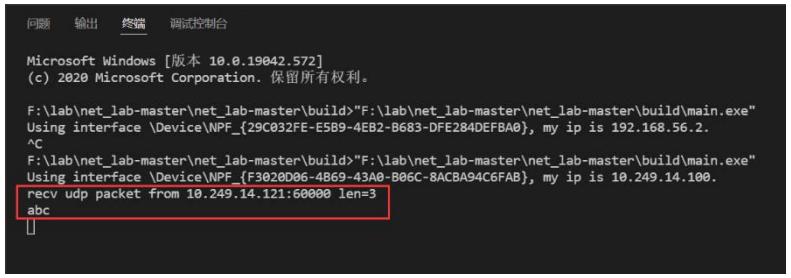
- 自查你自己电脑网卡的IP地址
- 将NET_IF_IP改为**同一个网段内**

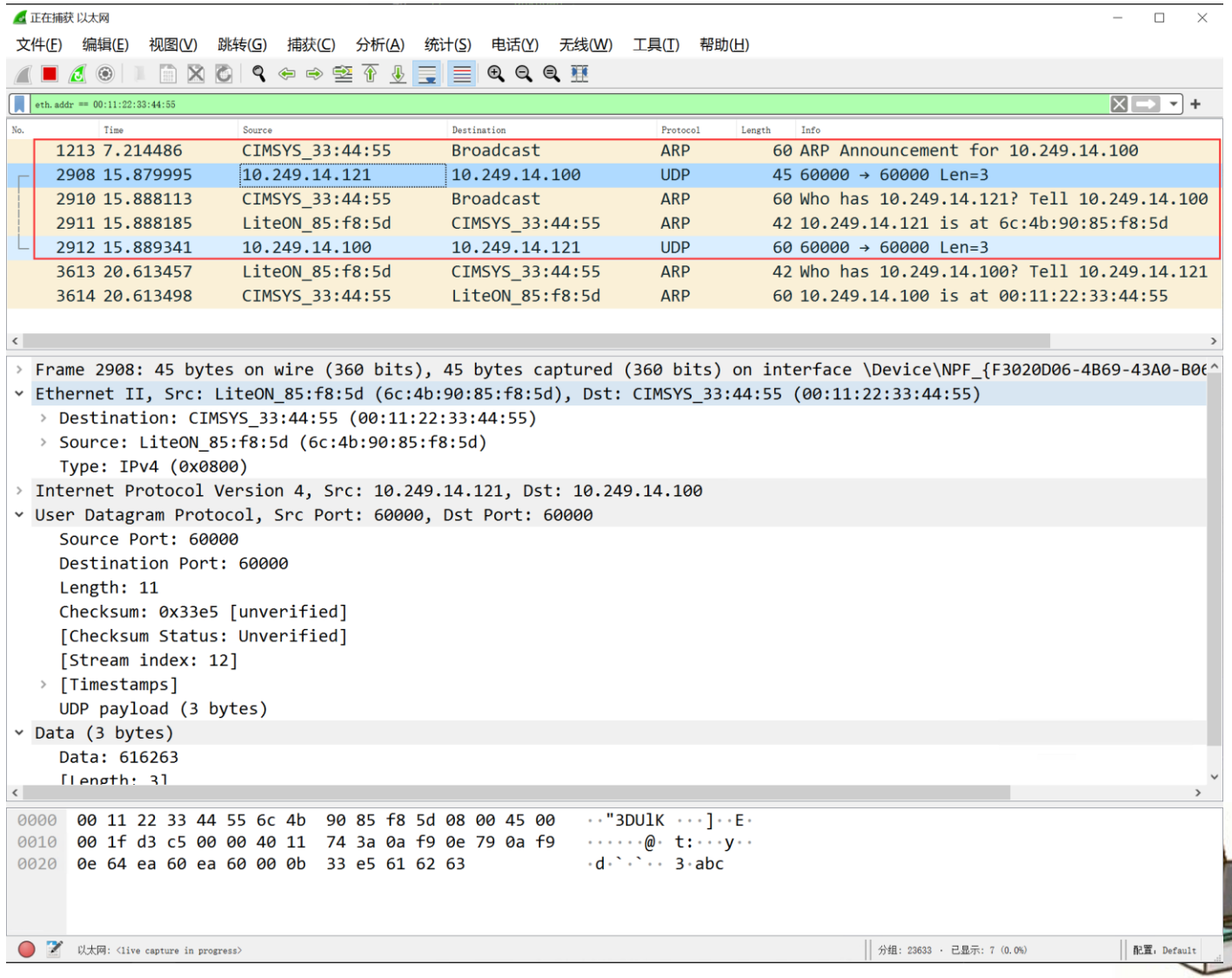


实验自测

✓ UDP自测结果







只有敲代码才能
感受到温暖

作业提交

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

同学们
请开始实验吧！