



哈尔滨工业大学  
Harbin Institute of Technology

# 计算机网络 课程实验报告

实验名称	Ipv4 分组收发与转发					
姓名	董琦		院系	数据科学与大数据		
班级	2003501		学号	120L020701		
任课教师	刘亚维		指导教师	刘亚维		
实验地点	G001		实验时间	2022/10/19		
实验课表现	出勤、表现得分(10)		实验报告 得分(40)		实验总分	
	操作结果得分(50)					
教师评语						

**实验目的：****IPv4分组收发实验目的：**

IPv4协议是互联网的核心协议，它保证了网络节点（包括网络设备和主机）在网络层能够按照标准协议互相通信。IPv4地址唯一标识了网络节点和网络的连接关系。在我们日常使用的计算机的主机协议栈中，IPv4协议必不可少，它能够接收网络中传送给本机的分组，同时也能根据上层协议的要求将报文封装为IPv4分组发送出去。

本实验通过设计实现主机协议栈中的IPv4协议，让学生深入了解网络层协议的基本原理，学习IPv4协议基本的分组接收和发送流程。

另外，通过本实验，学生可以初步接触互联网协议栈的结构和计算机网络实验系统，为后面进行更为深入复杂的实验奠定良好的基础。

**IPv4 分组转发实验目的：**

通过前面的实验，我们已经深入了解了IPv4协议的分组接收和发送处理流程。本实验需要将实验模块的角色定位从通信两端的主机转移到作为中间节点的路由器上，在IPv4分组收发处理的基础上，实现分组的路由转发功能。

网络层协议最为关注的是如何将IPv4分组从源主机通过网络送达目的主机，这个任务就是由路由器中的IPv4协议模块所承担。路由器根据自身所获得的路由信息，将收到的IPv4分组转发给正确的下一跳路由器。如此逐跳地对分组进行转发，直至该分组抵达目的主机。IPv4分组转发是路由器最为重要的功能。

本实验设计模拟实现路由器中的IPv4协议，可以在原有IPv4分组收发实验的基础上，增加IPv4分组的转发功能。对网络的观察视角由主机转移到路由器中，了解路由器是如何为分组选择路由，并逐跳地将分组发送到目的主机。本实验中也会初步接触路由表这一重要的数据结构，认识路由器是如何根据路由表对分组进行转发的。

**实验内容：****IPv4分组收发实验内容：****1) 实现IPv4分组的基本接收处理功能**

对于接收到的IPv4分组，检查目的地址是否为本地地址，并检查IPv4分组头部中其它字段的合法性。提交正确的分组给上层协议继续处理，丢弃错误的分组并说明错误类型。

**2) 实现IPv4分组的封装发送**

根据上层协议所提供的参数，封装IPv4分组，调用系统提供的发送接口函数将分组发送出去。

**IPv4分组转发实验功能：****1) 设计路由表数据结构。**

设计路由表所采用的数据结构。要求能够根据目的IPv4地址来确定分组处理行为（转发情况下需获得下一跳的IPv4地址）。路由表的数据结构和查找算法会极大的影响路由器的转发性能，有兴趣的同学可以深入思考和探索。

**2) IPv4分组的接收和发送。**

对前面实验（IP实验）中所完成的代码进行修改，在路由器协议栈的IPv4模块中能够正确完成分组的接收和发送处理。具体要求不做改变，参见“IP实验”。

**3) IPv4分组的转发。**

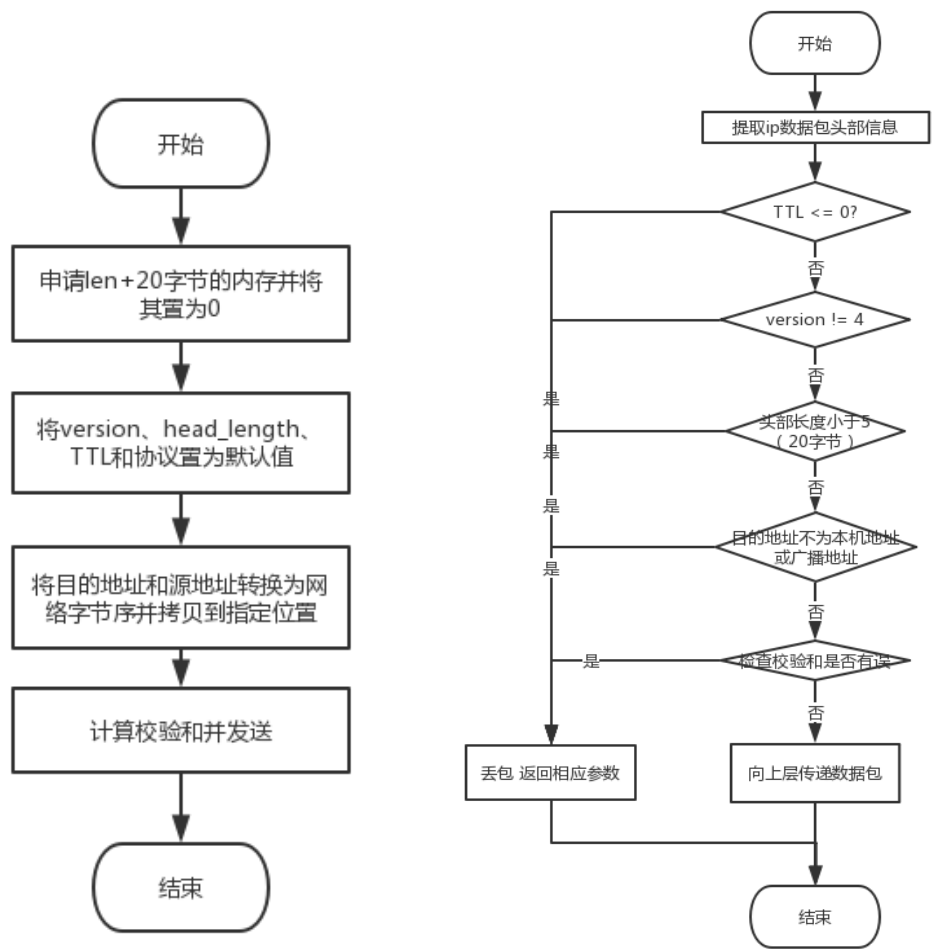
对于需要转发的分组进行处理，获得下一跳的IP地址，然后调用发送接口函数做进一步处理。

**实验过程：**

以文字描述、实验结果截图等形式阐述实验过程，必要时可附相应的代码截图或以附件形式提交。

(一) IP分组收发实验

1. 发送函数和接收函数的流程图



左图为发送函数；右图为接收函数。

2. 没有使用新建的数据结构。

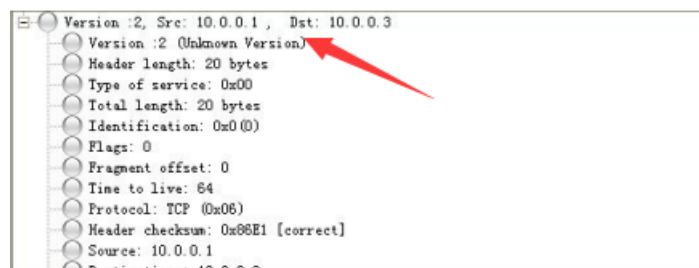
3. IPv4数据包头部信息如下图所示：



## 1) 版本号字段错误检验原理

可以看到版本号在第0字节的高4位，因此仅需利用位运算的原理，将第0字节的高4位提取出来，并且与0进行比较即可。具体实现如下：

```
if (version != 4)           //提取版本号判断是否为4
{
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_VERSION_ERROR);
    return 1;
}
```



## 2) 头部长度的（IP Head Length）字段检验原理

头部长度的信息是以4字节为单位，存储在IP数据包头部的第0字节的低4位，所以仅需利用位运算提取出低4位的信息；并且根据最少的IP数据包的头部信息为20字节，所以低于5的头部长度的字段是错误的，具体实现如下：

```
unsigned char headLen = pBuffer[0] & 0xF;
if (headLen < 5)       //提取头部长度的判断是否小于5
{
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_HEADLEN_ERROR);
    return 1;
}
```



## 3) 生存时间（Time to live, TTL）字段检验原理

生存时间字段在IP数据包的头部第8字节，因此仅仅需要将对应的字节提取出来，与0进行比较。如果TTL小于0说明其已经过期，丢弃即可，具体的实现如下：

```
unsigned char ttl = pBuffer[8];
if (ttl == 0)      //提取ttl,如果为0,出错
{
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_TTL_ERROR);
    return 1;
}
```

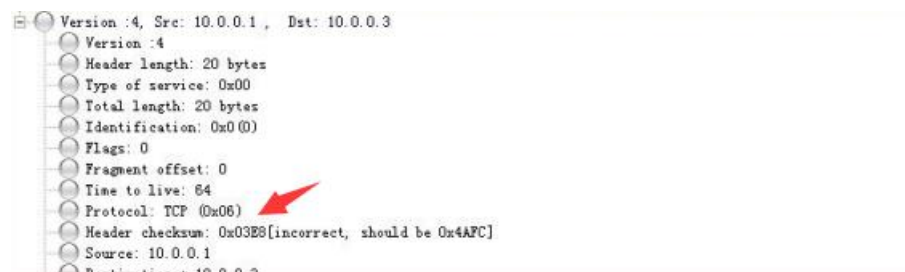


#### 4) 头部校验和（Header Checksum）字段检验原理

头部校验和字段在IP数据包的头部第10个字节，并且根据之前进行计算时取反的性质，将所有的头部信息进行与checksum生成时相同的计算步骤，得到的结果应该为全1；否则说明头部校验和错误。具体的实现如下：

```
unsigned short check_sum(char* buffer)
{
    unsigned int checkSum = 0;
    for (int i = 0; i < 10; i++)    // 头部数据每16b加起来
    {
        checkSum += ((unsigned short*)buffer)[i];
    }
    checkSum = (checkSum >> 16) + (checkSum & 0xFFFF); //进位翻卷
    checkSum = ~checkSum;    // 反码
    return checkSum;
}

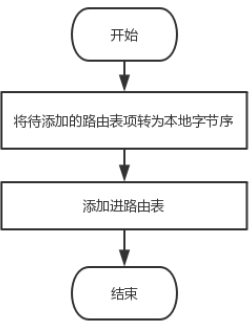
if (check_sum(pBuffer)) // 如果校验和计算函数的返回值不为0,出错
{
    ip_DiscardPkt(pBuffer, STUD_IP_TEST_CHECKSUM_ERROR);
    return 1;
}
```



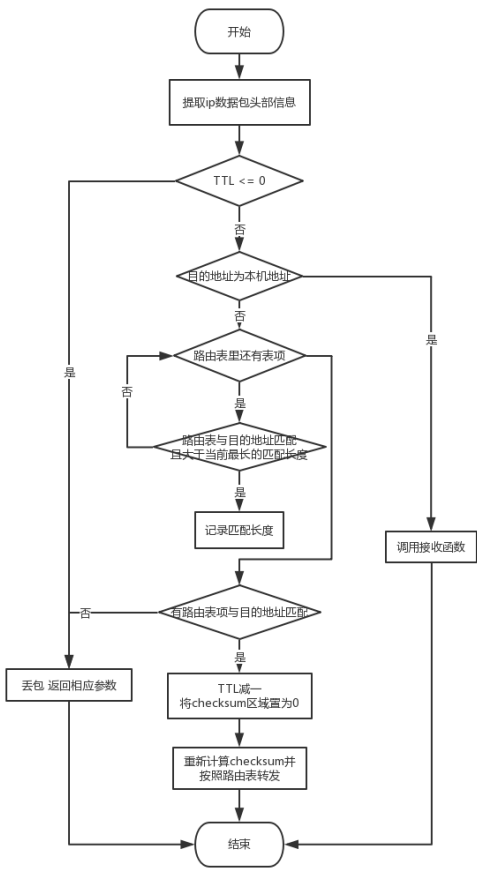
### （二）IP分组转发实验

1. 对于路由表初始化函数stud\_Route\_Init(), 使用全局变量作为路由表，所以初始化时没有再利用到该函数，所以没画对应的流程图。

下面是路由增加和路由转发的实现流程图



上图为路由增加stud\_route\_add()函数流程图。



上图为系统处理收到的IP分组的stud\_fwd\_deal()函数的流程图。

2. 新建的数据结构说明：路由表使用的数据结构为c++内置的STL中的vector。初始化路由表，就是新建一个vector链表；增加路由表项，就是在vector的尾部增加新的项即可；在vector进行查找，此处只能进行线性搜索，从vector的头部依次进行搜索。

实验结果：

采用演示截图、文字说明等方式，给出本次实验的实验结果。

(一) IP分组收发实验

1. version字段错误

```
00033 if (version != 4)
00034 {
00035     // version is not 4
00036     ip_DiscardPkt(pBuffer, STUD_IP_TEST_VERSION_ERROR);
00037     return 1;
00038 }
00039 if (head_length < 5)
00040 {
00041     // head length is not 20 bytes
00042     ip_DiscardPkt(pBuffer, STUD_IP_TEST_HEADLEN_ERROR);
00043     return 1;
00044 }
00045 if (destination != getIpv4Address() && destination != 0xffff)
00046 {
```

变量名	值
version	3
ttl	64
checksum	30433
head_length	5

可以看到在版本号错误的实验样例中，给出的IP数据包的version段为3，与IPv4的数据包的头部version为4不符。

2. 头部长度 (IP Head Length) 错误

```
00039 if (head_length < 5)
00040 {
00041     // head length is not 20 bytes
00042     ip_DiscardPkt(pBuffer, STUD_IP_TEST_HEADLEN_ERROR);
00043     return 1;
00044 }
00045 if (destination != getIpv4Address() && destination != 0xffff)
00046 {
00047     ip_DiscardPkt(pBuffer, STUD_IP_TEST_DESTINATION_ERROR);
00048     return 1;
00049 }
00050 unsigned long sum = 0;
00051 unsigned long temp = 0;
00052 int i;
```

变量名	值
version	4
ttl	64
checksum	26849
head_length	3

可以看到在实验测试中给出的样例为头部长度为3，这与最少的IP头部长度字段为5不符。

3. 生存时间 (TTL) 错误

```

00027 | if (ttl == 0)
00028 | {
00029 |     // TTL error
00030 |     ip_DiscardPkt(pBuffer, STUD_IP_TEST_TTL_ERROR);
00031 |     return 1;
00032 | }
00033 | if (version != 4)
00034 | {
00035 |     // version is not 4
00036 |     ip_DiscardPkt(pBuffer, STUD_IP_TEST_VERSION_ERROR);
00037 |     return 1;
00038 | }
00039 | if (head_length < 5)
00040 | {

```

变量名	值
version	4
ttl	0
checksum	-22815
head_length	5

可以看到在实验中给出的带有错误TTL数据段的数据包，在TTL字段的长度为0，意味着该数据包应该进行丢弃。

#### 4. 校验和（Header Checksum）字段错误

根据发送方checksum的计算公式可知，在接收方重新计算checksum的结果应该为全1（即16个1），在实验给出的有关校验和错误的样例中，如下

```

00060 | unsigned short l_word = sum & 0xffff;
00061 | unsigned short h_word = sum >> 16;
00062 | if (l_word + h_word != 0xffff)
00063 | {
00064 |     ip_DiscardPkt(pBuffer, STUD_IP_TEST_CHECKSUM_ERROR);
00065 |     return 1;
00066 | }
00067 |
00068 | ip_SendtoUp(pBuffer, length);
00069 | return 0;
00070 | }
00071 |
00072 | int stud_ip_Upsend(char *pBuffer, unsigned short len, unsigned
00073 |                   unsigned int dstAddr, byte protocol, byte t

```

变量名	值
head_length	5
l_word	39398
h_word	0



win7 [正在运行] - Oracle VM VirtualBox

管理 控制 视图 网络 设备 帮助

网络协议开发实验平台 - C:\Users\win732\Desktop\lab4.cpp

系统(S) 文件(F) 编辑(E) 视图(V) 调试(D) 协议编辑(P) 扩展协议实验(E) 帮助(H)

新建 打开 存盘 编译 执行 继续 停止 进入 跳进 跳出 断点 编辑 分析 组包 交互 帮助

编号	时间	源地址	目的地址	协议	数据包描述	实验描述
1	Tue Nov 09 14:46:20.376 2...	10.0.255.243	10.0.255.241	IP	Version 4, Src...	2.1 发送IP包
2	Tue Nov 09 14:46:21.814 2...	10.0.0.1	10.0.0.4	TCP	Bogus TCP h...	2.2 正确接收IP包
3	Tue Nov 09 14:46:23.798 2...	10.0.0.1	10.0.0.4	TCP	Bogus TCP h...	2.3 校验和错误的IP包
4	Tue Nov 09 14:46:25.814 2...	10.0.0.1	10.0.0.4	TCP	Bogus TCP h...	2.4 TTL错误的IP包
5	Tue Nov 09 14:46:27.798 2...	10.0.0.1	10.0.0.4	TCP	Bogus TCP h...	2.5 标志错误的IP包
6	Tue Nov 09 14:46:29.814 2...	10.0.0.1	10.0.0.4	TCP	TCP 16384 > ...	2.6 头部长度错误的IP包
7	Tue Nov 09 14:46:31.845 2...	10.0.0.1	192.168.88.51	TCP	Bogus TCP h...	2.7 错误目标地址的IP包

Ethernet II, Src: 00:00:04:00:00:0A, Dst: 00:00:01:00:00:0A  
Version 4, Src: 10.0.255.243, Dst: 10.0.255.241  
Data(17 bytes)

0000 00 0D 01 00 00 0A 00 0D 04 00 00 0A 00 00 45 00  
0010 00 25 55 F8 00 00 12 00 3F 04 0A 00 FF F3 0A 00  
0020 FF F1 31 71 61 7A 78 73 77 32 33 65 64 63 76 66  
0030 72 34 00

网络协议开发实验平台 - C:\Users\win732\Desktop\lab5.cpp

系统(S) 文件(F) 编辑(E) 视图(V) 调试(D) 协议编辑(P) 扩展协议实验(E) 帮助(H)

新建 打开 存盘 编译 执行 继续 停止 进入 跳进 跳出 断点 编辑 分析 组包 交互 帮助

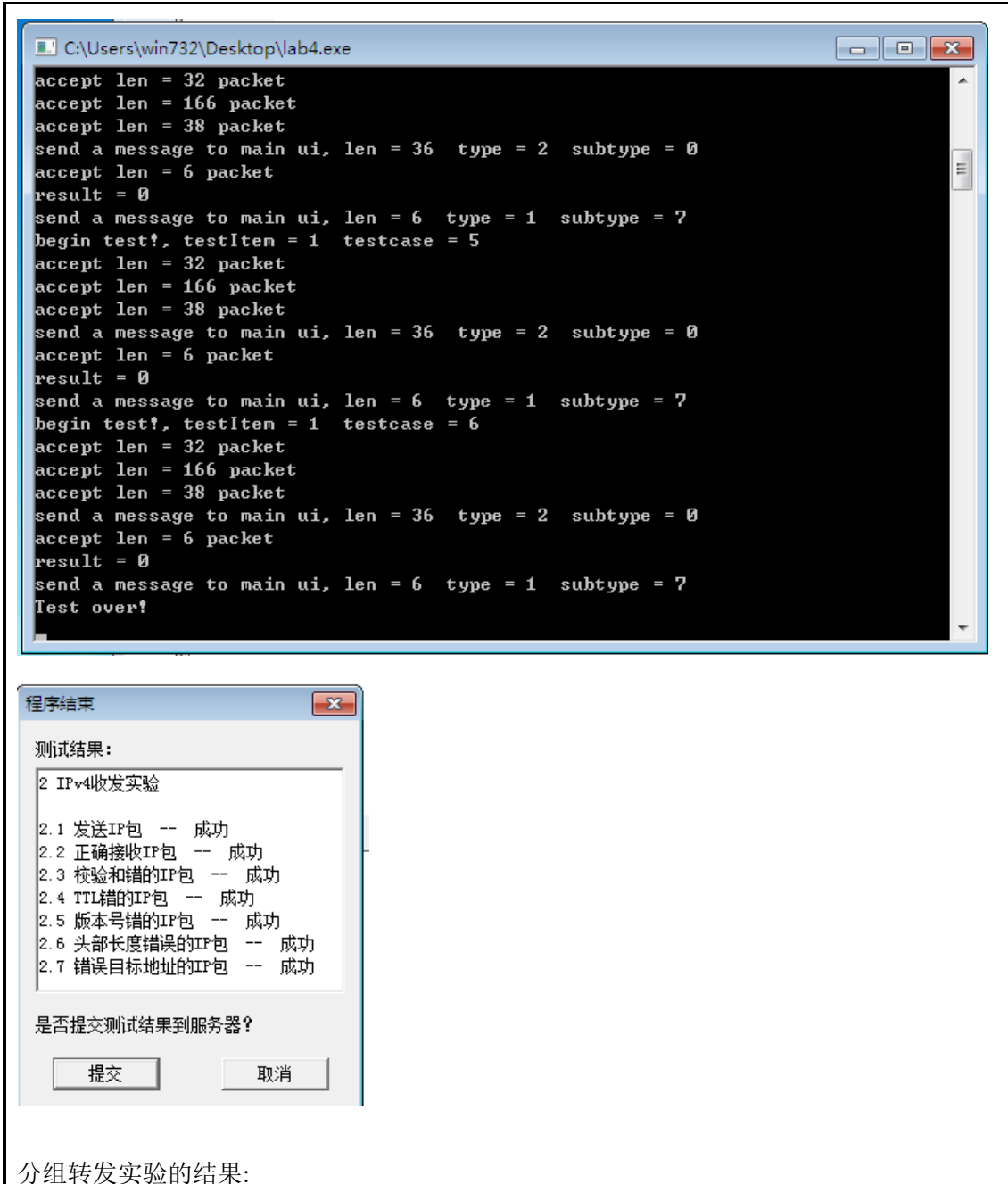
编号	时间	源地址	目的地址	协议	数据包描述	实验描述
1	Tue Nov 09 14:42:01.814 2...	10.0.0.1	10.0.0.3	TCP	Bogus TCP h...	3.1 本地接收实验
2	Tue Nov 09 14:42:09.814 2...	10.0.0.1	16.0.0.3	TCP	Bogus TCP h...	3.2 无法获得路由信息
3	Tue Nov 09 14:42:17.814 2...	10.0.0.1	11.0.0.3	TCP	Bogus TCP h...	3.3 正确转发实验
4	Tue Nov 09 14:42:17.814 2...	10.0.0.1	11.0.0.3	TCP	Bogus TCP h...	3.3 正确转发实验

Ethernet II, Src: 00:0D:01:00:00:0A, Dst: 00:0D:03:00:00:0A  
Destination : 00:0D:03:00:00:0A  
Source : 00:0D:01:00:00:0A  
TYPE : IP (0x0800)  
Version :4, Src: 10.0.0.1, Dst: 10.0.0.3  
Data(17 bytes)(invalid TCP header)

0000 00 0D 03 00 00 0A 00 0D 01 00 00 0A 08 00 45 00  
0010 00 25 00 00 00 00 40 06 66 E1 0A 00 00 01 0A 00  
0020 00 03 31 71 61 7A 78 73 77 32 33 65 64 63 76 66  
0030 72 34 00

可以看到，重新计算的结果为0x99E6，这与16位全1不符，所以发生了校验和错误。

分组收发实验的结果:



The image shows a Windows command prompt window titled "C:\Users\win732\Desktop\lab4.exe" displaying the output of a network test. The output consists of several lines of text indicating packet reception and message sending. Below the command prompt is a "程序结束" (Program End) dialog box with a "测试结果:" (Test Results) section listing seven test cases, all marked as successful. At the bottom of the dialog box, there is a question "是否提交测试结果到服务器?" (Whether to submit test results to the server?) and two buttons: "提交" (Submit) and "取消" (Cancel).

```

accept len = 32 packet
accept len = 166 packet
accept len = 38 packet
send a message to main ui, len = 36 type = 2 subtype = 0
accept len = 6 packet
result = 0
send a message to main ui, len = 6 type = 1 subtype = 7
begin test!, testItem = 1 testcase = 5
accept len = 32 packet
accept len = 166 packet
accept len = 38 packet
send a message to main ui, len = 36 type = 2 subtype = 0
accept len = 6 packet
result = 0
send a message to main ui, len = 6 type = 1 subtype = 7
begin test!, testItem = 1 testcase = 6
accept len = 32 packet
accept len = 166 packet
accept len = 38 packet
send a message to main ui, len = 36 type = 2 subtype = 0
accept len = 6 packet
result = 0
send a message to main ui, len = 6 type = 1 subtype = 7
Test over!
    
```

程序结束

测试结果:

- 2 IPv4收发实验
- 2.1 发送IP包 -- 成功
- 2.2 正确接收IP包 -- 成功
- 2.3 校验和错的IP包 -- 成功
- 2.4 TTL错的IP包 -- 成功
- 2.5 版本号错的IP包 -- 成功
- 2.6 头部长度错误的IP包 -- 成功
- 2.7 错误目标地址的IP包 -- 成功

是否提交测试结果到服务器?

提交 取消

分组转发实验的结果:



