

## 实验 4：IPv4 分组收发实验

### 1. 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。

### 2. 实验环境

- 接入到 Netriver 网络实验系统服务器的主机
- Windows XP、Windows 7/8/10 或苹果 OS
- 开发语言：C 语言

### 3. 实验要求

根据计算机网络实验系统所提供的上下层接口函数和协议中分组收发的主要流程，独立设计实现一个简单的 IPv4 分组收发模块。要求实现的主要功能包括：

- 1) IPv4 分组的基本接收处理，能够检测出接收到的 IP 分组是否存在如下错误：校验和错、TTL 错、版本号错、头部长度的错、错误目标地址；
- 2) IPv4 分组的封装发送；

注：不要求实现 IPv4 协议中的选项和分片处理功能

## 4. 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。

## 5. 参考内容

在主机协议栈中,IPv4 协议主要承担辨别和标识源 IPv4 地址和目的 IPv4 地址的功能,一方面接收处理发送给自己的分组,另一方面根据应用需求填写目的地址并将上层报文封装发送。IPv4 地址可以在网络中唯一标识一台主机,因而在相互通信时填写在 IPv4 分组头部中的 IPv4 地址就起到了标识源主机和目的主机的作用。在后面 IPv4 分组的转发实验中,我们还将深入学习 IPv4 地址在分组转发过程中对选择转发路径的重要作用。

在两个主机端系统通信的环境中,网络的拓扑可以简化为两台主机直接相连,中间的具体连接方式可以抽象为一条简单的链路,如图 4-1 所示。IPv4 分组收发实验就是要在实验系统客户端的开发平台上,实现 IPv4 分组的接收和发送功能。

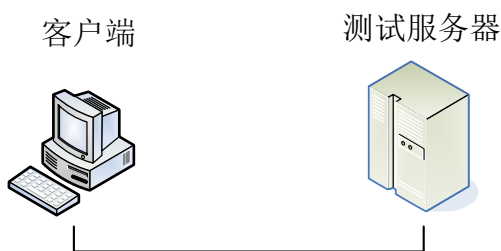


图 4-1 实验环境网络拓扑结构

## 5.1 NetRiver 网络实验系统简介

整个 NetRiver 网络实验系统由 3 部分构成：实验管理服务器、测试服务器和客户端，三者之间可以互相通信，系统的结构如图 4-2 所示。

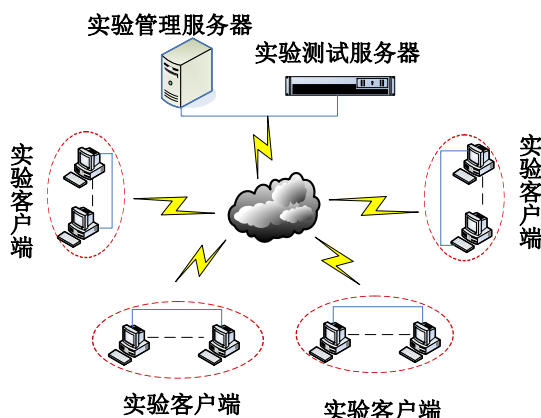


图 4-2 NetRiver 系统网络实验环境

实验管理服务器用于保存所有用户信息，包括用户名、用户密码、实验进度、测试结果等等。实验管理服务器还存储了网络实验的所有测试用例。支持用户用户管理和认证功能，用户登录成功后，服务器会向客户端发送当前测试用例列表，并管理用户提交的实验代码和测试结果。该服务器还能接受到测试服务器的连接请求及向测试服务器发送测试用例内容。

测试服务器会根据实验内容与客户端进行协议通信，完成协议一致性测试。测试服务器从实验管理服务器上获取测试用例内容，测试用例描述了详细的测试过程，测试服务器和客户端按照测试用例完成这一过程。测试服务器向客户端发送协议分组，也接收客户端发来的协议分组，并判断客户端协议行为的正确性。所有的测试结果都是有测试服务器进行判定，再反馈给客户端，并通知实验管理服务器对其进行保存。

客户端是学生实验的操作平台，它不仅可以连接并登录实验管理服务器和测试服务器，还提供了一个可以编辑、编译、调式和执行实验代码的软件开发环境。在测试过程中，客户端软件还能够显示并分析协议测试分组的内容。

## 5.2 NetRiver 实验系统学生实验流程

整个实验流程图如图 4-3 所示。

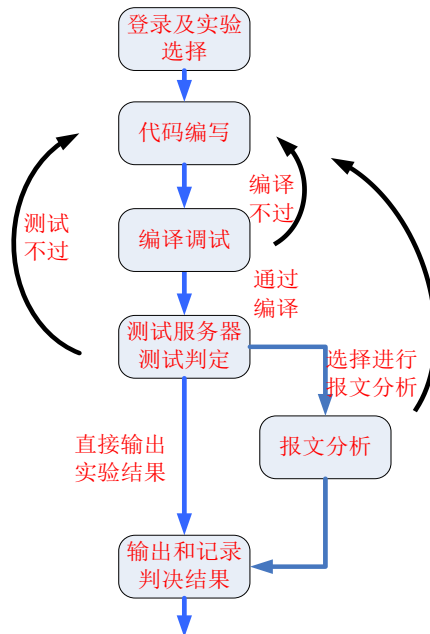


图 4-3 NetRiver 系统实验整体流程

### 5.2.1 登录和选择实验

安装好客户端软件后，学生需要登录系统并选择改次试验的内容和具体的测试用例，如图 4-4 所示。

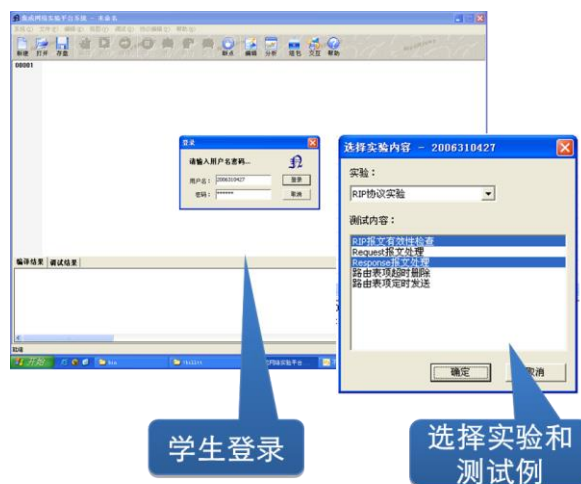


图 4-4 学生登录、选择实验和测试用例

### 5.2.2 实验代码编辑、编译和运行

如果第一次做某个实验，在“文件”菜单中选择“新建”，系统会自动创建该实验的模板代码并显示在编辑器当中。该模板将给出实验必须的头文件引用、系统函数接口声明以及需要学生完成的函数接口。学生只需在此基础上定义自己需要的数据结构，完成需要实现的函数内容即可。学生可以随时保存或另存代码文件，系统提供了代码的上传和下载服务，可以把代码文件上传到服务器上或者下载。注意，服务器只会为每个用户的每个实验保留一份代码，所以每次上传都会覆盖上次上传的文件。用户在编写完实验代码后可以编译。只有代码被正确编译后，用户才能执行相关的测试操作。代码的编辑和编译示意图如图 4-5 所示。

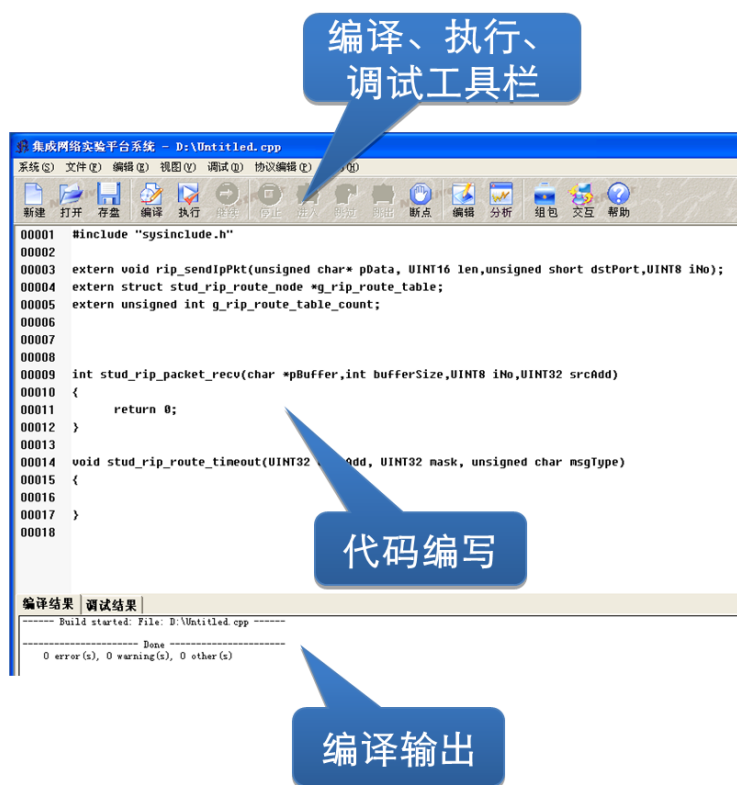


图 4-5 实验代码编辑和编译示意图

代码编译成功后，学生可以运行程序，执行测试用例或者对代码进行调试。NetRiver 提供了一整套代码编写和调试的环境，包括断点设置、单步执行和查看运行时变量等。

在执行或调试程序的时候，学生可以在报文分析界面查看客户端和服务器发出的报文，如图 4-6 所示。

系统会在执行过程当中显示执行结果和测试结果，如图 4-7 所示。

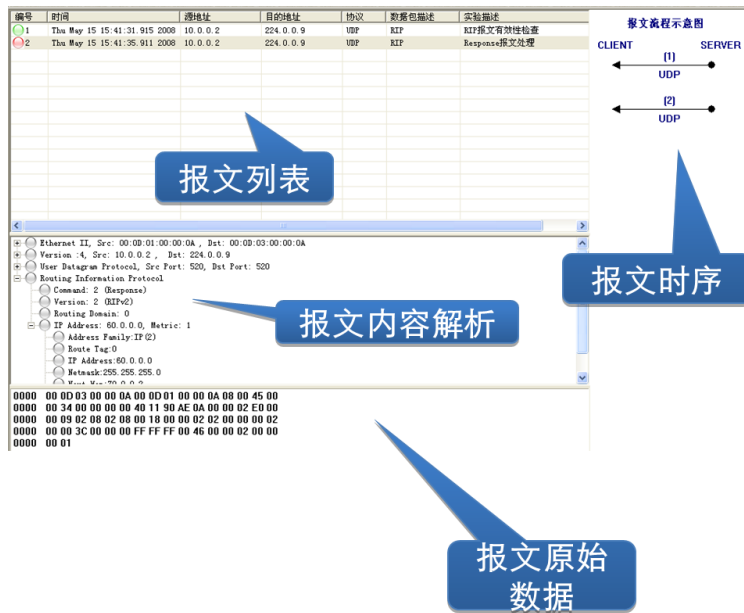


图 4-6 报文分析器界面

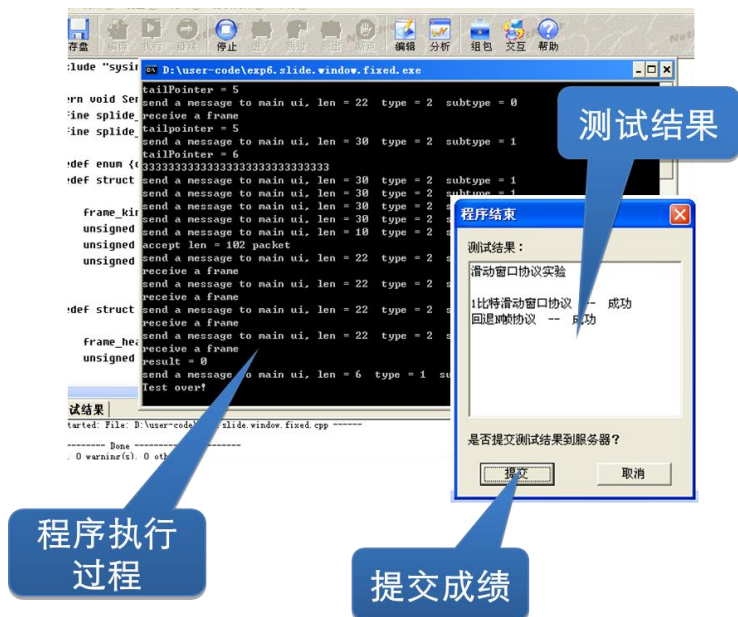


图 4-7 程序执行过程和结果

### 5.2.3 输出和记录判决结果

在学生完成实验后，可以进入在线实验管理系统查看系统自动给出的判决结果，该结果是由系统的测试服务器与学生程序交互后，按照测试用例自动给出，如图 4-8 所示。

ID	学号	学生姓名	系	课程	实验名称	日期	完成情况
12	2005011342	李德	计算机科学与技术系	计55	滑动窗口协议实验	2008-04-16	0
13	2005080033	金圣	计算机科学与技术系	计55	滑动窗口协议实验	2008-04-25	0
14	2005011347	陈帅	计算机科学与技术系	计55	KIP协议实验	2008-04-09	0
15	2005011349	谢耀	计算机科学与技术系	计55	KIP协议实验	2008-04-18	0
16	2005011294	郭旭	计算机科学与技术系	计53	KIP协议实验	2008-05-08	0
17	2005010709	楼宇	计算机科学与技术系	计53	滑动窗口协议实验	2008-04-17	0
18	2005011296	罗森	计算机科学与技术系	计50	KIP协议实验	2008-05-15	0
19	2006011306	朱晨	计算机科学与技术系	计60	滑动窗口协议实验	2008-04-10	0
20	2005011300	徐博	计算机科学与技术系	计53	KIP协议实验	2008-04-10	0
21	2005011283	汪汀	计算机科学与技术系	计50	滑动窗口协议实验	2008-04-17	0
22	2005011281	黄刚	计算机科学与技术系	计53	滑动窗口协议实验	2008-04-24	0
23	2005011303	赵星	计算机科学与技术系	计53	KIP协议实验	2008-05-20	0
24	2005011298	特力	计算机科学与技术系	计53	滑动窗口协议实验	2008-04-24	0

图 4-8 输出和记录判决结果

## 5.3 处理流程

客户端接收到测试服务器发送来的 IPv4 分组后，调用接收接口函数 `stud_ip_recv()`（图 4-9 中接口函数 1）。学生需要在这个函数中实现 IPv4 分组接收处理的功能。接收处理完成后，调用接口函数 `ip_SendtoUp()` 将需要上层协议进一步处理的信息提交给上层协议（图 4-9 中接口函数 2）；或者调用函数 `ip_DiscardPkt()` 丢弃有错误的分组并报告错误类型（图 4-9 中函数 5）。

在上层协议需要发送分组时，会调用发送接口函数 `stud_ip_Upsend()`（图 4-9 中接口函数 3）。学生需要在这个函数中实现 IPv4 分组封装发送的功能。根据所传参数完成 IPv4 分组的封装，之后调用接口函数 `ip_SendtoLower()` 把分组交给下层完成发送（图 4-9 中接口函数 4）。



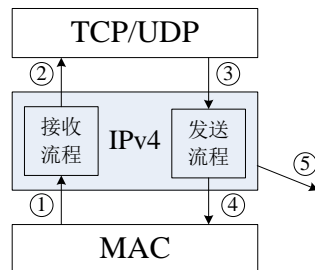


图 4-9 实验接口函数示意图

### 5.3.1 接收流程

在接口函数 `stud_ip_recv()` 中，需要完成下列处理步骤（仅供参考）：

- ① 检查接收到的 IPv4 分组头部的字段，包括版本号（Version）、头部长（IP Head length）、生存时间（Time to live）以及头校验和（Header checksum）字段。对于出错的分组调用 `ip_DiscardPkt()` 丢弃，并说明错误类型。
- ② 检查 IPv4 分组是否应该由本机接收。如果分组的地址是本机地址或广播地址，则说明此分组是发送给本机的；否则调用 `ip_DiscardPkt()` 丢弃，并说明错误类型。
- ③ 如果 IPV4 分组应该由本机接收，则提取得到上层协议类型，调用 `ip_SendtoUp()` 接口函数，交给系统进行后续接收处理。

### 5.3.2 发送流程

在接口函数 `stud_ip_Upsend()` 中，需要完成下列处理步骤（仅供参考）：

- ① 根据所传参数（如数据大小），来确定分配的存储空间的大小并申请分组的存储空间。
- ② 按照 IPv4 协议标准填写 IPv4 分组头部各字段，标识符（Identification）字段可以使用一个随机数来填写。（注意：部分字段内容需要转换成网络字节序）
- ③ 完成 IPv4 分组的封装后，调用 `ip_SendtoLower()` 接口函数完成后续的发送处理工作，最终将分组发送到网络中。



## 5.4 IPv4 分组头部格式

IPv4 分组头部格式如图 4-10 所示。

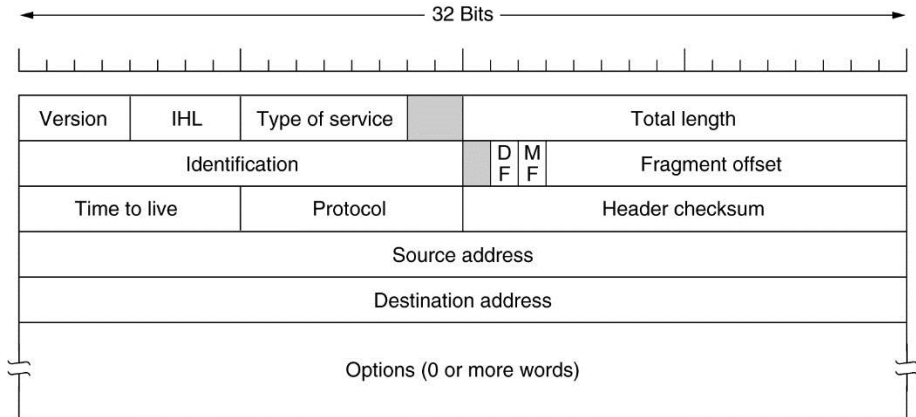


图 4-10 IPv4 分组头部格式

## 5.5 接口函数说明

### 5.5.1 需要实现的接口函数

#### 1) 接收接口

```
int stud_ip_recv(char * pBuffer, unsigned short length)
```

参数:

pBuffer: 指向接收缓冲区的指针, 指向 IPv4 分组头部

length: IPv4 分组长度

返回值:

0: 成功接收 IP 分组并交给上层处理

1: IP 分组接收失败

#### 2) 发送接口

```
int stud_ip_Upsend(char* pBuffer, unsigned short len, unsigned int srcAddr, unsigned int dstAddr, byte protocol, byte ttl)
```

参数:

pBuffer: 指向发送缓冲区的指针, 指向 IPv4 上层协议数据头部

len: IPv4 上层协议数据长度

srcAddr: 源 IPv4 地址

dstAddr: 目的 IPv4 地址

protocol: IPv4 上层协议号

ttl: 生存时间 (Time To Live)

返回值:

- 0: 成功发送 IP 分组
- 1: 发送 IP 分组失败

### 5.5.2 系统提供的接口函数

#### 1) 丢弃分组

void ip\_DiscardPkt(char \* pBuffer ,int type)

参数:

pBuffer: 指向被丢弃分组的指针

type: 分组被丢弃的原因, 可取以下值:

STUD_IP_TEST_CHECKSUM_ERROR	//IP 校验和出错
STUD_IP_TEST_TTL_ERROR	//TTL 值出错
STUD_IP_TEST_VERSION_ERROR	//IP 版本号错
STUD_IP_TEST_HEADLEN_ERROR	//头部长度的错
STUD_IP_TEST_DESTINATION_ERROR	//目的地址错

#### 2) 发送分组

void ip\_SendtoLower(char \*pBuffer ,int length)

参数:

pBuffer: 指向待发送的 IPv4 分组头部的指针

length: 待发送的 IPv4 分组长度

#### 3) 上层接收

void ip\_SendtoUp(char \*pBuffer, int length)

参数:

pBuffer: 指向要上交的上层协议报文头部的指针

length: 上交报文长度

#### 4) 获取本机 IPv4 地址

unsigned int getIpv4Address( )

参数: 无

## 6. 实验报告

- 1) 要求给出发送和接收函数的实现程序流程图;
- 2) 给出自己所新建的数据结构的说明 (如果有);
- 3) 要求给出版本号 (Version)、头部的长度 (IP Head length)、生存时间 (Time to live) 以及头校验和 (Header checksum) 字段的错误

检测原理，并根据实验具体情况给出错误的具体数据，例如如果为头部长度错，请给出收到的错误的 IP 分组头部长度字段值为多少。