

# 计算机网络原理实验报告

学院 计算机学院 专业 计算机科学与技术 班级 10012006

学号 2020303245 姓名 夏卓 实验时间: 2022/11/12

## 一、实验名称:

基于 UDP 协议通信编程

## 二、实验目的:

掌握 UDP 协议通信流程, 以及 UDP 协议通信应用系统设计与实现原理; 掌握 UDP 协议通信时序关系, 并分析采用这种时序关系的原因; 掌握利用 UDP 协议实现两台计算机之间的单向通信、双向通信和多媒体文件传输。

## 三、实验环境:

Win10, Intelx86

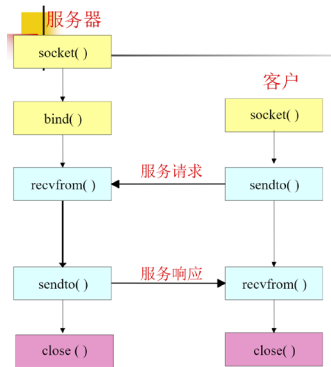
## 四、实验内容及步骤:

### 实验内容:

1. 单向通信: 客户端将从键盘输入任意字符串发送给服务器, 服务器接收到后将该字符串在屏幕上打印, 并将客户端 IP 地址和 PORT 打印;
2. 双向通信: (1) 客户端将从键盘输入任意字符串发送给服务器, 服务器接收到后将该字符串在屏幕上打印, 并将客户端 IP 地址和 PORT 打印; (2) 服务器将客户端发送来的字符串发送给客户端, 客户端打印字符串和服务器 IP 地址和 PORT;
3. 传输多媒体文件: 客户端将一个视频文件发送给服务器, 在服务器上能播放; 在发送过程中将“网络接口-停止”(3 到 6 秒), 检测可否正常传输文件。

### 实验步骤:

(1) 理解 UDP 协议的通信流程; 掌握 Socket 编程接口的使用方法



函数名	协议	使用者
SOCKET socket( af, type, protocol );	UDP	S/C
int bind( SOCKET s, const struct sockaddr FAR * seradd, int seraddlen )	UDP	S
int sendto (sockid, buf, buflen, flags, destadd, addrlen)	UDP	S/C
int recvfrom (sockid, buf, buflen, flags, suradd, suraddrlen)	UDP	S/C
int closesocket( SOCKET s )	UDP	S/C

- (2) 编写程序，利用 UDP 协议实现客户端到服务器单向和双向数据的传输和接收。

程序参考书上 P180-196，需要注意根据实际情况，更改客户端的目的 IP 地址及端口号与服务器的实际 IP 地址与端口号一致，可以在命令行使用 ipconfig 命令进行查看。另外最好打开防火墙，以免通信信息被防火墙屏蔽，使用无线热点进行连接，效果更好。

如下图所示，需要更改的 IP 地址即是局域网 IPv4 的地址：

```
无线局域网适配器 WLAN:

    连接特定的 DNS 后缀 . . . . . : 
    IPv6 地址 . . . . . : fd31:1927:f25a:5:959e:64ee:171d:c633
    临时 IPv6 地址. . . . . : fd31:1927:f25a:5:94:9734:e92:14b5
    本地链接 IPv6 地址. . . . . : fe80::105a:cd46:3b91:4c50%15
    IPv4 地址 . . . . . : 10.30.217.218
    子网掩码 . . . . . : 255.255.0.0
    默认网关. . . . . : 10.30.0.1
```

- (3) 编写程序，利用 UDP 协议实现客户端到服务器单向文件 (\*.avi) 的传输；在文件传输过程中，将网络分别中断 1s、5s、10s 等时间，检测是否可以正常传输文件。

由于待传输文件较小，通信时间很短，为了能够实现中途断网的目的，我们在文件传输过程中使用了 sleep 函数让客户端休眠了 2s，以便让服务器有足够的时间断网后重连：

```
if (flag)
{
    printf("start to sleep\n");
    Sleep(2000);
    printf("sleep end\n");
    flag = false;
}
```

## 五、实验结果：

### 1. 单向通信

客户端：

```
(base) PS D:\大学\大三上\计网\实验1\客户端> .\UDP_single_client
这里是客户端
Hello,server!
```

服务器：

选择 D:\project1\Project1\Debug\Project1.exe

```
这里是客户端--Server received from Client ip:[192.168.26.124],port:[57354]
Hello,server!--Server received from Client ip:[192.168.26.124],port:[57354]
```

### 2. 双向通信

客户端：

```
(base) PS D:\大学\大三上\计网\实验1\客户端> .\UDP_double_client
Client start to send data:
Hello!
client receive data from server:Hello!
server ip:[192.168.26.31],port:[6666]
11111
client receive data from server:11111
server ip:[192.168.26.31],port:[6666]
```

服务器:

```
D:\project1\Project1\Debug\Project1.exe
server start to receive data:
Server received from Client Hello!
Client ip:[192.168.26.124],client port:[50921]
Server received from Client 11111
Client ip:[192.168.26.124],client port:[50921]
```

### 3. 传输多媒体文件

客户端:

```
(base) PS D:\大学\大三上\计网\实验1\客户端> .\UDP_fileTrans_client
client start to send file!
file send end successfully!
```

服务器:

```
Microsoft Visual Studio 调试控制台
server start to recieve file
server recieve data end successssfully

D:\project1\Project1\Debug\Project1.exe (进程 11644) 已退出，代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口。 . . .
```

中途断网:

```
(base) PS D:\大学\大三上\计网\实验1\客户端> .\UDP_fileTrans_client
client start to send file!
start to sleep
sleep end
file send end successfully!
```

实验结果为客户端并不会关心服务器是否收到数据，即使中途断网时间很短，客户端也会直接发送完所有数据，导致服务器无法收到完整数据。

## 六、 实验总结

### UDP C/S 模型通信问题

#### 1. UDP 通信为什么必须是客户端首先向服务器发送数据？

因为服务器并不知道客户端的 IP 地址和端口号，无法向客户端发送数据。只有当客户端首先向服务器主动发出服务请求，服务器接收到请求后，才能得到客户端的 IP 地址及端口号，组成完整的通信五元组，此后服务器才可以向客户进程发送应答。

#### 2. 客户端何时知道通信五元组？

由于必须是客户端首先向服务器发送数据，因此要求客户端一开始就知道通信五元组的信息，并根据通信五元组向服务器主动发出服务请求。

#### 3. 服务器何时知道通信五元组？

只有服务器进程收到客户进程服务请求之后，服务器才知道通信五元组

### TCP 与 UDP 编程区别

基于连接与无连接:

TCP 存在建立连接和释放连接三次握手,而 UDP 直接发送数据,无连接过程;

TCP 保证数据传输可靠性,UDP 不保证。

TCP 采用字节流模式,而 UDP 采用数据报模式。

对系统资源的要求:

TCP 较多,UDP 少;

TCP 程序结构较复杂,UDP 程序结构较简单。

系统函数调用不同:

socket()的参数不同;

UDP Server 不需要调用 listen 和 accept;

TCP 收发数据不必指定对方地址,使用 send/recv 函数;

UDP 每次收发数据均需要指定对方地址,使用 sendto/recvfrom 函数;

TCP 客户进程标识信息在 accept 时服务器才能确定;

UDP 客户进程表示信息在服务器接收到数据的同时就能确定。

在传输多媒体文件的过程中,若中途断网,则基于 UDP 的通信过程会丢失数据,而当断网时间不长时,基于 TCP 的通信过程不会丢失数据。

**教师评语:**

**成绩:** \_\_\_\_\_ **教师签名:** \_\_\_\_\_ **批阅日期:** \_\_\_\_\_