

# 高级网络编程

## 实验报告

实验名称: UDP C/S结构程序的开发、uping程序解析

实验日期: 2020年5月13日

学生姓名: 黄文政

学 号: 71Y17111

# 一、实验目的

1. 掌握UDP C/S结构程序的开发
2. 弄懂ping指令的原理，编写一个ping程序

# 二、实验环境

ubuntu 18.04 LTS

# 三、实验内容

## UDP C/S结构程序的开发

### 1.设计思路

本次实验以回射程序为例进行UDP C/S结构的程序开发。（代码参照课件）

对于客户端，首先构造sockaddr\_in，设置好协议族、服务器的IP地址、端口等信息；其次创建一个套接口，并将套接口类型指定为SOCK\_DGRAM，表示使用UDP；然后在接收到终端的输入内容后，使用SendTo函数将内容发送至服务器；最后使用Recvfrom接收服务器的回射，并打印内容。

```

int main(int argc, char**argv){
    int sockfd;
    struct sockaddr_in servaddr;
    if(argc!=2){
        err_quit("usage:udp_echo_c <ipaddress>");
    }

    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family=AF_INET;
    servaddr.sin_port=htons(SERV_PORT);
    inet_pton(AF_INET, argv[1], &servaddr.sin_addr);
    sockfd=Socket(AF_INET, SOCK_DGRAM, 0);

    dg_cli(stdin, sockfd, (SA*)&servaddr, sizeof(servaddr));
    exit(0);
}

```

图 1: UDP客户端部分代码

```

vincent@vincent-ubuntu18:~/NetworkProgramming/3$ sudo ./udp_echo_c 192.168.0.115
hello
hello

```

图 2: UDP客户端运行结果

对于服务端，同样构造sockaddr\_in，设置好协议族、端口等信息；创建一个使用UDP通信的套接口；其次使用 bind()函数将套接口和sockaddr\_in进行绑定，让该套接口对指定端口进行监听；接着使用Recvfrom()函数接收客户端发来内容；最后将收到的内容通过SendTo()函数进行反射。

```

int main(int argc, char**argv){
    int sockfd;
    struct sockaddr_in servaddr, cliaddr;
    sockfd=Socket(AF_INET, SOCK_DGRAM, 0);

    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family=AF_INET;
    servaddr.sin_addr.s_addr=htonl(INADDR_ANY);
    servaddr.sin_port=htons(SERV_PORT);
    bind(sockfd, (SA*)&servaddr, sizeof(servaddr));

    dg_echo(sockfd, (SA*)&cliaddr, sizeof(cliaddr));
}

```

图 3: UDP服务端部分代码

```

vincent@vincent-ubuntu18:~/NetworkProgramming/3$ sudo ./udp_echo_s
server receives: hello

client port:26766

```

图 4: UDP服务器运行结果

## uping程序解析

### 1.设计思路

uping程序流程介绍:

- 1.读取输入的IP地址，若为域名，则通过gethostbyname()获取对应的IP地址。
- 2.使用上一步的IP地址构建sockaddr\_in，准备发送ICMP报文。
- 3.分别构建三个ICMP数据包，并设置icmp\_seq为1、2、3，将icmp\_id设置为main函数的进程号，发送ICMP包。
- 4.根据icmp\_id的值判断所收到的是否为目标主机返回的数据包，若是则计算并打印出间隔时间、ttl、rtt等信息。

## 2.运行结果

```
root@vincent-ubuntu18:~/NetworkProgramming/3# sudo ./uping www.bilibili.com
PING www.bilibili.com(110.43.34.72): 56 bytes data in ICMP packets.
64 byte from 110.43.34.72: icmp_seq=1 ttl=55 rtt=19.000 ms
64 byte from 110.43.34.72: icmp_seq=2 ttl=55 rtt=19.000 ms
64 byte from 110.43.34.72: icmp_seq=3 ttl=55 rtt=19.000 ms

-----PING statistics-----
3 packets transmitted, 3 received , %0 lost
root@vincent-ubuntu18:~/NetworkProgramming/3# sudo ./uping 14.215.177.39
PING 14.215.177.39(14.215.177.39): 56 bytes data in ICMP packets.
64 byte from 14.215.177.39: icmp_seq=1 ttl=56 rtt=13.000 ms
64 byte from 14.215.177.39: icmp_seq=2 ttl=56 rtt=13.000 ms
64 byte from 14.215.177.39: icmp_seq=3 ttl=56 rtt=13.000 ms

-----PING statistics-----
3 packets transmitted, 3 received , %0 lost
root@vincent-ubuntu18:~/NetworkProgramming/3#
```

图 5: uping执行结果

## 四、实验总结

在UDP程序开发中，主要遇到的问题是sys/socket.h中sendto的错误，换成unp.h中的Sendto即可。

系统自带的Ping是一种计算机网络工具，用来测试数据包能否透过IP协议到达特定主机。Ping的运作原理是向目标主机传出一个ICMP的请求回显数据包，并等待接收回显回应数据包。程序会按时间和成功响应的次数估算丢包率和数据包往返时间（网络时延）。而uping则大体上实现了这样的功能，完成了对ping程序的模拟。