



江南大学
JIANGNAN UNIVERSITY

物联网工程学院

TCP/IP 课程实验报告

班 级：物联 1601

姓 名：尹达恒

学 号：1030616134

指导老师：马君霞

2018~2019 第一学期

2018 年 12 月 26 日

目录

实验七 基于 ICMP 协议的 ping 应用程序综合实验	2
1 实验目的及要求	2
2 实验环境	2
3 实验内容及步骤	2
3.1 基于 ICMP 协议的 ping 程序基本原理	2
3.2 基于 ICMP 协议的 ping 程序编写	3
3.3 本机回环测试	4
3.4 远程互通测试	4
4 实验结果	5
4.1 本地回环测试结果	5
4.2 远程互通 IP 记录	5
4.3 远程互通测试结果 1(主机 1 启动 ping)	6
4.4 远程互通测试结果 2(主机 2 启动 ping)	6
5 问题及心得	6

实验七 基于 ICMP 协议的 ping 应用程序综合实验

1 实验目的及要求

- 了解和掌握创建原始套接口的编程方法,注意与创建 SOCK_DGRAM 和 SOCK_STREAM 类型套接口的区别;
- 编写一个基于 ICMP 协议的 Ping 应用程序;
- 掌握 select(), SendEchoRequest(), WaitForEchoReply(), gethostbyname(), sendto() 以及 recvfrom() 等函数的用法。

2 实验环境

- 操作环境: Windows 10;
- 编程环境: Visual Studio 2015;
- 程序原理: ICMP 网络程序设计;
- 程序使用 Visual C++ 下的 “Win32 Console Application”。

3 实验内容及步骤

3.1 基于 ICMP 协议的 ping 程序基本原理

使用 SOCK_RAW 套接口编写 ping 程序测试网络的连通性。ping 程序通过生成一个 ICMP “回应请求” (Echo Request), 将其发送到打算查询的目标主机, 检测是否可以收到目标主机的应答, 以此测试网络的连通性。本实验中的 ping 程序是实际操作系统中提供的 ping 程序的简化版, 只进行回送测试, 测试时需要提供一个目标主机的 IP 地址信息。需要注意, 使用 ping 程序测试时可达的机器也并不能保证一个客户机就一定可以与目标主机的某个进程顺利连接; 但若 ping 程序测试不成功, 则可以确定目标机一定与该主机不能建立任何连接。

3.2 基于 ICMP 协议的 ping 程序编写

- 调试环境：Visual Studio 2015;
- 程序名称：TCP7.exe;
- 主机 IP 地址：由系统指定;
- 程序功能：启动程序后 ping 程序通过 ICMP 协议对输入的 IP 地址进行 4 次发送和接收的测试，并显示测试结果;
- 命令格式：点分十进制形式的服务器 IP 地址;
- 命令举例：192.168.137.1。

ping 程序的具体实现步骤

1. 定义 IP 和 ICMP 协议头;
2. 定义回送请求和应答数据包的结构;
3. 在主程序中初始化 Winsock 协议栈 (使用 WSAS() 函数);
4. 在主程序中调用 ping 函数;
5. 释放 Winsock 协议栈。

ping 函数的主要功能

1. 创建一个原始套接口;
2. 根据主机名查询主机地址;
3. 输出 Ping 程序要测试的目标主机地址;
4. 控制 ping 程序进行 4 次发送和接收的测试 (发送和接收分别使用 SendEchoRequest() 和 WaitForEchoReply() 函数来实现);
5. 计算传输时间，并输出提示信息;
6. 关闭原始套接口。

3.3 本机回环测试

- 测试环境：Visual Studio 2015
- 测试步骤：
 1. 在一台主机上启动 ping 程序；
 2. 在 ping 程序中输入 IP 地址 “127.0.0.1” 进行 ping 操作；
 3. 观察记录实验结果；
 4. 关闭程序。

3.4 远程互通测试

- 测试环境：Visual Studio 2015
- 测试步骤：
 1. 将两台主机连入同一个网络；
 2. 分别在两台主机上的命令行窗口输入命令 “ipconfig” 查看并记录各自的 IP 地址；
 3. 在一台主机上启动 ping 程序；
 4. 在 ping 程序中输入另一台主机的 IP 地址进行 ping 操作；
 5. 观察记录实验结果；
 6. 关闭程序；
 7. 在另一台主机上启动 ping 程序并重复步骤 4 至步骤 6。

4 实验结果

4.1 本地回环测试结果

程序运行结果：图 1.1

```
PS E:\MyPrograms\Visual Studio 2015\Projects\TCP7\Debug> .\TCP7.exe 127.0.0.1  
Pinging 127.0.0.1 [127.0.0.1] with 32 bytes of data:  
  
Reply from:127.0.0.1:bytes=32 time=16ms TTL=128  
Reply from:127.0.0.1:bytes=32 time=0ms TTL=128  
Reply from:127.0.0.1:bytes=32 time=0ms TTL=128  
Reply from:127.0.0.1:bytes=32 time=0ms TTL=128
```

图 1.1: 本地回环测试结果

4.2 远程互通 IP 记录

主机 1：图 1.2

```
本地链接 IPv6 地址. . . . . : fe80::cd69:78c3:f849:6fed%4  
IPv4 地址 . . . . . : 192.168.137.1  
子网掩码 . . . . . : 255.255.255.0  
默认网关. . . . . :
```

图 1.2: 主机 1IP

主机 2：图 1.3

```
本地链接 IPv6 地址. . . . . : fe80::f886:cd0d:63e8:5411%11  
IPv4 地址 . . . . . : 192.168.137.236  
子网掩码 . . . . . : 255.255.255.0  
默认网关. . . . . : 192.168.137.1
```

图 1.3: 主机 2IP

4.3 远程互通测试结果 1(主机 1 启动 ping)

主机 1: 图 1.4

```
PS D:\TCP7\Debug> .\TCP7.exe 192.168.137.236

Pinging 192.168.137.236 [192.168.137.236] with 32 bytes of data:

Reply from:192.168.137.236:bytes=32 time=0ms TTL=128
Reply from:192.168.137.236:bytes=32 time=0ms TTL=128
Reply from:192.168.137.236:bytes=32 time=0ms TTL=128
Reply from:192.168.137.236:bytes=32 time=0ms TTL=128
```

图 1.4: 远程互通测试主机 1 结果

4.4 远程互通测试结果 2(主机 2 启动 ping)

主机 2: 图 1.5

```
PS E:\MyPrograms\Visual Studio 2015\Projects\TCP7\Debug> .\TCP7.exe 192.168.137.1

Pinging 192.168.137.1 [192.168.137.1] with 32 bytes of data:

Reply from:192.168.137.1:bytes=32 time=0ms TTL=128
Reply from:192.168.137.1:bytes=32 time=0ms TTL=128
Reply from:192.168.137.1:bytes=32 time=0ms TTL=128
Reply from:192.168.137.1:bytes=32 time=0ms TTL=128
```

图 1.5: 远程互通测试主机 2 结果

5 问题及心得

- 问题: 在 Visual Stdio 中将 “#include “stdafx.h”” 放在头文件声明最后时会出现库文件找不到的错误: 图 1.6
 - 原因: Visual Stdio 自动生成的 stdafx.h 文件和一些重要的库文件有关, 此文件的 include 声明必须要在其他 include 声明之前, 否则其他 include 声明就可能出现找不到的情况。
 - 解决: 将 “#include “stdafx.h”” 放在文件开头。
- 心得:
 1. 实践是检验真理的唯一标准;
 2. 实验是巩固知识的最快捷径;
 3. 掌握了基于 ICMP 协议 ping 程序的原理;
 4. 对 ICMP 协议的理解更加深入;
 5. 精进了代码水平。

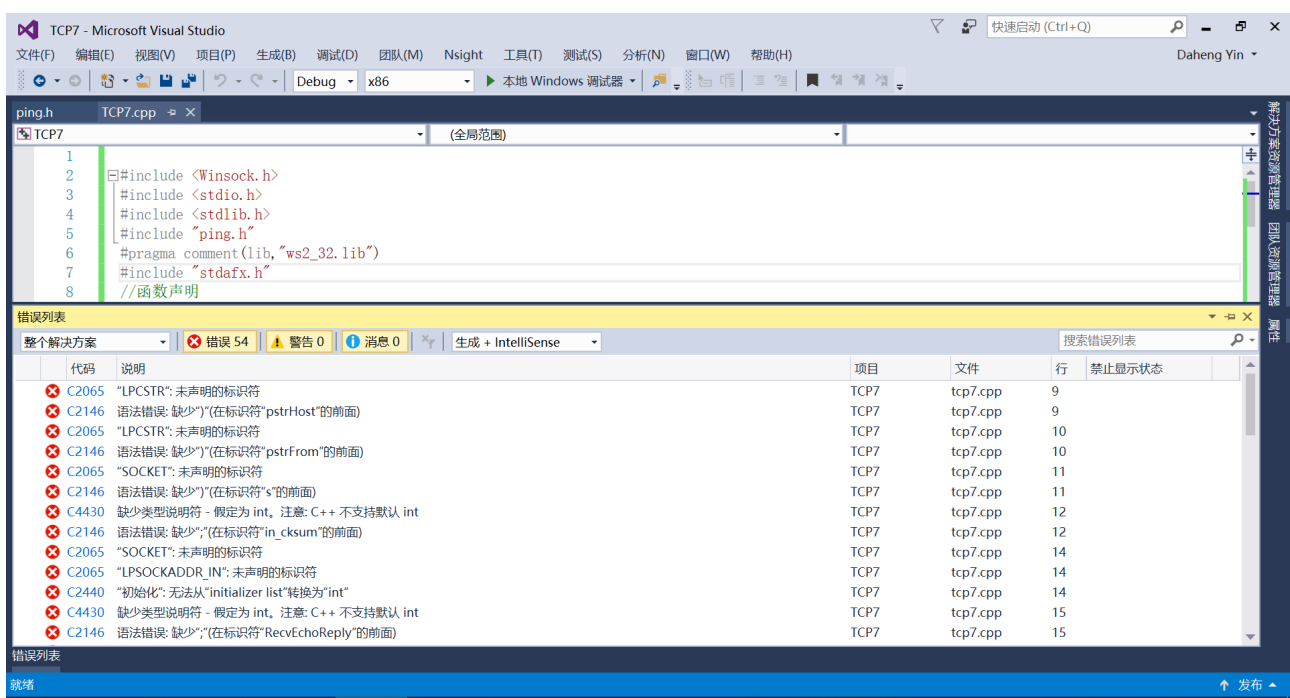


图 1.6: 错误信息