# Netsimul计算机网络教学系统 实验手册

## 概述

本实验手册是Netsimul计算机网络协议栈教学系统的附属文档，旨在提供一份参考性的实验文档，给出了实验内容与实验要求。但是，未必需要学生完成列出的所有实验。

### 系统环境

目前系统仅在Linux操作系统下进行了调试运行以及测试。系统开发使用Debian 8（Jessie）64位操作系统，通过虚拟机部署系统，进行调试运行，虚拟机使用Debian 8 32位（使用32位减少虚拟机资源需求）。表1.1给出了系统开发使用到的其他软件需求：

表1.1 系统开发软件需求

|  |  |
| --- | --- |
| 类型 | 名称及版本号 |
| 操作系统 | Debian 32/64位 8.6 |
| 构建工具 | CMake 3.0.2 |
| 编译器 | G++ 4.9.2 |
| 调试器 | GDB 7.7.1 |
| 版本管理工具 | Git 2.1.4 |
| 编辑器 | Vim 8.0 |

系统可以不依赖于虚拟机直接运行，即直接运行在物理机上。使用虚拟机可以避免稍微有些复杂的环境配置过程。

### 系统架构

我们设计的系统在运行时将包括至少两个进程：协议栈进程和应用程序进程，应用程序进程可以有多个。图1-1展示了系统的架构设计。

应用程序和协议栈分别运行在不同进程，通过共享内存的方式互相通信，以进行数据的接收和发送，通过信号进行同步。我们把完成应用程序和协议栈互相通信的模块称为 Socket 接口层，分为应用程序 Socket 和协议栈 Socket。应用程序 Socket为应用程序提供数据接收和发送的接口，通过共享内存和信号与协议栈 Socket 交互。协议栈 Socket 把从应用程序 Socket 收到的消息传递给协议栈发送，或把从协议栈收到的数据包传递给应用程序 Socket。

协议栈作为一个独立的进程一定先于应用程序运行，以通过 Socket 接口层向应用程序提供网络服务。参照 TCP/IP 模型，协议栈由传输层，网络层，数据链路层和物理层构成。物理层封装了 libpcap 的接口，是系统接收和发送数据包的基础。

实际上，图 1-1中省略了邻居子系统，从理论上来说，邻居子系统既可以划分到网络层，也可以划分到数据链路层，我们将其划分到了网络层。

另外，应该注意，协议栈模拟的物理层并不同于真实的协议栈中的物理层，物理层通常集成在网卡设备中，这里只是从教学的角度出发，为了体现协议栈完整的体系结构，我们借助于 Libpcap 进行数据包的接收和发送，称其为“物理层”。实际上，Libpcap 最终会将数据包交给网卡进行数据包的发送，以及从网卡接收数据包。

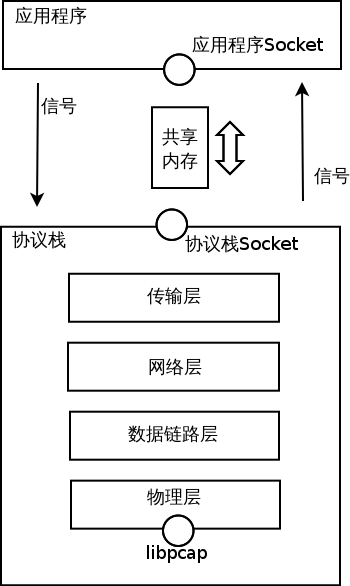


图1-1 系统架构设计图

在我们的实现中，协议栈程序名为Protocol，需要以超级用户的身份运行。

### 实验安排

系统已经实现了两台主机之间的数据发送和接收，即实现了一个计算机网络协议栈的基本功能。实验内容是将系统已经实现的部分留白，学生自行实现留白的部分以将协议栈补充完整，然后调试运行，抓包分析验证实现结果。建议使用Tcpdump进行抓包分析。

实验安排大致按照从简至难，自上而下的顺序，建议安排如下：

实验一，网络通讯程序：时间服务器。通过Netsimul协议栈实现一个时间服务器，包括服务端和客户端。建议5分。

实验二，网络通讯程序：回音程序。通过Netsimul协议栈实现一个回音程序，包括服务端和客户端。建议5分。

实验三，Socket的数据发送和接收。完成Socket数据发送和接收相关接口sendto，recvfrom，listen，accept，connect，send，recv。建议15分。

实验四，UDP协议的发送和接收。建议10分。

实验五，TCP协议三次握手。建议10分。

实验六，TCP协议四次挥手。建议10分。

实验七，TCP协议的发送和接收。建议15分。

实验八，TCP协议窗口与重传机制。（Advanced）建议20分。

实验九，TCP协议流量控制。（Advanced）建议20分。

实验十，TCP协议拥塞控制。（可选）建议20分。

实验十一，网络层报文的发送与接收。（选项设置）建议10分。

实验十二，网络层分片与重组。建议15分。

实验十三，邻居子系统：ARP协议。建议15分。

实验十四，路由子系统。（可选）建议20分。

实验十五，数据链路层成帧与解析。建议10分。

实验十六，物理层的发送与接收。建议10分。

## 实验内容

对于每个实验，实现实验要求之后均需要使用抓包工具（建议Tcpdump）验证实验结果，实验要求中不再逐一列出。

1. 网络通讯程序：时间服务器

#### 实验目标

1. 掌握简单的Socket应用程序编程
2. 掌握基于UDP协议的网络通信程序编程

#### 实验要求

实现一个简单的时间服务器以及客户端。客户端发送任意内容到服务端，服务端回复当前系统时间。

相关文件：DemoTimeServer.cpp，DemoTimeClient.cpp

相关方法：main()

建议5分

1. 网络通讯程序：回音程序

#### 实验目标

1. 掌握简单的Socket应用程序编程
2. 掌握基于TCP协议的网络通信程序编程

#### 实验要求

实现一个简单的回音程序，包括服务端和客户端，客户端发送任意内容到服务端，服务端回复客户端以相同内容。

相关文件：DemoEchoServer.cpp，DemoEchoClient.cpp

相关方法：main()

建议5分。

1. Socket的数据发送和接收接口

#### 实验目标

1. 掌握Socket的数据发送和接收原理
2. 学习Socket相关数据结构的设计
3. 学习共享内存与信号数据交互

#### 实验要求

完成Socket数据发送和接收相关接口sendto，recvfrom，listen，accept，connect，send，recv（为了降低实现难度，假定数据发送不会超过缓冲区大小4096字节）。

相关文件：Socket.cpp，ProtoSocket.cpp

相关方法：

1. CSocket.sendto()

UDP套接字发送数据方法

1. CSocket.recvfrom()

UDP套接字接收数据方法

1. CSocket.listen()

TCP套接字监听方法

1. CSocket.accept()

TCP套接字接收socket方法

1. CSocket.connect()

TCP套接字发起连接方法

1. CSocket.send()

TCP套接字发送数据方法

1. CSocket.recv()

TCP套接字接收数据方法

1. CProtoSocket对应CSocket以上方法的处理方法，handleSendTo()，handleRecvFrom()，……，handleRecv()。

建议15分。

1. UDP协议的发送和接收

#### 实验目标

1. 理解数据报的含义
2. 掌握UDP协议头部结构
3. 掌握UDP协议的数据发送和接收

#### 实验要求

完成UDP协议的数据发送和接收接口。

相关文件：UDP.cpp

相关方法：

1. CUDP.send()

UDP协议发送数据方法

1. CUDP.received()

UDP协议接收数据方法

建议10分。

1. TCP协议三次握手

#### 实验目标

1. 学习相关数据结构的设计
2. 掌握TCP协议三次握手

#### 实验要求

完成TCP协议三次握手。

相关文件：TCP.cpp

相关方法：

1. CTCP.listen()

TCP协议监听方法

1. CTCP.connect()

TCP协议连接方法

1. CTCP.recvStateProcess()

TCP协议连接状态处理方法

1. CTCP.recvListen()

TCP协议监听套接字接收到连接请求方法

1. CTCP.newConnection()

TCP协议连接建立完成方法

建议10分。

1. TCP协议四次挥手

#### 实验目标

1. 掌握TCP协议四次挥手

#### 实验要求

完成TCP协议四次挥手。

相关文件：TCP.cpp

相关方法：

1. CTCP.recvEstablished()

TCP协议连接建立后接收数据方法

1. CTCP.recvStateProcess()

TCP协议连接状态处理方法

1. CTCP.close()

TCP协议连接关闭方法

建议10分。

1. TCP协议的发送和接收

#### 实验目标

1. 理解数据流的含义
2. 学习相关数据结构的设计
3. 掌握TCP协议的数据发送和接收

#### 实验要求

实现TCP协议数据流的发送和接收。

相关文件：TCP.cpp

相关方法：

1. CTCP.send()

TCP协议发送数据方法

1. CTCP.received()

TCP协议接收数据方法

1. CTCP的doSend()，\_\_doSend()，sendNoData()辅助方法

建议15分。

1. TCP协议窗口与重传机制

#### 实验目标

1. 掌握相关数据结构的设计
2. 掌握滑动窗口协议
3. 掌握重传原理

#### 实验要求

实现滑动窗口协议，以及超时重传。

相关文件：TCP.cpp

相关方法：自行创建

建议20分。

1. TCP协议流量控制

#### 实验目标

1. 掌握相关数据结构的设计
2. 掌握流量控制原理

#### 实验要求

实现TCP协议流量控制。

相关文件：TCP.cpp

相关方法：自行创建

建议20分。

1. TCP协议拥塞控制

#### 实验目标

1. 掌握相关数据结构的设计
2. 掌握拥塞控制原理

#### 实验要求

实现TCP协议拥塞控制。

相关文件：TCP.cpp

相关方法：自行创建

建议20分。

1. 网络层报文的发送与接收

#### 实验目标

1. 学习相关数据结构的设计
2. 掌握IP报文头部结构
3. 掌握报文的发送与接收

#### 实验要求

在不考虑分片的情况下，实现报文的发送与接收。

相关文件：Network.cpp

相关方法：

1. CNetwork.send()

网络层报文的发送数据方法

1. CNetwork.received()

网络层报文的接收数据方法

建议10分。

1. 网络层分片与重组

#### 实验目标

1. 掌握相关数据结构的设计
2. 掌握报文分片与重组原理

#### 实验要求

考虑分片与重组的情况，实现报文的发送与接收。

相关文件：Network.cpp

相关方法：

1. CNetwork.fragment()

网络层报文分片方法

1. CNetwork.defragment()

网络层报文碎片接收方法

1. CNetwork.reasm()

网络层报文组装方法

1. CNetwork.clear()

网络层报文碎片清理方法

建议15分。

1. 邻居子系统：ARP协议

#### 实验目标

1. 掌握相关数据结构的设计
2. 掌握ARP数据包结构
3. 掌握ARP协议的原理

#### 实验要求

实现邻居子系统IPv4协议：ARP协议。

相关文件：Neighbor.cpp，ARP.cpp

相关方法：

1. CARP.sendDatagram()

ARP协议报文发送方法

1. CARP.cache()

ARP协议缓存地址映射方法

1. CARP.sendARP()

ARP协议ARP请求方法

1. CARP.recvARP()

ARP协议ARP请求接收方法

1. CARP.replyARP()

ARP协议ARP响应方法

1. CARP.processPendingDatagrams()

ARP协议找到地址映射后，发送对应已缓存的报文方法

建议15分。

1. 路由子系统

#### 实验目标

1. 掌握相关数据结构的设计
2. 掌握路由原理
3. 掌握OSPF与RIP路由协议

#### 实验要求

实现路由子系统，完成跨网段的数据传输，至少实现路由协议OSPF或RIP之一。

相关文件：自行创建

相关方法：自行创建

建议20分。

1. 数据链路层成帧与解析

#### 实验目标

1. 掌握以太网帧结构
2. 掌握以太网帧的封装与解析

#### 实验要求

实现数据链路层以太网帧的封装与解析。

相关文件：Link.cpp

相关方法：

1. CLink.transmit()

数据链路层发送数据帧方法

1. CLink.received()

数据链路层接收数据帧方法

建议10分。

1. 物理层的发送与接收

#### 实验目标

1. 学习通过libpcap 发送和接收数据包

#### 实验要求

利用libpcap的接口，将数据包经由物理网卡发送，以及从物理网卡接收数据包。

相关文件：Hardware.cpp

相关方法：

1. CHardware.transmit()

物理层发送数据包方法

1. CHardware::getPacket()方法

物理层接收数据包方法

1. CHardware.detectDevices()

物理层检测硬件设备（网卡）方法

建议10分。