**《计算机网络》课内实验**

**实 验 指 导 手 册**

**授课教师：**

**实验指导教师：**

**教学对象：软件工程专业**

**开课时间： 第五学期**

**北京邮电大学计算机学院**

**2022年8月**

**实验三：传输层实验（3学时）**

1. **实验目的**

通过本实验理解和掌握TCP、UDP协议功能和协议过程，掌握套接字（Socket）编程方法。

1. **实验任务**
2. TCP协议过程分析：在Linux系统中，通过Linux网络命令或自行开发的程序，实现TCP连接上的数据传输，仿真数据传输过程中的丢包，使用抓包工具抓取TCP连接上传输的TCP报文，并分析TCP协议过程：分析TCP连接建立过程、TCP差错控制过程，分析TCP连接三次握手过程中的TCP报文的端口号、顺序号、确认号字段的值；分析TCP数据报文、确认报文的顺序号、确认号字段的值；根据IP分组的长度字段（IP分组头长度字段、IP分组总长度字段），分析IP分组中所携带的TCP报文的长度；根据TCP报文中TCP报文头字段长度，分析TCP报文中携带的数据长度；分析TCP协议出错重传机制、重传的TCP数据报文的时间间隔。
3. 基于UDP协议实现提供可靠通信服务的协议软件：完成基于UDP协议的可靠通信协议设计并编码实现，在实验报告中描述该协议软件的需求分析、设计、实现、测试。
4. **实验内容**
5. 下面以自行开发的TCP应用程序为例，说明TCP协议过程分析部分的实验内容：

基于套接字编程接口（Socket API）开发基于TCP协议的客户端程序、服务器端程序，实现TCP客户端程序向TCP服务器端程序传输数据。

在TCP客户端或TCP服务器端模拟TCP连接上传输的数据的丢失：可以在TCP客户端程序中调用iptables命令设置IP分组过滤规则，以丢弃本机所发出或接收到的IP分组，通过调用iptables命令取消之前设置的分组丢弃设置，以模拟TCP报文传输过程中的丢失；也可以在TCP数据传输过程中，使用iptables命令，使得TCP应用服务器所在主机丢弃接收到的IP分组，然后再使用iptables命令恢复TCP服务器端IP分组的接收，使TCP服务器不再丢弃客户端发送来的IP分组，以此来模拟TCP分组的丢失。

观察TCP分组丢失情况下，TCP差错控制功能的实现过程。

客户端向服务器端发送完数据后，关闭客户端和服务器端的TCP连接。判断客户端和服务器端应用程序发送和接收到的数据是否相同。

1. 基于套接字编程接口，开发基于UDP协议的可靠通信协议软件，设计基于UDP协议实现可靠通信的协议机制和协议消息定义，完成基于UDP协议的可靠通信协议软件的设计与实现。
2. **实验环境**
3. 使用一台或两台桌面版的Linux系统主机：如果所使用的计算机的资源（CPU能力、内存大小、硬盘大小）足够的话，可以考虑通过虚拟机平台（VirtualBox或VMWare Workstations环境）在所使用的计算机上创建两个Linux虚拟机。也可使用两台物理主机。

如果计算机上的资源不够的话，也可以使用一台Linux系统主机或Linux虚拟机。本实验指导书中后续实验内容以在一台Linux虚拟机上的操作为例。

1. 抓包软件：Wireshark软件
2. **实验进度安排**

本实验3学时。在课内实验前：

1. 学习Linux系统环境中的iptables命令、tcpdump命令的功能与使用。
2. 学习Linux环境下套接字编程接口的使用。
3. 阅读实验指导书，理解本实验内容，然后设计实验方案，进行实验。

实验过程中和课内实验过程中，学生与教师交互实验中的实验内容及遇到的问题。

1. **实验方法与步骤**

下面分两部分说明实验方法和实验步骤。下面的实验内容以在一台桌面版本的Ubuntu操作系统主机中的实验来进行说明。**同学在阅读此指导书的基础上，设计自己的实验方案，得到自己的实验数据，并进行实验结果分析。**

1. 实验内容一：TCP协议过程分析
2. 学习Linux系统环境下的套接字编程接口，编写TCP协议客户端应用程序和TCP协议服务器端应用程序，实现TCP协议客户端程序向TCP协议服务器端程序发送数据，TCP协议服务器端应用程序能够正确接收到客户端应用程序发送的数据。需要客户端程序能够发送一定的数据量给服务器端程序，以便在数据传输过程中模拟TCP报文的丢失。TCP协议服务器端程序可以仅支持一个TCP客户端的连接。

图1所示的TCP客户端应用程序与同一Linux主机上、在TCP端口号6666上侦听的TCP服务器端应用程序建立了TCP连接之后，接收用户输入的字符串后，先后10次调用套接字函数send()函数，向TCP服务器端程序发送该字符串10次，然后关闭与服务器端应用程序的TCP连接后，退出客户端程序。

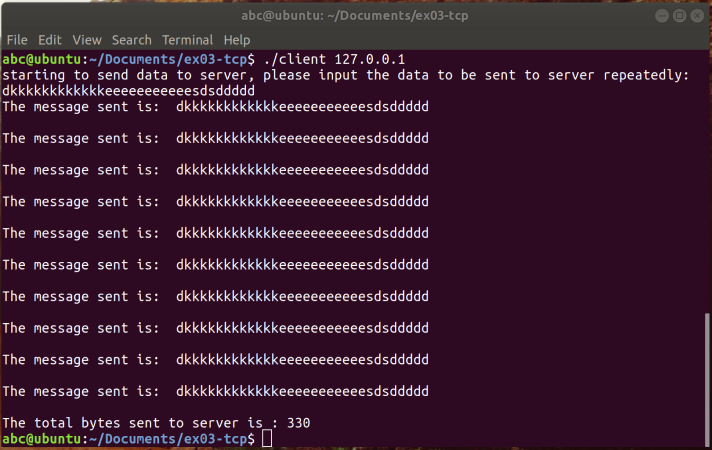


图1

图2所示的TCP服务器端程序在TCP协议6666号端口侦听TCP客户端的连接请求。该TCP服务器端应用程序申请、使用的侦听套接字（Listening socket）和连接套接字（Connected socket）工作在缺省的工作方式（即，阻塞方式）下。在响应了TCP客户端的连接请求后，TCP服务器端程序调用套接字函数recv()函数接收TCP连接上的数据，并将所接收到的数据打印出来。TCP服务器端程序循环调用recv()函数，不断从TCP连接上接收数据，直至recv()函数返回值为0（返回值值为0表示TCP连接对端实体关闭了TCP连接）或返回值小于0（返回值小于0表示recv函数执行异常）。图2所示的服务器端程序的输出，对应的TCP服务器端应用程序执行过程中调用了3次recv()函数，通过前两次recv()函数调用，TCP服务器端应用程序分别接收到了132个字节和198个字节数据；第三次调用recv()函数，recv()函数的返回值等于0，然后，TCP服务器端应用程序以此判断对应的TCP客户端程序关闭了TCP连接，因此，在shell窗口中输出“The client closed the connection”提示信息。

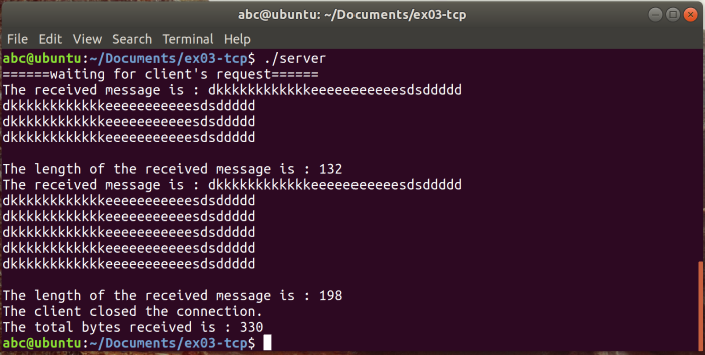


图2

图3为在上述图1和图2所示的TCP客户端和TCP服务器端应用程序建立好TCP连接之前，在另一Shell窗口中，使用sudo tcpdump -i lo tcp port 6666 -w "tcpdump.log"命令抓取在环回接口（lo）上传递的、端口号为6666的TCP协议消息。在TCP客户端与TCP服务器端应用程序通信结束后，在图3所示窗口中，用“Ctrl+C”终止此tcpdump命令的执行，tcpdump命令退出，并将抓取的报文写入文件tcpdump.log中。

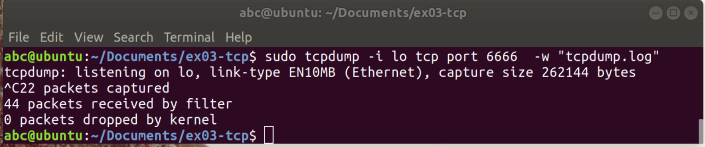


图3

图4为使用“sudo tcpdump -r tcpdump.log”命令显示tcpdump.log文件中存储的消息。

图5为用wireshark抓包软件打开图3中形成的tcpdump.log抓包文件。在Wireshark窗口中，可以看到抓取的TCP客户端和TCP服务器端之间传递的TCP消息。参见图6，对应于图1和图2所示的这次通信过程，TCP客户端在与TCP服务器端建立了TCP连接后，客户端TCP协议向服务器端TCP协议发送了9条携带数据的TCP报文（消息编号：4、6、8、10、12、14、16、18、20），其中编号为20的TCP报文携带的数据长度是66字节，编号为20的TCP报文的FIN比特位为1，也是一条关闭TCP连接的TCP报文（参见图6）。

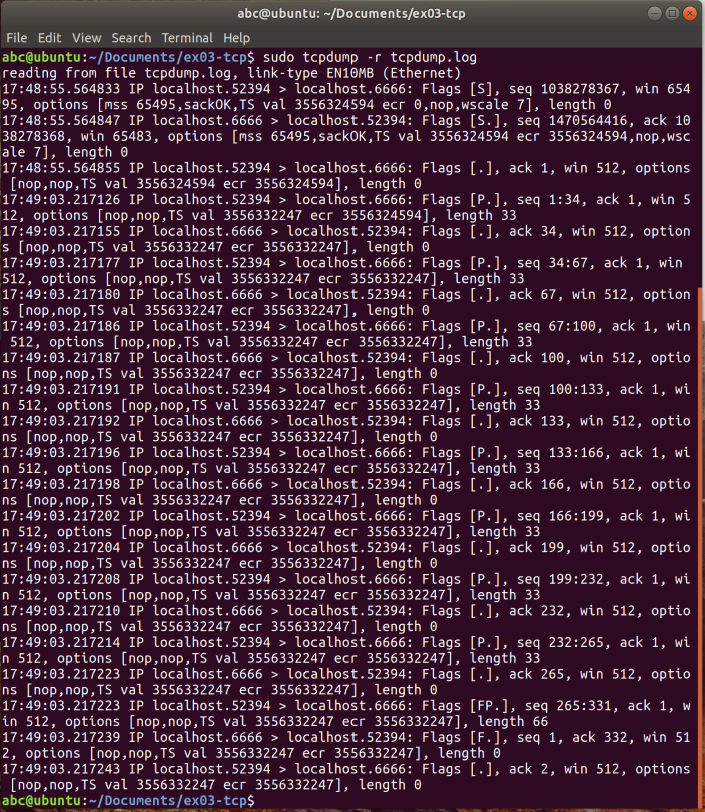


图4

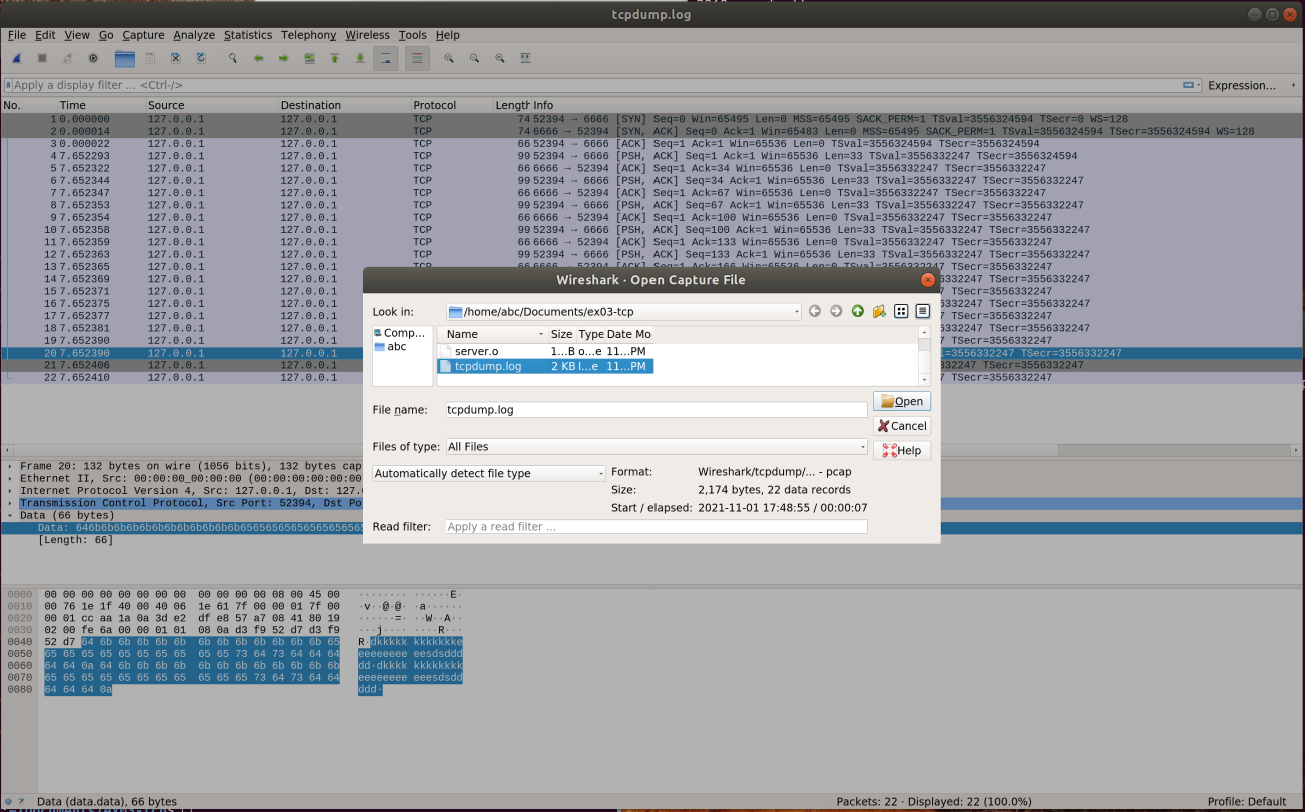


图5

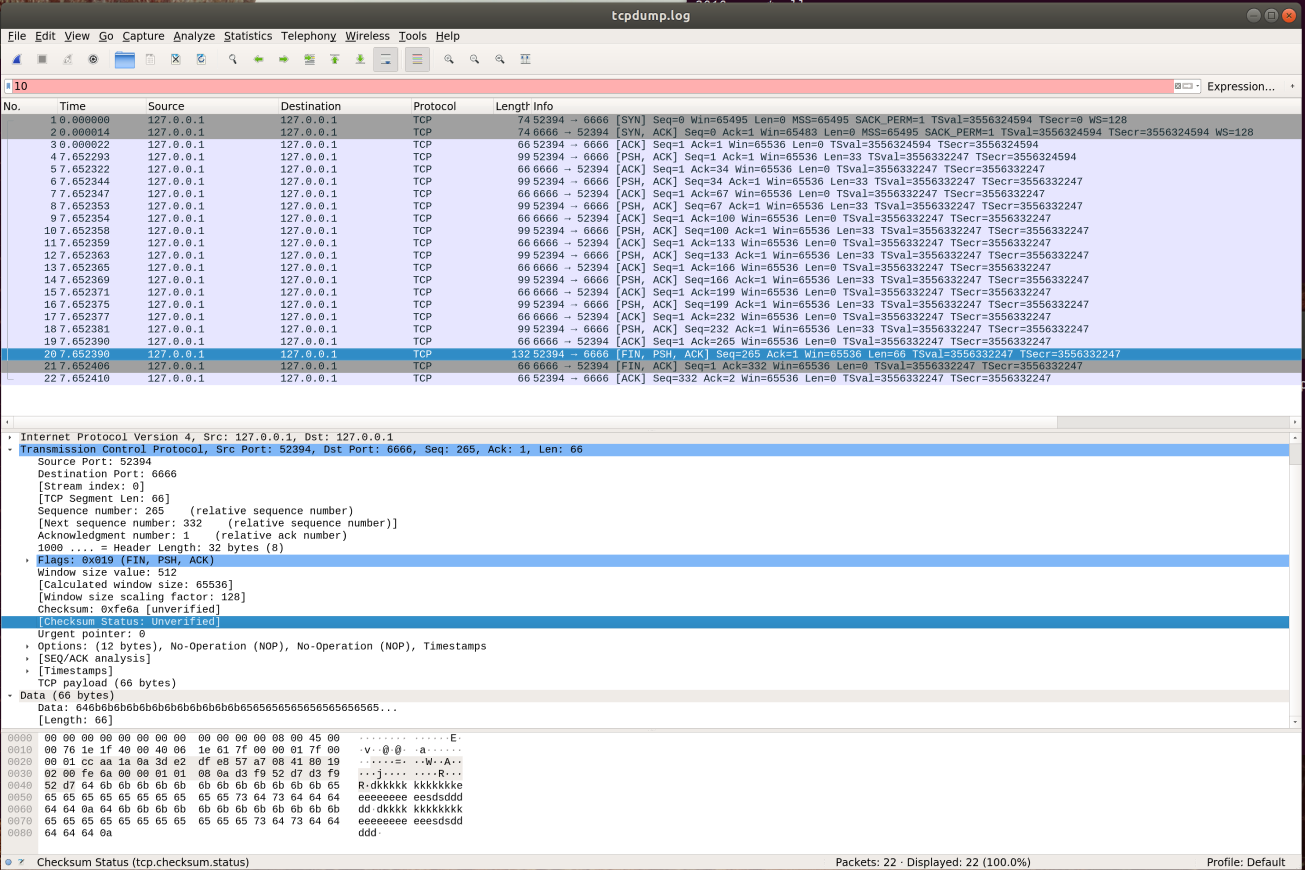


图6

1. 再次启动TCP服务器端应用程序，在启动服务器端程序之前，执行sudo tcpdump -i lo tcp port 6666 -w "tcpdump3.log"命令，以抓取TCP客户端与TCP服务器端之间传送、且TCP端口（包括源TCP端口以及目的TCP端口）为6666的TCP报文，所抓取的TCP报文存入tcpdump3.log文件中。参见图7。

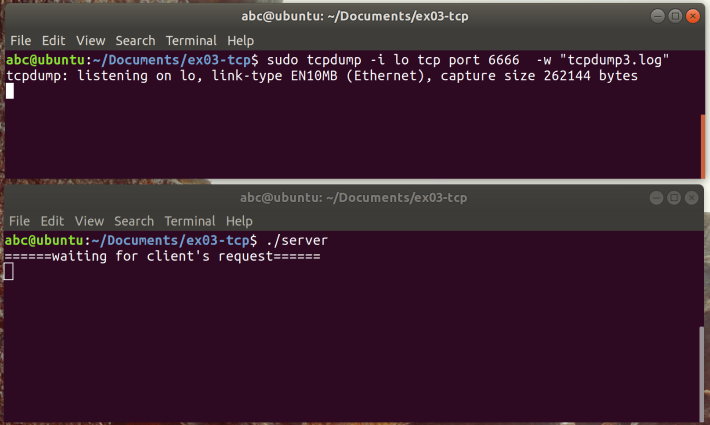


图7

此后启动TCP客户端程序，参见图8。所启动的客户端程序首先建立与TCP服务器端程序的TCP连接，然后提示用户输入待发送的数据。但启动TCP客户端程序后，我们先不按照客户端程序的提示输入需要发送的数据，待设置了IP分组过滤规则后再输入待发送的数据。

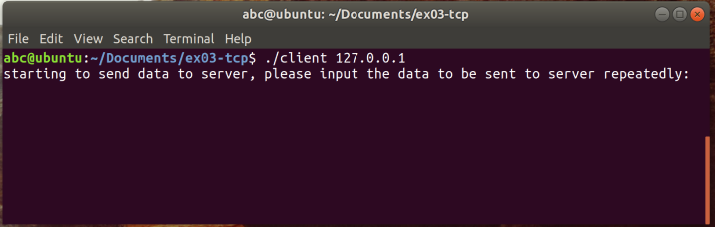


图8

此后在另一Shell窗口中，使用命令“sudo iptables -A INPUT -s 127.0.0.1 -j DROP”使得此Linux主机丢弃所接收到的源IP地址为127.0.0.1的IP分组。这个iptables命令的执行将会丢弃此主机接收到的、封装在IP分组中的TCP报文，以此来模拟TCP报文的丢失。参见图9。

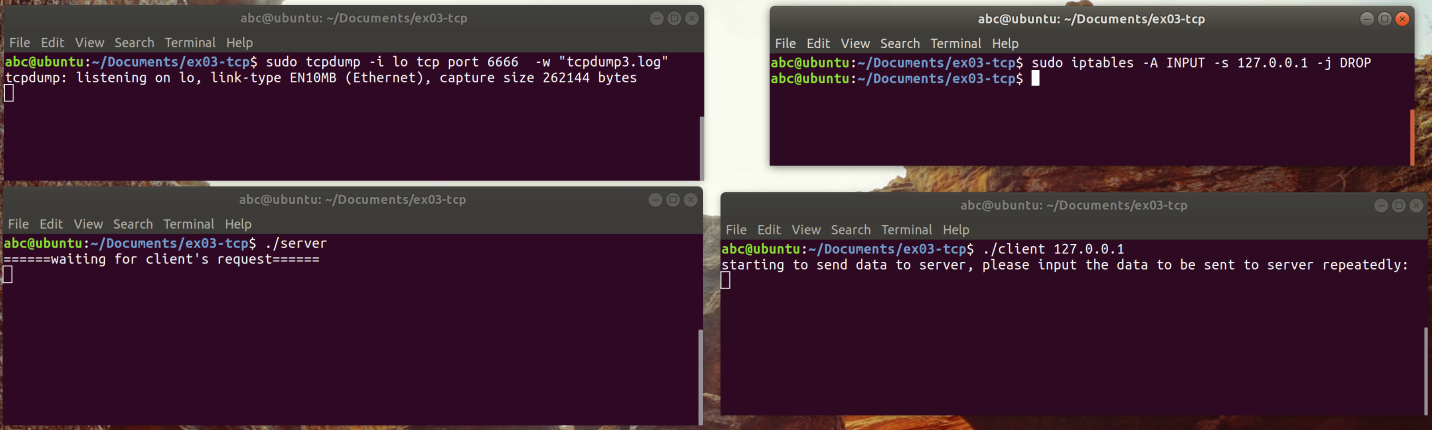


图9

此后，在TCP客户端程序窗口输入需要发送的数据字符串，参加图10。在输入了需要发送的数据后，及时执行“sudo iptables -D INPUT -s 127.0.0.1 -j DROP”命令，以便及时删除之前通过iptables命令设置的IP分组过滤规则。在此先后两次iptables命令执行的中间时间段，客户端发送给服务器端的数据被丢弃掉了。当删除掉IP分组过滤规则后，可以看到服务器端接收到了客户端发送的数据。因为，TCP协议重传了没有得到确认消息的数据。

注：如果在通过iptables命令设置过滤IP分组规则后，若没有及时删除此过滤规则，而此时客户端程序已经退出的话，有可能观察不到服务器端接收到数据。对此情况，可考虑在客户端发送完数据后，不关闭TCP连接也不退出客户端程序的执行；等删除掉过滤IP分组的规则后，再关闭TCP连接。

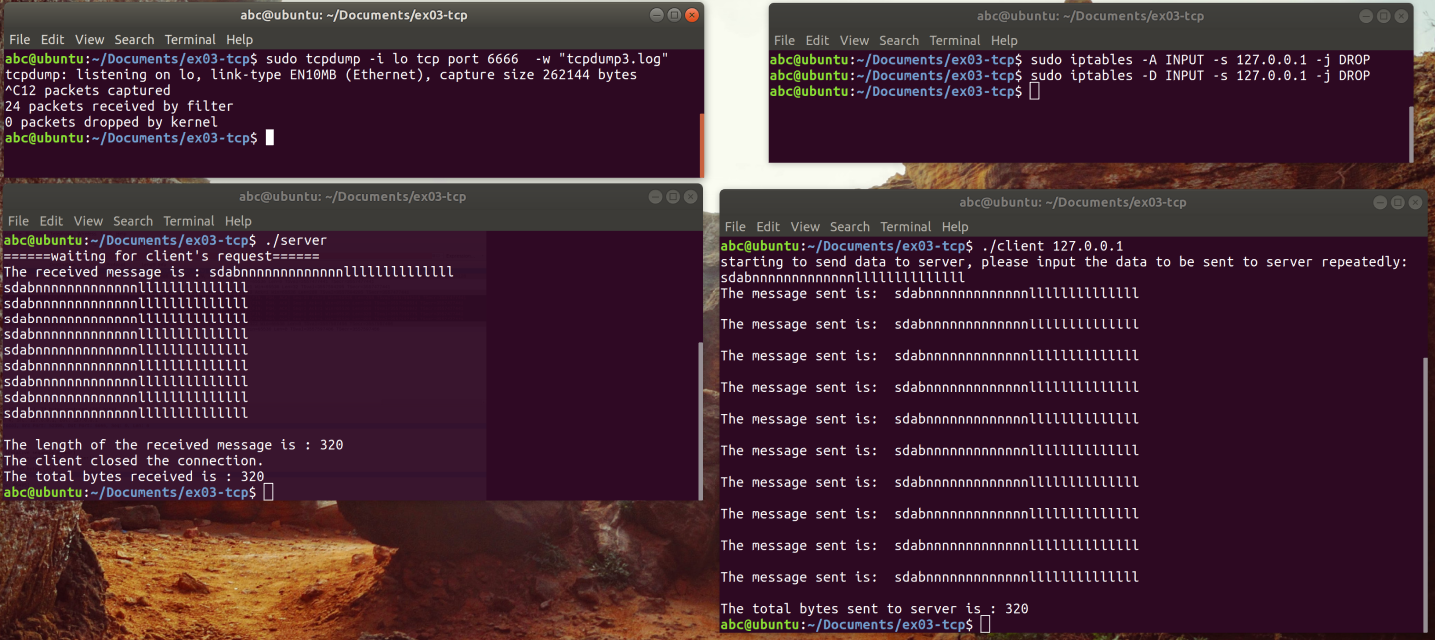


图10

图11为用wireshark抓包软件打开图10中形成的tcpdump3.log抓包文件。图11中，编号为1、2、3的数据包是TCP连接的三次握手数据包。编号为4的数据包为客户端TCP协议发出的第一个数据包，TCP报文中的数据长度为32字节。编号为5的数据包为客户端TCP协议发出的第二个数据包，TCP报文中携带的数据长度为288个字节，编号为5的TCP报文同时也是客户端发出的拆除TCP连接的报文。编号为6、7、8、9、10的包为客户端TCP协议重传的数据包，每个TCP数据包中携带的数据长度为320字节。**注意这些重传的TCP报文间的时间间隔。**编号为11的包为服务器端TCP协议给客户端发出的响应消息，对所收到的320个字节的数据进行确认；同时，编号为11的数据包也是一个拆除TCP连接的报文，也是对客户端拆除TCP连接的请求消息的响应消息。编号为12包为客户端对服务器端发出的编号为11的消息的响应消息。

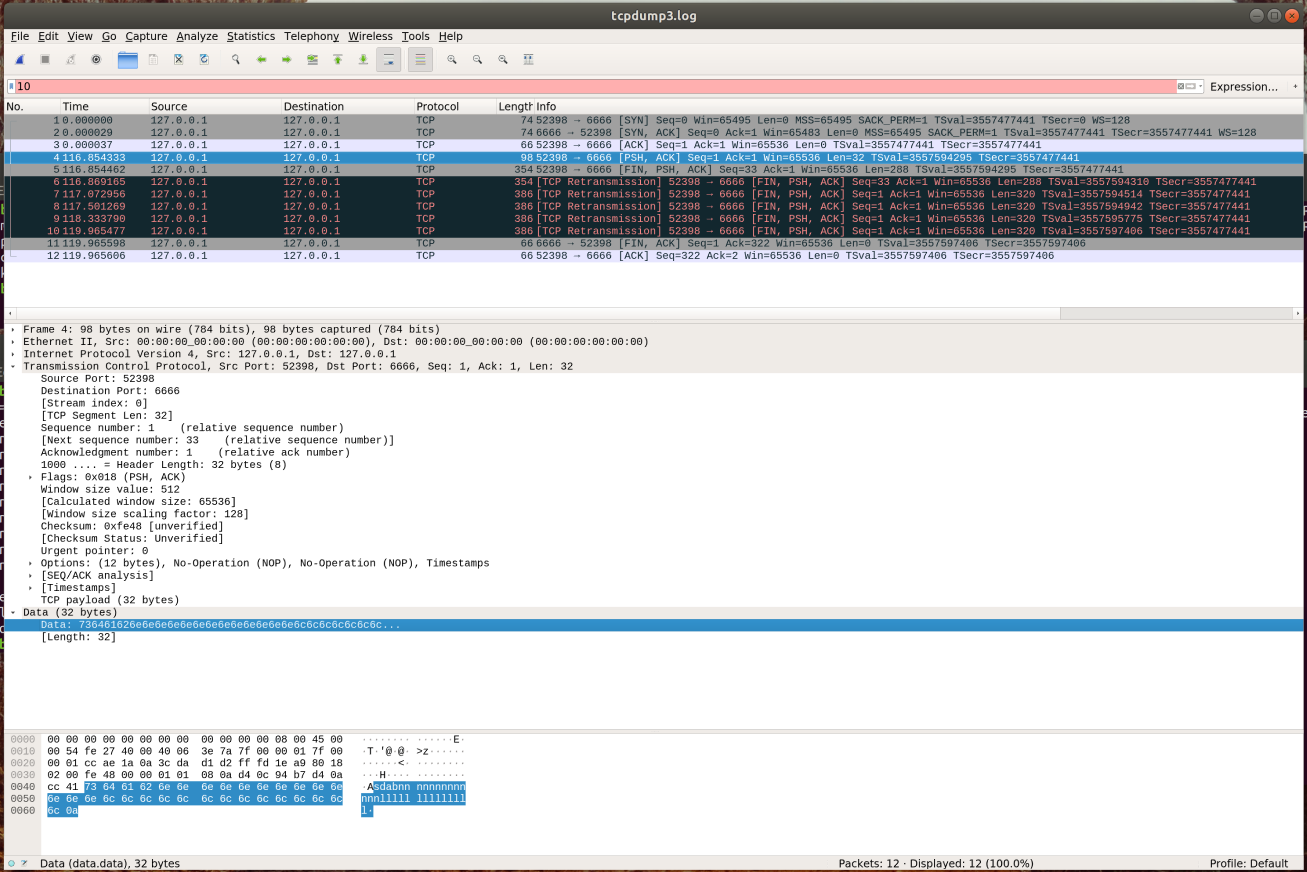


图11

也可在客户端应用程序中调用“iptables”命令，以设置过滤IP分组的规则，以及删除掉已设置的IP分组过滤规则。图12为在客户端程序中调用“iptables”命令设置IP分组过滤规则（图12中代码行第49行）和删除IP分组过滤规则（图12中代码行第53行）的代码片段。

图13为运行修改后的客户端程序与服务器端程序的通信过程的结果截图。将用tcpdump抓包后的结果存入tcpdump4.log文件中。注意：在使用此客户端（client）时，因为此客户端程序中需要调用iptables命令，而iptables命令的执行需要超级用户权限，所以执行此客户端程序时使用的命令是“sudo ./client 127.0.0.1”，以便在客户端程序执行到调用iptables命令时，不需要用户再输入超级用户的密码。

图14为用wireshark抓包软件打开上述所形成的抓包文件tcpdump4.log文件。可以看到TCP客户端在与TCP服务器端建立起TCP连接后，客户端TCP协议发出了两个TCP数据报文（图14中的编号为4、6的报文），每个TCP报文中携带的数据是52字节长。然后，客户端的TCP协议发出的后续报文（编号为7的报文）携带了416个字节数据。编号为8的报文为TCP服务器重发的响应报文，编号为8的报文对客户端TCP协议发出的第一个数据包再次进行确认。编号为9的报文为客户端TCP协议对所发出的第二个数据报文（编号为6的报文）的重发。编号为10的报文对客户端TCP协议发出的第二个、第三个TCP数据报文（编号为9、编号为7的报文）进行了确认。

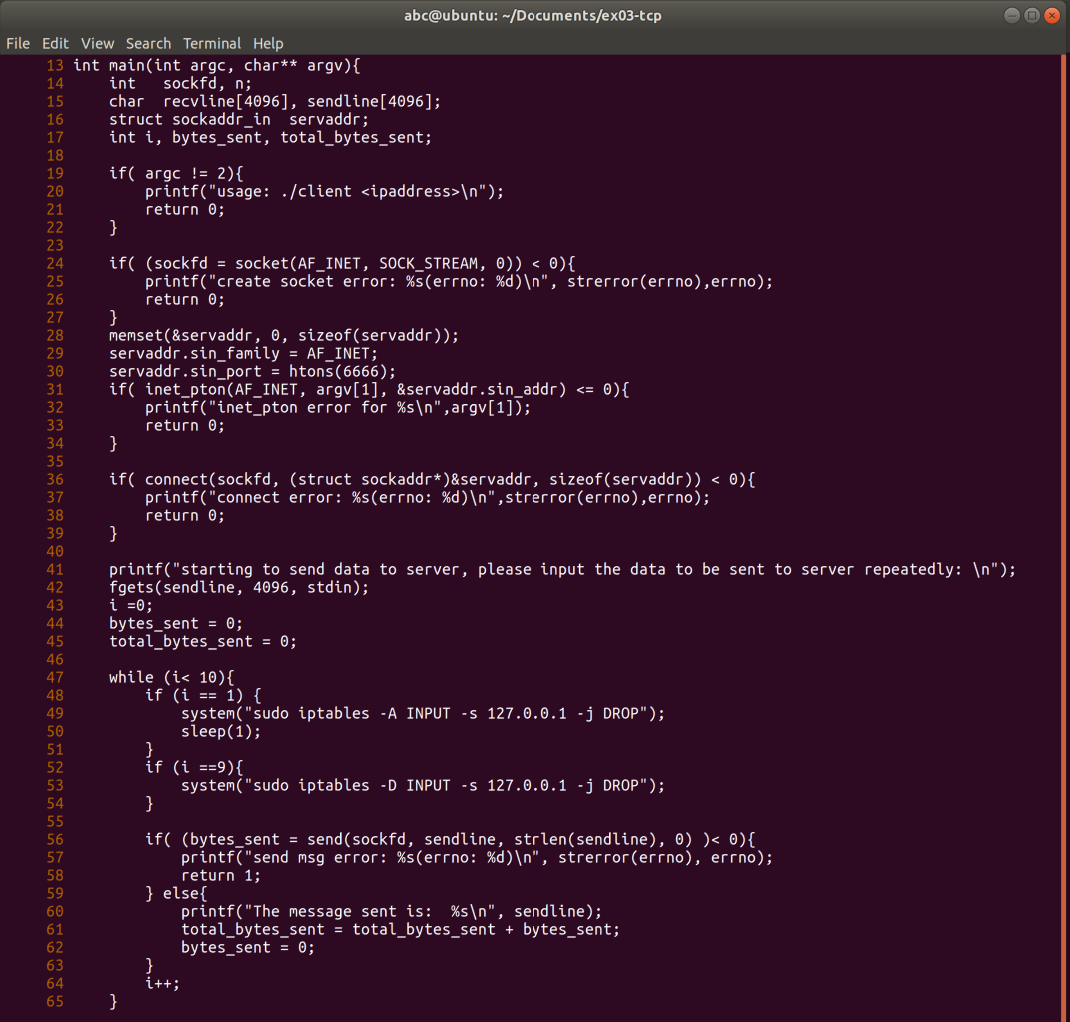


图12

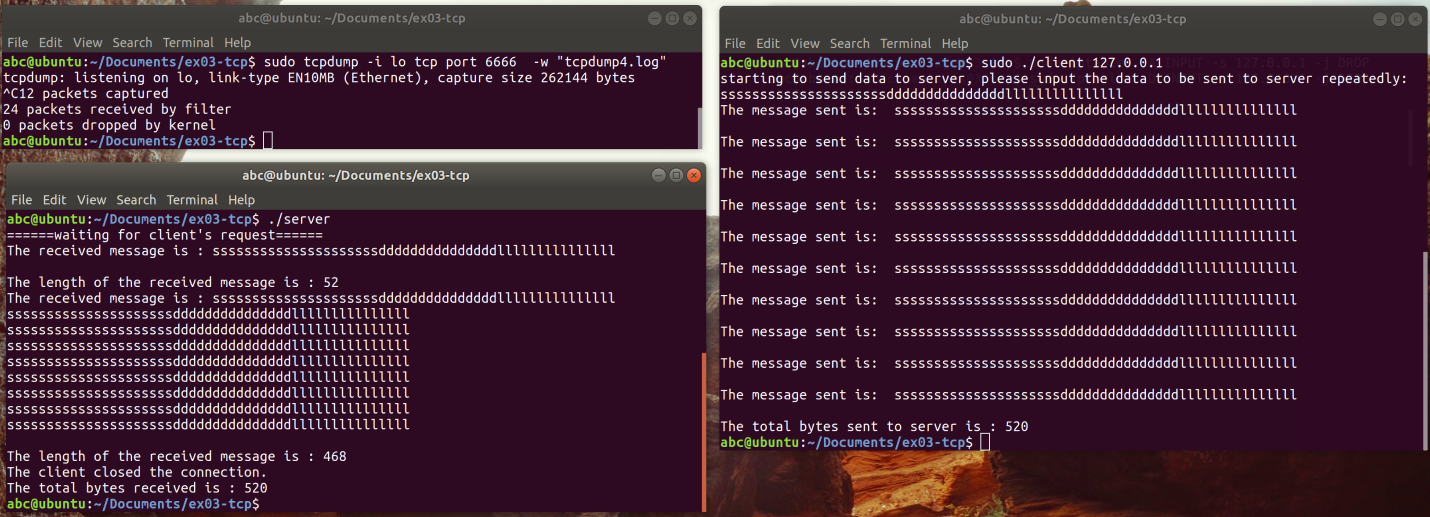


图13

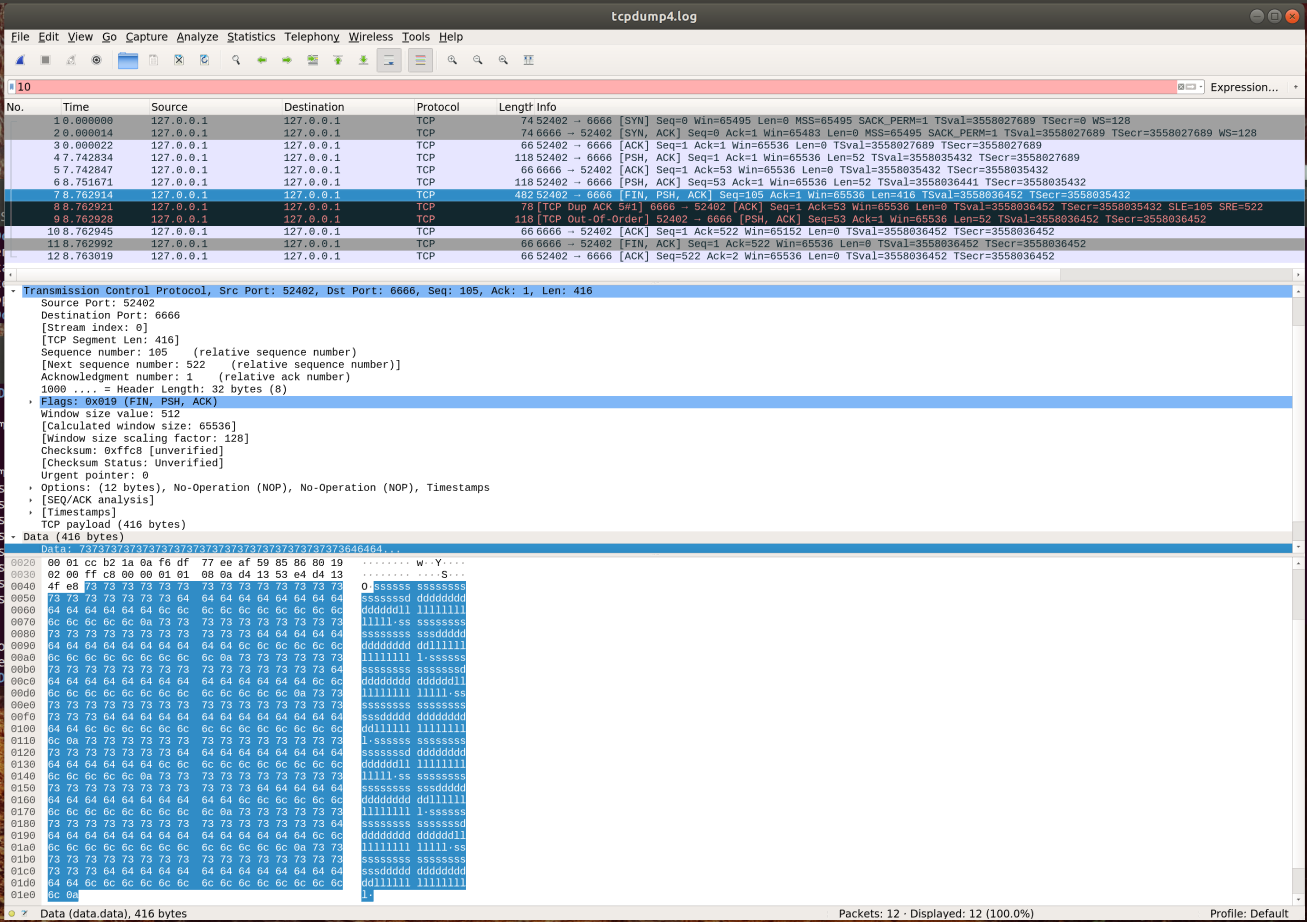


图14

1. 实验内容二：开发基于UDP协议的可靠通信协议软件
2. 使用套接字编程接口函数，开发基于UDP协议的客户端和服务器端程序，实现客户端程序和服务器程序之间数据的发送、接收。对基于套接字函数开发使用UDP协议进行通信的程序，可参考本课程实验二中的例子程序。
3. UDP客户端程序和UDP服务器端程序可运行在同一Linux主机中。可参考本实验指导书中第一部分实验内容“TCP协议过程分析”中使用的iptables命令模拟UDP报文的丢失，也可采用其他方法模拟UDP报文的丢失。可不考虑先发的IP分组后到达接收方的情况。
4. 设计基于UDP协议的可靠通信协议，实现在UDP报文丢失的情况下，UDP客户端应用程序与UDP服务器端应用程序之间能够正确收发数据。描述该协议的工作原理，给出该协议消息格式定义，说明该协议消息中各个字段的作用。可参考TCP协议差错控制机制、数据链路层的回退N步的滑动窗口协议、选择性重传的滑动窗口协议或其他协议的差错控制机制设计本协议差错控制机制。
5. **实验管理规则及实验验收**

实验完成后填写并提交实验报告。

对本实验的第一部分实验内容，需要描述实验方案，实验过程，给出实验过程系统界面截图，分析TCP连接建立过程、TCP差错控制过程，分析TCP连接三次握手过程中的TCP报文的端口号、顺序号、确认号字段的值；分析TCP数据报文、确认报文的顺序号、确认号字段的值；根据IP分组的长度字段（IP分组头长度字段、IP分组总长度字段），分析IP分组中所携带的TCP报文的长度；根据TCP报文中的TCP报文头字段长度，分析TCP报文中携带的数据长度；分析TCP协议出错重传机制、重传的TCP数据报文的时间间隔。

对本实验第二部分实验内容，实验报告需描述基于UDP协议的可靠通信协议的协议机制、协议消息的字段组成及每个字段的作用，描述协议软件工作原理，给出系统实现结果。

实验报告需要提交所开发程序的源程序。

实验报告需要对实验过程描述清晰、规范、准确。