TCP网络传输机制实验二报告

张磊 2017K8009922027

一、实验题目

TCP 网络传输机制实验二

二、 实验内容

实验内容一

1. 运行给定网络拓扑（tcp_topo.py）
2. 在节点 h1 上执行 TCP 程序

   在节点 h1 上运行 TCP 协议栈的服务器模式 `./tcp_stack server 10001`
3. 在节点 h2 上执行 TCP 程序

   在节点 h2 上运行 TCP 协议栈的客户端模式 `./tcp_stack client 10.0.0.1 10001`

   client 向 server 发送数据，server 将数据 echo 给 client
4. 使用 tcp_stack.py 替换其中任意一端，对端都能够正确收发数据

实验内容二

1. 修改 tcp_apps.c（以及tcp_stack.py），使之能够收发文件
2. 执行 create_randfile.sh，生成待传输数据文件 client-input.dat
3. 运行给定网络拓扑（tcp_topo.py）
4. 在节点 h1 执行 TCP 程序

在节点 h1 上运行 TCP 协议栈的服务器模式 `./tcp_stack server 10001`

5. 在节点 h2 上执行 TCP 程序

   在节点 h2 上运行 TCP 协议栈的客户端模式 `./tcp_stack client 10.0.0.1 10001`

   client 向 serve r发送数据，server 将收到的数据存储到文件 server-output.dat
6. 使用 tcp_stack.py 替换其中任意一端，对端都能够正确收发数据

三、实验流程

实验内容一

1. 在tcp_sock.c中实现tcp_sock_read函数和tcp_sock_write函数
2. 在tcp_in.c中增加对收取数据包的处理
3. 在节点h1上运行 `./tcp_stack server 10001` 启动TCP协议栈服务器模式
4. 在节点h2上运行 `./tcp_stack client 10.0.0.1 10001` 启动TCP协议栈客户端模式
5. client端向server端发送数据，server端收到数据后echo给client端
6. 保持节点h2不变，将节点h1的程序替换为tcp_stack.py `python tcp_stack.py server 10001` 再次进行实验
7. 保持节点h1不变，将节点h2的程序替换为tcp_stack.py `python tcp_stack.py client 10.0.0.1 10001` 再次进行实验

实验内容二

1. 修改tcp_apps.c，tcp_stack.py
2. 执行 `./create_randfile.sh` 生成待传输的数据文件client-input.dat
3. 在节点h1上运行 `./tcp_stack server 10001` 启动TCP协议栈服务器模式
4. 在节点h2上运行 `./tcp_stack client 10.0.0.1 10001` 启动TCP协议栈客户端模式
5. client端向server端发送数据，server端收到数据后存储到server-output.dat
6. 保持节点h2不变，将节点h1的程序替换为tcp_stack.py `python tcp_stack.py server 10001` 再次进行实验
7. 保持节点h1不变，将节点h2的程序替换为tcp_stack.py `python tcp_stack.py`

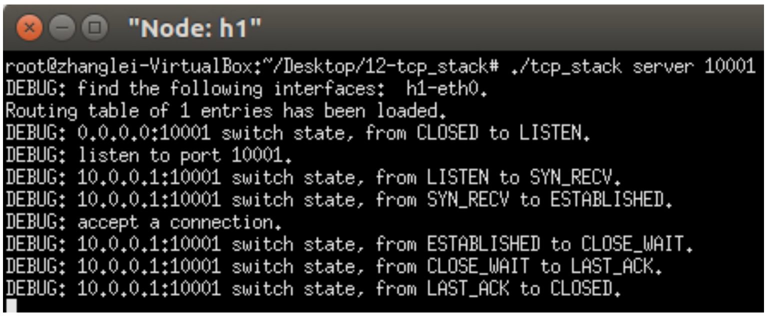
`client 10.0.0.1 10001` 再次进行实验

8. 每次实验server端完成数据接收后，执行 `md5sum server-output.dat` 和 `md5sum client-input.dat` 比较两个文件是否完全相同

四、实验结果

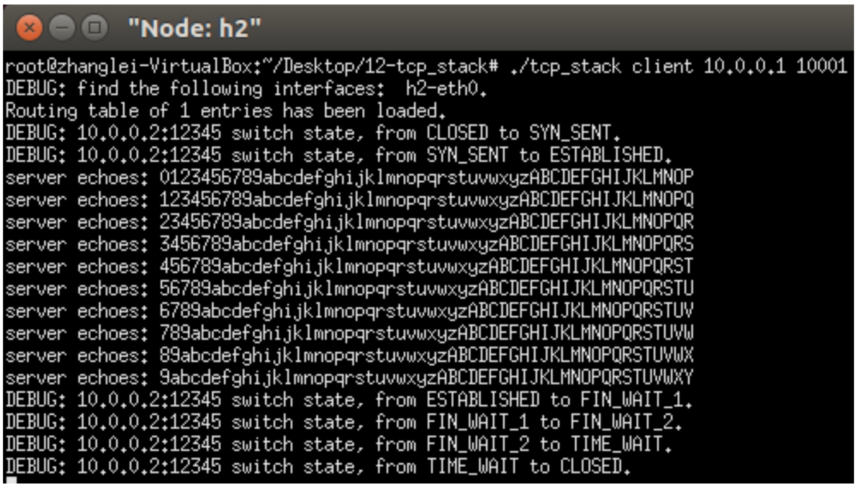
实验内容一

1. **h1和h2均为tcp_stack.c程序**



```
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack# ./tcp_stack server 10001
DEBUG: find the following interfaces:  h1-eth0.
Routing table of 1 entries has been loaded.
DEBUG: 0.0.0.0:10001 switch state, from CLOSED to LISTEN.
DEBUG: listen to port 10001.
DEBUG: 10.0.0.1:10001 switch state, from LISTEN to SYN_RECV.
DEBUG: 10.0.0.1:10001 switch state, from SYN_RECV to ESTABLISHED.
DEBUG: accept a connection.
DEBUG: 10.0.0.1:10001 switch state, from ESTABLISHED to CLOSE_WAIT.
DEBUG: 10.0.0.1:10001 switch state, from CLOSE_WAIT to LAST_ACK.
DEBUG: 10.0.0.1:10001 switch state, from LAST_ACK to CLOSED.
```

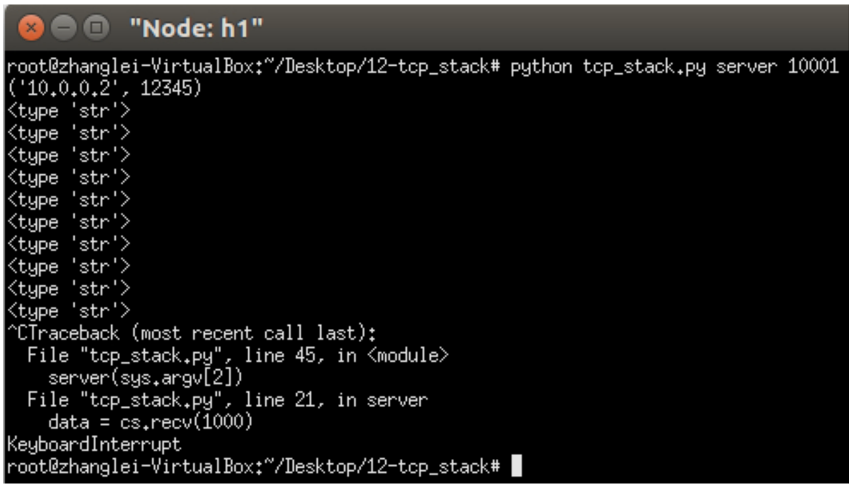
h1 output



```
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack# ./tcp_stack client 10.0.0.1 10001
DEBUG: find the following interfaces:  h2-eth0.
Routing table of 1 entries has been loaded.
DEBUG: 10.0.0.2:12345 switch state, from CLOSED to SYN_SENT.
DEBUG: 10.0.0.2:12345 switch state, from SYN_SENT to ESTABLISHED.
server echoes: 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOP
server echoes: 123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQ
server echoes: 23456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQR
server echoes: 3456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRS
server echoes: 456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRST
server echoes: 56789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTU
server echoes: 6789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUV
server echoes: 789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVW
server echoes: 89abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWX
server echoes: 9abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXY
DEBUG: 10.0.0.2:12345 switch state, from ESTABLISHED to FIN_WAIT_1.
DEBUG: 10.0.0.2:12345 switch state, from FIN_WAIT_1 to FIN_WAIT_2.
DEBUG: 10.0.0.2:12345 switch state, from FIN_WAIT_2 to TIME_WAIT.
DEBUG: 10.0.0.2:12345 switch state, from TIME_WAIT to CLOSED.
```

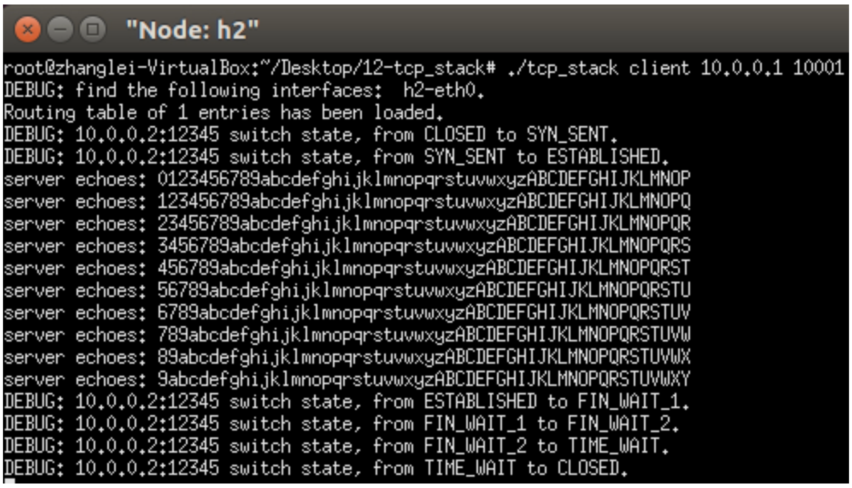
h2 output

2. **h1为tcp_stack.py程序，h2为tcp_stack.c程序**



```
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack# python tcp_stack.py server 10001
('10.0.0.2', 12345)
<type 'str'>
<type 'str'>
<type 'str'>
<type 'str'>
<type 'str'>
<type 'str'>
<type 'str'>
<type 'str'>
<type 'str'>
<type 'str'>
^CTraceback (most recent call last):
  File "tcp_stack.py", line 45, in <module>
    server(sys.argv[2])
  File "tcp_stack.py", line 21, in server
    data = cs.recv(1000)
KeyboardInterrupt
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack# 
```

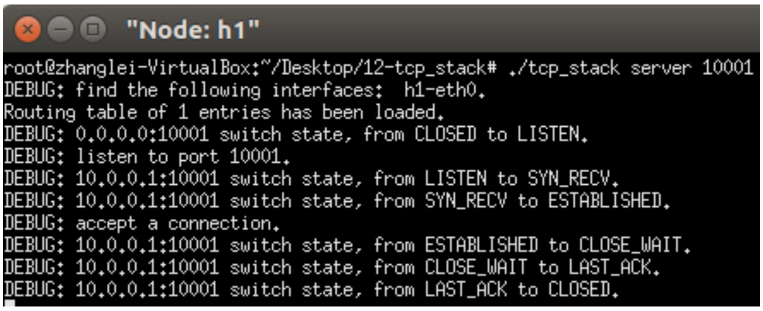
h1 output



```
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack# ./tcp_stack client 10.0.0.1 10001
DEBUG: find the following interfaces:  h2-eth0.
Routing table of 1 entries has been loaded.
DEBUG: 10.0.0.2:12345 switch state, from CLOSED to SYN_SENT.
DEBUG: 10.0.0.2:12345 switch state, from SYN_SENT to ESTABLISHED.
server echoes: 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOP
server echoes: 123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQ
server echoes: 23456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQR
server echoes: 3456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRS
server echoes: 456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRST
server echoes: 56789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTU
server echoes: 6789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUV
server echoes: 789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVW
server echoes: 89abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWX
server echoes: 9abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXY
DEBUG: 10.0.0.2:12345 switch state, from ESTABLISHED to FIN_WAIT_1.
DEBUG: 10.0.0.2:12345 switch state, from FIN_WAIT_1 to FIN_WAIT_2.
DEBUG: 10.0.0.2:12345 switch state, from FIN_WAIT_2 to TIME_WAIT.
DEBUG: 10.0.0.2:12345 switch state, from TIME_WAIT to CLOSED.
```

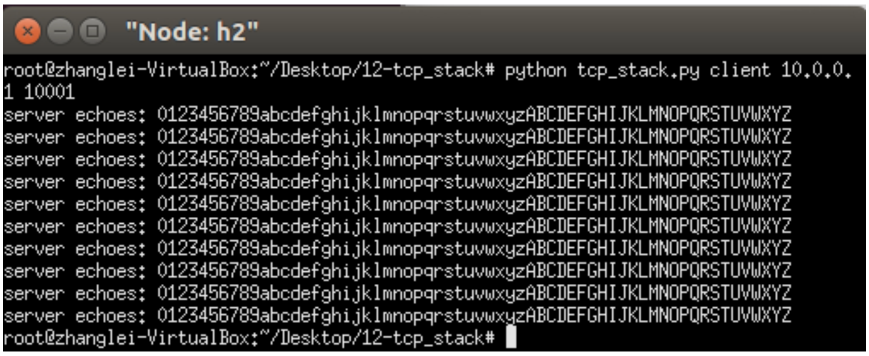
h2 output

3. **h1为tcp_stack.c程序，h2为tcp_stack.py程序**



```
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack# ./tcp_stack server 10001
DEBUG: find the following interfaces:  h1-eth0.
Routing table of 1 entries has been loaded.
DEBUG: 0.0.0.0:10001 switch state, from CLOSED to LISTEN.
DEBUG: listen to port 10001.
DEBUG: 10.0.0.1:10001 switch state, from LISTEN to SYN_RECV.
DEBUG: 10.0.0.1:10001 switch state, from SYN_RECV to ESTABLISHED.
DEBUG: accept a connection.
DEBUG: 10.0.0.1:10001 switch state, from ESTABLISHED to CLOSE_WAIT.
DEBUG: 10.0.0.1:10001 switch state, from CLOSE_WAIT to LAST_ACK.
DEBUG: 10.0.0.1:10001 switch state, from LAST_ACK to CLOSED.
```

h1 output

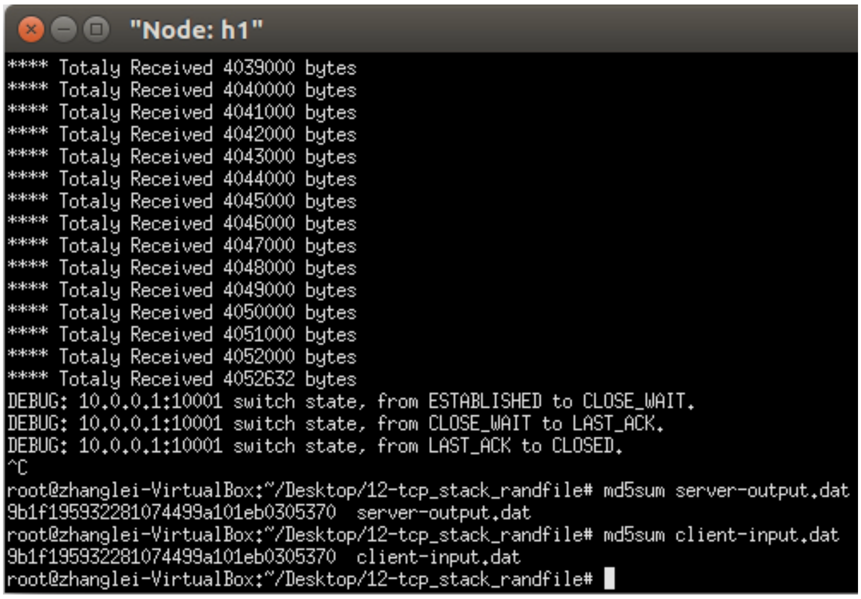


```
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack# python tcp_stack.py client 10.0.0.
1 10001
server echoes: 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
server echoes: 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
server echoes: 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
server echoes: 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
server echoes: 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
server echoes: 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
server echoes: 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
server echoes: 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
server echoes: 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
server echoes: 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack# 
```

h2 output

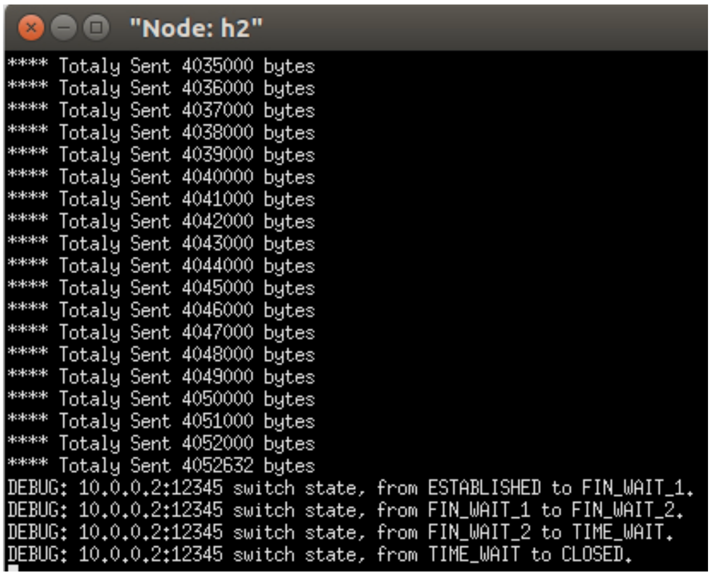
实验内容二

1. **h1和h2均为tcp_stack.c程序**



```
**** Totaly Received 4039000 bytes
**** Totaly Received 4040000 bytes
**** Totaly Received 4041000 bytes
**** Totaly Received 4042000 bytes
**** Totaly Received 4043000 bytes
**** Totaly Received 4044000 bytes
**** Totaly Received 4045000 bytes
**** Totaly Received 4046000 bytes
**** Totaly Received 4047000 bytes
**** Totaly Received 4048000 bytes
**** Totaly Received 4049000 bytes
**** Totaly Received 4050000 bytes
**** Totaly Received 4051000 bytes
**** Totaly Received 4052000 bytes
**** Totaly Received 4052632 bytes
DEBUG: 10.0.0.1:10001 switch state, from ESTABLISHED to CLOSE_WAIT.
DEBUG: 10.0.0.1:10001 switch state, from CLOSE_WAIT to LAST_ACK.
DEBUG: 10.0.0.1:10001 switch state, from LAST_ACK to CLOSED.
^C
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack_randfile# md5sum server-output.dat
9b1f195932281074499a101eb0305370  server-output.dat
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack_randfile# md5sum client-input.dat
9b1f195932281074499a101eb0305370  client-input.dat
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack_randfile# 
```

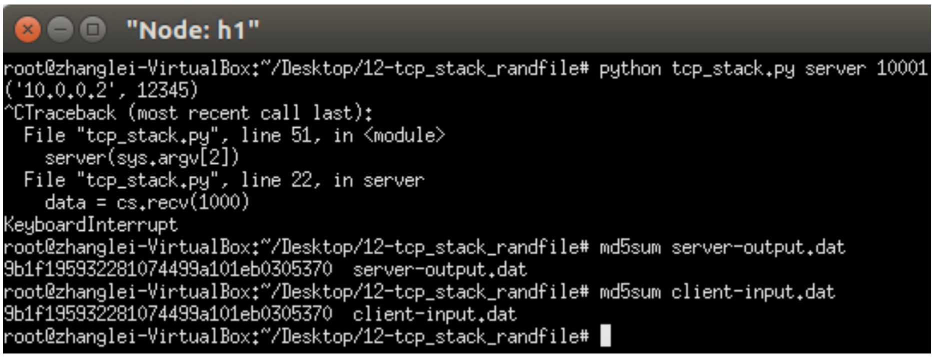
h1 output



```
**** Totaly Sent 4035000 bytes
**** Totaly Sent 4036000 bytes
**** Totaly Sent 4037000 bytes
**** Totaly Sent 4038000 bytes
**** Totaly Sent 4039000 bytes
**** Totaly Sent 4040000 bytes
**** Totaly Sent 4041000 bytes
**** Totaly Sent 4042000 bytes
**** Totaly Sent 4043000 bytes
**** Totaly Sent 4044000 bytes
**** Totaly Sent 4045000 bytes
**** Totaly Sent 4046000 bytes
**** Totaly Sent 4047000 bytes
**** Totaly Sent 4048000 bytes
**** Totaly Sent 4049000 bytes
**** Totaly Sent 4050000 bytes
**** Totaly Sent 4051000 bytes
**** Totaly Sent 4052000 bytes
**** Totaly Sent 4052632 bytes
DEBUG: 10.0.0.2:12345 switch state, from ESTABLISHED to FIN_WAIT_1.
DEBUG: 10.0.0.2:12345 switch state, from FIN_WAIT_1 to FIN_WAIT_2.
DEBUG: 10.0.0.2:12345 switch state, from FIN_WAIT_2 to TIME_WAIT.
DEBUG: 10.0.0.2:12345 switch state, from TIME_WAIT to CLOSED.
```

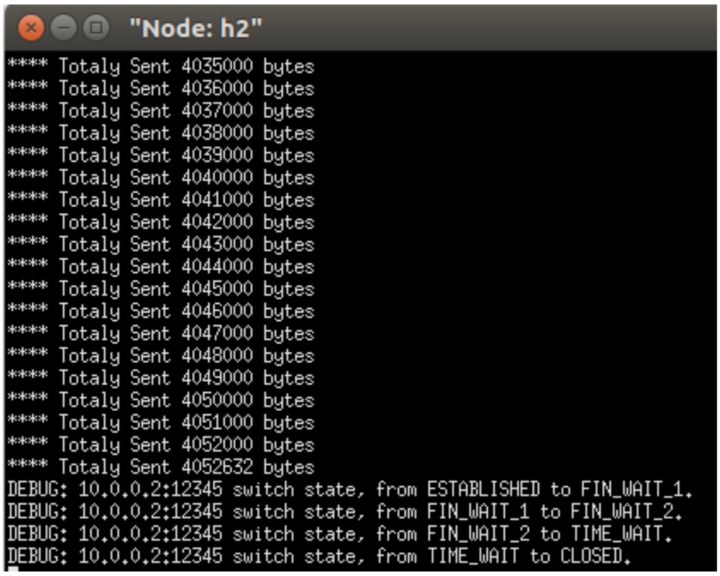
h2 output

2. **h1为tcp_stack.py程序，h2为tcp_stack.c程序**



```
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack_randfile# python tcp_stack.py server 10001
('10.0.0.2', 12345)
^CTraceback (most recent call last):
  File "tcp_stack.py", line 51, in <module>
    server(sys.argv[2])
  File "tcp_stack.py", line 22, in server
    data = cs.recv(1000)
KeyboardInterrupt
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack_randfile# md5sum server-output.dat
9b1f195932281074499a101eb0305370  server-output.dat
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack_randfile# md5sum client-input.dat
9b1f195932281074499a101eb0305370  client-input.dat
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack_randfile# 
```

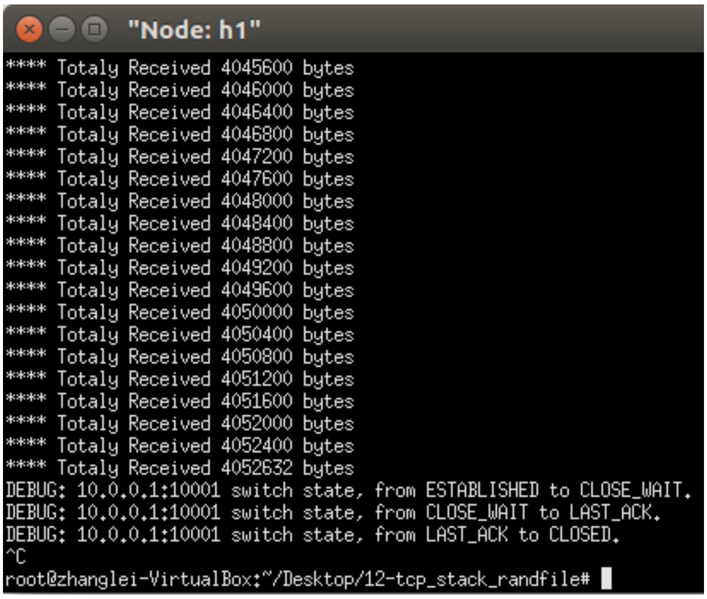
h1 output



```
**** Totaly Sent 4035000 bytes
**** Totaly Sent 4036000 bytes
**** Totaly Sent 4037000 bytes
**** Totaly Sent 4038000 bytes
**** Totaly Sent 4039000 bytes
**** Totaly Sent 4040000 bytes
**** Totaly Sent 4041000 bytes
**** Totaly Sent 4042000 bytes
**** Totaly Sent 4043000 bytes
**** Totaly Sent 4044000 bytes
**** Totaly Sent 4045000 bytes
**** Totaly Sent 4046000 bytes
**** Totaly Sent 4047000 bytes
**** Totaly Sent 4048000 bytes
**** Totaly Sent 4049000 bytes
**** Totaly Sent 4050000 bytes
**** Totaly Sent 4051000 bytes
**** Totaly Sent 4052000 bytes
**** Totaly Sent 4052632 bytes
DEBUG: 10.0.0.2:12345 switch state, from ESTABLISHED to FIN_WAIT_1.
DEBUG: 10.0.0.2:12345 switch state, from FIN_WAIT_1 to FIN_WAIT_2.
DEBUG: 10.0.0.2:12345 switch state, from FIN_WAIT_2 to TIME_WAIT.
DEBUG: 10.0.0.2:12345 switch state, from TIME_WAIT to CLOSED.
```

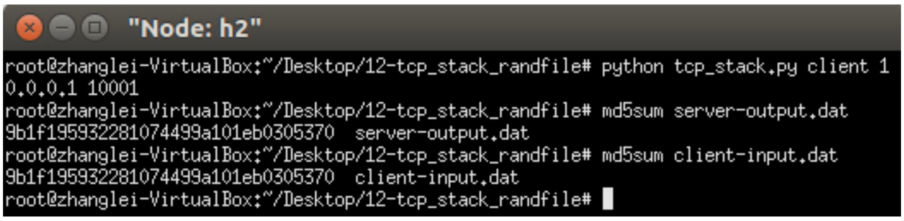
h2 output

3. **h1为tcp_stack.c程序，h2为tcp_stack.py程序**



```
**** Totaly Received 4045600 bytes
**** Totaly Received 4046000 bytes
**** Totaly Received 4046400 bytes
**** Totaly Received 4046800 bytes
**** Totaly Received 4047200 bytes
**** Totaly Received 4047600 bytes
**** Totaly Received 4048000 bytes
**** Totaly Received 4048400 bytes
**** Totaly Received 4048800 bytes
**** Totaly Received 4049200 bytes
**** Totaly Received 4049600 bytes
**** Totaly Received 4050000 bytes
**** Totaly Received 4050400 bytes
**** Totaly Received 4050800 bytes
**** Totaly Received 4051200 bytes
**** Totaly Received 4051600 bytes
**** Totaly Received 4052000 bytes
**** Totaly Received 4052400 bytes
**** Totaly Received 4052632 bytes
DEBUG: 10.0.0.1:10001 switch state, from ESTABLISHED to CLOSE_WAIT.
DEBUG: 10.0.0.1:10001 switch state, from CLOSE_WAIT to LAST_ACK.
DEBUG: 10.0.0.1:10001 switch state, from LAST_ACK to CLOSED.
^C
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack_randfile#
```

h1 output



```
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack_randfile# python tcp_stack.py client 1
0.0.0.1 10001
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack_randfile# md5sum server-output.dat
9b1f19593228107449 9a101eb0305370  server-output.dat
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack_randfile# md5sum client-input.dat
9b1f195932281074499a101eb0305370  client-input.dat
root@zhanglei-VirtualBox:~/Desktop/12-tcp_stack_randfile#
```

h2 output

五、实验分析

实验内容一

1. 观察实验结果，当 h1 和 h2 都使用我们自己编写的 tcp_stack.c 文件时，client 端成功发送数据，server 端成功接收到 client 端发送的数据并 echo 给了 client 端，并成功关闭 TCP 连接，实验成功
2. 观察实验结果，h1 使用 tcp_stack.py 程序，h2 使用 tcp_stack.c 程序，client 端成功发送数据，server 端成功接收到 client 端发送的数据并 echo 给了 client 端，并成功关闭 TCP 连接，实验成功
3. 观察实验结果，h1 使用 tcp_stack.c 程序，h2 使用 tcp_stack.py 程序，client 端成功发送数据，server 端成功接收到 client 端发送的数据并 echo 给了 client 端，并成功关闭 TCP 连接，实验成功
4. 当 h1 使用 tcp_stack.py 程序时，最后需要收到一个空数据包才能退出循环，这里我直接使用 `ctrl+c` 中断程序执行

实验内容二

1. 观察实验结果，当 h1 和 h2 都使用我们自己编写的 tcp_stack.c 文件时，client 端成功发送数据，server 端成功接收到 client 端发送的数据并存储到 server-output.dat 文件中，并成功关闭TCP连接，使用 `md5sum` 程序检测收到的文件和发送的文件，结果相同，证明两个文件完全相同，实验成功
2. 观察实验结果，h1 使用 tcp_stack.py 程序，h2 使用 tcp_stack.c 程序，client 端成功发送数据，server 端成功接收到 client 端发送的数据并存储到 server-output.dat 文件中，并成功关闭 TCP 连接，使用 `md5sum` 程序检测收到的文件和发送的文件，结果相同，证明两个文件完全相同，实验成功
3. 观察实验结果，h1 使用 tcp_stack.c 程序，h2 使用 tcp_stack.py 程序，client 端成功发送数据，server 端成功接收到 client 端发送的数据并存储到 server-output.dat 文件中，并成功关闭 TCP 连接，使用 `md5sum` 程序检测收到的文件和发送的文件，结果相同，证明两个文件完全相同，实验成功
4. 当 h1 和 h2 都使用 tcp_stack.c 文件和 h1 使用 tcp_stack.py 程序，h2 使用 tcp_stack.c 程序时，客户端可以以较快速率发送数据
5. 当 h1 使用 tcp_stack.c 文件，h2 使用 tcp_stack.py 程序时，发送速率较低，且每次最多只能发送 500bytes 的数据，实验中，我使用的是 400bytes ，当一次发送的数据量超过 500bytes 时，或者发送速率过快时会产生丢包，主要现象是server端可能会出现只能读取 456byets 数据的情况

六、反思总结

1. 本次实验代码量较小，需要实验的内容也比较简单，但是还是碰到了一些比较刁钻的 bug，比如在 ring_buffer 结构体中添加了 lock 变量为互斥锁后，忘记了初始化 lock，导致后续的实验中进程卡死，耗费了较多时间；
2. 在实验二中，可能是由于最后的 server 进程没有正确执行到 tcp_close 和 fclose，导致我的最后一部分接收到的数据存留在缓冲区，而没有写入到 server-output.dat 文件中，耗费了我一个下午的时间，最后请老师帮忙查看，在 fwrite 函数后执行 fflush 函数将缓冲区的数据写入到 server-output.dat 中，成功解决了这个问题
3. 在 DEBUG 的时候，由于怀疑 server 程序有问题， 所以在 每次接受时加了一个 sleep 函数，结果导致可能会出现第一个发送的数据包丢失的现象出现

4. 当需要 sleep 小数秒时间的时候，需要采用 usleep 函数，如果采用 sleep 函数，貌似对睡眠时间小于 1 的处理是直接归零了，这也导致我一开始对发送速率的错误控制

七、参考文献

- [1] 中国科学院大学 2020 春计算机网络研讨课 12-网络传输机制实验二实验课件
- [2] 中国科学院大学 2020 春计算机网络研讨课 12-网络传输机制实验二实验附件