

四川大学计算机学院、软件学院

实 验 报 告

学号：2021141430158 姓名：杨融鑫 专业：计算机科学与技术 班级：行政三班 第7周

课程名称	网络编程	实验课时	3 个课时
实验项目	编写支持心跳应答的 Client 程序，并抓包分析	实验时间	3 个小时
实验目的	学会分析三路握手过程； 学会编写 Client 程序发送普通数据消息(从键盘输入)； 学会编写 Client 程序接收并发送心跳应答 学会分析四路挥手过程		
实验环境	Visual Studio Code Python 3.10		

<p>实 验 内 容 (算法、程序、步骤和方法)</p>	<ol style="list-style-type: none"> 1. 编写 Client 程序完成发送普通数据消息功能 2. 利用多线程技术编写心跳应答功能 3. 对所编写的程序进行测试 4. 利用多线程技术编写服务器向客户端发送消息功能 5. 使用线程锁实现临界区的安全读写
<p>(接上) 实验内容 (算法、程序、步骤和方法)</p>	

三路握手过程

tcp.port==12001					
No.	Time	Source	Destination	Protocol	Length Info
370	13.943682	127.0.0.1	127.0.0.1	TCP	56 4324 → 12001 [SYN] Seq=0 Min=65535 Len=0 MSS=65495 WS=256 SACK_PERM
371	13.943738	127.0.0.1	127.0.0.1	TCP	56 12001 → 4324 [SYN, ACK] Seq=0 Ack=1 Min=65535 Len=0 MSS=65495 WS=256 SACK_PERM
372	13.943779	127.0.0.1	127.0.0.1	TCP	44 4324 → 12001 [ACK] Seq=1 Ack=1 Win=2619648 Len=0

客户端发送普通数据消息：

498	18.439598	127.0.0.1	127.0.0.1	TCP	47 4324 → 12001 [PSH, ACK] Seq=14 Ack=10 Win=2619648 Len=3
499	18.439645	127.0.0.1	127.0.0.1		
636	26.527471	127.0.0.1	127.0.0.1		
637	26.527498	127.0.0.1	127.0.0.1		
638	26.527546	127.0.0.1	127.0.0.1		
639	26.527563	127.0.0.1	127.0.0.1		
736	32.847545	127.0.0.1	127.0.0.1		
737	32.847580	127.0.0.1	127.0.0.1		
802	36.528795	127.0.0.1	127.0.0.1		
803	36.528840	127.0.0.1	127.0.0.1		
804	36.528906	127.0.0.1	127.0.0.1		

Frame 498: 47 bytes on wire (376 bits), 47 bytes captured (376 bits) on interface Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 4324, Dst Port: 12001, Seq: 14, Ack: 10, Win: 2619648, Len: 3
Data (3 bytes)

Wireshark - 分组 498 - Adapter for loopback traffic capture

> Frame 498: 47 bytes on wire (376 bits), 47 bytes captured (376 bits) on interface Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 4324, Dst Port: 12001, Seq: 14, Ack: 10, Win: 2619648, Len: 3
> Data (3 bytes)

0000 02 00 00 00 45 00 00 2b d5 9e 40 00 40 06 00 00E+..@...
0010 7f 00 00 01 7f 00 00 01 10 e4 2e e1 69 46 64 d4iFdM
0020 03 ce 96 6b 50 18 27 f9 7e 09 00 00 31 32 33kP...-123

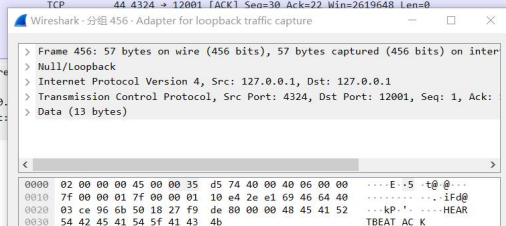
数据记录
和计算

```
ClientServer.py
1 import socket
2 import threading
3 import time
4
5 class HeartbeatSender(threading.Thread):
6     def run(self):
7         while True:
8             message = input("Enter message to send : ")
9             self.send(message)
10
11     def send(self, message):
12         s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
13         s.connect(("127.0.0.1", 12001))
14         s.send(message.encode())
15
16 if __name__ == '__main__':
17     s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
18     s.bind(("127.0.0.1", 12001))
19     s.listen(5)
20     print("Server is listening on port 12001")
21     while True:
22         (c, a) = s.accept()
23         print("Accepted connection from", a)
24         t = HeartbeatSender(c)
25         t.start()
26         c.close()
27
28 # 终端输出
29 C:\Users\杨融鑫\Desktop>python ClientServer.py
30 Connected to server.
31 Enter message to send : 123
32 Enter message to send : 服务器的群发消息为:456
```

Client 接收并发送心跳应答：

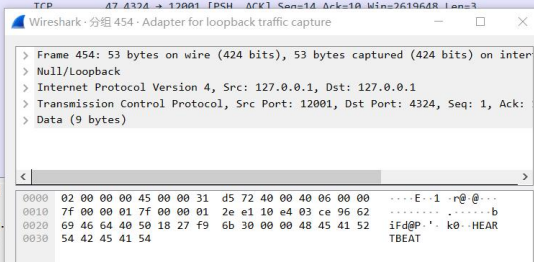
454	16.524988	127.0.0.1	127.0.0.1	TCP	53	12001 → 4324 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=9
455	16.525014	127.0.0.1	127.0.0.1	TCP	44	4324 → 12001 [ACK] Seq=1 Ack=10 Win=2619648 Len=0
456	16.525142	127.0.0.1	127.0.0.1	TCP	57	4324 → 12001 [PSH, ACK] Seq=1 Ack=10 Win=2619648 Len=13
457	16.525159	127.0.0.1	127.0.0.1	TCP	44	12001 → 4324 [ACK] Seq=10 Ack=14 Win=2619648 Len=0
498	18.439598	127.0.0.1	127.0.0.1	TCP	47	4324 → 12001 [PSH, ACK] Seq=14 Ack=10 Win=2619648 Len=3
499	18.439645	127.0.0.1	127.0.0.1	TCP	44	12001 → 4324 [ACK] Seq=10 Ack=17 Win=2619648 Len=0
636	26.527471	127.0.0.1	127.0.0.1	TCP	53	12001 → 4324 [PSH, ACK] Seq=10 Ack=17 Win=2619648 Len=9
637	26.527498	127.0.0.1	127.0.0.1	TCP	44	4324 → 12001 [ACK] Seq=17 Ack=19 Win=2619648 Len=0
638	26.527546	127.0.0.1	127.0.0.1	TCP	57	4324 → 12001 [PSH, ACK] Seq=17 Ack=19 Win=2619648 Len=13
639	26.527563	127.0.0.1	127.0.0.1	TCP	44	12001 → 4324 [ACK] Seq=19 Ack=30 Win=2619648 Len=0
736	32.847545	127.0.0.1	127.0.0.1	TCP	47	12001 → 4324 [PSH, ACK] Seq=19 Ack=30 Win=2619648 Len=3
737	32.847580	127.0.0.1	127.0.0.1	TCP	44	4324 → 12001 [ACK] Seq=30 Ack=22 Win=2619648 Len=0
802	36.528795	127.0.0.1	127.0.0.1			
803	36.528840	127.0.0.1	127.0.0.1			
804	36.528906	127.0.0.1	127.0.0.1			

Frame 456: 57 bytes on wire (456 bits), 57 bytes captured on interface eth0, 0 bytes captured on filter "Null/Loopback"
 Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 Transmission Control Protocol, Src Port: 4324, Dst Port: 12001, Seq: 1, Ack: 10, Win: 2619648, Len: 13 (13 bytes)



372	13.943779	127.0.0.1	127.0.0.1	TCP	44	4324 → 12001 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
454	16.524988	127.0.0.1	127.0.0.1	TCP	53	12001 → 4324 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=9
455	16.525014	127.0.0.1	127.0.0.1	TCP	44	4324 → 12001 [ACK] Seq=1 Ack=10 Win=2619648 Len=0
456	16.525142	127.0.0.1	127.0.0.1	TCP	57	4324 → 12001 [PSH, ACK] Seq=1 Ack=10 Win=2619648 Len=13
457	16.525159	127.0.0.1	127.0.0.1	TCP	44	12001 → 4324 [ACK] Seq=10 Ack=14 Win=2619648 Len=0
498	18.439598	127.0.0.1	127.0.0.1	TCP	47	4324 → 12001 [PSH, ACK] Seq=14 Ack=10 Win=2619648 Len=3
499	18.439645	127.0.0.1	127.0.0.1	TCP	44	12001 → 4324 [ACK] Seq=10 Ack=17 Win=2619648 Len=0
636	26.527471	127.0.0.1	127.0.0.1	TCP	53	12001 → 4324 [PSH, ACK] Seq=10 Ack=17 Win=2619648 Len=9
637	26.527498	127.0.0.1	127.0.0.1	TCP	44	4324 → 12001 [ACK] Seq=17 Ack=19 Win=2619648 Len=0
638	26.527546	127.0.0.1	127.0.0.1	TCP	57	4324 → 12001 [PSH, ACK] Seq=17 Ack=19 Win=2619648 Len=13
639	26.527563	127.0.0.1	127.0.0.1	TCP	44	12001 → 4324 [ACK] Seq=19 Ack=30 Win=2619648 Len=0
736	32.847545	127.0.0.1	127.0.0.1	TCP	47	12001 → 4324 [PSH, ACK] Seq=19 Ack=30 Win=2619648 Len=3
737	32.847580	127.0.0.1	127.0.0.1	TCP	44	4324 → 12001 [ACK] Seq=30 Ack=22 Win=2619648 Len=0
802	36.528795	127.0.0.1	127.0.0.1			
803	36.528840	127.0.0.1	127.0.0.1			
804	36.528906	127.0.0.1	127.0.0.1			

Frame 454: 53 bytes on wire (424 bits), 53 bytes captured on interface eth0, 0 bytes captured on filter "Null/Loopback"
 Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 Transmission Control Protocol, Src Port: 12001, Dst Port: 4324, Seq: 1, Ack: 10, Win: 2619648, Len: 9 (9 bytes)



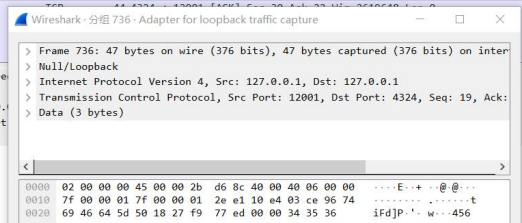
四路挥手过程：

1105	58.882827	127.0.0.1	127.0.0.1	TCP	44	4512 → 12001 [FIN, ACK] Seq=4 Ack=4 Win=2619648 Len=0
1106	58.882881	127.0.0.1	127.0.0.1	TCP	44	12001 → 4512 [ACK] Seq=4 Ack=5 Win=2619648 Len=0
1107	58.882970	127.0.0.1	127.0.0.1	TCP	44	12001 → 4512 [FIN, ACK] Seq=4 Ack=5 Win=2619648 Len=0
1108	58.882909	127.0.0.1	127.0.0.1	TCP	44	4512 → 12001 [ACK] Seq=5 Ack=5 Win=2619648 Len=0

附加功能：服务器向客户端群发消息

639	26.527563	127.0.0.1	127.0.0.1	TCP	44	12001 → 4324 [ACK] Seq=19 Ack=30 Win=2619648 Len=0
736	32.847545	127.0.0.1	127.0.0.1	TCP	47	12001 → 4324 [PSH, ACK] Seq=19 Ack=30 Win=2619648 Len=3
737	32.847580	127.0.0.1	127.0.0.1	TCP	44	4324 → 12001 [ACK] Seq=30 Ack=22 Win=2619648 Len=0
802	36.528795	127.0.0.1	127.0.0.1			
803	36.528840	127.0.0.1	127.0.0.1			
804	36.528906	127.0.0.1	127.0.0.1			

Frame 736: 47 bytes on wire (376 bits), 47 bytes captured on interface eth0, 0 bytes captured on filter "Null/Loopback"
 Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 Transmission Control Protocol, Src Port: 12001, Dst Port: 4324, Seq: 19, Ack: 30, Win: 2619648, Len: 3 (3 bytes)



	<div><div>ClientServer.py</div><div>Debug HeartbeatCheck! ▾</div><div>C: > Users > 杨融鑫 > Desktop > 网络编程 > ClientServer.py > ...</div><div><div>1 import socket</div><div>2 import threading</div><div>3 import time</div><div>4</div><div>5 class HeartbeatSender(threading.Thread):</div></div><div><div>问题</div><div>输出</div><div>调试控制台</div><div>终端</div><div>JUPYTER</div></div><div>版权所有 (C) Microsoft Corporation。保留所有权利。</div><div>尝试新的跨平台 PowerShell https://aka.ms/pscore6</div><div>PS C:\Users\杨融鑫\Desktop\html_文件> & 'C:\Users\杨融鑫\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\杨融鑫\.vscode\extensions\ms-python.python-2022.18.2\pythonFiles\lib\python\debug\adapter/../../debugpy\launcher' '4302' '--' 'c:\Users\杨融鑫\Documents\Tencent Files\1379718091\FileRecv\HeartbeatCheckServer.py'</div><div>Listening on 127.0.0.1:12001</div><div>Enter message to send: Received from ('127.0.0.1', 4324): 123</div><div>456</div></div>
结 论 (结 果)	<div>三路握手过程：</div> <div>客户端向服务器发送 SYN 包（SYN=1，seq=x）。</div> <div>服务器接收到 SYN 包，向客户端发送 SYN-ACK 包（SYN=1，ACK=1，seq=y，ack=x+1）。</div> <div>客户端接收到 SYN-ACK 包，向服务器发送 ACK 包（ACK=1，seq=x+1，</div>

	<p>ack=y+1)，建立连接。</p> <p>客户端发送普通数据消息：</p> <p>客户端从键盘输入消息，并将消息发送给服务器。</p> <p>Client 接收并发送心跳应答：</p> <p>客户端在一个单独的线程中监听来自服务器的心跳消息。</p> <p>如果收到心跳消息，则向服务器发送心跳应答。</p> <p>四路挥手过程：</p> <p>客户端向服务器发送 FIN 包（FIN=1，seq=4）。</p> <p>服务器接收到 FIN 包后，向客户端发送 ACK 包（ACK=1，seq=4，ack=5）。</p> <p>服务器关闭写入方向的连接，向客户端发送 FIN 包（FIN=1，seq=4，ack=5）。</p> <p>客户端接收到服务器的 FIN 包，向服务器发送 ACK 包（ACK=1，seq=5，ack=5），关闭连接。</p> <p>附加功能：服务器向客户端群发消息</p> <p>服务器从键盘输入消息，并将消息发送给客户端。</p>
小 结	<p>通过这次实验，我们深入了解了 TCP 中 socket 连接的建立和关闭过程，以及在网络编程中实现心跳应答的重要性。同时，通过抓包分析，加深了对网络通信过程中数据包交换的理解，为我们后续的网络编程开发提供了有益的经验 and 知识，也了解到实现一个可以完成基本功能的网络程序不难，但是想要做到性能高，正常退出，还是很有困难的，特别是这次困扰了我很久</p>

图（实验装置的结构示意图），再配以相应的文字说明，这样既可以节省许多文字说明，又能使实验报告简明扼要，清楚明白。

数据记录和计算 指从实验中测出的数据以及计算结果。

结论（结果） 即根据实验过程中所见到的现象和测得的数据，作出结论。

小结 对本次实验的体会、思考和建议。

备注或说明 可写上实验成功或失败的原因，实验后的心得体会、建议等。

注意：

- 实验报告将记入实验成绩；
- 每次实验开始时，交上一次的实验报告，否则将扣除此次实验成绩。