��TCP����

�������������������������

��������tcp�������wireshark���TCP�������������������
“Hello World�”�����������������������������

������

��go1.8��

����

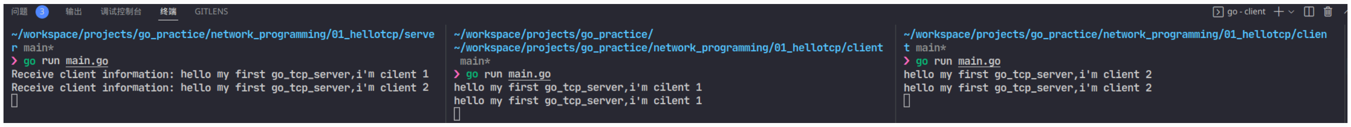
```
//tcp client
func main()  {
    conn ,err := net.Dial("tcp","127.0.0.1:8888")
    if err != nil {
        fmt.Println("connected failed, error message:",err)
        return
    }
    defer conn.Close()
    inputReader := bufio.NewReader(os.Stdout)
    for {
        input, _ := inputReader.ReadString('\n')    //������
        inputInfo := strings.Trim(input,"\r\n")
        if strings.ToUpper(inputInfo) == "q"{
            return  //����q���
        }
        _,err = conn.Write([]byte(inputInfo))   //����
        if err != nil{
            return
        }
        buf := [512]byte{}
        n,err := conn.Read(buf[:])
        if err != nil{
            fmt.Println("get information failed, error message",err)
            return
        }
        fmt.Println(string(buf[:n]))
    }
}

//TCP server�
func process(conn net.Conn) {
 defer conn.Close() //����
 for {
  reader := bufio.NewReader(conn)
  var buf [128]byte
  n, err := reader.Read(buf[:]) //����
  if err != nil {
   fmt.Println("Failed to connect to client, error message:", err)
  }
  recvStr := string(buf[:n])
  fmt.Println("Receive client information:", recvStr)
  conn.Write([]byte(recvStr)) //����
 }
}

func main() {
 listen, err := net.Listen("tcp", "127.0.0.1:8888")
 if err != nil {
  fmt.Println("Listen failed error message:", err)
  return
 }
 for {
  conn, err := listen.Accept() //����
```

```
   if err != nil {
    fmt.Println("Establishing connection failed, error message:", err)
    continue
   }
   go process(conn) //����goroutine����
  }
 }
```

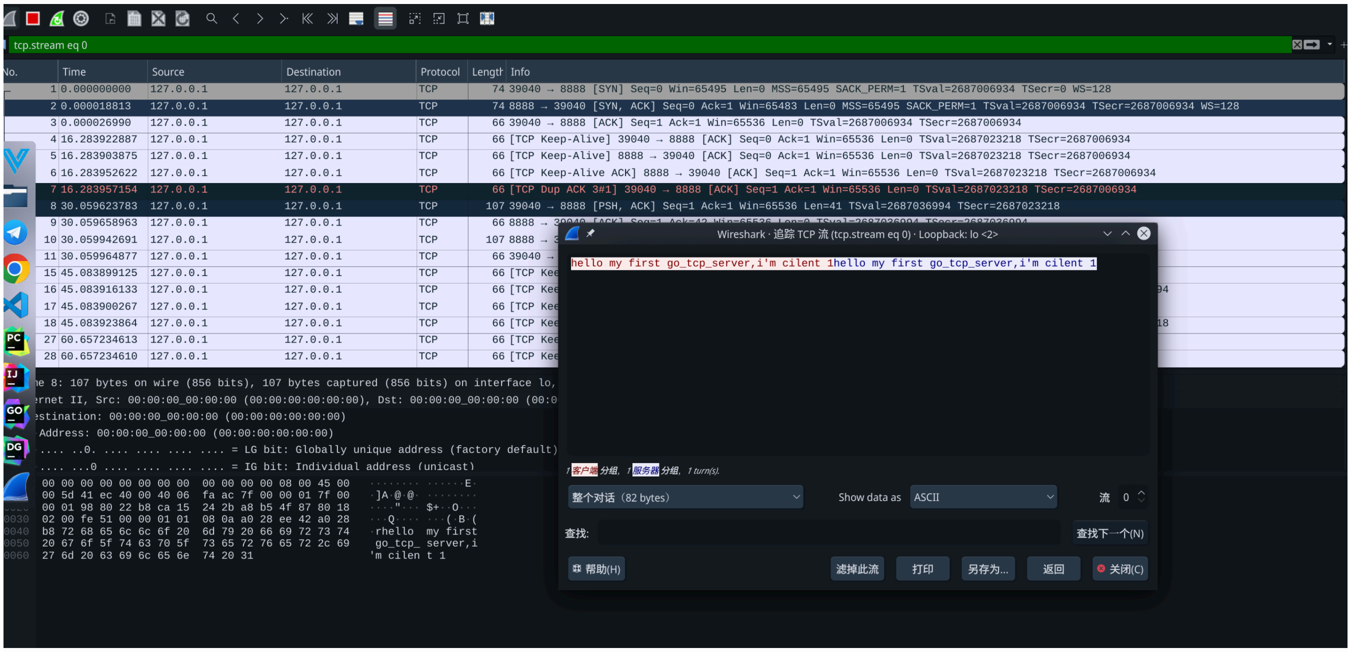
����



```
~/workspace/projects/go_practice/network_programming/01_hellotcp/serve
r main*
❯ go run main.go
Receive client information: hello my first go_tcp_server,i'm cilent 1
Receive client information: hello my first go_tcp_server,i'm client 2
▯
```

```
~/workspace/projects/go_practice/
~/workspace/projects/go_practice/network_programming/01_hellotcp/client
 main*
❯ go run main.go
hello my first go_tcp_server,i'm cilent 1
hello my first go_tcp_server,i'm cilent 1
▯
```

```
~/workspace/projects/go_practice/network_programming/01_hellotcp/clien
t main*
❯ go run main.go
hello my first go_tcp_server,i'm client 2
hello my first go_tcp_server,i'm client 2
▯
```



tcp.stream eq 0

```
No.  Time          Source     Destination  Protocol Length Info
   1 0.000000000   127.0.0.1  127.0.0.1    TCP       74 39040 → 8888 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2687006934 TSecr=0 WS=128
   2 0.000018813   127.0.0.1  127.0.0.1    TCP       74 8888 → 39040 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=2687006934 TSecr=2687006934 WS=128
   3 0.000026990   127.0.0.1  127.0.0.1    TCP       66 39040 → 8888 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2687006934 TSecr=2687006934
   4 16.283922887  127.0.0.1  127.0.0.1    TCP       66 [TCP Keep-Alive] 39040 → 8888 [ACK] Seq=0 Ack=1 Win=65536 Len=0 TSval=2687023218 TSecr=2687006934
   5 16.283903875  127.0.0.1  127.0.0.1    TCP       66 [TCP Keep-Alive] 8888 → 39040 [ACK] Seq=0 Ack=1 Win=65536 Len=0 TSval=2687023218 TSecr=2687006934
   6 16.283952622  127.0.0.1  127.0.0.1    TCP       66 [TCP Keep-Alive ACK] 8888 → 39040 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2687023218 TSecr=2687006934
   7 16.283957154  127.0.0.1  127.0.0.1    TCP       66 [TCP Dup ACK 3#1] 39040 → 8888 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2687023218 TSecr=2687006934
   8 30.059623783  127.0.0.1  127.0.0.1    TCP      107 39040 → 8888 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=41 TSval=2687036994 TSecr=2687023218
   9 30.059658963  127.0.0.1  127.0.0.1    TCP       66 8888 → 39040 [ACK] Seq=1 Ack=42 Win=65536 Len=0 TSval=2687036994 TSecr=2687036994
  10 30.059942691  127.0.0.1  127.0.0.1    TCP      107 8888 → 3
  11 30.059964877  127.0.0.1  127.0.0.1    TCP       66 39040 →
  15 45.083899125  127.0.0.1  127.0.0.1    TCP       66 [TCP Kee
  16 45.083916133  127.0.0.1  127.0.0.1    TCP       66 [TCP Kee                                                      94
  17 45.083900267  127.0.0.1  127.0.0.1    TCP       66 [TCP Kee
  18 45.083923864  127.0.0.1  127.0.0.1    TCP       66 [TCP Kee                                                      18
  27 60.657234613  127.0.0.1  127.0.0.1    TCP       66 [TCP Kee
  28 60.657234610  127.0.0.1  127.0.0.1    TCP       66 [TCP Kee
```

```
ne 8: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface lo,
ernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:0
estination: 00:00:00_00:00:00 (00:00:00:00:00:00)
 Address: 00:00:00_00:00:00 (00:00:00:00:00:00)
 .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
 .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
```

```
     00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ········ ······E·
     00 5d 41 ec 40 00 40 06  fa ac 7f 00 00 01 7f 00   ·]A·@·@· ········
     00 01 98 80 22 b8 ca 15  24 2b a8 b5 4f 87 80 18   ····"··· $+··O···
0030 02 00 fe 51 00 00 01 01  08 0a a0 28 ee 42 a0 28   ···Q···· ···(·B·(
0040 b8 72 68 65 6c 6c 6f 20  6d 79 20 66 69 72 73 74   ·rhello  my first
0050 20 67 6f 5f 74 63 70 5f  73 65 72 76 65 72 2c 69    go_tcp_ server,i
0060 27 6d 20 63 69 6c 65 6e  74 20 31                  'm cilen t 1
```

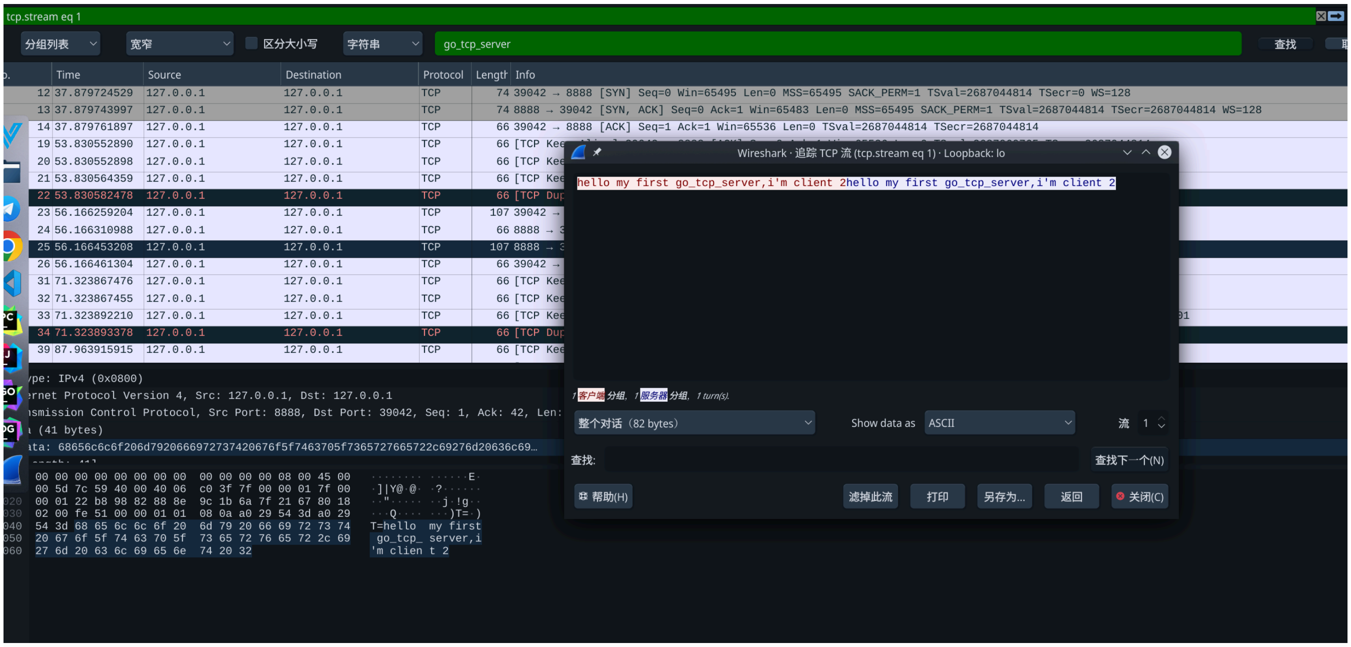
Wireshark · 追踪 TCP 流 (tcp.stream eq 0) · Loopback: lo <2>

```
hello my first go_tcp_server,i'm cilent 1hello my first go_tcp_server,i'm cilent 1
```

1 客户端 分组, 1 服务器 分组, 1 turn(s).

整个对话（82 bytes） Show data as ASCII 流 0

查找:



```
o.    Time          Source       Destination  Protocol Length Info
   12 37.879724529  127.0.0.1    127.0.0.1    TCP        74 39042 → 8888 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=2687044814 TSecr=0 WS=128
   13 37.879743997  127.0.0.1    127.0.0.1    TCP        74 8888 → 39042 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=2687044814 TSecr=2687044814 WS=128
   14 37.879761897  127.0.0.1    127.0.0.1    TCP        66 39042 → 8888 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=2687044814 TSecr=2687044814
   19 53.830552890  127.0.0.1    127.0.0.1    TCP        66 [TCP Kee
   20 53.830552898  127.0.0.1    127.0.0.1    TCP        66 [TCP Kee
   21 53.830564359  127.0.0.1    127.0.0.1    TCP        66 [TCP Kee
   22 53.830582478  127.0.0.1    127.0.0.1    TCP        66 [TCP Dup
   23 56.166259204  127.0.0.1    127.0.0.1    TCP       107 39042 →
   24 56.166310988  127.0.0.1    127.0.0.1    TCP        66 8888 → 3
   25 56.166453208  127.0.0.1    127.0.0.1    TCP       107 8888 → 3
   26 56.166461304  127.0.0.1    127.0.0.1    TCP        66 39042 →
   31 71.323867476  127.0.0.1    127.0.0.1    TCP        66 [TCP Kee
   32 71.323867455  127.0.0.1    127.0.0.1    TCP        66 [TCP Kee
   33 71.323892210  127.0.0.1    127.0.0.1    TCP        66 [TCP Kee                                                          01
   34 71.323893378  127.0.0.1    127.0.0.1    TCP        66 [TCP Dup
   39 87.963915915  127.0.0.1    127.0.0.1    TCP        66 [TCP Kee
```

```
ype: IPv4 (0x0800)
ernet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
nsmission Control Protocol, Src Port: 8888, Dst Port: 39042, Seq: 1, Ack: 42, Len:
a (41 bytes)
ata: 68656c6c6f206d7920666972737420676f5f7463705f7365727665722c69276d20636c69…
```

```
     00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ········ ······E·
     00 5d 7c 59 40 00 40 06  c0 3f 7f 00 00 01 7f 00   ·]|Y@·@· ·?······
020  00 01 22 b8 98 82 88 8e  9c 1b 6a 7f 21 67 80 18   ··"····· ··j·!g··
030  02 00 fe 51 00 00 01 01  08 0a a0 29 54 3d a0 29   ···Q···· ···)T=·)
040  54 3d 68 65 6c 6c 6f 20  6d 79 20 66 69 72 73 74   T=hello  my first
050  20 67 6f 5f 74 63 70 5f  73 65 72 76 65 72 2c 69    go_tcp_ server,i
060  27 6d 20 63 6c 69 65 6e  74 20 32                  'm clien t 2
```

```
hello my first go_tcp_server,i'm client 2hello my first go_tcp_server,i'm client 2
```

1 客户端 分组, 1 服务器 分组, 1 turn(s).

整个对话（82 bytes）　Show data as ASCII　流 1

查找:　查找下一个(N)