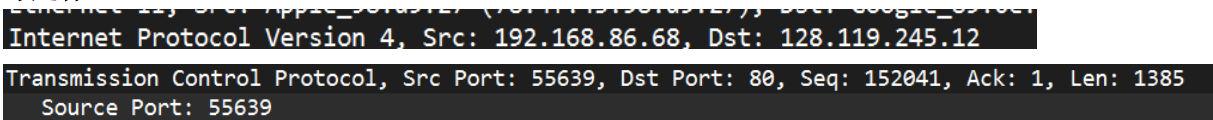


计算机学院 计算机网络 课程实验报告

实验题目：TCP 协议		学号：202200400053
日期：2024-04-12	班级： 2 班	姓名： 王宇涵
Email：1941497679@qq.com		
<p>实验方法介绍：</p> <p>使用 wireShark 进行抓包，并进行数据包的分析，理解 TCP 协议的组成和数据包字段代表的含义。</p>		
<p>实验过程描述：</p> <p>一、捕获从计算机到远程服务器的批量 TCP 传输，并查看捕获的轨迹</p> <p>回答下列问题</p> <p>1. 将 alice.txt 文件传输到 gaia.cs.umass.edu 的客户端计算机（源）使用的 IP 地址和 TCP 端口号是什么？</p> <div><p>Internet Protocol Version 4, Src: 192.168.86.68, Dst: 128.119.245.12</p><p>Transmission Control Protocol, Src Port: 55639, Dst Port: 80, Seq: 152041, Ack: 1, Len: 1385</p><p>Source Port: 55639</p></div> <p>答：IP 地址：192.168.86.68 TCP 端口号：55639</p> <p>2. gaia.cs.umass.edu 的 IP 地址是多少？它在哪个端口号上发送和接收此连接的 TCP 段？</p> <p>答：IP 地址：128.119.245.12 TCP 端口号：80</p> <p>二、TCP 基础</p> <p>3. 什么是用于启动客户端计算机和 gaia.cs.umass.edu 之间 TCP 连接的 TCP SYN 段的序列号？</p> <p>这个 TCP 报文段中的什么内容将该报文段标识为 SYN 报文段？此会话中的 TCP 接收方是否足够使用选择性确认？</p>		

```

Source Port: 55639
Destination Port: 80
[Stream index: 0]
▶ [Conversation completeness: Incomplete, DATA (15)]
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 4236649187
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
1011 .... = Header Length: 44 bytes (11)
▶ Flags: 0x002 (SYN)
Window: 65535
[Calculated window size: 65535]
Checksum: 0xa1e4 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0

```

▶ TCP Option - SACK permitted

答：序列号为 0, Flags(0x002)会标记 SYN 为 1, 可以使用选择性确认.

4. 什么是序列号 gaia.cs.umass.edu 发送到客户端计算机以回复 SYN 的 SYNACK 段的长度? 段中是什么将段标识为 SYNACK 段? SYNACK 段中的 ACK 字段的值是多少? gaia.cs.umass.edu 如何确定该值?

```

Sequence Number: 0 (relative sequence number)

```

```

1010 .... = Header Length:
Flags: 0x012 (SYN, ACK)

```

```

Acknowledgment Number: 1 (relative ack number)

```

```

Flags: 0x012 (SYN, ACK)
000. .... = Reserved: Not set
...0 .... = Accurate ECN: Not set
.... 0... = Congestion Window Reduced: Not set
.... .0.. = ECN-Echo: Not set
.... ..0. = Urgent: Not set
.... ...1 = Acknowledgment: Set
.... ....0... = Push: Not set
.... .... .0.. = Reset: Not set
▼ .... .... ..1. = Syn: Set

```

答：序列号为 0, 将 Flags 段标识为 SYN, ACK, ACK 字段的值为 1, 通过将 Flags 设为(0x012)确定这个值. 即 Syn 及 Acknowledgment 位设为 1, 1

5. 包含 HTTP POST 命令头的 TCP 段的序列号是多少?

该 TCP 报文段的有效负载(数据)字段中包含多少字节的数据? 传输的文件 alice.txt 中的所有数据是否都适合这个单独的段?

```

153 0.147682 192.168.86.68 128.119.245.12 HTTP 1451 POST /wireshark-labs/lab3-1-reply.htm HTTP/1.1 (text/plain)

```

```

Sequence Number: 152041 (relative sequence number)
Sequence Number (raw): 4236801228

```

```

TCP payload (1385 bytes)

```

答：TCP 序列号为 152041, 有效负载中有 1385bytes 数据, alice.txt 中所有的数据都适合这个单独的段.

6. 将包含 HTTP“POST”的 TCP 段视为 TCP 连接数据传输部分的第一个段。

TCP 连接数据传输部分中的第一个数据段（包含 HTTP POST 的数据段）何时发送？

第一个包含数据的段的 ACK 是在什么时间收到的？

第一个包含数据的分段的 RTT 是多少？

第二个承载数据的 TCP 段及其 ACK 的 RTT 值是多少？

是什么估计 RTT 收到第二个数据承载段的 ACK 后的值（参见正文中的第 3.5.3 节）？

假设在收到第二段的 ACK 后进行此计算，

4	0.024047	192.168.86.68	128.119.245.12	TCP	1514	55639 → 80 [ACK] Seq=1 Ack=1 Win=131712 Len=1448 TSval=725607532 TSecr=3913851370 [TCP segment of a reassembled PDU]
7	0.052671	128.119.245.12	192.168.86.68	TCP	66	80 → 55639 [ACK] Seq=1 Ack=1449 Win=31872 Len=0 TSval=3913851399 TSecr=725607532
5	0.024048	192.168.86.68	128.119.245.12	TCP	1514	55639 → 80 [ACK] Seq=1449 Ack=1 Win=131712 Len=1448 TSval=725607532 TSecr=3913851370 [TCP segment of a reassembled PDU]
8	0.052676	128.119.245.12	192.168.86.68	TCP	66	80 → 55639 [ACK] Seq=1 Ack=2897 Win=34816 Len=0 TSval=3913851400 TSecr=725607532

```
[SEQ/ACK analysis]
[This is an ACK to the segment in frame: 4]
[The RTT to ACK the segment was: 0.028624000 seconds]
[iRTT: 0.022505000 seconds]
```

```
[SEQ/ACK analysis]
[This is an ACK to the segment in frame: 5]
[The RTT to ACK the segment was: 0.028628000 seconds]
[iRTT: 0.022505000 seconds]
```

答：第一个数据段于 0.024047 时发送。第一个包含数据的段 ACK 收到的时间为 0.052671。RTT 为 $0.052671 - 0.024047 = 0.028624$ ；第二个承载数据的 TCP 段 RTT 为 $0.052676 - 0.024048 = 0.028628$

计算： $\text{EstimatedRTT} = (1 - \alpha) * \text{EstimatedRTT} + \alpha * \text{SampleRTT} = (1 - 0.125) * 0.028624 + 0.125 * 0.028628 = 0.02864025$

4	0.024047	192.168.86.68	128.119.245.12	TCP	1514	55639 → 80 [ACK] Seq=1 Ack=1 Win=131712 Len=1448 TSval=725607532 TSecr=3913851370 [TCP segment of a reassembled PDU]
5	0.024048	192.168.86.68	128.119.245.12	TCP	1514	55639 → 80 [ACK] Seq=1449 Ack=1 Win=131712 Len=1448 TSval=725607532 TSecr=3913851370 [TCP segment of a reassembled PDU]
6	0.024049	192.168.86.68	128.119.245.12	TCP	1514	55639 → 80 [ACK] Seq=2339 Ack=1 Win=131712 Len=1448 TSval=725607532 TSecr=3913851370 [TCP segment of a reassembled PDU]
7	0.052671	128.119.245.12	192.168.86.68	TCP	66	80 → 55639 [ACK] Seq=1 Ack=1449 Win=31872 Len=0 TSval=3913851399 TSecr=725607532
8	0.052676	128.119.245.12	192.168.86.68	TCP	66	80 → 55639 [ACK] Seq=1 Ack=2897 Win=34816 Len=0 TSval=3913851400 TSecr=725607532
9	0.052774	192.168.86.68	128.119.245.12	TCP	1514	55639 → 80 [ACK] Seq=4345 Ack=1 Win=131712 Len=1448 TSval=725607560 TSecr=3913851399 [TCP segment of a reassembled PDU]

7. 前四个承载数据的 TCP 段中每个段的长度（标头加有效负载）是多少？

答：均为 1514

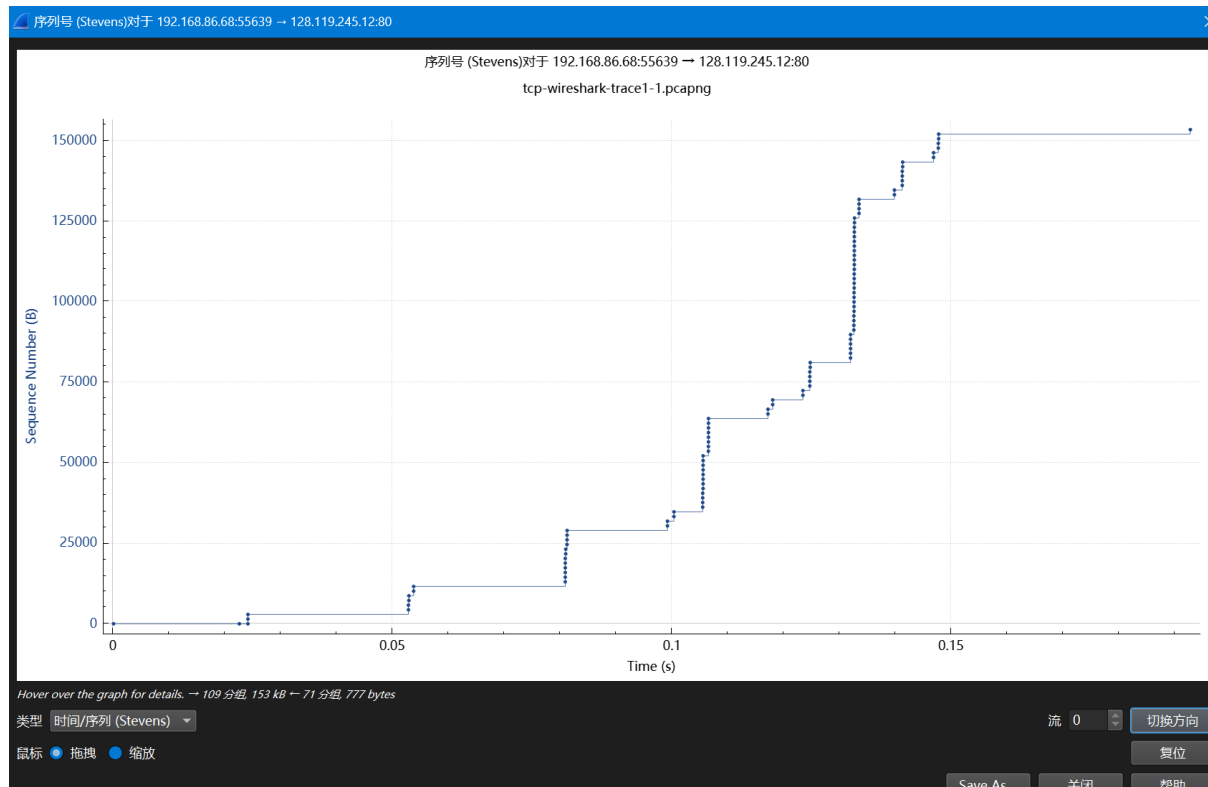
8. gaia.cs.umass.edu 在前四个数据承载 TCP 段中向客户端通告的最小可用缓冲

区空间是多少？接收方缓冲区空间的缺乏是否曾限制前四个数据传输段的发送方？

答：Win 均为 131712。缓存空间充足，不会使发送方受限制。

9. 跟踪文件中是否有重传的段？为了回答这个问题，您（在跟踪中）检查了什

么？



答：绘制 stevens 图发现，序号随着时间单调不递减，因此没有出现重传的段。

10. 在从客户端发送到 gaia.cs.umass.edu 的前 10 个数据承载段中，接收方通常会在 ACK 中确认多少数据？您能否识别出接收方在这前 10 个数据承载段中每隔一个接收到的段（参见正文中的表 3.2）进行 ACK 的情况？

Len=1448	
7 0.052671	128.119.245.12 192.168.86.68 TCP 66 80 -> 55639 [ACK] Seq=1 Ack=1449 Win=31872 Len=0 TSval=3913851399 TSecr=725607532
8 0.052676	128.119.245.12 192.168.86.68 TCP 66 80 -> 55639 [ACK] Seq=1 Ack=2897 Win=34816 Len=0 TSval=3913851400 TSecr=725607532
13 0.053626	128.119.245.12 192.168.86.68 TCP 66 80 -> 55639 [ACK] Seq=1 Ack=4345 Win=37760 Len=0 TSval=3913851400 TSecr=725607532
16 0.080768	128.119.245.12 192.168.86.68 TCP 66 80 -> 55639 [ACK] Seq=1 Ack=5793 Win=40576 Len=0 TSval=3913851421 TSecr=725607560
17 0.080771	128.119.245.12 192.168.86.68 TCP 66 80 -> 55639 [ACK] Seq=1 Ack=7241 Win=43520 Len=0 TSval=3913851422 TSecr=725607560
18 0.080772	128.119.245.12 192.168.86.68 TCP 66 80 -> 55639 [ACK] Seq=1 Ack=8689 Win=46336 Len=0 TSval=3913851422 TSecr=725607560
19 0.080772	128.119.245.12 192.168.86.68 TCP 66 80 -> 55639 [ACK] Seq=1 Ack=10137 Win=49280 Len=0 TSval=3913851422 TSecr=725607560
28 0.081120	128.119.245.12 192.168.86.68 TCP 66 80 -> 55639 [ACK] Seq=1 Ack=11585 Win=52224 Len=0 TSval=3913851429 TSecr=725607560
29 0.081123	128.119.245.12 192.168.86.68 TCP 66 80 -> 55639 [ACK] Seq=1 Ack=13033 Win=55040 Len=0 TSval=3913851429 TSecr=725607560
34 0.099073	128.119.245.12 192.168.86.68 TCP 66 80 -> 55639 [ACK] Seq=1 Ack=14481 Win=57984 Len=0 TSval=3913851449 TSecr=725607588
37 0.100226	128.119.245.12 192.168.86.68 TCP 66 80 -> 55639 [ACK] Seq=1 Ack=15929 Win=60928 Len=0 TSval=3913851449 TSecr=725607588
40 0.105364	128.119.245.12 192.168.86.68 TCP 66 80 -> 55639 [ACK] Seq=1 Ack=18825 Win=66688 Len=0 TSval=3913851454 TSecr=725607588
41 0.105368	128.119.245.12 192.168.86.68 TCP 66 80 -> 55639 [ACK] Seq=1 Ack=21721 Win=72448 Len=0 TSval=3913851455 TSecr=725607588
42 0.105369	128.119.245.12 192.168.86.68 TCP 66 80 -> 55639 [ACK] Seq=1 Ack=23169 Win=75392 Len=0 TSval=3913851455 TSecr=725607588
43 0.105369	128.119.245.12 192.168.86.68 TCP 66 80 -> 55639 [ACK] Seq=1 Ack=24617 Win=78208 Len=0 TSval=3913851456 TSecr=725607588
56 0.106412	128.119.245.12 192.168.86.68 TCP 66 80 -> 55639 [ACK] Seq=1 Ack=26065 Win=81152 Len=0 TSval=3913851456 TSecr=725607588
57 0.106415	128.119.245.12 192.168.86.68 TCP 66 80 -> 55639 [ACK] Seq=1 Ack=28961 Win=86912 Len=0 TSval=3913851456 TSecr=725607588

答：接收方通常会在 ACK 中确认 1448 个数据，前十个段中没有看到每隔一个接受到的段进行 ACK 的情况。

11. TCP 连接的吞吐量（单位时间传输的字节数）是多少？解释一下你是如何计算这个值的。

第一个数据包发送

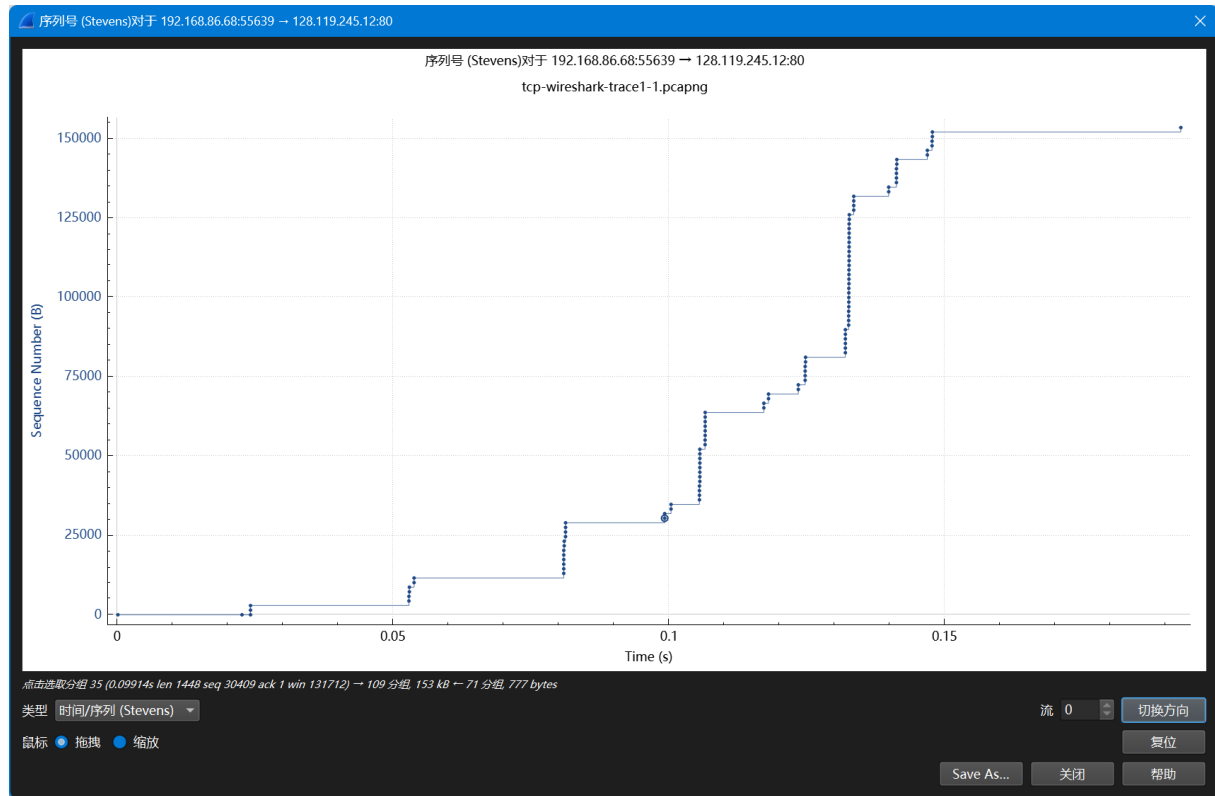
4 0.024847	192.168.86.68	128.119.245.12	TCP	1514 55639 -> 80 [ACK] Seq=1 Ack=1 Win=131712 Len=1448 TSval=725607532 TSecr=3913851370 [TCP segment of a reassembled PDU]
------------	---------------	----------------	-----	--

最后一个数据包发送

180 0.192732 192.168.86.68 128.119.245.12 TCP 66 55639 → 80 [ACK] Seq=153426 Ack=778 Win=130944 Len=0 TSval=725607691 TSecr=3913851542

答：总时间为最后一个数据包发送时间减去第一个数据包发送时间，总数据量为 153426 bytes
则吞吐量为 $153426 * 8 / (0.192732 - 0.024047) = 7276331.62$ bps

三、TCP 拥塞控制



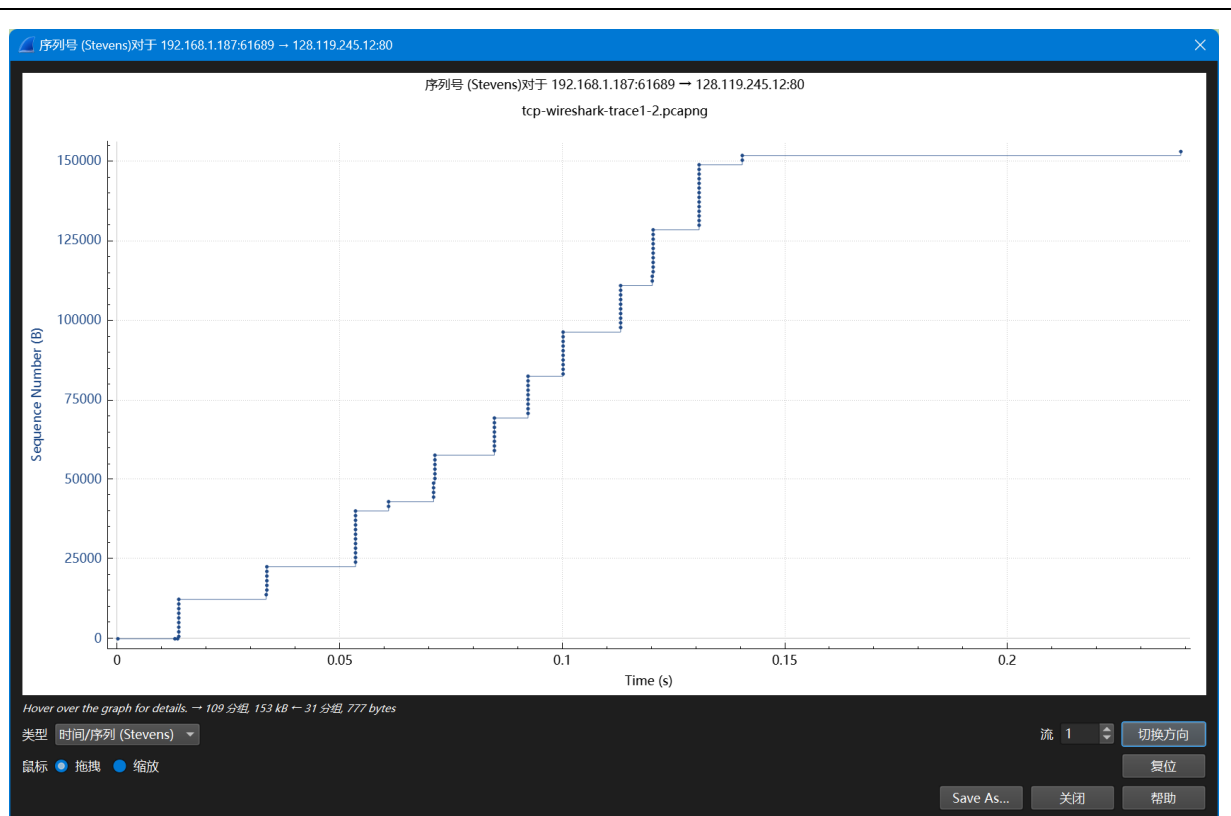
12. 使用时序图（Stevens）绘图工具，用于查看从客户端发送到 gaia.cs.umass.edu 服务器的片段的序列号与时间图。考虑发送的数据包的“舰队” $t=0.025, t=0.053, t=0.082$ 和 $t=0.1$ 。评论一下这看起来 TCP 是否处于慢启动阶段、拥塞避免阶段或其他阶段？

答: $t=0.025$ ：慢启动阶段； $t=0.053$ ：慢启动阶段； $t=0.082$ ：慢启动阶段； $t=0.1$ ：发生了丢包(延迟)，cwnd 置为 1。

13. 这些分段“舰队”似乎具有一定的周期性。对于这段时期你有什么想说的吗？

答：我们可以看出是周期性增长的。

14. 回答上述两个问题，了解您将文件从计算机传输到 gaia.cs.umass.edu 时收集的跟踪信息



答：t = 0 的时候是慢启动开始的时候，但无法明显的看出慢启动阶段的结束和拥塞避免的开始，但也可以看出一定的周期性，并没有明显的增长趋势。

分析：

在这个实验中，我对 TCP 协议的行为进行了详细的研究。

我研究了 TCP 在提供可靠数据传输方面使用的序列号和确认号；

观察了 TCP 的拥塞控制算法——慢启动和拥塞避免——的运作方式；

并了解了 TCP 的接收端流量控制机制。我还简要讨论了 TCP 连接建立，并调查了 TCP 连接之间的性能（吞吐量和往返时间）。

结论：

1. TCP 通过使用序列号和确认号来实现可靠的数据传输。序列号用于对发送的数据进行编号，确认号用于确认已经接收到的数据。
2. TCP 的拥塞控制算法包括慢启动和拥塞避免。慢启动阶段中，TCP 发送方以指数增长的速度增加拥塞窗口大小；在拥塞避免阶段中，TCP 发送方以线性增长的速度增加拥塞窗口大小，以避免网络拥塞。
3. TCP 的接收端流量控制机制通过接收窗口大小来限制发送方发送的数据量，确保接收端能够处理数据。
4. TCP 连接的性能受到吞吐量和往返时间的影响。通过调整 TCP 参数和网络条件，可以优化 TCP 连接的性能。