TCP

2022年11月29日　　13:33

transmission control protocol

==========

https://zhuanlan.zhihu.com/p/430799766
https://juejin.cn/post/7028003193502040072

3次握手，4次挥手

TCP传输分为3个阶段

1. 建立连接
2. 数据传输
3. 释放连接

TCP头部格式

源端口，16bit，源端口号，用来识别 发送 该TCP报文段的 应用进程。

目的端口，16bit，目的端口号，用来识别 接受 该TCP报文段的 应用进程。

序号，32bit，取值范围[0,2^32-1]，序号增加到最后一个后，下一个序号又回到0。指出 本TCP报文段 数据载荷 的第一个字节的 序号。

确认号，32bit，uint，最大+1变成0。 指出 期望 收到对方下一个TCP 报文段的 数据载荷的第一个字节的 序号，同时 也是对 之前收到的 所有数据的确认。如果确认号是 n，那么说明 <= n-1 的数据都已经被正确接收，希望获得 序号为 n 的数据。

确认标志位ACK，取值为1 时， 确认号 字段才有效。TCP规定，在连接建立后，所有传送的TCP报文段 都应该将 ACK 设置为 1

数据偏移，占4bit，并以4字节为单位。用来指出 TCP 报文段的 数据载荷部分的 起始处 距离 TCP 报文段的 起始处有多远。 这个字段实际上 是指出 TCP报文段的 首部长度。

窗口，16bit，以字节为单位。指出发送 本报文段的 一方的 接收窗。

同步标志位SYN，在TCP连接 建立时 用来同步序号。

终止标志位FIN：用来释放TCP连接。

复位标志位RST：用来复位TCP连接。

推送标志位PSH：接收方的 TCP 收到 该标志位为1 的报文 会 尽快上交应用程序，而不必等到接受缓存 都填满后 再向上交付。

校验和：16bit，检查范围 包括 TCP 报文段的 首部 和 数据载荷 2部分。在计算校验和时，要在 TCP 报文段的 前面加上 12字节的 伪首部。

紧急指针：16bit，以字节为单位，用来指明 紧急数据的长度。

填充，由于 选项的 长度可变，因此使用 填充来 确保报文段首部 能被4整除（因为 数据偏移 字段， 也就是 首部长度字段，是以 4字节 为单位的）

TCP的连接建立
建立连接的过程叫做握手，需要在 客户端 和 服务器 之间 交换 3个TCP 报文段，称为 三报文握手。
采用三报文握手，主要是为了防止 已 失效的 连接 请求 报文段 突然 又传送到了，而产生错误。

TCP的连接建立 需要解决以下 3个问题

1. 使 TCP 双方都能 确知 对方的 存在
2. 使 TCP 双方 能够 协商 一些 参数 （ 最大窗口值是否使用窗口扩大选项和时间戳选项，以及 服务质量等）
3. 使 TCP 双方 都能 对 运输实体资源 （例如 缓存大小 连接表中的项目 等） 进行分配。

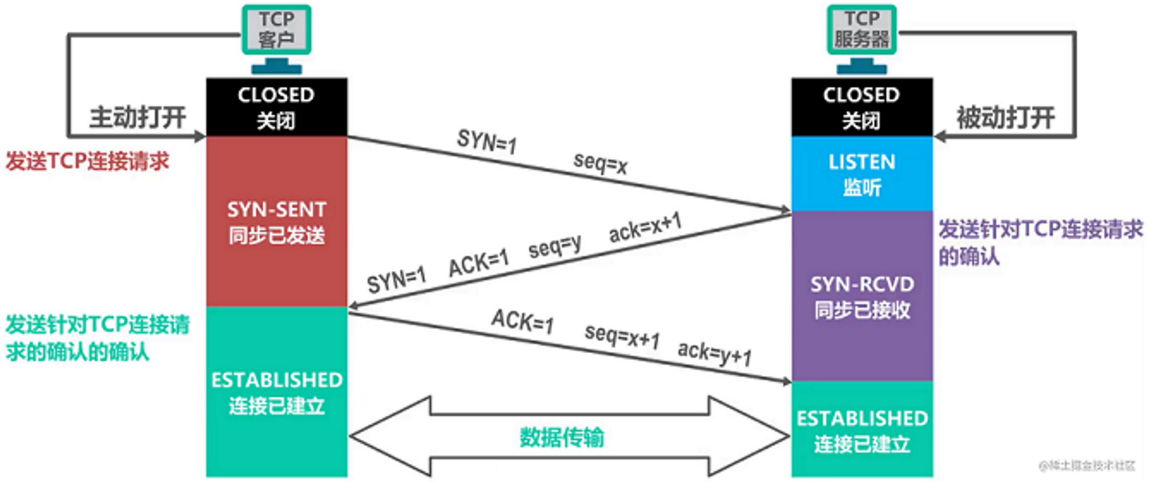
主动发起TCP连接建立 的称为 客户端
被动等待 TCP连接建立 的称为 服务器

1. 最开始，2端 的 TCP 进程都处于 关闭状态。
2. 服务器TCP进程 创建 传输控制块，用来 存储 TCP连接中的 一些重要信息，例如，TCP连接表，指向发送 和 接受 缓存的 指针，指向 重传队列的 指针，当前的 发送 和 接受 序号 等。之后 就 准备 接受 TCP客户端进程 的 连接请求，此时 TCP服务器进程 进入 监听状态，等待 TCP客户端进程的 连接请求。
3. TCP客户单 也需要 首先 创建 传输控制块，然后 再打算建立。TCP服务器进程 是 被动等待 来自TCP客户端进程的 连接请求，所以被称为 被动打开连接。
4. TCP连接时 向TCP服务器 进程 发送 TCP连接请求 报文段，并进入 同步已发送状态。
    a. TCP连接请求报文首部中的 同步位SYN 被设置为 1，表明这是一个 TCP 连接请求报文段。
    b. 序号字段seq 被设置为 一个 初始值 x 作为TCP客户端进程 所选择的 初始序号。
5. 由于TCP连接建立 是由 TCP 客户端进程主动发起的，因此成为 主动打开连接。 注意TCP

规定 SYN被设置为1 的报文段 不能携带数据 但要 消耗一个 序号。

6. TCP服务器进程 收到 TCP 连接请求报文段后，如果 同意建立连接，则向 客户端进程 发送 TCP 连接 请求 确认报文段，并进入到 同步已接收状态。
   a. 该报文段 首部中的 同步位 SYN 和 确认位ACK 都设置为1， 表明这是一个 TCP连接请求。
   b. 序号字段SEQ被设置了一个 初始值 y，作为 TCP服务器进程 所选择的 初始序号。
   c. 确认号 字段ACK 的值 被设置为 x+1， 这是对TCP 客户进程 所选择的 初始序号 SEQ 的确认。

   注意，这个报文段 也不能携带数据，因为 SYN是1，但同样要消耗一个 序号。
7. TCP客户进程 收到 TCP 连接请求 确认报文段后，还要向 TCP服务器 进程 发送一个 普通的TCP确认报文段 并进入 连接已建立状态。
   a. 该报文段首部中的 确认位ACK 为1，表明是一个 普通的TCP确认报文段
   b. 序号字段SEQ 被设置为 x+1， 这是因为 TCP客户端 发送的第一个TCP 报文的序号是 x，并且 不携带数据，所以 第二个报文段的 序号是 x + 1
   c. 确认号字段ACK 被设置为 y + 1， 这是对 TCP服务器进程 所选择的 初始序号的确认。
8. TCP服务器收到 该确认报文段后 也进入 连接已建立状态，现在 TCP双方都进入了 连接已建立 状态，它们可以基于 已建立好 的TCP 连接进行 可靠的数据传输了。



注意，TCP规定，普通的TCP 确认报文段 可以携带数据。但如果不携带数据 则不消耗 序号，在这种情况下，下一个数据报文段的序号仍然是 x + 1。
。。。？不消耗序号的话，下一个 不应该 还是 x ？

两次握手的问题
客户端发送 建立TCP连接请求，
但是超时，客户端重试，这次 服务器收到，并返回 连接建立 报文给 客户端，
数据传输，
挥手释放连接。
客户端发送的第一个请求 送达服务器，服务器 返回 连接建立报文 给客户端， 但是客户端已经关闭了，所以不会有响应， 服务器就一直维持着 这个连接，等待 客户端的数据。

在不可靠的信道中想要模拟出可靠的，双向的传输最少也得是三次通信，两次只能建立可靠的单向传输。

---

https://juejin.cn/post/7132499826771656734

四次挥手

1. 客户端主动停止连接，向服务器发送 FIN=1 的报文
2. 服务器收到后，返回一个 确认报文
3. 服务器 确定自己也没有数据 要发送给 客户端时，服务器会向 客户端 发送 FIN=1 的报文，并等待确认
4. 客户端在收到 服务器的 FIN=1 的报文是，会 立即返回 一个 确认报文，并进入 2MSL 等待阶段。
   在2MSL 时间范围内，如果客户端收到服务器端 超时重传的 FIN=1 的报文，则说明 客户端发出的确认，服务端 没有收到，所以此时客户端 会重新发送 确认并 重置计时器，在2MSL只有，没有收到服务端的 超时重传 FIN=1 的报文，则客户端正常关闭。服务器在收到 确认后也是正常关闭的。

2MSL MSL是Maximum Segment Lifetime英文的缩写,中文可以译为“报文最大生存时间”

---

https://juejin.cn/post/7075945250115551239

TCP/IP模型	OSI模型	作用
应用层http	应用层	为应用程序提供服务
应用层http	表示层	数据格式转化，数据加密
应用层http	会话层	建立，管理，维护会话
传输层TCP	传输层	建立，管理，维护 端对端的连接
网络层IP	网络层	IP选址 及 路由选择
链路层(物理网卡)	链路层	提供介质访问 和 链路管理
链路层(物理网卡)	物理层	物理传输

TCP/IP 协议族
TCP/IP 是 Internet 最基本的协议，Internet 国际互联网络的基础，由网络层的 IP 协议和 传输层的 TCP 协议组成。

TCP/IP概念层模型	功能	TCP/IP协议族
应用层	文件传输，电子邮件，文件服务，虚拟终端	TFTP,HTTP,SNMP,FTP,SMTP,DNS,Telnet

应用层	数据格式化，代码转换，数据加密	没有协议
应用层	解除 或 建立 与别的接点 的联系	没有协议
传输层	提供端对端的接口	TCP UDP
网络层	为数据包选择路由	IP,ICMP,RIP,OSPF,BGP,IGMP
链路层	传输地址的帧以及错误检测功能	SLIP,CSLIP,PPP,ARP,RARP,MTU
链路层	以二进制数据形式在物理媒体上传输数据	ISO2110,IEEE802,IEEE802.2

一次完整的http请求的过程

DNS域名解析（本地浏览器缓存，操作系统缓存或者DNS服务器）
三次握手建立TCP连接
客户端发起HTTP请求
服务器响应HTTP请求
客户端解析html代码，并请求HTML代码中的资源
客户端渲染展示内容
关闭TCP连接
1.0 连接不可以复用　　2.0 IO多路复用，一次TCP连接可以进行多次http请求

为什么需要四次挥手?
TCP是全双工（即客户端和服务端可以相互发送和接收请求）所以需要双方都确认关闭连接

为什么需要TIME_WAIT状态?
客户端的最后一次应答报文可能在网络传输中会丢失，所以客户端还得进行一次重传，确保可靠的终止TCP/IP 保证迟来的TCP报文有足够的时间，被识别并丢弃

==========

==========

==========

==========

==========

==========

==========

==========

==========

==========

==========