[文章] 作之间 · 2020-10-09
[文章] 每周总结四篇
[文章] 作之间 · 2020-10-09
[文章] Redux 管理数据的流程
[文章] 作之间 · 2020-10-09
[文章] 管理数据的流程
[文章] 作之间 · 2020-10-09
[文章] 作之间 · 2020-10-09
[文章] 作之间 · 2020-10-09
[文章] 作之间 · 2020-10-09
[文章] 作之间 · 2020-10-09
[文章] 作之间 · 2020-10-09
[文章] 作之间 · 2020-10-09
[文章] 作之间 · 2020-10-09
[文章] 作之间 · 2020-10-09
[文章] 作之间 · 2020-10-09
[文章] HTTP 缓存 · 2020-10-09
[文章] HTTP 缓存 · 2020-10-09
[文章] 入门到 · 2020-10-09
[文章] 作之间 · 2020-10-09

TCP

首先思考为啥要设计可靠性的呢？为了进一步的分析，我们来将目标不设计可靠性，

`可靠性目标：UDP 是 TCP 的目标是什么？`

TCP 是 面向 UDP 的基本，通过提供可靠性的数据传输服务的协议。它提供了一种面向连接的、可靠的、全双工的、字节流的传输。

头部

从这个图上我们都可以看到 TCP 头部比 UDP 头部要长的多。

对于 TCP 头部来说，以下几个字段很重要的：

- Sequence number，这个序号保证了 TCP 传输的报文都是有序的，对端可以通过序号顺序的拼接报文
- Acknowledgment number，这个序号表示期望接收到的字节的下一个序号Ack号是多少，表示接收方期望收到的下一个报文
- Window Size，窗口大小，表示接收方还有多少缓冲区，用于流量控制
- 标志位：
  - URG：紧急指针（urgent pointer）有效的意思，表示紧急数据，一般会标记出来的紧急数据的最后一个字节的偏移量，通常配合一定位于应用层的紧急数据的协议使用。紧急指针与紧急数据一起使用，
  - ACK：确认序号有效，表示确认序号有效，这时，TCP 还规定在连接建立后所有的传输报文段都必须 ACK 置为1；
  - PSH：这个 置为1 表示接收端应该立即将数据 push 给应用层，而不是等到缓冲区满或超时才push；
  - RST：重置连接，表示当前连接有错，需要重新建立连接；
  - SYN：建立连接，SYN=1，ACK=0时，表示当前报文是第一个握手报文，SYN=1，ACK=1时，表示当前报文是一个同意建立连接的应答；
  - FIN：这个 置为1 表示发送端已经没有任务了，可以关闭连接了。

状态机

TCP 的状态机是连接的，并且与建立和终止连接的过程有关系，图下所展示的就是这个状态机图的一部分。

在这之前要了解一下重要的状态缩写：SYT，表指的是主动发起连接的那端的状态缩写，接收端的状态缩写。

建立连接三次握手

首先要说明这连接的第一端的状态有两，被动连接和一端的主动连接，不要搞混了这两连接状态。TCP 的建立连接的基本步骤是：SYN 可以看一下图上的SYN。

此时，两端的 SYN 已经 SYT 状态，连接状态分别是：对方端的连接 SYN 。连接建立 LISTEN 状态，对方的连接等待了解连接状态。

第一次握手

客户端的连接状态发送SYN报文后，进入SYT等待状态等待接收端的SYN应答。服务端连接 SYN 状态接收 SYN 报文。

第二次握手

服务端的连接状态发送SYN报文后，进入SYT等待状态等待接收端的ACK应答。服务端连接 SYN 状态接收 SYN 报文。

第三次握手

客户端的连接状态发送ACK报文后，进入ESTABLISHED状态。服务端的连接 SYN 状态接收 ACK 报文。

断开连接四次挥手

SYN 的建立连接的三次握手，不是 ACK 的四次挥手。

第一次握手

客户端 FIN 的 ACK 报文，进入 FIN_WAIT_1 状态。

第二次握手

服务端 FIN 的 ACK 报文，进入 CLOSE_WAIT 状态。

第三次握手

服务端 FIN 的 ACK 报文，进入 LAST_ACK 状态。

第四次握手

客户端 FIN 的 ACK 报文，进入 TIME_WAIT 状态。

滑动窗口

停止等待 ARQ

正常传输过程

超时重传

丢失报文段

丢失 ACK 报文

流量控制

拥塞控制

慢开始

拥塞避免

快重传

快恢复

Zero 窗口

拥塞处理

粘包问题

常用算法

常用命令

实现

总结

参考

登录 / 注册
输入评论（支持Markdown）
发表评论

全部评论 (10)

最新 | 点赞最多 | 最早

最长记录 Guest · 1小时前
这个 TCP 的 Zero 窗口，一定是 Bug 吗？
0 | 回复

Ian Wang Guest · 1小时前
这个 TCP 的 Zero 窗口，一定是 Bug 吗？
0 | 回复

刘思凡_品牌 admin Guest · 1小时前
Nice article!
0 | 回复

Yoyoyo Guest · 1小时前
这个 TCP 的 Zero 窗口，一定是 Bug 吗？
0 | 回复

Wang Ziz Guest · 1小时前
Nice article!
0 | 回复

Ian Wang Guest · 1小时前
这个 TCP 的 Zero 窗口，一定是 Bug 吗？
0 | 回复

Ian Wang Guest · 1小时前
这个 TCP 的 Zero 窗口，一定是 Bug 吗？
0 | 回复

Ian Wang Guest · 1小时前
这个 TCP 的 Zero 窗口，一定是 Bug 吗？
0 | 回复

Ian Wang Guest · 1小时前
这个 TCP 的 Zero 窗口，一定是 Bug 吗？
0 | 回复

查看更多回复 ∨

小强爱学习 Guest · 1小时前
这个 TCP 的 Zero 窗口，一定是 Bug 吗？
0 | 回复

**Ian Wang