

问题一：UDP 与 TCP 的区别是什么？

前言：网络协议是每个前端工程师都必须掌握的知识，我们将先来学习传输层中的两个协议：UDP 以及 TCP。对于大部分工程师来说最常用的协议也就是这两个了，并且面试中经常会提问的也是关于这两个协议的区别。

首先 UDP 协议是面向无连接的，也就是说不需要在正式传递数据之前先连接起双方。

然后 UDP 协议只是数据报文的搬运工，不保证有序且不丢失的传递到对端，并且 UDP 协议也没有任何控制流量的算法，总的来说 UDP 相较于 TCP 更加的轻便。

1. 面向无连接

首先 UDP 是不需要和 TCP 一样在发送数据前进行三次握手建立连接的，想发数据就可以开始发送了。

并且也只是数据报文的搬运工，不会对数据报文进行任何拆分和拼接操作。

具体来说就是：

- 在发送端，应用层将数据传递给传输层的 UDP 协议，UDP 只会给数据增加一个 UDP 头标识下是 UDP 协议，然后就传递给网络层了
- 在接收端，网络层将数据传递给传输层，UDP 只去除 IP 报文头就传递给应用层，不会任何拼接操作

1. 不可靠性

首先不可靠性体现在无连接上，通信都不需要建立连接，想发就发，这样的情况肯定不可靠。

并且收到什么数据就传递什么数据，并且也不会备份数据，发送数据也不会关心对方是否已经正确接收到数据了。

再者网络环境时好时坏，但是 UDP 因为没有拥塞控制，一直会以恒定的速度发送数据。即使网络条件不好，也不会对发送速率进行调整。这样实现的弊端就是在网络条件不好的情况下可能会导致丢包，但是优点也很明显，在某些实时性要求高的场景（比如电话会议）就需要使用 UDP 而不是 TCP。

1. 高效

虽然 UDP 协议不是那么的可靠，但是正是因为它不是那么的可靠，所以也就没有 TCP 那么复杂了，需要保证数据不丢失且有序到达。

因此 UDP 的头部开销小，只有八字节，相比 TCP 的至少二十字节要少得多，在传输数据报文时是很高效的。

UDP Header																															
0								1								2								3							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Source port																Destination port															
Length																Checksum															

UDP 头部包含了以下几个数据

- 两个十六位的端口号，分别为源端口（可选字段）和目标端口
- 整个数据报文的长度
- 整个数据报文的检验和（IPv4 可选 字段），该字段用于发现头部信息和数据中的错误

1. 传输方式

UDP 不止支持一对一的传输方式，同样支持一对多，多对多，多对一的方式，也就是说 UDP 提供了单播，多播，广播的功能。

1. 适合使用的场景

UDP 虽然对比 TCP 有很多缺点，但是正是因为这些缺点造就了它高效的特性，在很多实时性要求高的地方都可以看到 UDP 的身影。

直播

想必大家都看过直播吧，大家可以考虑下如果直播使用了基于 TCP 的协议会发生什么事情？

TCP 会严格控制传输的正确性，一旦有某一个数据对端没有收到，就会停止下来直到对端收到这个数据。这种问题在网络条件不错的情况下可能并不会发生什么事情，但是在网络情况差的时候就会变成画面卡住，然后再继续播放下一帧的情况。

但是对于直播来说，用户肯定关注的是最新的画面，而不是因为网络条件差而丢失的老旧画面，所以 TCP 在这种情况下无用武之地，只会降低用户体验。

王者荣耀

虽然我具体不知道王者荣耀底层使用了什么协议，但是对于这类实时性要求很高的游戏来说，UDP 是跑不了的。

为什么这样说呢？首先对于王者荣耀来说，用户体量是相当大的，如果使用 TCP 连接的话，就可能会出现服务器不够用的情况，因为每台服务器可供支撑的 TCP 连接数量是有限制的。

再者，因为 TCP 会严格控制传输的正确性，如果因为用户网络条件不好就造成页面卡顿然后再传输旧的游戏画面是肯定不能接受的，毕竟对于这类实时性要求很高的游戏来说，最新的游戏画面才是最需要的，而不是老旧的画面，否则角色都不知道死多少次了。

小结

最后总结一下这一章节的内容：

- UDP 相比 TCP 简单的多，不需要建立连接，不需要验证数据报文，不需要流量控制，只会把想发的数据报文一股脑的丢给对端
- 虽然 UDP 并没有 TCP 传输来的准确，但是也能在很多实时性要求高的地方有所作为