**QUIC简单介绍**

QUIC（Quick UDP Internet Connections）是谷歌推出的一套基于UDP的传输协议，它实现了TCP + HTTPS + HTTP/2的功能，目的是保证可靠性的同时降低网络延迟。因为UDP是一个简单传输协议，基于UDP可以摆脱TCP传输确认、重传、慢启动等因素，其建立安全连接只需要一个往返时间（RTT），它还实现了HTTP/2多路复用、头部压缩等功能。

**为什么需要QUIC：**

近三十年来，TCP协议发展得非常缓慢；

很多网络中间层，比如防火墙、网关等，都强依赖于TCP指定的各类规则，所以TCP的修改很容易由于这些中间环节的存在而受到干扰；

TCP是由操作系统在内核层面实现的， 导致TCP的迭代受限于操作系统的升级。

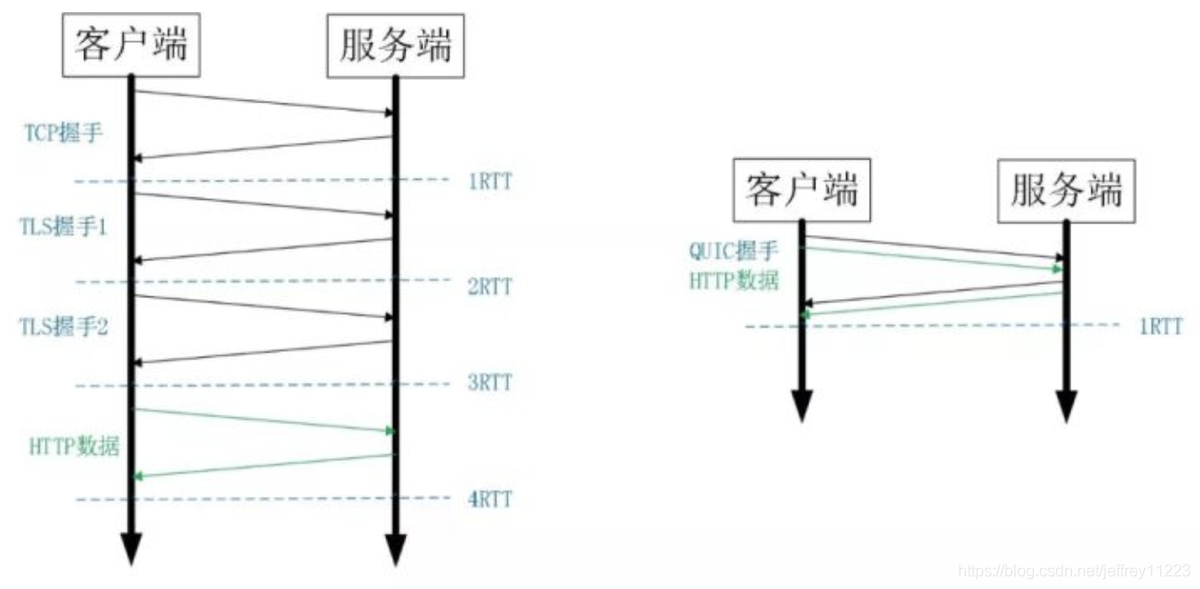
**应用场景：**

弱网络环境下能够提升 20% 以上的访问速度；

频繁切换网络的情况下，不会断线，不需要重连，用户无任何感知。

**QUIC相比于TCP的关键特性：**

**1、链接耗时更短**



（a）TCP + TLS (b) QUIC

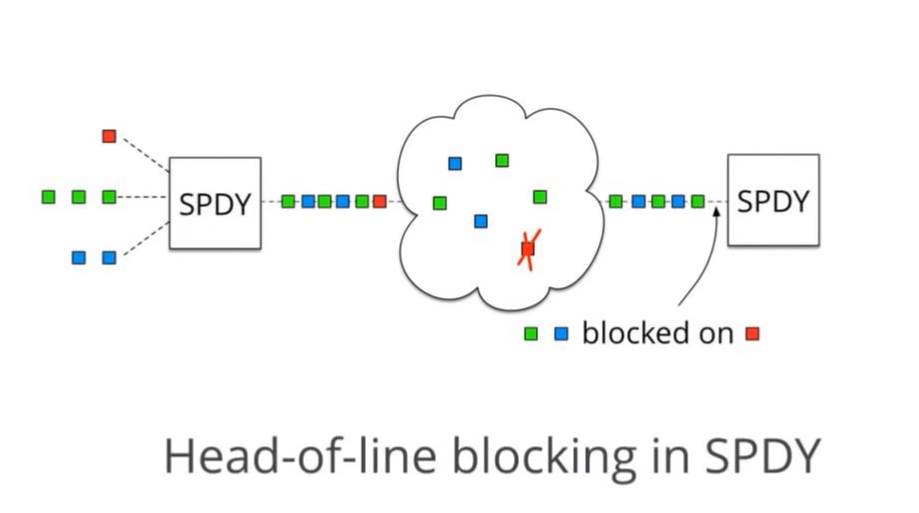
对于HTTPS页面，需要花在连接阶段的耗时：TCP三次握手（花费1RTT）+ TLS握手（完全握手耗费2RTT，简化握手需要1RTT）= 2 ~ 3RTT，而QUIC只用耗费0-1RTT的连接耗时（仅首次连接需要1RTT），就能做到等同于TCP+TLS连接的效果（0RTT就等同于发给服务器的第一个包就带有请求数据）。

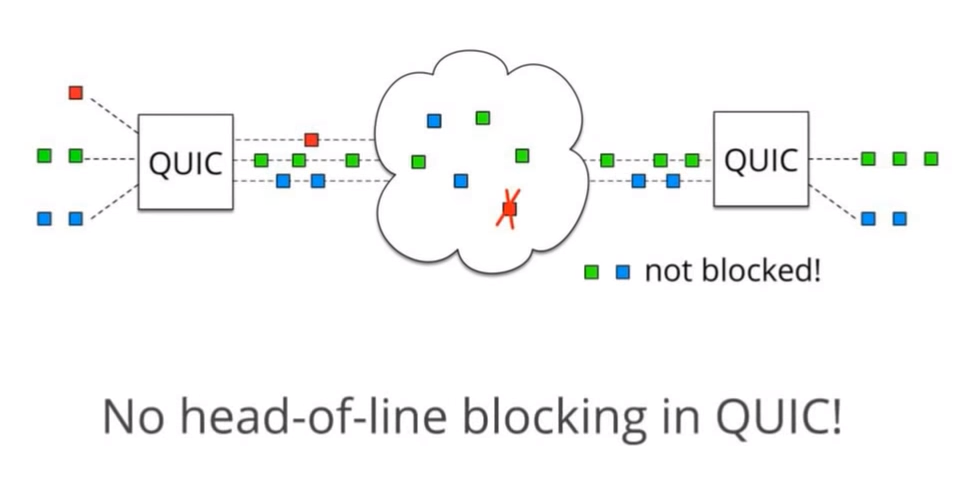
原理：

1. UDP是不面向连接的，事先不需要握手过程就可以直接传输数据；
2. HTTPS的TLS握手的一个主要目的在于约定一个对称秘钥，用于之后的数据传输，QUIC在这一步用了经典的秘钥加密算法（DH算法）来实现对称秘钥的交换。
3. 第一次连接时，如果客户端之前没有连接过此服务器，那么他会发送一个Hello Packet，服务器接到后会回复一个数据包，里面含有安全证书以及对此客户端唯一的SYN cookie。客户端接到包之后，会保存此SYN cookie，其类似于令牌，能够验证客户端身份。这样建立连接只需要1个RTT。
4. 因为客户端保留了一份服务器的安全证书，当其想要再次连接服务器时，客户端假设这个安全证书没有过期，还是有效的，因此其直接加密一个Hello Packet后并发送后，不用等回复就可以直接加密其他的数据包并发送。服务器成功解码后，验证了客户端的安全性之后，就可以继续处理接下来收到的数据包，此时延时是0个RTT。

**2、可以避免队头阻塞**

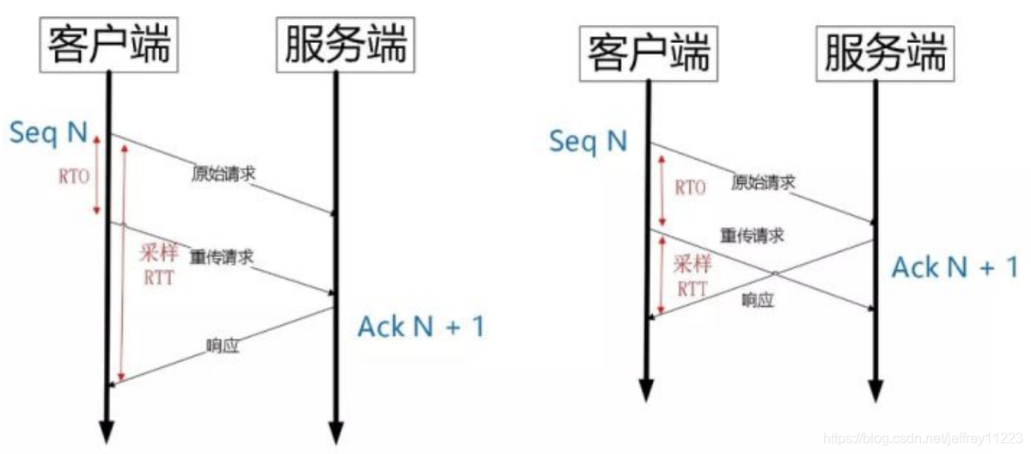
对于TCP来说，因为需要保证包接收时的顺序，因此当某个包丢失需要等待重传时，位于其后的所有包即使已经到达并被保存在接收缓存中，仍然只能等到丢失的包到达后才能被确认接收，即此时被阻塞，由此导致了所谓的队头阻塞问题。而在QUIC中因为采用了多路复用机制，其丢包、流控都是基于stream的，丢失的包只会影响其在的特定的那一条stream，其他stream中的数据将能够正常到达。



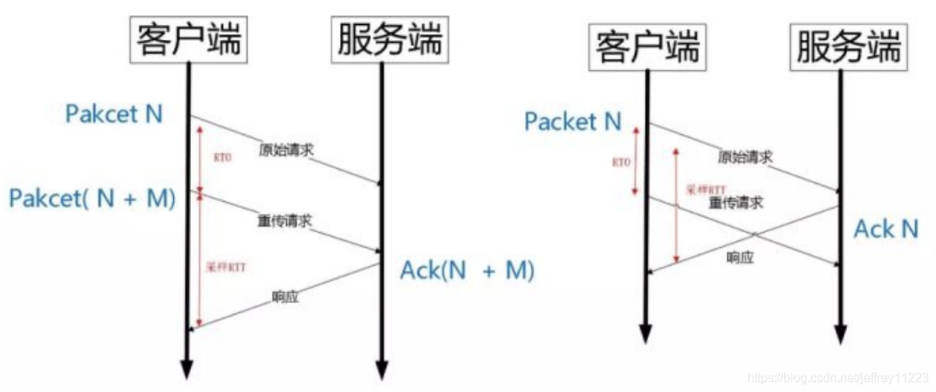


**3、没有歧义的重传:**

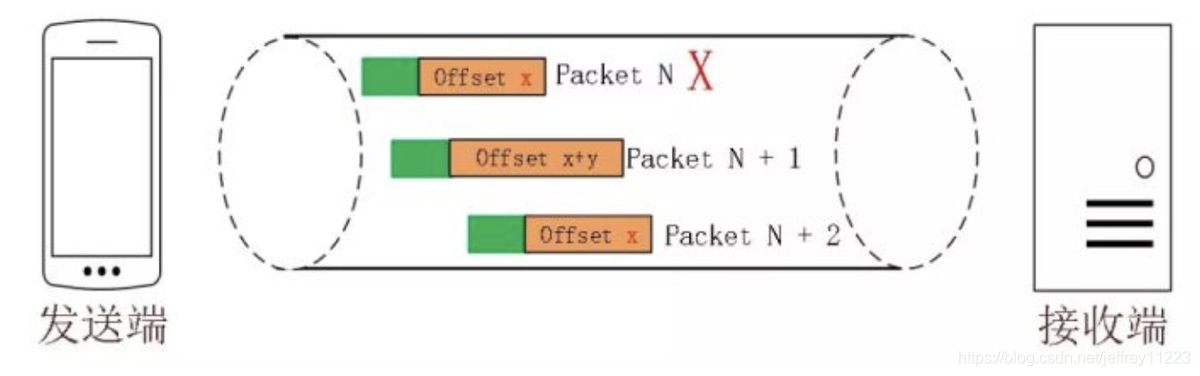
TCP 重传的包的Sequence Number和原始包的Sequence Number 是一致的，也正是由于这个特性，产生了 TCP 重传的歧义问题：



超时事件 RTO 发生后，客户端发起重传，然后接收到了 Ack 数据。由于Sequence Number一样，这个 Ack 到底是原始请求的响应还是重传请求的响应呢？这就间接导致了RTT计算的歧义。



QUIC使用 Packet Number 代替了 TCP 的 Sequence Number，并且每个 Packet Number 都严格递增，也就是说就算 Packet N 丢失了，重传的 Packet N 的 Packet Number 已经不是 N，而是一个比 N 大的值，这就解决了RTT计算的歧义问题。



QUIC 引入了一个叫 Stream Offset 的概念。假设 Packet N 丢失了，发起重传，重传的 Packet Number 是 N+2，但是它的 Stream 的 Offset 依然是 x，这样就算 Packet N + 2 是后到的，依然可以将 Stream x 和 Stream x+y 按照顺序组织起来。

**4、前向纠错&重传**

TCP采用重传机制，而QUIC采用纠错机制。

TCP发生丢包时，需要一个等待延时判断发生了丢包，然后再启动重传机制，这个过程会造成一定的阻塞，影响传输时间。QUIC则采用一种更主动的方案，有点类似RAID5，每n个包额外发一个校验和包。如果这n个包中丢了一个包，可以通过其他包及校验和恢复出来，完全不需要重传。

**5、链接迁移**

当客户端IP或端口发生变化时（这在移动端比较常见），TCP连接基于两端的ip:port四元组标示，因此会重新握手，而UDP不面向连接，不用握手。其上层的QUIC链路由于使用了64位Connection id作为唯一标识，四元组变化不影响链路的标示，也不用重新握手。因此网络链接状态变化时不会增加额外的握手重连耗时。

**参考**：https://www.mobibrw.com/2020/23190