

31 | HTTP3: 甩掉TCP、TLS的包袱, 构建高效网络

前面两篇文章我们分析了HTTP/1和HTTP/2, 在HTTP/2出现之前, 开发者需要采取很多变通的方式来解决HTTP/1所存在的问题, 不过HTTP/2在2018年就开始得到了大规模的应用, HTTP/1中存在的一大堆缺陷都得到了解决。

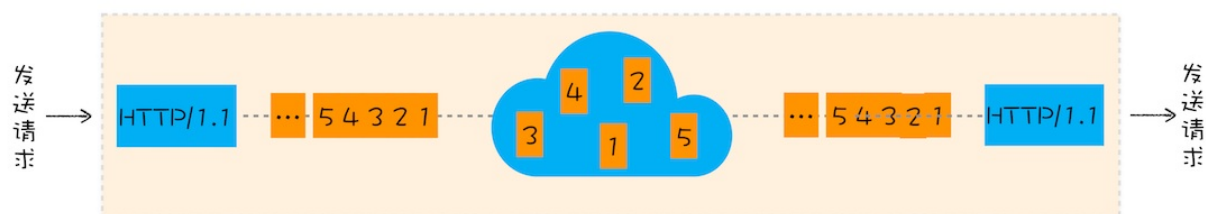
HTTP/2的一个核心特性是使用了多路复用技术, 因此它可以通过一个TCP连接来发送多个URL请求。多路复用技术能充分利用带宽, 最大限度规避了TCP的慢启动所带来的问题, 同时还实现了头部压缩、服务器推送等功能, 使得页面资源的传输速度得到了大幅提升。在HTTP/1.1时代, 为了提升并行下载效率, 浏览器为每个域名维护了6个TCP连接; 而采用HTTP/2之后, 浏览器只需要为每个域名维护1个TCP持久连接, 同时还解决了HTTP/1.1队头阻塞的问题。

从目前的情况来看, HTTP/2似乎可以完美取代HTTP/1了, 不过HTTP/2依然存在一些缺陷, 于是就有了HTTP/3。和通常一样, 介绍HTTP/3之前, 我们先来看看HTTP/2到底有什么缺陷。

TCP的队头阻塞

虽然HTTP/2解决了应用层面的队头阻塞问题, 不过和HTTP/1.1一样, HTTP/2依然是基于TCP协议的, 而TCP最初就是为了单连接而设计的。你可以把TCP连接看成是两台计算机之前的一个虚拟管道, 计算机的一端将要传输的数据按照顺序放入管道, 最终数据会以相同的顺序出现在管道的另外一头。

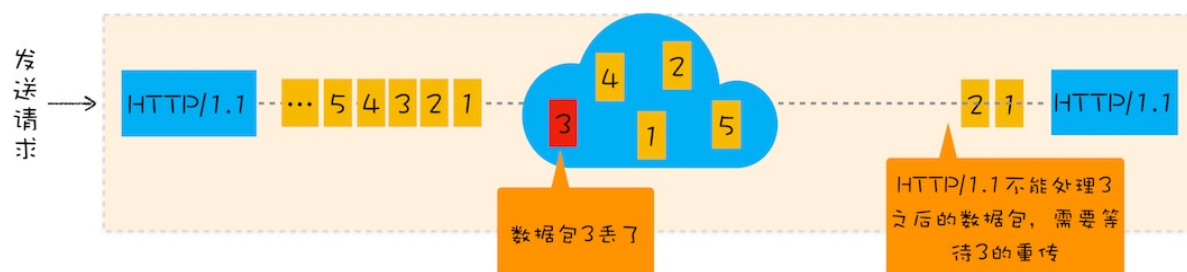
接下来我们就来分析下HTTP/1.1协议栈中TCP是如何传输数据的。为直观理解, 你可以参考下图:



正常情况下的TCP传输数据过程

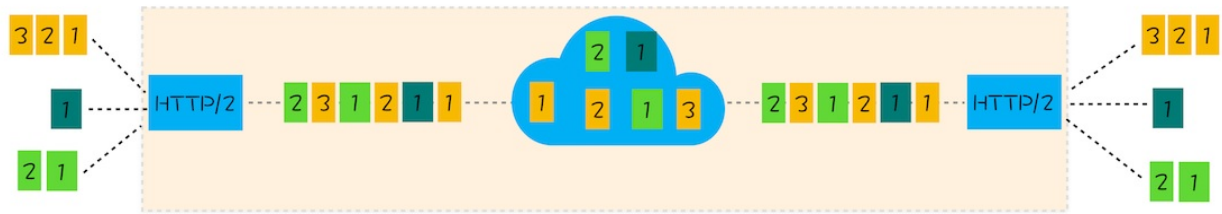
通过上图你会发现, 从一端发送给另外一端的数据会被拆分为一个个按照顺序排列的数据包, 这些数据包通过网络传输到了接收端, 接收端再按照顺序将这些数据包组合成原始数据, 这样就完成了数据传输。

不过, 如果在数据传输的过程中, 有一个数据因为网络故障或者其他原因而丢包了, 那么整个TCP的连接就会处于暂停状态, 需要等待丢失的数据包被重新传输过来。你可以把TCP连接看成是一个按照顺序传输数据的管道, 管道中的任意一个数据丢失了, 那之后的数据都需要等待该数据的重新传输。为了直观理解, 你可以参考下图:



我们就把在TCP传输过程中，由于单个数据包的丢失而造成的阻塞称为TCP上的队头阻塞。

那队头阻塞是怎么影响HTTP/2传输的呢？首先我们来看正常情况下HTTP/2是怎么传输多路请求的，为了直观理解，你可以参考下图：



HTTP/2 多路复用

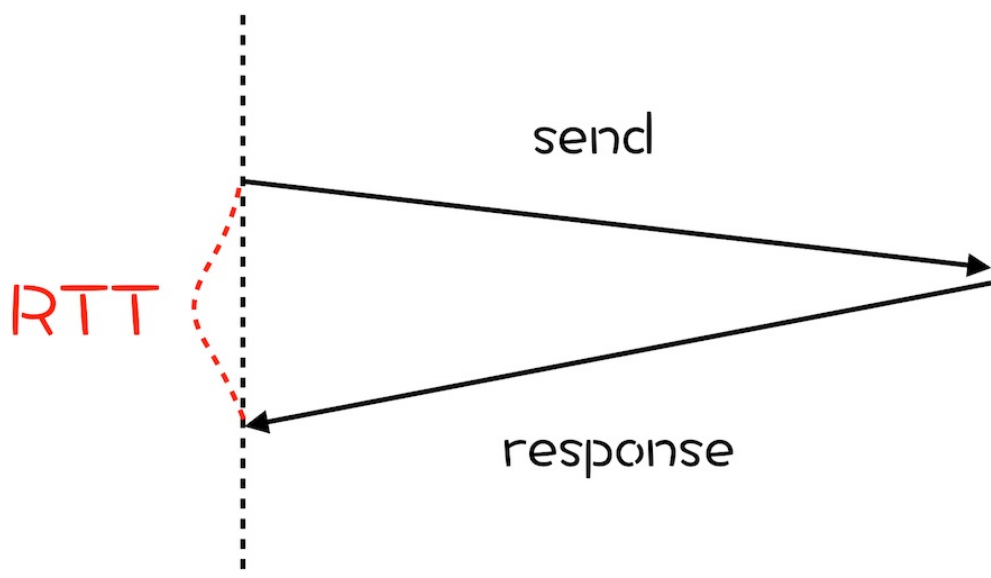
通过该图，我们知道在HTTP/2中，多个请求是跑在一个TCP管道中的，如果其中任意一路数据流中出现了丢包的情况，那么就会阻塞该TCP连接中的所有请求。这不同于HTTP/1.1，使用HTTP/1.1时，浏览器为每个域名开启了6个TCP连接，如果其中的1个TCP连接发生了队头阻塞，那么其他的5个连接依然可以继续传输数据。

所以随着丢包率的增加，HTTP/2的传输效率也会越来越差。有测试数据表明，当系统达到了2%的丢包率时，HTTP/1.1的传输效率反而比HTTP/2表现得更好。

TCP建立连接的延时

除了TCP队头阻塞之外，TCP的握手过程也是影响传输效率的一个重要因素。

为了搞清楚TCP协议建立连接的延迟问题，我们还是先回顾下网络延迟的概念，这会有助于你对后面内容的理解。网络延迟又称为RTT（Round Trip Time）。我们把从浏览器发送一个数据包到服务器，再从服务器返回数据包到浏览器的整个往返时间称为RTT（如下图）。RTT是反映网络性能的一个重要指标。



那建立TCP连接时，需要花费多少个RTT呢？下面我们来计算下。

我们知道HTTP/1和HTTP/2都是使用TCP协议来传输的，而如果使用HTTPS的话，还需要使用TLS协议进行安全传输，而使用TLS也需要一个握手过程，这样就需要有两个握手延迟过程。

1. 在建立TCP连接的时候，需要和服务端进行三次握手来确认连接成功，也就是说需要在消耗完1.5个RTT之后才能进行数据传输。
2. 进行TLS连接，TLS有两个版本——TLS1.2和TLS1.3，每个版本建立连接所花的时间不同，大致是需要1~2个RTT，关于HTTPS我们到后面到安全模块再做详细介绍。

总之，在传输数据之前，我们需要花掉3~4个RTT。如果浏览器和服务器的物理距离较近，那么1个RTT的时间可能在10毫秒以内，也就是说总共要消耗掉30~40毫秒。这个时间也许用户还可以接受，但如果服务器相隔较远，那么1个RTT就可能需要100毫秒以上了，这种情况下整个握手过程需要300~400毫秒，这时用户就能明显地感受到“慢”了。

TCP协议僵化

现在我们知道了TCP协议存在队头阻塞和建立连接延迟等缺点，那我们是不是可以通过改进TCP协议来解决这些问题呢？

答案是：**非常困难**。之所以这样，主要有两个原因。

第一个是**中间设备的僵化**。要搞清楚什么是中间设备僵化，我们先要弄明白什么是中间设备。我们知道互联网是由多个网络互联的网状结构，为了能够保障互联网的正常工作，我们需要在互联网的各处搭建各种设备，这些设备就被称为中间设备。

这些中间设备有很多种类型，并且每种设备都有自己的目的，这些设备包括了路由器、防火墙、NAT、交换机等。它们通常依赖一些很少升级的软件，这些软件使用了大量的TCP特性，这些功能被设置之后就很少更新了。

所以，如果我们在客户端升级了TCP协议，但是当新协议的数据包经过这些中间设备时，它们可能不理解包的内容，于是这些数据就会被丢弃掉。这就是中间设备僵化，它是阻碍TCP更新的一大障碍。

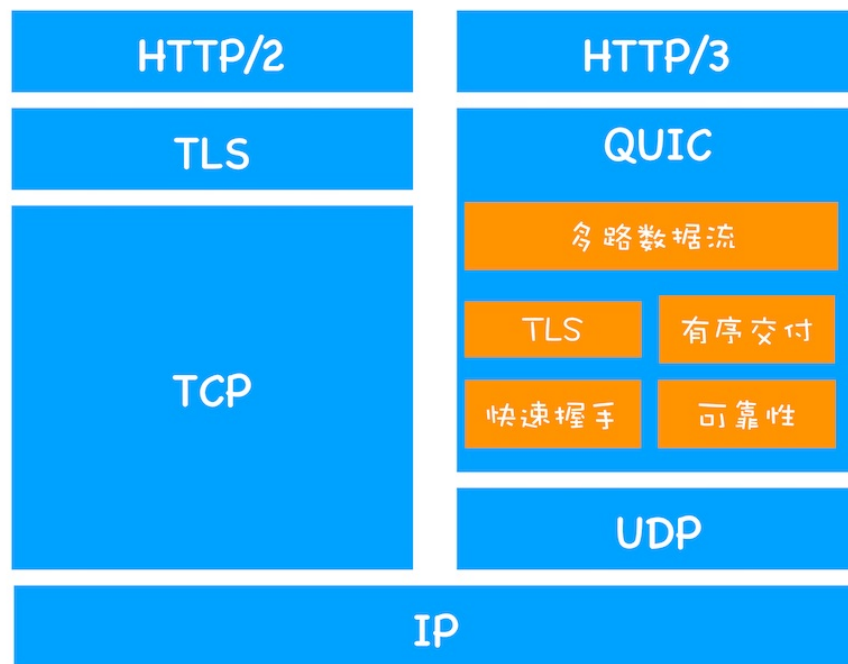
除了中间设备僵化外，**操作系统也是导致TCP协议僵化的另外一个原因**。因为TCP协议都是通过操作系统内核来实现的，应用程序只能使用不能修改。通常操作系统的更新都滞后于软件的更新，因此要想自由地更新内核中的TCP协议也是非常困难的。

QUIC协议

HTTP/2存在一些比较严重的与TCP协议相关的缺陷，但由于TCP协议僵化，我们几乎不可能通过修改TCP协议自身来解决这些问题，那么解决问题的思路是绕过TCP协议，发明一个TCP和UDP之外的新的传输协议。但是这也面临着和修改TCP一样的挑战，因为中间设备的僵化，这些设备只认TCP和UDP，如果采用了新的协议，新协议在这些设备同样不被很好地支持。

因此，HTTP/3选择了一个折衷的方法——UDP协议，基于UDP实现了类似于TCP的多路数据流、传输可靠

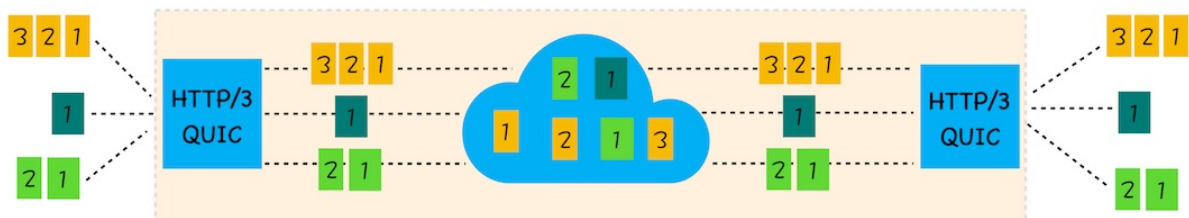
性等功能，我们把这套功能称为**QUIC协议**。关于HTTP/2和HTTP/3协议栈的比较，你可以参考下图：



HTTP/2和HTTP/3协议栈

通过上图我们可以看出，HTTP/3中的QUIC协议集合了以下几点功能。

- **实现了类似TCP的流量控制、传输可靠性的功能。**虽然UDP不提供可靠性的传输，但QUIC在UDP的基础之上增加了一层来保证数据可靠性传输。它提供了数据包重传、拥塞控制以及其他一些TCP中存在的特性。
- **集成了TLS加密功能。**目前QUIC使用的是TLS1.3，相较于早期版本TLS1.3有更多的优点，其中最重要的一点是减少了握手所花费的RTT个数。
- **实现了HTTP/2中的多路复用功能。**和TCP不同，QUIC实现了在同一物理连接上可以有多个独立的逻辑数据流（如下图）。实现了数据流的单独传输，就解决了TCP中队头阻塞的问题。



QUIC协议的多路复用

- **实现了快速握手功能。**由于QUIC是基于UDP的，所以QUIC可以实现使用0-RTT或者1-RTT来建立连接，这意味着QUIC可以用最快的速度来发送和接收数据，这样可以大大提升首次打开页面的速度。

HTTP/3的挑战

通过上面的分析，我们相信在技术层面，HTTP/3是个完美的协议。不过要将HTTP/3应用到实际环境中依然面临着诸多严峻的挑战，这些挑战主要来自于以下三个方面。

第一，从目前的情况来看，服务器和浏览器端都没有对HTTP/3提供比较完整的支持。Chrome虽然在数年前就开始支持Google版本的QUIC，但是这个版本的QUIC和官方的QUIC存在着非常大的差异。

第二，部署HTTP/3也存在着非常大的问题。因为系统内核对UDP的优化远远没有达到TCP的优化程度，这也是阻碍QUIC的一个重要原因。

第三，中间设备僵化的问题。这些设备对UDP的优化程度远远低于TCP，据统计使用QUIC协议时，大约有3%~7%的丢包率。

总结

好了，今天就介绍到这里，下面我来总结下本文的主要内容。

我们首先分析了HTTP/2中所存在的一些问题，主要包括了TCP的队头阻塞、建立TCP连接的延时、TCP协议僵化等问题。

这些问题都是TCP的内部问题，因此要解决这些问题就要优化TCP或者“另起炉灶”创造新的协议。由于优化TCP协议存在着诸多挑战，所以官方选择了创建新的QUIC协议。

HTTP/3正是基于QUIC协议的，你可以把QUIC看成是集成了“TCP+HTTP/2的多路复用+TLS等功能”的一套协议。这是集众家所长的一个协议，从协议最底层对Web的文件传输做了比较彻底的优化，所以等生态相对成熟时，可以用来打造比现在的HTTP/2还更加高效的网络。

虽说这套协议解决了HTTP/2中因TCP而带来的问题，不过由于是改动了底层协议，所以推广起来还会面临着巨大的挑战。

关于HTTP/3的未来，我有下面两点判断：

1. 从标准制定到实践再到协议优化还需要走很长一段路；
2. 因为动了底层协议，所以HTTP/3的增长会比较缓慢，这和HTTP/2有着本质的区别。

思考时间

那你来总结一下，HTTP/3都做了哪些性能上的改进？它所面临的挑战又是什么？

欢迎在留言区与我分享你的想法，也欢迎你在留言区记录你的思考过程。感谢阅读，如果你觉得这篇文章对你有帮助的话，也欢迎把它分享给更多的朋友。

浏览器工作原理与实践

>>> 透过浏览器看懂前端本质

李兵

前盛大创新院高级研究员



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。