

02-TCP协议：如何保证页面文件能被完整送达浏览器？

在衡量Web页面性能的时候有一个重要的指标叫“FP（First Paint）”，是指从页面加载到首次开始绘制的时长。这个指标直接影响了用户的跳出率，更快的页面响应意味着更多的PV、更高的参与度，以及更高的转化率。那什么影响FP指标呢？其中一个重要的因素是网络加载速度。

要想优化Web页面的加载速度，你需要对网络有充分的了解。而理解网络的关键是要对网络协议有深刻的认识，不管你是使用HTTP，还是使用WebSocket，它们都是基于TCP/IP的，如果你对这些原理有足够了解，也就清楚如何去优化Web性能，或者能更轻松地定位Web问题了。此外，TCP/IP的设计思想还有助于拓宽你的知识边界，从而在整体上提升你对项目的理解和解决问题的能力。

因此，在这篇文章中，我会给你重点介绍在Web世界中的TCP/IP是如何工作的。当然，协议并不是本专栏的重点，这篇文章我会从我的角度结合HTTP来分析网络请求的核心路径，如果你想对网络协议有更深入的理解，那我推荐你学习刘超老师的《趣谈网络协议》专栏，以及陶辉老师的《Web协议详解与抓包实战》视频课程。

好，接下来我们回到正题，开始今天的内容。在网络中，一个文件通常会被拆分为很多数据包来进行传输，而数据包在传输过程中又有很大概率丢失或者出错。那么如何保证页面文件能被完整地送达浏览器呢？

这篇文章将站在数据包的视角，给出问题答案。

一个数据包的“旅程”

下面我将分别从“数据包如何送达主机”“主机如何将数据包转交给应用”和“数据是如何被完整地送达应用程序”这三个角度来为你讲述数据的传输过程。

互联网，实际上是一套理念和协议组成的体系架构。其中，协议是一套众所周知的规则 and 标准，如果各方都同意使用，那么它们之间的通信将变得毫无障碍。

互联网中的数据是通过数据包来传输的。如果发送的数据很大，那么该数据就会被拆分为很多小数据包来传输。比如你现在听的音频数据，是拆分成一个个小的数据包来传输的，并不是一个大的文件一次传输过来的。

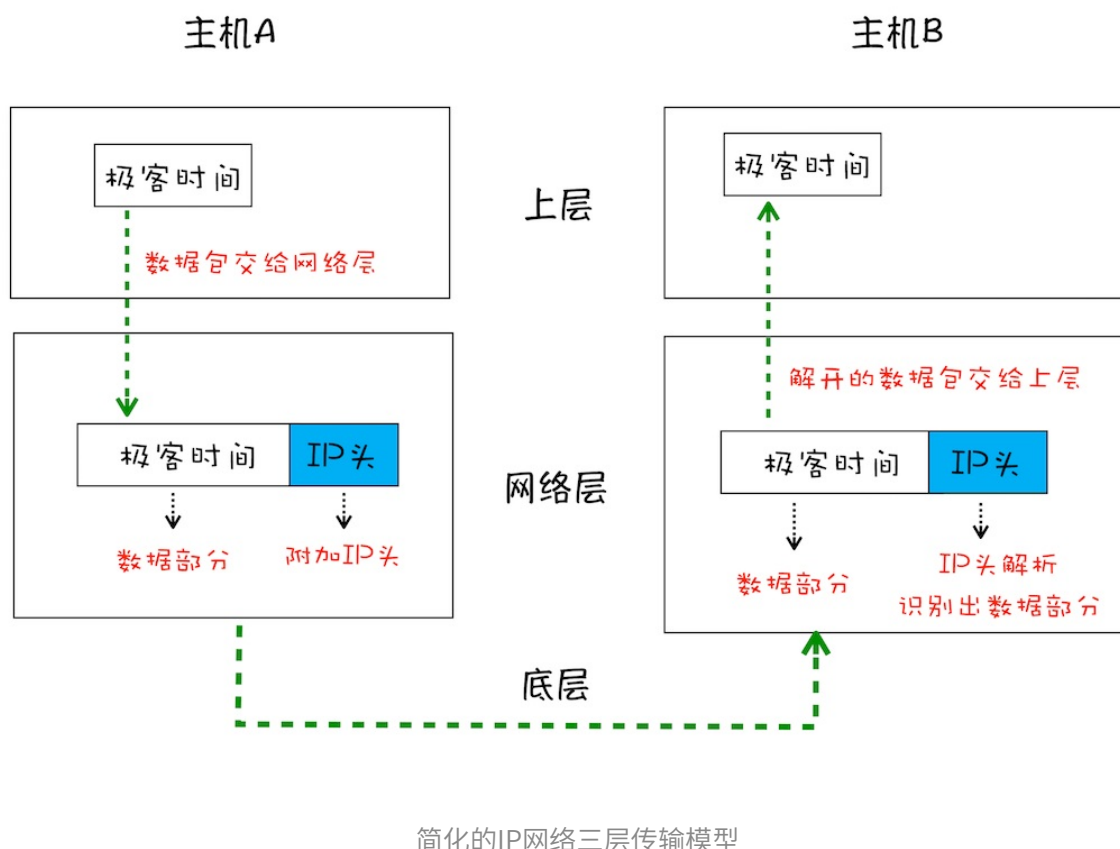
1. IP：把数据包送达目的主机

数据包要在互联网上进行传输，就要符合**网际协议**（Internet Protocol，简称**IP**）标准。互联网上不同的在线设备都有唯一的地址，地址只是一个数字，这和大部分家庭收件地址类似，你只需要知道一个家庭的具体地址，就可以往这个地址发送包裹，这样物流系统就能把物品送到目的地。

计算机的地址就称为IP地址，访问任何网站实际上只是你的计算机向另外一台计算机请求信息。

如果要想把一个数据包从主机A发送给主机B，那么在传输之前，数据包上会被附加上主机B的IP地址信息，这样在传输过程中才能正确寻址。额外地，数据包上还会附加上主机A本身的IP地址，有了这些信息主机B才可以回复信息给主机A。这些附加的信息会被装进一个叫**IP头**的数据结构里。**IP头是IP数据包开头的信息，包含IP版本、源IP地址、目标IP地址、生存时间等信息。**如果你要详细了解IP头信息，可以参考[该链接](#)。

为了方便理解，我先把网络简单分为三层结构，如下图：



下面我们一起来看下一个数据包从主机A到主机B的旅程：

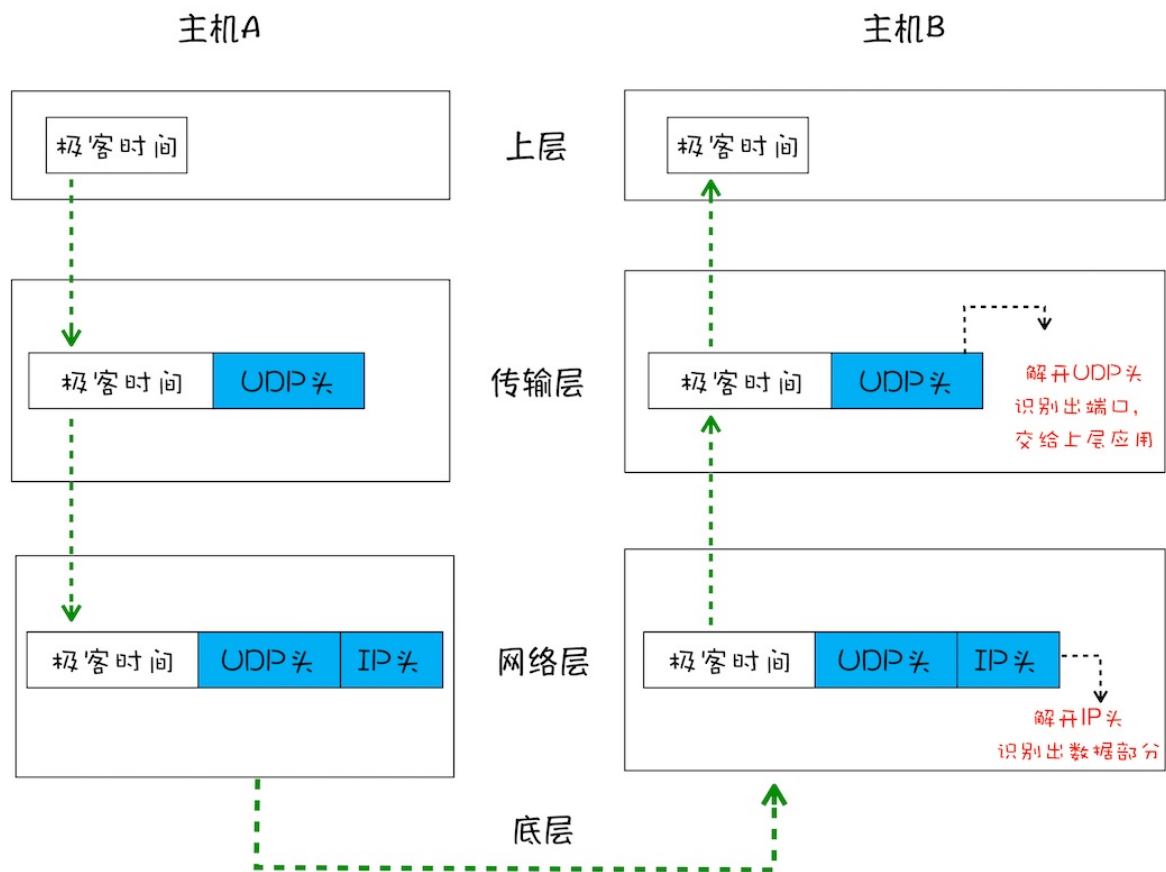
- 上层将含有“极客时间”的数据包交给网络层；
- 网络层再将IP头附加到数据包上，组成新的 **IP数据包**，并交给底层；
- 底层通过物理网络将数据包传输给主机B；
- 数据包被传输到主机B的网络层，在这里主机B拆开数据包的IP头信息，并将拆开来的数据部分交给上层；
- 最终，含有“极客时间”信息的数据包就到达了主机B的上层了。

2. UDP：把数据包送达应用程序

IP是非常底层的协议，只负责把数据包传送到对方电脑，但是对方电脑并不知道把数据包交给哪个程序，是交给浏览器还是交给王者荣耀？因此，需要基于IP之上开发能和应用打交道的协议，最常见的是“**用户数据包协议**（User Datagram Protocol）”，简称**UDP**。

UDP中一个最重要的信息是**端口号**，端口号其实就是一个数字，每个想访问网络的程序都需要绑定一个端口号。通过端口号UDP就能把指定的数据包发送给指定的程序了，所以**IP通过IP地址信息把数据包发送给指定的电脑，而UDP通过端口号把数据包分发给正确的程序**。和IP头一样，端口号会被装进UDP头里面，UDP头再和原始数据包合并组成新的UDP数据包。UDP头中除了目的端口，还有源端口号等信息。

为了支持UDP协议，我把前面的三层结构扩充为四层结构，在网络层和上层之间增加了传输层，如下图所示：



简化的UDP网络四层传输模型

下面我们一起来看下一个数据包从主机A旅行到主机B的路线：

- 上层将含有“极客时间”的数据包交给传输层；
- 传输层会在数据包前面加上**UDP头**，组成新的UDP数据包，再将新的UDP数据包交给网络层；
- 网络层再将IP头附加到数据包上，组成新的IP数据包，并交给底层；
- 数据包被传输到主机B的网络层，在这里主机B拆开IP头信息，并将拆开来的数据部分交给传输层；
- 在传输层，数据包中的UDP头会被拆开，**并根据UDP中所提供的端口号，把数据部分交给上层的应用程序**；
- 最终，含有“极客时间”信息的数据包就旅行到了主机B上层应用程序这里。

在使用UDP发送数据时，有各种因素会导致数据包出错，虽然UDP可以校验数据是否正确，但是对于错误的数据包，**UDP并不提供重发机制，只是丢弃当前的包，而且UDP在发送之后也无法知道是否能达到目的地。**

虽说**UDP不能保证数据可靠性，但是传输速度却非常快**，所以UDP会应用在一些关注速度、但不那么严格要求数据完整性的领域，如在线视频、互动游戏等。

3. TCP：把数据完整地送达应用程序

对于浏览器请求，或者邮件这类要求数据传输可靠性（reliability）的应用，如果使用UDP来传输会存在**两个问题**：

- 数据包在传输过程中容易丢失；

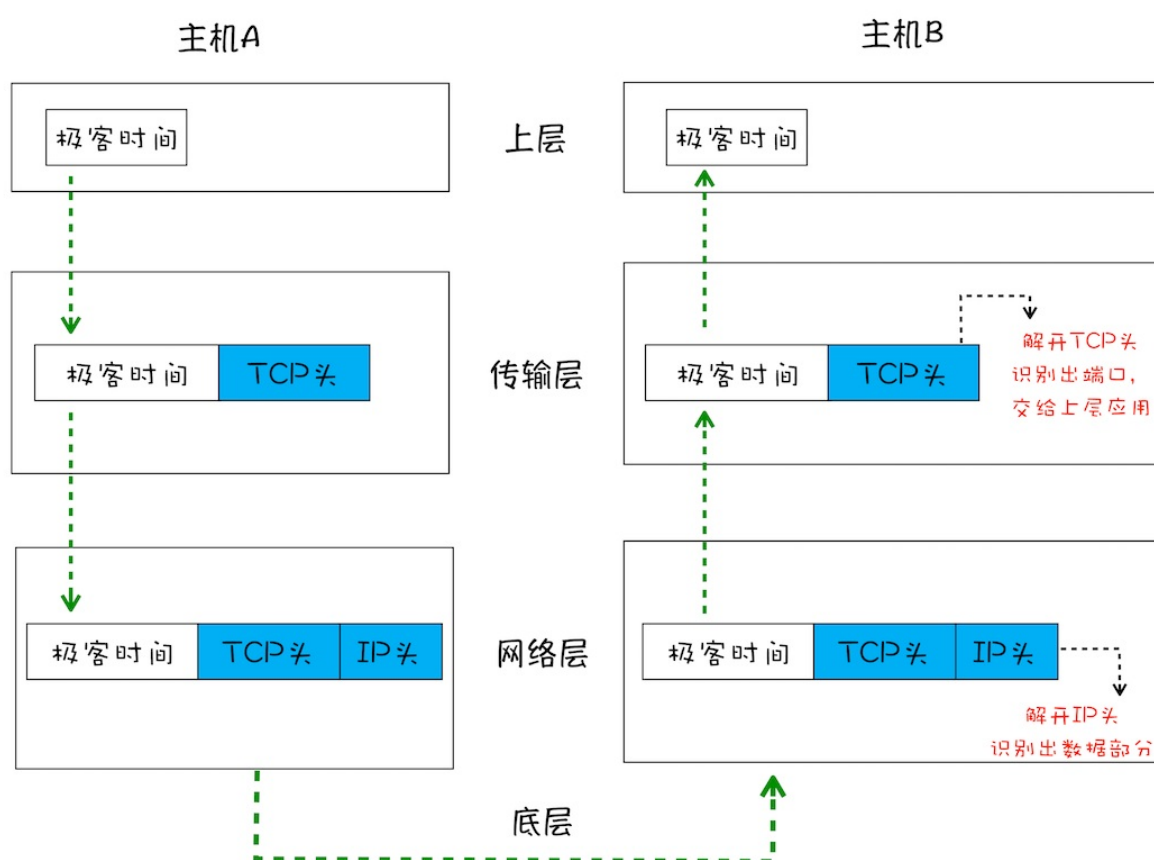
- 大文件会被拆分成很多小的数据包来传输，这些小的数据包会经过不同的路由，并在不同的时间到达接收端，而UDP协议并不知道如何组装这些数据包，从而把这些数据包还原成完整的文件。

基于这两个问题，我们引入TCP了。**TCP（Transmission Control Protocol，传输控制协议）是一种面向连接的、可靠的、基于字节流的传输层通信协议。**相对于UDP，TCP有下面两个特点：

- 对于数据包丢失的情况，TCP提供重传机制；
- TCP引入了数据包排序机制，用来保证把乱序的数据包组合成一个完整的文件。

和UDP头一样，TCP头除了包含了目标端口和本机端口号外，还提供了用于排序的序列号，以便接收端通过序号来重排数据包。

下面看看TCP下的单个数据包的传输流程：

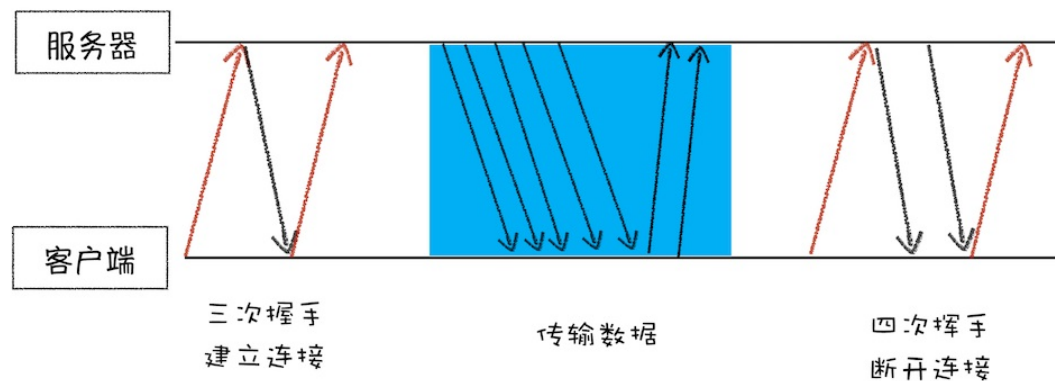


简化的TCP网络四层传输模型

通过上图你应该可以了解一个数据包是如何通过TCP来传输的。TCP单个数据包的传输流程和UDP流程差不多，不同的地方在于，通过TCP头的信息保证了一块大的数据传输的完整性。

下面我们再看下**完整的TCP连接过程**，通过这个过程你可以明白TCP是如何保证重传机制和数据包的排序功能的。

从下图可以看出，一个完整的TCP连接的生命周期包括了“**建立连接**”“**传输数据**”和“**断开连接**”三个阶段。



一个TCP连接的生命周期

- **首先，建立连接阶段。**这个阶段是通过“三次握手”来建立客户端和服务端之间的连接。TCP 提供面向连接的通信传输。**面向连接**是指在数据通信开始之前先做好两端之间的准备工作。所谓**三次握手**，是指在建立一个TCP连接时，客户端和服务端总共要发送三个数据包以确认连接的建立。
- **其次，传输数据阶段。**在该阶段，**接收端需要对每个数据包进行确认操作**，也就是接收端在接收到数据包之后，需要发送确认数据包给发送端。所以当发送端发送了一个数据包之后，在规定时间内没有接收到接收端反馈的确认消息，则判断为数据包丢失，并触发发送端的重发机制。同样，一个大的文件在传输过程中会被拆分成很多小的数据包，这些数据包到达接收端后，接收端会按照TCP头中的序号为其排序，从而保证组成完整的数据。
- **最后，断开连接阶段。**数据传输完毕之后，就要终止连接了，涉及到最后一个阶段“四次挥手”来保证双方都能断开连接。

到这里你应该就明白了，TCP为了保证数据传输的可靠性，牺牲了数据包的传输速度，因为“三次握手”和“数据包校验机制”等把传输过程中的数据包的数量提高了一倍。

总结

好了，这一节就到这里，下面我来做一个简单的总结。

- 互联网中的数据是通过数据包来传输的，数据包在传输过程中容易丢失或出错。
- IP负责把数据包送达目的主机。
- UDP负责把数据包送达具体应用。
- 而TCP保证了数据完整地传输，它的连接可分为三个阶段：建立连接、传输数据和断开连接。

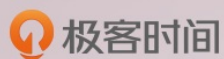
其实了解TCP协议，是为了全方位了解HTTP，包括其实际功能和局限性，之后才会更加深刻地理解为什么要推出HTTP/2，以及为什么要推出QUIC协议，也就是未来的HTTP/3。这是一个由浅入深、循序渐进的过程，希望你能稳扎稳打，学好这每一步、每一个协议，后面“水到自然渠成”。

思考时间

今天这篇文章我没有讲HTTP协议，但是相信你应该听说过，**HTTP协议是基于TCP协议的**，那么今天我留给你的问题是：你怎么理解HTTP和TCP的关系？

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。

精选留言：

• Dongz 2019-08-08 01:37:34

HTTP协议和TCP协议都是TCP/IP协议簇的子集。

HTTP协议属于应用层，TCP协议属于传输层，HTTP协议位于TCP协议的上层。

请求方要发送的数据包，在应用层加上HTTP头以后会交给传输层的TCP协议处理，应答方接收到的数据包，在传输层拆掉TCP头以后交给应用层的HTTP协议处理。建立 TCP 连接后会顺序收发数据，请求方和应答方都必须依据 HTTP 规范构建和解析HTTP报文。[11赞]

作者回复2019-08-08 07:16:48

赞

• 高斯定律 2019-08-08 01:16:40

这个tcp讲的非常清晰 一次就听明白了、tcp是个梯子，http就是利用梯子来搬运货物 [4赞]

作者回复2019-08-08 07:17:31

有点形象

• pyhhou 2019-08-09 02:10:45

请教老师一下，TCP/IP 建立连接和断开连接要经历三次握手和四次挥手，那么 TCP 和 HTTP 建立连接和断开连接是不是也要经历这么一个过程，还是说另有别的考虑？

• ytd 2019-08-08 23:33:48

http架构在tcp之上，主要用在web服务器和浏览器上，它比tcp更加面向开发者。

• 美美 2019-08-08 20:30:21

tcp传送数据时 浏览器端就做渲染处理了么？如果前面数据包丢了 后面数据包先来是要等么？类似的那种实时渲染怎么处理？针对数据包的顺序性？

- hogg 2019-08-08 19:26:25
我记得在网络工程里有一句话,下层为上层提供服务,TCP为HTTP提供差错校验,超时重传的机制吧.

作者回复2019-08-08 20:45:04

下层为上层提供服务，这句很到位

- 琪琪fq 2019-08-08 18:39:30
http协议是超文本协议，浏览器发出http请求，TCP会把请求向底层传递知道web服务器，然后web服务器返回http请求的response，浏览器渲染数据

作者回复2019-08-08 20:45:41

可以这么理解

- 许童童 2019-08-08 11:37:32
你怎么理解 HTTP 和 TCP 的关系？
HTTP是建立在TCP协议之上的，属于应用层，TCP提供给HTTP可靠的连接，HTTP给应用提供更方便的使用接口。

- 高斌 2019-08-08 09:03:01
TCP抽象了HTTP的需求

- XWL 2019-08-08 08:50:03
那丢包一般是什么原因

作者回复2019-08-08 21:28:51

网络问题，线路故障，路由错误等底层的问题都有可能导致丢包

- Snow同學 2019-08-08 08:28:56
好希望更新刚快一些 哈哈

作者回复2019-08-08 10:10:00

一周三更

- 金波 2019-08-08 08:22:59
只是简单罗列了下

作者回复2019-08-08 10:13:14

这节内容主要是熟悉TCP过程，理解了这个流程对理解HTTP有非常大的帮助，TCP如果深入了讲，也是非常复杂的，所以本节的目的，是尽量以最通俗的方式让大家理解TCP过程。

- Dongz 2019-08-08 01:18:25
HTTP协议在网络分层中基于TCP协议之上,和