

02 | TCP协议：如何保证页面文件能被完整送达浏览器？

2019-08-08 李兵

《浏览器工作原理与实践》

课程介绍 >



讲述：李兵

时长 12:08 大小 11.12M



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

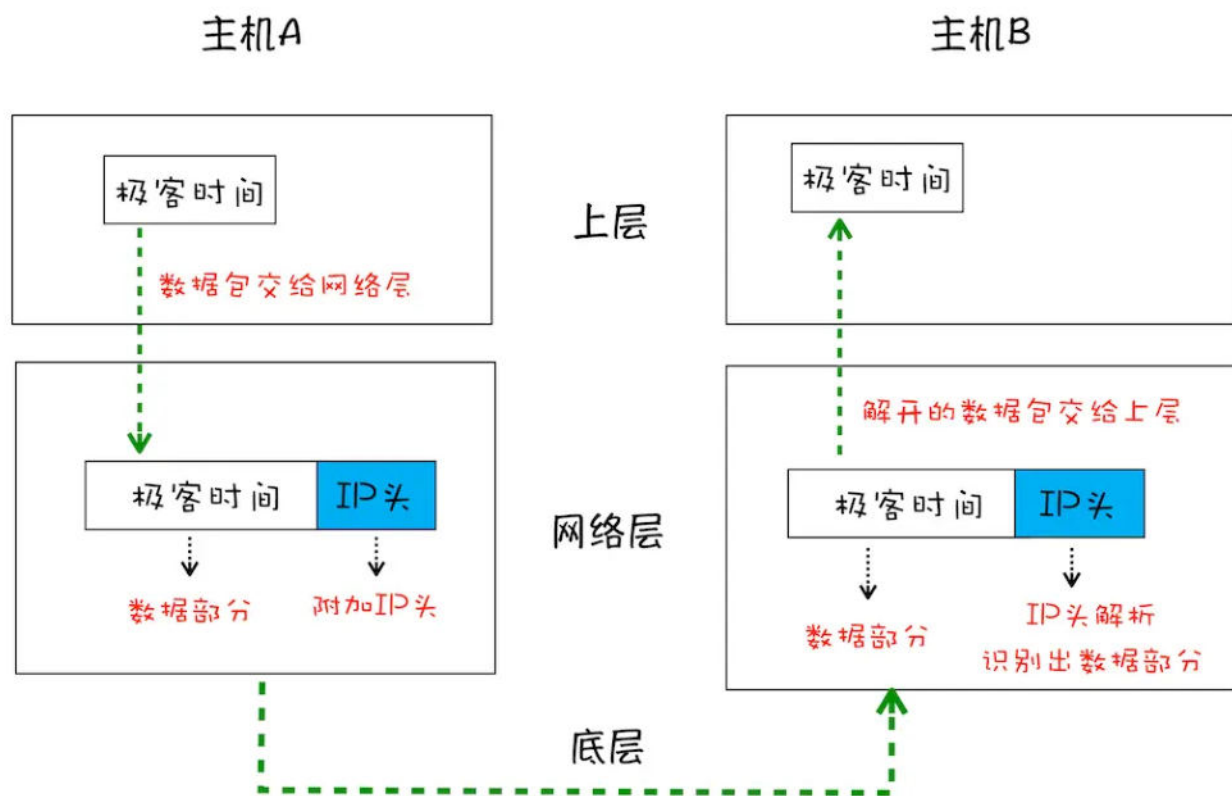
数据包要在互联网上进行传输，就要符合**网际协议**（Internet Protocol，简称 **IP**）标准。互联网上不同的在线设备都有唯一的地址，地址只是一个数字，这和大部分家庭收件地址类似，你只需要知道一个家庭的具体地址，就可以往这个地址发送包裹，这样物流系统就能把物品送到目的地。

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：



简化的 IP 网络三层传输模型

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。

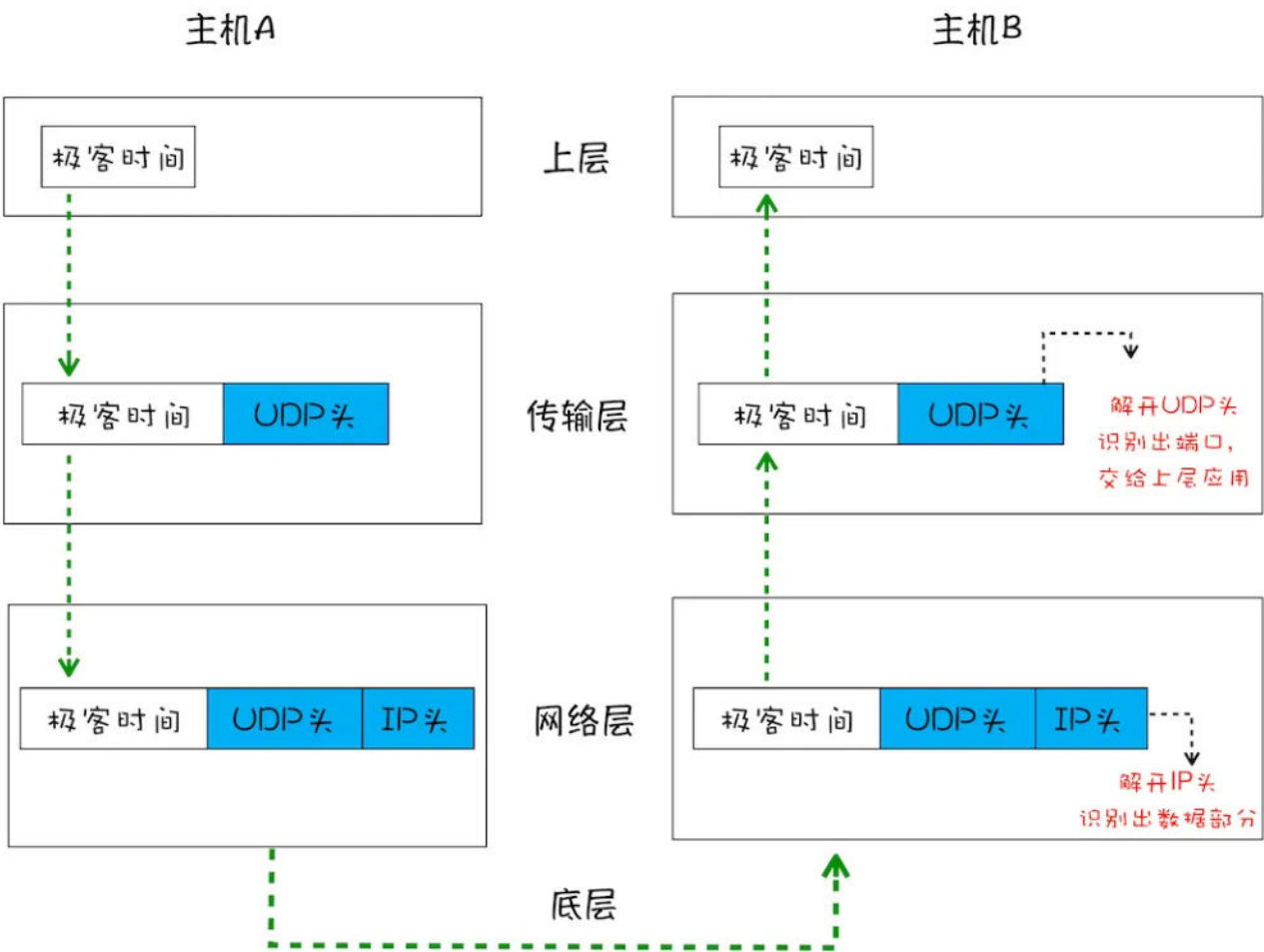
2. UDP：把数据包送达应用程序

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附加上 **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。

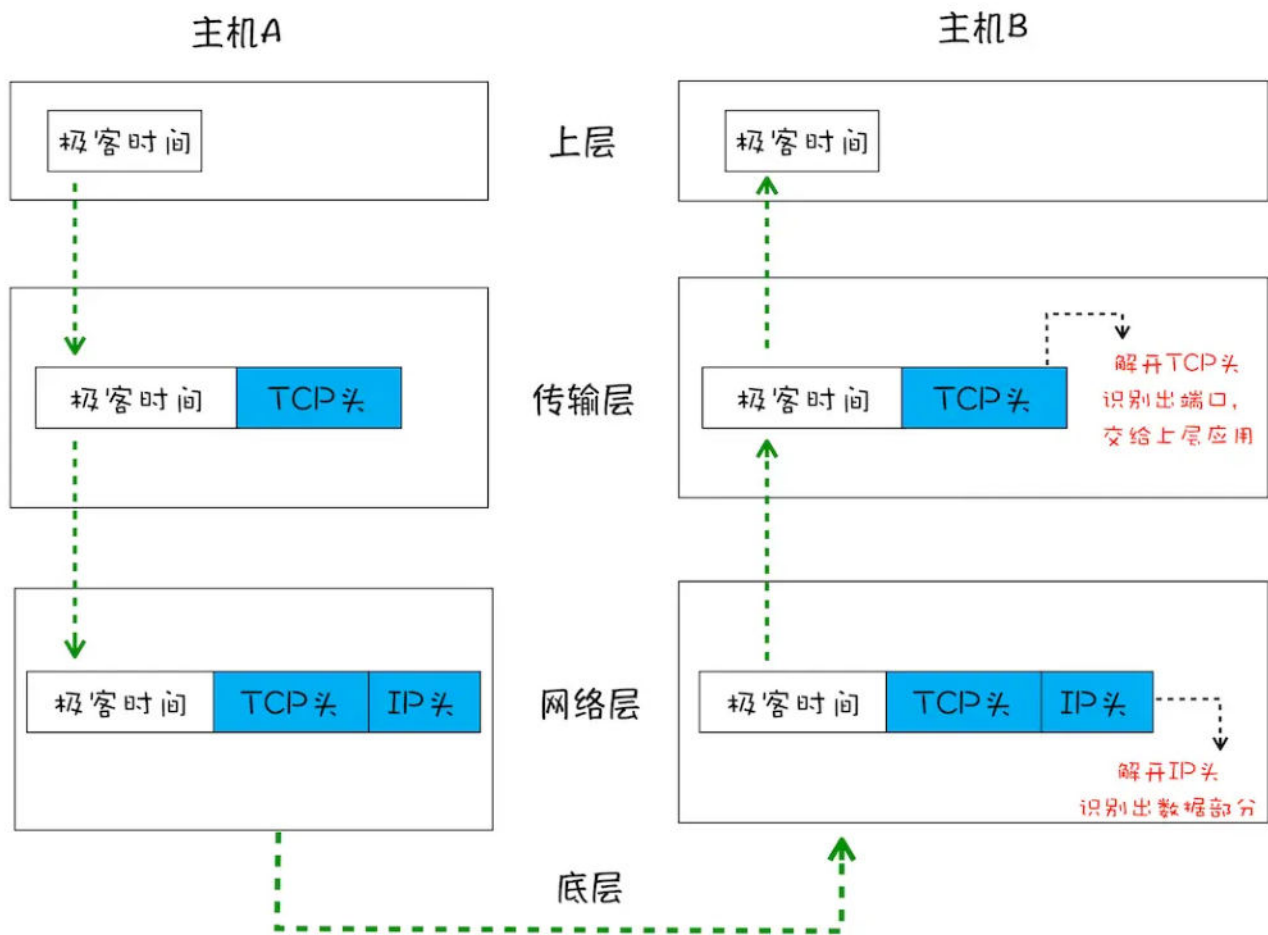
基于这两个问题，我们引入 TCP 了。**TCP（Transmission Control Protocol，传输控制协议）是一种面向连接的、可靠的、基于字节流的传输层通信协议。**相对于 UDP，TCP 有下面两个特点：

- 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：



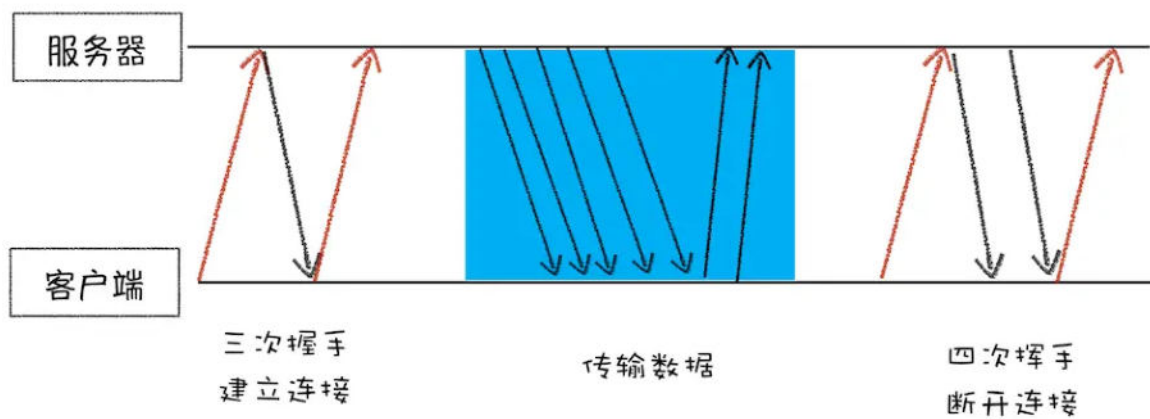
简化的 TCP 网络四层传输模型

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，我希望你能稳扎稳打，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。

分享给需要的人，Ta 订阅超级会员，你将得 50 元

Ta 单独购买本课程，你将得 20 元

 生成海报并分享

 赞 72

 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 01 | Chrome 架构：仅仅打开了 1 个页面，为什么有 4 个进程？

下一篇 03 | HTTP 请求流程：为什么很多站点第二次打开速度会很快？



JVM + NIO + Spring

各大厂面试题及知识点详解

限时免费



精选留言 (85)

写留言



Dongz

2019-08-08

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。

作者回复: 赞

共 4 条评论 >

187



白糖炒栗子~

2020-04-06

在评论区也学到了很多，小总结：

1. 浏览器可以同时打开多个页签，他们端口一样吗？如果一样，数据怎么知道去哪个页签？

端口一样的，网络进程知道每个tcp链接所对应的标签是那个，所以接收到数据后，会把数据分发给对应的渲染进程。



2. 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？

接收到http响应头中的content-type类型时就开始准备渲染进程了，响应体数据一旦接受到便开始做DOM解析了！基于http不用担心数据包丢失的问题，因为丢包和重传都是在tcp层解决的。http能保证数据按照顺序接收的（也就是说，从tcp到http的数据就已经是完整的了，即便是实时渲染，如果发生丢包也得在重传后才能开始渲染）

3. http 和 websocket都是属于应用层的协议吗？

都是应用层协议，而且websocket名字取的比较有迷惑性，其实和socket完全不一样，可以把websocket看出是http的改造版本，增加了服务器向客户端主动发送消息的能力。

4. 关于 "数据在传输的过程中有可能会丢失或者出错"，丢失的数据包去哪里了？凭空消失了吗？出错的数据包又变成啥了？为什么会出错？

比如网络波动，物理线路故障，设备故障，恶意程序拦截，网络阻塞等等

共 4 条评论 >

👍 109



sheldon

2019-08-09

现在的浏览器可以同时打开多个页签，他们端口一样吗？如果一样，数据怎么知道去哪个页签？

作者回复: 端口一样的，网络进程知道每个tcp链接所对应的标签是那个，所以接收到数据后，会把数据分发给对应的渲染进程

共 9 条评论 >

👍 87



Cris

2019-08-09

http 和 websocket都是属于应用层的协议吗？

作者回复: 是的，他们都是应用层协议，而且websocket名字取的比较有迷惑性，其实和socket完全不一样，你可以把websocket看出是http的改造版本，增加了服务器向客户端主动发送消息的能力。



👍 44



美美

2019-08-08

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？



作者回复: 接收到http响应头中的content-type类型时就开始准备渲染进程了,

响应体数据一旦接受到便开始做DOM解析了!

基于http不用担心数据包丢失的问题, 因为丢包和重传都是在tcp层解决的。http能保证数据按照顺序接收的!

共 3 条评论 >

👍 43



高斯定律

2019-08-08

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

作者回复: 有点形象

共 3 条评论 >

👍 31



江霖

2020-07-23

谢谢李冰老师, 断断续续的花了很长时间终于要学完了, 回顾完整个学习过程, 感触颇多也和老师一样来做个总结和同学们分享:

1. 首先知识方面的收获:

学习完整个课程之后我对浏览器的整体架构和如何运作的有了一个宏观的理解, 明白了浏览器是由哪些部分构成, 这些部分是怎么配合来完成任务以及浏览器是如何与服务器交互的。

这让我的前端后端散装知识能够联系到一起, 形成了一张知识网, 感觉更容易记忆了

在一些比较重要的知识点上如页面的渲染, javascript的运行机制和网络安全等方面跟随老师的文章进行深度的挖掘, 对这些知识有了更深层次的理解, 掌握程度更高, 工作中遇到的比较难解决的问题时也能很快的有一个清晰的思路

2. 如何学习一门知识的思路

以前我的学习方式是贪多而且杂, 分不清主次, 所有的东西搅在一起, 学了不少但始终感觉没法融会贯通和更深入的理解, 多数知识只是停留到如何应用这一层

学习了老师的课程中知识的同时也学习如何学习一门知识的思路:

- 1) 搭建整个知识的框架(体), 化繁为简对这门知识有个宏观整体的理解
- 2) 对于框架的每个部分进行拆分相关的知识点拆分成一个系列(面)
- 3) 之后再深入挖掘每个系列中的每个知识点(线)
- 4) 在每个知识点中分析其原因, 解决的问题, 历史, 定义等(点), 并使用简练有逻辑的语言配合图表将每个点讲清楚
- 5) 每章总结, 化繁为简, 梳理出关键的知识点形成记忆的主干, 配合框架让知识形成树状结



构。知识体系是根->每章总结是主干->讲解的知识点原因，历史等是叶子

6) 每章后的思考，学而不思则罔，思而不学则殆。对知识进行更进一步的思考，能够加深对知识的理解并且检查是否有欠缺的地方，计划下一步的学习方案等

3. 如何将知识写成文章

首先对整个知识需要有深度的了解

文章的思路就是学习的思路

考虑受众进行整体设计，确定文章的定位和目标，懂得舍弃，杂项影响主干完成的知识点作为加餐补充

使用总分的形式拆分章节

使用写作技巧多用图表，转折链接各个章节，简练的语言

说实话我是个喜欢白嫖的程序员，对于知识付费一向不以为然，这门课程是我的第一个网课，完全改变了我对知识付费的看法。

课程真的性价比超高很多东西真的不是靠自己短时间学习能够掌握的，跟着网课系统学习是一种很高效的学习方式，跟着老司机不翻车，之后感兴趣的话可以深度学习也成为一名合格的老司机

希望学习到的东西能够应用到工作中，学习知识和记录笔记等方面

感谢老师，祝老师越写越好，给我们带来更多更精彩的课程

共 2 条评论 >

👍 30



黄紫茜

2019-08-22

我想问下，输入url回车后理论上是不是生成http请求报文，然后传给TCP，加上TCP首部，然后进行三次握手，将http请求报文数据传递，四次挥手，请求报文到达服务端，然后服务端在返回响应报文，返回到过程也是要三次握手，传递数据，四次挥手？因为网上看到不少文章都说先三次握手，然后发起http请求。

作者回复：

先通过三次握手建立tcp链接，链接建立好之后，发送http请求行和http请求头给服务器，然后服务器返回响应行，响应头和响应体，最终完成后通过四次挥手断开tcp链接！

共 2 条评论 >

👍 27



hogger

2019-08-08

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: 下层为上层提供服务, 这句很到位

共 2 条评论 >

👍 18



我是辣妈

2019-08-10

- 1、IP 负责把数据包送达目的主机。
- 2、UDP 负责把数据包送达具体应用。
- 3、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。

老师, 这里面第二条, UDP和TCP都是把数据包送达具体应用应用的吧?

作者回复: 对的, 只不过udp有可能会丢包, 而tcp采用了很多复杂的机制, 保证了数据的完整性



👍 15



Cris

2019-09-02

老师, tcp和udp都是传输层协议, 这两个是同时存在还是只能选其一, 我看了这期感觉是选其一?

作者回复: 二选一



👍 11



尔冬橙

2019-08-26

老师, 那我打开谷歌浏览器里面开了10个页面, 那就是至少有40个进程? windows下我控制面板里能看到40个google.exe? 插件进程页面间能公用么

作者回复: 不用这么多啊, 浏览器进程 网络进程, GPU进程都是共同的, 你打开十个页面一般也就十几个进程。

共 3 条评论 >

👍 11



sam

2019-08-21

这篇文章可以当作平时购物的流程就很清晰的理解:

数据包： 我们可以理解为我们买的東西；

TCP/UDP头： 买卖人电话、姓名等；

IP头： 买卖双方地址；

TCP和UDP不同点在于，TCP会把购买的物品直接送到购买人手中，而UDP可能只是送到蜂巢如此之类的。

不过我有个疑问，三次握手的过程当中是否也是需要走三次传输流程？

作者回复: 对的

共 2 条评论 >

👍 10



阿桐

2019-08-22

文章前面从网络层引到传输层，再从 udp 过渡到 tcp，语言通俗易懂，图示也很清晰，看的意犹未尽。相比之下，后面的 【一个 tcp 连接的生命周期】老师用的笔墨就少了点，其实还是蛮期待老师能对这块有更为细致的讲解。

作者回复: 嗯，tcp细节要讲就多了，这块我只是介绍了tcp的一个大的建立连接，传输数据，断开链接的过程。理解这个过程是理解http的基础。展开来讲的话需要一节内容了，这块极客时间上有很多老师都讲的不错。



👍 6



レイン小雨

2019-08-11

想问一下老师，关于 "数据在传输的过程中有可能会丢失或者出错"，丢失的数据包去哪里了？凭空消失了吗？出错的数据包又变成啥了？为什么会出错？

作者回复: 比如网络波动，物理线路故障，设备故障，恶意程序拦截，网络阻塞等等

共 6 条评论 >

👍 5



HoSalt

2020-04-09

老师你好，有些问题想问下

『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』

1. 基于字节流是什么意思？

2. UDP不是基于字节流吗，那UDP是基于什么？



3. TCP性能没有UDP好，消耗的性能更多是不是用在了消息确认这套机制上，而非三次握手和四次挥手上，难道建立链接和断开链接很费时吗？平时应用中都是说通过减少http的连接数来提升性能

共 2 条评论 >

👍 3



Cshine 🌸🌸

2020-05-06

感觉很清晰，层层递减，打call！！



👍 2



学习

2020-01-12

IP协议是高速公路，TCP协议是货车，HTTP协议是货物。

共 1 条评论 >

👍 2



レイン小雨

2019-08-11

http协议是应用层的协议，是最上层的网络协议，tcp / udp是两种位于传输层的协议。http协议位于tcp / upd上层。



👍 2



snow

2020-08-27

如果客户端向服务器请求一个数据，建立了tcp连接，服务器发送了数据，中间断网了，服务器没有收到客户端的回复消息，然后继续重发，如果一直断网，服务器就一直重发吗？还是说tcp就能够检测到网络不通了，或者人家已经关闭应用了？



👍 1

