

# HTTP1.0和HTTP1.1的区别

## 1.长连接

HTTP1.0使用非持久连接，一个TCP连接只能发送一个HTTP请求和响应，完成后自动断开连接

HTTP1.1支持持久连接（长连接），一个TCP连接长时间持续有效，可以传送多个HTTP请求和响应

如何建立长连接

HTTP1.0需要在HTTP头加入Connection：Keep-Alive才可以启用长连接，HTTP1.1默认启动了长连接通过给Keep-Alive

赋值close可以进行关闭。

优点：减少了建立和关闭连接的消耗和延迟。

## 2.缓存控制

在HTTP1.0中主要使用header里的If-Modified-Since, Expires来作为缓存判断的标准, HTTP1.1则引入了更多的缓存控制策略例如Entity tag, If-Unmodified-Since, If-Match, If-None-Match等更多可供选择的缓存头来控制缓存策略.

## 3.节约带宽

在HTTP头部引入了range字段，可以用于告知服务器只返回文件的某个部分。在一个 Range 首部中，可以一次性请求多个部分，服务器会以 multipart 文件的形式将其返回。如果服务器返回的是范围响应，需要使用 206Partial Content 状态码。假如所请求的范围不合法，那么服务器会返回416.

Range: bytes=start-end

1

另外一个节约带宽的方法就是压缩要传送的数据。Content-Encoding列出了对当前实体消息（消息荷载）应用的任何编码类型，以及编码的顺序。在请求消息中加入Accept-Encoding头域，它可以告诉服务器客户端能够解码的编码方式。

## 4.Host域

HTTP1.0中默认每台服务器都绑定唯一的一个IP地址,所以请求消息中url并没有传递主机名,也就是hostname，但后面的虚拟主机技术使得一个物理服务器上面可以存在多个虚拟主机，他们共享一个ip地址。

HTTP1.1的请求消息和响应消息都应支持Host头域，且请求消息中如果没有Host头域会报告一个错误（400 Bad Request）。此外，服务器应该接受以绝对路径标记的资源请求。

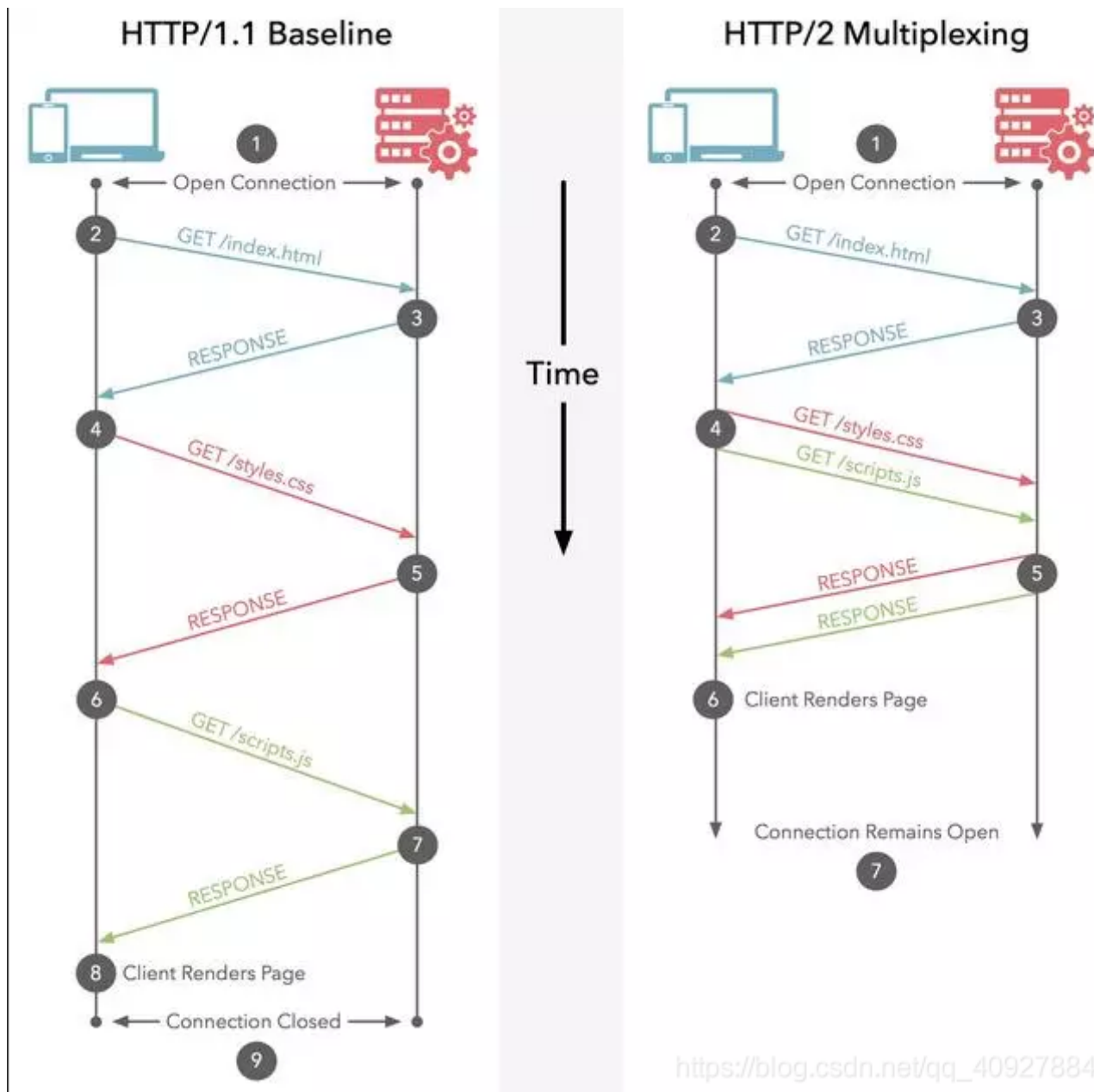
## 5.错误提示

http1.1版本新增了24个错误状态响应码。

# http1.0和http2.0的区别

## 1、多路复用和二进制分帧

多路复用允许单一的 HTTP/2 连接同时发起多重的请求-响应消息。看个例子：



[https://blog.csdn.net/qq\\_40927884](https://blog.csdn.net/qq_40927884)

从这张图可以看出，请求index.html页面的时候，浏览器会去请求style.css和scripts.js的文件。左边的图是顺序加载两个文件的，右边则是并行加载两个文件。

我们知道，TCP连接相当于两根管道（双工，一个用于服务器到客户端，一个用于客户端到服务器），管道里面数据传输是通过字节码传输，传输是有序的，每个字节都是一个一个来传输。

例如客户端要向服务器发送Hello、World两个单词，只能是先发送Hello再发送World，没办法同时发送这两个单词。不然服务器收到的可能就是HWeolrlod（注意是穿插着发过去了，但是顺序还是不会乱）。这样服务器就懵b了。

接上面的问题，能否同时发送Hello和World两个单词能，当然也是可以的，可以将数据拆成包，给每个包打上标签。发的时候是这样的①H ②W ①e ②o ①l ②r ①l ②l ①o ②d。这样到了服务器，服务器根据标签把两个单词区分开来。实际的发送效果如下图：



One HTTP/2 Connection's multiplexing

二进制分帧可以实现上面的效果，二进制分帧层在应用层(HTTP/2)和传输层(TCP or UDP)之间。在二进制分帧层中，HTTP/2 会将所有传输的信息分割为帧 (frame)，并对它们采用二进制格式的编码，其中首部信息会被封装到 HEADER frame，而相应的 Request Body 则封装到 DATA frame 里面。

HTTP 性能优化的关键并不在于高带宽，而是低延迟。TCP 连接会随着时间进行自我「调谐」，起初会限制连接的最大速度，如果数据成功传输，会随着时间的推移提高传输的速度。这种调谐则被称为 TCP 慢启动。由于这种原因，让原本就具有突发性和短时性的 HTTP 连接变的十分低效。

HTTP/2 通过让所有数据流共用同一个连接，可以更有效地使用 TCP 连接，让高带宽也能真正的服务于 HTTP 的性能提升。

## 2、首部压缩

假定一个页面有80个资源需要加载（这个数量对于今天的Web而言还是挺保守的），而每一次请求都有1400字节的消息头（着同样也并不少见，因为Cookie和引用等东西的存在），至少要7到8个来回去“在线”获得这些消息头。这还不包括响应时间——那只是从客户端那里获取到它们所花的时间而已。这全都由于TCP的慢启动机制，它会基于对已知有多少个包，来确定还要来回去获取哪些包 – 这很明显的限制了最初的几个来回可以发送的数据包的数量。相比之下，即使是头部轻微的压缩也可以是让那些请求只需一个来回就能搞定——有时候甚至一个包就可以了。这种开销是可以被节省下来的，特别是当你考虑移动客户端应用的时候，即使是良好条件下，一般也会看到几百毫秒的来回延迟。

为解决这个问题，支持 HTTP/2 的浏览器和服务端之间会压缩http报文头部，具体原理是：

- 1、维护一份相同的静态字典（Static Table），包含常见的头部名称，以及特别常见的头部名称与值的组合；
- 2、维护一份相同的动态字典（Dynamic Table），可以动态的添加内容；
- 3、支持基于静态哈夫曼码表的哈夫曼编码（Huffman Coding）；

静态字典的作用有两个：

- 1) 对于完全匹配的头部键值对，例如“:method :GET”，可以直接使用一个字符表示；
- 2) 对于头部名称可以匹配的键值对，例如“cookie :xxxxxxx”，可以将名称使用一个字符表示。

同时，浏览器和服务端都可以向动态字典中添加键值对，之后这个键值对就可以使用一个字符表示了。需要注意的是，动态字典上下文有关，需要为每个 HTTP/2 连接维护不同的字典。在传输过程中使用，使用字符代替键值对大大减少传输的数据量。

## 3、服务器的主动推送机制

当浏览器请求一个网页时，服务器将会发回HTML，在服务器开始发送JavaScript、图片和CSS前，服务器需要等待浏览器解析HTML和发送所有内嵌资源的请求。服务器推送服务通过“推送”那些它认为客户

端将会需要的内容到客户端的缓存中，以此来避免往返的延迟。

http1.0和http2.0区别概括：

- 1、HTTP/2采用二进制格式而非文本格式
- 2、HTTP/2是完全多路复用的，而非有序并阻塞的——只需一个连接即可实现并行
- 3、使用报头压缩，HTTP/2降低了开销
- 4、HTTP/2让服务器可以将响应主动“推送”到客户端缓存中

### **http1.1的长连接和http2.0的多路复用到底有啥区别？**

长连接：同一个域名访问同一个文件的多个请求都可以复用同一个tcp连接（不用像1.0一样 每次请求都需要重新建立连接）

依然存在的问题：1.多个请求只能被串行处理（数据基于文本，只能按顺序传输）；2.访问多个不同的文件依然会建立多个请求。

多路复用：同一个域名访问多个文件的请求也可以复用同一个tcp连接，且多个请求可以被并行处理。

并行实现原理：http2.0引入二进制数据帧和流的概念（数据帧对每一个数据进行标识，可以不按顺序传输，从而实现并行）

---

版权声明：本文为CSDN博主「o3otxl」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/o3otxl/article/details/108573563>