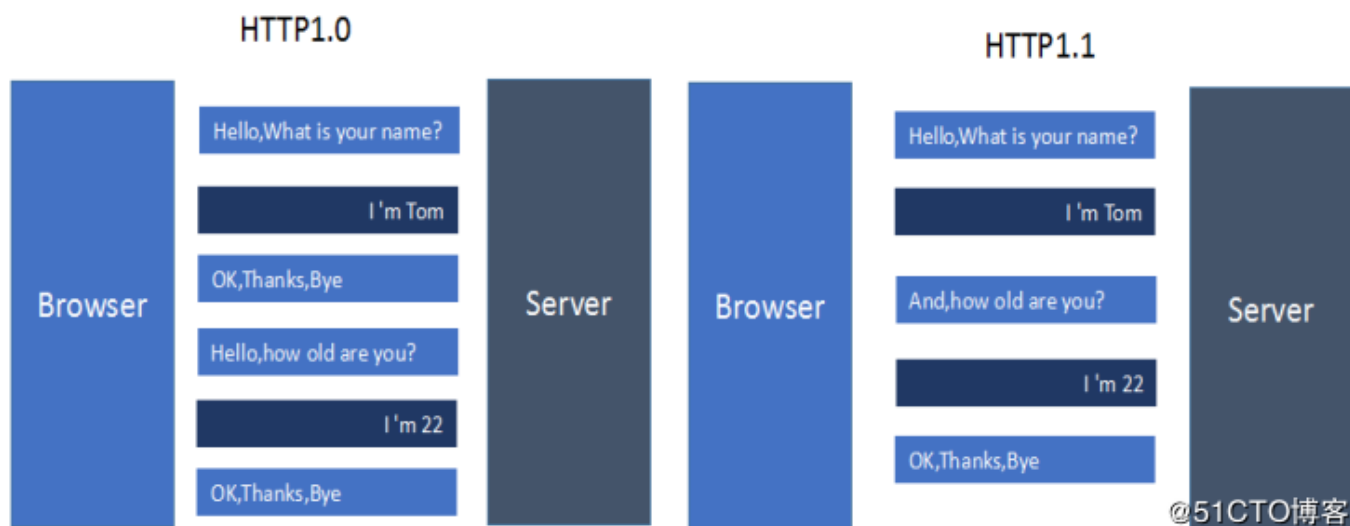


HTTP 1.0、1.1区别

HTTP1.0和HTTP1.1区别

版本	HTTP1.0	HTTP1.1
连接方面	使用 非持久连接，即在非持久连接下，一个tcp连接只传输一个web对象。每次请求和响应都需要建立一个单独的连接，每次连接只是传输一个对象，严重影响客户机和服务器的性能	默认使用持久连接(然而,HTTP/1.1协议的客户机和服务器可以配置成使用非持久连接)在持久连接下,不必为每个Web对象的传送建立一个新的连接,一个连接中可以传输多个对象。在一个TCP连接上可以传送多个HTTP请求和响应，减少了建立和关闭连接的消耗和延迟。HTTP 1.1的持续连接，也需要增加新的请求头来帮助实现，例如，Connection请求头的值为Keep-Alive时，客户端通知服务器返回本次请求结果后保持连接；Connection请求头的值为close时，客户端通知服务器返回本次请求结果后关闭连接。HTTP 1.1还提供了与身份认证、状态管理和Cache缓存等机制相关的请求头和响应头。并且不允许同时存在两个并行的响应。
缓存方面	主要使用header里的If-Modified-Since,Expires来做为缓存判断的标准	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带宽优化及网络连接的使用
状态码	无状态	新增了24个错误状态响应码，如409（Conflict）表示请求的资源与资源的当前状态发生冲突；410（Gone）表示服务器上的某个资源被永久性的删除
宽带优化	存在一些浪费带宽的现象，例如客户端只是需要某个对象一部分，而服务器却将整个对象送过来了，并且不支持断点续传功能	支持只发送header信息(不带任何body信息)，如果服务器认为客户端有权限请求服务器，则返回100，否则返回401。客户端如果接收到100，才开始把请求body发送到服务器。这样当服务器返回401的时候，客户端就可以不用发送请求body了，节约了带宽。
Host头	HTTP1.0中认为每台服务器都绑定一个唯一的IP地址，因此，请求消息中的URL并没有传递主机名(hostname)。但随着虚拟主机技术的发展，在一台物理服务器上可以存在多个虚拟主机（Multi-homed Web Servers），并且它们共享一个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）

HTTP1.0和HTTP1.1区别预览图



HTTP1.0和HTTP1.1现存的问题

1. HTTP1.x在传输数据时，每次都需要重新建立连接，无疑增加了大量的延迟时间，特别是在移动端更为突出
2. HTTP1.x在传输数据时，所有传输的内容都是明文，客户端和服务端都无法验证对方的身份，无法保证数据的安全性
3. HTTP1.x在使用时，header里携带的内容过大，增加了传输的成本，并且每次请求header基本不怎么变化，尤其在移动端增加用户流量
4. 虽然HTTP1.x支持了keep-alive，来弥补多次创建连接产生的延迟，但是keepalive使用多了同样会给服务端带来大量的性能压力，并且对于单个文件被不断请求的服务(例如图片存放网站)，keep-alive可能会极大的影响性能，因为它在文件被请求之后还保持了不必要的连接很长时间

HTTPS

为解决安全问题，网景在1994年创建了HTTPS。HTTPS就是安全版的HTTP，目前大型网站基本实现全站HTTPS

HTTPS简介

1. HTTPS协议需要到CA申请证书，一般免费证书很少，需要交费
2. HTTP协议运行在TCP之上，所有传输的内容都是明文，HTTPS运行在SSL/TLS之上，SSL/TLS运行在TCP之上，所有传输的内容都经过加密的
3. HTTP和HTTPS使用的是不同的连接方式，端口不同，前者是80，后者是443
4. HTTPS可以有效的防止运营商劫持，解决了防劫持的一个大问题
5. HTTPS 中的SSL握手等过程降低用户访问速度，但是只要经过合理优化和部署，HTTPS 对速度的影响完全可以接受

SPDY

SPDY：2009年,谷歌研发，综合HTTPS和HTTP两者有点于一体的传输协议

SPDY主要特点

1. 降低延迟，针对HTTP高延迟的问题，SPDY优雅的采取了多路复用（multiplexing）。多路复用通过多个请求stream共享一个tcp连接的方式，解决了HOL blocking的问题，降低了延迟同时提高了带宽的利用率
2. 请求优先级（request prioritization）。多路复用带来一个新的问题是，在连接共享的基础之上有可能导致关键请求被阻塞。PDY允许给每个request设置优先级，重要的请求就会优先得到响应。比如浏览器加载首页，首页的html内容应该优先展示，之后才是各种静态资源文件，脚本文件等加载，可以保证用户能第一时间看到网页内容
3. header压缩。HTTP1.x的header很多时候都是重复多余的。选择合适的压缩算法可以减小包的大小和数量
4. 基于HTTPS的加密协议传输，大大提高了传输数据的可靠性
5. 服务端推送（server push），采用了SPDY的网页，例如网页有一个style.css的请求，在客户端收到style.css数据的同时，服务端会将style.js的文件推送给客户端，当客户端再次尝试获取style.js时就可以直接从缓存中获取到，不用再发请求了

HTTP2.0

HTTP2.0是SPDY的升级版

HTTP2.0特性

- 1. 头信息和数据体都是二进制，称为头信息帧和数据帧
- 2. 复用TCP连接，在一个连接里，客户端和浏览器都可以同时发送多个请求或回应，且不用按顺序一一对应，避免了“队头堵塞”，此双向的实时通信称为多工（Multiplexing）
- 3. 引入头信息压缩机制（header compression），头信息使用gzip或compress压缩后再发送；客户端和服务端同时维护一张头信息表，所有字段都会存入这个表，生成一个索引号，不发送同样字段，只发送索引号，提高速度
- 4. HTTP/2 允许服务器未经请求，主动向客户端发送资源，即服务器推送（server push）

HTTP2.0和SPDY的区别

- 1. HTTP2.0 支持明文 HTTP 传输，而 SPDY 强制使用 HTTPS
- 2. HTTP2.0 消息头的压缩算法采用 HPACK，而非 SPDY 采用的 DEFLATE

HTTP1.1和HTTP2.0的区别

版本	HTTP1.1	HTTP2.0
多路复用	在HTTP/1.1协议中，浏览器客户端在同一时间针对同一域名的请求有一定数据限制。超过限制数目的请求会被阻塞	HTTP2.0使用了多路复用的技术，做到同一个连接并发处理多个请求，而且并发请求的数量比HTTP1.1大了好几个数量级。而这个强大的功能则是基于“二进制分帧”的特性。
首部压缩	不支持header数据的压缩	使用HPACK算法对header的数据进行压缩，这样数据体积小了，在网络上传输就会更快
服务器推送		当我们对支持HTTP2.0的web server请求数据的时候，服务器会顺便把一些客户端需要的资源一起推送到客户端，免得客户端再次创建连接发送请求到服务器端获取。这种方式非常适合加载静态资源