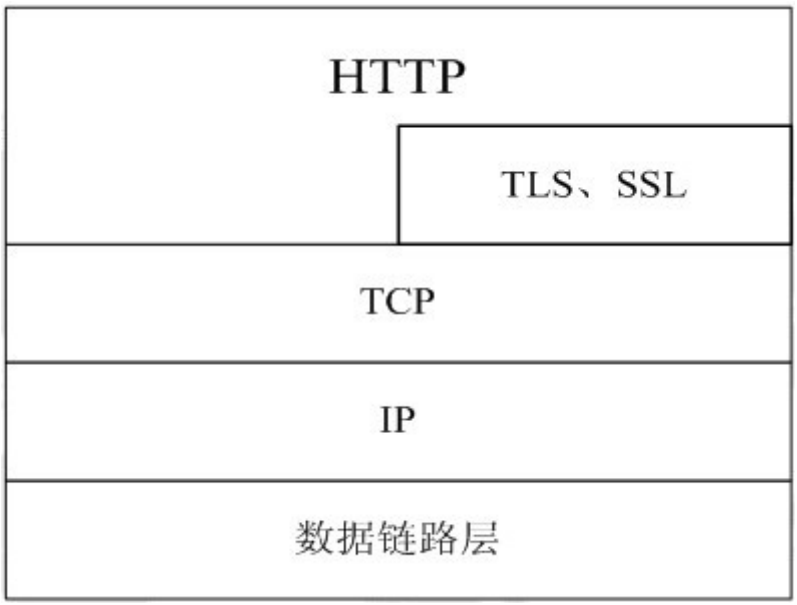
HTTP协议

笔记本：	HTTP协议		
创建时间：	2018/5/10 0:14	**更新时间：**	2018/5/10 9:32
作者：	ly		

HTTP协议

超文本传输协议（HyperText Transfer Protocol）是互联网上应用最为广泛的一种网络协议。版本：HTTP1.1
HTTP协议通常承载与TCP协议上，有时也承载与TLS或SSL协议层之上。
HTTPS：



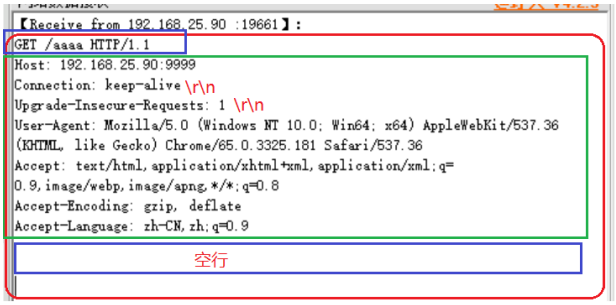
请求/响应（Request/Response）模型

HTTP协议是一种无状态的协议，以及客户端和服务器端不需要建立持久的链接。

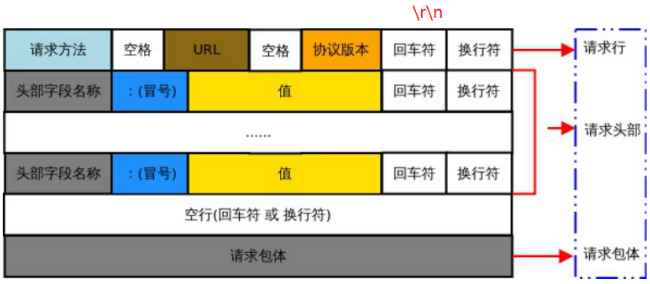
无状态协议：对于事务处理没有记忆能力，服务器不知道客户端是什么状态。也就是说，打开一个服务器上的网页和你之前打开这个服务器上的网页之间没有任何联系。

请求报文

报文示例：



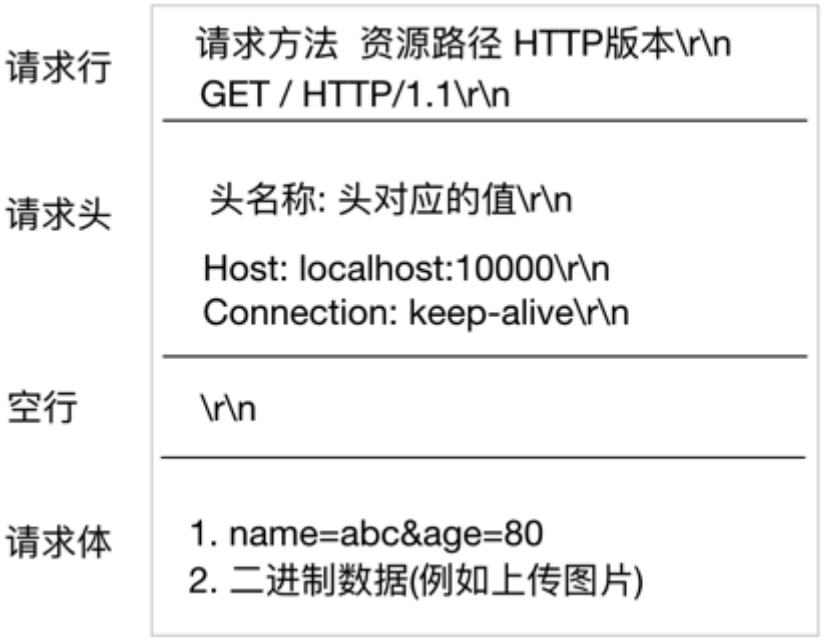
上网输入网址，一般情况下，请求包体没有的




```
GET /aaaa HTTP/1.1
Host: 192.168.25.90:9999
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=
0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
```

请求报文格式说明：

HTTP请求报文由请求行，请求头部，空行，请求包体4个部分组成。



请求行：由**方法字段**，**URL字段**和**HTTP协议版本字段**3部分组成。 常用的HTTP请求方法有GET, POST.

使用GET方法时，请求参数和对应的值附加在URL后面，利用一个问号（ “？” ）代表URL的结尾与请求参数的开始，传递参数长度受限制，因此GET方法不适合用于上传数据。通过GET方法来获取网页时，参数会显示在浏览器地址栏上，因此保密性很差。

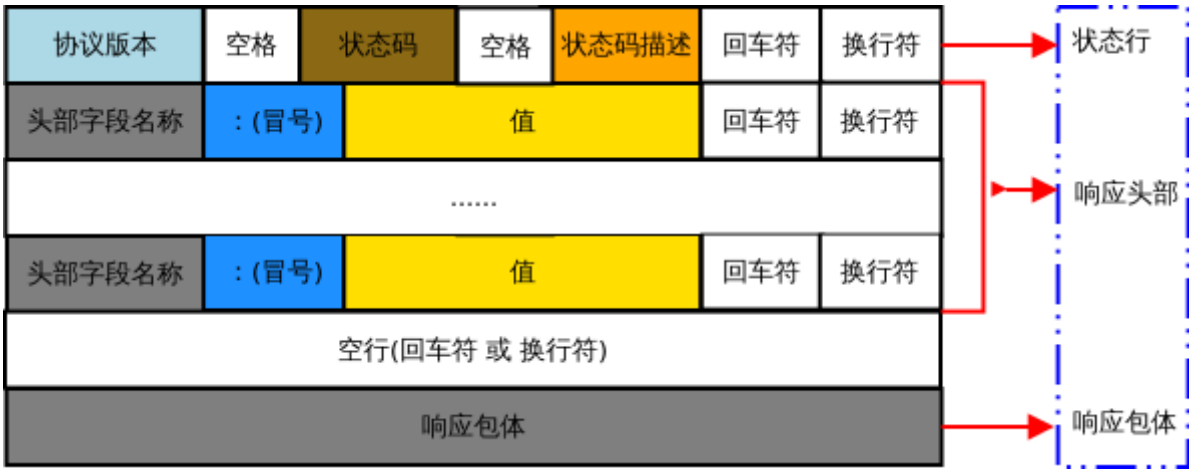
使用POST方法时，当客户端给服务器提供信息较多时可以使用POST方法，POST方法向服务器提交数据，比如完成表单数据的提交，将数据提交给服务器处理，长度没有限制，因为POST携带的数据，在HTTP的请求正文中，以名称/值的形式出现，可以传输大量数据。

空行：最后一个请求头之后是一个空行，发送回车符和换行符，通知服务器以下不再有请求头。

请求包体：请求包体不再GET方法中使用，而是POST方法中使用。

响应报文格式说明

HTTP响应报文由**状态行**，**响应头部**，**空行**，**响应包体**4个部分组成



示例

```
▼ Response Headers      view parsed
  HTTP/1.1 200 OK
  Bdpagetype: 1
  Bdqid: 0xc521e78f000022d7
  Cache-Control: private
  Connection: Keep-Alive
  Content-Encoding: gzip
  Content-Type: text/html; charset=utf-8
  Cxy_all: baidu+b4645d86e41fa4eeb56295d99da52cb3
  Date: Thu, 26 Apr 2018 01:52:07 GMT
  Expires: Thu, 26 Apr 2018 01:51:14 GMT
  Server: BWS/1.1
  Set-Cookie: BDSVRTM=0; path=/
  Set-Cookie: BD_HOME=0; path=/
  Set-Cookie: H_PS_PSSID=1993_1465_21107_22160; path=/; domain=.baidu.com
  Strict-Transport-Security: max-age=172800
  Vary: Accept-Encoding
  X-Powered-By: HPHP
  X-Ua-Compatible: IE=Edge,chrome=1
  Transfer-Encoding: chunked
```

状态行由HTTP**协议版本字段**，**状态码**和**状态码的描述文本**3个部分组成，他们之间使用空格隔开。

状态码：状态码由三位数字组成，第一位数字表示响应的类型，常用的状态码有五大类如下所示：

状态码	含义
1XX	表示服务器已接收了客户端请求，客户端可继续发送请求

2XX	表示服务器已成功接收到请求并进行处理
3XX	表示服务器要求客户端重定向
4XX	表示客户端的请求有非法内容
5XX	表示服务器未能正常处理客户端的请求而出现意外错误

常见的状态码举例：

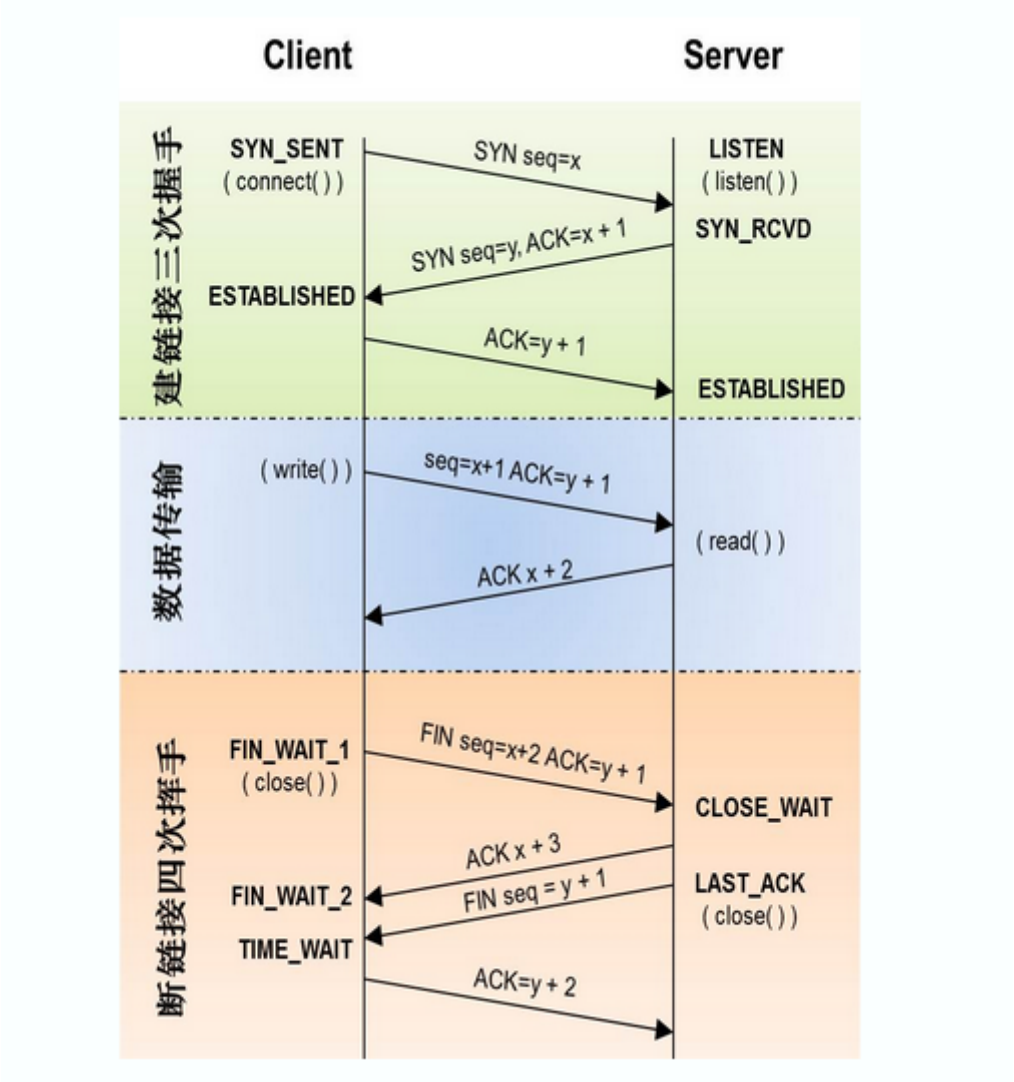
状态码	含义
200 OK	客户端请求成功 ****常用*****
400错误请求	请求报文有语法错误
401未经授权	未授权
403禁止	服务器拒绝服务
404未找到	请求的资源不存在 ****常用****
500内部服务器错误	服务器内部错误
503服务器不可用	服务器临时不能处理客户端请求（稍后可能可以）

响应包体：服务器返回给客户端的文本信息。

长连接和短连接

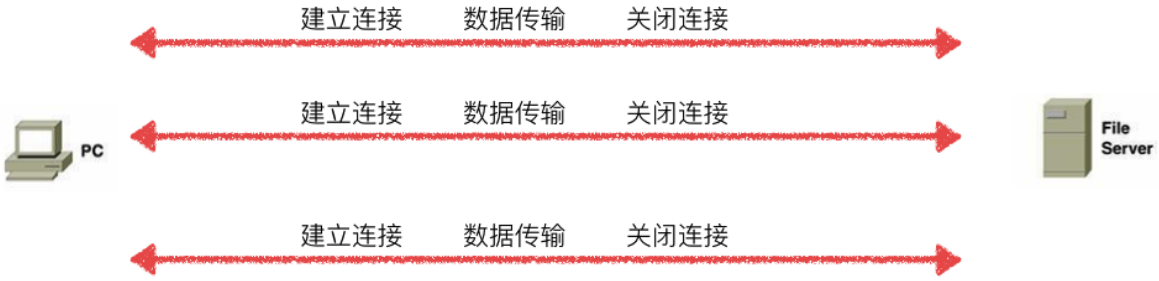
HTTP的长连接和短连接本质上是TCP长连接和短连接。

TCP通信的整个过程：



短连接（HTTP默认模式）

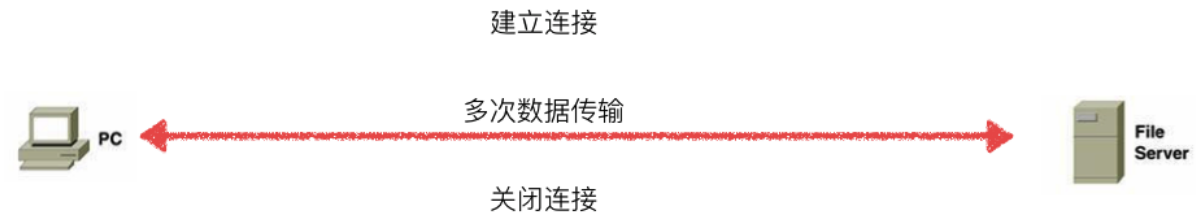
建立连接——数据传输——关闭连接...建立连接——数据传输——关闭连接



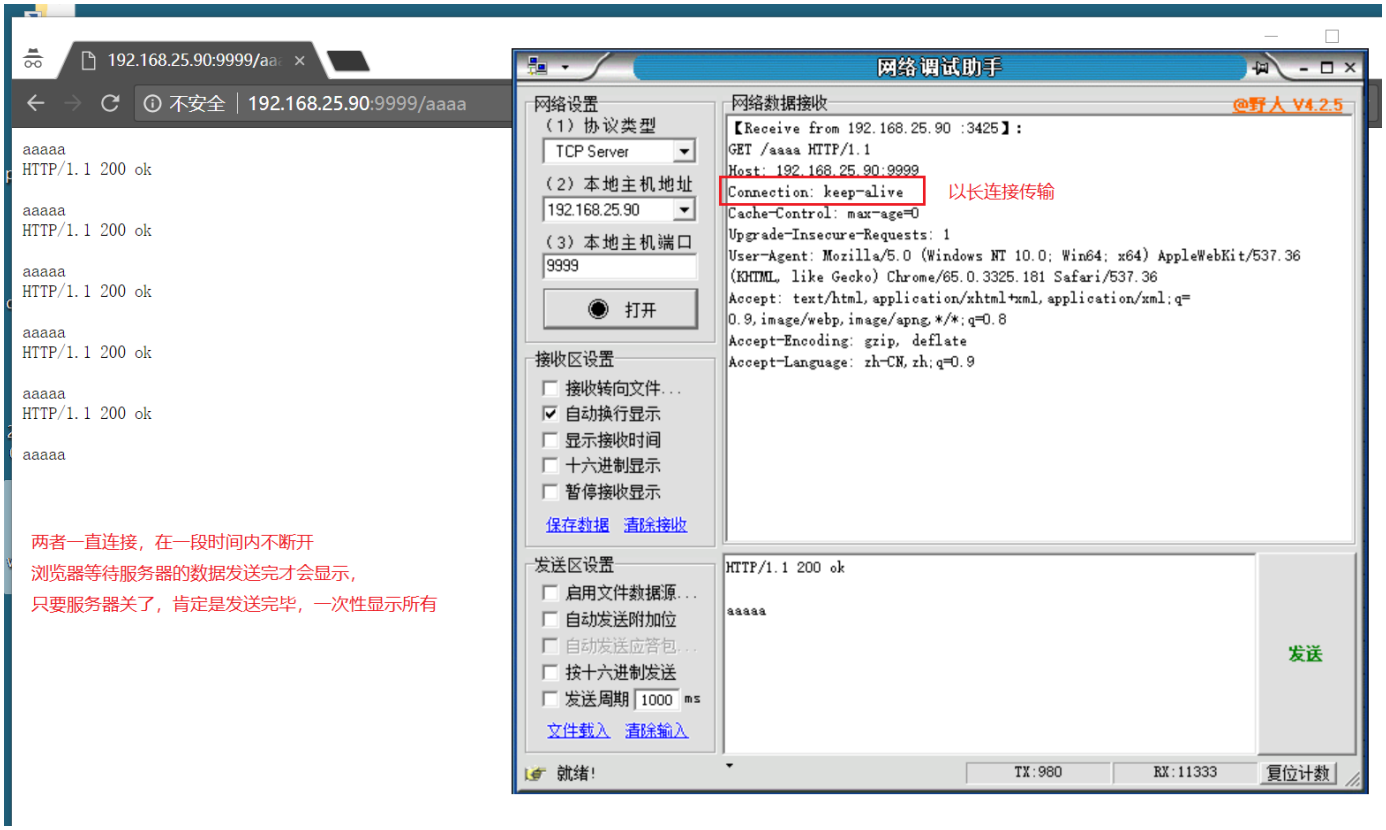
浏览器和服务器每进行一次HTTP操作，就建立一次连接，但任务结束就中断连接。

长连接

建立连接——数据传输...（保持连接）...数据传输——关闭连接



从 HTTP/1.1起，默认使用长连接，用以保持连接特性。使用长连接的HTTP协议，会在响应头有加入这行代码：



长/短连接的优缺点

长连接

优点：连接可以省去较多的TCP建立和关闭的操作，减少浪费，节约时间。对于频繁请求资源的客户来说，较适用长连接。
缺点：随着客户端连接越来越多，server早晚有扛不住的时候，这时候server端需要采取一些策略，如关闭一些长时间没有读写事件发生的连接，这样可以避免一些恶意连接导致server端服务受损

短连接
优点：短连接对于服务器来说管理较为简单,存在的连接都是有用的连接，不需要额外的控制手段
缺点：但如果客户请求频繁，将在TCP的建立和关闭操作上浪费时间和带宽。