

2.2Web和HTTP

2.2.1HTTP概况

1. Web的应用层协议是超文本传输协议(HTTP),是web的核心
2. HTTP由客户程序和服务器程序实现
3. 两个程序运行在不同的端系统钟,通过交换HTTP报文进行会话
4. HTTP定义了报文的结构以及客户和服务器进行报文交换的方式
5. Web网页由对象组成,一个对象只是一个文件,文件可以通过一个URL地址寻址
6. URL由两部分组成,存放对象的服务器名称和对象的路径名
7. Web浏览器实现了HTTP的客户端,Web服务器实现了HTTP的服务端
8. 交互过程:用户请求Web页面时,浏览器向服务器发出该页面对象的HTTP请求报文,服务器对该请求的对象进行响应
9. HTTP使用TCP作为它的支撑运输协议
10. HTTP不会保存关于客户的任何信息,所以HTTP是无状态协议

2.2.2非持续连接和持续连接

非持续连接:每个请求/响应都经一个单独的TCP连接发送

持续连接:所以的请求/响应经相同的TCP连接发送

RTT定义:往返时间,一个短分组从客户到服务器再返回客户所花费的时间

1. 采用非持续连接的HTTP
 1. 每个TCP连接在服务器发送一个对象后关闭,该连接不为其他对象持续下来
 2. 每个TCP连接只传输一个请求
 3. 每个对象都需要经历两倍RTT的交付时延,一个RTT创建TCP,一个RTT用于请求和接收
2. 采用持续连接的HTTP
 1. 采用HTTP1.1持续连接,服务器在发送响应后保持TCP连接
 2. 相同的客户和服务器之间,后续的请求/响应,通过相同的连接进行传送
 3. 一条连接经过一定时间间隔未被使用,则服务器关闭该连接
 4. HTTP默认使用带流水线的持续连接
 5. HTTP2在HTTP1.1的基础上,允许多个请求和回答交错

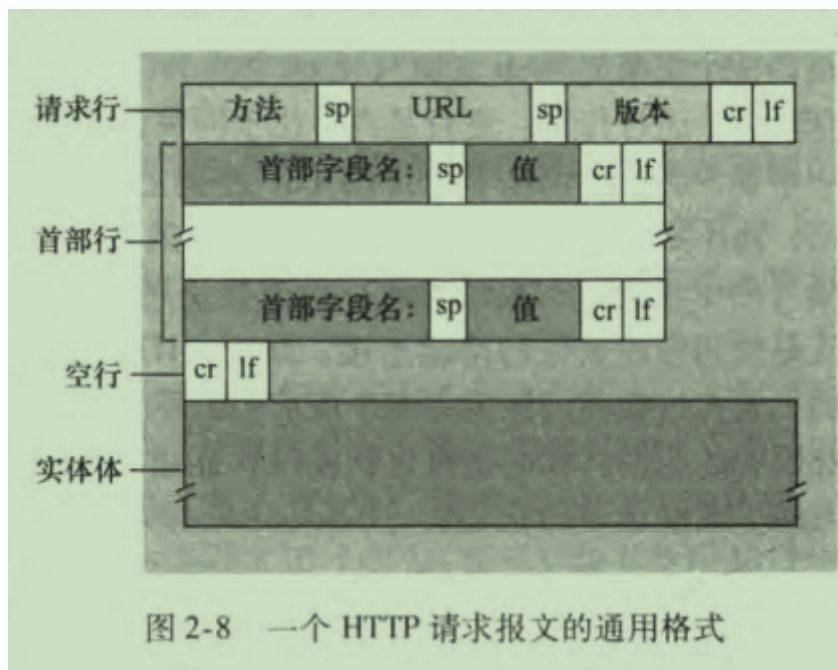
2.2.3HTTP报文格式

1. 请求报文

下面提供了一个典型的 HTTP 请求报文:

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

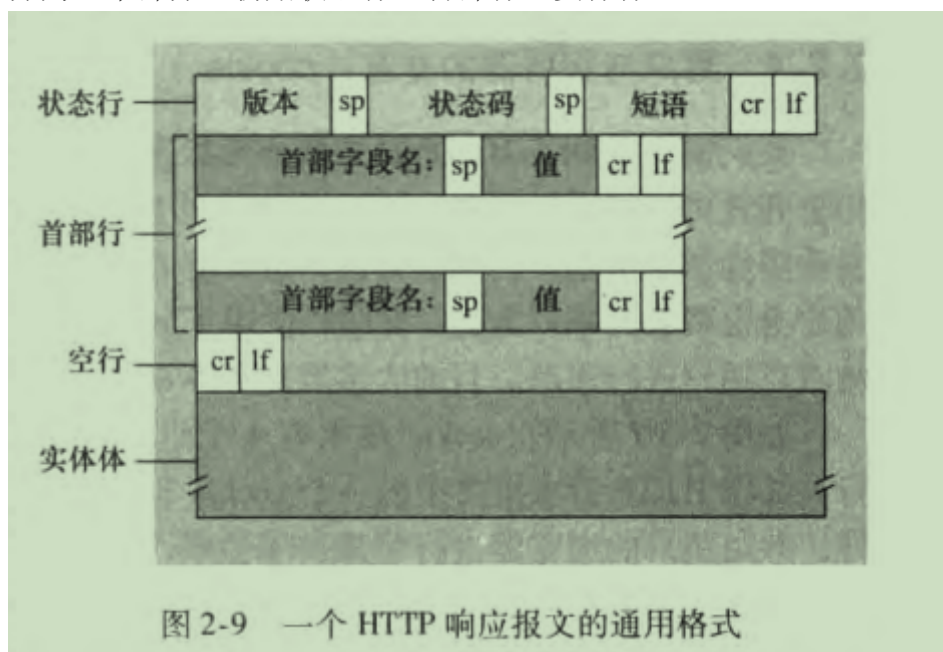
第一行为请求行,后继的行为首部行



2. 响应报文

```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 18 Aug 2015 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 18 Aug 2015 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html
(data data data data data ...)
```

分为三个部分，初始状态行，首部行，实体体



2.2.4用户与服务器的交互:cookie

网站对每个用户使用一个唯一的标识码(cookie)，这个识别码会索引到web后端数据库，会记录该用户在网站的行为

2.2.5Web缓存

Web缓存器也叫代理服务器，能够代表初始Web服务器来满足HTTP请求的网络实体

Web缓存器既是客户也是服务器

浏览器请求的情况：

- 1) 浏览器创建一个到 Web 缓存器的 TCP 连接，并向 Web 缓存器中的对象发送一个 HTTP 请求。
- 2) Web 缓存器进行检查，看看本地是否存储了该对象副本。如果有，Web 缓存器就向客户浏览器用 HTTP 响应报文返回该对象。
- 3) 如果 Web 缓存器中没有该对象，它就打开一个与该对象的初始服务器（即 `www.someschool.edu`）的 TCP 连接。Web 缓存器则在这个缓存器到服务器的 TCP 连接上发送一个对该对象的 HTTP 请求。在收到该请求后，初始服务器向该 Web 缓存器发送具有该对象的 HTTP 响应。
- 4) 当 Web 缓存器接收到该对象时，它在本地存储空间存储一份副本，并向客户的浏览器用 HTTP 响应报文发送该副本（通过现有的客户浏览器和 Web 缓存器之间的 TCP 连接）。

优点：

1. 减少对客户请求的响应时间
2. 能够减少一个机构的接入链路到因特网的通信量，减少通信量也就不需要增加带宽

2.2.6条件GET方法

用于更新缓存器中的对象副本

在GET方法后加入 `If-modified-since:`更新时间

```
GET /fruit/kiwi.gif HTTP/1.1
Host: www.exotiquecuisine.com
If-modified-since: Wed, 9 Sep 2015 09:23:24
```