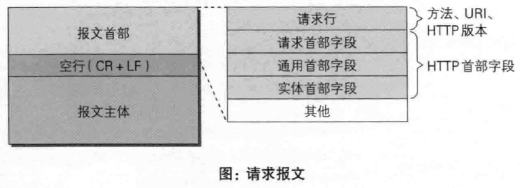
**HTTP协议**

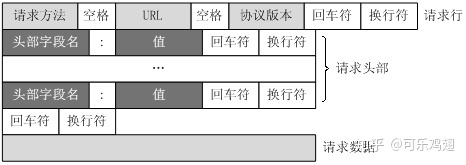
HTTP(Hyper Text Transfer Protocol，超文本传输协议)，用于从万维网服务器传输文本到本地浏览器的传送协议。HTTP协议是基于TCP的应用层协议，默认端口是80。它不关心数据传输的细节，主要是用来规定客户端和服务端的数据传输格式。HTTP是基于**请求与响应**模式的、无状态的、应用层的协议。

# HTTP请求报文



HTTP请求报文主要由**请求行**、**请求头部**、**空一行**、**请求正文**4部分组成。

1. 请求行：由请求方法、URL和协议名称和版本组成。
2. 请求头部：附加的、需要服务器知道的一些信息。请求头包含若干个属性，格式为“属性名：属性值”，服务端据此获取客户端的信息。
3. 空一行：
4. 请求体：它将一个页面表单中的组件值通过param1=value1&param2=value2的键值对形式编码成一个格式化串，它承载多个请求参数的数据。不但请求体可以传递请求参数，请求URL也可以通过类似于“/chapter15/user.html? param1=value1&param2=value2”的方式传递请求参数。



## 1.1. 请求行

### 1.1.1. 请求方法

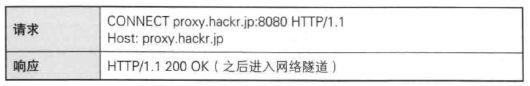
GET和POST是最常见的HTTP方法，除此以外还包括DELETE、HEAD、OPTIONS、PUT、TRACE。



1. **GET**：请求访问已被URL识别的资源。指定的资源经过服务器解析后返回相应内容。
2. **POST**：传输实体的主体。用GET方法也可以传输实体的主体，但一般不使用GET进行传输，而是用POST方法。
3. **PUT**：传输文件。要求在请求报文的主体中包含文件内容，然后保存到请求URP指定的位置。
4. **HEAD**：获得报文首部，用于确认URI的有效性以及资源更新的日期时间等。
5. **DELETE**：用来删除文件。按请求URI删除指定的资源。
6. **OPTIONS**：查询针对URI指定的资源支持的方法。
7. **TRACE**：追踪路径，回显服务器收到的请求。用于测试或诊断。
8. **CONNECT**：要求在与代理服务器通信时建立隧道，实现用隧道协议进行TCP通信。主要使用SSL/TLS协议把通信内容加密后经网络隧道传输。

CONNECT方法的格式如下：

CONNECT 代理服务器名：端口号 HTTP版本



### 1.1.2. URL

Uniform Resource Location，统一资源定位符，用于描述网上的资源。

格式：schema://host[:port#]/path/…/[?query-string]

1. Schema：协议，如http、https、ftp等
2. Host：域名或IP地址，定位服务器
3. Port：端口
4. Path：资源路径
5. Query-string：发送的参数

## 1.2. 请求头部

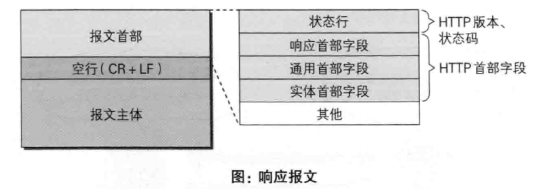
键值对的格式：

首部字段名：字段值

## 1.3. 使用Cookie的状态管理

Cookie技术通过在请求和响应报文中写入Cookie信息来控制客户端的状态。Cookie会根据从服务器端发送的响应报文内的一个叫做**Set-Cookie**的首部字段信息，通知客户端保存Cookie。当下次客户端再往该服务器发送请求时，客户端会自动在请求报文中加入Cookie值后送出去。服务端发现客户发送过来的Cookie后，会去检查究竟是从哪一个客户端发送来的连接请求，然后对比服务器上的记录，最后得到之前的状态信息。

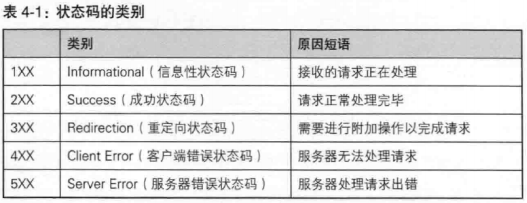
# HTTP响应报文



HTTP响应报文主要由**响应行**、**响应头部**、**空一行**、**响应正文**4部分组成。

响应行包括：响应结果的状态码、原因短语和HTTP版本。

## 2.1. 状态码



### 2XX成功

#### 200 OK

表示从客户端发来的请求在服务器被正常处理了。

#### 204 No Content

表示服务器接收的请求已处理成功，但在返回的响应报文中不含实体的主体部分。

#### 206 Partial Content

表示客户端进行了范围请求，而服务器成功执行了这部分的GET请求。响应报文中包含由Content-Range指定范围的实体内容。

### 3XX重定向

表明浏览器需要执行某些特殊的处理以正确处理请求。

#### 301 Moved Permanently

永久性重定向，表示请求的资源已被分配了新的URI，以后应使用资源现在的URI。

下方的请求URI，当指定资源路径的最后忘记添加斜杠”/”，就会产生301状态码：

<http://example.com/sample>

#### 302 Found

临时性重定向，表示请求的资源已被分配了新的URI，希望用户本次能够使用新的URI访问。

#### 303 See Other

表示由于请求对应的资源存在着另一个URI，应使用GET方法定向获取请求的资源。

#### 304 Not Modified

表示客户端发送附带条件的请求时，服务端允许请求访问资源，但未满足条件的情况。304状态码返回时，不包含任何响应的主体部分。

#### 307 Temporary Redirect

### 4XX 客户端错误

#### 400 Bad Request

请求报文中存在语法错误。浏览器会像200 OK一样对待该状态码。

#### 401 Unauthorized

表明发送的请求需要有通过HTTP认证的认证信息。若之前已进行过1次请求，则表示用户认证失败。

#### 403 Forbidden

表明对请求资源的访问被服务器拒绝了。服务器没有必要给出拒绝的理由，但如果想说明的话，可以在实体的主体部分描述原因。未获得文件系统的访问授权、访问权限出现某些问题等情况都可能是发生403的原因。

#### 404 Not Found

服务器上无法找到请求的资源。也可以在服务器拒绝请求且不想说明理由时使用。

### 5XX服务器错误

#### 500 Internal Server Error

表明服务器在执行请求时发生了错误。也可能是Web应用存在的bug或某些临时的故障。

#### 503 Service Unavailable

表明服务器暂时处于超负载或正在进行维护，现在无法处理请求。

# 3. 与HTTP协作的Web服务器

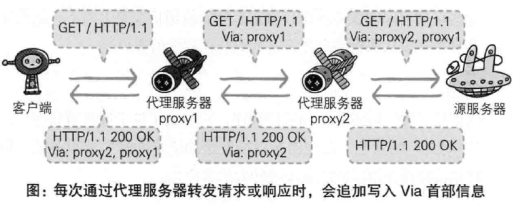
HTTP通信时，除了客户端和服务器，还有一些用于通信数据转发的应用程序，如代理、网关和隧道。

## 3.1. 通信数据转发程序

### 3.1.1. 代理

代理服务器的基本行为是接收客户端发送的请求后转发给其他服务器。代理不改变请求URI，会直接发送给前方持有资源的目标服务器。从源服务器返回的响应经代理服务器后再传给客户端。

HTTP通信过程中，可级联多台代理服务器。转发时，需要附加Via首部字段以标记经过的主机信息。

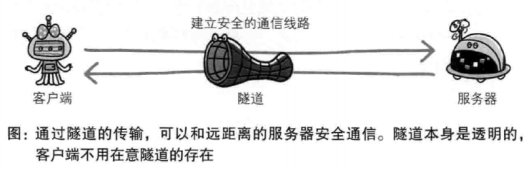


### 3.1.2. 网关

网关的工作机制和代理十分相似，网关能使通信线路上的服务器提供非HTTP协议服务。**利用网关能够提高通信的安全性**，因为可以在客户端和网关之间的通信线路上加密，以确保连接的安全。

### 3.1.3. 隧道

隧道可按要求建立起一条与其他服务器的通信线路，届时使用SSL等加密手段进行通信。隧道的目的是确保客户端能与服务器进行安全的通信。



## 3.2. 保存资源的缓存

缓存是指代理服务器或客户端本地磁盘内保存的资源副本。利用缓存可减少对源服务器的访问。缓存服务器是代理服务器的一种。**利用缓存可避免多次从源服务器转发资源，客户端可就近从缓存服务器上获取资源，服务器也不必多次处理相同的请求**。

源服务器上的资源更新时，缓存服务器需要向源服务器确认并更新。

# 4. HTTP首部

## 4.1. HTTP首部字段

使用首部字段是为了给浏览器和服务器提供报文主体大小、所使用的的语言、认证信息等内容。

HTTP首部字段的结构由首部字段名和字段值组成，中间用冒号“：”分隔。

首部字段名：字段值

### 4.1.1. HTTP首部字段类型

1. **通用首部字段：**请求报文和响应报文都会使用的首部。
2. **请求首部字段：**从客户端发送请求报文时使用的首部，补充了请求的附加内容、客户端信息、响应内容相关优先级等信息。
3. **响应首部字段：**从服务器向客户端返回响应报文时使用的首部，补充了响应的附加内容，也会要求客户端附加额外的内容信息。
4. **实体首部字段：**针对请求报文和响应报文的实体部分使用的首部。补充了资源内容更新时间等与实体有关的信息。

## 4.2. 通用首部字段

1. Cache-Control

操作缓存的工作机制。指令的参数可选，多个指令之间用“，”分隔。首部字段Cache-Control的指令可用于请求及时响应。

# 5. HTTPS

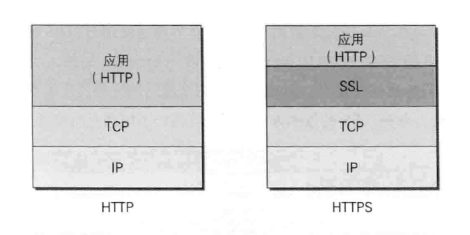
## HTTP的缺点

1. 通信使用明文传输，内容会被窃听；

2. 不验证通信方的身份，因此有可能遭遇伪装；

3. 无法证明报文的完整性，所以有可能内容被篡改（中间人攻击）。

## HTTP+加密+认证+完整性保护=HTTPS

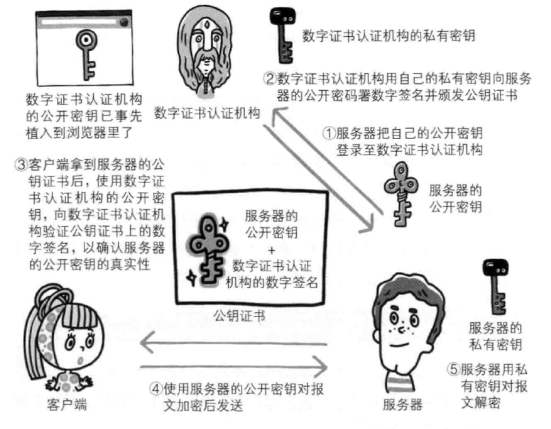


通常，HTTP直接和TCP通信。当使用SSL时，则演变成HTTP和SSL通信，SSL和TCP通信了。

HTTPS采用混合加密方式：

1. 使用非对称加密的方式将共享秘钥加密，再安全地交换；

2. 确保交换的秘钥是安全的前提下，使用共享秘钥加密方式进行通信。



## HTTPS的安全通信机制

首先建立TCP连接，然后开始SSL/TLS握手：

1. 客户端：发送Client Hello报文开始SSL通信。报文中包含客户端支持的SSL版本、加密套件列表（加密算法、秘钥长度等）；

2. 服务端：发送Server Hello，报文中包含SSL版本和加密组件。服务端的加密组件内容是从客户端的加密组件中筛选出来的；

3. 服务端：发送Certificate报文，报文中包含公开秘钥；

4. 服务端：发送Server Hello Done通知客户端，最初阶段的SSL握手协商部分结束；

5. 客户端以Client Key Exchange报文作为回应，报文中包含Pre-master secret（预制密钥）的随机密码串，该报文使用步骤3中的公开秘钥进行加密；

6. 客户端继续发送Change Cipher Spec报文，提示服务器，在此报文之后的通信会采用Pre-master secret进行加密；

7. 客户端发送Finished报文，该报文包含连接至今全部报文的整体校验值。若服务器能够正确解密该报文，则这次握手协商成功；

8. 服务端同样发送Change Cipher Spec报文；

9. 服务端：发送Finished报文；

10. SSL连接建立完成，双方开始HTTP通信；

11. 最后客户端断开连接，发送close\_notify报文；

12. TCP 4次挥手。

# 6. HTTP认证方式

服务器需要知道当前访问者是谁，即需要对当前访问的客户端进行认证。HTTP使用的认证方式包括：

1. BASIC认证（基本认证）；
2. DIGEST认证（摘要认证）；
3. SSL客户端认证；
4. FormBase认证（基于表单认证）；

此外，还有Windows统一认证（Keberos认证、NTLM认证）。

## 6.1. BASIC认证

BASIC认证步骤：

1. 当请求的资源需要BASIC认证时，服务器会随状态码401 Authorization Required，返回带WWW-Authenticate首部字段的响应。该字段内包含认证的方式（BASIC）和Request-URI安全域字符串。
2. 接收到状态码401的客户端为了通过BASIC认证，需要将用户ID和密码发送给服务器。发送的字符串内容由用户ID和密码构成，中间以冒号连接，再经过Base64处理。
3. 接收到包含首部字段Authorization请求的服务器，会对认证信息的正确性进行验证。如果验证通过，则返回一条包含Request-URI资源的响应。



特点：用户ID和密码只是经过Base64编码，仍旧是明文，极易被窃听；BASIC认证使用不够灵活，安全等级低。

## 6.2. DIGEST认证



DIGEST认证采用质询/响应的方式，但不会直接明文发送用户ID和密码。

认证步骤：

1. 当请求需认证的资源时，服务器会随状态码401 Authorization Required，返回带WWW-Authenticate首部字段的响应。该字段内包含质询响应方式认证所需的临时质询码（随机数，nonce）。
2. 接收到401状态码的客户端，返回的响应中包含DIGEST认证必须的首部字段Authorization信息。首部字段Authorization内必须包含username， realm，nonce，uri和response的字段信息。其中，realm和nonce就是之前从服务器接收到的响应中的字段。。username是realm限定范围内可进行认证的用户名。uri是Request-URI的值。response存放经过MD5运算后的密码字符串，形成响应码。
3. 接收到包含首部字段Authorization请求的服务器，会确认认证信息的正确性。认证通过后则返回包含Request-URI资源的响应。并且这是会在首部字段Authorization-Info写入一些认证成功的相关信息。

## 6.3. SSL客户端认证

SSL客户端认证是借由HTTPS的客户端证书完成认证的方式，根据客户端证书，服务器可确认方式是否来自已登录的客户端。

认证步骤：

1. 接收到需要认证资源的请求，服务器会发送Certificate Request报文，要求客户端提供客户端证书；
2. 用户选择将发送的客户端证书后，客户端会把客户端证书信息以Client Certificate报文的方式发送给服务器。
3. 服务器验证客户端证书通过后方可领取证书内客户端的公开密钥，然后开始HTTPS加密通信。

SSL客户端认证通常会和基于表单认证组合形成一种双因素认证来使用。SSL客户端认证需要支付一定的费用。

## 6.4. FormBase认证

客户端会向服务器上的Web应用程序发送登录信息，按登录信息的验证结果认证。