一、HTTP协议的主要特点可概括如下：  
1.支持客户/服务器模式。  
2.简单快速：客户向服务器请求服务时，只需传送请求方法和路径。请求方法常用的有GET、HEAD、POST。每种方法规定了客户与服务器联系的类型不同。由于HTTP协议简单，使得HTTP服务器的程序规模小，因而通信速度很快。  
3.灵活：HTTP允许传输任意类型的数据对象。正在传输的类型由Content-Type加以标记。  
4.无连接：无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开连接。采用这种方式可以节省传输时间。  
5.无状态：HTTP协议是无状态协议。无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。另一方面，在服务器不需要先前信息时它的应答就较快。

二、TCP/IP是互联网相关的各类协议族的总称

  http（超文本传输协议）是一个基于请求与响应模式的、无状态的、应用层的协议，常基于TCP的连接方式，HTTP1.1版本中给出一种持续连接的机制，绝大多数的Web开发，都是构建在HTTP协议之上的Web应用。

TCP/IP的分层管理：

按层次分为四层：

1. 应用层

应用层决定了向用户提供应用服务时t通信的活动。

TCP/IP协议族内预存了各类通用的应用服务，比如FTP（文本传输协议）和DNS（域名系统）。

HTTP协议也处于该层。

1. 传输层

传输层对上层应用层，提供处于网络连接中的两台计算机之间的数据传输。

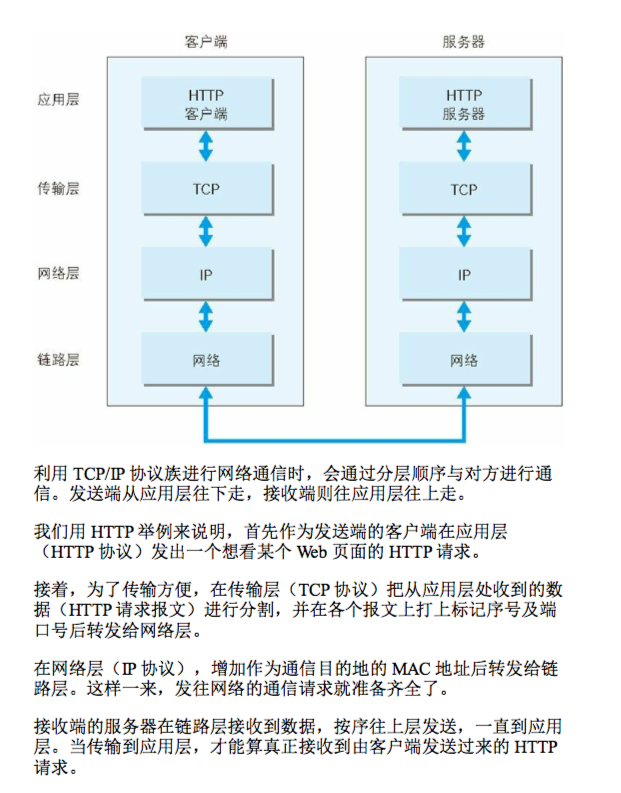
在传输层有两个性质不同的协议：TCP（传输控制协议）和UDP（用户数据报协议）

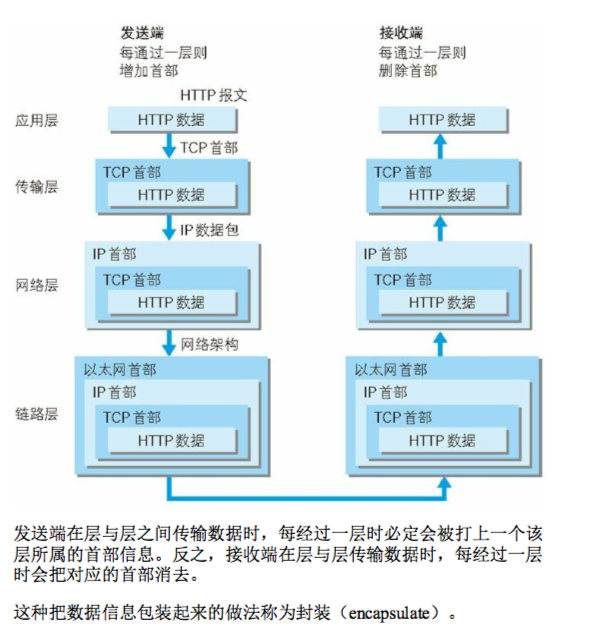
1. 网络层（网络互连层）

用来处理在网络上流动的数据包，数据包是网络传输的最小数据单位，该层规定了通过怎样的路径到达对方计算机，并把数据包传送给对方。与对方计算机之间通过多台计算机或网络设备进行传输时，网络层所起的作用就是在众多的选项内选择一条传输路线。

1. 数据链路层（网络接口层）

用来处理连接网络的硬件部分，包括控制操作系统，硬件的设备驱动，NIC（即网卡），及光纤等物理可见部分（还包括连接器等一切传输媒介）。硬件上的范畴均在链路层的作用范围之内。





IP协议：

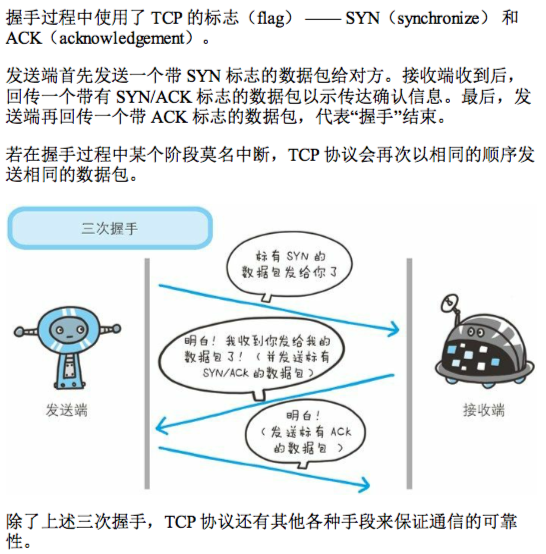
作用是把各种数据包传送给对方，要保证确实传送到对方那里，需要满足重要的条件，其中两个分别是IP地址和MAC地址，IP地址指明了节点被分配到的地址，MAC地址是指网卡所属的固定地址，ip地址可以和mac地址进行配对，ip地址可变换，但mac基本上不会改变。

在进行中转时，会利用下一站中转设备的mac地址来搜索下一个中转目标，这时采用ARP协议，ARP协议根据通信方的ip地址就可以反查出对应的mac地址。

TCP协议：

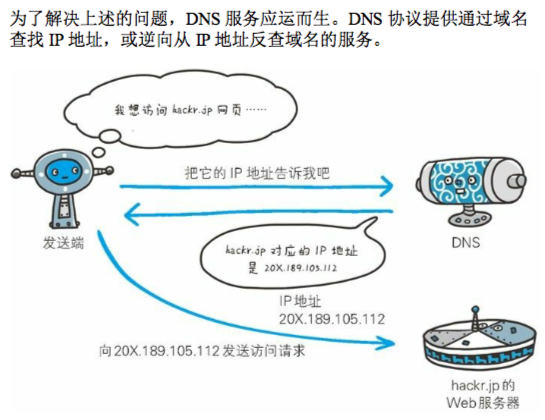
提供可靠的字节流服务，为了传送大数据才把数据分割，并且能够确认数据是否送达到对方。

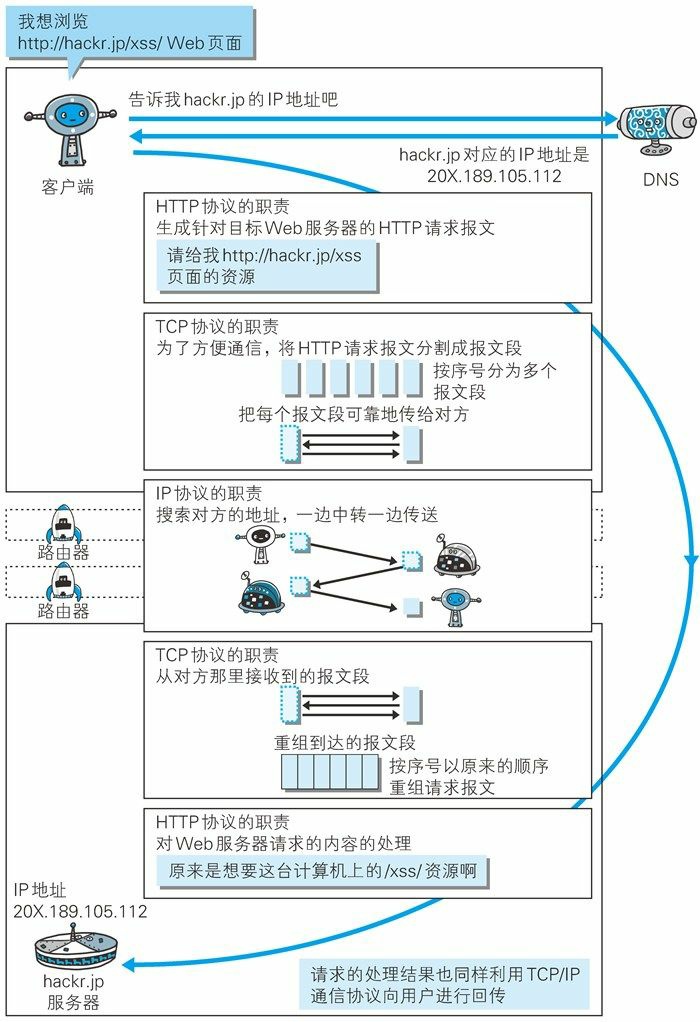
为了确认数据能到达目标，tcp协议采用了三次握手策略：



DNS服务：

提供域名到ip地址之间的解析服务，



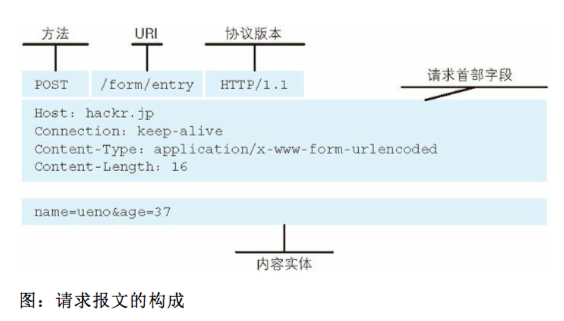


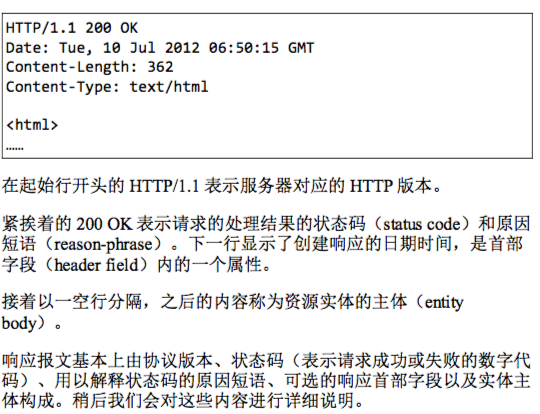
三、URL统一资源定位符

HTTP URL (URL是一种特殊类型的URI，包含了用于查找某个资源的足够的信息)的格式如下：  
  
http表示要通过HTTP协议来定位网络资源；host表示合法的Internet主机域名或者IP地址；port指定一个端口号，为空则使用缺省端口80；abs\_path指定请求资源的URI；如果URL中没有给出abs\_path，那么当它作为请求URI时，必须以“/”的形式给出，通常这个工作浏览器自动帮我们完成。  
eg:  
1、输入：[www.guet.edu.cn](http://www.guet.edu.cn/" \t "_blank)  
浏览器自动转换成：[http://www.guet.edu.cn/](http://www.guet.edu.cn/" \t "_blank)  
2、http:192.168.0.116:8080/index.jsp

四**、HTTP协议详解之请求篇**

http请求由三部分组成，分别是：请求行、消息报头、请求正文





五、告知服务器意图的HTTP方法

1、GET方法

用来请求访问已被URI识别的资源。

2、POST方法

传输实体的主体，

post功能与get相似，但是post的主要目的不是获取相应的主体内容。



3、put传输文件

就像FTP协议的文件上传一样，要求在请求报文的主体中包含文件内容，然后保存到请求URI指定的位置。

4、head 获得报文首部

和get方法一样，只是不返回报文主体部分，用于确认URI的有效性及资源更新的日期时间等。

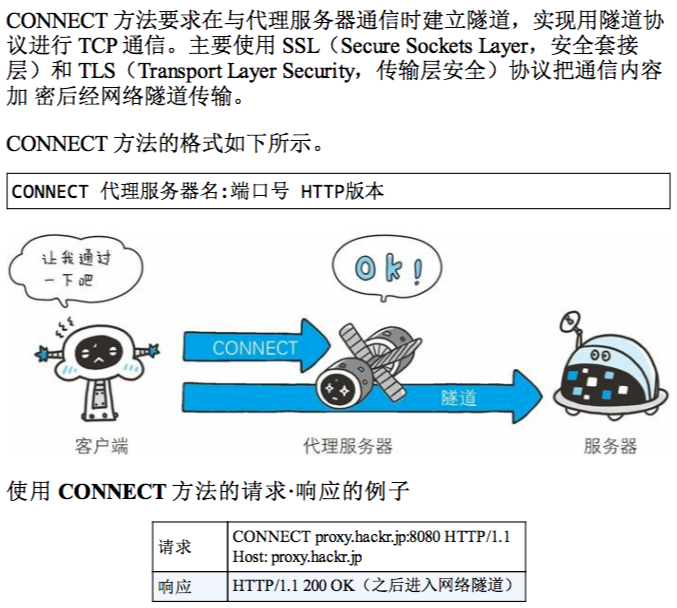
5、delete方法

与put相反的方法，按请求URI删除指定的资源。

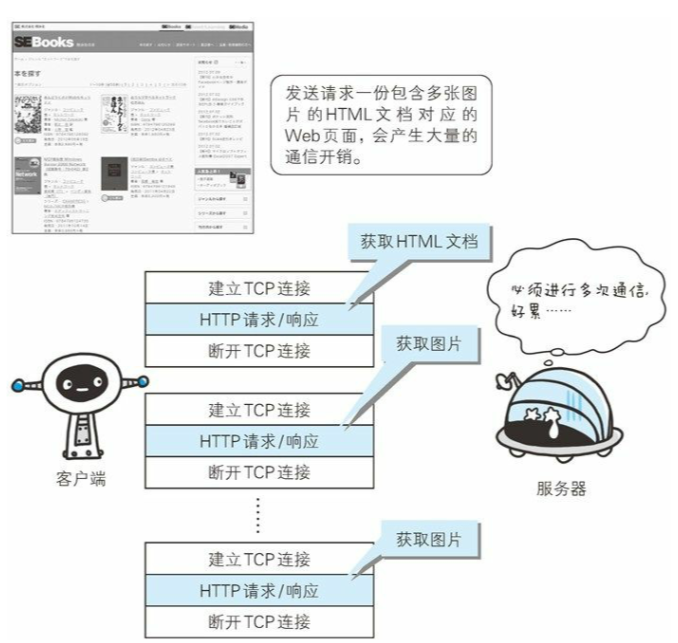
6、options询问支持的方法

用来查询针对请求URI指定的资源支持的方法。

7、connect要求用隧道协议连接代理



六、持久连接节省通信量



持久连接的特点是只要任意一端没有明确提出断开连接，则保持tcp连接状态。建立一次tcp连接以后进行多次请求和响应的交互。

七、管线化

不用等待响应即可发送下一个请求。

八、使用cookie的状态管理（无状态协议）

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值后发出去。

九、HTTP报文内的HTTP信息

1、HTTP报文（报文首部&报文主体）

由最早出现的空行（CR回车符+LF换行符）来划分，通常不一定要有报文主体。

一般有四种首部，分别是通用首部，请求首部，响应首部和实体首部。

2、报文主体和实体主体的差异

报文是HTTP通信的基本单位，由八位组字节流组成，通过HTTP通信传输。

实体作为请求或响应的有效载荷数据被传输，其内容由实体首部和实体主体组成。

HTTP报文的主体用于用于传输请求或响应的实体主体。

通常报文主体等于实体主体，只有当传输中进行编码操作时，实体主体的内容发生变化，才导致他和报文主体产生差异。

3、压缩传输的内容编码

内容编码指明应用在实体上的编码格式，并保持实体信息原样压缩。内容编码后的实体由客户端接收并进行解码。

Gzip，compress，deflate，identity（不进行编码）

4、分割发送的分块传输编码

将实体主体分成多个部分，每一块会用十六进制来标记块的大小，而实体主体最后一块会用0（CR+LF）来标记。

5、多部分对象集合

multipart/form-data 在web表单上传时使用

multipart/byteranges 状态码206 响应报文包含了多个范围的内容时使用

在HTTP报文中使用多部分对象集合时，需要在首部字段里加上content-type，

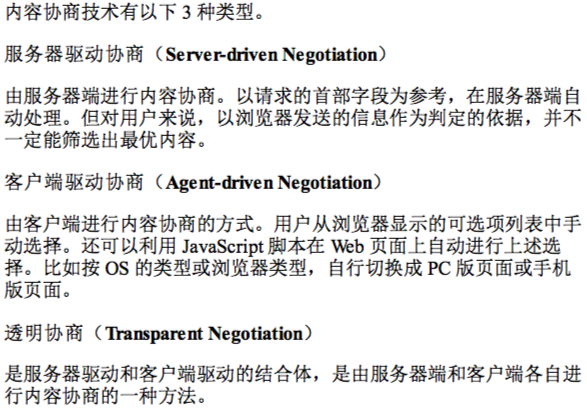
6、获取部分内容的的范围请求

要实现该功能需要指定下载的实体范围，像这样，指定范围发送的请求叫做范围请求。

会用到首部字段range来指定资源的byte范围。

响应会返回状态码为206partial content的响应报文，如果服务器无法响应范围请求则会返回状态码200 OK和完整的实体内容。

7、内容协商返回最合适的内容



十、返回结果的HTTP状态码



1、2XX成功

200 OK

204 NO CONTENT

返回的报文中不含实体主体部分也不允许返回任何实体的主体，浏览器显示的页面不发生更新。

206 partial content

进行了范围请求

2、3XX 重定向 （当这些响应状态码返回时，几乎所有浏览器会把post改成get，并删除请求报文的主体，之后请求会自动再次发送。）

301 moved permanently

永久性重定向 请求的资源已被分配了新的URI

302 Found

临时性重定向 希望用户本次能使用新的URI进行访问

303 see other

与302有相同的功能 但是303明确表示客户端应该使用get方法获取资源

304 Not modified

表示客户端发送附带条件的请求时，服务器端允许请求访问资源，但未满足条件的情况，

307 temporary redirect

临时重定向 与302有相同的含义

3、4XX客户端错误

400 Bad request

表示请求报文中存在语法错误

401 unauthorized

表示发送的请求需要有通过HTTP认证，浏览器初次接收到401响应时会弹出认证用的对话窗口。

403 forbidden

访问被服务器拒绝了。未获得文件系统的访问授权，访问权限出现某些问题等都是发生403的原因。

404 Not Found

服务器上无法找到请求的资源

4、5XX 服务器错误

500 Internal server error

服务器端在执行请求时发生了错误

503 service unavailable

服务器正在处于超负载或正在进行停机服务，现在无法处理请求。

状态码和状况的不一致：不少返回的状态码响应都是错误的，比如web应用程序内部发生错误，状态码依旧返回200 OK。

十一、与HTTP协作的web服务器

1、用单台虚拟主机实现多个域名

在相同的ip地址下，由于虚拟主机可以寄存多个不同主机名和域名的web网站，因此在发送HTTP请求时，必须在host首部内完整指定主机名或域名的URI。

2、通信数据转发程序

①代理（中间人角色）

请求和响应的转发会经过数台类似锁链一样连接起来的代理服务器，转发时需要附加via首部字段以标记出经过的主机信息。

使用代理服务器的理由有：利用缓存技术减少网络带宽的流量，组织内部针对特定网站的访问控制，以获取访问日志为主要目的。

缓存代理：缓存代理会预先将资源的副本保存在代理服务器上。

透明代理：不对报文做任何加工的代理类型。

②网关

可以由HTTP转化为其他协议通信，工作机制与代理相似。

③隧道

3、保存资源的缓存

①缓存服务器的优势在于利用缓存可以避免多次从源服务器转发资源。

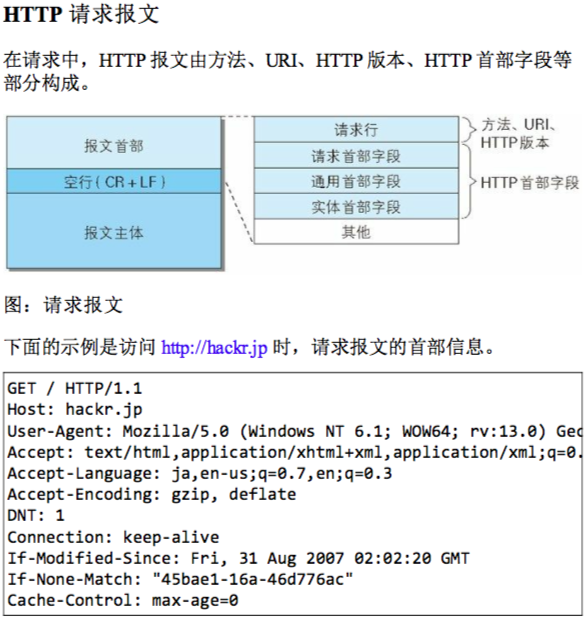
缓存过期后，缓存服务器会向源服务器确认资源的有效性。

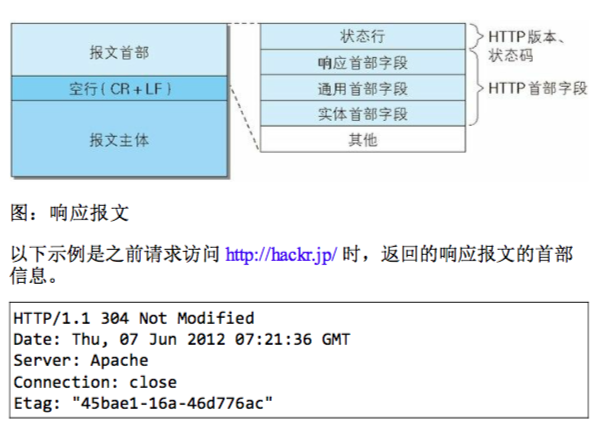
②客户端的缓存

把客户端缓存称为临时缓存文件。

十二、HTTP首部

1、





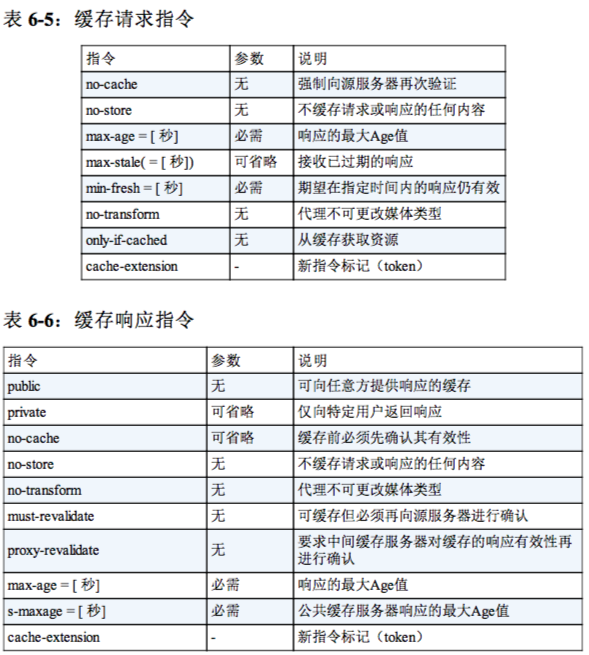
2、HTTP首部字段

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，所使用的语言，认证信息等内容。

3、通用首部字段

① cache-control

能够控制缓存的行为



表示是否能缓存的指令：

public：其他用户也可以利用缓存。与private相反。

no-cache：防止从缓存中返回过期的资源。无参数值的首部字段可以使用缓存，只能在响应指令中指定该参数。

控制可执行缓存的对象的指令

no-store：规定缓存不能在本地存储请求或响应的任一部分。

指定缓存期限和认证的指令

s-maxage：功能和max-age相同，但是该指令只适用于供多位用于使用的代理。当使用s-maxage时直接忽略对expires首部字段及max-age指令的处理。

Max-age：数值表示资源保存为缓存的最长时间。

②connection（控制不再转发给代理的首部字段，管理持久连接）

③Date代表创建HTTP报文的时间和日期

④pragma：属于通用首部字段，但是只在客户端发送的请求中。

⑤trailer：事先说明在报文主体后记录了哪些首部字段。

⑥transfer-encoding

规定了传输报文主体时采用的编码方式

⑦warning

告知用户一些关于缓存问题的警告

4、请求首部字段：用于补充请求的附加信息，客户端信息，对响应内容相关的优先级等内容。（权重q值）

from：使用代理时应尽可能包含from首部字段，但可能会因为代理不同，将电子邮件地址记录在user-agent首部字段内。

Host：唯一一个必须包含在请求内的首部字段，虚拟主机运行在同一个IP上，用host加以区分。若服务器为设定host值则发送一个空值即可。

If-match：（if-none-match）

只有当if-match的字段与ETag值匹配一致时，服务器才会接受请求。

If-modifi-since：（if-unmodified-since相反，如果在指定日期时间之后进行了更新，则返回412）

在指定该字段值的日期时间之后，如果请求的资源都没有过更新，则返回状态码304 not modified。

If-range：

字段值若是和ETag值或更新的日期时间匹配一致，那么就作为范围请求处理，若不一致，则忽略范围请求，返回全部资源。如果不使用这个则需要进行两次处理（会返回412）。

Referer：

只要查看referer就能知道请求的URI是从哪个web页面发起的。

1. 响应首部字段：用于补充响应的附加信息，服务器信息，以及对客户端的附加要求等信息。

Accept-range：可处理范围请求时为bytes，反之指定为none。

Age：告知客户端，源服务器在多久前创建了响应。代理创建响应时必须加上首部字段age。

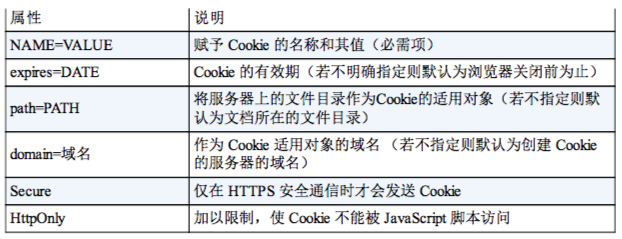
ETag：强etag值（无论实体发生多么细微的变化都会改变其值）和弱etag值（只用于提示资源是否相同，只有资源发生了根本的改变才会改变其值，这时会在字段值最开始处附加W/）

Vary：从代理服务器接收到源服务器包含vary指定项的响应之后，若再要进行缓存，仅对请求中含有相同vary指定首部字段的请求返回缓存，即使对相同资源进行请求，若vary指定的首部字段不相同，因此必须要从源服务器重新获取资源。

6、实体首部字段：用于补充内容的更新时间等与实体相关的信息。

7、为cookie服务的首部字段

①set-cookie（响应首部字段）



十三、确保web安全的https

1、加密处理防止被窃听

通信的加密：通过SSL（安全套接层）或TLS（安全层传输协议）的组合使用，加密HTTP的通信的内容。

与SSL组合使用的HTTP被称为HTTPS。

内容的加密：前提是要求客户端和服务器同时具备加密和解密机制。

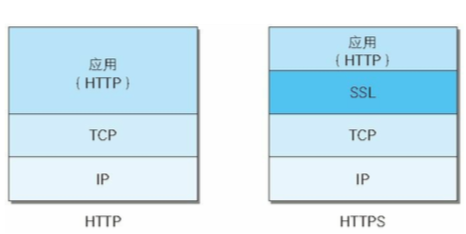
2、查明对手的证书

SSL不仅提供加密处理，而且还使用了一种被称为证书的手段，可用于确定方。

3、如何防止篡改

MD5和SHA-1等散列值校验的方法，以及用来确认文件的数字签名方法。

1. HTTP+加密+认证+完整性保护=HTTPS（身披SSL外壳的HTTP）



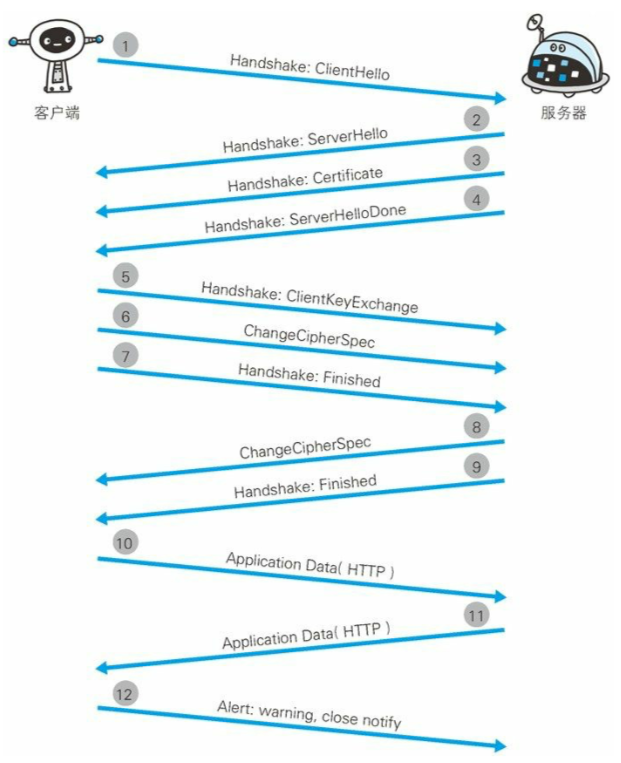
5、相互交换密钥的公开密钥加密技术

加密和解密同用一个密钥的方式称为共享密钥加密（对称密钥加密）。

公开密钥和私有密钥配对的方式称为公开密钥加密。

HTTPS采用混合加密机制，在交换密钥环节使用公开密钥加密，之后的建立通信交换报文阶段则使用共享密钥加密方式。

1. HTTPS的安全通信机制



1. 为什么不一直使用HTTPS

与纯文本通信相比，加密通信会消耗更多的CPU及内存资源，如果每次通信都加密，会消耗相当多的资源，平摊到一台计算机上时，能够处理的请求数目必定也将随之减少。只有在包含个人敏感数据时才利用HTTPS加密通信。

十四、确认访问用户身份的认证

1. basic认证

使用上不够便捷灵活，且达不到多数web网站期望的安全性等级，因此并不常用。

1. digest认证

不会像basic那样直接发送明文密码，采用质询响应方式，



1. SSL客户端认证（借由HTTPS的客户端证书完成认证的方式）

需要事先将客户端证书分发给客户端，且客户端必须安装此证书。

服务器验证客户端证书 验证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。

不仅依靠证书完成任务，一般会和基于表单认证组合（登录信息）形成一种双因素认证。

1. session管理及cookie应用

使用cookie来管理session，以弥补HTTP协议中不存在的状态管理功能。



另外，为减轻跨站脚本攻击（XSS）造成的损失，建议事先在cookie内加上httponly属性。

通常，一种安全的保存方法是，先利用给密码加salt的方式增加额外信息，再使用散列函数计算出散列值以后保存。

十五、基于HTTP的功能追加协议

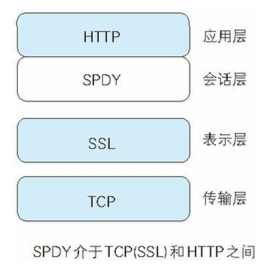
1. 消除HTTP瓶颈的SPDY（缩短web加载时间）

Ajax的解决办法：有效利用JavaScript和dom的操作，以达到局部web页面替换加载的异步通信手段，核心技术是名为XMLHttprequest的API，通过js脚本语言的调用就能和服务器进行HTTP通信。但是会导致大量请求产生。

Comet的解决办法：一旦服务器有内容更新了，comet不会让请求等待，而是直接给客户端返回响应，这是一种通过延迟应答，模拟实现服务器端向客户端推送的功能。先将响应置于挂起状态，当服务器端有内容更新时再返回该响应。但是一次连接的时间变长了。为了维持连接会消耗更多的资源。

1. spdy的设计与功能（通信中使用SSL）

spdy以会话层的形式加入，控制对数据的流动，但还是采用HTTP建立通信连接。



使用spdy以后HTTP协议额外获得以下功能：

多路复用流：所有请求的处理都在一条TCP连接上完成

赋予请求优先级

压缩HTTP首部

推送功能：支持服务器主动向客户端推送数据的功能

服务器提示功能：服务器可以主动提示客户端请求所需的资源，由于在客户端发现资源之前就可以获知资源的存在，因此在资源已缓存等情况下，可以避免发送不必要的请求。

SPDY基本上只是将单个域名（IP地址）的通信多路复用，所以当一个web网站上使用多个域名下的资源，改善效果就会收到限制。

1. 使用浏览器进行全双工通信的websocket：为了解决ajax和comet里XMLHttprequest附带的缺陷所引起的问题。

主要特点：

1. 推送功能：支持由服务器向客户端推送数据的推送功能，服务器可以直接发送数据而不必等待客户端的请求。
2. 减少通信量：websocket的首部信息很小。在HTTP连接建立以后需要完成一次握手步骤：

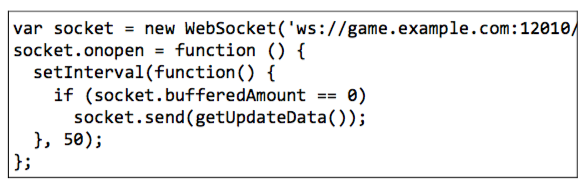
握手请求：



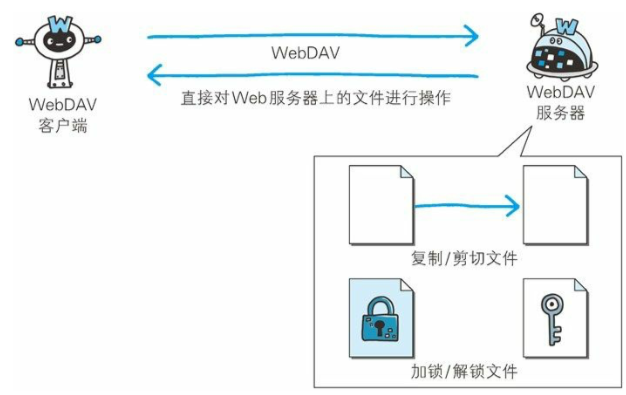
握手响应：



websocket API：

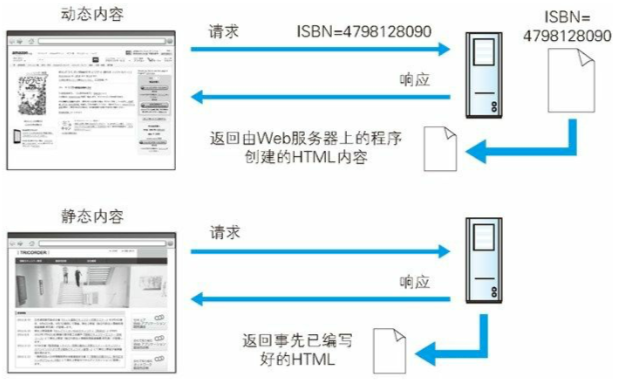


1. web服务器管理文件的webDAV：可对服务器上的内容直接进行文件复制，编辑等操作的分布式文件系统，



十六、web应用

1. 动态内容与静态内容



1. 与web服务器及程序协作的CGI（通用网关接口）

指web服务器在接收到客户端发送过来的请求后转发给程序的一组机制，在CGI作用下，程序会对请求内容做出相应的动作，比如创建HTML等动态内容，CGI通常由php，c等语言编写而成。

十七、web的攻击技术

1. 针对web应用的攻击模式
2. 以服务器为目标的主动攻击

攻击者通过直接访问web应用，把攻击代码传入。针对服务器上的资源进行攻击。

具有代表性的攻击有SQL注入攻击和OS命令注入攻击。

②以服务器为目标的被动攻击

利用全套策略执行攻击代码的执行模式。

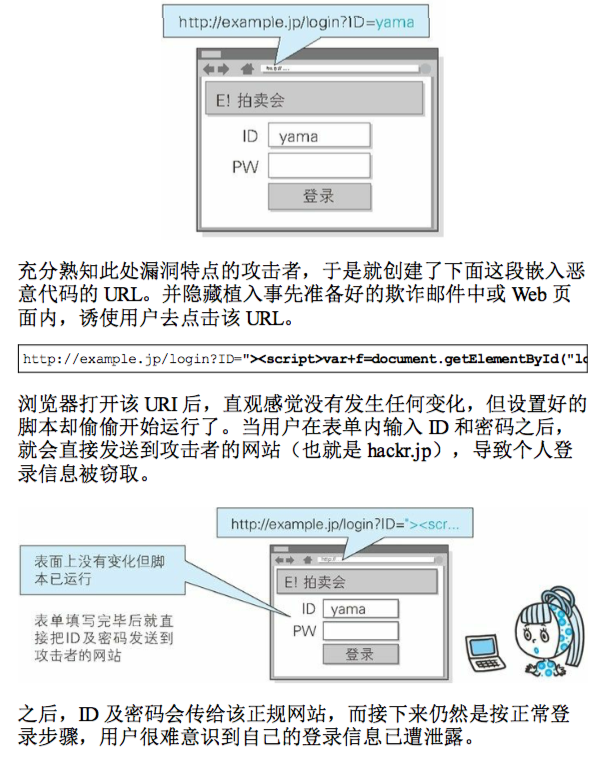
具有代表性的有跨站脚本攻击以及跨站点请求伪造。

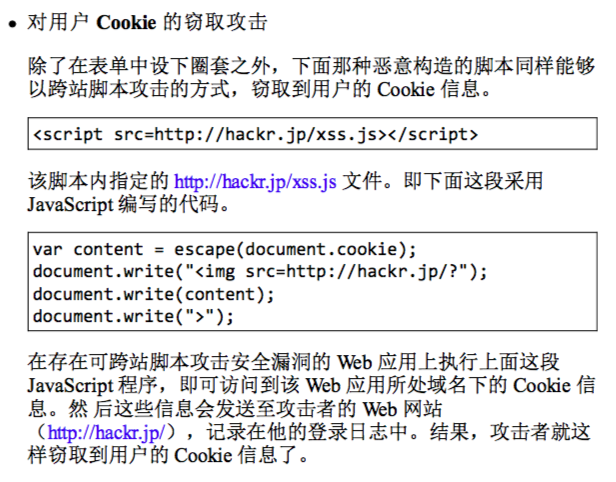
利用被动攻击可以发起对原本从互联网上无法直接访问的企业内网的攻击



1. 因输出值转义不完全引发的安全漏洞

①跨站脚本攻击XSS：指通过存在安全漏洞的web网站注册用户的浏览器内运行非法的HTML标签或JS进行的一种攻击，动态创建的HTML很可能存在安全漏洞，攻击者编写脚本设下陷阱，

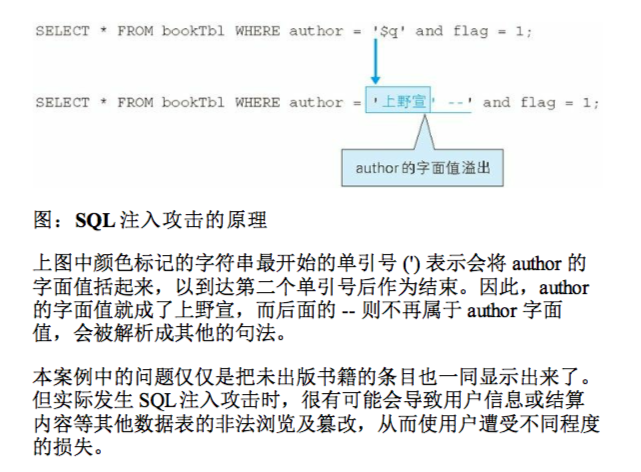




②SQL注入攻击

会执行非法SQL的SQL注入攻击

SQL注入是指针对web应用使用的数据库，通过运行非法的SQL而产生的攻击，



③OS命令注入攻击