# OSI七层模型

## OSI七层模型

-1）物理层：比特流传输。将数据转换为可通过物理介质传送的电子信号，相当于邮局的搬运工人

-2）数据链路层：提供介质访问、链路管理等。决定访问网络介质的方式。在此层将数据分帧，并处理流控制。本层指定拓扑结构并提供硬件寻址，相当于邮局的装拆箱工人

-3）网络层：寻址和路由选择。使用权数据路由经过大型网络，相当于邮局的排序工人。ip协议

-4）传输层：建立主机端到端连接。提供终端到终端的可靠连接，相当于公司的跑邮局的送信职员。tcp/udp协议

-5）会话层：建立、维护和管理会话。允许用户使用简单的名称建立连接，相当于公司中收寄信、写信封与拆信封的秘书

-6）表示层：处理数据格式、数据加密等。协商数据交换格式，相当于公司中简报老板、替老板写信的助理

-7）应用层：提供应用程序间通信。用户的应用程序和网络之间的接口老板。http协议

相关问题：

OSI七层模型分别对应着五层模型的哪一部分？

数据链路层是做什么的?数据链路层的流量控制？

各层之间的协议：

应用层：http协议

传输层：tcp协议，socket属于传输层的这么一个变成规范

网络层：ip协议

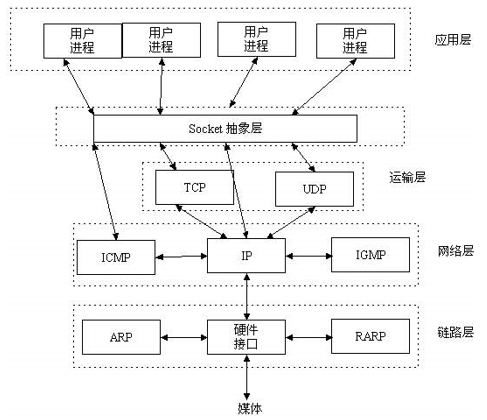
数据链路层：以太网协议

## TCP/IP协议簇

TCP/IP（Transaction Control Protocol/Internet Protocol）传输控制协议/网间协议，是一个工业标准的协议集，它是为广域网（WANs）设计的。

UDP（User Data Protocol，用户数据报协议）是与TCP相对应的协议，它属于TCP/IP协议簇中的一种。

TCP/IP协议族包括运输层、网络层、链路层。



TCP/UDP客户端、服务端的线程模型：

TCP的客户端和服务端的线程模型：客户端一般使用单线程模型，当有数据到来时启动线程读取，需要写入数据时开启线程进行数据写入。服务端一般使用多线程模型，一个线程负责接收tcp连接请求，每当接收到请求后开启一个线程处理它的读写事件。

UDP的客户端和服务端的线程模型：UDP的客户端和服务端比较简单，由于UDP数据包长度是确定的，只需要写入一个固定的缓存和读取一个固定的缓存空间即可。一般使用DatagramPacket包装一个udp数据包，使用DatagramSocket发送。

问：网络中的进程如何通信？

在本地可以通过进程PID唯一标识一个进程，但是在网络中不行。

其实TCP/IP协议簇已经帮我们解决了这个问题：“网络层的IP地址”可以唯一标识网络中的主机，而传输层的“协议+端口”可以唯一标识主机中的应用程序（进程），这样利用三元组（IP地址，协议，端口）就可以标识网络的进程了，网络中的进程通信就可以利用这个标志与其它进程进行交互。

使用TCP/IP协议的应用程序通常采用应用编程接口：套接字(Socket)，来实现网路进程之间的通信。就目前而言，几乎所有的应用程序都是采用socket，而现在又是网络时代，网络中进程通信是无处不在，这就是我为什么说“一切皆socket”。

总结，网络中通过socket进行进程间通信。

# Socket介绍

## 概述

Socket在应用层和传输层之间的一个抽象层，它把 TCP/IP 层复杂的操作抽象成几个简单的接口，实现进程在网络中的通信，进程间通信通过socket来进行。

说白了，socket就是在传输层把tcp/ip协议给封装了一下，我们程序员一般都是面向socket来编程的。

## Socket通信模型

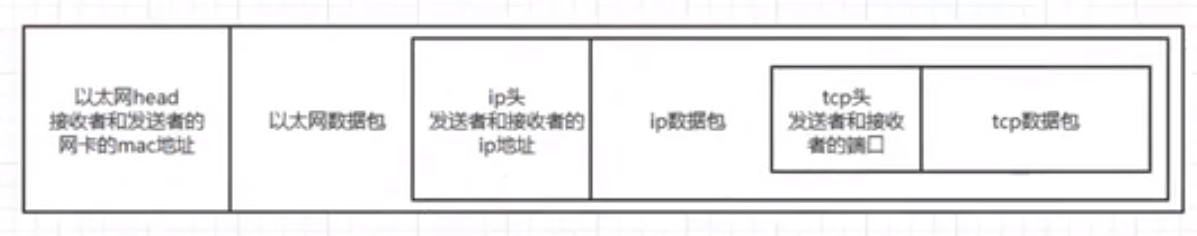
工作原理：先从服务器端说起。服务器端先初始化Socket，然后与端口绑定(bind)，对端口进行监听(listen)，调用accept阻塞等待客户端连接。在这时如果有个客户端初始化一个Socket，然后连接服务器(connect)，如果连接成功，这时客户端与服务器端的连接就建立了。客户端发送数据请求，服务器端接收请求并处理请求，然后把回应数据发送给客户端，客户端读取数据，最后关闭连接，一次交互结束。

其它底层，如建立连接和释放连接，都是基于tcp三次握手和四次挥手的规范来实现的。



Socket数据传输：

数据传输也是基于tcp协议的，把数据封装在tcp数据包，里面包括有tcp报头、端口号这些，然后封装在ip数据包，最后封装在以太网数据包进行传输。



# Http介绍

## 概述

http协议应用层协议，基于tcp/ip协议，是浏览器与服务器之间的数据传输协议。

http协议仅规定了数据传输格式，与数据如何传输无关(数据传输由tcp完成)。

http协议是一种数据包的格式，不涉及数据包的传输，数据包具体由tcp传输

## 特点

-1）无状态

http协议的无状态是针对每一次请求和响应的。http协议是应用层协议重点说明接收到的数据要如何解析，本身并不关心之前的请求是什么样，只要根据协议理解本次请求发过来的东西是什么就好。

对同一url请求没有记忆能力，当客户端再发送一次http请求时，http无法知道当前客户是“老用户”。

-2）长连接

http长连接是指建立一次tcp连接后长时间不关闭，供后面的请求使用，本质上是tcp的长连接。

目的：长连接省去了较多的tcp建立、关闭操作，使得每次发送请求不用重新建立tcp连接，减少开销。而保持连接的时间长短由服务端决定，比如tomcat中的server.xml文件就有相关配置。

http1.0默认短连接，http1.1默认长连接，短连接现在用的很少了。

实现：http长连接本质上是基于tcp连接的不关闭实现。而对于两次http请求来说依然没有直接关系的，所以说http是无状态的。

注：一个包含了许多图像的网页文件的多个请求和应答可以在一个连接中传输，但每个单独的网页文件的请求和应答仍然需要使用各自的连接。

保持连接有开销的，因此服务端会给连接设置超时时间，一段时间内连接上没请求数据就会关闭连接；以及设置最大连接数限制。

问：http的无状态和长连接矛盾吗？

无状态指对于每次http请求都是独立的，服务端没有保存客户端的状态。

长连接指tcp在单次请求完成后不关闭连接，下次再请求时可复用之前的连接。本质是tcp的长连接。

http无状态和长连接是不同层面上的东西，所以这两者之间不矛盾。

问：http连接超时情况？

1、服务端检测某个tcp连接超过一段时间没有发起请求，服务端会关闭这个连接。(在服务端设置超时值)

2、浏览器发起一个请求后，在等待服务端超过一段时间没有得到响应。(在浏览器设置超时值)

问：如果一个连接里面处理一个长事物，其他的请求会不会排队等待？

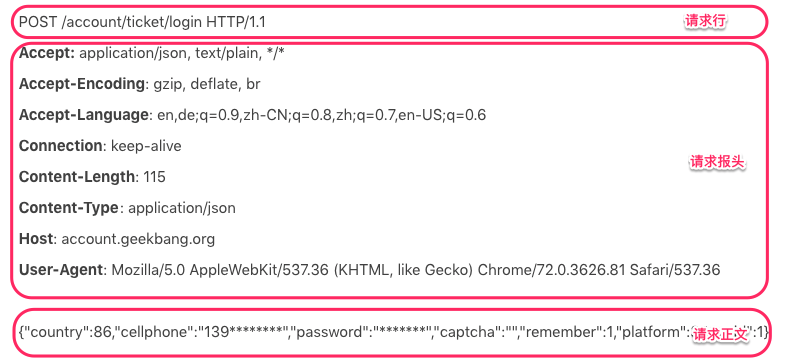
在http1.1中，请求是按顺序排队处理的，前面的http请求处理会阻塞后面的http请求。

虽然http pipelining 对连接请求做了改善，但是复杂度太大，并没有普及，这个问题在 http2.0 中得到了解决。

## http数据包

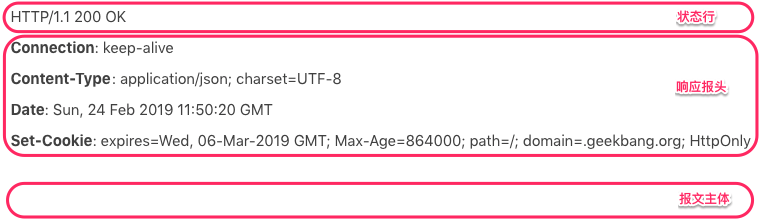
当http请求数据到达Tomcat后，Tomcat会把Http请求数据字节流解析成一个Request对象，这个Request对象封装了Http所有的请求信息。接着Tomcat把这个Request对象交给Web应用去处理，处理完后得到一个Response对象，Tomcat会把这个Response对象转换成Http格式的响应数据并发送给浏览器。

HTTP请求数据：请求行、请求头、请求正文



登录请求示例图

HTTP响应数据：状态行、响应头、报文主体



登录请求的响应示例图

## http状态码含义

状态码有三位数字组成，第一位数字定义了响应的类别，共分5类：

1XX：指示信息--表示请求已接收，继续处理

2XX：成功--表示请求已被成功接收、理解、接受

3XX：重定向--要完成请求必须进行更进一步的操作

4XX：客户端错误--请求有语法错误或请求无法实现

5XX：服务端错误--服务器未能实现合法的请求

## 浏览器输入URL后到获得页面过程？

问：从输入网址到获得页面的过程？浏览器输入URL发生什么？

1. 查询DNS，获取域名对应IP地址：-1）浏览器搜索自身的DNS缓存；-2）搜索操作系统的DNS缓存；-3）读取本地的HOST文件；-4）发起一个DNS的系统调用：1、宽带运营服务器查看本身缓存；2、运营服务器发起一个迭代DNS解析请求。

2. 浏览器获得域名对应的ip地址后，发起http三次握手

3. tcp/ip连接建立起来后，浏览器就可以向服务器发送http请求；

4. 服务器接收这个请求，根据路径参数，经过后端的一些处理生成html页面代码返回给浏览器；

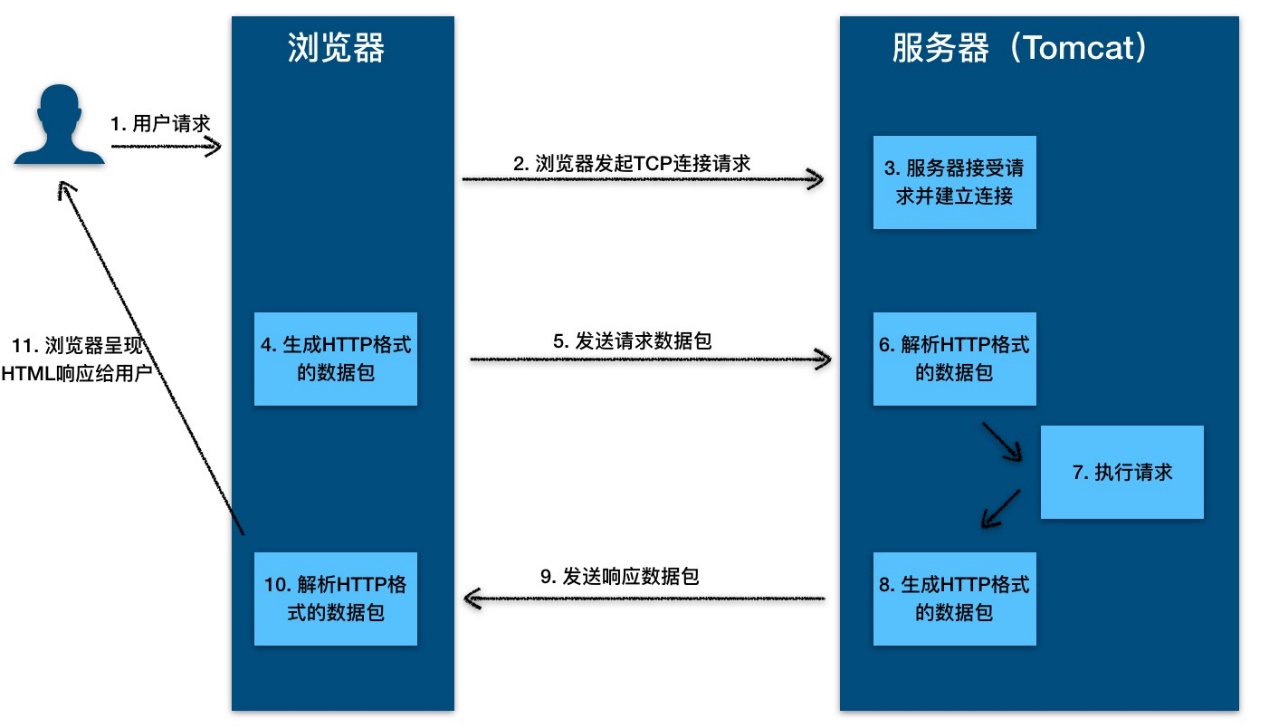
5. 浏览器拿到完整html页面代码开始解析和渲染，如果遇到引用的外部js、css、图片等资源，它们同样也是一个个http请求，都需经过上面步骤

6. 浏览器根据拿到的资源对页面进行渲染，最终把一个完整的页面呈现给用户。

## 发送http请求过程

浏览器从远程http服务器获取一个html文本过程中主要做两件事：

主要是两个过程：建立Socket连接和数据传输。建立连接需要经过tcp的三次握手；数据传输需要统一按照http协议规定的格式。



说明：

-1）用户通过浏览器访问某网址或点击某a标签；

-2）浏览器向服务器发送tcp连接请求；

-3）服务器接收浏览器的连接请求，并经过tcp三次握手后建立连接；

-4）浏览器将请求数据打包成http协议格式的数据包；

-5）浏览器将该数据包推入网络，数据包经过网络传输，最终达到服务端程序；

-6）服务端程序拿到这个数据包后，同样以http协议格式解包；

-7）服务器通过解包数据执行请求，生成响应结果；

-8）服务端将响应结果（html文本、图片）打包成http协议格式的数据包；

-9）服务器将响应数据包推入网络，数据包经过网络传输最终达到浏览器；

-10）浏览器拿到响应数据包后，以http协议的格式解包，然后解析数据，假设这里的数据是html；

-11）浏览器将html文件展现在页面上。

建立tcp连接说明：

在http工作开始之前web浏览器首先通过网络与web服务器建立连接。

连接是通过tcp实现，tcp协议与ip协议共同构建Internet，即著名的tcp/ip协议族，因此Internet又被称为tcp/ip网络。

http是比tcp更高层的应用层协议，根据规则，只有低层次协议建立之后才能进行更高层协议的连接，因此首先要建立tcp连接。一般tcp连接的端口是80。

Tomcat和Jetty作为http服务器，在这个过程中都做了什么？

接收并建立连接、解析请求数据、处理请求、发送响应

附：若有成千上万的浏览器请求同一个http服务器，因此Tomcat和Jetty为了提高服务的能力和并发度，往往会将自己要做的几个事情并行化，具体使用多线程技术。

## 为什么http不安全？

http通讯时容易出现中间人攻击行为。

http通讯时，若客户端c请求服务器s，可以通过网络抓包的形式来获取信息，甚至可以模拟服务器s端，来骗取与c端的通讯信息。

http协议被认为不安全是因为传输过程容易被监听者勾线监听、伪造服务器，而https协议主要解决的便是网络传输的安全性问题。

本地请求怎么被劫持？

修改host文件

中间人攻击？

参考https的中间人攻击。

有做过修改host文件来访问谷歌，这种就是通过中间人来访问谷歌的。

## http2.0介绍

http2.0特点：

1、多路复用技术。该技术可以只通过一个tcp连接就可以传输所有的请求数据。

2、新的编码机制。所有传输的数据都会被分割，并采用二进制格式编码。(以前httpb版本中通过文本的方式传输数据)

3、在http/2中有两个重要的概念：帧frame和流stream。帧代表着最小的数据单位，每个帧会标识出该帧属于哪个流，流也就是多个帧组成的数据流。

http2特点：https://http2.akamai.com/demo

# https介绍

## https流程

https流程：主要包括证书验证(非对称加密)和数据传输(对称加密)两大步骤。

--证书验证阶段：

-1）浏览器发起https请求，发送握手消息；

-2）服务端返回ssl证书、对称加密算法和公钥给客户端；

-3）客户端验证证书是否合法，若不合法则提示非安全证书。

--数据传输阶段：

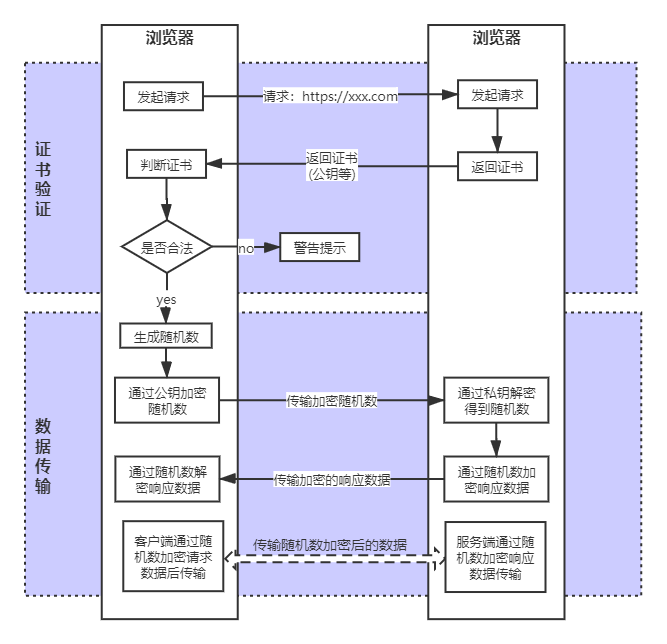
-1）当证书验证合法后，在本地生成随机数，并将随机数用证书里加密算法的公钥进行加密后发送给服务端；

-2）服务端通过私钥进行解密后得到随机数，并用此随机数作为密钥采用对称加密算法加密一段握手消息发给客户端；

-3）客户端收到消息后解密成功，则握手结束。

--发送请求：

客户端将请求参数通过随机数加密后传输，服务器使用随机数解密得到请求参数。



公钥、私钥和随机数的作用：

公钥和私钥的作用：客户端使用公钥对随机数加密传给服务端，服务端通过私钥解密得到随机数。

随机数作用：用于数据传输的对称加密算法的加密。

为什么数据传输是用对称加密？

1、非对称加密的加密解密效率非常低，而服务端与客户端之间存在大量的交互，非对称加密的效率无法接受；

2、在https的场景只有服务端保存了私钥，一对公私钥只能实现单向的加解密，所以https中内容传输加密采取的是对称加密，而不是非对称加密。

本地随机数被窃取怎么办？

描述：证书验证是采用非对称加密实现，但是传输过程是采用对称加密，而其中对称加密算法中重要的随机数是由本地生成并且存储于本地的，HTTPS如何保证随机数不会被窃取？

其实HTTPS并不包含对随机数的安全保证，HTTPS保证的只是传输过程安全，而随机数存储于本地，本地的安全属于另一安全范畴，应对的措施有安装杀毒软件、反木马、浏览器升级修复漏洞等。

问：如果我篡改了公钥呢？

公钥只是用来加密随机数的，篡改了会导致服务端拿不到随机数。

## https证书

https证书包含什么信息？

签发机构、用户信息、公钥、域名、有效期、指纹等信息。

浏览器如何验证ssl证书的合法性？

证书来源是否合法；

域名、有效期等信息是否正确；

证书是否被篡改，需要与ca服务器进行校验。

如何避免证书的冒用？

描述：证书是公开的，若要发起中间人攻击，我在官网上下载一份证书作为我的服务器证书，那客户端肯定会认同这个证书是合法的。

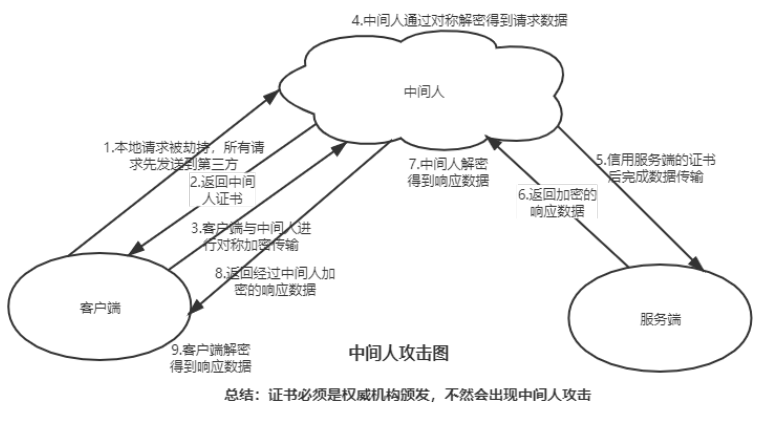
答案：证书来源不合法，且证书上的域名和网址上的域名不一致，客户端校验不通过。

证书的合法性依据是什么？

证书由ca认证机构颁发，不是随便一个机构都有资格颁发证书的。权威机构需要对其颁发的证书进行信用背书，我们能认为证书是合法的。但不同等级的权威机构对审核的要求也不一样，因此证书也分免费和付费的。

为什么需要ca认证机构颁发证书？

假设不存在认证机构，任何人都可以制作证书，这带来的安全风险就是经典的“中间人攻击”问题。



说明：

-1）本地请求被劫持（如DNS劫持等），所有请求均被发送到中间人的服务器；

-2）中间人服务器返回中间人自己的证书；

-3）客户端创建随机数，通过中间人证书的公钥对随机数加密后传送给中间人，然后凭随机数对传输内容进行加密传输；

-4）中间人因为拥有客户端的随机数，可以用过对称加密算法进行内容解密；

-5）中间人以客户端请求内容再向正规网站发起请求

-6）因为中间人与服务器的通信过程是合法的，正规网站通过建立的安全通道返回加密后的数据。

-7）中间人凭借与正规网站建立的对称加密算法对内容进行解密。

-8）中间人通过与客户端建立的对称加密算法对正规内容返回的数据进行加密传输。

-9）客户端通过与中间人建立的对称加密算法对返回结果数据进行解密。