# tcp/ip

1. tcp/ip的三次握手和四次挥手是什么概念，以及状态变化过程？

tcp/ip三次握手协议：客户端与服务端之间建立联系需要经过三次通讯确认通讯双方能正常发送接收消息，并建立连接的过程；第一次握手：客户发送SYN标识给服务端；第二次握手：服务端发送包含SYN/ACK给客户端；第三次：客户端收到消息后，发送ACK标识给服务端；连接建立完成。

tcp/ip四次挥手协议：服务器与客户端断开连接时，双方的一个确认过程；第一次挥手：客户端送FIN标识给服务端；第二次挥手：服务端收到客户端的消息后，发送ACK标识给客户端，通知客户端等待；

第三次挥手：服务端发送FIN标识给客户端；第四次挥手：客户端收到消息后，发送ACK标识给客户端，并关闭请求。

1. 建立连接需要3次，为什么断开连接需要4次？

断开连接需要4次挥手，因为在第二次挥手中服务端收到客户端的消息后，只是说明没有数据传输了，之前发送的需要一个处理过程，客户端进入等待状态，需要等待服务端的通知。

1. 三次握手有哪些不安全性？

三次握手过程：可能存在信息被窃取、信息被中间人篡改、服务端发送SYN攻击。

1. TCP和UDP的区别？TCP是通过什么方式来保证可靠性的
2. tcp四层网络模型和osi七层网络模型分别是什么？以及每一层的作用

tcp四层网络模型：（1）传输层：将要传输的HTTP报文增加TCP头；

（2）网络层：将传输层传过来的数据信息增加IP头（包括目的IP地址、源IP地址）；

（3）数据链路层：将网络层出入过来的数据信息增加MAC地址；

（4）物理层：将数据链路层出入过来的数据转化为二进制传输；

osi七层网络模型：（1）应用层：文件传输、服务、电子邮件、虚拟终端；

（2）表示层：数据格式化、代码转换、数据加密；

（3）会话层：解除或者建立与其他接点的联系；

（4）传输层、网络层、数据链路层、物理层与TCP/IP四层对应；

1. 什么是滑动窗口协议？它的实现原理是什么？

滑动窗口协议是网络传输中TCP控制流量传输的一种协议；原理：TCP协议通讯的两端都是全双工，接收端与发送端都有独立的缓冲区；

为了控制流量传输；TCP的接收端与维护端都维护一个窗口，发送端将缓冲中的数据读入到窗口，发送给接收端；当发送端收到接收端返回的数据后才进行下一段数据发送；每次成功发送数据并收到确认后，发送窗口就会在缓冲区按顺序移动。

1. 服务器上TIME\_WAIT状态的连接过多，怎么解决？

通过调整内核参数（/etc/sysctl.conf），然后执行/sbin/sysctl –p让参数生效：

（1）net.ipv4.tcp\_tw\_recycle = 1 ：表示开启TCP连接中TIME-WAIT sockets的快速回收，默认为0，表示关闭。

（2）net.ipv4.tcp\_fin\_timeout 修改系默认的 TIMEOUT 时间。

1. 什么是NIO、BIO、AIO？他们的区别？

BIO：同步阻塞I/O ；NIO：同步非阻塞I/O；AIO：异步非阻塞I/O

区别：BIO：客户端发送请求时，需要等待服务端有空闲线程；如果有的话，客户端其他线程会等待当前线程执行完再执行；

NIO：基于事件处理机制的同步非阻塞I/O；当socket有流可读或可写入socket时，操作系统会相应的通知引用程序进行处理，应用再将流读取到缓冲区或写入操作系统；NIO的最重要的地方是当一个连接创建后，不需要对应一个线程，这个连接会被注册到多路复用器上面，所以所有的连接只需要一个线程就可以搞定，当这个线程中的多路复用器进行轮询的时候，发现连接上有请求的话，才开启一个线程进行处理，也就是一个请求一个线程模式。

AIO：直接调用API的read或write方法即可。这两种方法均为异步的，对于读操作而言，当有流可读取时，操作系统会将可读的流传入read方法的缓冲区，并通知应用程序；对于写操作而言，当操作系统将write方法传递的流写入完毕时，操作系统主动通知应用程序。 即可以理解为，read/write方法都是异步的，完成后会主动调用回调函数；

1. 了解过多路复用吗？它是一个什么实现原理？

当创建一个连接后，这个连接会被注册到多路复用器上面，当这个线程中的多路复用器进行轮询的时候，发现连接上有请求的话，

才开启一个线程进行处理，也就是一个请求一个线程模式。

1. epool和select的区别是什么？

select：

最大并发数限制：使用32个整数的32位，即32\*32=1024来标识fd，虽然可修改，但是有以下第二点的瓶颈；

效率低：每次都会线性扫描整个fd\_set，集合越大速度越慢；

内核/用户空间内存拷贝问题。

epoll：

本身没有最大并发连接的限制，仅受系统中进程能打开的最大文件数目限制；

效率提升：只有活跃的socket才会主动的去调用callback函数；

省去不必要的内存拷贝：epoll通过内核与用户空间mmap同一块内存实现。

Epoll优势体体现在：高并发，且任一时间只有少数socket是活跃的；

引用网络上的一个事例：

Select： 收作业的班长一个个学生地去问有没有作业本，如果没有，它还得等待一段时间而后又继续问；

Epoll：收作业的班长第一次一个个学生地去问有没有作业本，只问一次，如果没有作业本，班长就告诉大家去哪里交作业本，当班长想要取作业本，就去那里看看或者等待一定时间后离开，有作业本到了就叫醒他，然后取走

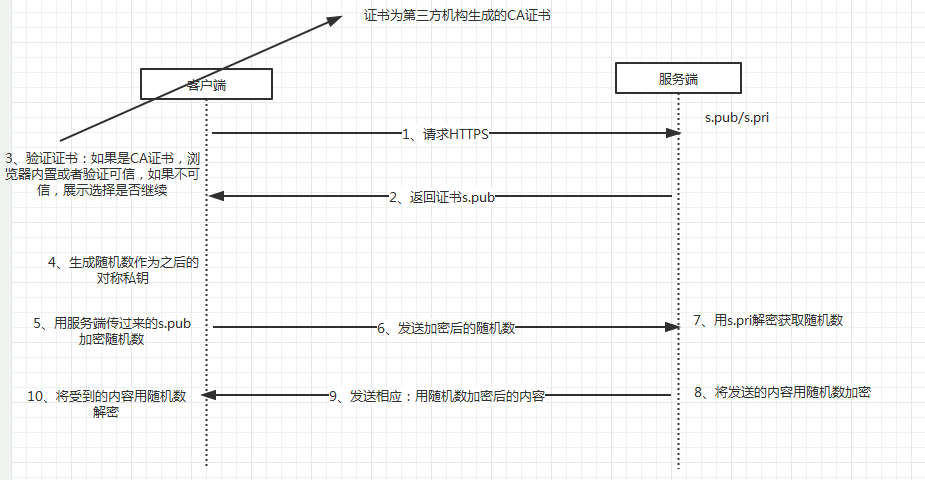
# http

1. http和https的区别

https在http的基础上增加了发送端与接收端证书检验；对传输中的报文进行了加密；

1. 简单说说https是如何保证安全传输的

简单来讲：https通过证书对之后通信中的对称秘钥进行了非对称加密；这样保证了对称秘钥不被中间人获取，就可以保证之后传输的信息不被中间人破解；具体流程如下图



1. https是不是绝对安全的？有没有办法被破解？

不是绝对安全的；中间人通过篡改证书，导致客户端安装不受信任证书。

1. http无状态协议，怎么理解无状态协议。如何实现有状态的请求

http无状态协议是指服务端不记录客户端发送的请求痕迹的状态；即：服务端认为客户端每次的请求都是第一次请求。

可以使用session会话实现一个有状态的请求。

1. 说说http协议中的302状态码的作用

302：重定向，当响应码为302时，表示服务器要求浏览器重新再发一个请求，服务器会发送一个响应头Location，它指定了新请求的URL地址。

1. 304缓存原理

当用户第一次请求index.html时，服务端会添加一个名为Last\_Modified, 这个头说明了index.html的最后修改时间，浏览器会把index.html内容，以及最后响应时间缓存下来。当用户第二次请求index.html时，在请求中包含一个名为If-Modified-Since(记录的是服务端传过来的index.html的最后修改时间)请求头，服务端收到请求后会将If-Modified-Since的值与index的最后修改时间对比，如果相同则告诉客户端，没有变化，直接显示index.html 。

1. http协议1.0和http协议1.1的区别

（1）HTTP 1.1支持长连接（PersistentConnection）和请求的流水线（Pipelining）处理

HTTP 1.0规定浏览器与服务器只保持短暂的连接，浏览器的每次请求都需要与服务器建立一个TCP连接，服务器完成请求处理后立即断开TCP连接，服务器不跟踪每个客户也不记录过去的请求。

HTTP 1.1则支持持久连接Persistent Connection, 并且默认使用persistent connection. 在同一个tcp的连接中可以传送多个HTTP请求和响应. 多个请求和响应可以重叠，多个请求和响应可以同时进行. 更加多的请求头和响应头(比如HTTP1.0没有host的字段).

HTTP 1.1的持续连接，也需要增加新的请求头来帮助实现，例如，Connection请求头的值为Keep-Alive时，客户端通知服务器返回本次请求结果后保持连接；Connection请求头的值为close时，客户端通知服务器返回本次请求结果后关闭连接。HTTP 1.1还提供了与身份认证、状态管理和Cache缓存等机制相关的请求头和响应头。

请求的流水线（Pipelining）处理，在一个TCP连接上可以传送多个HTTP请求和响应，减少了建立和关闭连接的消耗和延迟。例如：一个包含有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。 HTTP 1.1还允许客户端不用等待上一次请求结果返回，就可以发出下一次请求，但服务器端必须按照接收到客户端请求的先后顺序依次回送响应结果，以保证客户端能够区分出每次请求的响应内容。

（2）HTTP 1.1增加host字段

在HTTP1.0中认为每台服务器都绑定一个唯一的IP地址，因此，请求消息中的URL并没有传递主机名（hostname）。但随着虚拟主机技术的发展，在一台物理服务器上可以存在多个虚拟主机（Multi-homed Web Servers），并且它们共享一个IP地址。

HTTP1.1的请求消息和响应消息都应支持Host头域，且请求消息中如果没有Host头域会报告一个错误（400 Bad Request）。此外，服务器应该接受以绝对路径标记的资源请求。

1. 如何保证基于http协议的接口的安全性

（1）重要数据加密（2）非重要数据验签

1. http协议上传文件，数据如何传输？

在最初的 http 协议中，没有上传文件方面的功能。 rfc为 http 协议添加了这个功能。客户端的浏览器，如 Microsoft IE, Mozila, Opera 等，按照此规范将用户指定的文件发送到服务器。服务器端的网页程序，如 php, asp, jsp 等，可以按照此规范，解析出用户发送来的文件。

Microsoft IE, Mozila, Opera 已经支持此协议，在网页中使用一个特殊的 form 就可以发送文件。

绝大部分 http server ，包括 tomcat ，已经支持此协议，可接受发送来的文件。

各种网页程序，如 php, asp, jsp 中，对于上传文件已经做了很好的封装。

1. 说说http协议的优缺点

优点：灵活、无状态；

缺点：不安全；

1. 一次http请求的完整交互流程

域名解析 --> 发起TCP的3次握手 --> 建立TCP连接后发起http请求 --> 服务器响应http请求，浏览器得到html代码 --> 浏览器解析html代码，并请求html代码中的资源（如js、css、图片等） --> 浏览器对页面进行渲染呈现给用户

# 序列化

1. 什么是序列化，Java是如何实现序列化的

序列化就是一种用来处理对象流的机制，所谓对象流也就是将对象的内容进行流化。可以对流化后的对象进行读写操作，也可将流化后的对象传输于网络之间。序列化是为了解决在对对象流进行读写操作时所引发的问题。

Java中实现序列化的方式：实现java自带的Serializable接口；如果一个子类实现了序列化，父类没有实现，那么父类中的成员变量能否被序列化？

不会。

1. 你有了解过哪些序列化技术？以及他们之间的差异性？
2. XML、JSON、FastJSON、HESSION、ProtoBuf。
3. transient是干嘛的？

Transient用来声明瞬态的，变量修饰符；被Transient修饰的变量不会被序列化。

1. 有什么方法能够绕过transient的机制。这个实现机制的原理是什么？

在类中实现readObect、writeObject ,使用输入与输出流调用defaultWriteObject()与defaultReadObject();

*/\*\*  
 \* 通过实现writeObject与readObject实现瞬态变量序列化  
 \* @param objectOutputStream  
 \* @throws IOException  
 \*/*private void writeObject(ObjectOutputStream objectOutputStream)throws IOException{  
 objectOutputStream.defaultWriteObject();  
 objectOutputStream.writeObject(**age**);  
}  
private void readObject(ObjectInputStream objectInputStream)throws Exception{  
 objectInputStream.defaultReadObject();  
 **age** = (int) objectInputStream.readObject();  
}

1. serializable的安全性如何保证？

加密：md5等方式；

1. 有没有了解过protobuf，它的序列化实现原理是什么？

Protocol Buffers 是一种轻便高效的结构化数据存储格式，可以用于结构化数据串行化，或者说序列化。它很适合做数据存储或数据交换格式。可用于通讯协议、数据存储等领域的语言无关、平台无关、可扩展的序列化结构数据格式。Protobuf的编码方式使用的是varint，存储方式用的T-L-V(Tag-Length-Value)；整数编码：正数使用的是int32，负数使用的是ZigZag。

1. serialVersionUID的作用是什么？如果我不设置serialVersionUID,有没有问题？

serialVersionUID 用来表明类的不同版本间的兼容性。如果不设置serialVersionUID，会产生一个默认值。