# 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？

答：tcp/ip的三次握手是TCP/IP建立通信连接的过程，而四次挥手是断开连接的过程。以下的客户端和服务端都是相对而言。

在三次握手的过程中，客户端发起连接，向服务端发送SYN/SEQ数据包，自身状态变为SYN-SEND，表示自身已经发送了建立连接的确认包了，等待服务端的响应，服务端接收到客户端发送的数据包的时候，会向客户端发送一个SYN/ACK的数据包，表名服务端自身可以建立通信通道，客户端接收到这个数据包的时候，自身会最后以此发送数据包，自身状态会变为ESTABLESHED，表名客户端已经准备好建立连接了，服务端接收到这个数据包，状态也会变为ESTABLESHED，然后连接通道就建立了。

在四次挥手的过程中，客户端向服务端发送一个FIN的数据包，表明这边已经准备好关闭连接了，服务端接收到客户端的FIN的数据包的时候，会向客户端先发送一个ACK的数据包，只是表明收到啦客户端发送过来的数据包，但是还没有准备好关闭连接，因为服务端可能还有没有执行完的任务或者消息，当服务端任务已经全部执行完，可以关闭连接但是，会再次向客户端发送一个FIN的数据包，表明已经可以关闭连接了，客户端收到FIN的数据包的时候，会向服务端发送一个ACK的数据包，并且客户端会关闭连接了，服务端接收到也会关闭连接了。

1. 建立连接需要3次，为什么断开连接需要4次？

答：因为建立连接的时候，服务端在接收到客户端发送过来的FIN数据包的时候，可以直接向客户端发送ACK+SYN数据包，表明可以建立连接，但是在断开连接的时候，服务端接收到客户端的断开连接的请求之后，并不能马上向客户端发送ACK+FIN的数据包，因为可能还有任务或者消息没有处理完，只能先向客户端回复一个ACK的数据包，表明服务端已经收到了客户端发送的断开连接的请求，等任务或者消息全部处理完再向客户端发送一个FIN的数据包，表明可以关闭链接了，所以断开连接需要四次。

1. 三次握手有哪些不安全性？

答：1.DOS攻击：攻击者故意不完成三次握手连接，但是一直请求三次握手，导致服务器内存、CPU等资源耗尽，甚至服务器崩溃。

2.在TCP建立连接三次握手的过程中，伪造TCP的数据包进行攻击：a.攻击者监听服务器端发送的SYN/ACK的数据包；b.攻击者向服务器发送RST数据包，并且接着又发送SYN数据包，假装成正常的客户端，来建立三次握手的过程；c.服务端响应新连接，并发送连接响应报文SYN/ACK；d.攻击者再假冒A方对B方发送ACK包。

1. TCP和UDP的区别？TCP是通过什么方式来保证可靠性的

答：TCP和UDP都是基于IP协议的传输层协议，与UDP不同的是，TCP协议是一个稳定可靠的协议，他能解决IP协议提供的数据报文服务所产生的数据包丢失、重发、顺序错乱等等问题，而UDP没有解决此类问题。TCP协议利用滑动窗口机制实现对发送方的流量控制，这样服务端接收到一个数据祯才能接收下一个数据帧，保证TCP连接的可靠性；滑动窗口机制的窗口单位是字节不是报文段，发送端的发送窗口不能超过接收到的接收窗口给出的数值。

1. 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

答：TCP四层网络模型为：传输层、网络层、数据链路层、物理层：

传输层：主要有TCP和UDP两种协议，主要是在数据报文上拼接包含源端口号和目标端口号的头信息，

网络层：进行逻辑地址寻址，在传输层的基础上拼接包含目标IP地址和源IP地址等信息的IP头，

数据链路层：建立逻辑连接、硬件地址的寻址，并且拼接包含源MAC地址和目标MAC地址的MAC头信息，

物理层：建立、维护、断开物理连接，物理成传输的是二进制流；

OSI七层网络模型为：应用层、表示层、会话层、传输层、网络层、数据链路层、物理层：

应用层：网络服务与最终服务的一个接口，协议有http、ftp、smtp、dns、https等等，

表示层：数据的表示、安全、压缩，

会话层：建立、管理、结束会话，是指本地主机与服务端主机的会话，

传输层：主要有TCP和UDP两种协议，主要是在数据报文上拼接包含源端口号和目标端口号的头信息，

网络层：进行逻辑地址寻址，在传输层的基础上拼接包含目标IP地址和源IP地址等信息的IP头，

数据链路层：建立逻辑连接、硬件地址的寻址，并且拼接包含源MAC地址和目标MAC地址的MAC头信息，

物理层：建立、维护、断开物理连接，物理成传输的是二进制流；

1. 什么是滑动窗口协议？它的实现原理是什么？

答：滑动窗口协议是一种流量控制的技术，其实现原理：发送方和接收方都会维护一个数据帧的序列，这个序列称为滑动窗口。发送方的窗口大小由接收方的窗口大小决定，以免导致数据来不及接收进而数据溢出的现象。同时流量控制也能够很好的解决网络堵塞的所造成的问题，当客户端发送的数据帧发送出去了，但是没有接收到服务端发送过来的ack的确认信息的包的时候，后面的数据帧不会继续发送，接收到数据包才往后面的数据帧发送，这个过程称之为滑动。

1. 服务器上TIME\_WAIT状态的连接过多，怎么解决？

答：1. linux没有在sysctl或者proc文件系统暴露修改这个TIMEWAIT超时时间的接口，可以修改内核协议栈代码中关于这个TIMEWAIT的超时时间参数，重编内核，让它缩短超时时间，加快回收；  
 2. 利用SO\_LINGER选项的强制关闭方式，发RST而不是FIN，来越过TIMEWAIT状态，直接进入CLOSED状态。

1. 什么是NIO、BIO、AIO？他们的区别？

答：BIO为同步阻塞，服务器的实现模式为一个连接一个线程，即客户端有连接请求的时候服务器就需要创建一个线程去处理这个请求，如果这个连接不坐任何事情，会造成不必要的线程开销，不过可以通过线程池改善；适用于连接数目比较小而且固定的架构，jdk1.4之前只有这种模式；

NIO为同步非阻塞，服务器的实现模式为一个请求一个线程，即客户端发送的连接请求都会发送到注册到多路复用器上面，多路复用器轮询到连接有I/O请求时，才会分配线程去处理；适用于连接数目比较多且连接比较短的架构，比如聊天，jdk1.4开始支持。

AIO是异步非阻塞。服务器的实现模式为一个有效请求一个线程，即客户端的I/O请求都是由OS先完成了再通知服务器应用去启动线程并且进行处理。适用于连接数目比较多且连接比较长的架构，比如相册服务器，jdk1.7开始支持。

1. 了解过多路复用吗？它是一个什么实现原理？

答：多路复用是指统一传输媒介（线路）承载多路信号进行通信的方式。

1. epoll和select的区别是什么？

答：1.epoll没有最大并发丽连接的限制，而select由最大连接限制；

1. epoll相比select效率会高，select是每次都要线性扫描整个集合，集合越大，效率越低，而epoll只有活跃的socket才会主动的调用callback函数；
2. epoll相比select省去了内存拷贝的操作，节省了内存资源

# http

1. http和https的区别

答：1.http所传输的数据报文是明文的，容易被泄露和串改，https所传输的内容是经过加密处理的，不会被泄露和串改；

2.https相比http在传输层上面加了一层SSL（Socket安全层）或者TSL（传输安全层），网络分层略有不同；

1. 简单说说https是如何保证安全传输的

答：

1. https是不是绝对安全的？有没有办法被破解？

答：https也不是绝对安全的，是相对安全的，如果用户信赖了一些不安全的证书，可能会产生安全问题。

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

答：http请求是无状态的，每一次的http请求无法携带上一次的http请求的状态，都是相互独立的。为了实现有状态的请求，我们引入了cookie技术，在第一次的请求中服务器把状态信息写入到客户端的cookie中，然后下一次客户端发起请求就会携带cookie中的信息来实现一个状态信息的保存的效果。

1. 说说http协议中的302状态码的作用

答：302 Found 临时移动。与301类似。但资源只是临时被移动。客户端应继续使用原有URI       301 Moved Permanently 永久移动。请求的资源已被永久的移动到新URI，返回信息会包括新的URI，浏览器会自动定向到新URI。今后任何新的请求都应使用新的URI代替。

1. 304缓存原理

答：304 ：客户端有缓冲的文档并发出了一个条件性的请求。服务器告诉客户端，原来缓冲的文档还可以继续使用。

1. http协议1.0和http协议1.1的区别

答：A http1.1相比http1.0支持了长连接和请求的流水线处理；

B HTTP1.1 新增了字段host，其请求信息和响应信息都包含host头域，并且如果没有host头域或报400的一个错误；

C 新增状态码100（继续），客户端事先向服务端发送一个带头域的请求，如果服务端接收了这个请求，就会向客户端返回一个状态码100，表示客户端可以发送带实体的完整的请求了；

D http1.1引入了Chuked transfer-coding

E http1.1在http1.0的基础上加入了一些cache的新特性。当缓存对象的Age超过Expire时变为stale对象，cache不需要直接抛弃stale对象，而是与源服务器进行重新激活（revalidation）。

1. 如何保证基于http协议的接口的安全性

答：主要通过以下两种方式来保证http协议的接口的安全性：

一种是以支付宝等支付公司为代表的私钥公钥签名验证机制；

一种是普遍采用的参数签名验证机制；

1. http协议上传文件，数据如何传输？

答：使用http的PUT方法进行文件的传输：PUT要求在请求报文的主体中包含文件内容，然后保存到请求URL指定的位置处于安全考虑，一般web网站不使用此方法，若配合web的安全验证机制，或者架构采用REST标准的网站，就可能开放使用此方法

1. 说说http协议的优缺点

答：优点：1.基于应用级的接口使用方便

    2.要求的开发水平不高，容错性强

缺点： 1.传输速度慢，数据包大。

    2.如实现实时交互，服务器性能压力大

    3.数据传输安全性差

1. 一次http请求的完整交互流程

答：用户发送请求，然后请求被解析成http报文，在OSI四层网络模型中，该请求报文首先会进过传输层，传输层协议主要有TCP和UDP两种，以TCP为例，在我们的传输层上面会往http报文的头部拼接TCP头，TCP的头部信息会包含源端口号，目标服务器端口号；经过传输层之后，会进入到网络层，网络层都是基于IP协议的，在原有已经拼接了TCP头的基础上再拼接IP头，IP头会包含源IP地址和目标IP地址，通过这个IP地址，找到目标服务器；经过网络层我们的报文会进入到数据链路层，在数据链路层中会在原有基础上拼接MAC头，包含了目标MAC地址信息和源MAC地址信息，用这个MAC地址信息来确认目标服务机器以及我们客户端的机器MAC地址信息，这个MAC地址信息每台机器是唯一的，最后进入我们的物理层，在物理层，吧我们的报文信息转换成二进制流以便进行传输；而进入到服务器上面则是上述过程的逆过程，来一步步解析客户福安发送的信息。

# 序列化

1. 什么是序列化，Java是如何实现序列化的

答：序列化是指把对象信息转化为可存储的、可传输的二进制流的形式；JAVA通过实现serializable接口来实现序列化操作。

1. 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？

答：父类中的成员变量不会被序列化。

1. 你有了解过哪些序列化技术？以及他们之间的差异性？

答：我了解的常见的序列化技术有：

Java序列化：java语言本身的序列化，比较简单易用，但是不支持跨语言；

XML序列化：序列化形成的内容可读性强，可以比较容易的读取其中的信息，比如webservice就是使用的xml序列化，但是序列化产生的文件比较大，性能比较低，适用于对性能要求不高的场景；

JSON序列化：交互性强，可读性也还好，相比XML，其产生的数据量不是很大，应用比较广泛，前后端分离的项目肯定是使用json序列化，常见的json序列有阿里巴巴的fastJson和谷歌的GSON；

Hessian序列化：相比JAVA自身的序列化，支持跨语言，并且性能也比他好，dubbo应用的序列化框架是hessian不过 重构了以下；

Protobuf序列化：产生的数据量极小，效率极高，适用于那些对性能要求很高的场景，类似的高性能序列化框架还有protosuff、thrift；

1. transient是干嘛的？

答：transient是用来控制序列化的，在变量前面加上这个关键字，该变量就不会被序列化，在其反序列化过程中对应的变量会被赋予初始值；

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？

答：通过自己使用IO流来编写对应变量的序列化方法，来实现绕过transient。

1. serializable的安全性如何保证？

答：

1. 有没有了解过protobuf，它的序列化实现原理是什么？

答：protobuf是一种序列化框架，其序列化生成的数据量极小，序列化所花费的时间或者说序列化的效率也很高，十分适用于对性能要求很高的场景。

其序列化实现原理是通过varint算法实现的。

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

答：serialVersionUID的作用是保证客户端的对象的版本与服务端反序列化生成的版本一致所使用的；如果不设置serialVersionUID的话，也不要紧，因为JAVA编译器会自动对class信息写一个摘要信息来作为其版本信息，类似于指纹算法。