### 1，基础知识

* ip地址和端口号：ip地址是唯一的通讯实体，一个端口号对应一个通讯实体
* tcp/udp协议：udp是无连接协议，可以通过任何途径达到目的地，不可靠，但是效率高，可以用来做做视频聊天

tcp需要三次握手，是可靠的协议，传输效率低，

* URL：统一的资源定位器，
* InetAddress，java提供的一个地址类
* http连接的特点：客户端发送的每次请求都需要服务器回送响应，在请求结束后，会主动释放连接。从建立连接到关闭连接的过程称为一次连接。
* 网络分层：OSI七层模型（应用层，表示层，会话层，传输层，网络层，数据链路层，物理层）

### 2，Socket

创建socket实例，大部分基于



image.png

客户端连接：

1，创建socket对象

2，连接建立后，通过输出流向服务器端发送请求信息

3，通过输入流获取服务器响应的信息

4，关闭响应资源

服务端连接：

1，创建ServerSocket对象，绑定监听端口

2，通过accept()方法监听客户端请求

3，连接建立后，通过输入流读取客户端发送的请求信息

4，通过输入流向客户端发送信息

5，关闭相关资源

### 3，Socket建立网络连接的步骤：

建立Socket连接至少需要一对套接字，其中一个运行与客户端--ClientSocket，一个运行于服务端--ServiceSocket

1、服务器监听：服务器端套接字并不定位具体的客户端套接字，而是处于等待连接的状态，实时监控网络状态，等待客户端的连接请求。

2、客户端请求：指客户端的套接字提出连接请求，要连接的目标是服务器端的套接字。注意：客户端的套接字必须描述他要连接的服务器的套接字，

指出服务器套接字的地址和端口号，然后就像服务器端套接字提出连接请求。

3、连接确认：当服务器端套接字监听到客户端套接字的连接请求时，就响应客户端套接字的请求，建立一个新的线程，把服务器端套接字的描述

发给客户端，一旦客户端确认了此描述，双方就正式建立连接。而服务端套接字则继续处于监听状态，继续接收其他客户端套接字的连接请求。

### 4，Socket和http的区别

Http协议：简单的对象访问协议，对应于应用层。Http协议是基于TCP链接的。

tcp协议：对应于传输层

ip协议：对应与网络层

TCP/IP是传输层协议，主要解决数据如何在网络中传输；而Http是应用层协议，主要解决如何包装数据。

Socket是对TCP/IP协议的封装，Socket本身并不是协议，而是一个调用接口（API），通过Socket，我们才能使用TCP/IP协议。

Http连接：http连接就是所谓的短连接，及客户端向服务器发送一次请求，服务器端相应后连接即会断掉。

socket连接：socket连接及时所谓的长连接，理论上客户端和服务端一旦建立连接，则不会主动断掉；但是由于各种环境因素可能会是连接断开，比如说：服务器端或客户端主机down了，网络故障，或者两者之间长时间没有数据传输，网络防火墙可能会断开该链接已释放网络资源。所以当一个socket连接中没有数据的传输，那么为了位置连续的连接需要发送心跳消息，具体心跳消息格式是开发者自己定义的。

### 5，java的IO接口

基于字节，字符，磁盘，网络操作的IO接口

（BIO）阻塞式IO的通讯模型，如下图（） 

BIO数据在写入OutputStream

### 6， TCP 与 UDP 的区别

* TCP面向连接（如打电话要先拨号建立连接）;UDP是无连接的，即发送数据之前不需要建立连接
* TCP提供可靠的服务。也就是说，通过TCP连接传送的数据，无差错，不丢失，不重复，且按序到达;UDP尽最大努力交付，即不保证可靠交付
* TCP面向字节流，实际上是TCP把数据看成一连串无结构的字节流;UDP是面向报文的 UDP没有拥塞控制，因此网络出现拥塞不会使源主机的发送速率降低（对实时应用很有用，如IP电话，实时视频会议等）
* 每一条TCP连接只能是点到点的;UDP支持一对一，一对多，多对一和多对多的交互通信
* TCP首部开销20字节;UDP的首部开销小，只有8个字节
* TCP的逻辑通信信道是全双工的可靠信道，UDP则是不可靠信道

### 7，TCP 的 3 次握手和四次挥手

第一次握手：客户端发送syn包(syn=j)到服务器，并进入SYN\_SEND状态，等待服务器确认；

第二次握手：服务器收到syn包，必须确认客户的SYN（ack=j+1），同时自己也发送一个SYN包（syn=k），即SYN+ACK包，此时服务器进入SYN\_RECV状态；

第三次握手：客户端收到服务器的SYN＋ACK包，向服务器发送确认包ACK(ack=k+1)，此包发送完毕，客户端和服务器进入ESTABLISHED状态，完成三次握手。

握手过程中传送的包里不包含数据，三次握手完毕后，客户端与服务器才正式开始传送数据。理想状态下，TCP连接一旦建立，在通信双方中的任何一方主动关闭连接之前，TCP 连接都将被一直保持下去。断开连接时服务器和客户端均可以主动发起断开TCP连接的请求，断开过程需要经过“四次挥手”

第一次挥手：客户端发送报文告诉服务器没有数据要发送了

第二次挥手：服务端收到，再发送给客户端告诉它我收到了

第三次挥手：服务端向客户端发送报文，请求关闭连接

第四次挥手：客户端收到关闭连接的请求，向服务端发送报文，服务端关闭连接

### 8，TCP为什么三次握手不是两次握手，为什么两次握手不安全

为了实现可靠数据传输， TCP 协议的通信双方， 都必须维护一个序列号， 以标识发送出去的数据包中， 哪些是已经被对方收到的。 三次握手的过程即是通信双方相互告知序列号起始值， 并确认对方已经收到了序列号起始值的必经步骤

如果只是两次握手， 至多只有连接发起方的起始序列号能被确认， 另一方选择的序列号则得不到确认

### 9，为什么TCP是可靠的，UDP早不可靠的?为什么UDP比TCP快?

TCP/IP协议拥有三次握手双向机制，这一机制保证校验了数据，保证了他的可靠性。

UDP就没有了，udp信息发出后,不验证是否到达对方,所以不可靠。

### 10，Http的get和post的区别

http是应用层的协议，底层基于TCP/IP协议，所以本质上，get和post请求都是TCP请求。所以二者的区别都是体现在应用层上（HTTP的规定和浏览器/服务器的限制）

1.参数的传输方式：GET参数通过URL传递，POST放在Request body中。

2.GET请求在URL中传送的参数是有长度限制的，而POST没有。

3.对于GET方式的请求，浏览器会把http header和data一并发送出去，服务器响应200（返回数据）；而对于POST，浏览器先发送header，服务器响应100 continue，浏览器再发送data，服务器响应200 ok（返回数据）。不过要注意，并不是所有浏览器都会在POST中发送两次包，比如火狐

4.对参数的数据类型，GET只接受ASCII字符，而POST没有限制。

5.GET比POST更不安全，因为参数直接暴露在URL上，所以不能用来传递敏感信息。

6.GET请求只能进行url编码，而POST支持多种编码方式。

7.GET在浏览器回退时是无害的，而POST会再次提交请求。

8.GET产生的URL地址可以被Bookmark，而POST不可以。

9.GET请求会被浏览器主动cache，而POST不会，除非手动设置。

### 11，socket和http的区别

Http协议：简单的对象访问协议，对应于应用层。Http协议是基于TCP链接的。

tcp协议：对应于传输层

ip协议：对应与网络层

TCP/IP是传输层协议，主要解决数据如何在网络中传输；而Http是应用层协议，主要解决如何包装数据。

Socket是对TCP/IP协议的封装，Socket本身并不是协议，而是一个调用接口（API），通过Socket，我们才能使用TCP/IP协议。

Http连接：http连接就是所谓的短连接，及客户端向服务器发送一次请求，服务器端相应后连接即会断掉。

socket连接：socket连接及时所谓的长连接，理论上客户端和服务端一旦建立连接，则不会主动断掉；但是由于各种环境因素可能会是连接断开，比如说：服务器端或客户端主机down了，网络故障，或者两者之间长时间没有数据传输，网络防火墙可能会断开该链接已释放网络资源。所以当一个socket连接中没有数据的传输，那么为了位置连续的连接需要发送心跳消息，具体心跳消息格式是开发者自己定义的。

### 12，描述一次网络请求的流程

1)域名解析

浏览器会先搜索自身DNS缓存且对应的IP地址没有过期；若未找到则搜索操作系统自身的DNS缓存；若还未找到则读本地的hotsts文件；还未找到会在TCP/IP设置的本地DNS服务器上找，如果要查询的域名在本地配置的区域资源中，则完成解析；否则根据本地DNS服务器会请求根DNS服务器；根DNS服务器是13台根DNS，会一级一级往下找。

2）TCP三次握手

客户端先发送SYN=1，ACK=0，序列号seq=x报文；（SYN在连接建立时用来同步序号，SYN=1，ACK=0代表这是一个连接请求报文，对方若同意建立连接，则应在响应报文中使SYN=1，ACK=1）

服务器返回SYN=1，ACK=1，seq=y, ack=x+1；

客户端再一次确认，但不用SYN了，回复服务端, ACK=1, seq=x+1, ack=y+1

3）建立TCP连接后发起HTTP请求

客户端按照指定的格式开始向服务端发送HTTP请求，HTTP请求格式由四部分组成，分别是请求行、请求头、空行、消息体，服务端接收到请求后，解析HTTP请求，处理完成逻辑，最后返回一个具有标准格式的HTTP响应给客户端。

4）服务器响应HTTP请求

服务器接收处理完请求后返回一个HTTP响应消息给客户端，HTTP响应信息格式包括：状态行、响应头、空行、消息体

5）浏览器解析HTML代码，请求HTML代码中的资源

浏览器拿到html文件后，就开始解析其中的html代码，遇到js/css/image等静态资源时，向服务器发起一个http请求，如果返回304状态码，浏览器会直接读取本地的缓存文件。否则开启线程向服务器请求下载。

6）浏览器对页面进行渲染并呈现给用户

7）TCP的四次挥手

当客户端没有东西要发送时就要释放连接（提出中断连接可以是Client也可以是Server），客户端会发送一个FIN=1的没有数据的报文，进入FIN\_WAIT状态，服务端收到后会给客户端一个确认，此时客户端不能发送数据，但可接收信息。

### 13，Http位于TCP/IP 模型中的第几层？为什么说Http是可靠的数据传输协议？

TCP/IP 的五层模型：

从下到上：物理层-数据链路层-网络层-传输层-应用层

其中Tcp/ip 位于模型中的网络层，处于同一层的还有ICMP（网络控制信息协议），http位于模型中的应用层

由于tcp/ip是面向链接的可靠协议，而http是在传输层基于tcp/ip西仪的，所以说http也是可靠的数据传输协议

### 14，http，https的区别

此处延伸：https的实现原理

1、https协议需要到ca申请证书，一般免费证书较少，因而需要一定费用。

2、http是超文本传输协议，信息是明文传输，https则是具有安全性的ssl加密传输协议。

3、http和https使用的是完全不同的连接方式，用的端口也不一样，前者是80，后者是443。

4、http的连接很简单，是无状态的；HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的网络协议，比http协议安全。

https实现原理：

（1）客户使用https的URL访问Web服务器，要求与Web服务器建立SSL连接。

（2）Web服务器收到客户端请求后，会将网站的证书信息（证书中包含公钥）传送一份给客户端。

（3）客户端的浏览器与Web服务器开始协商SSL连接的安全等级，也就是信息加密的等级。

（4）客户端的浏览器根据双方同意的安全等级，建立会话密钥，然后利用网站的公钥将会话密钥加密，并传送给网站。

（5）Web服务器利用自己的私钥解密出会话密钥。

（6）Web服务器利用会话密钥加密与客户端之间的通信。