1. **TCP三次握手**

所谓三次握手(Three-way Handshake)，是指建立一个TCP连接时，需要客户端和服务器总共发送3个包。

1. 客户端向服务端发送连接请求报文字段，该报文段头部SYN（发送一个新连接）=1，ACK（确定序号有效） =0，seq（序号） = x。此时进去了syn——send状态
2. 服务端收到了请求报文端后，若同意连接，则发送一个应答，SYN = 1,ACK = 1,seq = y,ack = x=1
3. 客户端收到连接同意的应答后，还要向服务端发送一个确认报文字段，表明服务端发来的连接同意已经收到。ACK = 1，seq = x+1；ack = y+1

三次原因： 是为了防止已经失效的连接请求报文段突然传到了服务端，当客户端的第一个请求因为各种原因在网络节点延迟，直到连接释放的某个时间才到服务端。

1. **TCP四次挥手**
   1. A向B发送连接释放请求，此时A进入了FIN\_WAIT状态。FIN（释放一个请求） = 1，seq = x，Ack = z
   2. B收到A连接释放请求后，会通知相关应用程序；告知A到B这个连接已经释放，此时B进入close—WAIT状态。Ack = x+1，seq = z
   3. 当B向A发完所有的数据后，向A发送连接释放请求；FIN = 1；ACK =1，seq = W ack = u+1
   4. 当A收到连接释放请求后，进入到2ms的TIME—WAIT状态。最后发ACK= 1，seq = x

CLOSE—WAIT

1 .可靠实现了TCP全双工链接的终止

因为无法保证最后收到的ack报文一定会被收到，这个状态可以用来重发可能丢失的ack报文

2.允许老的重复分节在网络中消逝

如果应用程序立即建立一个和刚关闭链接相似的连接，新的连接可能会受到老的连接吃到的报文影响。

1. Tcp可靠传输的实现

流量控制，拥塞控制

滑动串口协议：有一个发送窗口A和一个接收窗口B；发送窗口在没有收到B确认的情况下，A也可以连续把发送窗口的数据发送出去，但是在未确认前，该数据会暂时保留以便超时重传。

其维护了发送缓冲区和接收缓冲区。发送缓冲区有：准备发送的数据和已经发送但未

确认的数据。

接收缓冲区有：1.按序到达但未被应用程序读取的data；2.未按序到达的data

后退n帧协议：

选择重传：

流量控制怎么控制死锁？

持续计数器：当发送窗口接收到一个窗口为0的应答后，就启动持续计数器。时间一到就主动发送报文询问接收窗口的大小。

拥塞控制：

慢开始和拥塞避免算法；快重传和快恢复

慢开始：

1. 通信开始的时候，发送方的发送窗口设为1，并发送第一个分组M1.
2. 接收方收到M1后，返回确认应答，此时发送方发送窗口扩大两倍，并发送M2
3. 若发送窗口 》 慢开始门限，则使用拥塞避免算法，每次确认应答后将发送窗口+1
4. 如发送方出现了超时重传，则表明网络出现了阻塞；此时慢开始门限设为当前发送窗口的一半；发送窗口设定为1，启动拥塞避免算法

快：网络中可能未出现阻塞但是已经有预兆，可做出预防措施。

接收者如果收到一个乱序的分组化，就必须立即发出前一个正确分组的应答，让发送者知道有一个分组可能丢失了。

此时直接执行：1.慢开始将门限减半 2 将发送窗口减半 3使用拥塞避免算法

**Http 工作过程？**

 1 ) 、地址解析，

     如用客户端浏览器请求这个页面：[http://localhost.com:8080/index.htm](http://localhost:8080/simple.htm" \t "_blank)

     从中分解出协议名、主机名、端口、对象路径等部分，对于我们的这个地址

2）、封装HTTP请求数据包

     把以上部分结合本机自己的信息，封装成一个HTTP请求数据包

3）封装成TCP包，建立TCP连接（TCP的三次握手）

       HTTP是比TCP更高层次的应用层协议，根据规则，只有低层协议建立之后才能，才能进行更层协议的连接，因此，首先要建立TCP连接，一般TCP连接的端口号是80。这里是8080端口

 4）客户机发送请求命令

       建立连接后，客户机发送一个请求给服务器，请求方式的格式为：统一资源标识符（URL）、协议版本号，后边是MIME信息包括请求修饰符、客户机信息和可内容。

     5）服务器响应

     服务器接到请求后，给予相应的响应信息，其格式为一个状态行，包括信息的协议版本号、一个成功或错误的代码，后边是MIME信息包括服务器信息、实体信息和可能的内容。

        实体消息是服务器向浏览器发送头信息后，它会发送一个空白行来表示头信息的发送到此为结束，接着，它就以Content-Type应答头信息所描述的格式发送用户所请求的实际数据

     6）服务器关闭TCP连接

     一般情况下，一旦Web服务器向浏览器发送了请求数据，它就要关闭TCP连接，然后如果浏览器或者服务器在其头信息加入了这行代码

HTTP请求报文由3部分组成（**请求行+请求头+请求体**）：

**请求行：请求方法 URL协议/版本**

**请求头：每个头域都由一个域名 ：值域组成**

**Connection ：表明是否要持久连接**

**Host： 制定请求资源的主机和端口号**

**Accept: 接收的编码方法和接收的语言。**

**Cookie头域：**

**Http响应报文（状态行，消息报文，响应正文）**

**状态行：协议版本、状态代码，状态描述**

200 – 请求成功

301 – 资源(网页等)被永久转移到其它URL

303 ：重定向，从原url重新定向到新的url

401：请求未经授权

400：客户端请求语法有错误

403：服务器拒绝提供服务

404 – 请求的资源(网页等)不存在

500 – 内部服务器错误

**http1.0和http1.1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。

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

3.HTTP/1.1加入了一个新的状态码100（Continue）。客户端事先发送一个只带头域的请求，如果服务器因为权限拒绝了请求，就回送响应码401（Unauthorized）；如果服务器接收此请求就回送响应码100，客户端就可以继续发送带实体的完整请求了。

Http的SSL？

引入了权威证书颁发机构（CA）来解决

1. 服务端先把自己的公钥发送证书颁发机构，向其申请证书
2. 证书颁发机构自己也有一个公钥和私钥。机构用私钥来加密key1.