**HTTP协议：**

**超文本传输协议**，是一种建立在TCP上的**无状态**连接，整个基本的工作流程是客户端发送一个HTTP请求，说明客户端想要访问的资源和请求的动作，服务端收到请求之后，服务端开始处理请求，并根据请求做出相应的动作访问服务器资源，最后通过发送HTTP响应把结果返回给客户端。端口80

1. HTTP请求

状态行

GET /562f25980001b1b106000338.jpg HTTP/1.1

请求头

Host img.mukewang.com

User-Agent Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.106 Safari/537.36

Accept image/webp,image/\*,\*/\*;q=0.8

Referer http://www.imooc.com/

Accept-Encoding gzip, deflate, sdch

Accept-Language zh-CN,zh;q=0.8

空行

主体

Key：value

**状态行：**包括请求方式**Method**、资源路径**URL**、协议版本**Version**；

**请求头：**包括一些访问的域名、用户代理、Cookie等信息、HOST、编码；

**空行：**请求头部后面的空行是必须的，即使第四部分的请求数据为空，也必须有空行。

**主体：**参数，请求的数据参数

1. HTTP响应

**状态行：版本（1.1），状态码（200），状态消息为（ok）**

HTTP/1.1 200 OK

**响应头：用来说明客户端要使用的一些附加信息**

Date: Fri, 22 May 2009 06:07:21 GMT

Content-Type: text/html; charset=UTF-8

**空行：头部后必须要有一行空行**

**响应数据主体：**

<html>

<head></head>

<body>

<!--body goes here-->

</body>

</html>

1. HTTP状态码

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  **200（成功）**  
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  **400（客户端语法错误）401（无权限）403（拒绝提供服务）404（资源不存在）405（请求method错误）**  
5xx：服务器端错误 **500（服务器出现错误）501（服务器当前不能处理请求，一段时间后可以）**

**HTTPS协议：**

HTTPS就是安全的HTTP，在http与传输层之间加上了一个SSL对称加密与非对称加密。HTTPS = HTTP+ 加密 + 认证 + 完整性保护。端口443

**Https的连接过程：**

http连接是明文传输，HTTPS选择了握手时交换密钥的方案。握手过程中，服务器会发出一张证书（带着公钥），客户端用公钥加密了一段较短的数据S，并返回给服务器。服务器用私钥解开，拿到S。此时，握手步骤完成，S成为了一个被安全传输到对方手中的对称加密密钥。此后，服务器与我的请求响应，只需要用S作为密钥进行一次对称的加密就好。

**OSI七层模型：**

**应用层：**提供应用层服务，文件传输(FTP)，电子邮件(SMTP)，还有HTTP（超文本传输协议），DNS，和telnet

**表示层：**用于数据格式化，代码转换，数据加密，没有协议

**会话层：**解除或建立与别的接点的联系，没有协议

**传输层：**TCP UDP

**网络层：** IP  ICMP（ping主要实现）， OSPF(全局泛洪，主要用于IP选路)

**数据链路层：**ARP（地址解析协议，根据IP地址获得MAC地址）

**物理层：**中继器、集线器、双绞线等

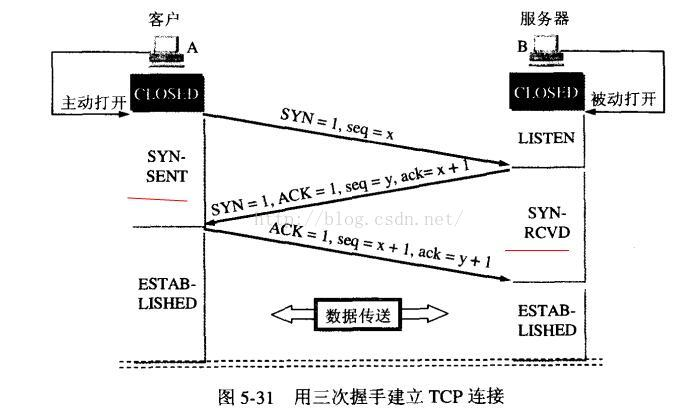
**TCP（传输控制协议）：**

是一种**面向连接的、可靠的、基于字节流**的传输层通信协议，在**传输层**，实际数据传输之前服务器和客户端要进行**三次握手，四次挥手**会话结束后结束连接，是1对1连接的

与UDP区别

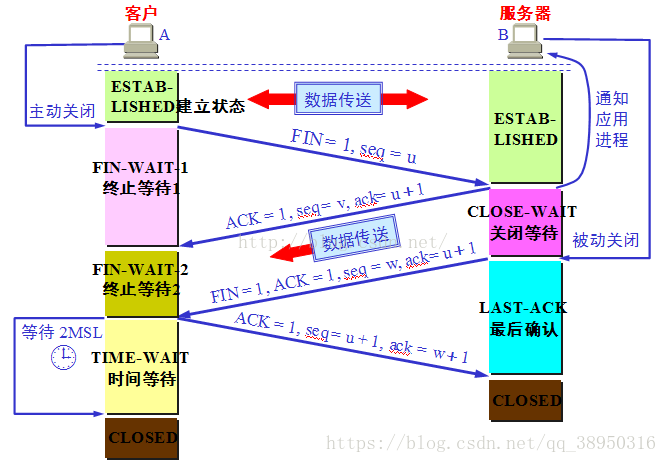
UDP是**无连接**的，没有超时重发机制，**传输速度快**，主要用于视频传输，**可以1对多广播**，**尽可能保证送达**

**TCP三次握手：**



1、假设A是客户端，B是服务端。A首先向B发出连接请求报文段，这个时候首部中的同步位SYN=1，同时选择一个初始的序号x。此时报文段不能携带数据。此时A进入到SYN\_SENT(同步已发送)状态。  
      2、B受到连接请求报文，同意建立连接，向A发出确认。确认报文中，SYN和ACK都置1，确认号是x+1,与此同时，自己选择一个初始序号y,这个报文也不能携带数据。此时B进入SYN\_RCVD（同步收到）状态。  
      3、A收到B的确认后，还要给B确认。这时可以携带数据，A进入到ESTABLISHED状态。这就是三次握手的过程。

**TCP四次挥手：**



1）客户端进程发出连接释放报文，并且停止发送数据。释放数据报文首部，FIN=1，其序列号为seq=u（等于前面已经传送过来的数据的最后一个字节的序号加1），此时，客户端进入FIN-WAIT-1（终止等待1）状态。 TCP规定，FIN报文段即使不携带数据，也要消耗一个序号。

2）服务器收到连接释放报文，发出确认报文，ACK=1，ack=u+1，并且带上自己的序列号seq=v，此时，服务端就进入了CLOSE-WAIT（关闭等待）状态。TCP服务器通知高层的应用进程，客户端向服务器的方向就释放了，这时候处于半关闭状态，即客户端已经没有数据要发送了，但是服务器若发送数据，客户端依然要接受。这个状态还要持续一段时间，也就是整个CLOSE-WAIT状态持续的时间。

3）客户端收到服务器的确认请求后，此时，客户端就进入FIN-WAIT-2（终止等待2）状态，等待服务器发送连接释放报文（在这之前还需要接受服务器发送的最后的数据）。

4）服务器将最后的数据发送完毕后，就向客户端发送连接释放报文，FIN=1，ack=u+1，由于在半关闭状态，服务器很可能又发送了一些数据，假定此时的序列号为seq=w，此时，服务器就进入了LAST-ACK（最后确认）状态，等待客户端的确认。

5）客户端收到服务器的连接释放报文后，必须发出确认，ACK=1，ack=w+1，而自己的序列号是seq=u+1，此时，客户端就进入了TIME-WAIT（时间等待）状态。注意此时TCP连接还没有释放，必须经过2∗∗MSL（最长报文段寿命）的时间后，当客户端撤销相应的TCB后，才进入CLOSED状态。

6）服务器只要收到了客户端发出的确认，立即进入CLOSED状态。同样，撤销TCB后，就结束了这次的TCP连接。可以看到，服务器结束TCP连接的时间要比客户端早一些。