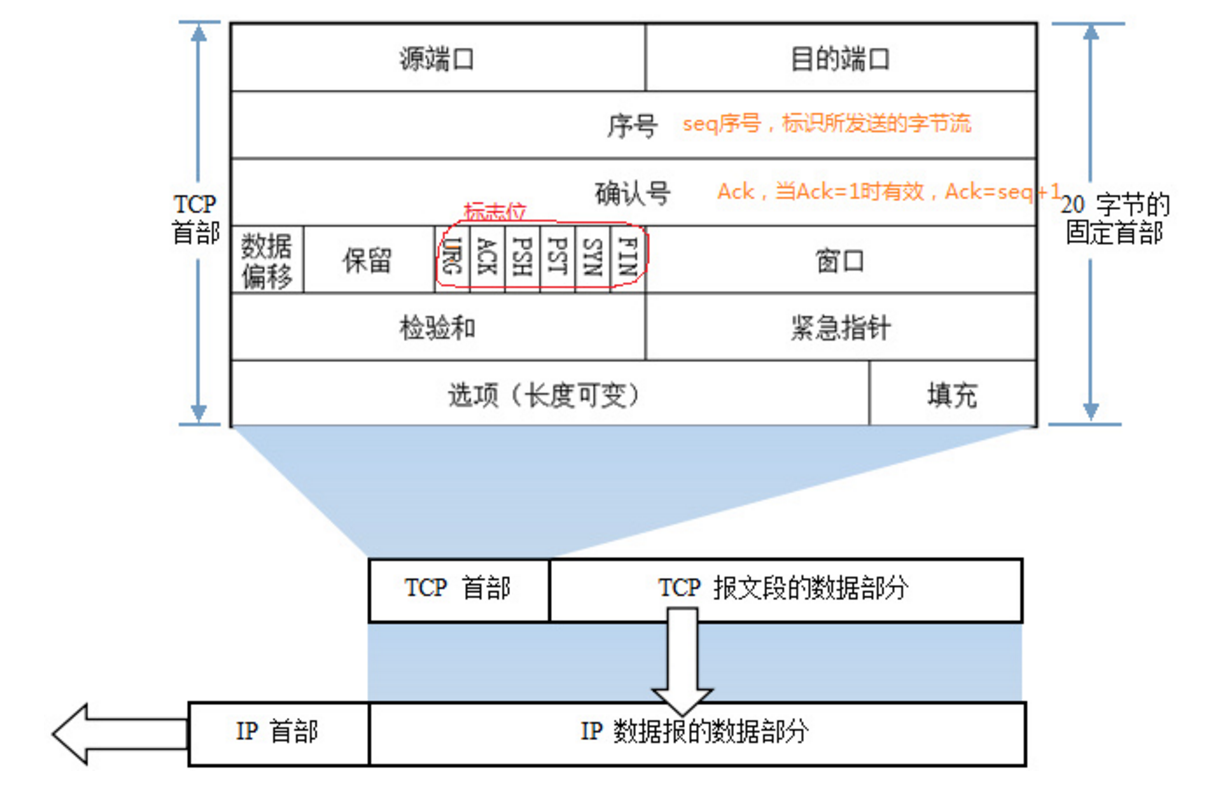
三次握手流程

1. 客户端发个请求“开门呐，我要进来”给服务器
2. 服务器发个“进来吧，我去给你开门”给客户端
3. 客户端有很客气的发个“谢谢，我要进来了”给服务器

四次挥手流程

1. 客户端发个“时间不早了，我要走了”给服务器，等服务器起身送他
2. 服务器听到了，发个“我知道了，那我送你出门吧”给客户端，等客户端走
3. 服务器把门关上后，发个“我关门了”给客户端，然后等客户端走（尼玛~矫情啊）
4. 客户端发个“我知道了，我走了”，之后自己就走了

**TCP报文格式**



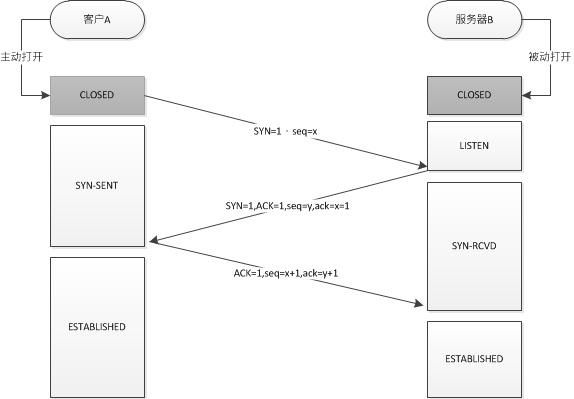
上图中有几个字段需要重点介绍下：  
  （1）序号：Seq序号，占32位，用来标识从TCP源端向目的端发送的字节流，发起方发送数据时对此进行标记。  
  （2）确认序号：Ack序号，占32位，**只有ACK标志位为1时，确认序号字段才有效，Ack=Seq+1**。  
  （3）标志位：共6个，即URG、ACK、PSH、RST、SYN、FIN等，具体含义如下：  
  （A）URG：紧急指针（urgent pointer）有效。

  （B）ACK：确认序号有效。  
  （C）PSH：接收方应该尽快将这个报文交给应用层。  
  （D）RST：重置连接。  
  （E）SYN：发起一个新连接。  
  （F）FIN：释放一个连接。

具体流程图如下：

**三次握手的过程（客户端我们用A表示，服务器端用B表示）**

前提：A主动打开，B被动打开

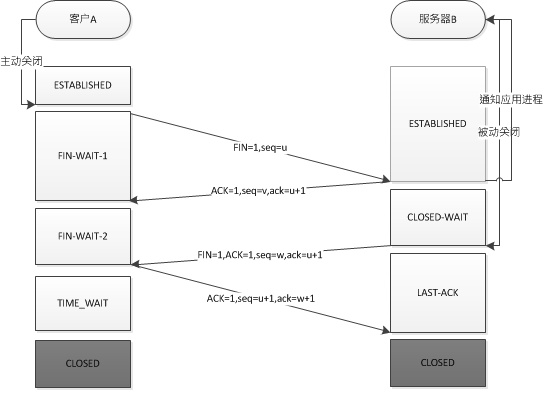


TCP三次握手   
（1）第一次握手：Client将标志位**SYN置为1**，随机产生一个**值seq=J**，并将该数据包发送给Server，Client进入**SYN\_SENT状态**，等待Server确认。   
（2）第二次握手：Server收到数据包后由标志位SYN=1知道Client请求建立连接，Server将标志位**SYN和ACK都置为1，ack=J+1，**随机产生一个值**seq=K，**并将该数据包发送给Client以确认连接请求，Server进入**SYN\_RCVD**状态。   
（3）第三次握手：Client收到确认后，检查**ack是否为J+1，ACK是否为1**，如果正确则将标志位**ACK置为1，ack=K+1**，并将该数据包发送给Server，Server检查**ack是否为K+1，ACK是否为1**，如果正确则连接建立成功，Client和Server进入**ESTABLISHED**状态，完成三次握手，随后Client与Server之间可以开始传输数据了。

SYN攻击：   
在三次握手过程中，Server发送SYN-ACK之后，收到Client的ACK之前的TCP连接称为半连接（half-open connect），此时Server处于SYN\_RCVD状态，当收到ACK后，Server转入ESTABLISHED状态。SYN攻击就是Client在短时间内伪造大量不存在的IP地址，并向**Server不断地发送SYN包**，Server回复确认包，并等待Client的确认，由于源地址是不存在的，因此，Server需要不断重发直至超时，这些伪造的SYN包将产时间占用未连接队列，导致正常的SYN请求因为队列满而被丢弃，从而引起网络堵塞甚至系统瘫痪。SYN攻击时一种典型的DDOS攻击，检测SYN攻击的方式非常简单，即当Server上有**大量半连接状态且源IP地址是随机的**，则可以断定遭到SYN攻击了，使用如下命令可以让之现行：   
#netstat -nap | grep SYN\_RECV

**四次分手的过程（客户端我们用A表示，服务器端用B表示）**

前提：A主动关闭，B被动关闭

 TCP四次挥手   
由于TCP连接时全双工的，因此，每个方向都必须要单独进行关闭，这一原则是当一方完成数据发送任务后，发送一个FIN来终止这一方向的连接，收到一个FIN只是意味着这一方向上没有数据流动了，即不会再收到数据了，但是在这个TCP连接上仍然能够发送数据，直到这一方向也发送了FIN。首先进行关闭的一方将执行主动关闭，而另一方则执行被动关闭，上图描述的即是如此。   
（1）第一次挥手：Client发送一个FIN=1，用来关闭Client到Server的数据传送，随机产生一个**值seq=u,**Client进入**FIN\_WAIT\_1**状态。   
（2）第二次挥手：Server收到FIN后，发送一个ACK=1给Client，确认序号为收到序号ack=u+1（与SYN相同，一个FIN占用一个序号），随机产生一个**值seq=v,**Server进入**CLOSE\_WAIT**状态。   
（3）第三次挥手：Server发送一个**FIN=1**，用来关闭Server到Client的数据传送，**ACK=1,seq=w,ack=u+1;**Server进入**LAST\_ACK**状态。   
（4）第四次挥手：Client收到FIN后，Client进入**TIME\_WAIT**状态，接着发送一个ACK=1给Server，确认序号为收到序号+1**,ack=w+1，seq=u+1**;Server进入**CLOSED**状态，完成四次挥手。   
上面是一方主动关闭，另一方被动关闭的情况，实际中还会出现同时发起主动关闭的情况

**（2）为什么建立连接是三次握手，而关闭连接却是四次挥手呢？**这是因为服务端在LISTEN状态下，收到建立连接请求的SYN报文后，把ACK和SYN放在一个报文里发送给客户端。而关闭连接时，当收到对方的FIN报文时，仅仅表示对方不再发送数据了但是还能接收数据，己方也未必全部数据都发送给对方了，所以己方可以立即close，也可以发送一些数据(ACK)给对方后，再发送**FIN报文**给对方来表示同意现在关闭连接，因此，**己方ACK和FIN一般都会分开发送**。