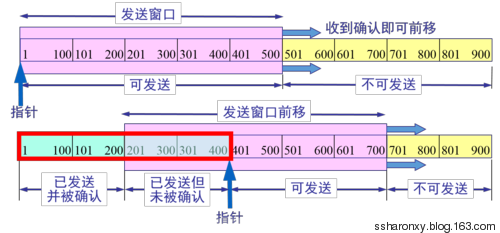
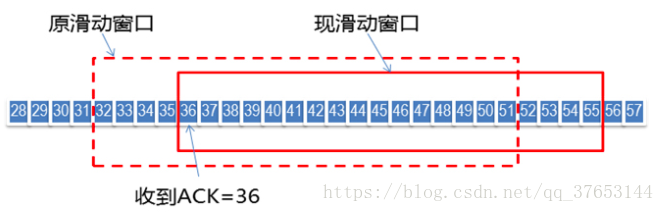
1. 浏览器输入url到显示页面的过程？
2. DNS解析
3. 建立TCP连接
4. 发送http请求
5. 服务器处理请求并返回http报文
6. 浏览器解析渲染页面
7. 连接结束
8. 滑动窗口和流量控制



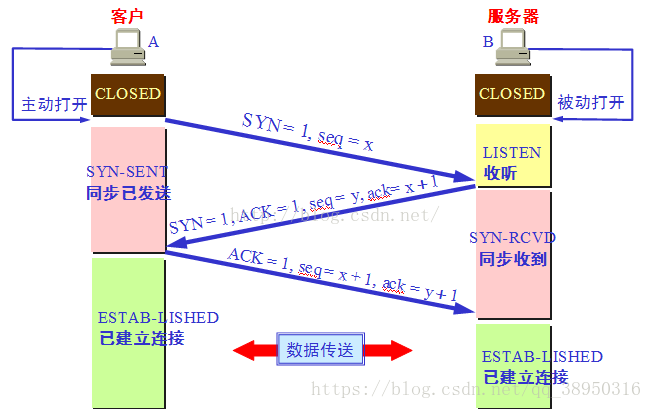


如上图所示， 随着发送端接收到返回的数据ACK，滑动窗口也随之右移。发送端根据接收端返回的ACK可以得到两个重要的信息：一是接收端期望收到的下一个字节序号；二是当前的窗口大小（再结合发送端已有的其他信息可以得出还能发送多少字节数据）。需要注意的是：发送窗口的左边界只能右移，因为它控制的是已发送并受到确认的数据，具有累积性，不能返回；右边界可以右移也可以左移。

TCP 利用滑动窗口实现流量控制。流量控制是为了控制发送方发送速率，保证接收方来得及接 收。 接收方发送的确认报文中的窗口字段可以用来控制发送方窗口大小，从而影响发送方的发 送速率。将窗口字段设置为 0，则发送方不能发送数据。

1. TCP的三次握手与四次挥手

三次握手：

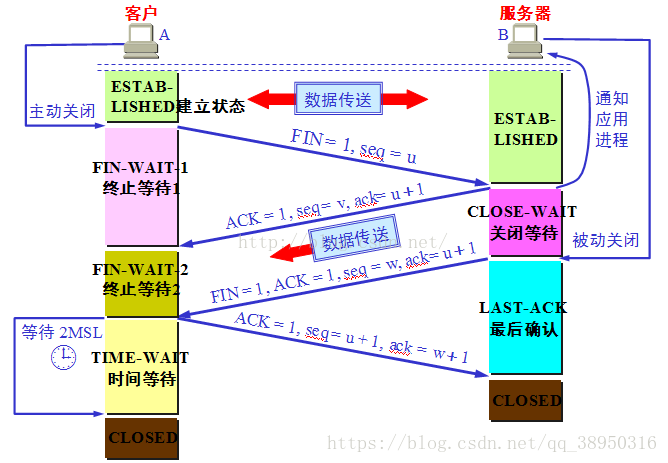


第一次握手：建立连接时，客户端发送syn包（syn=x）到服务器，并进入SYN\_SENT状态，等待服务器确认；SYN：同步序列编号（Synchronize Sequence Numbers）。

第二次握手：服务器收到syn包，必须确认客户的SYN（ack=x+1），同时自己也发送一个SYN包（syn=y），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=y+1），此包发送完毕，客户端和服务器进入ESTABLISHED（TCP连接成功）状态，完成三次握手。

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。

1. IP地址分类

A类地址： 1.0.0.0~126.0.0.0

B类地址：128.0.0.0 ~ 191.255.255.255

C类地址：192.0.0.0 ~ 223.255.255.255

D类地址：224.0.0.0 ~ 239.255.255.255 （作为多播使用）

保留（私有）地址：A类：10.0.0.0 – 10.255.255.255，B类：172.16.0.0 – 172.31.255.255，C类：192.168.0.0 – 192.168.255.255

1. TCP/IP四层模型和OSI七层模型

