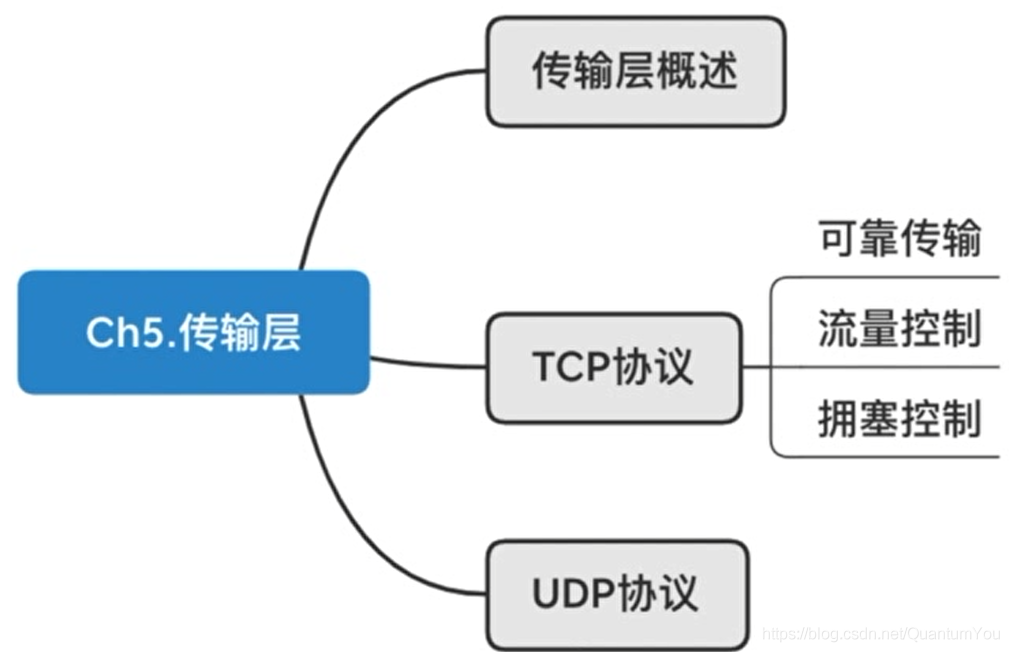
业精于勤，荒于嬉  
行成于思，毁于随

@[toc]  


# 传输层

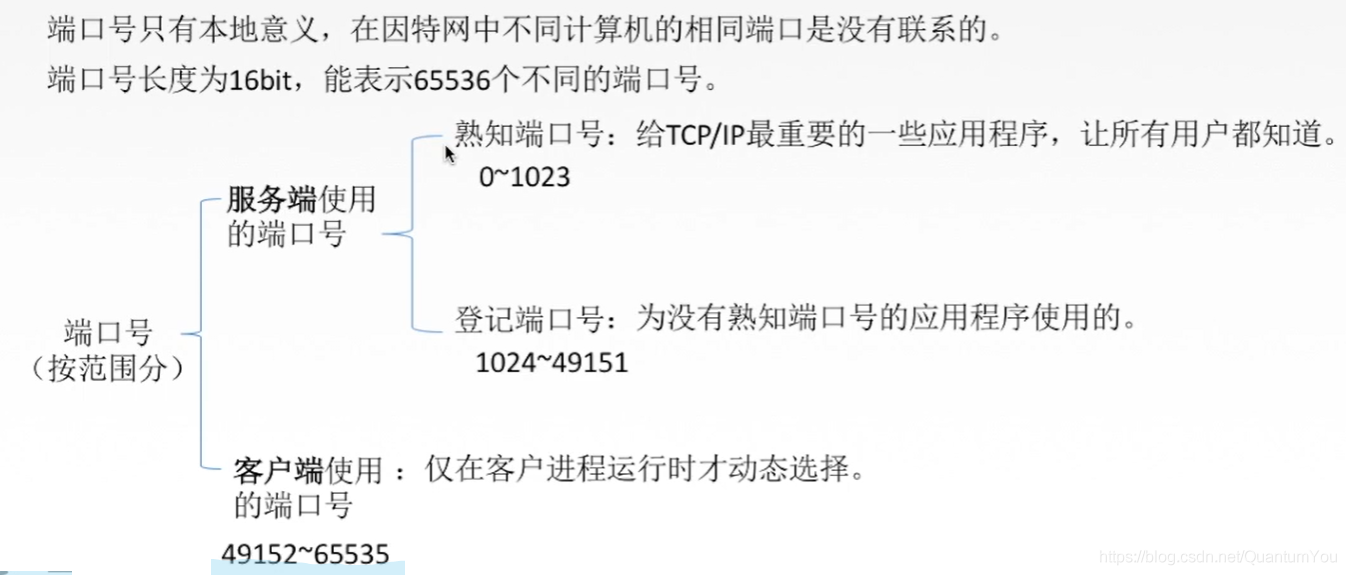
* **传输层概述**：只有主机才拥有的层次，为应用层提供通信服务 ，使用网络层的服务
* **传输层的功能**：  
  1、传输层提供进程和进程之间的逻辑通信。  
  2、复用和分用  
  3、传输层对收到的报文进行差错检测。  
  4、传输层的两者协议

## 传输层的两种协议



## 传输层的寻址与端口

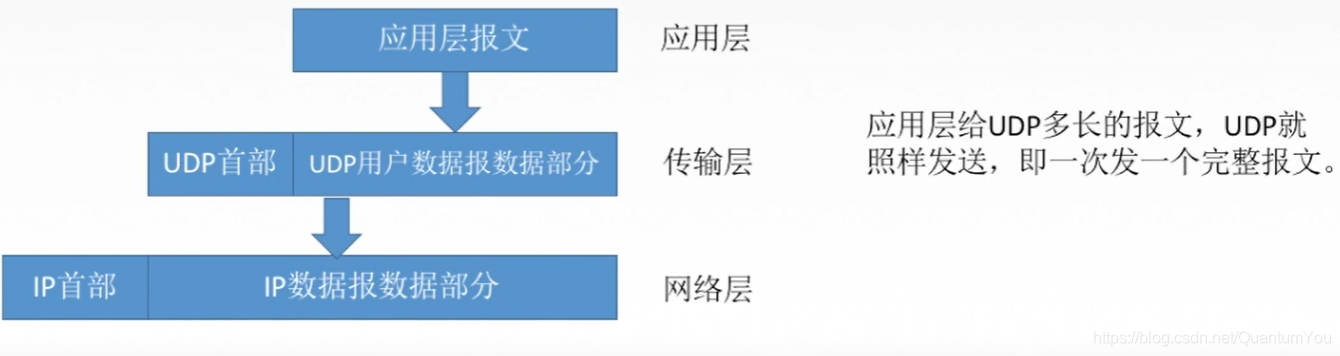
* 复用：应用层所有的应用进程都可以通过传输层再传输到网络层。
* 分用：传输层从网络层收到数据后交付指明的应用进程。
* 逻辑端口/软件端口 **端口**是传输层的SAP,标识主机中的应用进程。

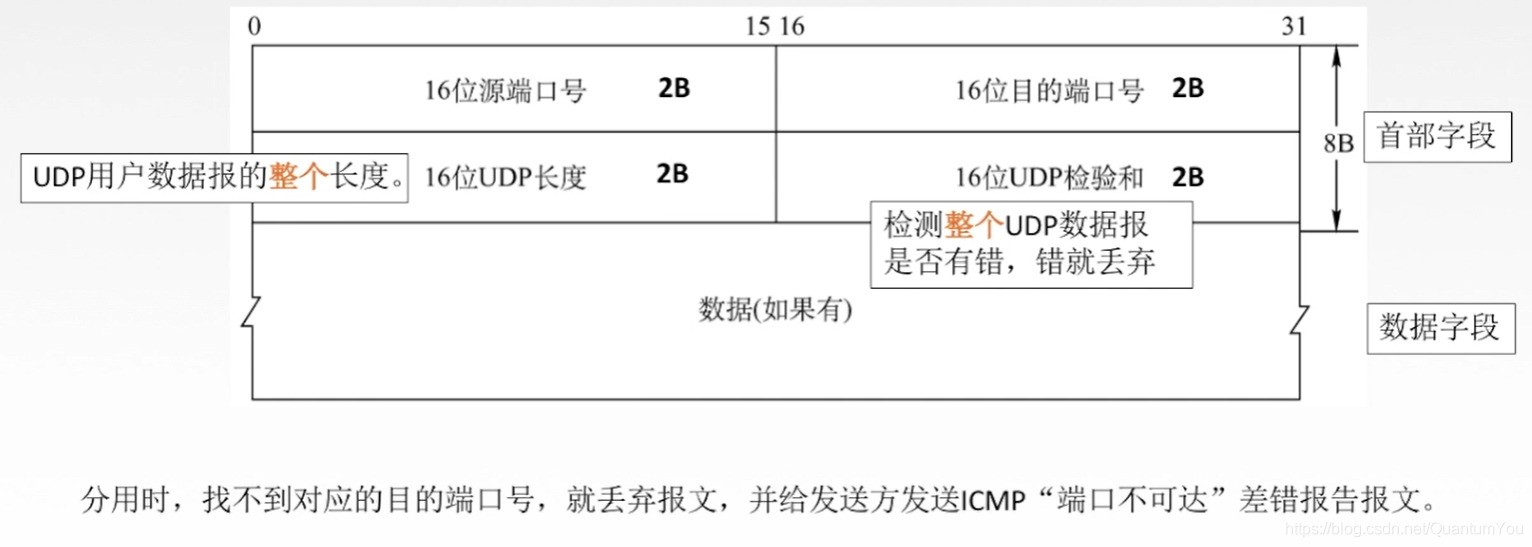

# UDP 协议

## 用户数据报协议UDP概述

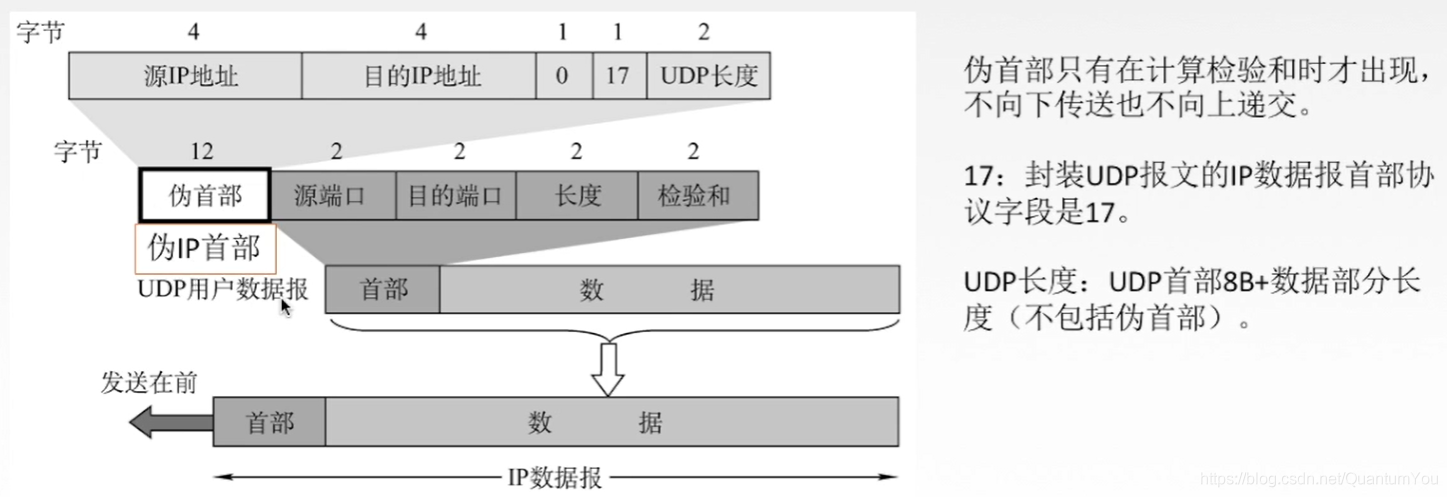
* UDP只在IP数据报服务之上增加了很少功能，即复用分用和差错检测功能。

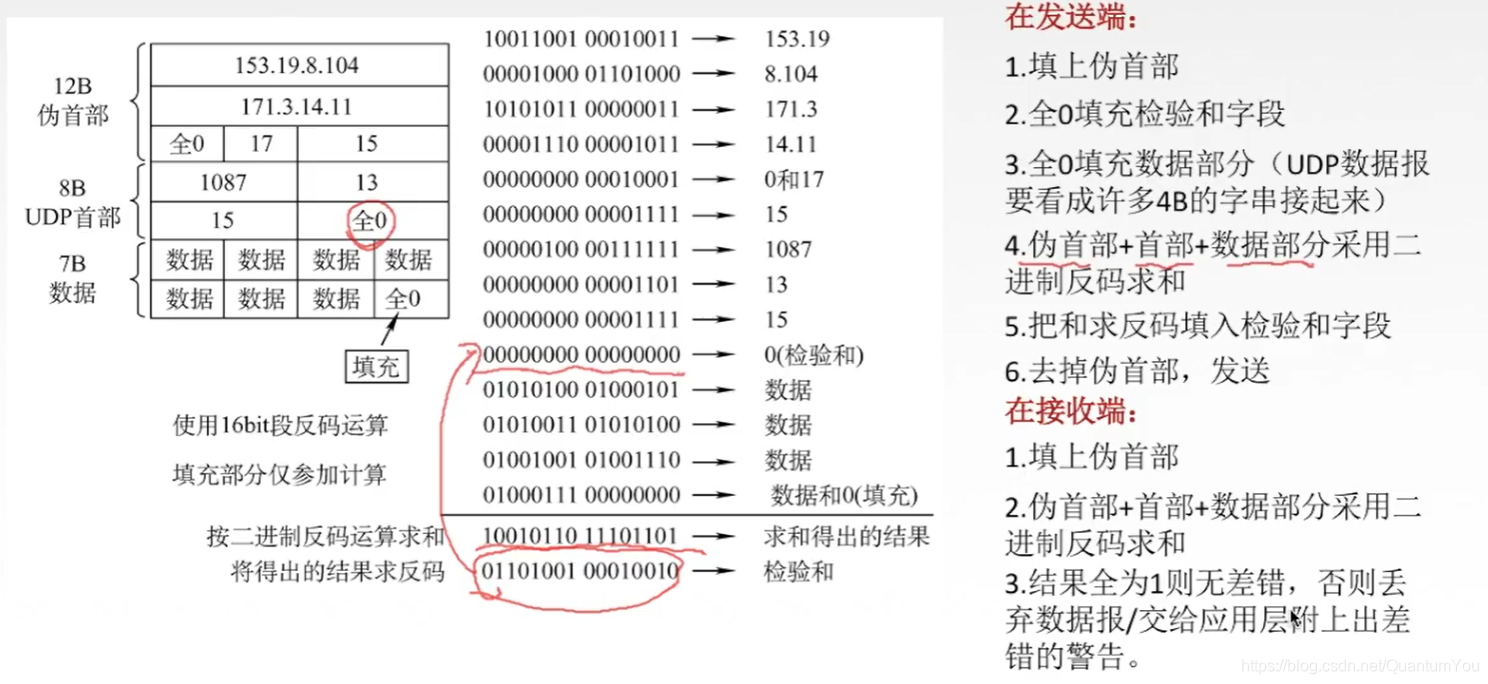
**UDP 的主要特点**：  
1、UDP是**无连接**的，减少开销和发送数据之前的时延。  
2、UDP使用最大努力交付，即**不保证可靠交付**。  
3、UDP是**面向报文**的，适合一次性传输少量数据的网络应用。  
4、UDP**无拥塞控制**，适合很多实时应用。  
5、UDP首部开销小，8B,TCP20B。  


## UDP 首部格式



## UDP 校验

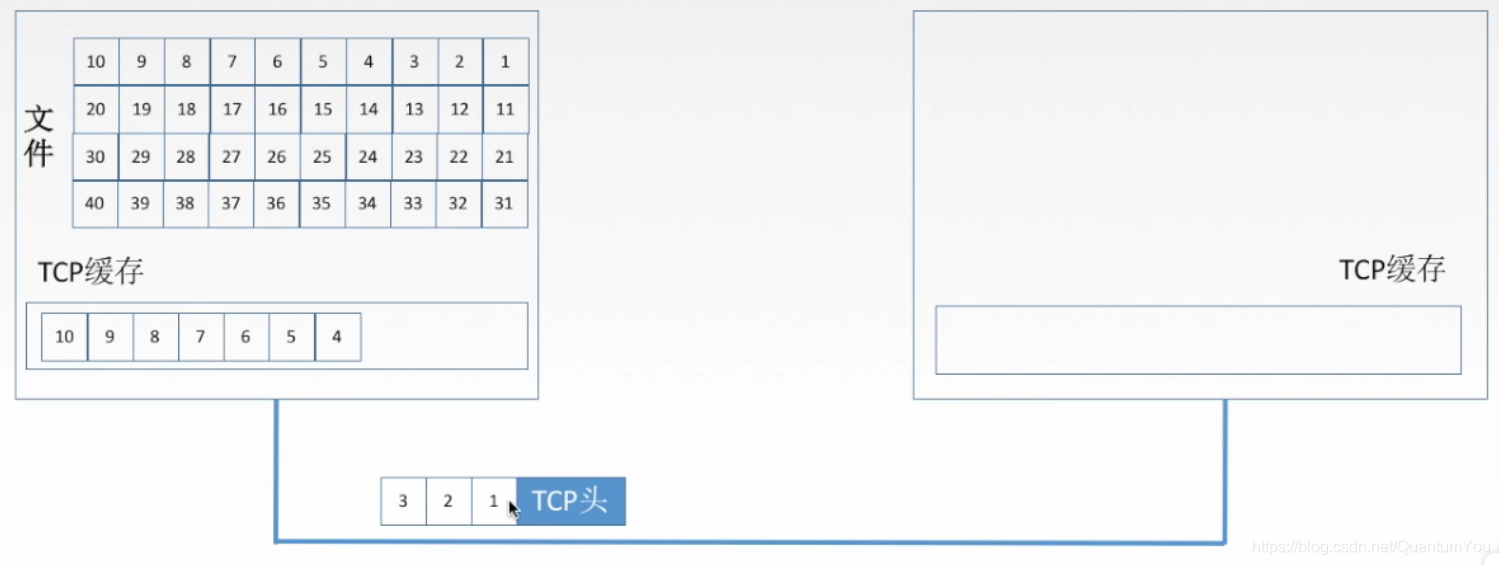




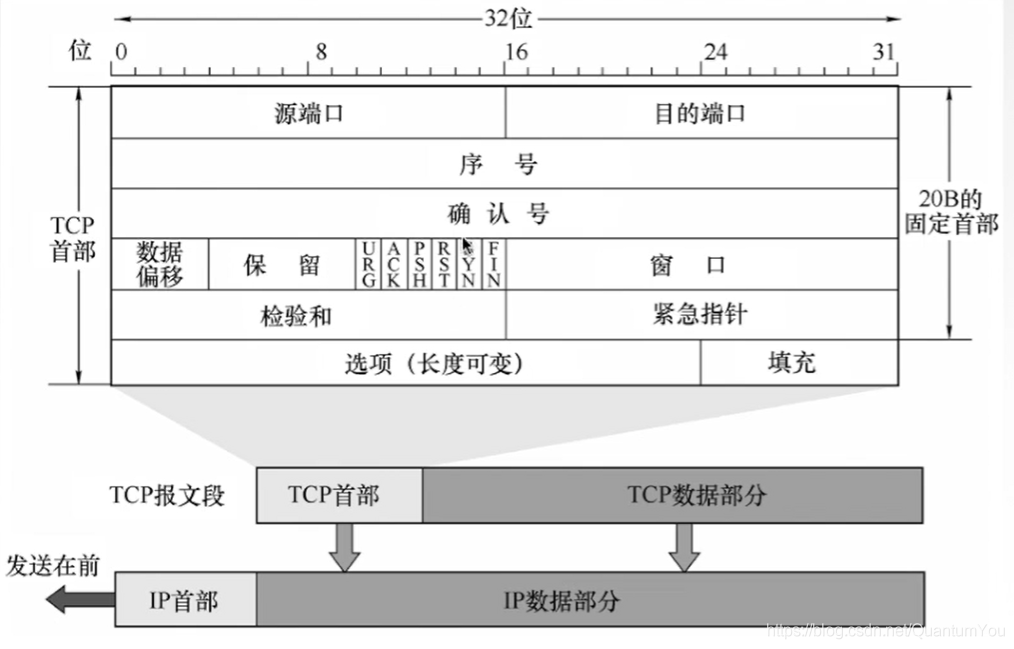
# TCP 协议

## TCP 协议特点

* 1、TCP是面向连接（虚连接）的传输层协议。
* 2、每一条TCP连接只能有两个端点，每一条TCP连接只能是点对点的。
* 3、TCP提供可靠交付的服务，无差错、不丢失、不重复、按序到达。 *可靠有序，不丢不重*
* 4、TCP提供全双工通信。  
  发送缓存：准备发送的数据&已发送但尚未收到确认的数据  
  接收缓存：按序到达但尚未被接受应用程序读取的数据&不按序到达的数据
* 5、TCP面向字节流：TCP把应用程序交下来的数据看成仅仅是一连串的无结构的字节流。
* 流：流入到进程或从进程流出的字节序列。



## TCP报文段首部格式 （重要）



* **序号**：在一个TCP连接中传送的字节流中的每一个字节都按顺序编号，本字段表示本报文段所发送数据的第一个字节的序号。
* **确认号**：期望收到对方下一个报文段的第一个数据字节的序号。若确认号为N,则证明到序号N-1为止的所有数据都己正确收到。
* **数据偏移**（首部长度）：TCP报文段的数据起始处距离TCP报文段的起始处有多远，以4B位单位，即1个数值是4B。

**6个控制位**

* **紧急位URG** : URG=1时，标明此报文段中有紧急数据，是高优先级的数据，应尽快传送，不用在缓存里排队，配合紧急指针字段使用。
* **确认位ACK** :ACK=1时确认号有效，在连接建立后所有传送的报文段都必须把ACK置为1.
* **推送位PSH** : PSH=1时，接收方尽快交付接收应用进程，不再等到缓存填满再向上交付。
* **复位RST** : RST=1时，表明TCP连接中出现严重差错，必须释放连接，然后再重新建立传输链接。
* **同步位SYN**: SYN=1时，表明是一个连接请求/连接接受报文。
* **终止位FIN** : FIN=1时，表明此报文段发送方数据已发完，要求释放连接。

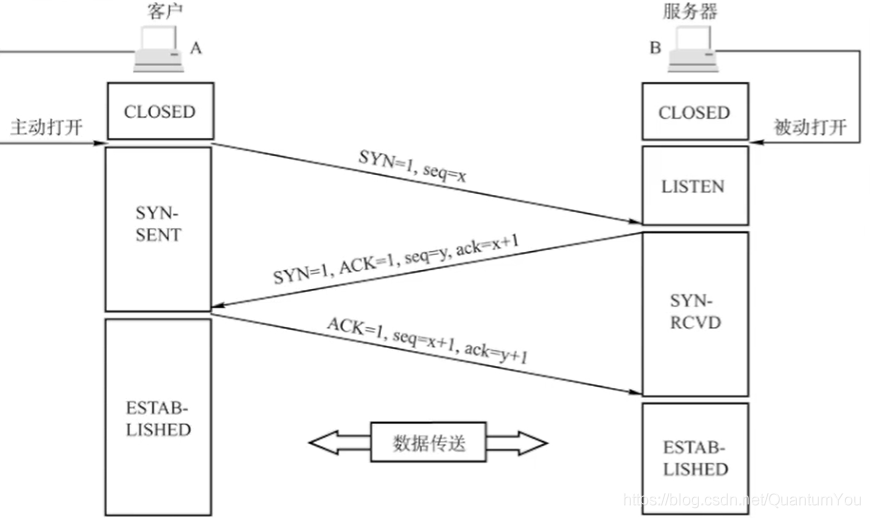
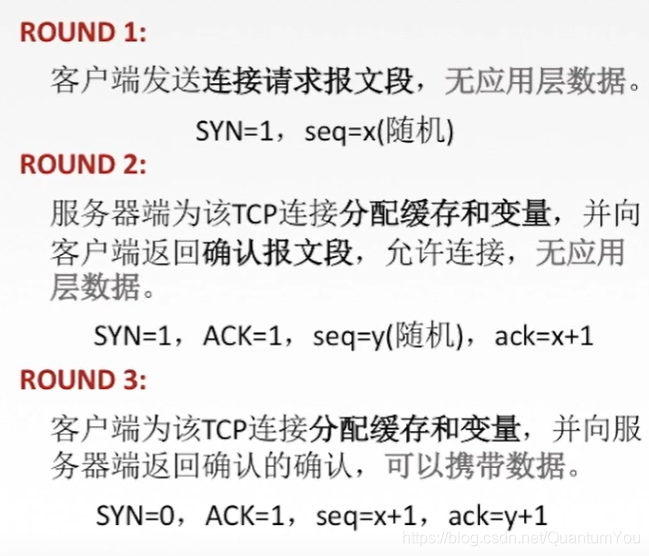
**窗口**：指的是发送本报文段的一方的接收窗口，即现在允许对方发送的数据量。  
**检验和**：检验首部+数据，检验时要加上12B伪首部，第四个字段为6。  
**紧急指针**：URG=1时才有意义，指出本报文段中紧急数据的字节数。  
**选项**：最大报文段长度MSS、窗口扩大、时间戳、选择确认。

## TCP 连接



* TCP连接的建立采用客户服务器方式，主动发起连接建立的应用进程叫做客户，而被动等待连接建立的应用进程叫服务器。

**具体连接建立过程如下**：  
假设运行在一台主机（客户）上的一个进程想与另一台主机（服务器）上的一个进程建立一条连接，客户应用进程首先通知客户TCP,他想建立一个与服务器上某个进程之间的连接，客户中的TcP会用以下步骤与服务器中的TCP建立一条TCP连接

  
三大阶段如下：  


## SYN 泛洪攻击

* SYN洪泛攻击发生在OSl第四层，这种方式利用τCP协议的特性，就是三次握手。攻击者发送TCP SYN，SYN是TCP三次握手中的第一个数据包，而当服务器返回ACK后，该攻击者就不对其进行再确认，那这个τCP连接就处于挂起状态，也就是所谓的半连接状态，服务器收不到再确认的话，还会重复发送ACK给攻击者。这样更加会浪费服务器的资源。攻击者就对服务器发送非常大量的这种TCP连接，由于每一个都没法完成三次握手，所以在服务器上，这些TCP连接会因为挂起状态而消耗CPU和内存，最后服务器可能死机，就无法为正常用户提供服务了。
* 解决方法 ： 设置SYN Cookie

## TCP 连接释放

* 参与一条TCP连接的两个进程中的任何一个都能终止该连接，连接结束后，主机中的“资源”（缓存和变量）将被释放。

