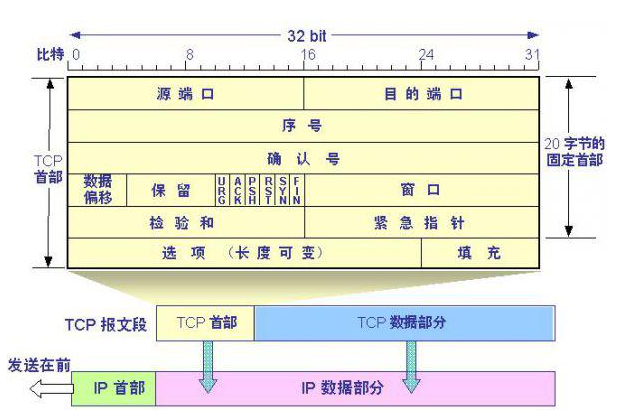
## Tcp协议

Tcp协议是一种可靠的，基于字节流的传输控制协议，它完成第四层，传输层所做的功能，用户数据包协议UDP 是同一层中另一个重要的协议；它位于IP层之上，应用层之下：应用层向tcp层发送8位字节表示的数据流，然后tcp会把数据流进行分段，分成适当长度的报文段，传送到IP层，最后传送到接收方的端口上；

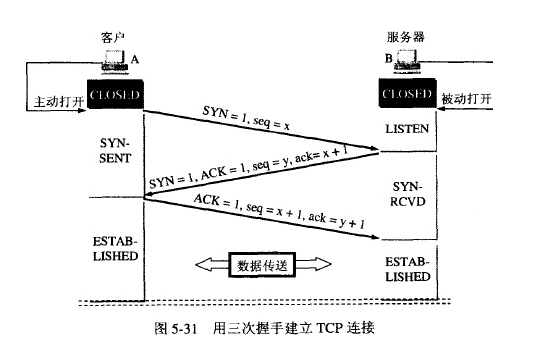
Tcp在传输过程中，为了保证不丢包，会在每个包上编一个序号，保证接受方也是按照顺序接收，然后接受方会对成功接收到的包返回一个确认（ack）；如果发送端在往返时间延迟内没收到确认包，会认为包被丢了而进行重新传送；tcp用一个“校验和”来检验数据是否有错误，在发送和接受时都会进行校验；

Tcp数据包格式图：



* 源端口：16位
* 目的端口：16位
* 序号：发送数据包的第一个字节的序列号，32位
* 确认号：确认序列号，32位
* 数据偏移：4位，值为tcp数据长度/4
* Urg： 1位，0或者1，表示紧急指针有效
* Ack：1位，表示确认号字段有意义，一旦连接建立，该标志总为1
* Psh： 1位，表示push功能，尽快将报文段提交到应用层
* Rst： 1位，重置tcp连接
* Syn： 1位，表示syn报文（在建立tcp连接时有用，当syn=1，ack=0时，这是一个连接请求报文段；对方同意连接，则返回syn=1,ack=1）
* Fin： 1位，表示没有数据需要发送了（在关闭tcp连接时有用）
* 校验和：16位

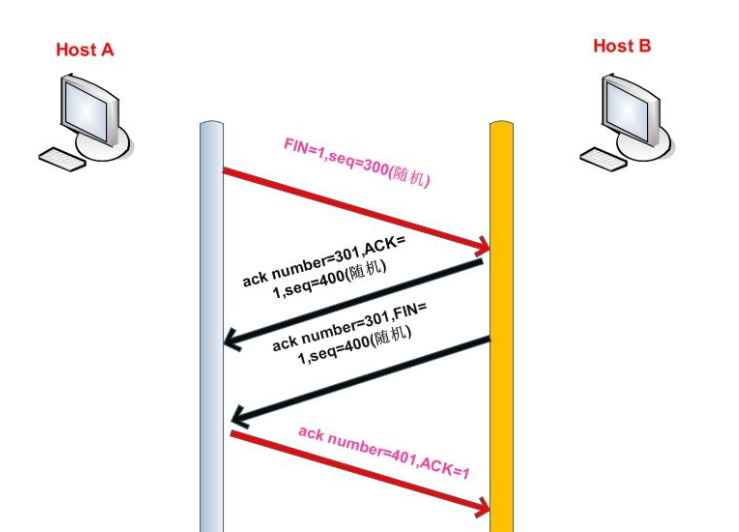
## 三次握手



* 第一步：客户端发送SYN到服务器，序列号为x
* 第二步: 服务器接收到syn报文，回送一个SYN（序列号为y）ACK（ack=x+1）报文
* 第三部：客户端接受到服务器的SYN报文，回送一个ACK报文（ack=y+1,序列号为x+1）
* 连接建立，开始传输数据

## 四次断开

由于tcp连接存在“半关闭状态”，所以断开连接需要四次



* 第一步：主机A会主动发出断开连接请求，执行“主动关闭”，主机A的tcp就会发送一个FIN数据包(序列号为x)，表示数据发送完毕；
* 第二步：主机B收到FIN数据包，会执行“被动关闭”，关闭该端口的数据连接，并返回一个ACK确认包（ack=x+1,序列号为y）
* 第三步：服务器会关闭该连接的进程，关闭该端口，这会触发tcp发出一个FIN数据包（ack=x+1）
* 第四步：客户端接受到这个FIN数据包后，确认了这个关闭数据包，发送一个确认消息ack数据包（ack=y+1），关闭连接

在某些特殊情况下，第一步的FIN会随着数据一起发送，第二步和第三步有时会一起发送，被合并为一个分节；

“半关闭状态”是指在第二步和第三步之间，从主机B到主机A传递数据是可能的，这称为“半关闭”；

任何一端都可以执行主动关闭，不仅仅是客户端；比如，在http/1.0中由服务器主动关闭；

Tcp连接仅仅适用于两端通信，广播和多播通信不适用tcp连接；