TCP连接的特点：

面向连接的，可套的字节流服务。

为应用层提供服务，不用考虑数据的内容。

TCP保证可靠性：

分块传输：数据被分割成最合适的块数据（UDP的数据报长度不变）

等待确认：通过定时器等待接收端发送确认回复，收不到确认则重发

确认回复：收到确认后发送确认回复

数据校验：保持首部和数据的校验和，检测数据传输过程有无变化

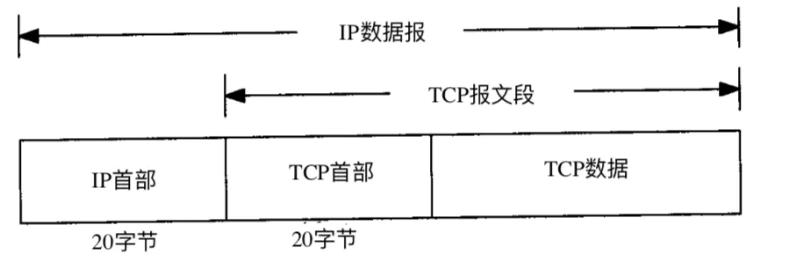
乱序重排：接收端能重排数据，以正确的数据发送给应用

重复丢弃：接收端能丢弃重复的数据包

流量缓冲：两段有固定大小的缓冲区（滑动窗口），防止速度不匹配丢数据

TCP报头

宏观



TCP数据被封在IP数据报中

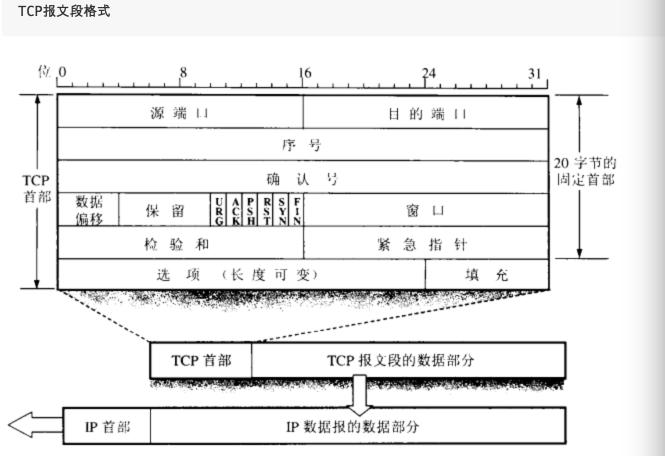
TCP：传输控制协议

面向连接的可靠传输协议，在完成了传输层基本工作的基础上还需要保证传输的可靠性

面向链接：传输数据前先通过三次我收建立端到端的虚拟链路

可靠传输：4种可靠传输机制：排序、确认、重传、流控（滑动窗口控制）

TCP报文头详解



1. 源端口和目的端口：各占两个字节，一般网络通信通过IP+端口唯一确定一个TCP连接。
2. 占四个字节，用来表示从发送端向接收端发送的数据字节流（seq）
3. 确认序号：占四个字节，包含发送确认的一段所期望的下一个序号，即ack=seq序号+1
4. 数据偏移：用来存储报文头的大小，默认为20个字节，最大为60个字节。
5. 保留字段：六位，可以忽略
6. 标志位：

URG（紧急）：为1时表示紧急指针有效

ACK（确认）：为1时表示确认序号字段有效

PSH（推送）：为1时表示尽快把报文推个应用层

RST（复位）：为1时表示链接异常，需要重新简历TCP连接

SYN（同步）：为1时表示在建立连接时同步序号

FIN（终止）：为1时表示发送数据完毕则断开连接

1. 接收窗口：占两个字节，用于流量控制和阻塞控制，表示当前接收缓冲区的大小，动态控制缓冲区大小。在计算机网络中，通常是用接收方的大小来控制发送方的数据发送量。TCP链接的一端根据缓冲区大小确定自己的接收窗口值，告诉对方，是对方可以确定发送数据的字节数。
2. 校验和：占2个字节，范围包括首部和数据两部分。是可以选的，默认不选。

MSS最长报文大小

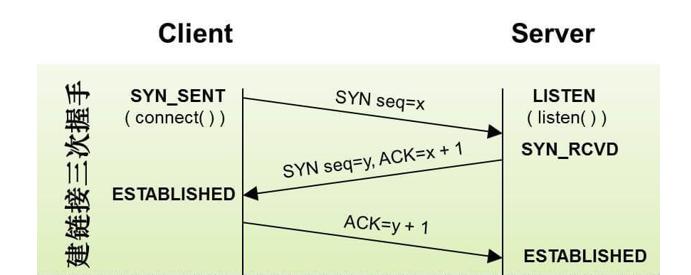
最常见的可选字段

MSS只能出现在SYN时传过来（第一次握手和第二次握手）

指明端口能接受的最大长度的报文段

建立连接时，双方都要发送MSS

三次握手流程



1、客户端连接服务端，发送32位\*\*\*syn=x，并且SYN置1。  
2、服务端监听客户端发送的\*\*\*，然后发送32位确认响应syn=y和ack=x+1，并将SYN、ACK置1。  
3、然后客户端接收以后，说明服务端接收成功，然后再返回ack=y+1，表示客户端已经接收成功可以开始连接  
详细过程：  
• 第一次握手：建立连接时，客户端发送syn包（syn=j）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=j+1），同时自己也发送一个SYN包（syn=k），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=k+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;

未连接队列

在三次握手协议中，服务器维护一个未连接队列，该队列为每个客户端的SYN包（syn=j）开设一个条目，该条目表明服务器已收到SYN包，并向客户发出确认，正在等待客户的确认包。这些条目所标识的连接在服务器处于SYN\_RECV状态，当服务器收到客户的确认包时，删除该条目，服务器进入ESTABLISHED状态。

TIME\_WAIT状态

TIME\_WAIT状态存在有两个原因。  
<1>可靠终止TCP连接。如果最后一个ACK报文因为网络原因被丢弃，此时server因为没有收到ACK而超时重传FIN报文，处于TIME\_WAIT状态的client可以继续对FIN报文做回复，向server发送ACK报文。  
<2>保证让迟来的TCP报文段有足够的时间被识别和丢弃。连接结束了，网络中的延迟报文也应该被丢弃掉，以免影响立刻建立的新连接。

**面试题**

问：为什么需要三次？

答：TCP是可靠的传输控制协议，三次握手能保证数据可靠传输又能提高传输效率。  
tcp连接是全双工的，数据在两个方向上能同时传递。  
所以要确保双方，同时能发数据和收数据  
第一次握手：证明了发送方能发数据  
第二次握手：ack确保了接收方能收数据，syn确保了接收方能发数据  
第三次握手：确保了发送方能收数据  
实际上是四个维度的信息交换，不过中间两步合并为一次握手了。  
四次握手浪费，两次握手不能保证“双方同时具备收发功能”

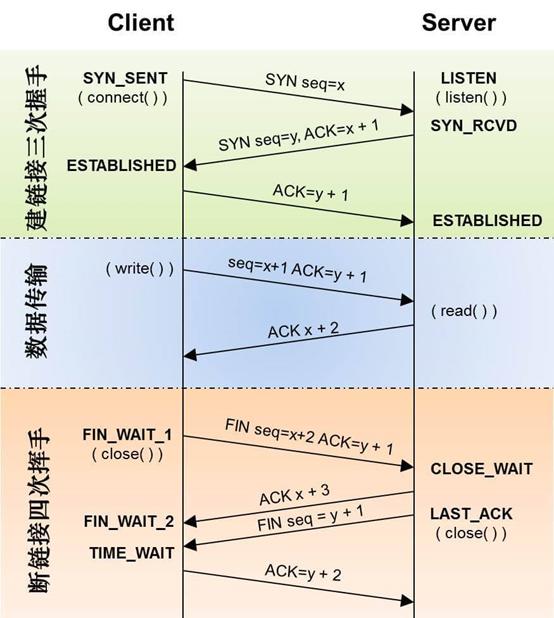
如果TCP的握手是俩次，会发生的情况有：

<1>如果client发给server的SYN报文因为网络原因，延迟发送。由于client没有收到server对SYN的确认报文，会重发SYN报文，服务器和回复ACK，连接建立。数据发送完毕，这条连接被正常关闭。这时，延迟的SYN报文发到了server，server误以为这是client重新发送的同步报文，又回复了一个ACK，和client建立了连接。  
<2>如果server给client发送的ACK报文因为网络原因，报文被丢弃，此时server认为已经建立好连接，但是client没有收到确认报文，认为没有建立好连接。client会重发SYN报文，此时server已经处于就绪状态，认为已经建立好连接。

如果TCP的握手是四次：

–1.client给server发送SYN同步报文；  
–2.server收到SYN后，给client回复ACK确认报文；  
–3.server给client发送SYN同步报文；  
–4.client给server发送ACK确认报文。  
第2.3步之间，server和client没有任何的数据交互，分开发送相当于多发了一次TCP报文段，SYN和ACK标识只是TCP报头的一个标识位。很明显，这两步可以合并，从而提高连接的速度和效率。

四次挥手流程



详细步骤：

1、客户端发送\*\*\*seq=x+2，确认号ack=y+1，并FIN置1；  
2、服务端接收以后，返回ack=x+3，并ACK置1；  
3、如果有数据没有传送完毕，等待传送完毕返回\*\*\*seq=y+1也就是返回确认号，并FIN置1，如果没有数据则合并ack=x+3，seq=y+1 FIN置1；  
4、最后客户端接收以后返回给服务端确认号ack=y+2来确认断开

**面试问题：**问：为什么连接的时候是三次握手，关闭的时候却是四次握手？

答：因为当Server端收到Client端的SYN连接请求报文后，可以直接发送SYN+ACK报文。其中ACK报文是用来应答的，SYN报文是用来同步的。但是关闭连接时，当Server端收到FIN报文时，很可能并不会立即关闭SOCKET，所以只能先回复一个ACK报文，告诉Client端，“你发的FIN报文我收到了”。只有等到我Server端所有的报文都发送完了，我才能发送FIN报文，因此不能一起发送。故需要四步握手。

问：为什么是四次？

答：TCP是全双工的连接，必须两端同时关闭连接，连接才算真正关闭。 如果一方已经准备关闭写，但是它还可以读另一方发送的数据。发送给FIN结束报文给对方对方收到后，回复ACK报文。当这方也已经写完了准备关闭，发送FIN报文，对方回复ACK。两端都关闭，TCP连接正常关闭。

问：为什么TIME\_WAIT状态需要经过2MSL(最大报文段生存时间)才能返回到CLOSE状态？

答：网络是不可靠的，有可以最后一个ACK丢失。所以TIME\_WAIT状态就是用来重发可能丢失的ACK报文。  
可以确保每成功建立一个TCP连接时，来自该连接先前化身的老的重复分组都已经在网络中消逝。

## **SCTP介绍**

SCTP (Stream Control Transmission Protocol)是一种传输协议，在TCP/IP协议栈中所处的位置和TCP、UDP类似，兼有TCP/UDP两者特征。  
SCTP是可以确保数据传输的，和TCP类似，也是通过确认机制来实现的。

### **SCTP和TCP的区别**

****1、 TCP是以字节为单位传输的，SCTP是以数据块为单位传输的****  
TCP接收端确认的是收到的字节数，SCTP接收端确认的是接收到的数据块。SCTP的这种数据块（被称为DATA CHUNK）通常会携带应用的一个数据包，或者说是应用要发送的一个消息。  
****2、TCP通常是单路径传输，SCTP可以多路径传输****  
TCP的两端都只能用一个IP来建立连接，连接建立之后就只能用这一对IP来相互收发消息了。如果这一对IP之间的路径出了问题，那这条TCP连接就不可用了。  
SCTP不一样的地方是，两端都可以绑定到多个IP上，只要有其中一对IP能通，这条SCTP连接就还可以用。  
****3、TCP是单流有序传输，SCTP可以多流独立有序/无序传输****  
SCTP可以在一个联合中支持多流机制，每个流（stream）都是独立的。  
每个流都有各自的编号，编码在SCTP报文中  
阻塞的流不会影响同一联合中的其他流，可以并行进行传输  
****4、TCP连接的建立过程需要三步握手，SCTP连接的建立过程需要四步握手****  
TCP连接建立过程，容易受到DoS攻击。在建立连接的时候，client端需要发送SYN给server端，server端需要将这些连接请求缓存下来。通过这种机制，攻击者可以发送大量伪造的SYN包到一个server端，导致server端耗尽内存来缓存这些连接请求，最终无法服务。  
SCTP的建立过程需要四步握手，server端在收到连接请求时，不会立即分配内存缓存起来，而是返回一个COOKIE。client端需要回送这个COOKIE，server端校验之后，从cookie中重新获取有效信息（比如对端地址列表），才会最终建立这条连接。这样，可以避免类似TCP的SYN攻击。  
****5、SCTP有heartbeat机制来管理路径的可用性****  
SCTP协议本身有heartbeat机制来监控连接/路径的可用性。