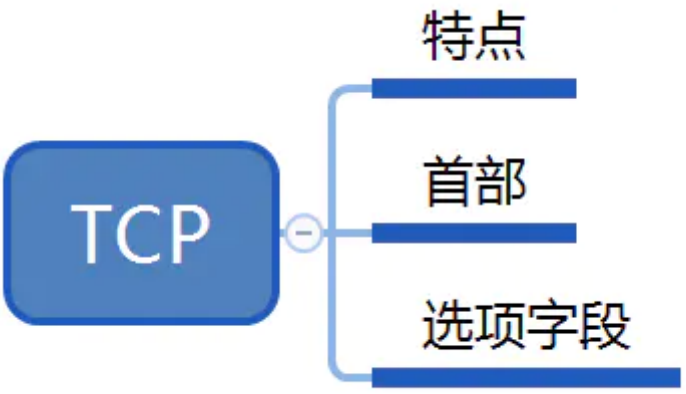
TCP协议

内容总览



1.TCP特点

(1) TCP 是**面向连接**的传输层协议。

> 应用程序是使用TCP协议之前，必须建立TCP连接。在传送数据完毕后，必须释放已建立的TCP连接。TCP连接是一条虚连接（逻辑连接），而不是一条真正的物理连接。

(2) 每一条TCP连接只能有**两个端点**，每条TCP连接只能是点对点的（一对一）。

(3) TCP提供**可靠交付**的服务。通过TCP连接传送的数据，**无差错、不丢失、不重复、并且按序达到。**

(4) **TCP提供全双工通信。**

TCP允许通信双方的应用进程在任何时候都能发送数据。TCP连接的两端都**设有发送缓存和接收缓存**，用来临时存放双方通信的数据。

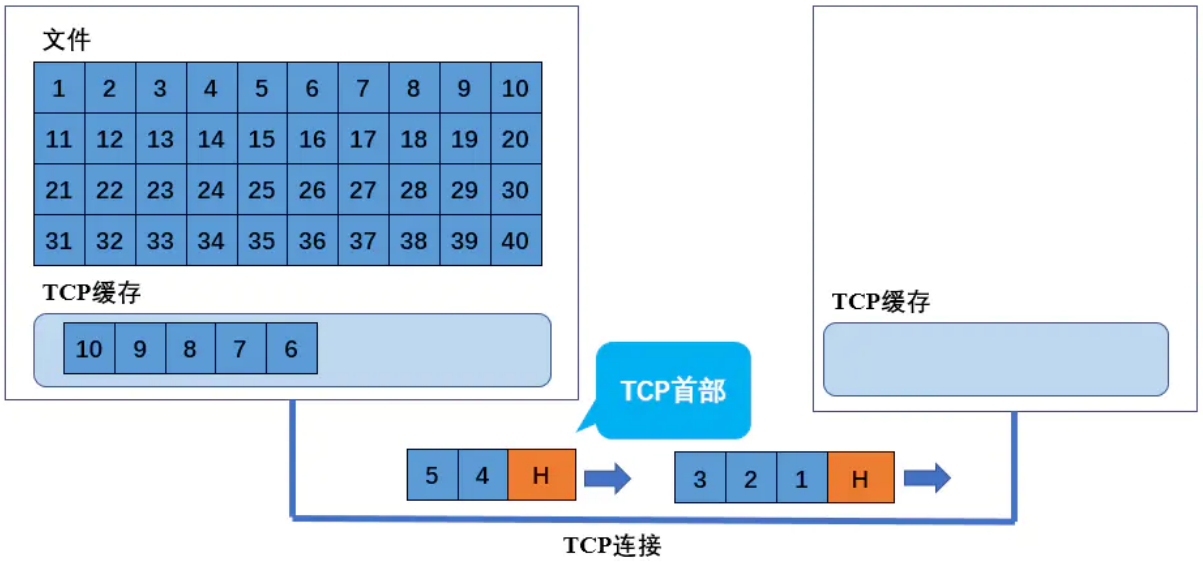
1. **发送缓存**：用于存放**准备发送的数据**和**已发送但尚未收到确认的数据**（因为TCP提供可靠交付服务，对于发送的数据必须要收到接收方的确认回复，如果数据丢失了，就可以从发送缓存中将刚发送的数据再发送一次，直到收到确认回复后，才会将这个数据从缓存中删除）。

2. **接收缓存**：用于存放**按序到达但尚未被接收应用程序读取的数据**和**不按序达到的数据。**

(5) TCP**面向字节流。**流是**指流入进程或从进程流出的字节序列。**

> 虽然应用程序和TCP的交互是一次一个数据块（大小可以不等），但是TCP把应用程序交下来的数据块看成仅仅一连串的**无结构的字节流。**

如下图所示，假如发送方要发送一个文件，会将这个文件按字节排序并编号，在发送时将这些字节放入TCP缓存中，从缓存中取出若干字节（可以不等）并加上TCP首部形成一个完整的报文段在链路上传送，最终到达接收方的TCP缓存中。



> TCP并不关心应用进程依次把多长的报文发送到TCP缓存中，而是根据接收方和当前网络拥塞的程度来决定一个报文段应包含多少个字节。所以如果应用进程传送到TCP缓存中的数据块太长，TCP就可以把它划分短一些再传送，而UDP是面向报文的，应用进程给出的数据块UDP就是按照给出的长度发送。

2.TCP的连接

每条TCP连接有两个端点，TCP连接的端点叫做**套接字（socket）或插口**，这里的套接字定义为：**端口号拼接到IP地址即构成了套接字。**

> 套接字的表示方法是在点分十进制的IP地址后面写上端口号，中间用冒号或逗号隔开。

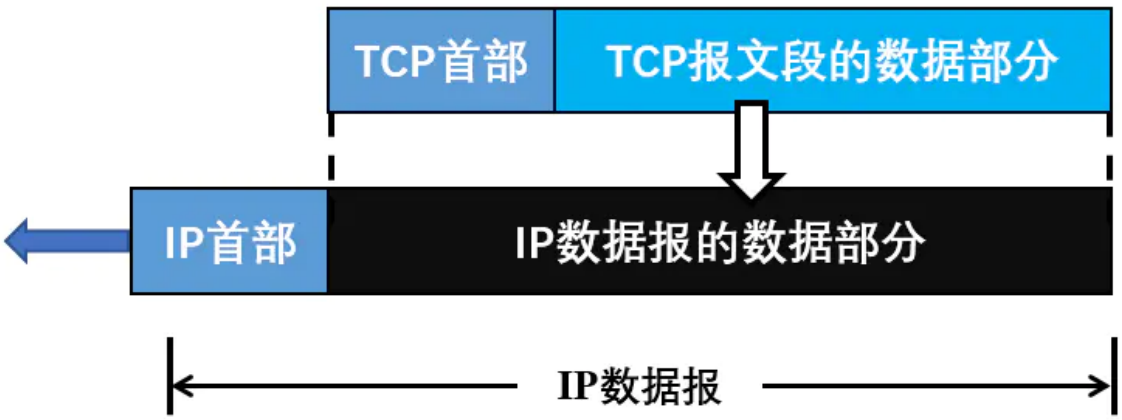
> **套接字 socket = （IP地址 : 端口号）**

例如，若IP地址为192.3.4.5，端口号是80，那么得到的套接字就是（192.3.4.5:80） **每条TCP连接唯一地被通信的两个端点（即两个套接字确定）所确定。**即：

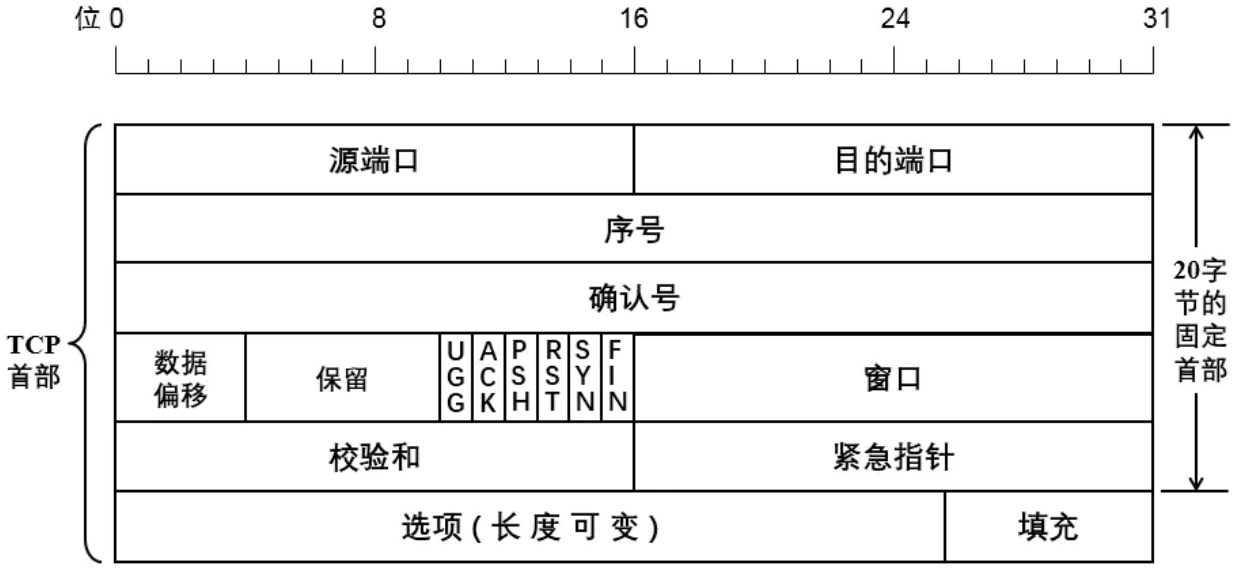
> TCP连接 = {socket1,socket2} = {（IP1 : prot1），（IP2 : prot2）}

3.TCP报文段的首部格式

一个TCP报文段分为**首部**和**数据部分**两部分。



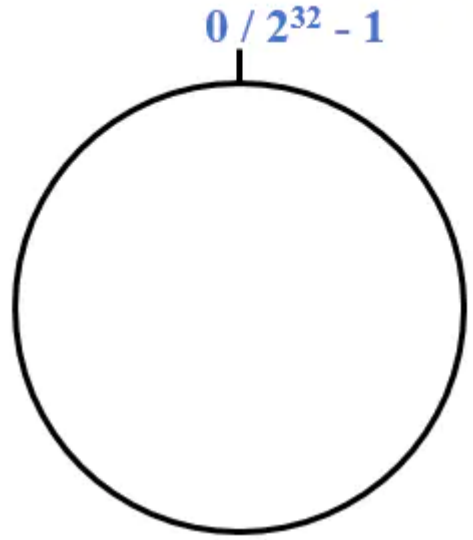
TCP报文段首部的前20个字节是固定的，后面有4n（n是整数）字节是根据需要而增加的选项。**因此TCP首部的最小字节是20字节。**



(1) **源端口和目的端口**：各占2自己，分别写入源端口号和目的端口号，TCP的分用功能也是通过端口实现的。

(2) **序号**：占4字节，序号范围是[0 , 2^32 - 1]，共 232个序号。TCP是面向字节流的，在一个TCP连接中传送的字节流中的**每一个字节都是按顺序编号。**首部中的序号字段值则指的是本报文段所发送的数据的**第一字节的序号。**

序号是可以重用的，当序号增加到 232 - 1后，下一个序号就又回到了0，所以序号逻辑上可以表示为一个循环数组。



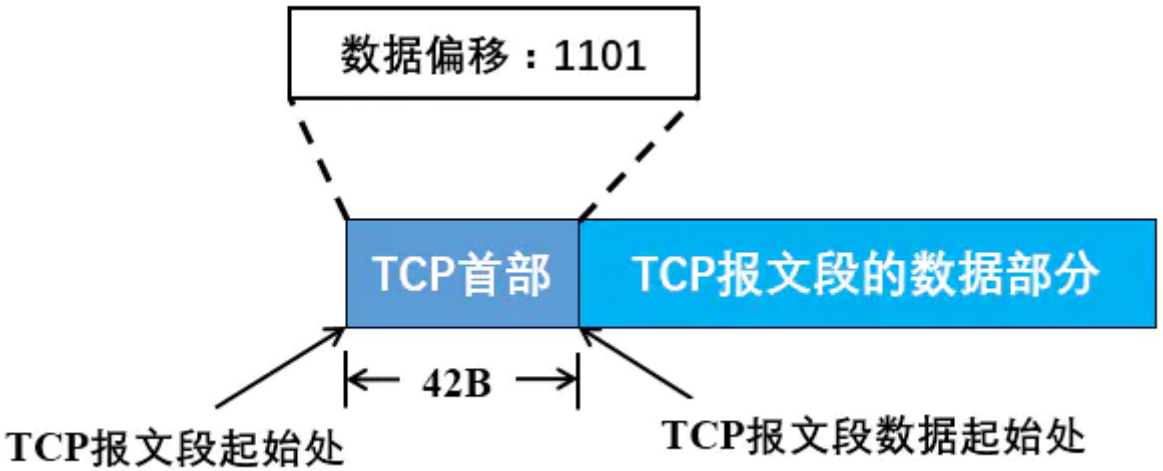
例如，若一个报文段的序号字段值是301，而携带的数据共有100字节，这就表明：本报文段的数据第一个字节的序号是301，最后一个字节的序号是400。如果还有下一个报文段，则其序号字段的的值应为401。

(3) **确认号**：占4字节，是**期望收到对方下一个报文段的第一个数据字节的序号。**

例如，B正确收到了A发送过来的一个报文段，其序号字段值是501，而该报文段的数据长度是200字节（序号501~700），这表明B正确收到了A发送的到序号700为止的数据，因此B期望收到A的下一个数据序号是701，TCP是可靠传输，收到数据后需要给发送方回复确认信息，所以B在收到数据后给A发送的确认收到的报文段中就把确认号置为701。

若确认号 = N,则表明：到序号N - 1为止的所有数据都已正确收到。

(4) **数据偏移**：占4位，单位：4B。它指出TCP报文段数据起始处距离TCP报文段的起始处有多远。**这个字段实际上是指出TCP报文段的首部长度。**



如上图所示，如果数据偏移字段的值为：1101（十进制13），所以数据偏移的值为13 × 4B = 42B，所以可知TCP报文段数据部分的起始处到TCP报文段的起始处（即TCP首部的）

数据偏移占4位，最大值为1111，即15，即数据偏移的值最大为60字节（TCP首部最大长度为60字节），又TCP首部有固定的20字节，所以TCP可选字段的长度不能超过40字节。

(5) **保留**：占6位，保留今后使用。

接下来是6个控制位

(6) **紧急URG（URGent）**：仅当URG = 1，表明后面的紧急指针字段才有效。**它表明系统此报文段有紧急数据，应尽快传送（相当于高优先级数据），而不要按照原来的排队顺序来传送。**

前面说到，在发送报文段时，需要将字节先存放在TCP缓存中，如果发送应用需要发送一个紧急指令，如中断指令（Control + C），如果不使用紧急数据，那么这两个字符就存在TCP缓存的末尾，直到前面的数据处理完才将这两个字符交给接收方。

当URG = 1时，发送应用进程就告诉发送方的TCP有紧急数据要传送，于是发送方TCP就把紧急数据插入到本报文段数据的最前面，而在紧急数据之后的数据仍是普通数据。

(7)**确认ACK（ACKnowledgment）**：仅当ACK = 1时确认号字段才有效。**TCP规定，在连接建立后所有传送的报文段都必须把ACK置1。**

(8) **推送PSH（PuSH）**：通常如果TCP缓存中字节很少，TCP会等待积累有足够多的字节后再构成报文段发送出去，当发送方将 PSH 置为 1时，并立即创建一个报文段发送出去，接收方TCP收到PSH = 1的报文段，就尽快地交付接收应用进程，而不再等到整个缓存都填满在向上交付。

> 这个字段适合在交互式的通信，在一端应用进程键入一个命令立即能够收到对方的响应，但是这种推送操作很少使用。

(9) **复位RST（ReSeT）**：当RST = 1时，表明TCP连接中出现了严重差错，必须释放连接，然后再重新建立传输连接。RST置为1还可以用来拒绝一个非法的报文段或拒绝打开一个连接。RST也可称为重建位或重置位。

(10) **同步SYN（SYNchronization）**：**在连接建立时用来同步序号。**当SYN = 1而ACK = 0时，表明这是一个连接请求报文段。对方同意建立连接，则应在响应的报文段中使用SYN = 1和ACK = 1.因此，SYN 置为1表示这是一个**连接请求或连接接收**报文。

(11) **终止FIN（FINis）**：用来释放一个连接。当FIN = 1时，表明此报文段的发送方的数据发送完毕，并要求释放传输连接。

(12) **窗口**：占2字节，是指发送本报文段的一方的**接收窗口。**窗口的值表示：从本报文段首部中的确认号算起，接收方目前允许对方发送的数据量。即**窗口值作为接收方让发送方设置其发送窗口的依据。**之所以要有这个限制，是因为接收方的数据缓存空间是有限的。

> 例如，A是发送方，B是接收方，B给A发送一个确认接收数据的报文，其确认号701（表示701之前的所有数据都已经正确收到，期望A下一个报文段的第一个数据字节序号为701），窗口字段的值是1000，这就是告诉发送方A：从701号算起，我的接收缓存还可以接收1000个字节数据，在你给我发送数据的时候，你需要考虑一下我的接收能力。

> **窗口字段明确指出了现在允许对方发送的数据量。窗口值是经常在动态变化。**

(13) **校验和**：占2字节。校验和字段校验的范围包括**首部**和**数据**这两个部分。和UDP用户数据报一样，在计算校验和时，需要在TCP报文段的前面加上12字节的**伪首部**。伪首部的格式和UDP伪首部的格式一样，只是需要将协议字段改为6，TCP协议号是6。

(14) **紧急指针**：紧急指针只有在URG= 1时才有意义，它和URG字段配合使用，它指出了报文段中紧急数据的字节数，因此，紧急指针指出了紧急数据的末尾在报文段中的位置。

> 即使窗口的值为0也可以发送紧急数据。

(15) **选项和填充**：长度可变，最长可达40字节。填充字段是为了使整个TCP首部的长度是4字节的整数倍。

4.几个选项字段

(1) **最大报文段长度MSS（Maximum Segment Size）**：是指**每个TCP报文段中数据字段的最大长度**，即TCP报文段长度减去TCP首部长度。**它是为了考虑网络利用率。**

> TCP报文段的数据部分，需要加上TCP首部至少20个字节（即没有可变部分）和IP首部至少20字节才能组成一个IP数据报。若选择较小的MSS长度，例如，MSS是1，即TCP报文段数据部分长度为1个字节，那么IP层传输的数据报的开销至少有40个字节，这就导致网络的利用率降低。
> 反之，如果MSS非常大，那么在IP层传输时就需要分片，在终点时需要将收到的各个分片重新组装成原来的TCP报文段，这样也会使网络开销变大。
> 因此，MSS应尽可能大些，只要在IP层输层传输时不需要分片就行。默认值是536字节长，因此所有在互联网上的主机都应能接受的报文段长度是536 + 20 = 556字节。

(2) **窗口扩大**：占3字节，这个选项是为了**扩大窗口。**TCP首部的窗口字段长度是16位，因此最大的窗口大小是64K字节，窗口扩大中有一个字节表示**移位值S。**新的窗口值等于TCP首部中的窗口位数从16增加到（16 + S）。移位值允许使用的最大值是14，相当于窗口最大值增加到2(16+14) - 1。

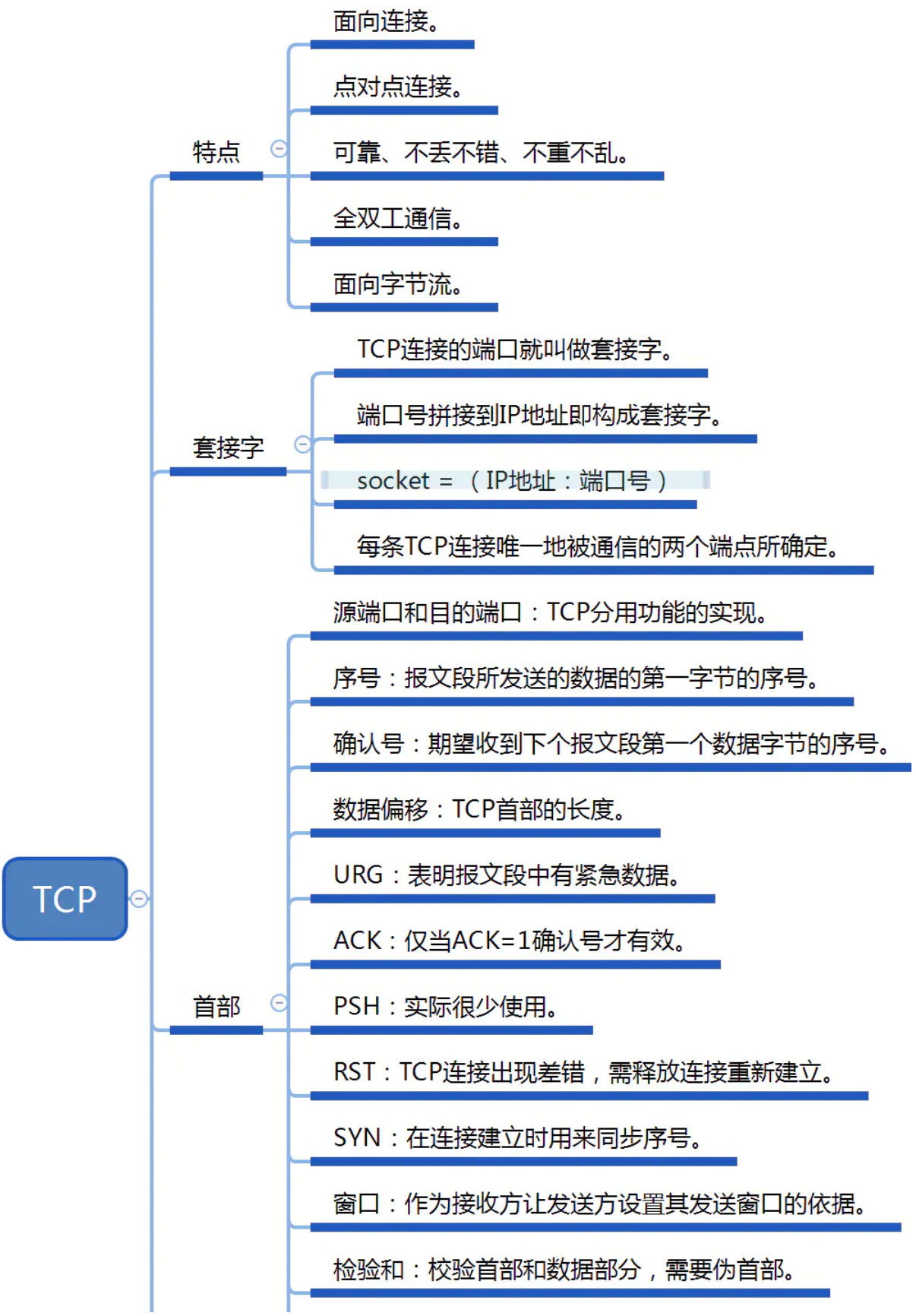
> 窗口扩大选项在双方初始建立TCP连接时进行协商，如果某一端实现了窗口扩大，当它不需要扩大其窗口时，可发送S = 0的选项，使窗口大小回到16。

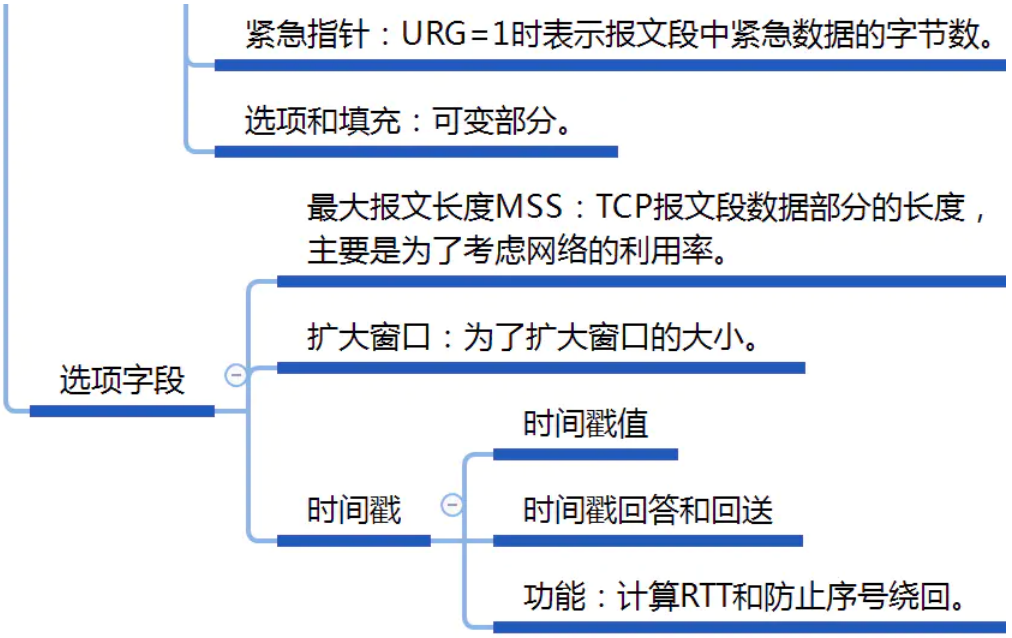
(3) **时间戳**：占10字节，其中最主要的是**时间戳值字段**（4字节）和**时间戳回送回答字段**（4字节）。时间戳选项有以下两个功能：

1. **计算往返时间RTT。**发送方在报文段时把当前时钟的时间值放入时间戳字段，接收方在确认该报文段时把时间戳字段复制到时间戳回送回答字段。因此，发送方在接收到确认报文后，就可以准确的计算出RTT来。

2. **用于处理TCP序号超过2^32的情况**，这又称为**防止序号绕回PAWS**（Protect Against Wrapped Sequence numbers）。

5.总结




紧急指针：URG=1时表示报文段中紧急数据的字节数。
选项和填充：可变部分。
选项字段
最大报文长度MSS：TCP报文段数据部分的长度，主要是为了考虑网络的利用率。
扩大窗口：为了扩大窗口的大小。
时间戳
时间戳值
时间戳回答和回送
功能：计算RTT和防止序号绕回。