粘包和半包

什么情况下会出现粘包和半包的问题呢？

TCP会出现粘包和半包的问题，UDP不会。

为什么TCP会出现粘包和半包的问题呢？

因为TCP是基于链接的可靠的流协议。而流消息是没有明确的边界的，所以无法判断哪一段流属于一个消息。

什么是粘包？

粘包：发送两条消息的时候，接收方一次性读取的两条消息，这种情况就叫粘包。（比如一条消息是ABC，一条消息是DEF，接收方读取到的却是ABCD）

什么是半包？

半包：发送一条消息的时候，接收方接收到的是两条消息，这种情况就叫半包。（比如一条消息是ABCD，接收方接收到的是一条ABC，一条D）

什么情况下会出现粘包？

一种情况是发送的消息的大小小于套接字缓冲区的大小。

一种情况是接收方没有及时接收。

什么情况下会出现半包？

一种情况是发送的消息的大小大于套接字缓冲区的大小。

一种情况是发送的消息的大小大于最大传输单元（MTU），必须进行拆包。

为什么要有缓冲区？

如果不使用缓冲区的话，每次写入操作CPU都会直接写到磁盘上，速度是很慢的。那如果先将写入的内容放在缓冲区，达到一定的大小之后，再一次性的写入磁盘上，这样的速度是会比较快的。总而言之，就是写入缓冲区的速度会比写入磁盘的速度快。

如何解决粘包和半包的问题？

有三种方法：

一种是：固定缓冲区的大小。服务端和客户端都使用相同的缓冲区大小，如果发送的数据不足缓冲区的大小的话，就使用空字符弥补到缓冲区的大小。（缺点是：当发送的数据较小的时候，会使用空字符进行弥补，增加了网络传输的负担。）

一种是：封装请求协议。将请求的数据封装成两部分，数据的大小+数据正文。读取的时候，先读取数据的大小，然后根据大小读取数据正文部分。（缺点是：编码成本较高。）

一种是：以特殊字符结尾，按行读取。

总而言之，上述的三种方法都是为了能够知道流的边界。