1. 基于帧同步下发需求的UDP方案
   1. 需求

帧同步下发 不可漏帧， 消息顺序必须保障

TCP已满足这两点，需要相对TCP的优势

* 1. 主体规则

每帧下发信息 附带 帧序号

客户端成功收到后，立刻回执通知服务器 正常收到

假设服务器收到第m帧的回执

则在下发第n帧消息的时候，实际下发 m->n帧的信息

* 1. 举例

服务器                     客户端  
发送1  
发送1 2                    收到1 发送回执1  
收到回执1  
发送2 3                      
发送2 3 4                  收到2 3发送回执3  
发送2 3 4 5               收到1 2（丢弃）  
收到回执3  
发送4 5 6

1. 与直接使用TCP相比的优势
   1. TCP 的顺序保障原理

基础原理：

发送，确认，再发送下一个（拆分数据为固定大小块）

发送 -------》

《------ 确认

发送-------》

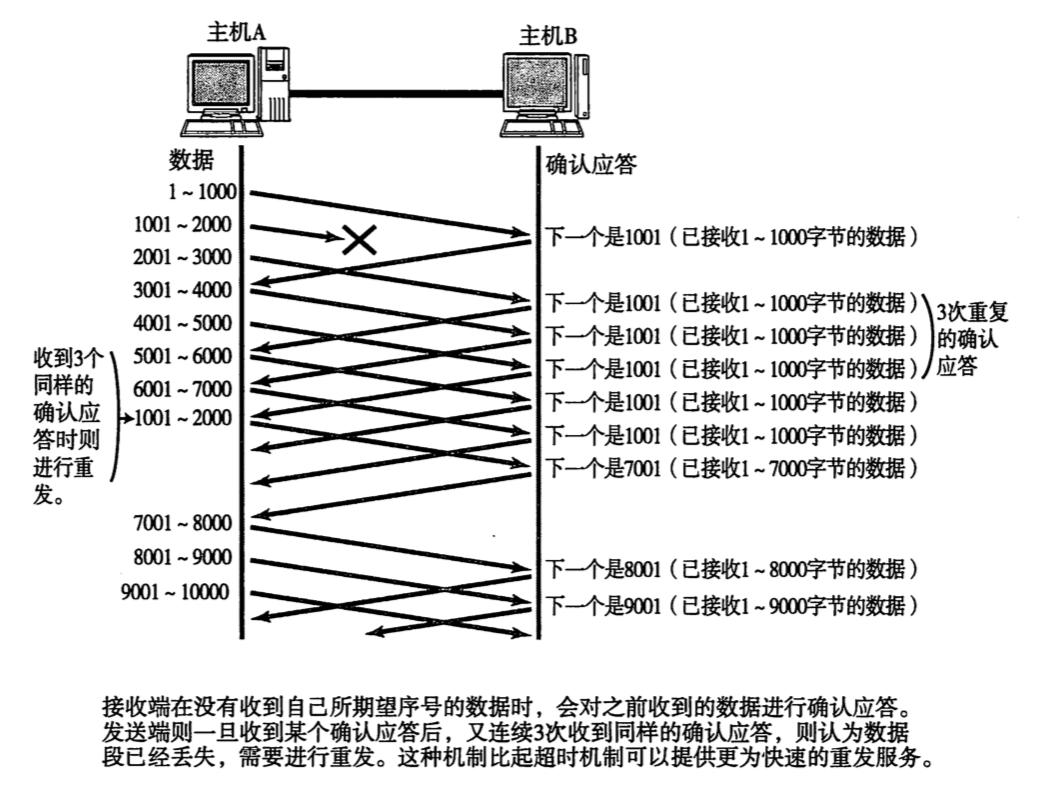
实际优化

窗口（缓冲区， 多个确认等待）

窗口大小就是指无需等待确认应答而可以继续发送数据的最大值

确认应答丢失，会被更高序号的确认应答确认

三次同样序号的确认应答，可确认数据包丢失，进行重传



* 1. 上述UDP方案相比的优劣

UDP

优势：丢失后的“重传”更及时，适应更及时的响应需求

劣势：更多的“重传”量（在帧同步操作指令的前提小，消息总量较小，认为可以接受）