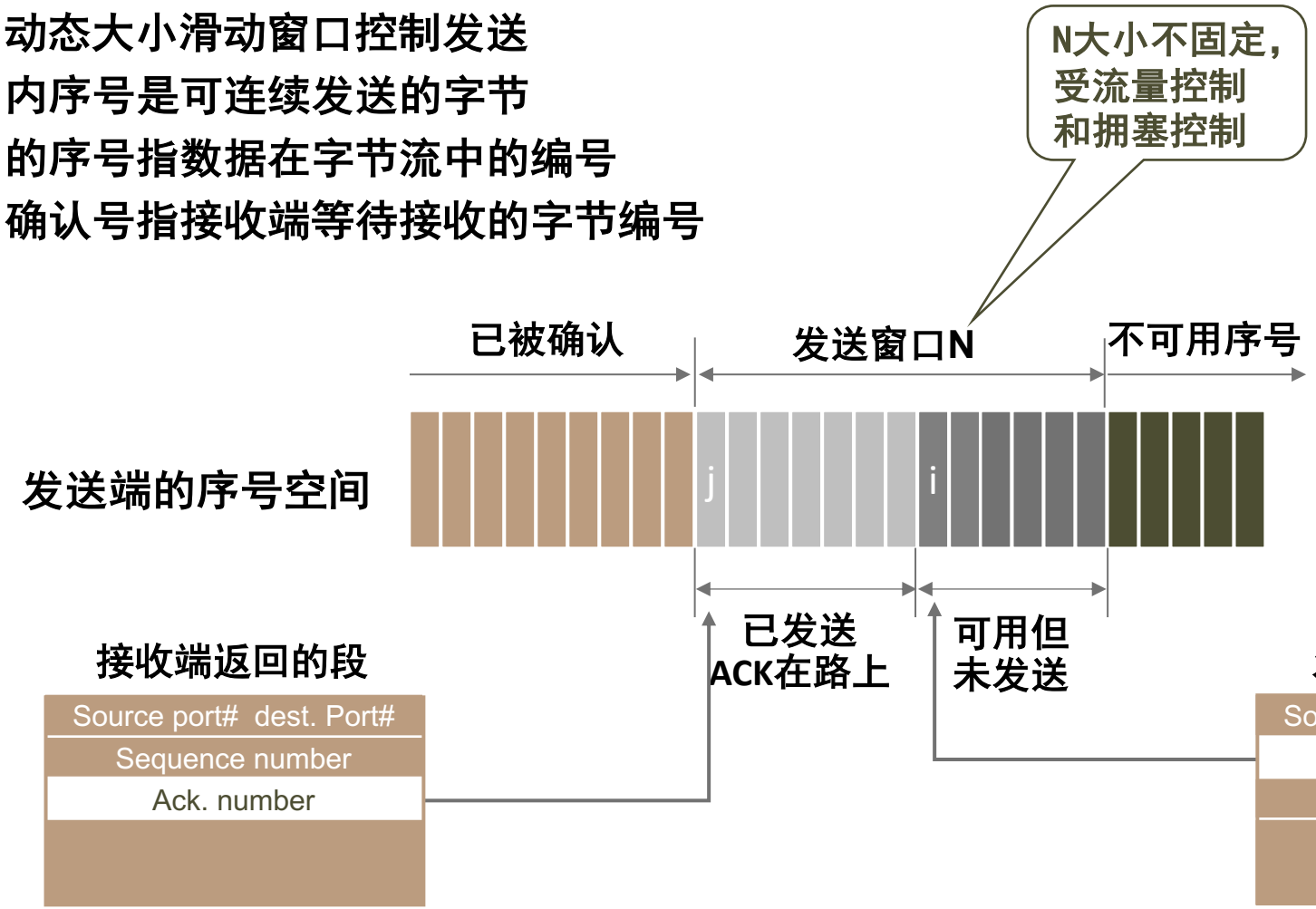
案例学习之
TCP的可靠数据传输

TCP段编号以及确认号

- TCP采用动态大小滑动窗口控制发送
- 发送窗口内序号是可连续发送的字节
- 发出段中的序号指数据在字节流中的编号
- 返回段中确认号指接收端等待接收的字节编号



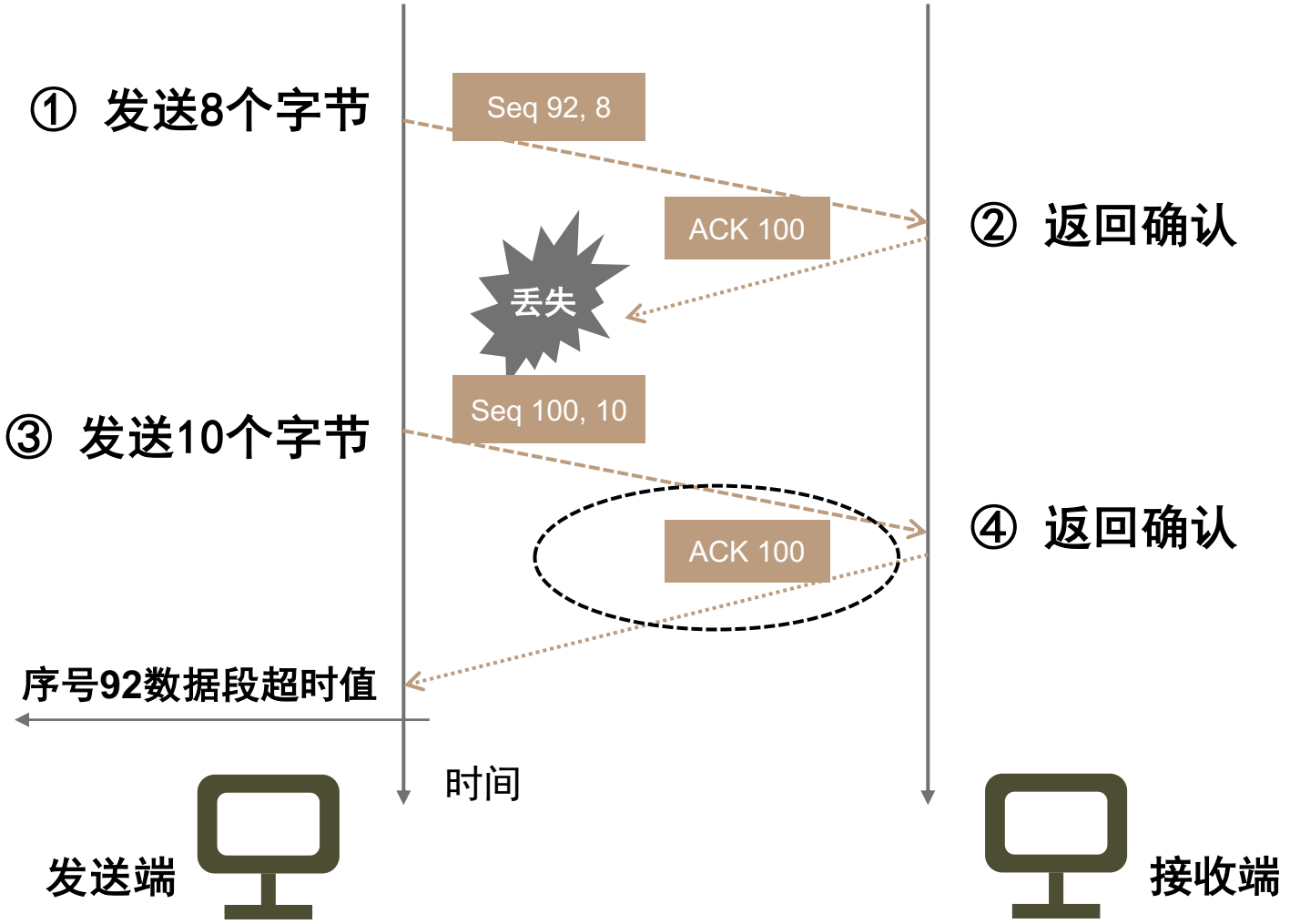
TCP的控制比特

- SYN
- ACK
- FIN
- URG
- PSH
- RST
- ECN
- CWR

TCP的累计确认机制

累计确认优点

- 后续确认段包含了之前传输正确的隐含信息
- 确认段丢失不一定导致发送端的重传



只要某段的确认在该段超时之前到达发送端，因累计确认含义，发送端就能肯定该段已经被正确接收无需重发。

Seq 92, 8 序号92，长度8字节的TCP数据段

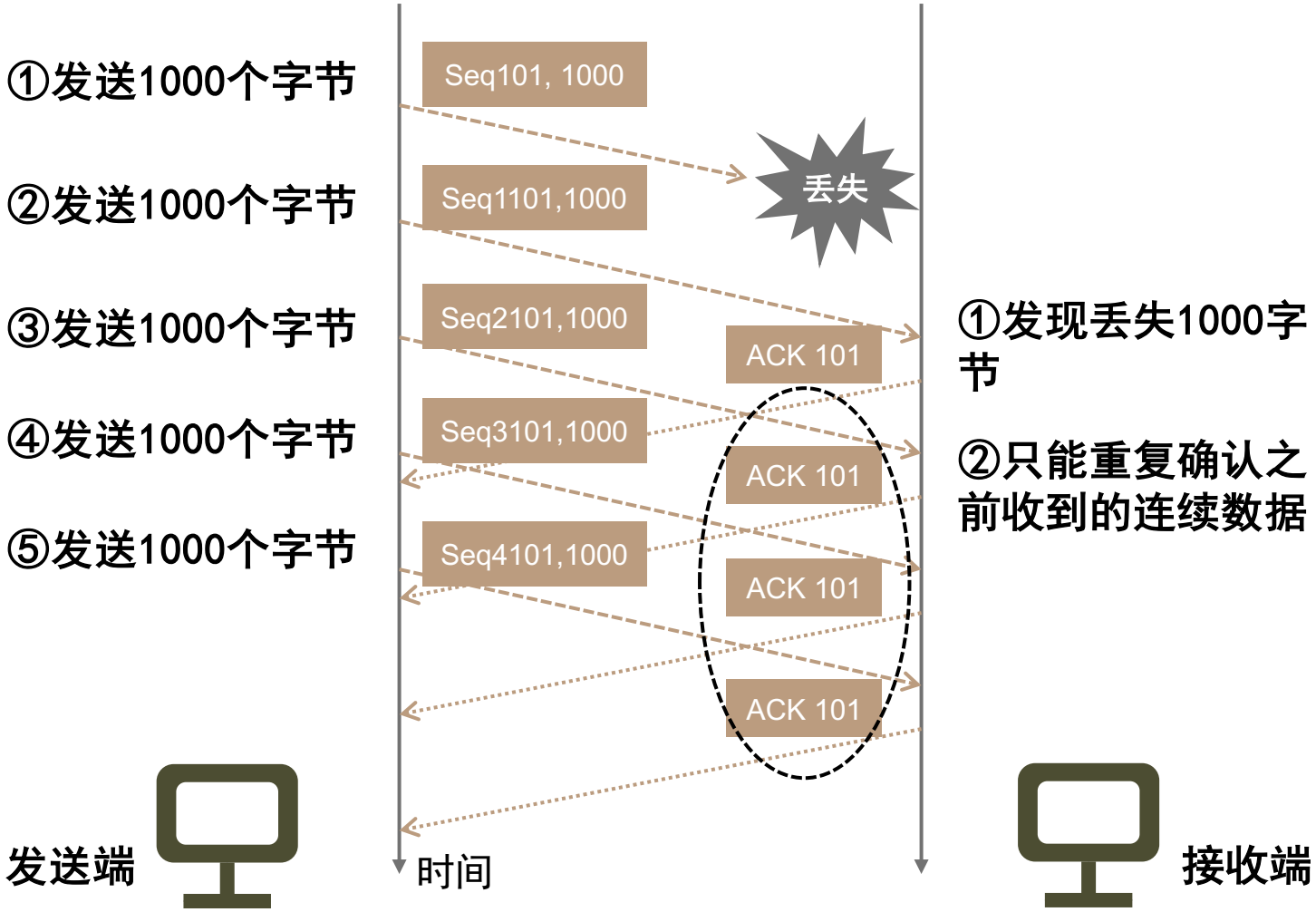
TCP的可靠性保证

示例：发送端的初始序号为101，
发送窗口大小=5000

累计确认缺点

- 发送端不能收到所有成功发送的段的确认信息
- 只知道已收到的数据流中的某一个位置信息
- 缺乏全部成功传送的信息
  →累计确认的效率降低

TCP没有否定确认机制，接收端只能通过重复确认信息来报告出错情况。



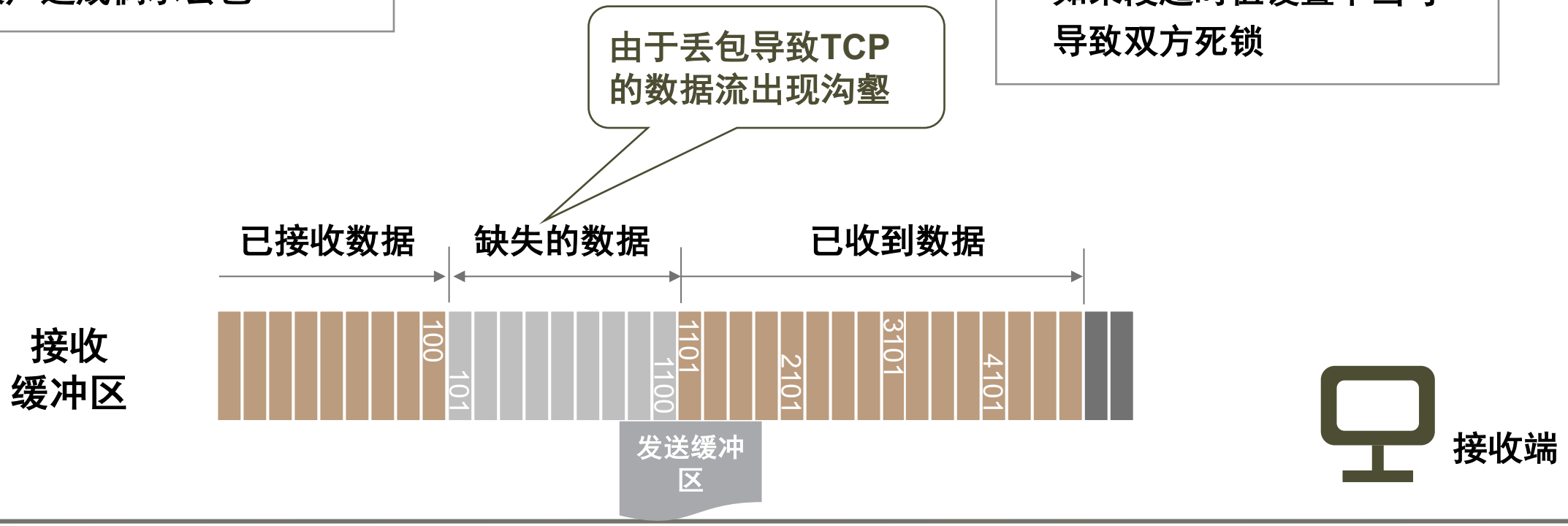
TCP丢失数据原因和后果

丢失数据原因

- 网络拥塞造成途径路由器无缓冲区可用，IP包被丢弃
- 突发噪声造成偶尔丢包

丢失数据后果

- 已经收到的数据不能立即上交给应用进程
- 如果段超时值设置不当可导致双方死锁

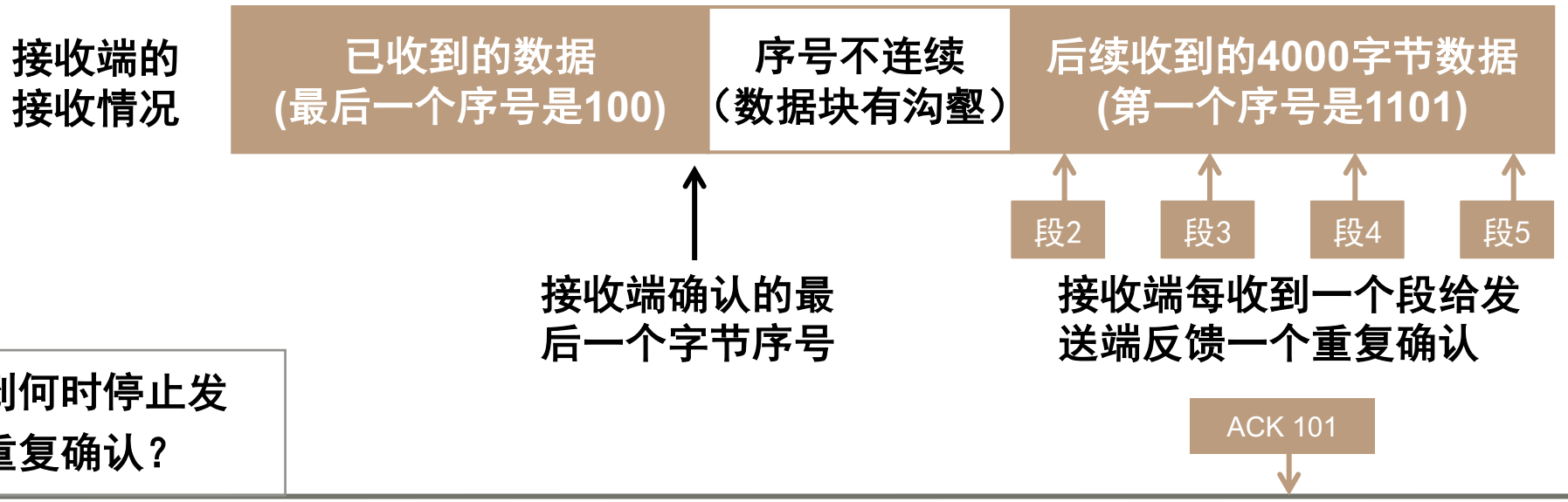


TCP快速重传机制

快速重传：发送端检测到三个重复ACK立即重传该ACK所指的段，而不是等待该段超时后再重传。

基于超时重发机制不足

- 发送端等待一定时间才能重发可能丢失的段
- 加剧了端-端的延迟



? 直到何时停止发送重复确认？

示例：快速重传机制的应用

发送端

① 发送1000个字节

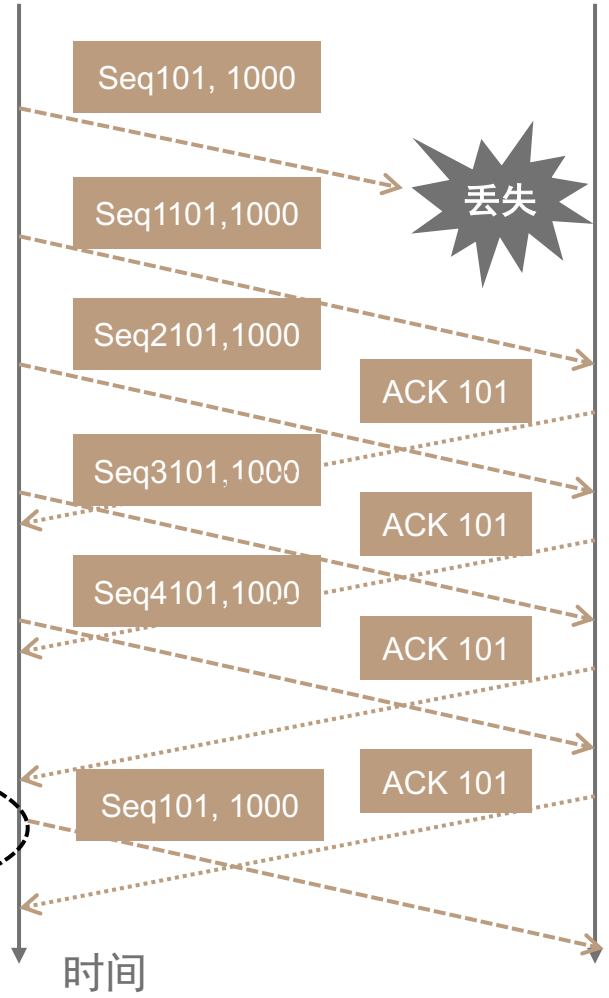
② 发送1000个字节

③ 发送1000个字节

④ 发送1000个字节

⑤ 发送1000个字节

⑥不等超时，立即重传第一段



接收端

① 发现丢失1000字节

② 只能重复确认之前收到的连续数据

③ 补齐丢失数据并上交给应用进程