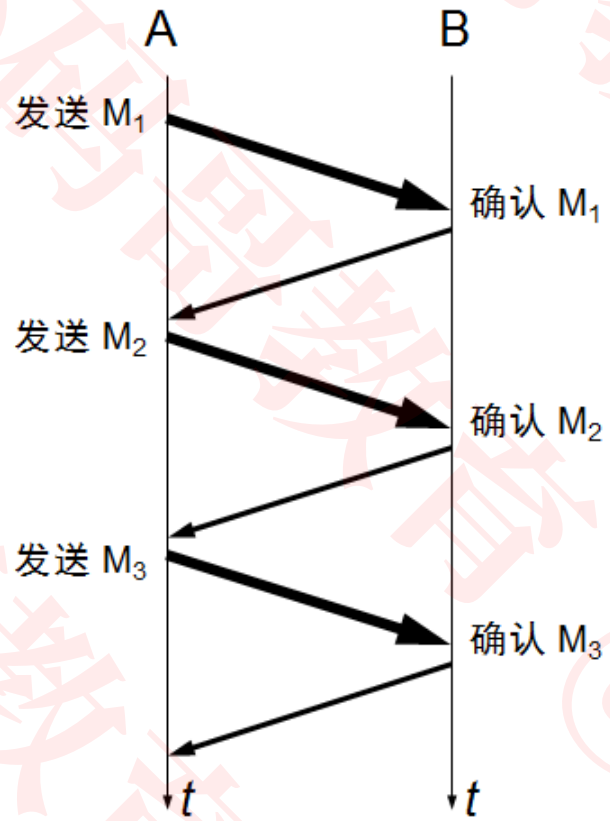
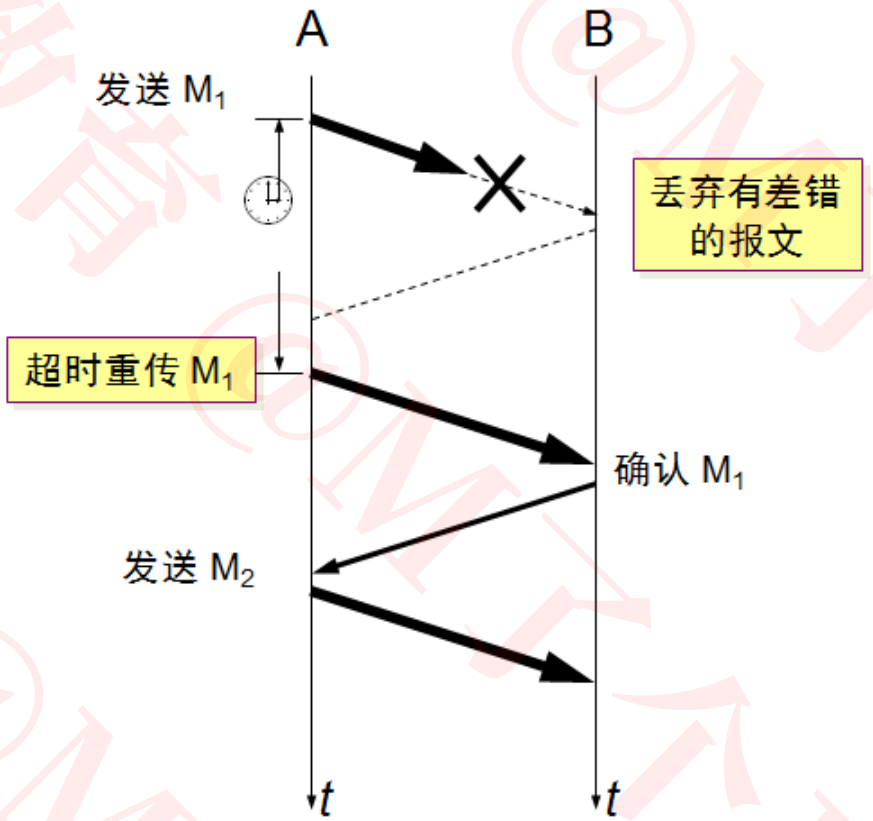
TCP – 可靠传输 – 停止等待ARQ协议

■ ARQ（Automatic Repeat–reQuest），自动重传请求

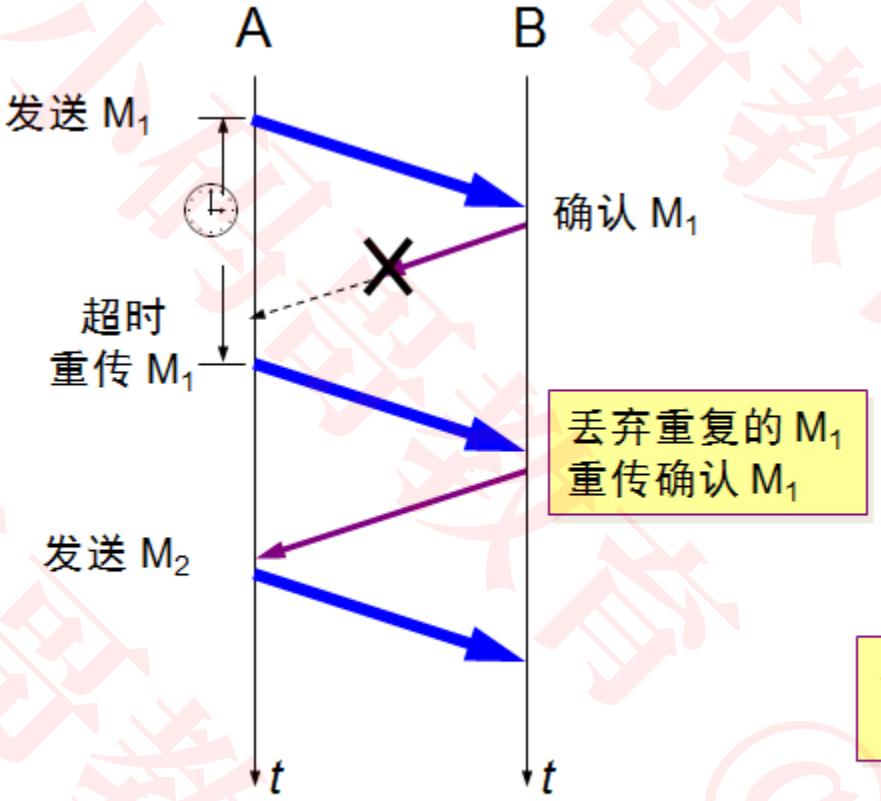


(a) 无差错情况

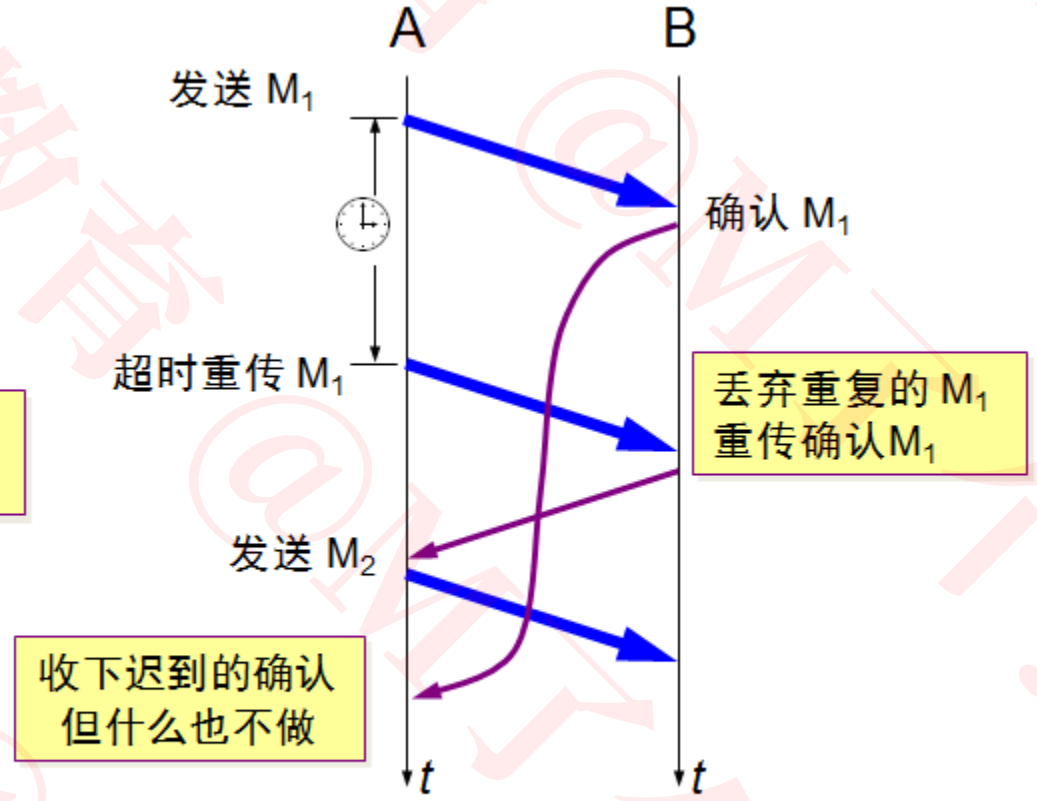


(b) 超时重传

TCP – 可靠传输 – 停止等待ARQ协议



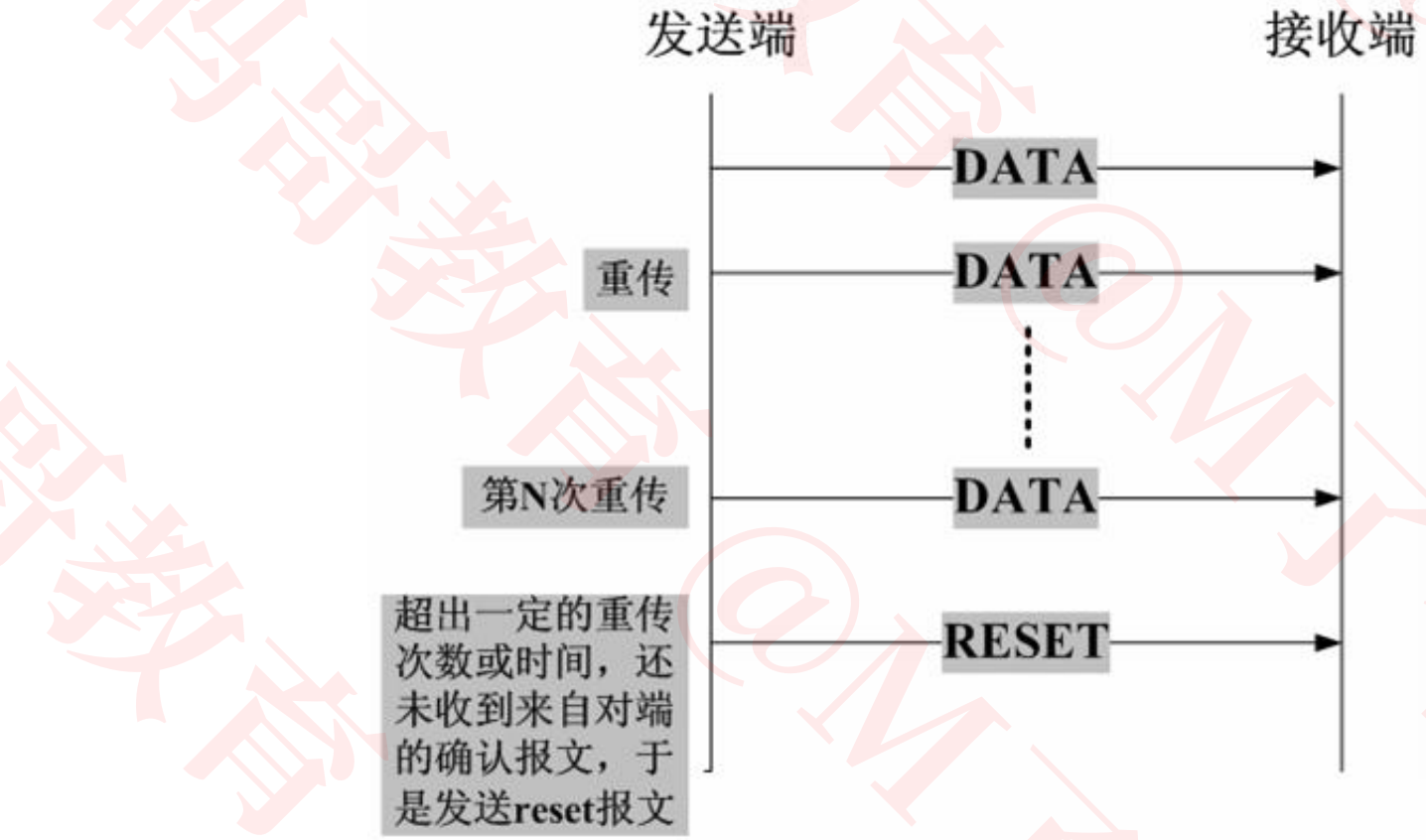
(a) 确认丢失



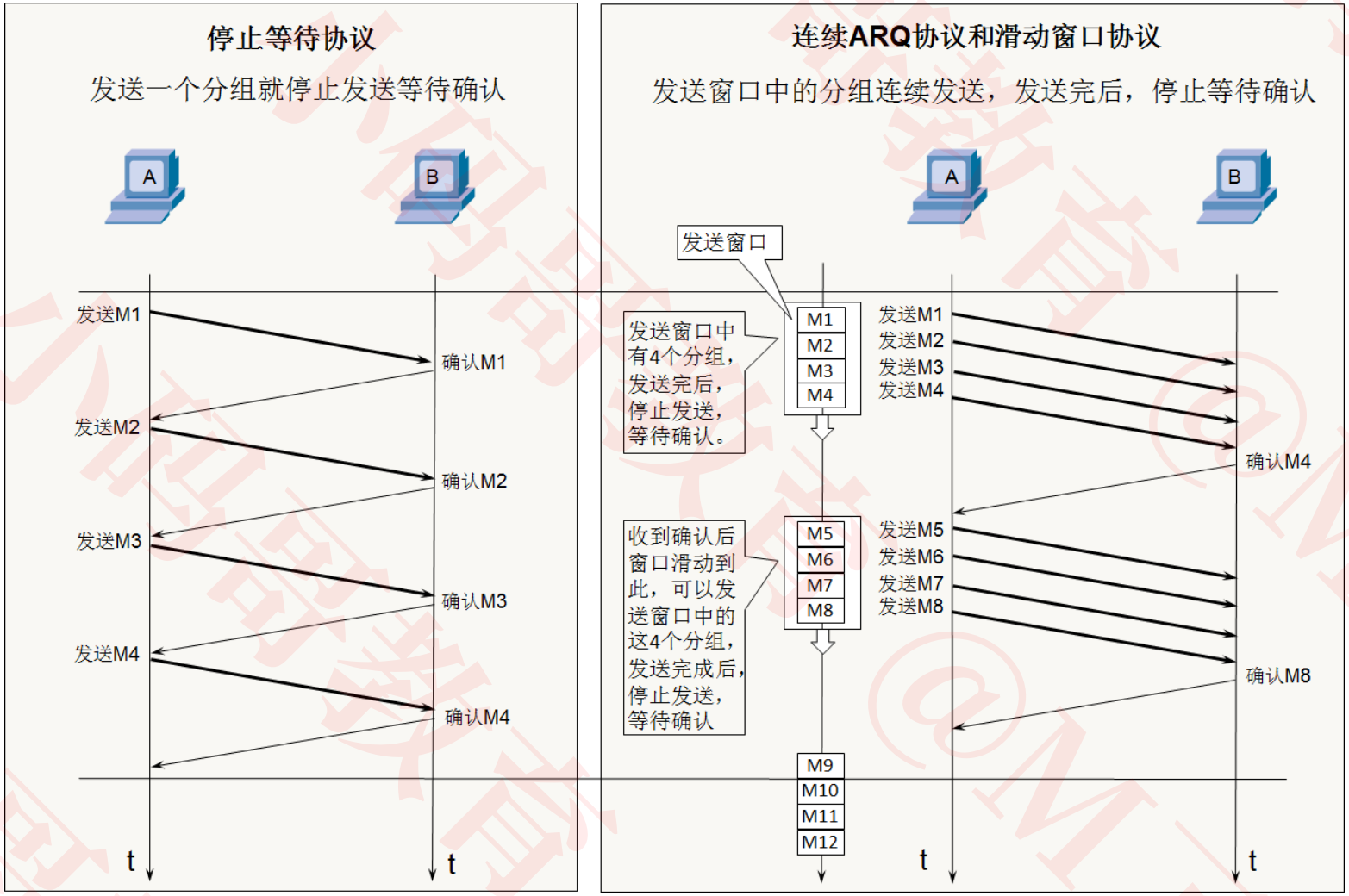
(b) 确认迟到

疑问

- 若有个包重传了N次还是失败，会一直持续重传到成功为止么？
- 这个取决于系统的设置，比如有些系统，重传5次还未成功就会发送reset报文（RST）断开TCP连接



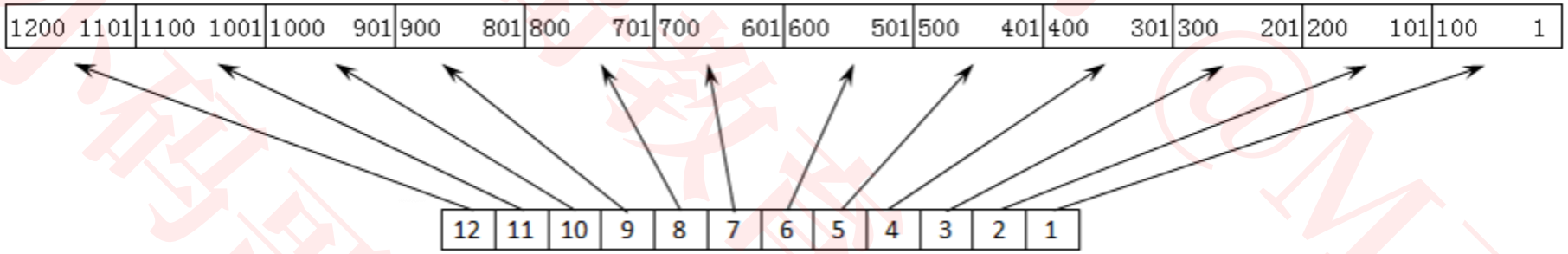
TCP – 可靠传输 – 连续ARQ协议 + 滑动窗口协议



- 如果接收窗口最多能接收4个包
  - 但发送方只发了2个包

- 接收方如何确定后面还有没有2个包?
  - 等待一定时间后没有第3个包
  - 就会返回确认收到2个包给发送方

TCP – 可靠传输 – 连续ARQ协议 + 滑动窗口协议



- 现在假设每一组数据是100个字节，代表一个数据段的数据
- 每一组给一个编号

TCP – 可靠传输 – SACK（选择性确认）

- 在TCP通信过程中，如果发送序列中间某个数据包丢失（比如1、2、3、4、5中的3丢失了）
- TCP会通过重传最后确认的分组后续的分组（最后确认的是2，会重传3、4、5）
- 这样原先已经正确传输的分组也可能重复发送（比如4、5），降低了TCP性能

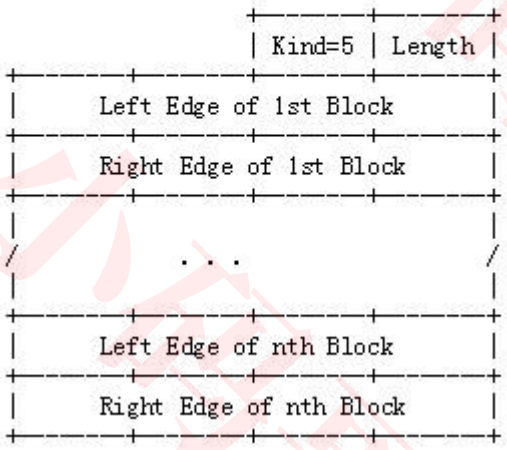
- 为改善上述情况，发展出了SACK（Selective acknowledgment，选择性确认）技术
  - 告诉发送方哪些数据丢失，哪些数据已经提前收到
  - 使TCP只重新发送丢失的包（比如3），不用发送后续所有的分组（比如4、5）

TCP – 可靠传输 – SACK（选择性确认）

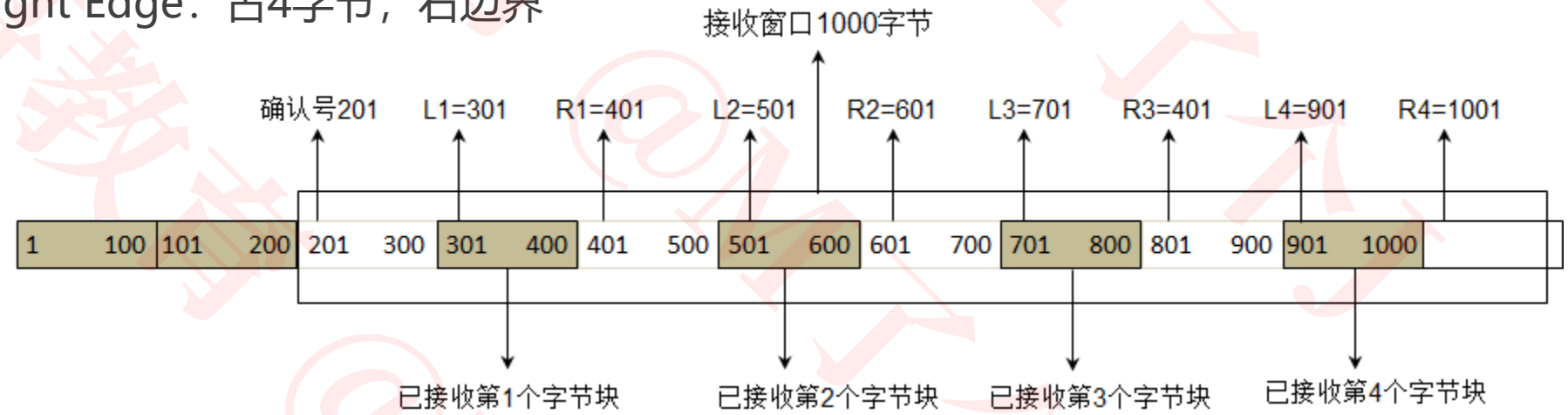
TCP SACK Option:

Kind: 5

Length: Variable



- SACK信息会放在TCP首部的选项部分
  - Kind：占1字节。值为5代表这是SACK选项
  - Length：占1字节。表明SACK选项一共占用多少字节
  - Left Edge：占4字节，左边界
  - Right Edge：占4字节，右边界



- 一对边界信息需要占用8字节，由于TCP首部的选项部分最多40字节，所以
  - SACK选项最多携带4组边界信息
  - SACK选项的最大占用字节数 = 4 * 8 + 2 = 34

小码哥教育 SEEMYGO

思考一个问题

- 为什么选择在传输层就将数据“大卸八块”分成多个段，而不是等到网络层再分片传递给数据链路层？
  - 因为可以提高重传的性能
  - 需要明确的是：可靠传输是在传输层进行控制的
    - ✓ 如果在传输层不分段，一旦出现数据丢失，整个传输层的数据都得重传
    - ✓ 如果在传输层分了段，一旦出现数据丢失，只需要重传丢失的那些段即可