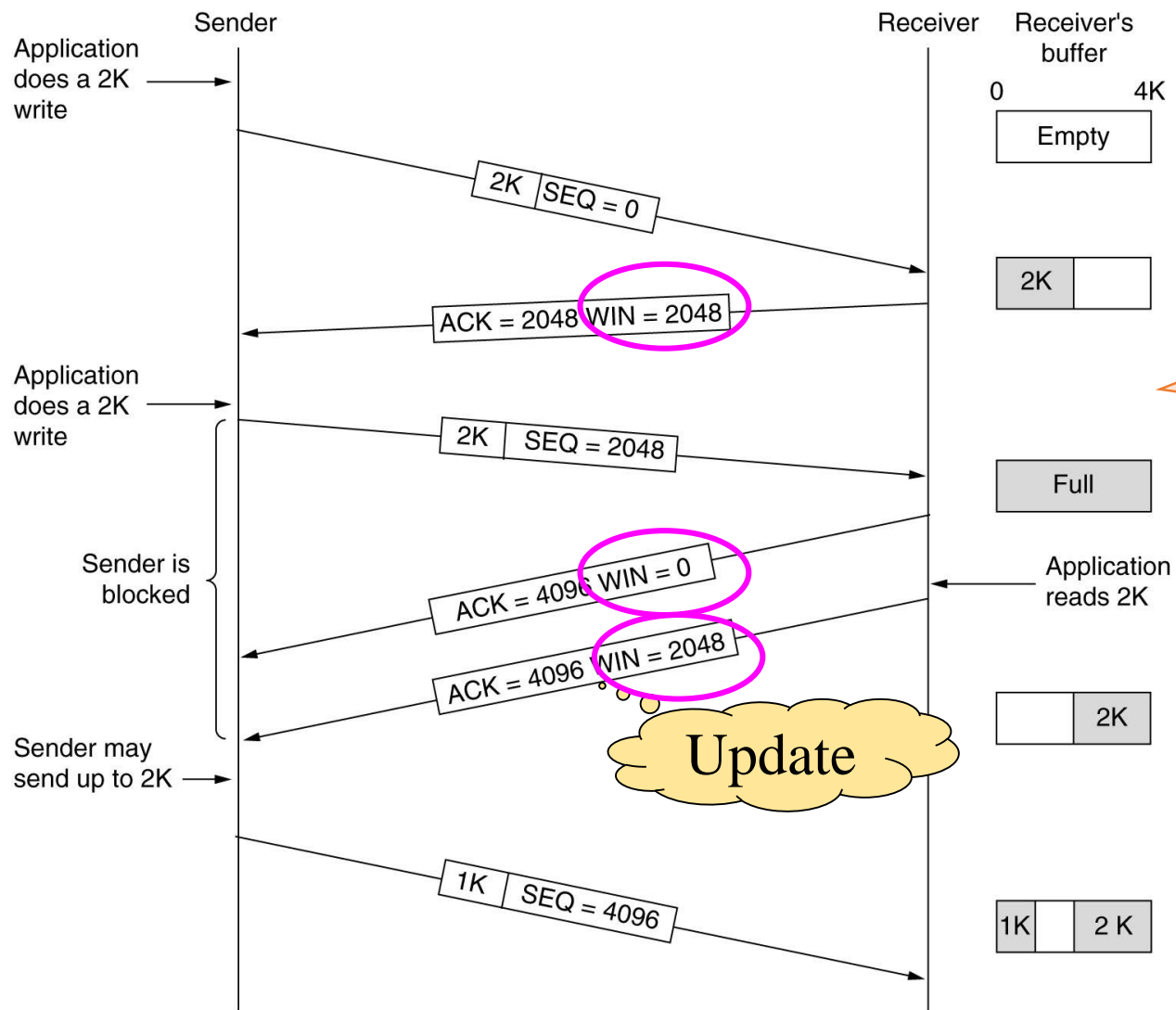


## 第六章 传输层

# TCP 传输策略

# TCP 传输策略



窗口尺寸受制  
于接收方！



# TCP 传输层策略

- 当窗口数为 0 时，发送者不能正常发送数据段，除非：
  - **Urgent** 数据。比如，用户想杀掉远端机器上的进程的时候，可以发送数据
  - 发送者可以发送一个字节的数据段，以便让接收者再次发送期待接收的字节号和窗口数（**避免死锁**）

发送者不需要马上发送应用程序产生的数据

接收者也不需要马上发送应答（当收到数据的时候）



考虑一个指向某交互式编辑器（远程）的TELNET 连接，该编辑器对用户的每次击键都作出响应，在最坏的情况下：

- ▼ 当用户敲入一个字符的时候，被送到传输实体，创建一个21字节的数据段，在传到网络层，变成了41字节的IP分组
- ▼ 接收方（运行着编辑器的远端机）收到这个信息后，会立发送一个40字节的确认分组（20字节的 TCP段头和20字节的IP头）

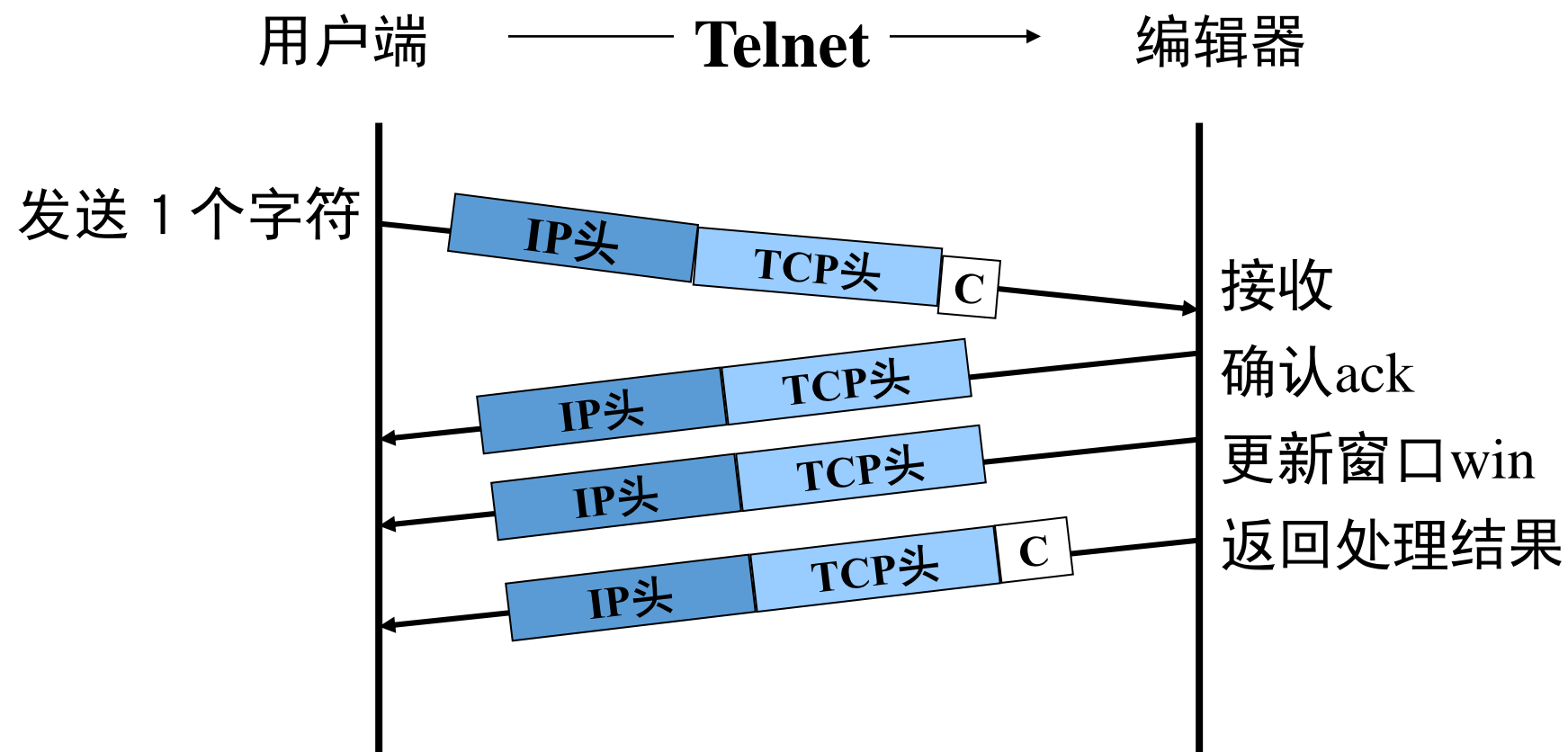
考虑一个指向某交互式编辑器（远程）的TELNET 连接，该编辑器对用户的每次击键都作出响应，在最坏的情况下：

❖ 随后，当编辑器读取出这个字节，TCP实体发送一个窗口更新，这个分组也是40字节

❖ 最后，当编辑器处理了这个字符，它发送一个41字节的分组作为该字符的回显

总共累计起来，对于每个敲入的字符，需要至少 162 字节的带宽（还没有考虑到链路层的开销），发送4个数据段。

# 远程交互telnet的最坏情形图示



处理一个字符需要传送4次数据段，占用162字节带宽



## 怎样优化接收端

接收端可以推迟500ms发送确认分组和窗口更新窗口，以便  
可以免费搭载在处理后的回显分组内（free ride）



## 怎样优化发送端

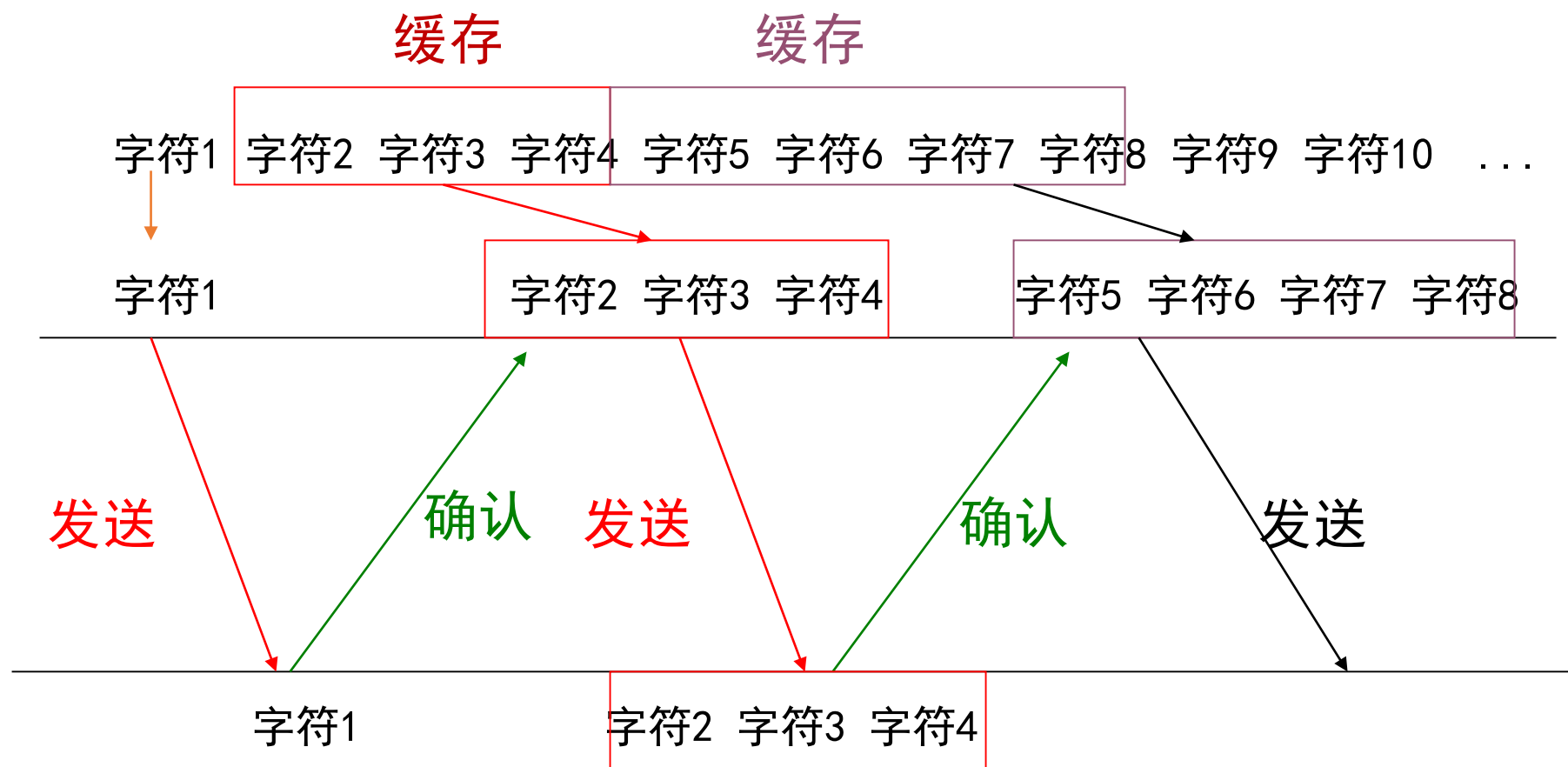
Nagle's algorithm (1984):

- 当数据以一次一字节的速度到达的时候，只发送第一个字节，然后将后续的字节缓存起来，直到发出的字节得到确认
- 将缓存起来的字节在一个数据段中发出，再继续缓存，直到发出的数据得到确认
- Nagle算法在很多TCP上实现，但是**有些时候最好禁用**，比如：当一个X-Windows应用在互联网运行的时候，鼠标的移动事件必须发送给远程计算机，把这些移动事件收集起来一批一批发送出去，使得鼠标的移动极不连贯





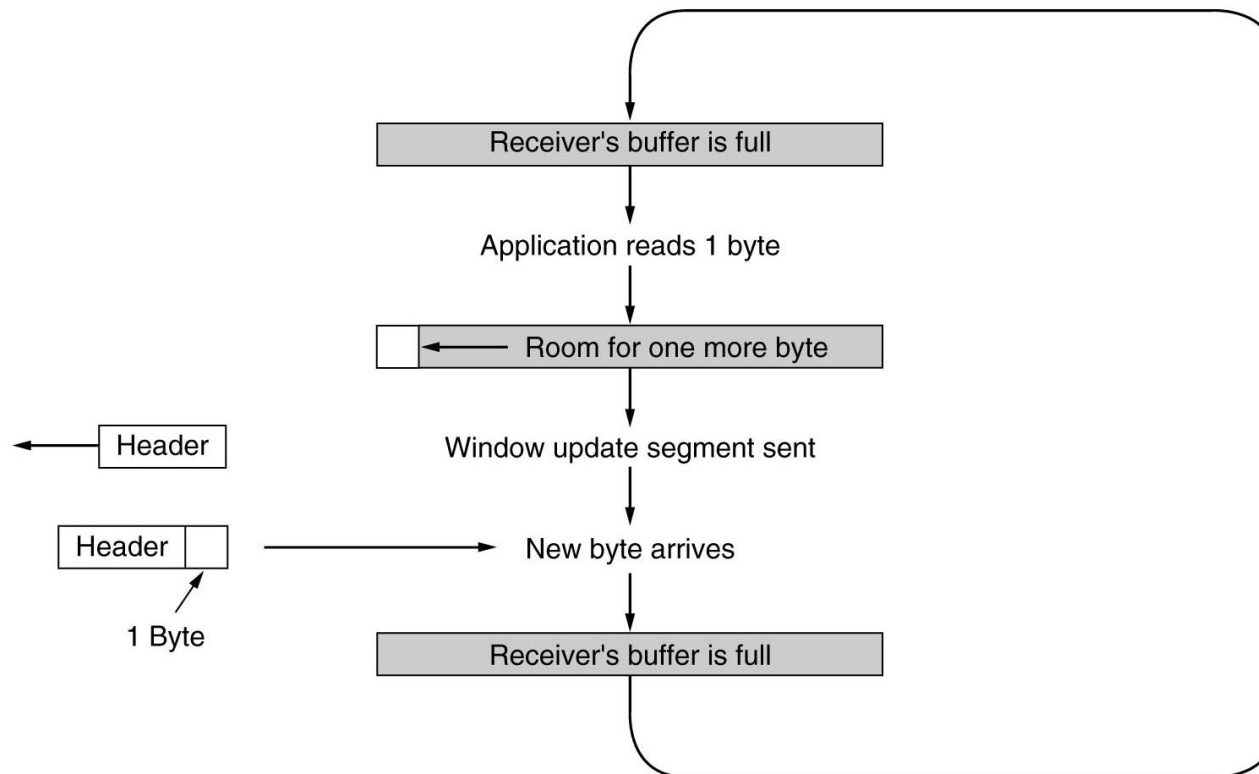
# Nagle's 算法图示



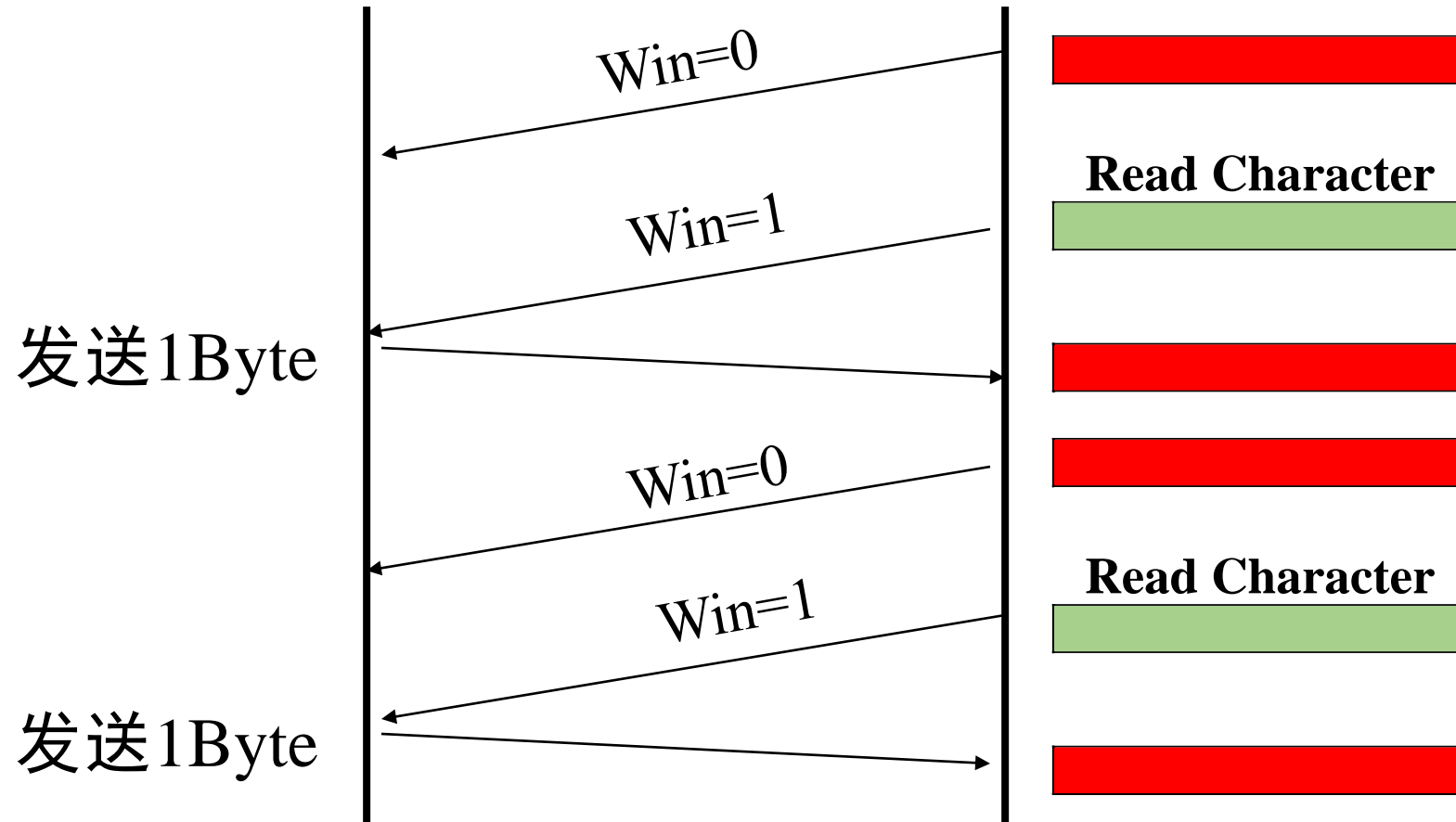


# 傻瓜窗口综合症

另一个使TCP性能退化的问题是傻瓜窗口综合症（**silly window syndrome problem**）：当有大块数据被传递给发送端TCP实体，但接收端的交互式应用每次只读取一个字节的时候，问题就来了



# 每次接收1字节

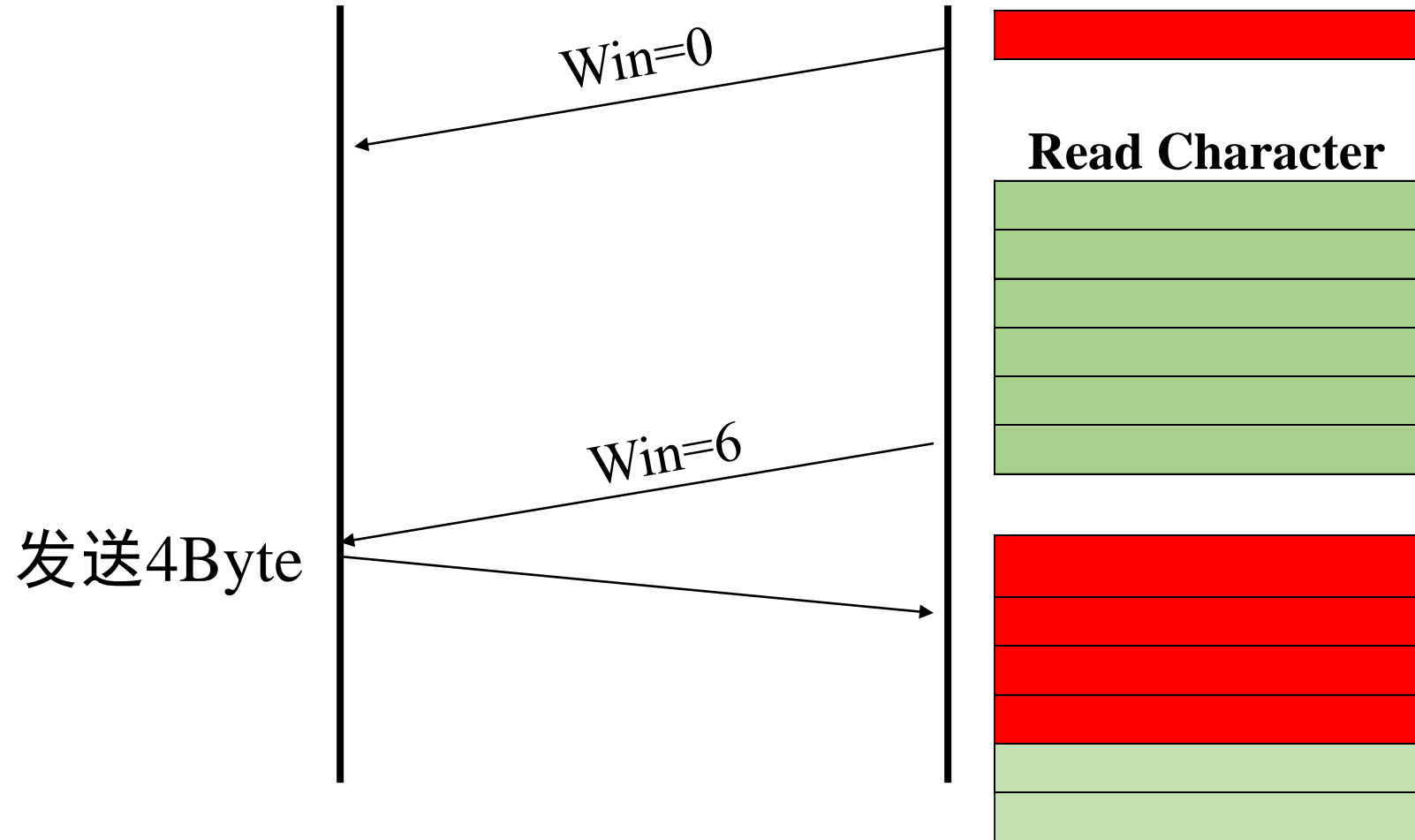




# 傻瓜窗口综合症

- **Clark解决方案**：阻止接收方发送只有1个字节的窗口更新，相反，它必须等待一段时间，当有了一定数量的空间之后再告诉发送方
- 接收方可以可以维护一个内部缓冲，且阻塞上层应用的 READ 请求，直到它有大块的数据提供

# Clark解决方案





# 发送方和接收方

## 发送方

(Nagle's algorithm)

- 尽量不发送数据含量小的数据段
- 缓存应用层的数据, 达到一定量再发送

## 接收方

(Clark's solution)

- 不请求对方发送短数据段(window size)
- 延迟窗口变更信息, 使接收缓冲区足够大



## 小结

- ❑ TCP传输的是全双工的字节流。
- ❑ TCP适配收发双方的数据流量
  - Window size
- ❑ TCP还需要效率
  - 发方优化: Nagle's algorithm
  - 收方优化: Clark's solution

# 思考题

- ❑ TCP怎样进行流控？
- ❑ 怎样提高TCP传输的效率？
- ❑ TCP的发方怎样优化？
- ❑ 傻瓜窗口综合症的症状是怎样的？
- ❑ TCP的收方怎样优化？



谢谢观看

# 致谢

本课程课件中的部分素材来自于：（1）清华大学出版社出版的翻译教材《计算机网络》（原著作者：Andrew S. Tanenbaum, David J. Wetherall）；（2）思科网络技术学院教程；（3）网络上搜到的其他资料。在此，对清华大学出版社、思科网络技术学院、人民邮电出版社、以及其它提供本课程引用资料的个人表示衷心的感谢！

对于本课程引用的素材，仅用于课程学习，如有任何问题，请与我们联系！