

案例学习之 TCP的流量控制



TCP的流量控制

流量控制：用于控制发送端的发送行为，以免快速发送端发送的数据将接收端淹没。流量控制仅与发送者和接收者之间的端-端通信有关。

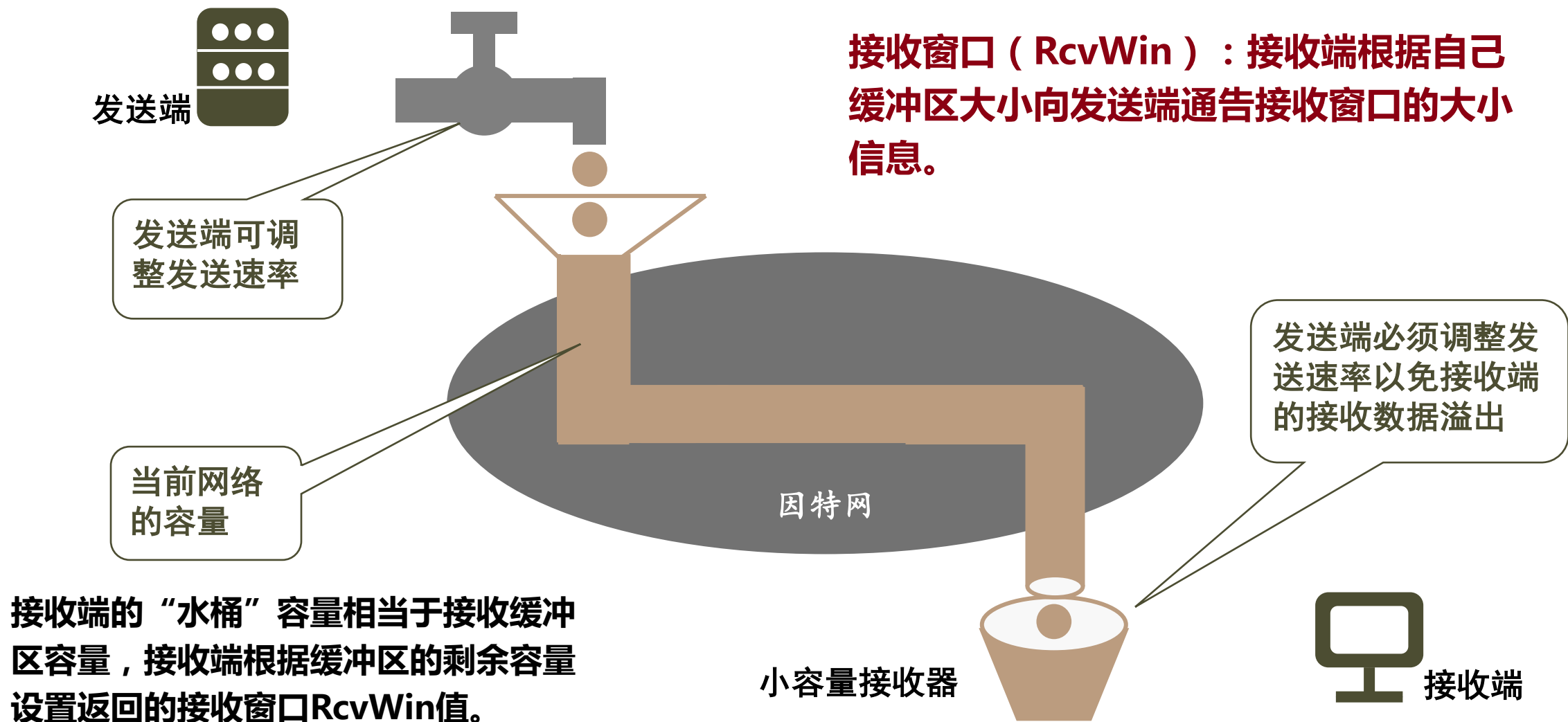
假设：在某个互联网中（无拥塞）

- 网络提供的数据速率为100Gbps
- 客户机是个人PC
- 文件服务器发送速率1Gbps

试问：如果服务器按全速率发送怎样？



流量控制的接收窗口



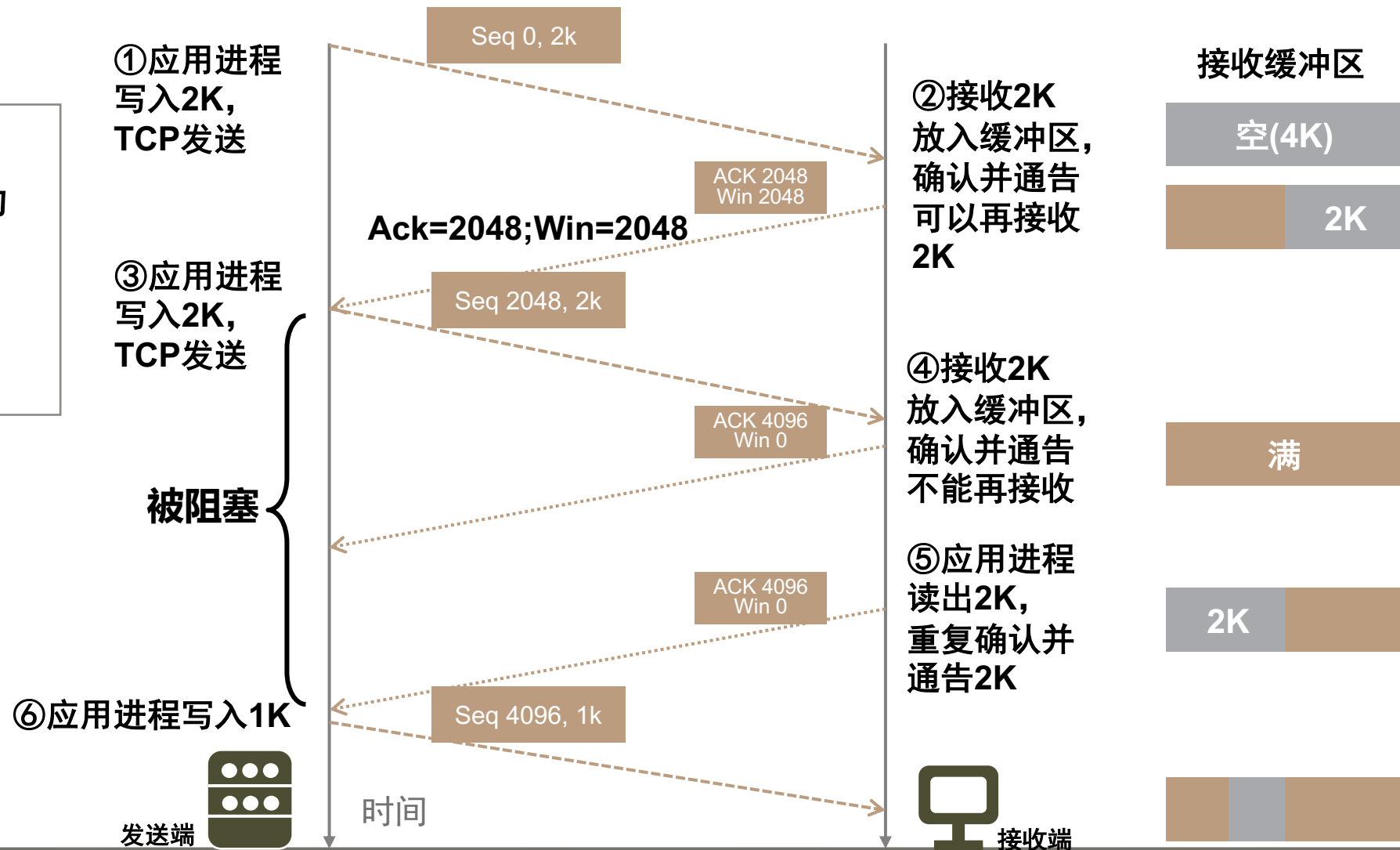
TCP的流量控制机制

TCP流量控制机制

- TCP采用大小可变的滑动窗口来控制发送速度
- 接收端通过window字段反馈当前可接收字节数

接收端返回的段

Source port#	dest. Port#
Sequence number	
Acknowledgement number	
	window

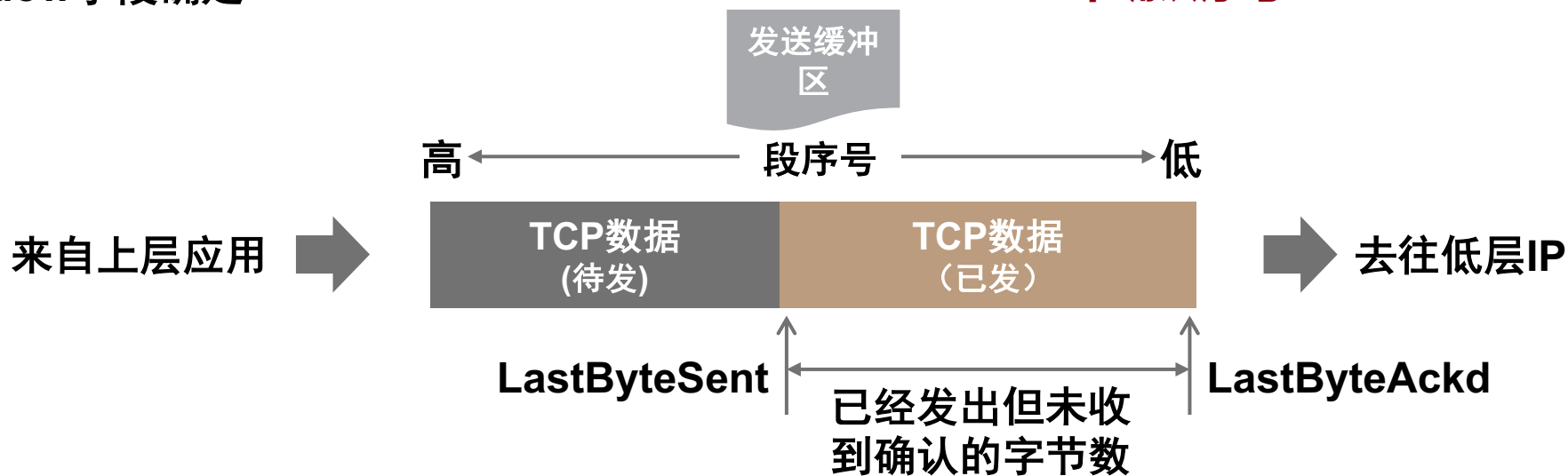


TCP的流量控制之发送端

TCP发送端维护一个变量“RcvWin”记录接收端能接收的字节数。

- 大小可变，由收到段的段头中window字段确定

- LastByteSent : 发送端发出的最后一个字节的序号
- LastByteAckd : 发送端从网络接收的最后一个确认序号



这两个变量满足什么关系才能达到流量控制?

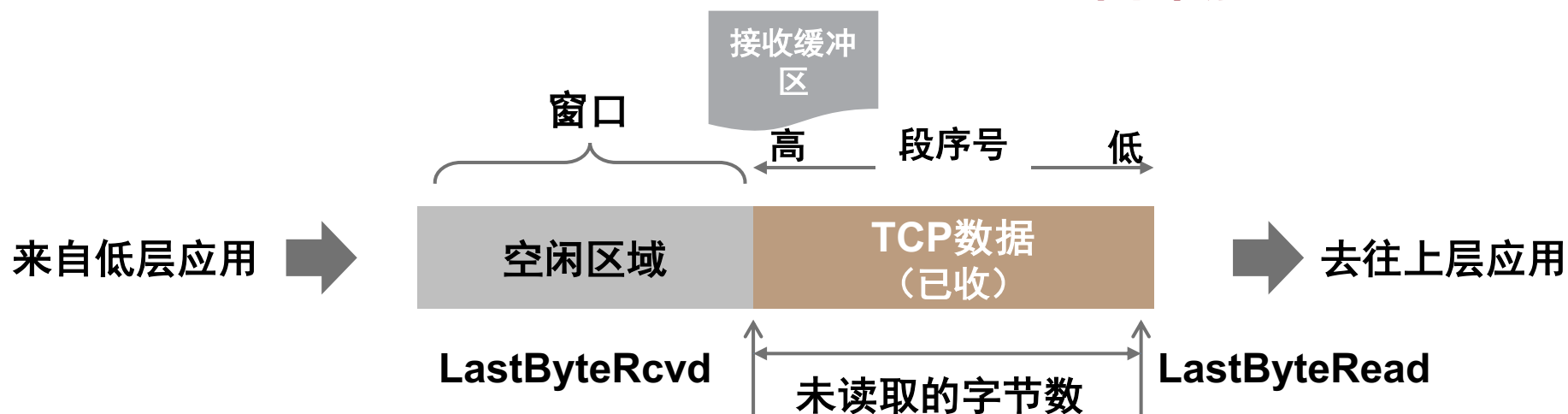
$$\text{LastByteSent} - \text{LastByteAckd} \leq \text{RcvWin}$$



TCP的流量控制接收端

TCP接收端通过段头部的“window”字段给发送端反馈可以接收的字节数。

- LastByteRead : 上层应用程序接收的最后一个字节序号
- LastByteRcvd : 接收端从网络接收的最后一个字节序号



?

发送端至多能
发送多少字节?

- 未读取的字节数 = $\text{LastByteRcvd} - \text{LastByteRead}$
- 空闲缓冲区大小 = 总缓冲区长度 - 未读取字节数



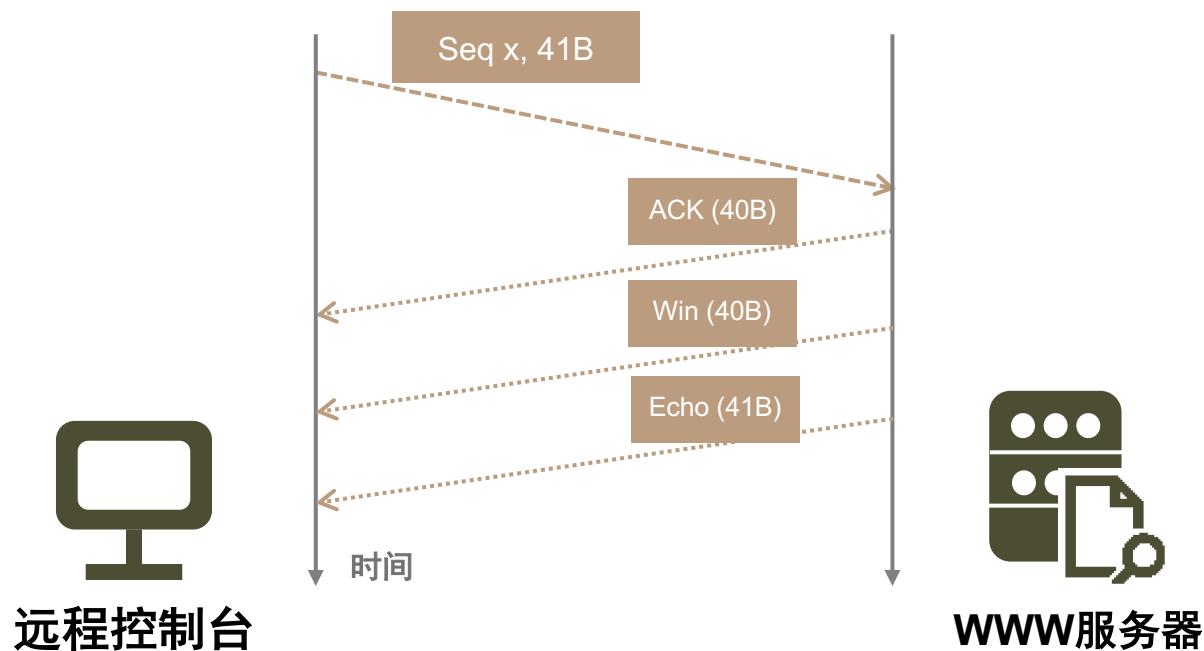
TCP传输性能问题一

考虑：telnet/login远程登陆网络应用

- ① 本地击键一次
- ② 远程主机确认
- ③ 远程主机反馈“窗口更新”
- ④ 远程主机回显字符

如果不做额外的处理，对于
每次击键输入的一个字符，
需发送4个段共162字节。

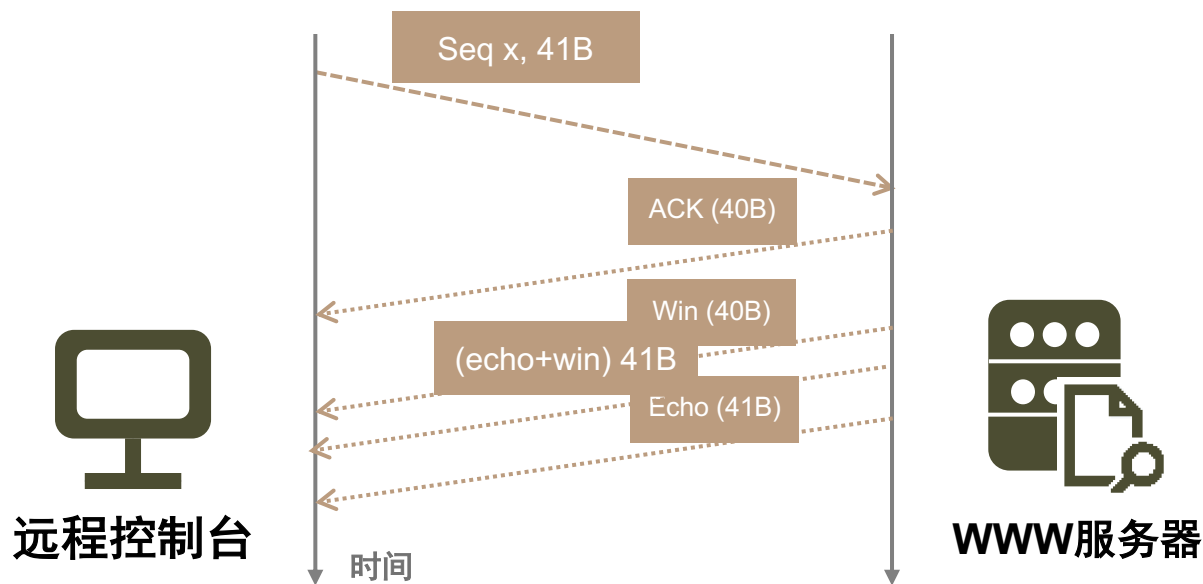
```
>telnet net.pku.edu.cn  
netlab>ipconfig
```



问题一的解决方案——Nagle

发送端优化方法

将“确认(ACK)”和“窗口更新(window)”段延迟一段时间以便搭载在后续其他数据中传输。



Nagle算法（针对发送端）

- 如果已传数据未确认之前发送端应用程序又生成了额外的数据，则照常将数据放入输出缓冲区，但并不发送
- 直到缓冲区中的数据足够填满一个MSS
- 如确认到达后发送端仍处于等待状态，则发送缓冲区中累积的所有数据

只有第一段是小报文，后续段因缓冲积攒多了一些数据，从而提高了发送端的数据传输效率。

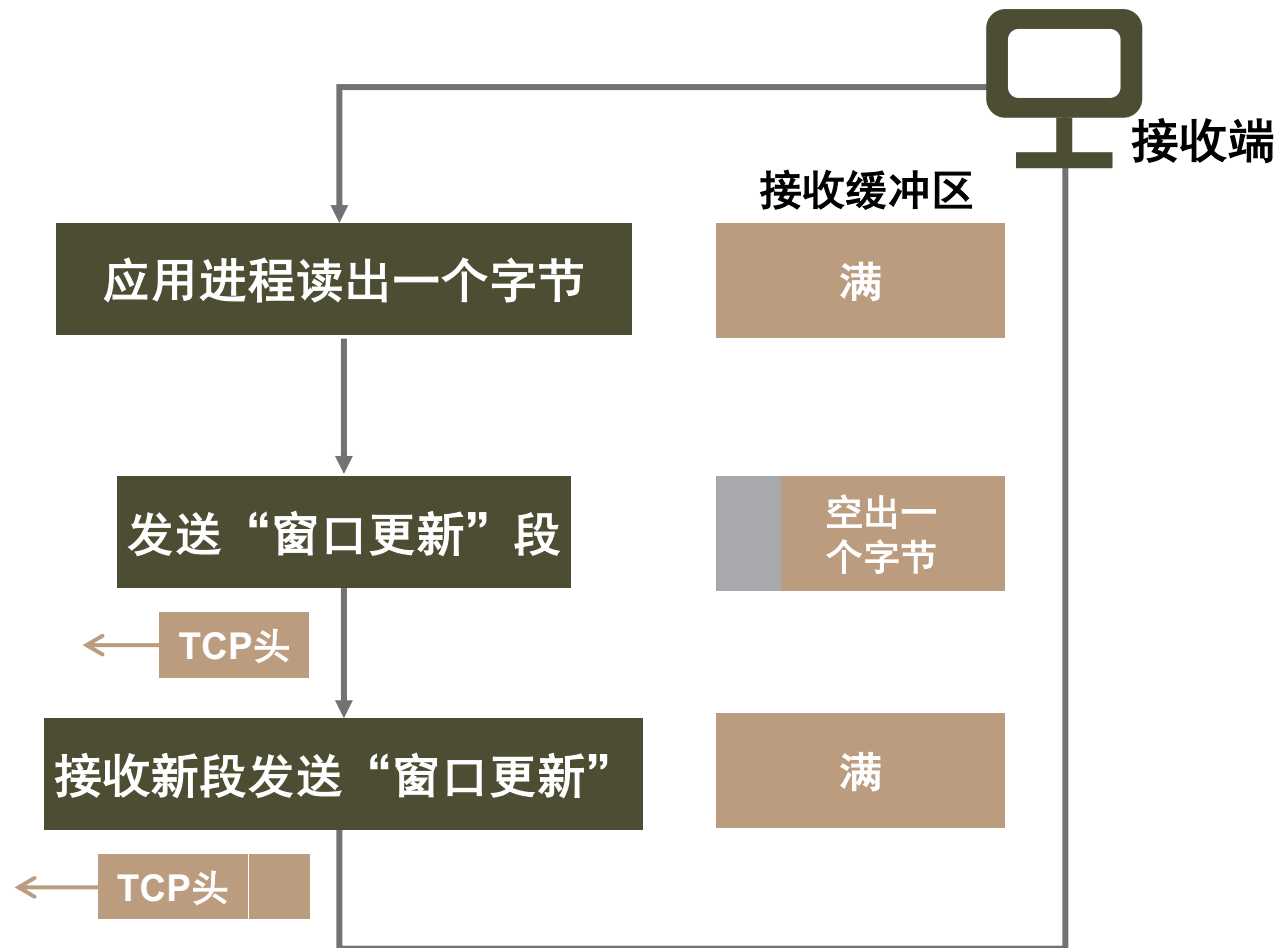


TCP传输性能问题二

“低能”窗口综合症

- 当接收端的应用程序每次仅读一个字节时
- 接收端通告一个小的可用窗口值将导致发送端产生短段
- 发送端的段仅含少量数据

后果：浪费网络带宽，增加计算开销



性能问题二解决方案——Clark

接收端优化方法

- TCP对收到的段进行确认，但要等到接收窗口可用空间达到启发式策略所指定的限度之后才发出“窗口增大”通告。
- 在接收窗口大小不够所指定的限度时推迟发送确认。

Clark方法（针对接收端）

- 通告大小为“0”的Window后，要等到接收缓冲区可用空间达到总空间的一半或MSS之后才发送“窗口更新”通告。

TCP规范说明：

- 发送端使用启发技术（Nagle）避免传输少量数据
- 接收端使用启发技术（Clark）防止反馈微小增量值的窗口通告

