



西北師範大學
NORTHWEST NORMAL UNIVERSITY

计算机网络

第五章 运输层 (3)

陈旺虎

chenwh@nwnu.edu.cn



第5章 运输层

5.1 运输层协议概述

5.2 运输层协议概述

5.3 传输控制协议 **TCP** 概述

5.4 可靠传输的工作原理

4.5 **TCP** 报文段的首部格式

5.6 **TCP** 可靠传输的实现

5.7 **TCP**的流量控制

5.8 **TCP**的拥塞控制

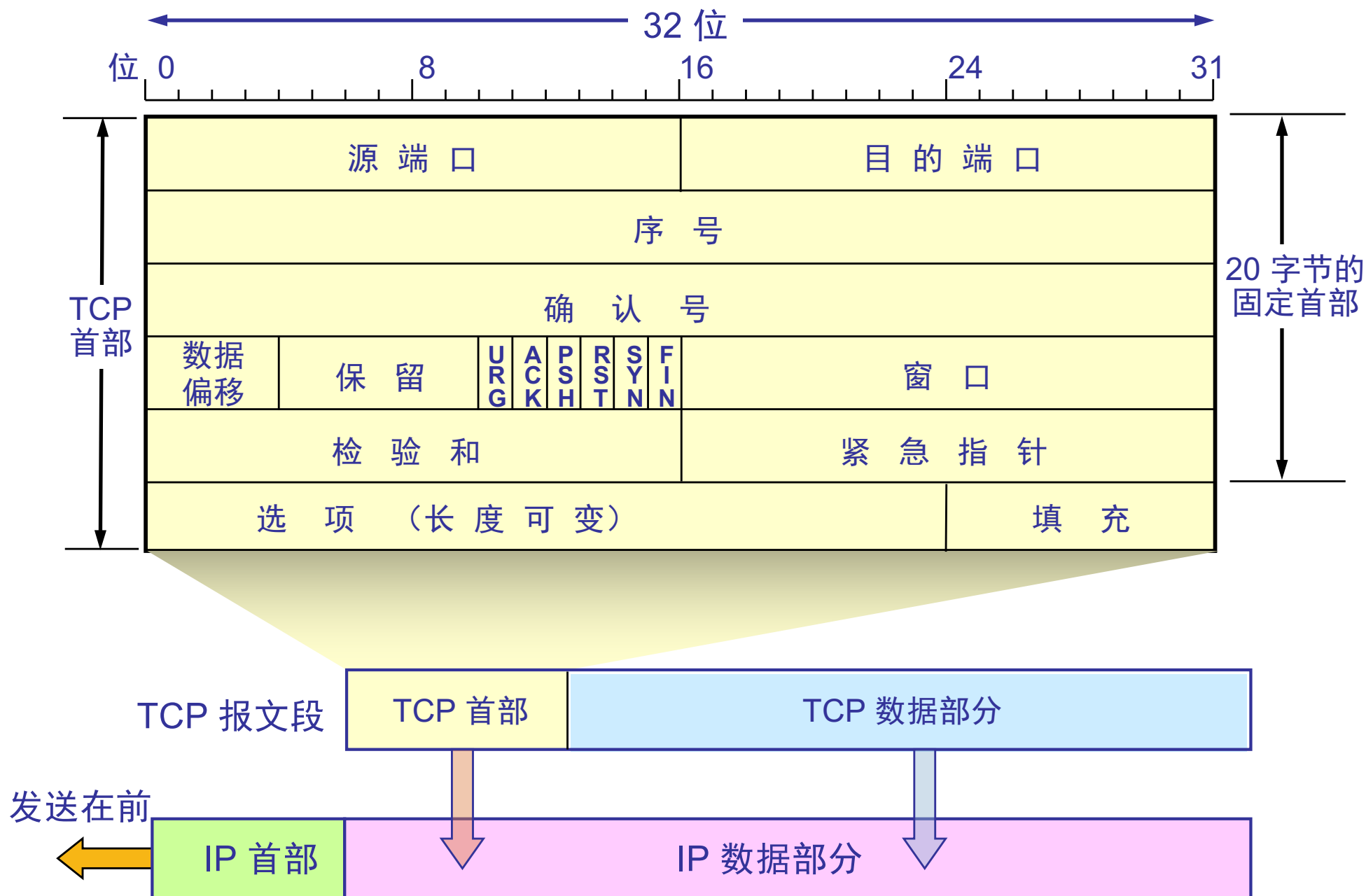
5.9 **TCP**的连接管理

Review



- 可靠传输的原理
- TCP协议的特点

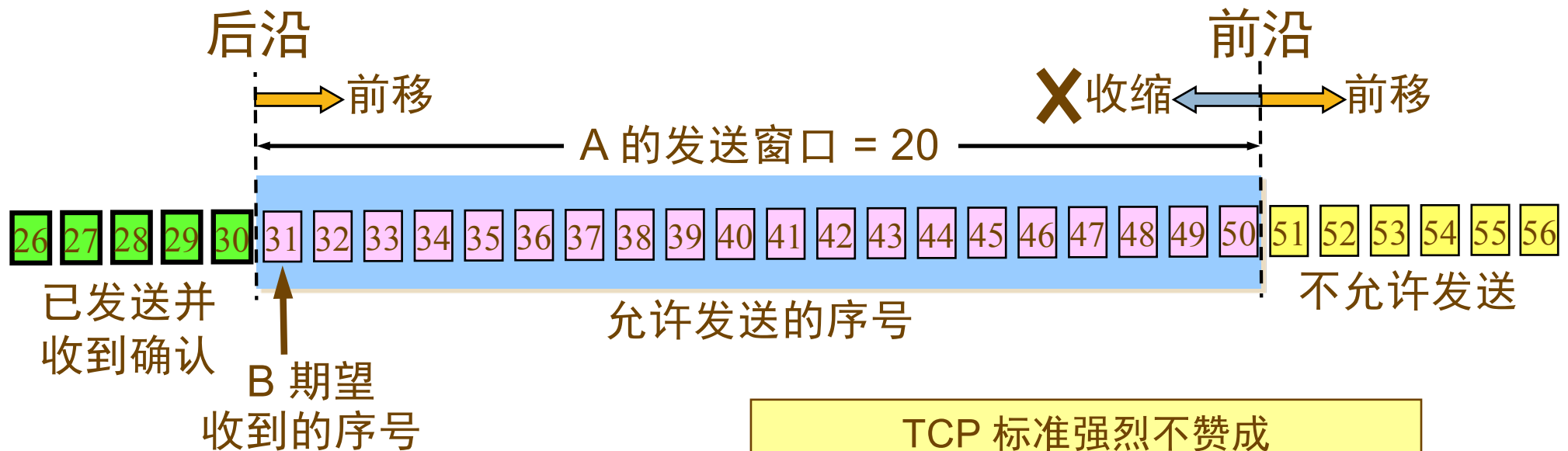
TCP 报文段的首部格式



5.6 TCP 可靠传输的实现

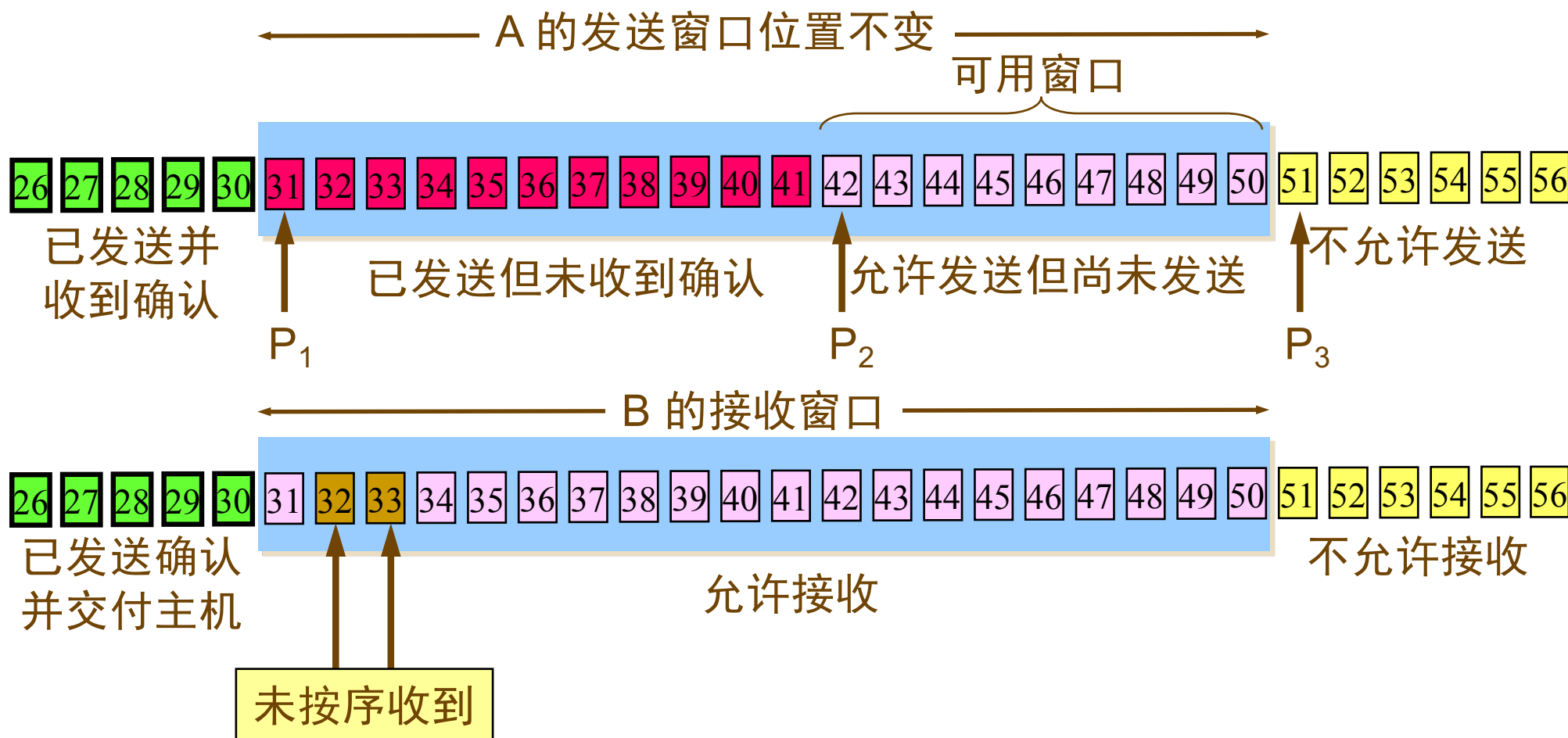
5.6.1 以字节为单位的滑动窗口

根据 B 给出的窗口值
A 构造出自己的发送窗口



TCP 标准强烈不赞成
发送窗口前沿向后收缩

A 发送了 11 个字节的数据

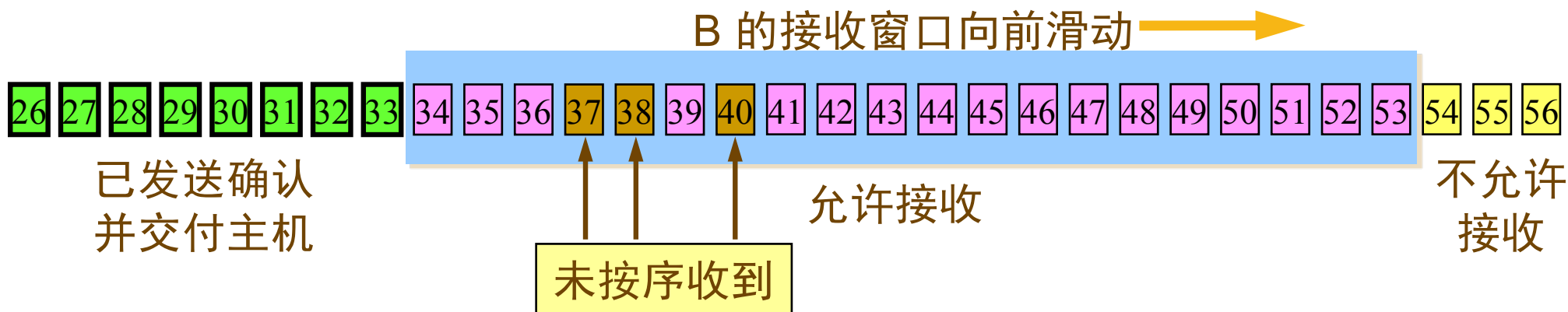
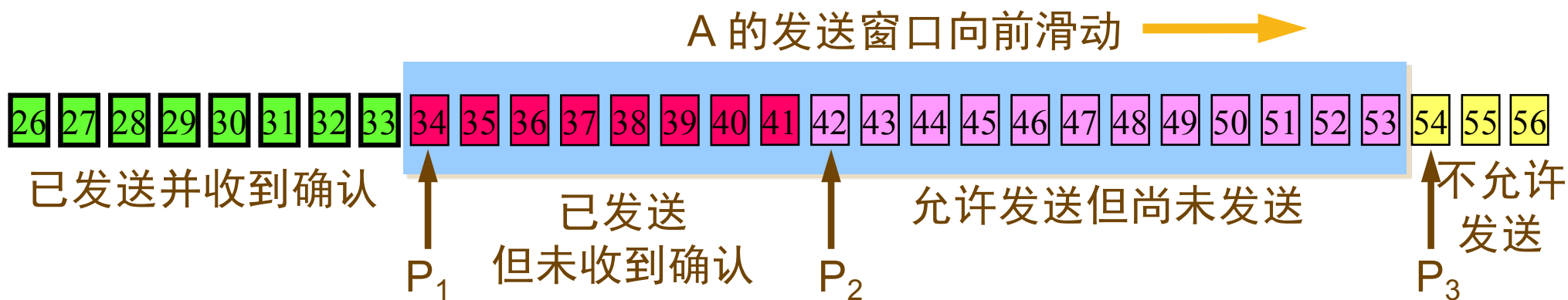


$P_3 - P_1 = A$ 的发送窗口（又称为通知窗口）

$P_2 - P_1 =$ 已发送但尚未收到确认的字节数

$P_3 - P_2 =$ 允许发送但尚未发送的字节数（又称为可用窗口）

A 收到新的确认号，发送窗口向前滑动



先存下，等待缺少的数据的到达

发送缓存与接收缓存的作用

- 发送缓存用来暂时存放：
 - ▣ 应用程序传送给发送方 TCP 准备发送的数据
 - ▣ TCP 已发送出但尚未收到确认的数据
- 接收缓存用来暂时存放：
 - ▣ 按序到达的、但尚未被接收应用程序读取的数据
 - ▣ 不按序到达的数据

需要强调三点

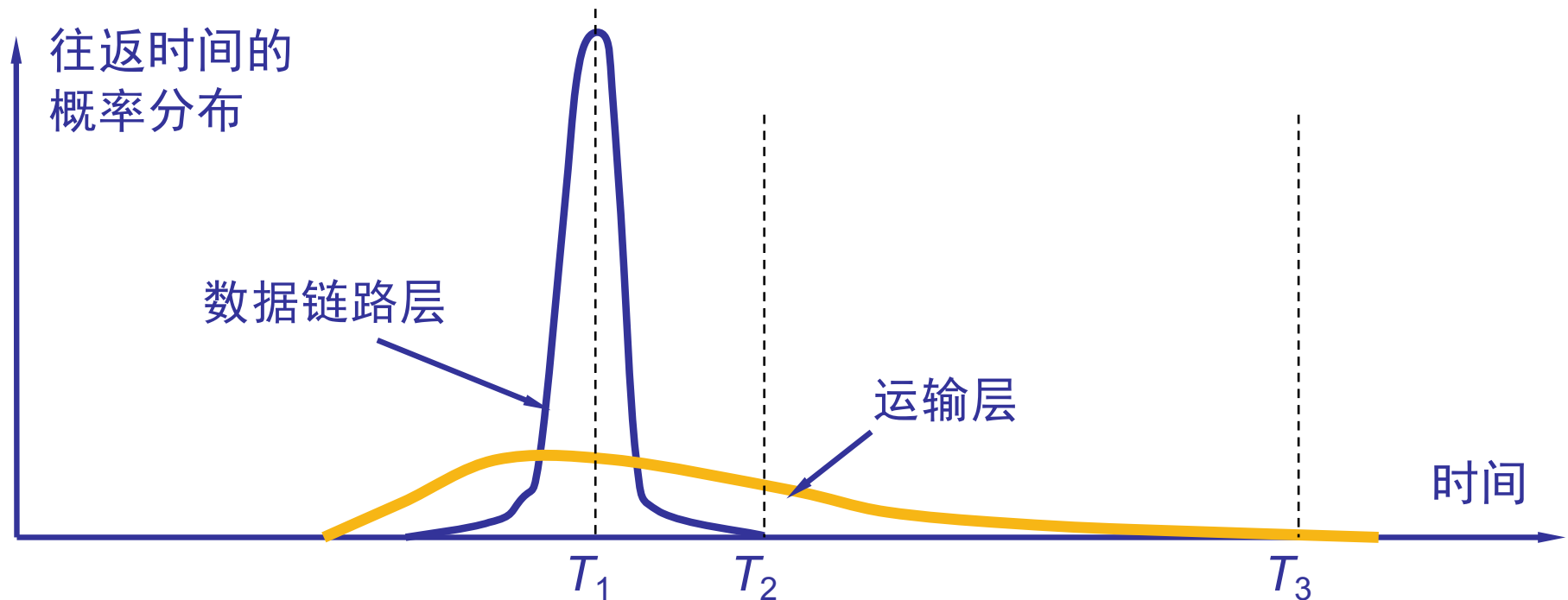
- A的发送窗口并不总是和B的接收窗口一样大（因为有一定的时间滞后）
- TCP 标准没有规定对不按序到达的数据应如何处理
 - ▣ 通常是先临时存放在接收窗口中，等到字节流中所缺少的字节收到后，再按序交付上层的应用进程
- TCP 要求接收方必须有累积确认的功能，这样可以减小传输开销

5.6.2 超时重传时间的选择

- 重传机制是 TCP 中最重要和最复杂的问题之一
 - ▣ TCP 每发送一个报文段，就对这个报文段设置一次计时器。
 - ▣ 只要计时器设置的重传时间到但还没有收到确认，就要重传这一报文段
- 超时重传时间如何选择？
- 超时重传时间与报文的往返时间有关（RTT）

往返时延的方差很大

- 由于 **TCP** 的下层是一个互联网环境，**IP** 数据报所选择的路由变化很大。因而运输层的往返时间的方差也很大。



加权平均往返时间

- 加权平均往返时间 RTT_S （又称为平滑的往返时间）。
- 第一次测量到 RTT 样本时， RTT_S 值就取为所测量到的 RTT 样本值。以后每测量到一个新的 RTT 样本，就按下式重新计算一次 RTT_S ：

$$\begin{aligned} \text{新的 } RTT_S &= (1 - \alpha) \times (\text{旧的 } RTT_S) \\ &\quad + \alpha \times (\text{新的 } RTT \text{ 样本}) \end{aligned}$$

- $0 \leq \alpha < 1$ 。若 α 很接近于零，表示 RTT 值更新较慢，越稳定。若选择 α 接近于 1，则表示 RTT 值更新较快。
- RFC 2988 推荐的 α 值为 $1/8$ ，即 0.125。

超时重传时间 RTO

- RTO (RetransmissionTime-Out)应略大于 RTT_S

RFC 2988 建议:

$$RTO = RTT_S + 4 \times RTT_D$$

- RTT_D 是 **RTT** 的偏差的加权平均值。

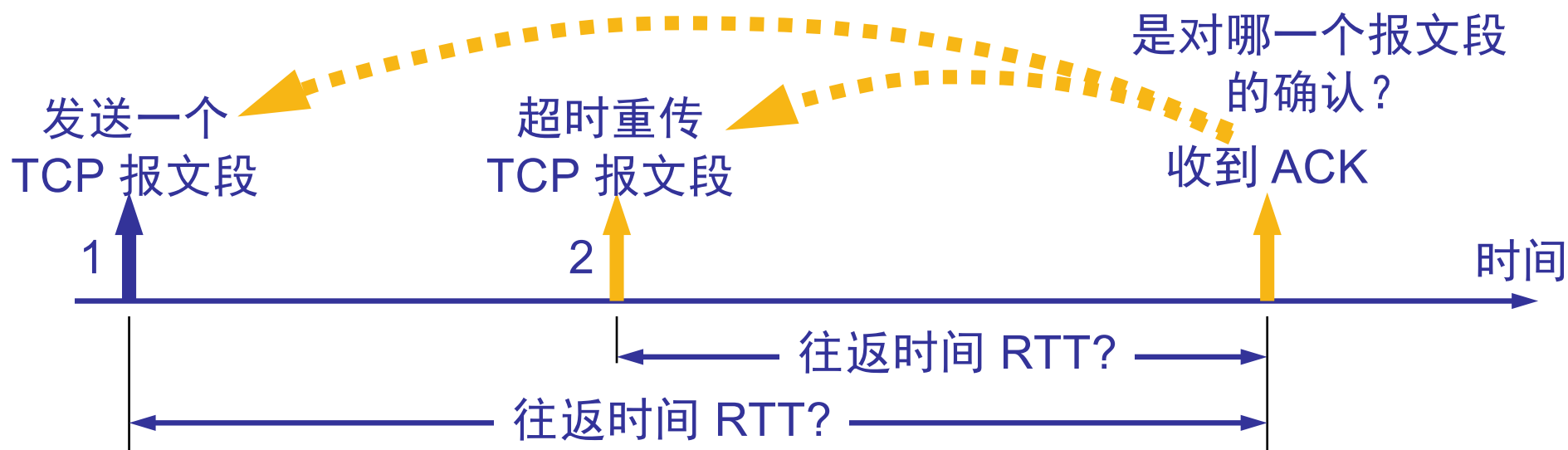
第一次测量时, RTT_D 值取为测量到的 RTT 样本值的一半; 在以后的测量中:

$$\begin{aligned} \text{新的 } RTT_D &= (1 - \beta) \times (\text{旧的 } RTT_D) \\ &\quad + \beta \times |RTT_S - \text{新的 } RTT \text{ 样本}| \end{aligned}$$

- β 是个小于 1 的系数, 其推荐值是 1/4, 即 0.25。

往返时间的测量相当复杂

- TCP 报文段 1 没有收到确认。重传后，收到了确认报文段 ACK。
- 如何判定此确认报文段是对原来的报文段 1 的确认，还是对重传的报文段 2 的确认？



Karn 算法



- 在计算平均往返时间 RTT 时，只要报文段重传了，就不采用其往返时间样本
- 这样得出的加权平均平均往返时间 RTT_S 和超时重传时间 RTO 就较准确

修正的 Karn 算法

- 报文段每重传一次，就把 RTO 增大一些

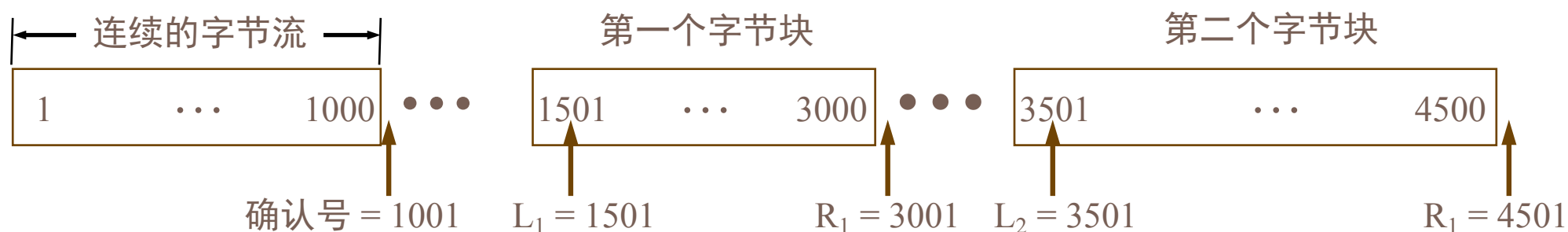
$$\text{新的 RTO} = \gamma \times (\text{旧的 RTO})$$

- 系数 γ 的典型值是 2。
- 当不再发生报文段的重传时
 - 根据报文段的往返时延更新平均往返时延 RTT 和超时重传时间 RTO 的数值
- 实践证明，这种策略较为合理

5.6.3 选择确认 (Selective ACK)

- 接收方收到了和前面的字节流不连续的两个字节块
- 如果这些字节的序号都在接收窗口之内，那么接收方就先收下这些数据，但要把这些信息准确地告诉发送方，使发送方不要再重复发送这些已收到的数据

接收到的字节流序号不连续



- 和前后字节不连续的每一个字节块都有两个边界：
左边界和右边界。
- 第一个字节块的左边界 $L_1 = 1501$ ，但右边界 $R_1 = 3001$ 。
- 第二个字节块的左边界 $L_2 = 3501$ ，而右边界 $R_2 = 4501$ 。

RFC 2018的规定

- 如果要使用选择确认，那么在建立 **TCP** 连接时，就要在 **TCP** 首部的选项^{选项}中加上“允许 **SACK**”的选项
- 如果使用选择确认，那么原来首部中的“确认号字段”的用法仍然不变。
- 增加 **SACK** 选项，以便报告收到的不连续的字节块的边界
- **TCP**首部选项的长度最多只有 **40** 字节，指明一个边界要用 **4** 字节，因此最多只能指明 **4**个字节块的边界信息

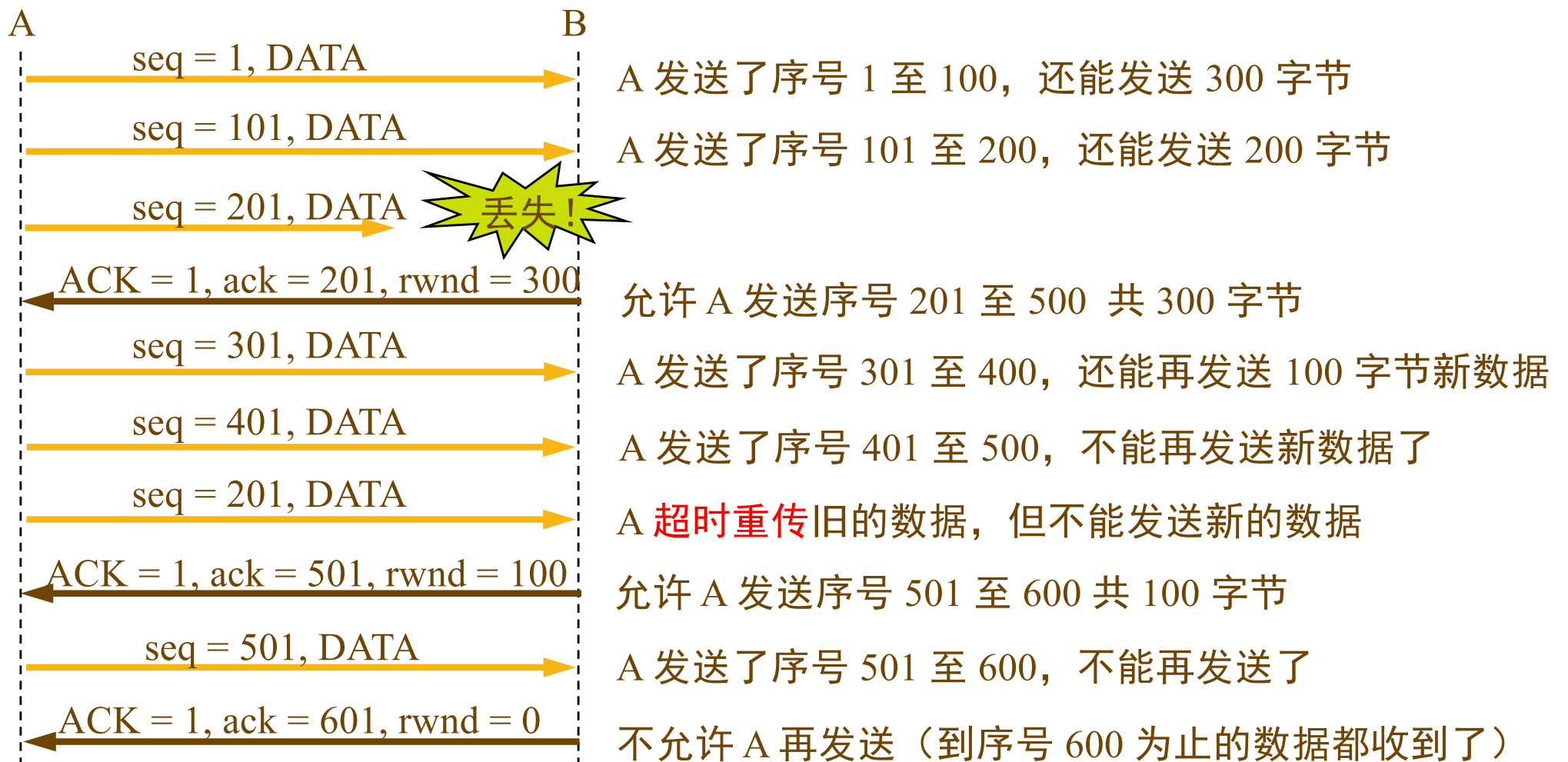
5.7 TCP的流量控制

5.7.1 利用滑动窗口实现流量控制

- 一般说来，我们总是希望数据传输得更快一些
 - ▣ 但是，如果发送方数据发送过快，接收方就可能来不及接收，造成数据的丢失。
- 流量控制(flow control)
 - ▣ 就是控制发送方的发送速率，让接收方来得及接收
 - ▣ 利用滑动窗口机制可以很方便地在 TCP 连接上实现流量控制

流量控制举例

A 向 B 发送数据。在连接建立时，B 告诉 A：“我的接收窗口 $rwnd = 400$ （字节）”。



0窗口死锁

- 如果B发送rwnd=0的报文段，则A收到后停止数据发送
- 假设B有了新的存储空间，并发送rwnd=400的报文段，但该报文丢失
- 则双方出现了相互等待：
 - ▣ A等待B发送非零窗口通知
 - ▣ B等待A发送数据
- 也就是说，出现了0窗口死锁

持续计时器 (persistence timer)

- 解决解决0窗口死锁
 - ▣ TCP 为每一个连接设有一个持续计时器。
 - ▣ 只要 TCP 连接的一方收到对方的零窗口通知，就启动持续计时器
 - ▣ 若持续计时器设置的时间到期，就发送一个零窗口探测报文段（仅携带 1 字节的数据）
 - ▣ 对方在确认探测报文段时，给出现在的窗口值
 - ▣ 若窗口仍然是零，则收到这个报文段的一方就重新设置持续计时器。
 - ▣ 若窗口不是零，则死锁的僵局就可以打破了

5.7.2 必须考虑传输效率

- TCP可以用不同的机制来控制报文段的发送时机:
- 第一种机制：缓存数据量
 - ▣ 缓存中存放的数据达到MSS 字节时，就组装成一个TCP 报文段发送出去
- 第二种机制：推送
 - ▣ 由发送方的应用进程指明要求发送报文段，即 TCP 支持的推送(push)操作
- 第三种机制：计时器
 - ▣ 发送方的一个计时器期限到了，把当前已有的缓存数据装入报文段（但长度不能超过 MSS）发送出去

面临的问题

- (1) 控制报文段的发送时机是非常复杂的
 - ▣ 交互式应用中，报文中有效数据量过小，发送的报文段和确认报文段远大于有效数据量，导致传输效率很低
 - ▣ 应尽量推迟发回确认，并尽量使用捎带确认
- Nagle算法
 - ▣ 发送应用程序要发送的第1个字节
 - ▣ 缓存后续的数据
 - ▣ 收到对前一个报文段的确认后，发送下一个报文段
 - ▣ 当到达的数据达到发送窗口的一半，或者达到MSS的长度时，发送数据

□ (2) 糊涂窗口综合症

- ▣ 交互式应用中，每次交付非常少的字节，同样导致传输效率过低

▣ 例如：

- 接收缓存满
- 交付一个报文段，只有一个字节
- 确认，通知接收窗口的大小为1字节

- ▣ 解决：接收方等待（**MSS**或者接受缓存一半空闲），然后发送确认报文

练习

- 33. 在OSI参考模型中，自下而上第一个提供端到端服务的层次是
 - ▣ A. 数据链路层
 - ▣ B. 传输层
 - ▣ C. 会话层
 - ▣ D. 应用层

- 35. 数据链路层采用了后退N帧（GBN）协议，发送方已经发送了编号为0~7的帧。当计时器超时时，若发送方只收到0、2、3号帧的确认，则发送方需要重发的帧数是
 - ▣ A. 2 B.3 C.4 D.5

- 36. 以太网交换机进行转发决策时使用的PDU地址是
 - ▣ A. 目的物理地址 B. 目的IP地址
 - ▣ C. 源物理地址 D. 源IP地址

□ 38. 主机甲和主机乙间已建立一个TCP连接，主机甲向主机乙发送了两个连续的TCP段，分别包含300字节和500字节的有效载荷，第一个段的序列号为200，主机乙正确接收到两个段后，发送给主机甲的确认序列号是

□ A. 500 B. 700 C. 800 D. 1000

Thanks, Any Question?

The End.

chenwh@nwnu.edu.cn

