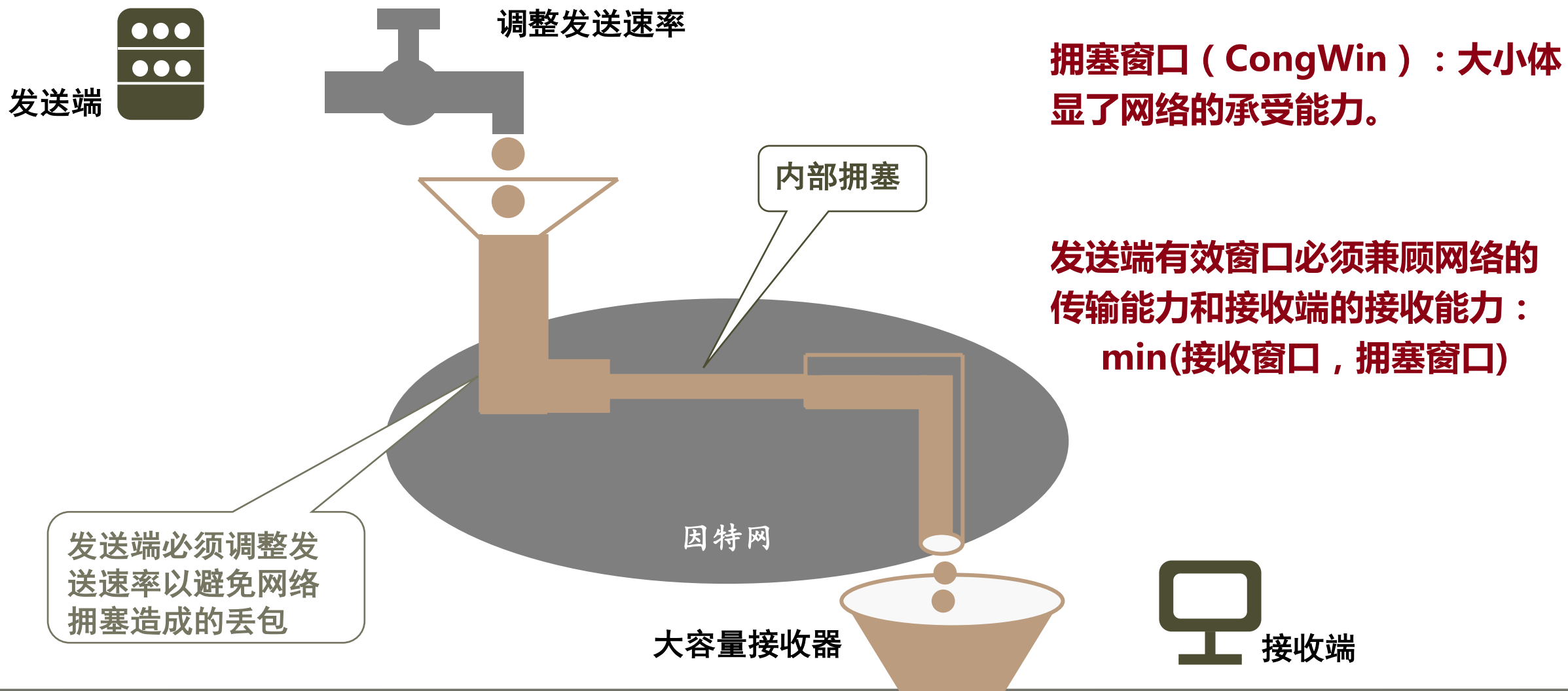


# 案例学习之 TCP的拥塞控制机制



# 拥塞控制的拥塞窗口



# TCP拥塞控制的基本思想

**TCP采用端-端拥塞控制机制：限制发送端通过连接注入网络的流量来缓解拥塞。**

- 连接两端TCP实体以多种方式监测传输路径是否出现拥塞
- 一旦监测到拥塞立即骤降发送速率
- 拥塞缓解后再逐步增加发送速率



Q1

TCP发送端如何知晓端-端路径上是否发生拥塞？

Q2

TCP发送端如何限制发往连接上数据的速率？

Q3

TCP发送端采用何种算法改变数据的发送速率？



# TCP如何知晓路径上发生拥塞？

## 发生“丢包事件”

- 段的计时器超时
- 收到三个重复ACK

## 路由器显式通知

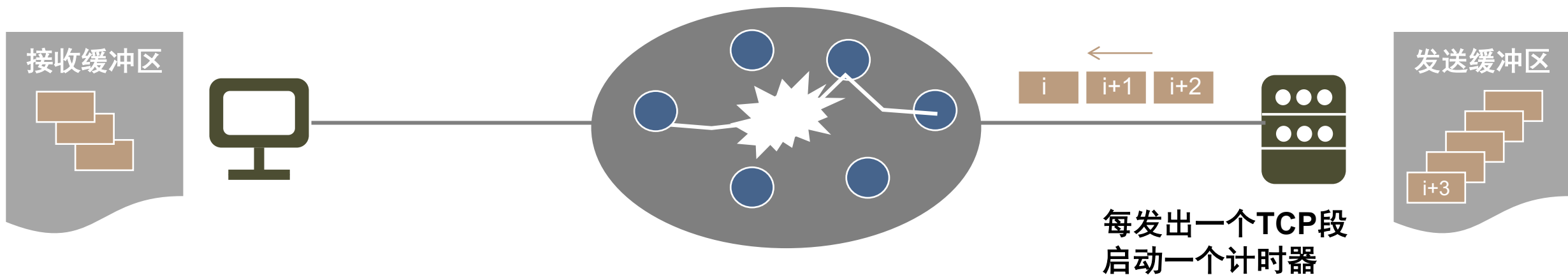
- 路由器监测到出现拥塞后通知连接一端

Q1

TCP发送端如何知晓端-端路径上是否发生拥塞？



# 何谓“丢包”？



- ① 在发送端和接收端之间路径上一个路由器的缓冲区溢出，数据报(携带TCP段*i*)被丢弃。

路由器缓冲区耗尽，TCP段*i*, *i*+1, *i*+2...都将被丢弃

- ② 在计时器超时前收到三个重复ACK（确认该TCP段之前的段）

由于传输错误TCP段*i*被丢弃，但TCP段*i*+1, *i*+2...都到达目的地

在计时器超时前没有收到接收端反馈的确认，即可认为相应的TCP段已经被丢弃

接收端每收到一个正确TCP段发送重复确认，发送端收到3个重复确认则启动快速重传机制

注意：图中为叙述简便，每个TCP段的编号递增1



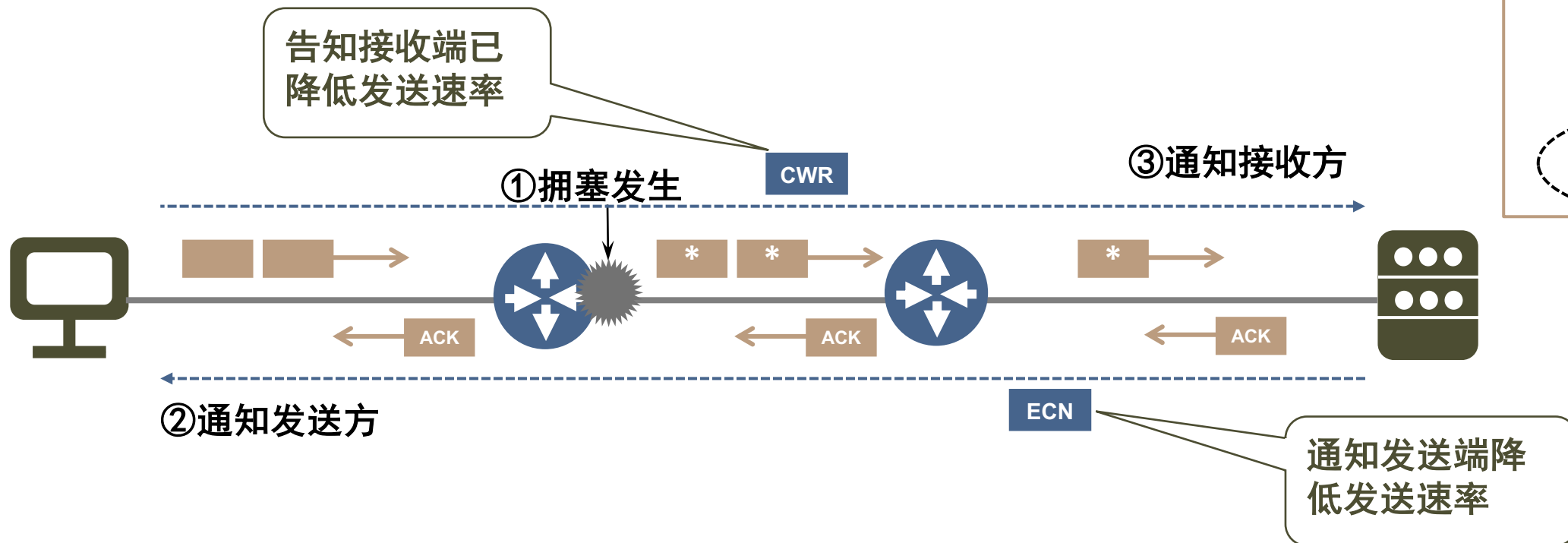
# 路由器显式通知

- 路由器：当检测到发生拥塞时在包头设置一个特殊比特
- 接收端：拷贝并“捎带”该特殊比特在ACK中发给发送端
- 发送端：监测带有ECN比特的ACK数量并据此调整传输速率

RFC3168

TCP的控制比特

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



# 路由器的拥塞检测方法

路由器通过监测出境线路和其他资源的使用情况来推测发生拥塞的可能性。

- 每个路由器周期性地对出境线路的瞬间利用率进行取样 $f$ ，求得该条线路的利用率 $u$
- 一旦 $u$ 超过了某个给定阈值，则该条出境线路就进入“警告状态”
- 每新到达一个包，路由器检查其出境线路的状态
- 如果该条线路处于“警告状态”，就意味着有可能发生拥塞

假设线路的利用率用实型变量 $u$ 表示， $u$ 的取值范围在0.0~1.0之间

$$u_{\text{new}} = a * u_{\text{old}} + (1 - a) * f$$

$$a = \begin{cases} 0 & u_{\text{new}} = f \\ 1 & u_{\text{new}} = u_{\text{old}} \end{cases}$$

- $a$ 为常数，7/8是经验值
- $f$ 为瞬间采样值

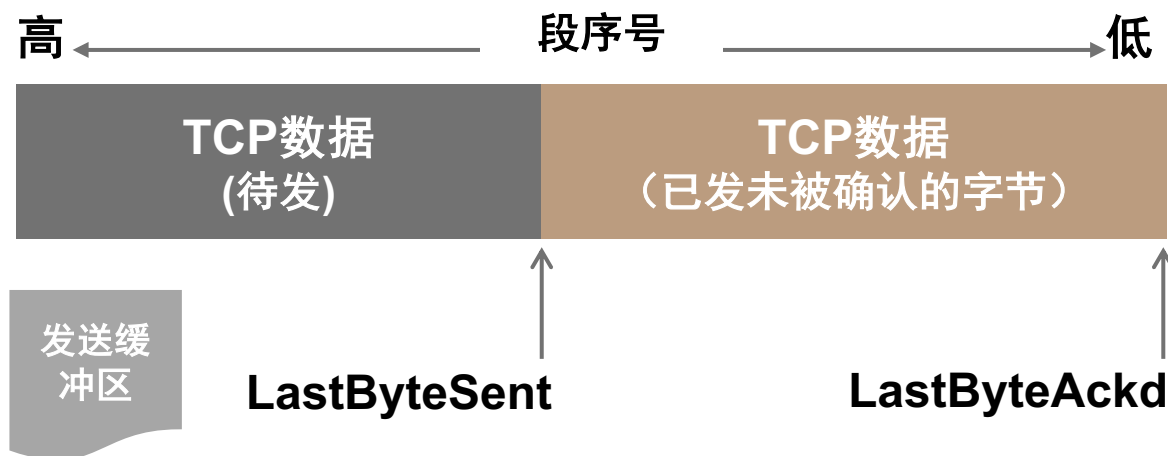
当路由器检测出线路利用率接近饱和时，通过显式方式通知发送端降低发送速率。



# TCP如何限制发送速率?

- CongWin : 控制TCP发送速率的拥塞窗口
- RcvWin : 接收端通报可以接收的字节数

有效窗口 : 必须同时兼顾网络和接收端能力  
 $\min(\text{接收窗口}, \text{拥塞窗口})$



$$\text{LastByteSent} - \text{LastByteAckd} \leq \min\{\text{CongWin}, \text{RcvWin}\}$$

Q2

TCP发送端如何限制发往连接上数据的速率?

只要控制拥塞窗口的大小就可以限制发送端的数据发送速率。

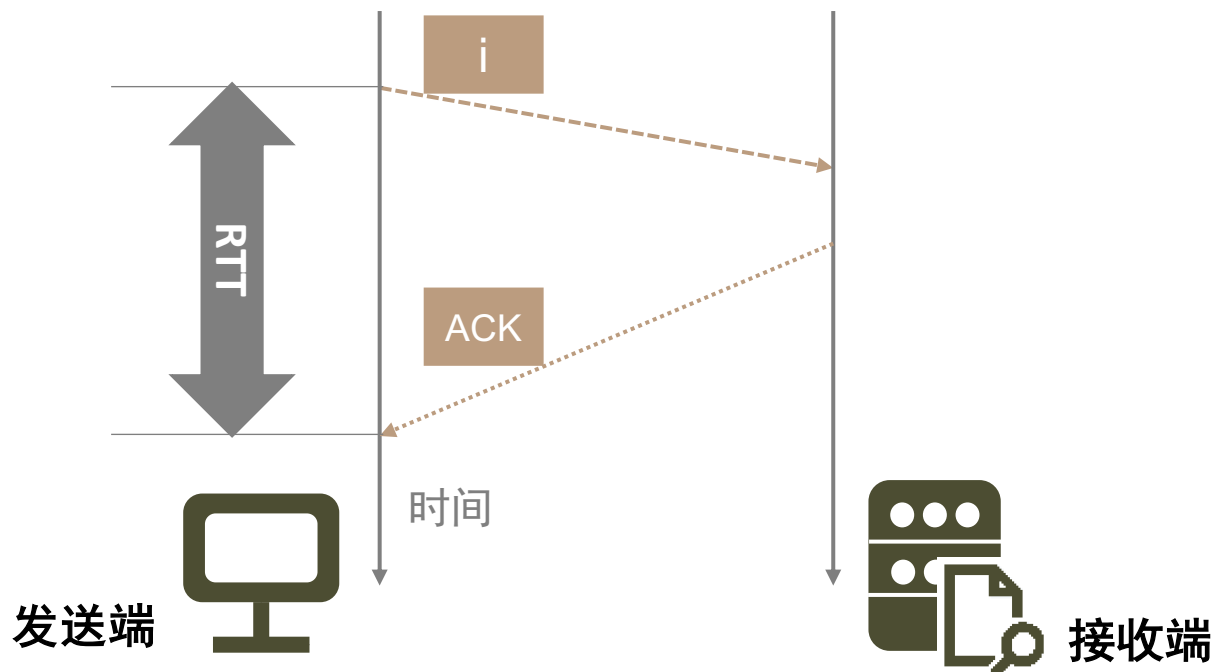




# 限制发送速率的两大参数

- CongWin : 控制TCP发送速率的拥塞窗口
- RcvWin : 接收端通报可以接收的字节数
- 有效窗口 :  $\min(\text{接收窗口}, \text{拥塞窗口})$

**RTT : TCP段开始发送到该段ACK返回的时间**



## 两大关键参数

- TCP发送端发送速率近似为  $\text{CongWin}/\text{RTT bps}$
- 发送端通过调整CongWin来限制发送速率

Q3

TCP发送端采用何种算法改变数据的发送速率？

