

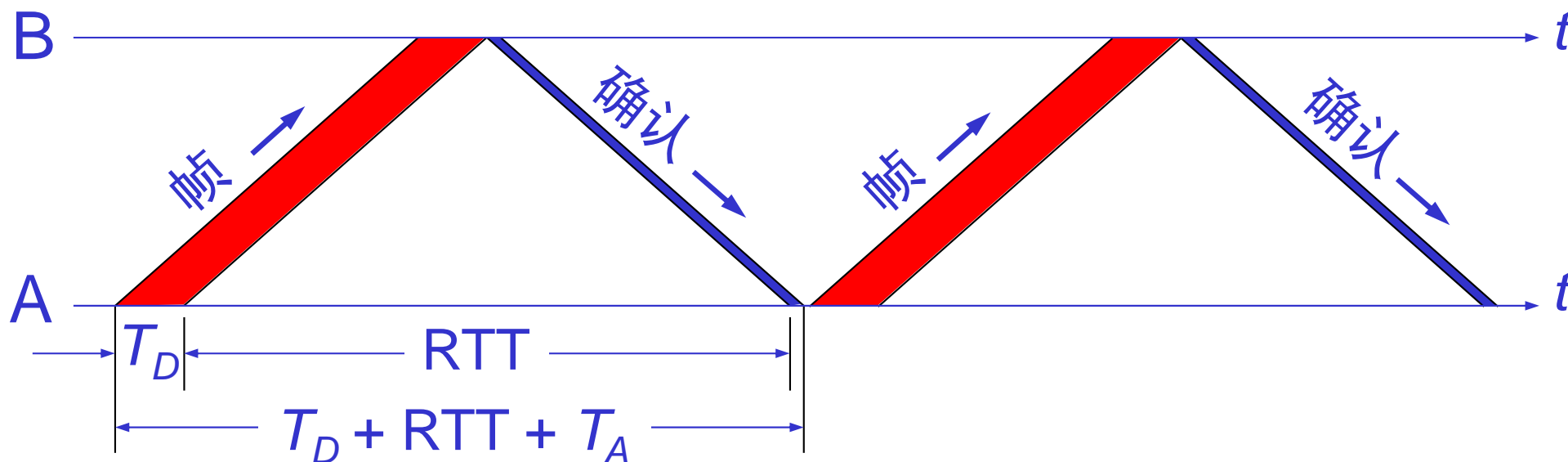
## 3.4 滑动窗口协议

## 一、简介

- 传输链路存在时延，而**ARQ**协议在同一时刻仅有一个帧在链路上传输(数据帧或确认帧)，其对信道的利用率较低
  - 请回忆第1章介绍的时延带宽积概念
- **ARQ**协议的信道利用率过低(尤其当传输时延较长时)

$$U = \frac{T_D}{T_D + RTT + T_A}$$

$T_D$ : 发送数据帧的时间  
 $RTT$ : 往返时延/环路时延  
 $T_A$ : 发送应答帧的时间



## 3.4 滑动窗口协议

---

### 二、滑动窗口协议原理(sliding window protocol)

- 滑动窗口协议的基本思想
  - 允许发送方连续发送多个帧
  - 通过滑动窗口实现流量控制
    - 每个待发送的帧都有一个序列号
    - 发送方维护一个发送窗口，它包含一组序列号，对应允许它发送的帧
    - 接收方维护一个接收窗口，对应允许它接收的帧
- 问题：发送窗口和接收窗口的宽度如何确定？
  - 接收窗口宽度：
  - 发送窗口宽度：

## 3.4 滑动窗口协议

### 二、滑动窗口协议原理

- 发送方

- 发送窗口内的序列号代表允许它发送的帧

- 窗口内最大的序列号称为窗口上边界，或窗口上沿、前沿
    - 窗口内最小的序列号称为窗口下边界，或窗口下沿、后沿

- 每当从网络层得到一个数据包，将其组成帧发出后，发送窗口的上边界+1

- 发送窗口下边界的帧被接收方确认后，下边界+1

- 接收方

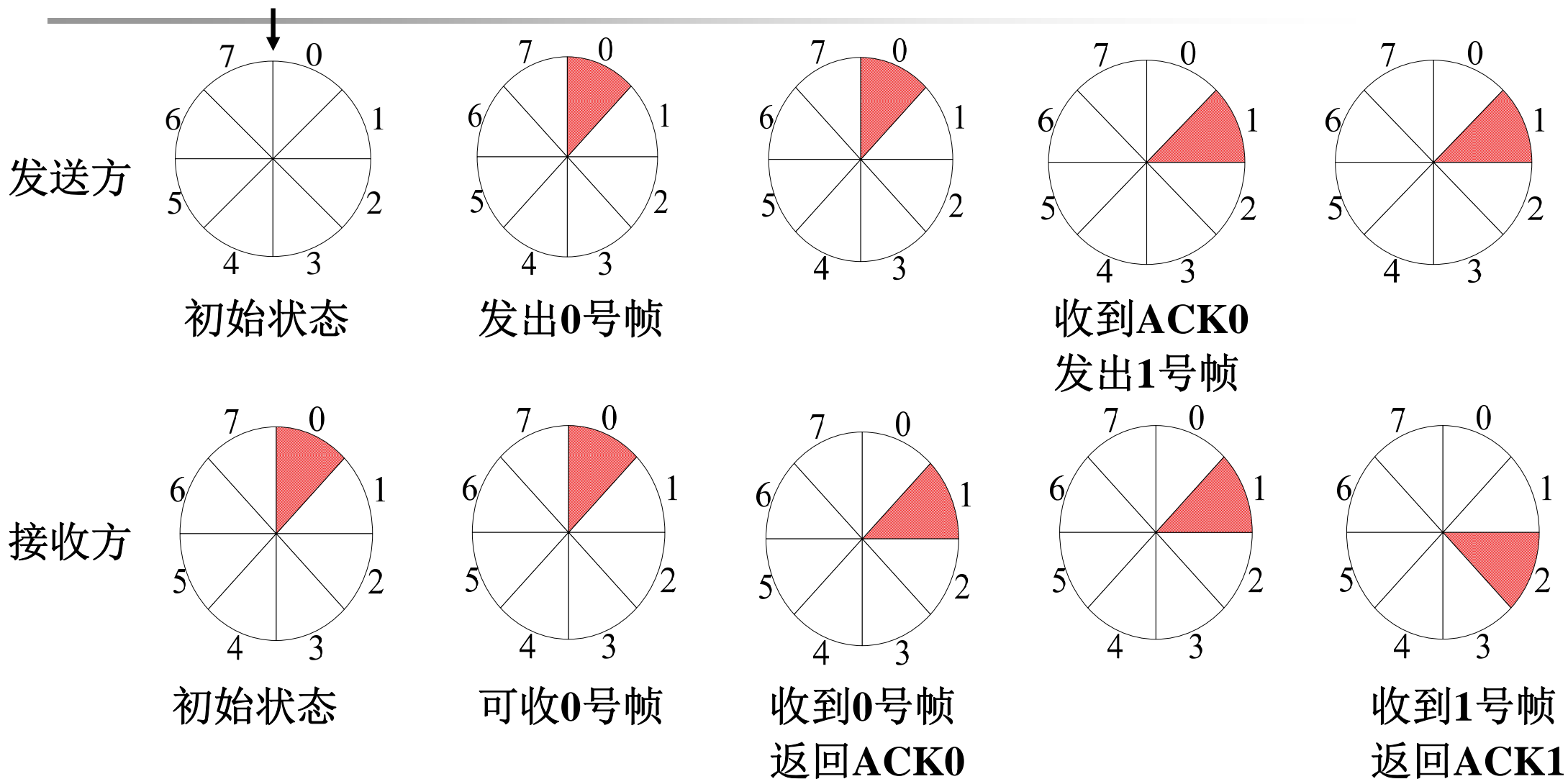
- 接收窗口内的序列号代表它可以接收的帧

- 收到的帧序列号等于窗口下边界时，将该帧上交网络层，并返回确认帧，同时整个窗口向前移动1个位置

- 如果收到帧序列号落在接收窗口之外，则将其丢弃

- 注意：接收窗口总是保持固定大小

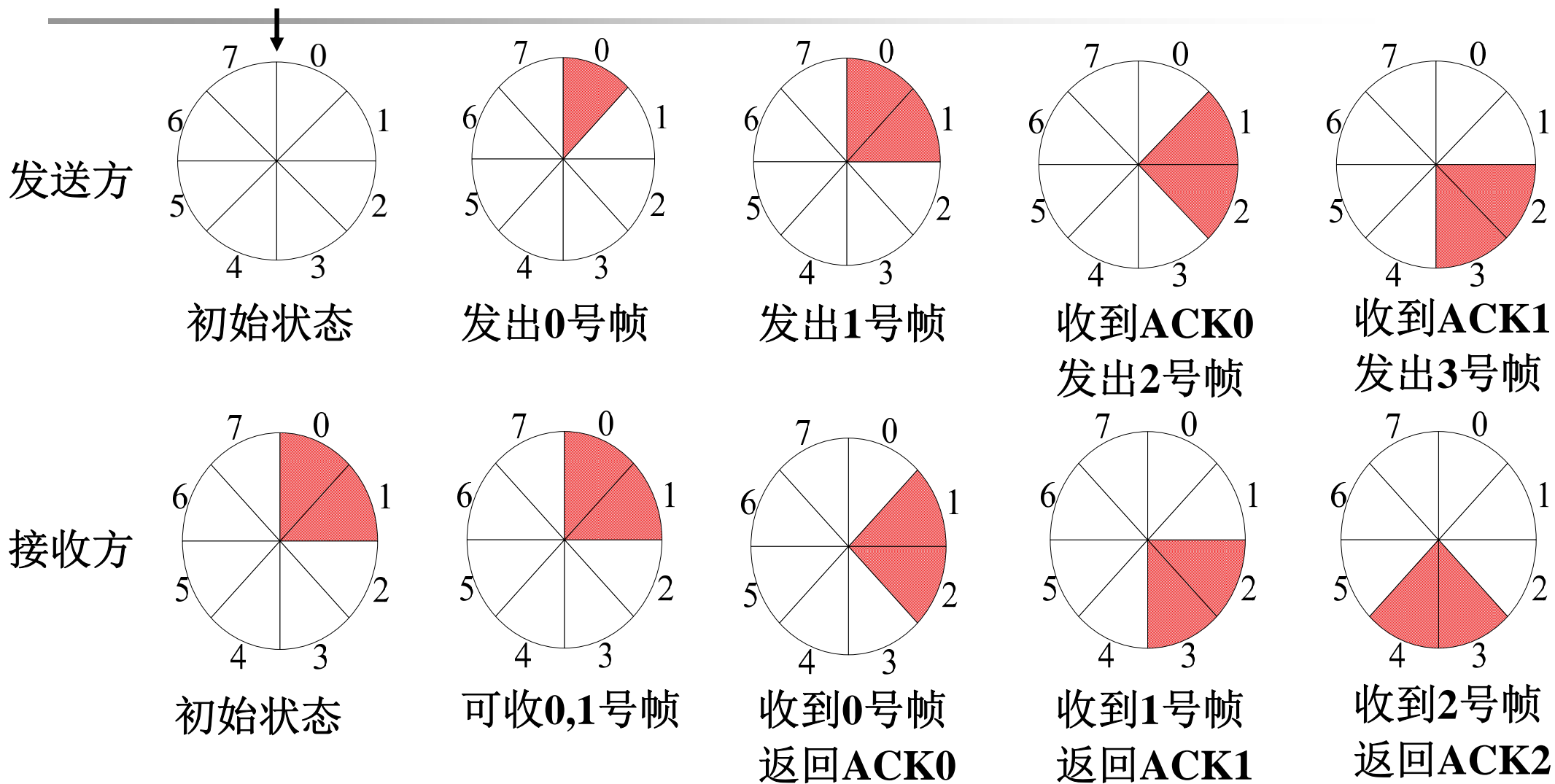
## 3.4 滑动窗口协议



假设：发送窗口 $W_T = 1$ ，接收窗口 $W_r = 1$

窗口最大尺寸为1的滑动窗口协议称为1位滑动窗口协议，即为ARQ协议

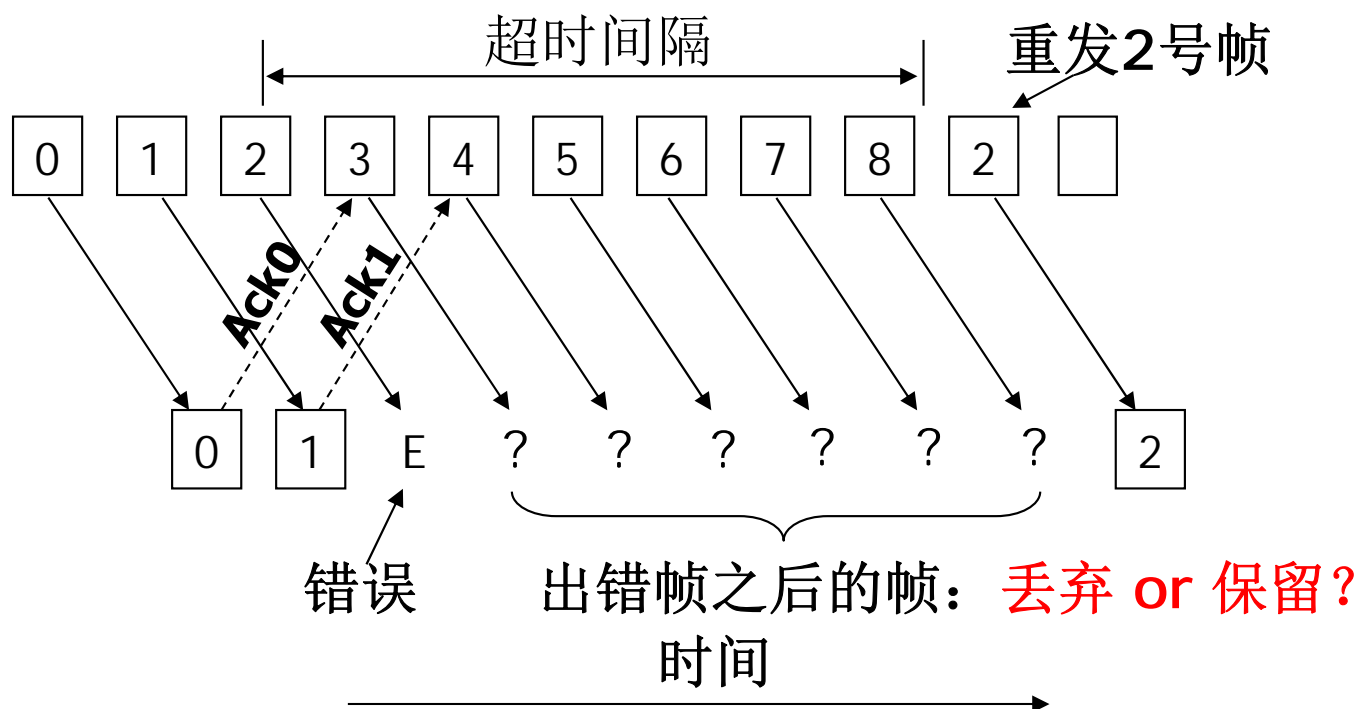
## 3.4 滑动窗口协议



假设：发送窗口  $W_T = 2$ ，接收窗口  $W_r = 2$

## 3.4 滑动窗口协议

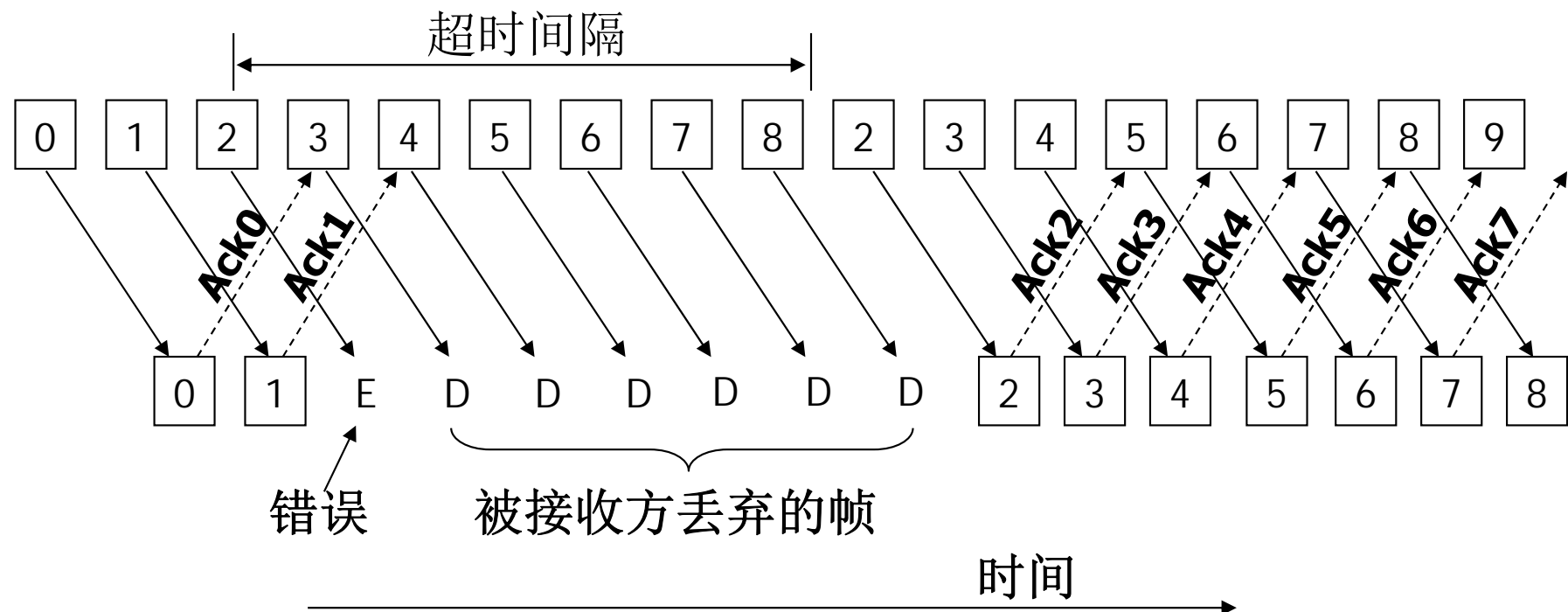
- 一个问题：连续发送的多个帧中，某一个帧出错或丢失，但后续帧已被正确接收，如何处理？
  - 必须遵守的规则：数据链路层需按顺序向网络层上交帧
  - 方案一：出错帧后的帧丢弃，从出错帧开始重新发送  
→ 后退N帧
  - 方案二：出错帧后的帧保留，只重发出错帧  
→ 选择性重传



## 3.4 滑动窗口协议

### 二、后退N帧的滑动窗口协议(Go back N)

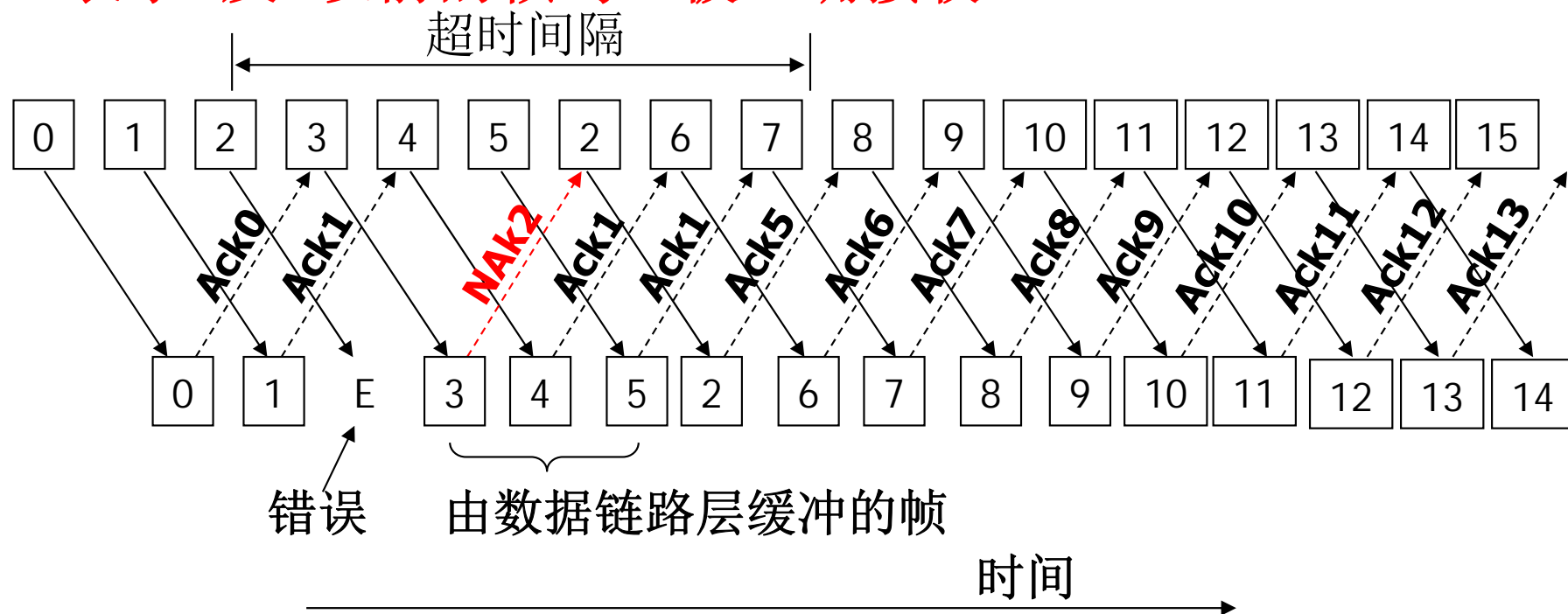
- 当某帧出错时，该帧之后的帧全被丢弃，从出错帧开始重新发送
- 实际上此时接收窗口宽度为1





### 三、选择性重传的滑动窗口协议(selective repeat)

- 当某帧出错时，只选择性地重发该帧，该帧之后发送的帧由接收方数据链路层缓冲，收到重发的出错帧后上交给网络层
- 当接收方检测到出错帧时，发送一个否定的确认(NAK, Negative Acknowledgement)
  - 带来的好处：发送方可以尽快重发出错帧，而不必等到超时
  - 问：如果数据帧不是出错而是丢失？如果确认帧出错或丢失？
- 该协议要求接收方能够临时性缓冲接收窗口内的帧
- **ACK $n$ 表示 $n$ 及 $n$ 以前的帧均已被正确接收**



## 3.4 滑动窗口协议

---

**2009年的一道考研题：**

数据链路层采用后退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**

## 3.5 点对点协议 PPP

- 早期的数据链路层协议：**HDLC(High-level Data Link Control)**

## 3.5 点对点协议 PPP

---

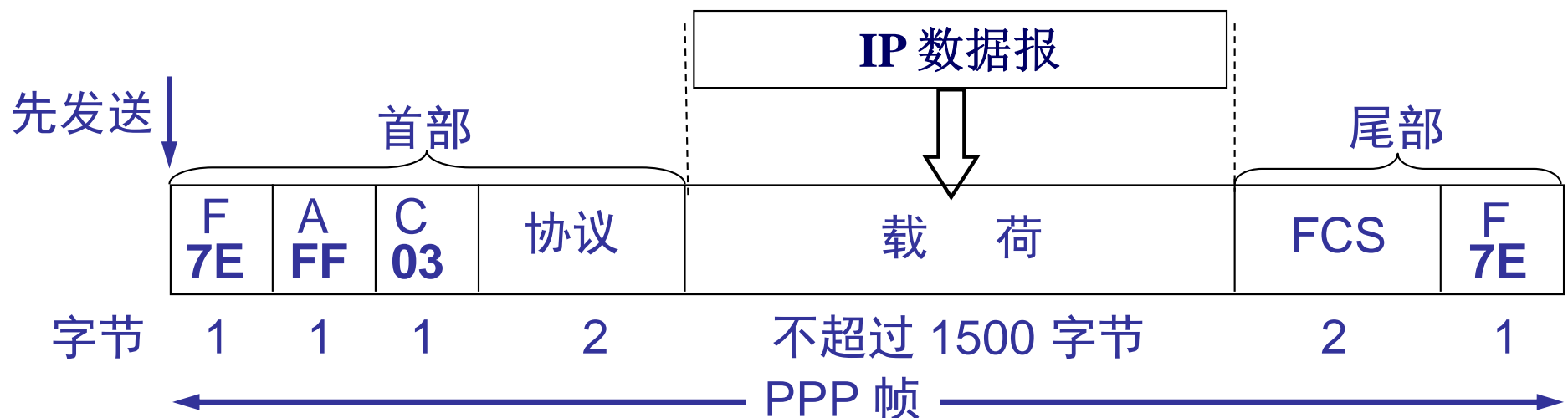
### 一、PPP 协议的特点

- **PPP (Point-to-Point Protocol):** 一种数据链路层协议，广泛应用于点到点链路的数据传输
  - 拨号上网、光纤传输、...
- **1992 年制订，1993 和 1994 年修订，成为 Internet 标准 RFC 1661**
- **PPP 协议有三个组成部分**
  - 将 IP 数据报封装到串行链路的方法
  - 链路控制协议 LCP (Link Control Protocol)
  - 网络控制协议 NCP (Network Control Protocol)

## 3.5 点对点协议 PPP

### 二、PPP的帧格式

- PPP 是面向字节的，所有的 PPP 帧的长度都是整数字节
- 标志字段F: =0x7E (二进制: 01111110)
- 地址字段A: 置为 0xFF, 实际上不起作用
- 控制字段C: 通常置为 0x03
- 协议字段: 2字节, 用于识别信息字段(又称为载荷,payload)的类型
  - 0x0021: PPP 帧的信息字段是IP 数据报
  - 0xC021: 信息字段是 PPP 链路控制数据(LCP)
  - 0x8021: 信息字段是网络控制数据(NCP)
- 校验字段FCS: 2字节的CRC校验



## 3.5 点对点协议 PPP

---

- **PPP的透明传输问题(帧边界识别)**
  - 同步传输(如SONET/SDH): **零比特填充**
  - 异步传输: **字符填充**
- **字符填充**
  - 信息字段(载荷)中的每一个**0x7E** → **0x7D, 0x5E**
  - 信息字段中的每一个**0x7D** → **0x7D, 0x5D**
  - 信息字段中的每一个ASCII 码控制字符(小于 **0x20** 的字符)前面加入 **0x7D**, 且编码改变
    - 例: **0x03** → **0x7D, 0x23**
  - 问: 接收端如何处理?

## 3.5 点对点协议 PPP

注意：只要有连续5个1，无论后面是0或1，都填入0

- 零比特填充

- 发送端：只要发现有 5 个连续 1，则立即填入一个 0
- 接收端：对帧中的比特流进行扫描，每当发现 5 个连续1时，就把这 5 个连续 1 后的一个 0 删除

信息字段中出现了和标志字段 F 完全一样的 8 比特组合

0 1 0 0 1 1 1 1 1 0 0 0 1 0 1 0  
会被误认为是标志字段 F

发送端在 5 个连 1 之后填入 0 比特再发送出去

0 1 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0  
发送端填入 0 比特

在接收端把 5 个连 1 之后的 0 比特删除

0 1 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0  
接收端删除填入的 0 比特

## 3.5 点对点协议 PPP

- PPP支持两种身份认证协议
  - PAP(Password Authentication Protocol)
  - CHAP(Challenge-Handshake Authentication Protocol)

### 三、PPP 协议的工作状态

