

## Ch-03 数据链路层

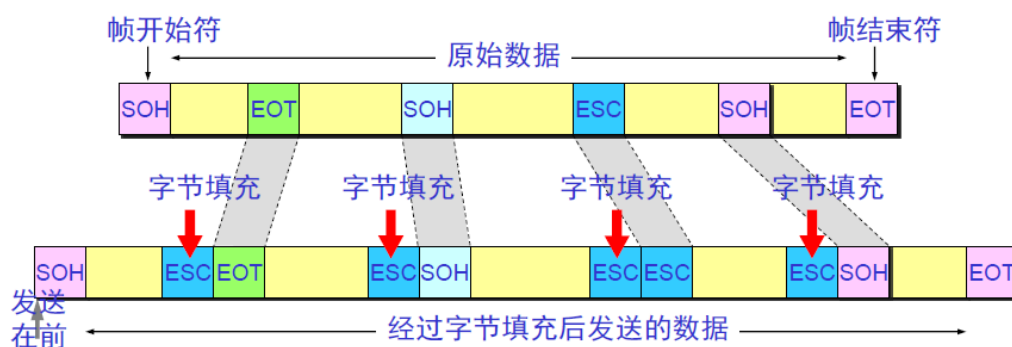
数据链路层：负责在一个结点与其相邻结点之间经过一条链路传递数据

### 成帧

帧是链路层的基本单位，用于承载数据、实施差错控制

组帧的方法：

- 字节计数法：利用头部的一个字段表示该帧字节数
  - 问题：若头部字节出错，会导致失去同步
- 封装成帧：在一段数据的前后分别添加首部和尾部，构成一个帧
  - 首部和尾部的作用是帧定界
  - 若帧头或帧尾在数据部分出现则出现歧义
- 字符填充实现透明传输
  - 数据中出现帧头或帧尾时，在其之前插入一个转义符
  - 接收端将数据送往高层前删除之前插入的转义符
  - 若转义符也在数据中出现，则在转义符之前插入一个转义符，当接收端收到连续两个转义符时就删掉一个



- 比特填充
  - 帧标志  $F$  为 01111110，采用零比特填充
  - 发送端在连续发出 5 个 1 后，填入一个 0 比特再发送
  - 接收端在连续接收 5 个 1 后，将其后的 0 比特删除

## 差错检错

差错检测方法：在信息位中加入冗余位构成码字

奇偶校验：加入冗余位，使码字中 1 的个数为奇数（或偶数）

检错码：无法判定错误的位置

- 一维奇校验：可检测单比特错
- 交织与检验和
  - 突发长度：第 1 位错与最后 1 位错之间的比特数
  - 交织：将突发错转换为随机错

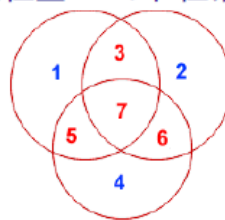
纠错码：判定错误的位置

- 二维偶校验：可检测并纠正单比特错
- 汉明码：校验位 + 信息位

### 纠错码举例：汉明码

- 校验位：(p1, p2, p4, p8, p16, ...)  $r$  位，位于  $2^n$ ， $n=0,1,\dots,r-1$
  - 信息位：(m3, m5, m6, m7, m9, ...)  $m$  位
  - 第  $K$  位信息的校验位来自  $K$  的 2 幂次之和，如  $5=1+4$
- 例如 (7,4) 汉明码，码长 7，信息位长  $m=4$ ，校验位长  $r=3$  ( $=7-4$ )
- 信息位 (m3, m5, m6, m7)，校验位(p1, p2, p4)
    - $3=1+2$ ;  $5=1+4$ ;  $6=2+4$ ;  $7=1+2+4$
    - (m3, m5, m6, m7) 分别影响 (1,2)(1,4)(2,4)(1,2,4)
  - 根据校验位的计算结果，可定位差错的位置——纠 1 位错

7	6	5	4	3	2	1	
D	D	D	P	D	P	P	码字
D	-	D	-	D	-	P	p1偶校验
D	D	-	-	D	P	-	p2偶校验
D	D	D	P	-	-	-	p4偶校验

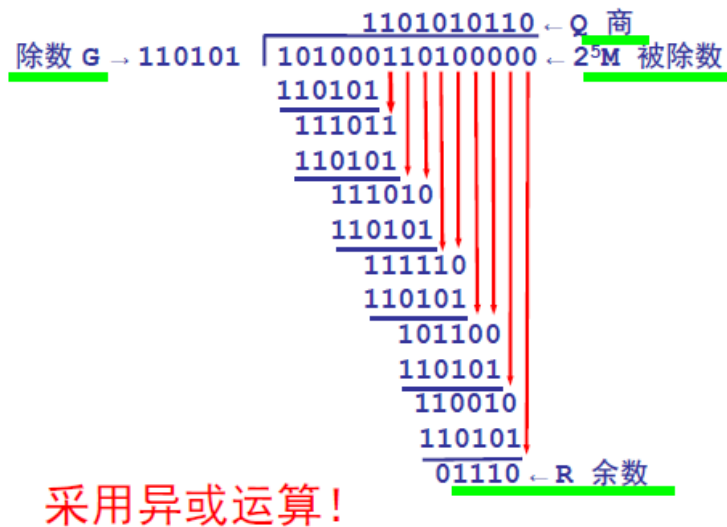


“校验位都是  $2^n$ ，数据位都不是  $2^n$ ，至少影响 2 个校验位”——来自常可

循环冗余检验码

- CRC
- 生成多项式

# 求循环冗余检验码的过程



## 差错控制方法

- 前向纠错：采用纠错码，接收端发现差错并纠正
- 自动请求重传：接收端检测到差错时，通知发送端重传，直至收到正确数据
- 混合纠错：检错码 + 纠错码

## 选择重传

### 自动请求重传 ARQ (Automatic Repeat reQuest):

- 采用检错码，接收端通过译码发现传输错误，但是无法纠正，因此采用自动请求重传的方式

### 停等协议：收端收到帧后发送确认帧，发端收到确认帧后发送新的帧

- 有差错情况，超时重传
- 收端用帧校验检测并丢弃错帧，发端增加超时定时器
- 当超时定时器到但未收到确认帧时，重传已发送的帧
- 数据帧和确认帧都要带序号
- 简单，信道利用率低

## 实现停等协议的要点

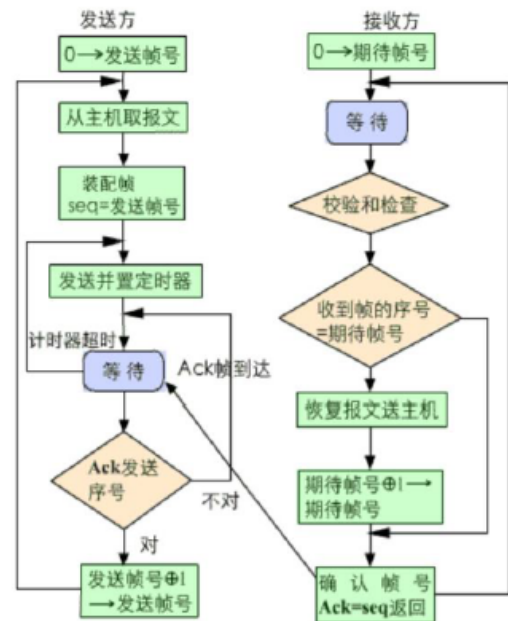
### ■ 发送方：

- 发送数据帧后缓存
- 启动超时定时器
- 收到确认帧后清除缓存，发送新数据帧
- 当定时器到，重传缓存帧

### ■ 接收方：

- 收到错误的帧，丢弃
- 收到正确的数据帧时，发送确认；若是**新数据帧**则保存；若为**重复帧**则丢弃

### ■ 发送端自动对错帧重传，称为ARQ (Automatic Repeat reQuest)

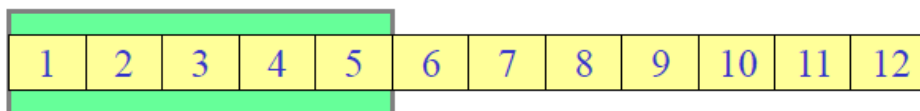


连续 ARQ: 滑动窗口

- 发送方连续发送多个帧，无需每发完一个就停下来等待确认
- 提高了信道利用率
- 错帧时，从错帧开始重新发（回退N）
- 接收端只接受序号连续的帧
- 确认帧序号表示期望收到的下一个帧的序号

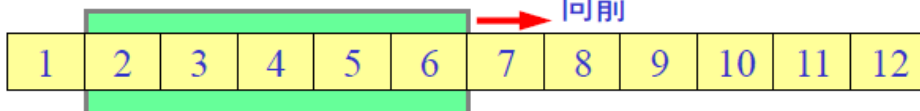
## 连续 ARQ 协议

发送窗口



(a) 发送方维持发送窗口（例如 5）

发送窗口



(b) 收到一个确认后发送窗口向前滑动

- 发送方：只允许发送在发送窗口内的W个帧；每发送一个帧启动一个定时器，等待确认
- 接收方：仅接收序号连续的帧，如收到0号帧则响应ACK1，收到1号帧则响应ACK2；ACK<sub>i</sub>表示期望收到的下一个帧的序号
- 发送方：收到ACK<sub>i</sub>则取消定时器，调整发送窗口为[i, i+W-1]；若第j个帧定时器到，则调整发送窗口为[j, j+W-1]

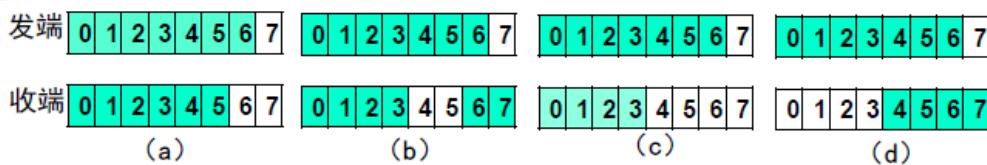
窗口的最大值:

- 若接收窗口为 1, 发送窗口不超过  $2^n - 1$
- 对选择重传协议, 接收窗口最大值为  $2^n / 2$

选择重传协议: 接收端可以接收序号不连续但仍在接收窗口内的数据帧

- 避免重传已经正确到达接收端的数据帧, 但接收端要缓存多帧数据

## 选择重传ARQ: 接收窗口



- 设帧序号用3位编码, 发送窗口为7, 接收窗口为6
- (a) 发送0~6帧后收端确认0~5, 但确认帧全部丢失, 发端窗口为0~6, 发端重发0~6
- (b) 收端发送了确认帧后, 接收窗口调整为6~3;
- 问题: 收端在接收窗口内的帧都应是新帧, 但是6帧为新帧而0~3帧为旧帧, 在接收窗口中新旧序号重叠, 收端无法区分;
- 解决: 避免接收窗口中新旧序号重叠! 接收窗口为4 ( $=2^n/2$ )
- (c) 发送0-6帧, 收端确认0~3, 但确认帧全部丢失, 发端重发0~6帧
- (d) 接收端发送了确认帧后窗口为4~7, 接收了4~6帧拒收了0~3帧; 确认4~6帧后, 接收窗口调整为7~2; 发送窗口调整为7~5, 继续.....

累计确认: 收端连续多个只发一个确认帧

捎带确认: 暂缓确认, 将确认信息搭载在下一个数据帧上