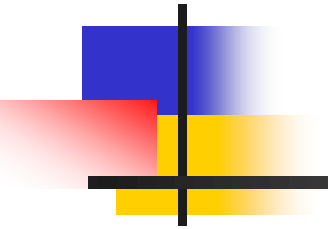


第 3 章 数据链路层
——组帧及检错

刘志敏
liuzm@pku.edu.cn

链路层的功能

- 相邻结点间的数据传输

(1) 链路管理：建立链路、拆除链路

(2) **同步**：组帧、解帧

(3) 流量控制

(4) **差错控制**

(5) 区分数据信息和控制信息

(6) **透明传输**

(7) 寻址

提纲

数据链路层简介

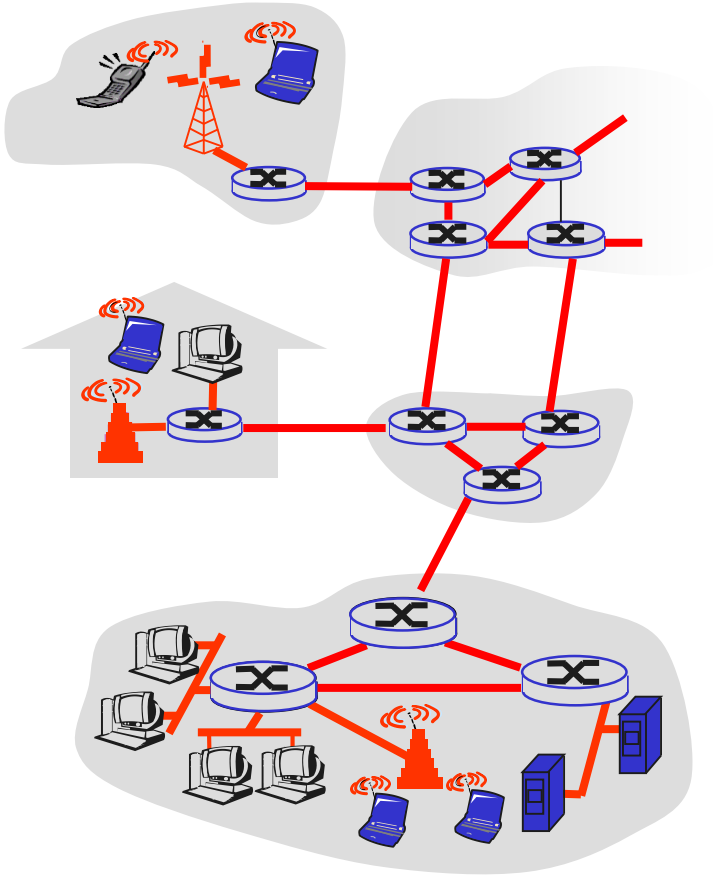
成帧：同步、透明传送

差错控制技术：检错及纠错

数据链路层：简介

名词术语：

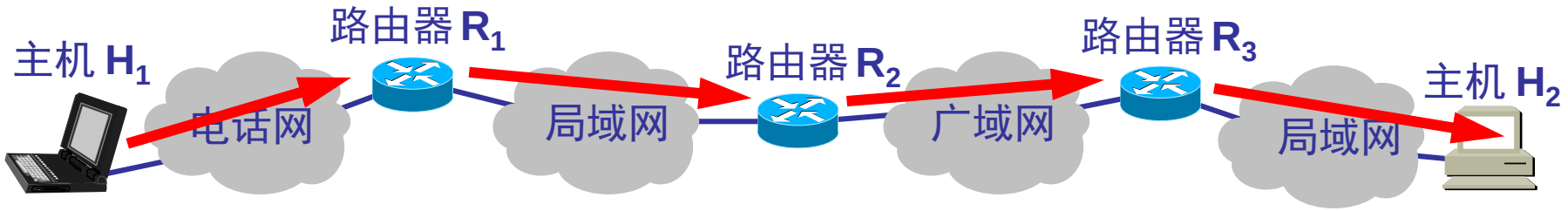
- 结点(node)：主机、路由器等
- 链路(link)：连接相邻结点之间的信道
  - 有线链路
  - 无线链路
- 帧(frame)：在数据链路层上传输数据的基本单位，用于承载数据。



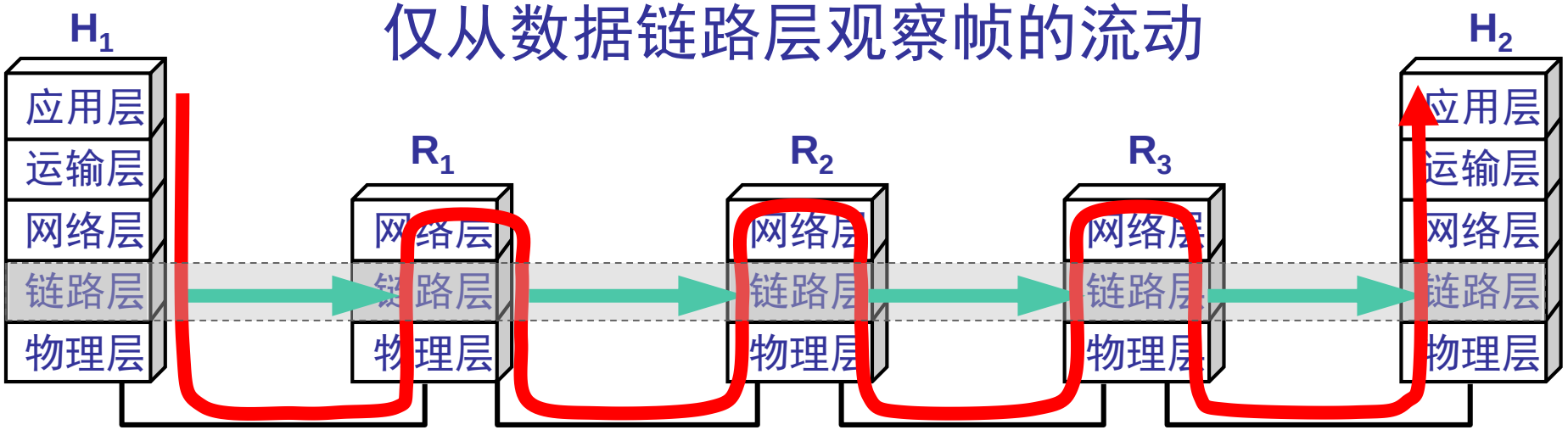
数据链路层 负责在一个结点与其**相邻结点之间**经过一条链路传递数据

数据链路层的模型

主机 H_1 向 H_2 发送数据

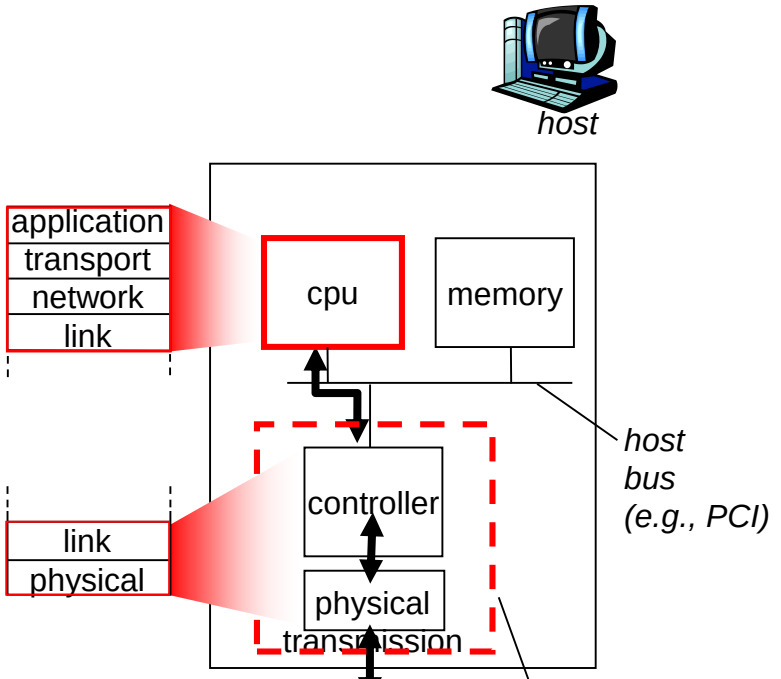


仅从数据链路层观察帧的流动

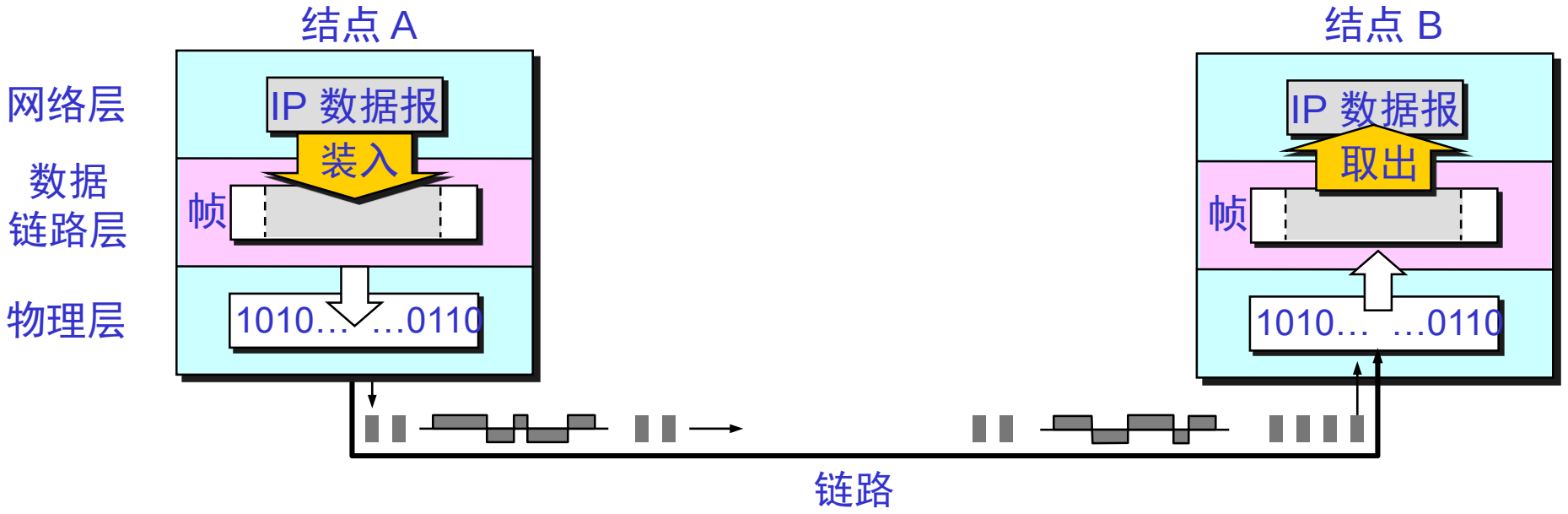


链路层在哪里实现？

- 在每个主机中
- 链路实现是在适配器，即网卡（NIC）
  - 以太网卡，802.11 网卡等
  - 网卡实现链路层及物理层的功能
- 与主机的系统总线连接
- 为软件、硬件及固件的结合



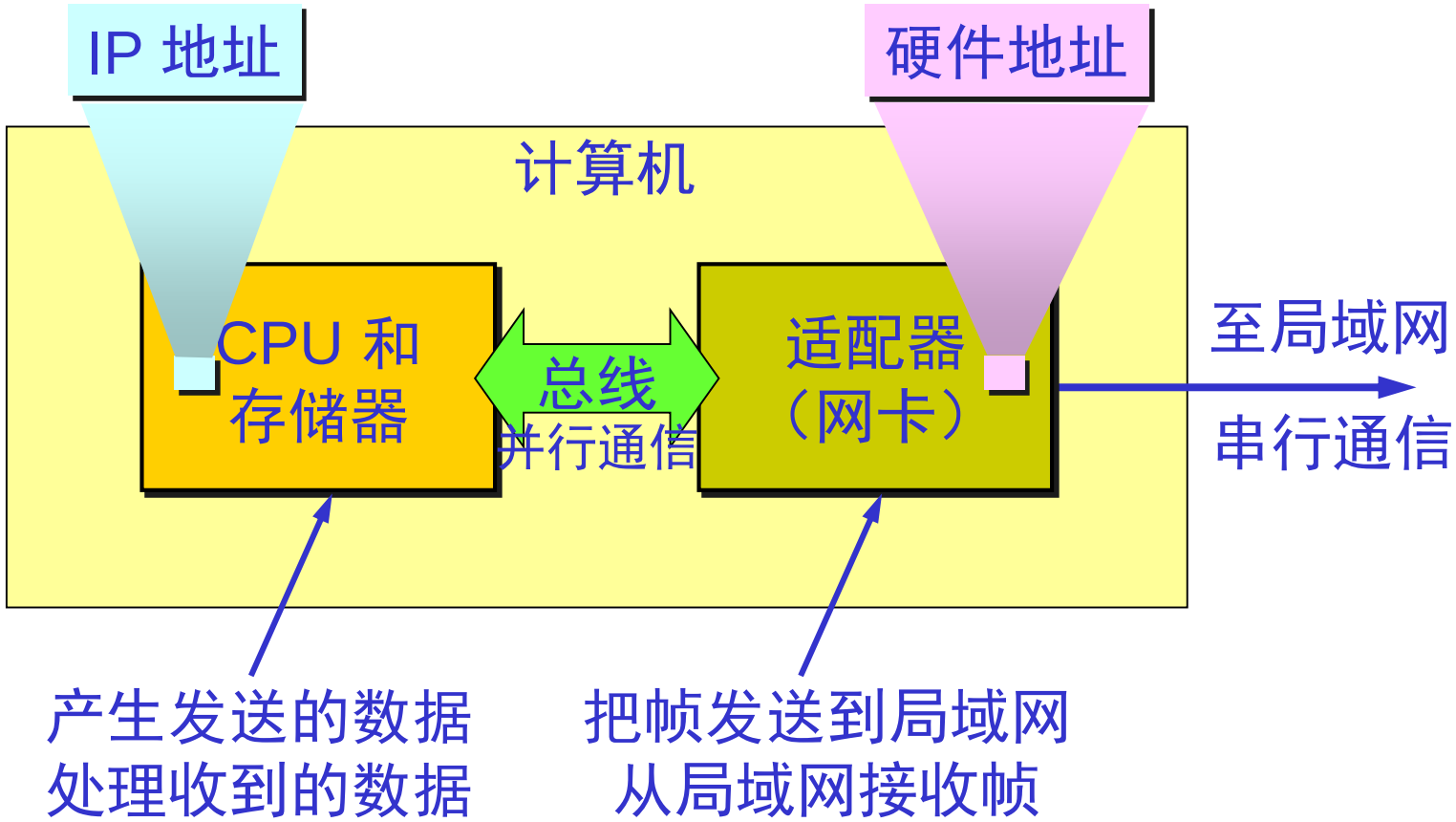
适配器之间的通信



- 发端
  - 将数据封装为帧
  - 增加校验位，接收地址，实施流量控制等
- 收端
  - 检查错误，检查接收地址，实施流量控制等
  - 提取数据，交由高层处理

网卡的作用

- 串/并变换、数据缓存、实现数据链路协议（由设备驱动程序提供）

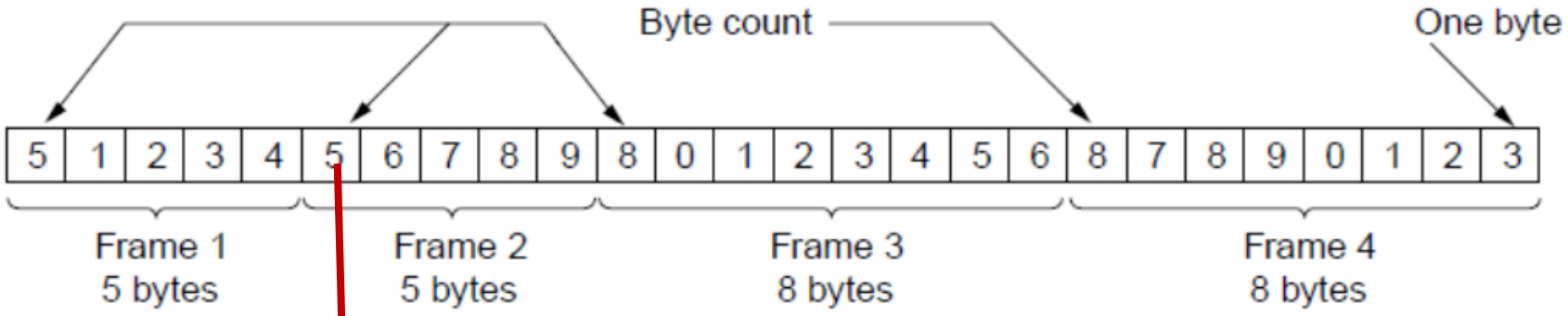


组帧有哪些方法？

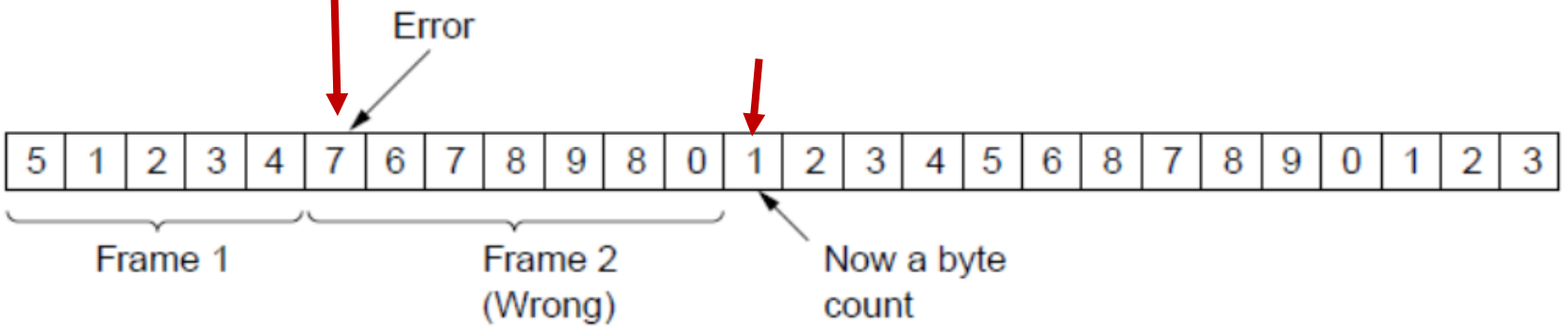
- 帧为链路层的基本单位，用于承载数据、实施差错控制等
- 如何确定帧的开始？异步传输中的起始位，同步传输中的标志位。
- 组帧的几种方法
  - 字节计数：
  - 固定帧长度：RS232
  - 标志位：以太网
  - 其它？

组帧的方法

- 字节计数法：利用头部的一个字段来标识该帧的字节数。

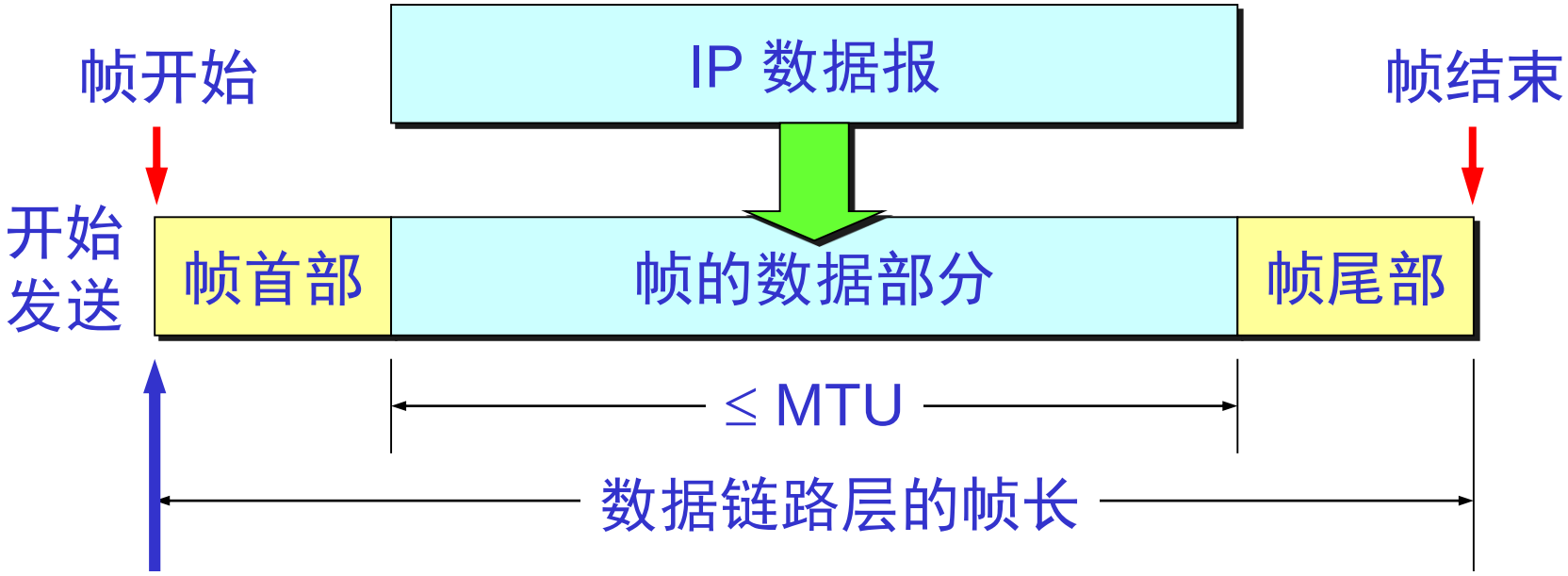


- 问题：若头部字节出错，导致失去同步。
- 若能将一个特殊的字节作为帧的首尾标志，则帧失步后就可以重新同步。

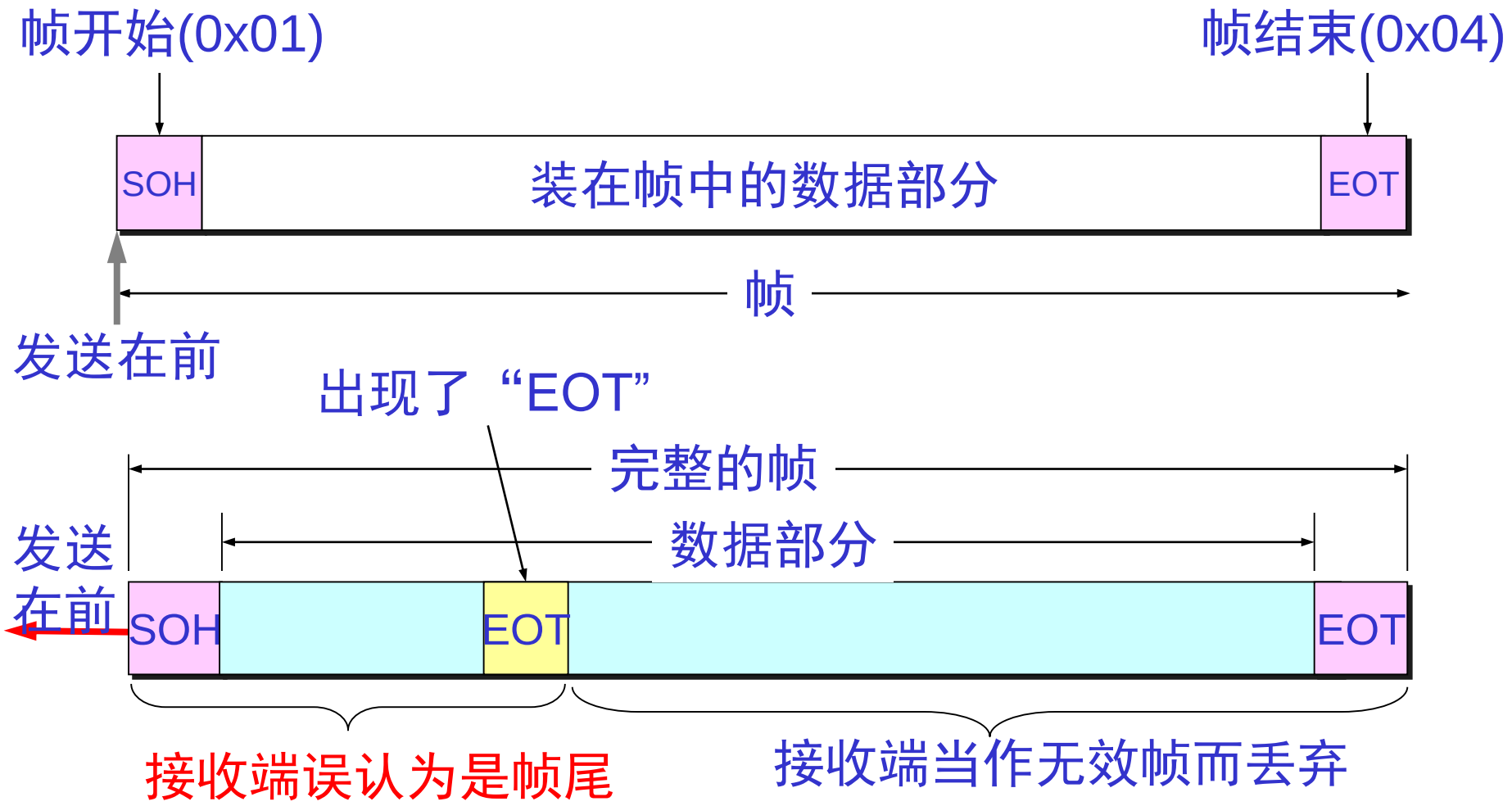


封装成帧

- 封装成帧(framing)就是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



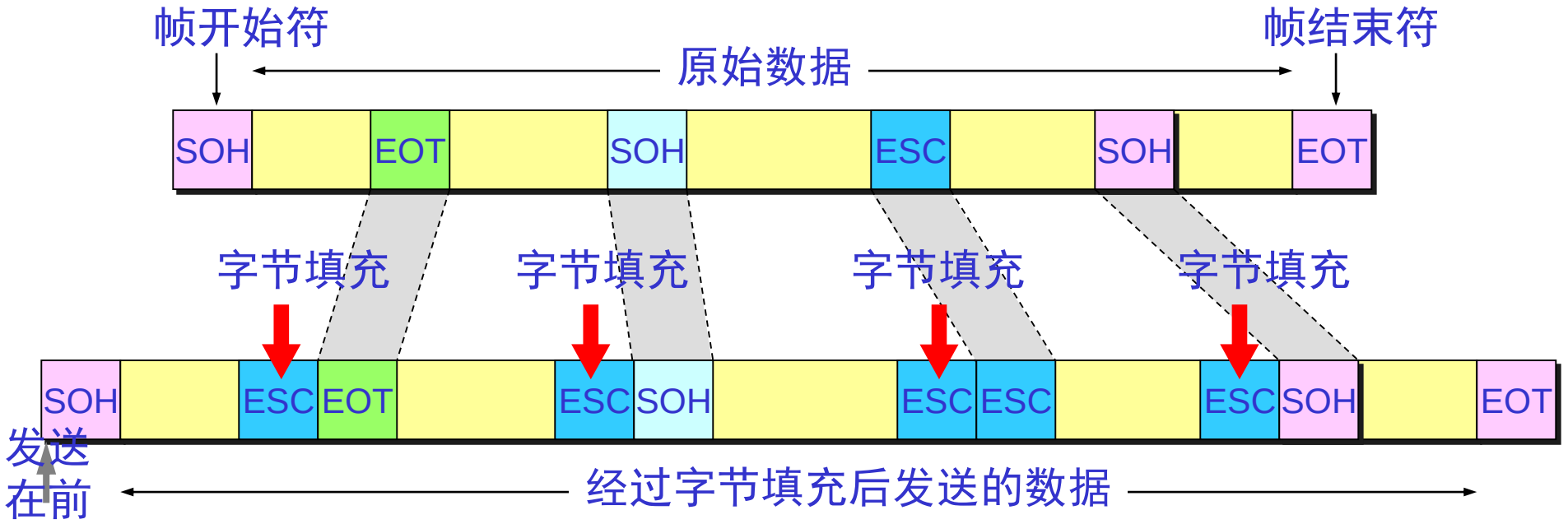
帧定界及透明传输



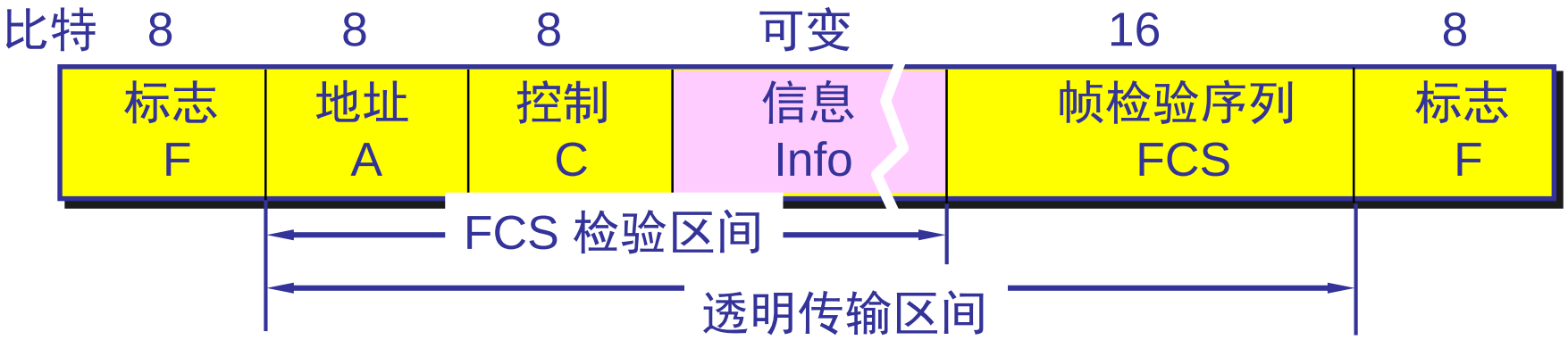
- 如何保证帧头、帧尾不在数据部分中出现？

用字符填充实现透明传输

- 发送端若在数据中出现“SOH”或“EOT”，则在其之前插入一个转义符“ESC”(0x1B)。
- 接收端将数据送往高层之前删除插入的转义符
- 如果转义符也在数据中出现，则在转义符之前插入一个转义符。当接收端收到连续的两个转义符时就删除一个



用比特填充实现透明传输



HDLC，帧标志F为01111110，采用零比特填充

数据中某一段比特组合恰好与 F 字段一样

0 1 0 0 1 1 1 1 1 1 0 0 0 1 0 1 0

会被误认为是 F 字段

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

填入 0 比特

在接收端将 5 个连 1 之后的 0 比特删除，恢复原样

0 1 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0

删除插入比特 0

练习题

- 若字符编码为 A:01000111 B:11100011 FLAG:01111110 ESC:11100000 为传输4个字符的帧 A B ESC FLAG，请分别给出在字节计数、字节填充、比特填充等成帧方法下所发送帧的比特序列。

- 若采用字节填充的方法发送数据段 A B ESC C ESC FLAG FLAG D，请给出经过填充后的输出。

提纲

数据链路层简介

成帧：同步、透明传送

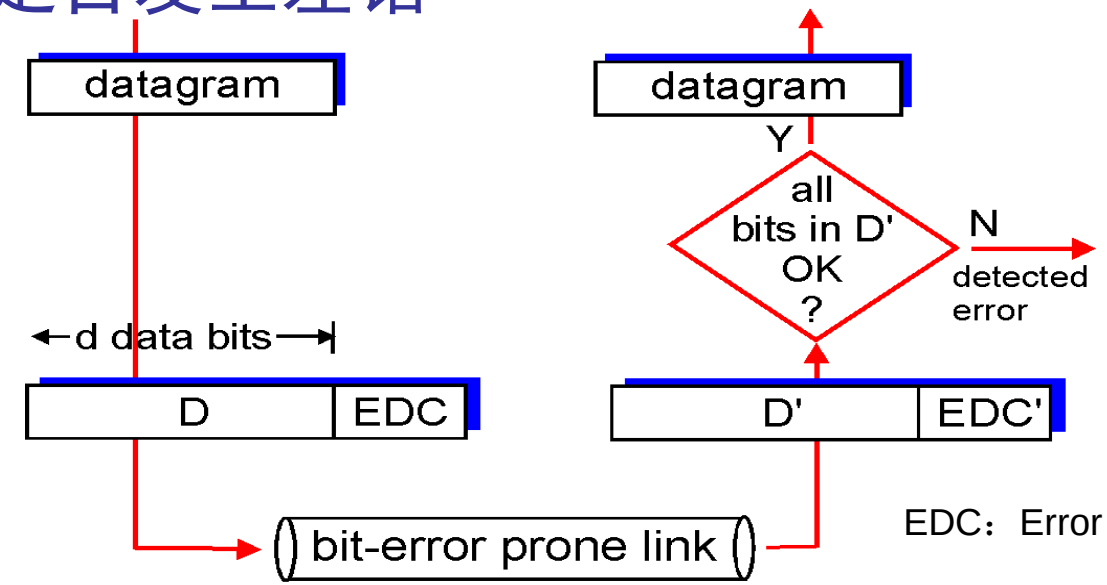
差错控制技术：检错及纠错

差错

- 差错：数字通信系统中接收位与发送位不一致
- 信道误码率：P_b=误码位数/发送的总位数
- 若误码率为P_b，帧长为L位，
  则无差错帧的概率$P_1=(1-P_b)^L$
- 例：ISDN 64Kbps信道，90%时间里误码率低于10^{-6}。设帧长1000b，求每天传输帧中错帧的数量？
  解：正确帧的概率$P_1=(1-10^{-6})^{1000}=0.999$
  每天传输帧数=64K/1000×3600×24=5.5×10^6，错帧总数为5.5×10^3

差错检测

- 差错检测方法：在信息位中加入冗余位
- 信息位：要发送的数据
- 冗余位：按某种关系对信息位运算所得到的数据
- 发送过程：信息位+冗余位 构成码字，发送码字
- 接收过程：检查信息位和冗余位之间的关系（校验过程），以发现传输过程中是否发生差错

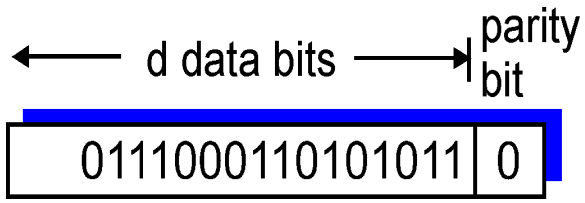


EDC：Error Detection and correction

差错检测：奇偶校验

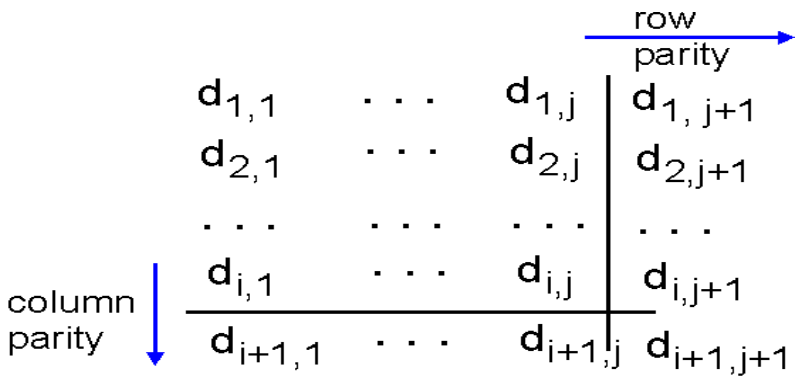
- 奇偶校验：增加冗余位，使码字中“1”的个数为奇数（或偶数），称为奇（或偶）校验

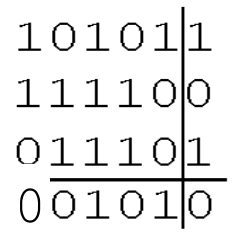
一维奇校验：可检测单比特错

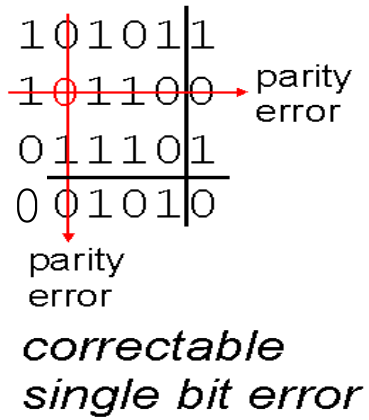


检错码：无法判定错误的位置

二维偶校验：可检测并纠正单比特错







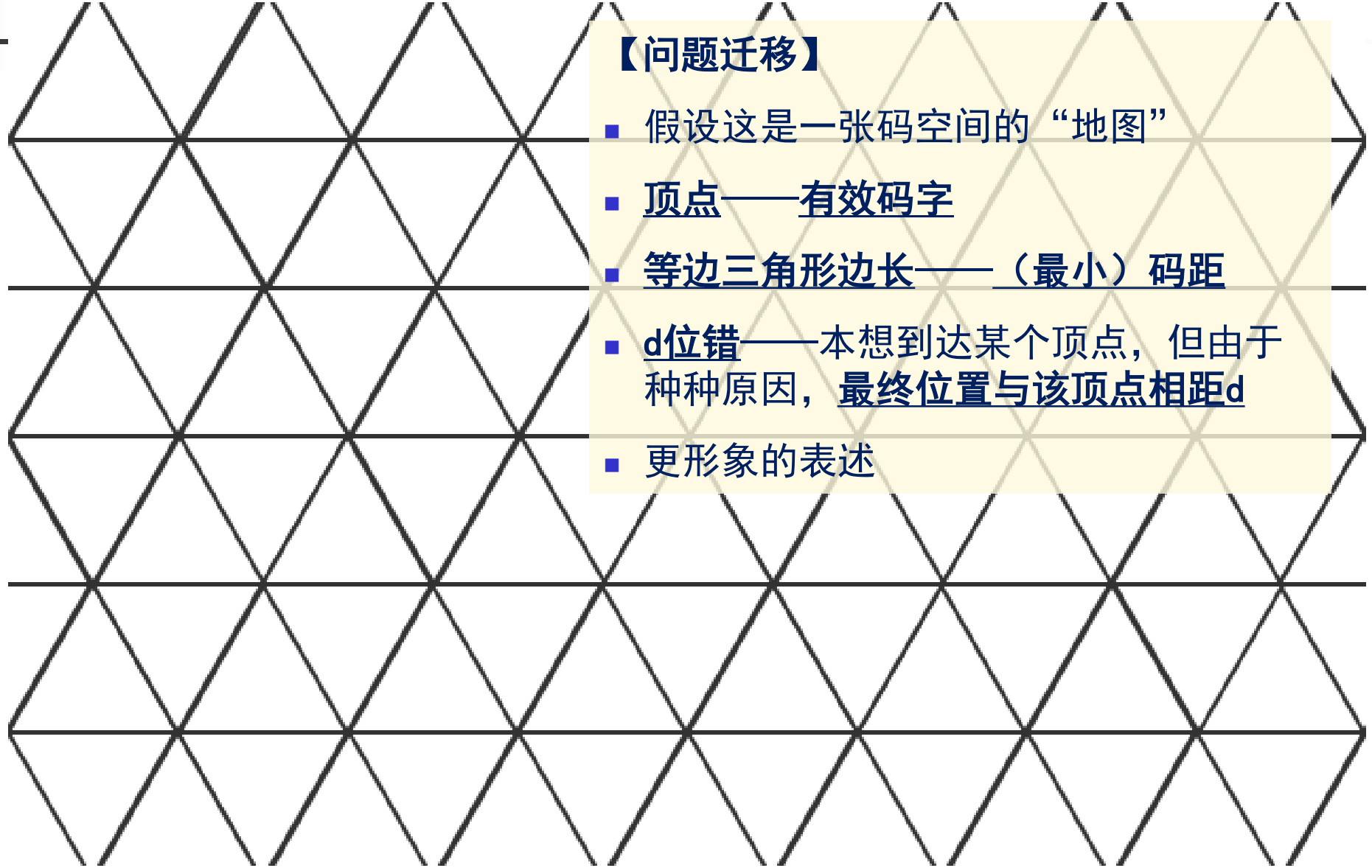
纠错码：判定作为的位置

差错编码分类

- 检错码：能自动发现差错的编码，如奇偶校验码、循环冗余码CRC(Cyclic Redundancy Code)
- 纠错码：不仅能发现差错而且能定位差错的位置的编码，如汉明码、卷积码等
- 定义：汉明距离（码距）为两个码字中不相同位的个数
- m位信息+r位校验：有2^{m+r}种可能，但只使用了2^m个码字，码距>1，码字空间稀疏
- 编码就是增大码距，解码就是找出距离码字最近的码字

码距、检错能力与编码效率

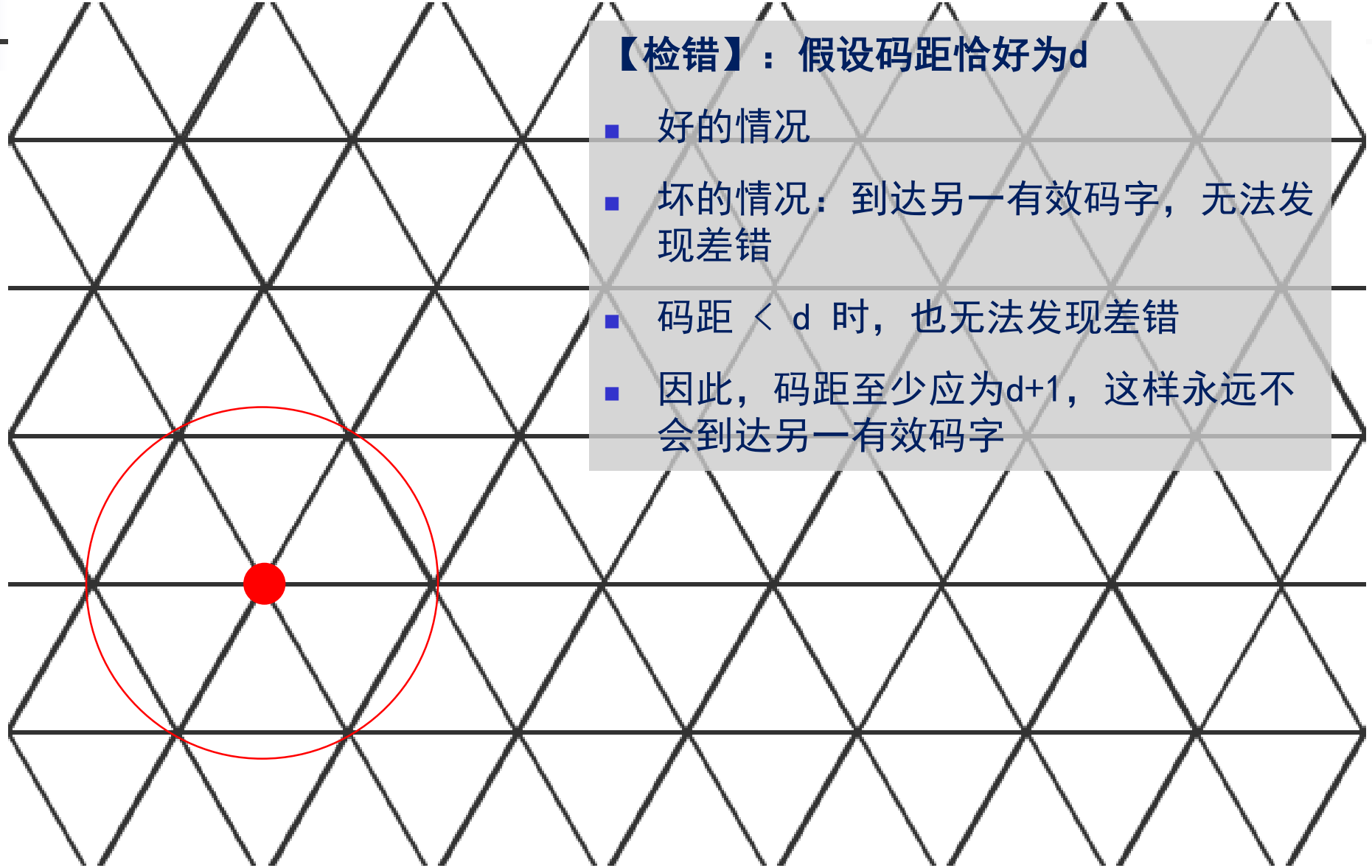
- 一维的奇偶校验码的码距是多少？ 2
- 检错能力是多少？ 1
- 若要检测d位错，则需要码距为d+1，因d位错不可能将一个码字变成另一个码字
- 若要纠正d位错，则需要码距为2d+1，因即使出现d位错，结果还是距其原有的码字更近
- 码率：用以衡量编码效率；若码长为n的码字中含有信息位m位，则码率=m/n
  - 纠错码的码率低，检错码的码率高
  -
    -



【问题迁移】

- 假设这是一张码空间的“地图”
- **顶点——有效码字**
- **等边三角形边长——（最小）码距**
- **d位错**——本想到达某个顶点，但由于种种原因，**最终位置与该顶点相距d**
- 更形象的表述

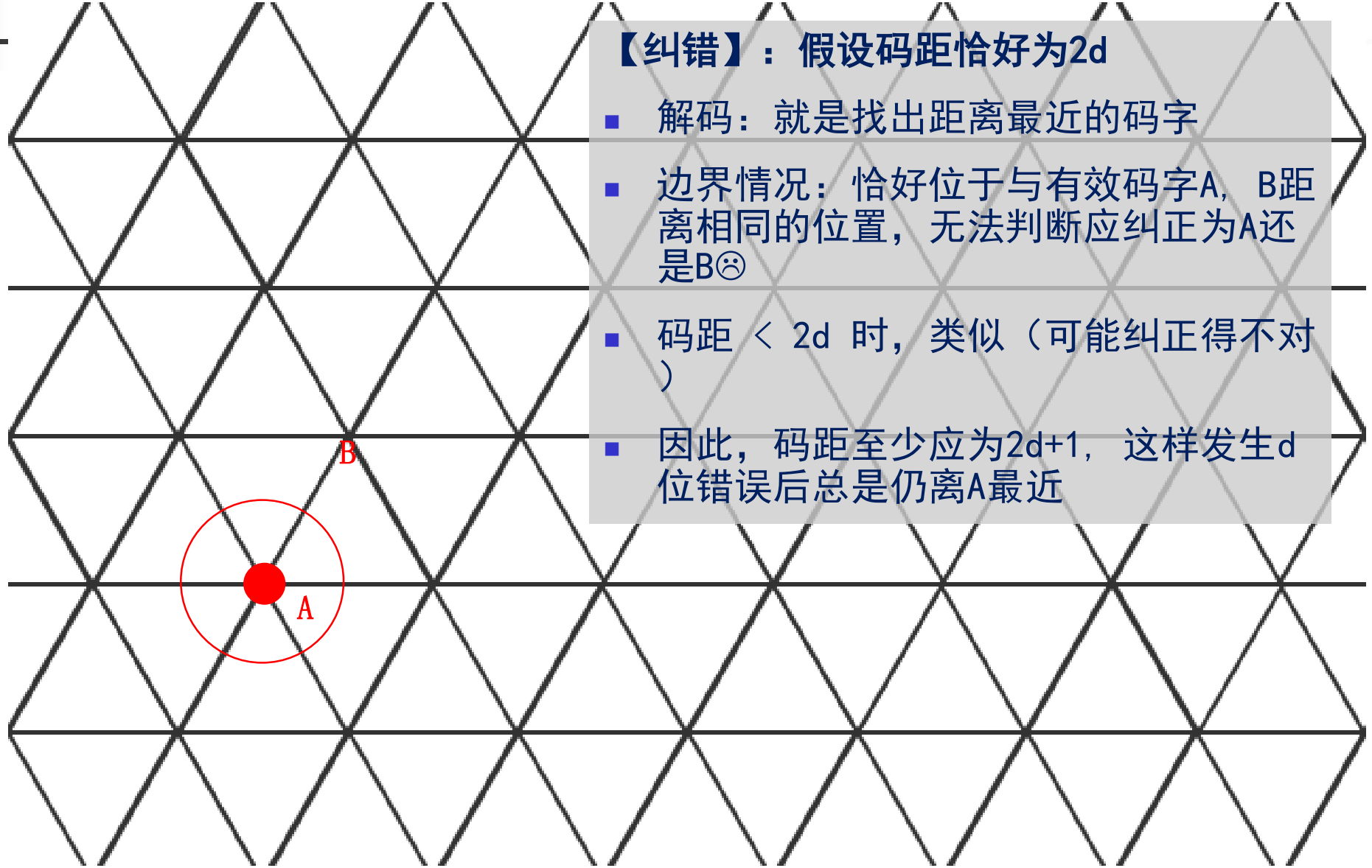
对码距的**直观理解** **来自**1600017739 庞悦



【检错】：假设码距恰好为d

- 好的情况
- 坏的情况：到达另一有效码字，无法发现差错
- 码距 < d 时，也无法发现差错
- 因此，码距至少应为d+1，这样永远不会到达另一有效码字

对码距的**直观理解** **来自**1600017739 庞悦



【纠错】：假设码距恰好为2d

- 解码：就是找出距离最近的码字
- 边界情况：恰好位于与有效码字A，B距离相同的位置，无法判断应纠正为A还是B☹
- 码距 < 2d 时，类似（可能纠正得不对）
- 因此，码距至少应为2d+1，这样发生d位错误后总是仍离A最近

对码距的**直观理解** **来自**1600017739 庞悦

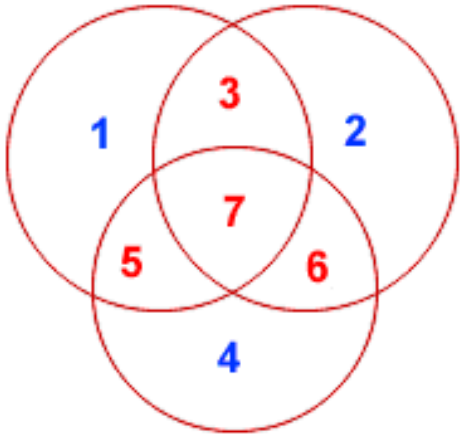
纠错码举例：汉明码

- 校验位：(p1, p2, p4, p8, p16, …) ***r*位，位于2^n，n=0,1,…,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



“检验位都是2^n，数据位都不是2^n，至少影响2个校验位”—来自常可

汉明码举例

- 对1101用(7,4)汉明码编码

7	6	5	4	3	2	1	
1	1	0		1			信息
1	-	0	-	1	-	0	P1偶校验
1	1	-	-	1	1	-	P2偶校验
1	1	0	0	-	-	-	P4偶校验

发送消息

1 1 0 0 1 1 0 ---------->

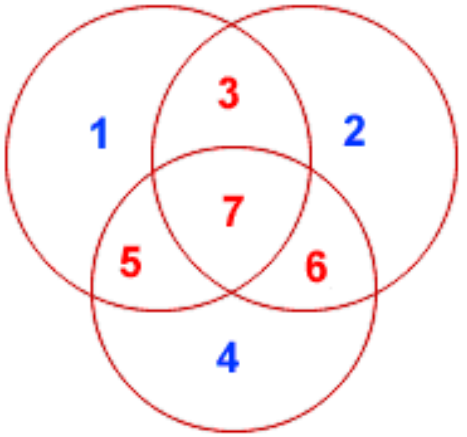
BIT: 7 6 5 4 3 2 1

接收消息

1 1 1 0 1 1 0

BIT: 7 6 5 4 3 2 1

7	6	5	4	3	2	1			
1	1	1	0	1	1	0	码字		
1	-	1	-	1	-	0	P1偶校验	NOT!	1
1	1	-	-	1	1	-	P2偶校验	OK!	0
1	1	1	0	-	-	-	P4偶校验	NOT!	1

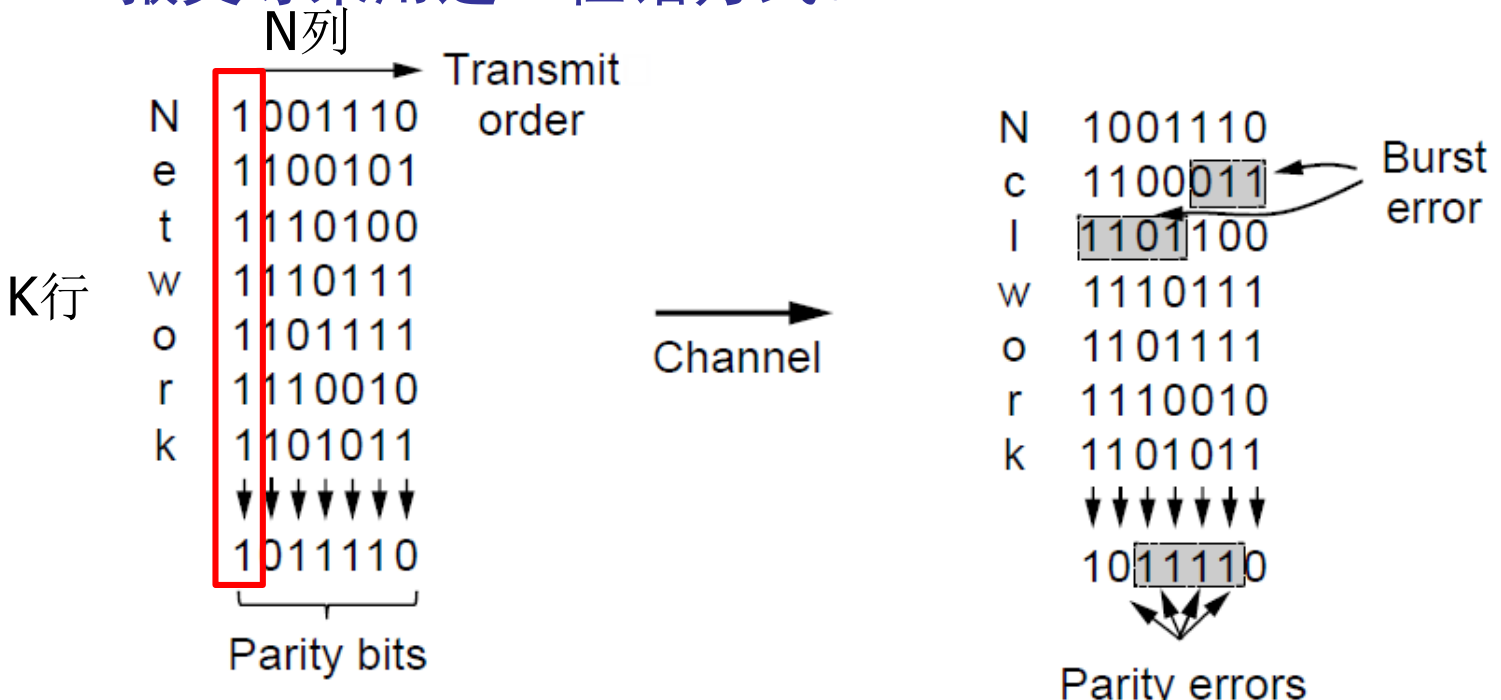


练习题

- 若收到12位汉明码0XE4F，假设首位为最低位，问该码的原始值是什么，设最多出现1位错。
- 举例说明：两维奇偶校验能够纠正和检测1比特错，某些2比特错能被检测但不能纠正，某些4比特错无法检测。

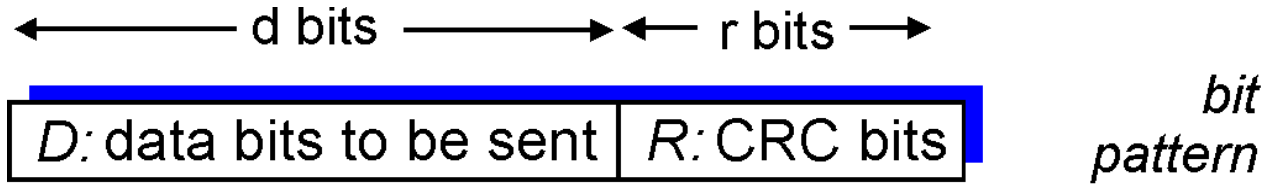
检错码：交织与校验和

- 突发错（时间上连续的错），突发长度：第1位错与最后1位错之间的比特数
- 码字的检错能力有限，奇偶校验码、汉明码，希望1个码字中最多只错1位。
- 如何使码字中最多只出现1位错？交织：将信息位组成n列k行，按列计算校验位，按行发送，将突发错转换为随机错；只要突发长度≤ n，则可以检错
- 检验和：对发送数据计算校验和，接收端实施同样计算以检错。例如IP头，TCP、UDP报文等采用这一检错方式。



检错码举例：循环冗余校验

- CRC(Cyclic Redundancy Check)：一种编码效率更高的检错码，用于Ethernet，802.11 WiFi，ATM
- d位数据位，组成一个二进制数 D；选择 r+1 位 生成式 G；计算 r 位的CRC位 R
- 发送端：使 <D,R> 被 G 整除（模 2）
- 接收端：已知G，用 <D,R> 除G，若余数为零，则保留，否则，检测到错误
- 可以检测突发长度小于等于r位的错



bit pattern

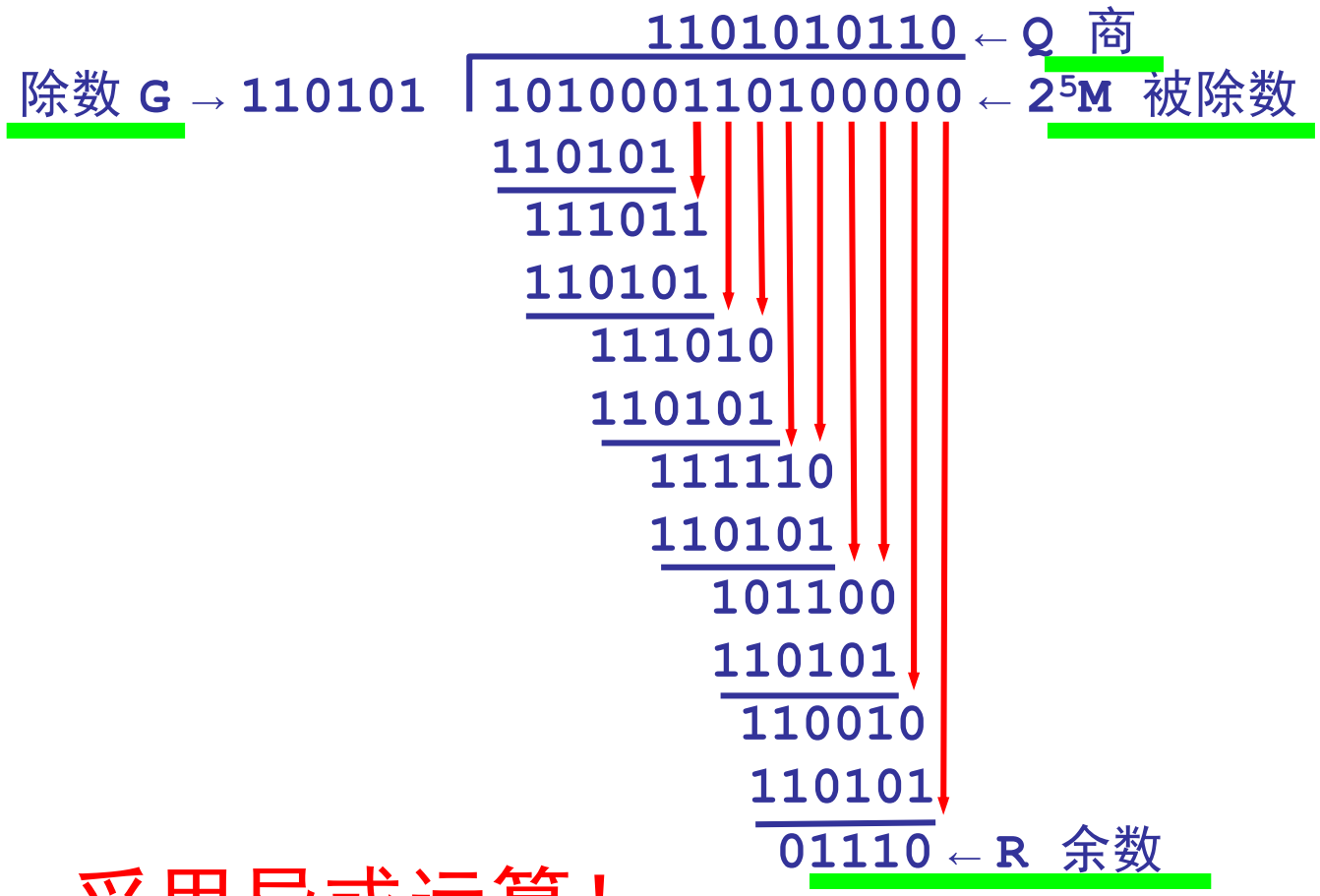
$D * 2^r$ XOR R

mathematical formula

编码及译码过程举例

- 数据 D = 1010001101（d位）;
  设 G = 110101, r+1 = 6, $2^5 D$，即D后添加5个0，再除以选定的G，得到商Q和余数R，由R得到冗余码
- 商 Q = 1101010110，余数R = 01110。将余数 R 作为冗余码添加在数据 D 后发送，即发送101000110101110
- 接收端收到码字，除以选定的*G*，若余数为0则无差错，提取发送的数据；否则有错，丢弃数据

求循环冗余检验码的过程



采用异或运算！

检错码的编码及译码

- 练习：使用CRC传输10011101，生成多项式为x^3+1
  （1）给出实际传输的比特序列
  （2）若第3个比特错，请说明在接收方可以检测出
  （3）给出一个传输错误的实例，使得接收方无法检测错误

- 为保证数据的无差错传送，接收端丢弃有差错的码字，那如何保证数据传输的可靠性？

循环冗余码（续）

- d位信息位与r位校验位组成码字T，对应的多项式分别为D、F、T

$$T = 2^r D + F$$

- （r+1）位特定比特序列，对应多项式为G；
- 若希望码字T能被G整除，问如何求F?

$$\frac{2^r D}{G} = Q + \frac{R}{G}, \quad T = 2^r D + R$$

- R ?= F，必须证明T可被G整除！

$$\frac{T}{G} = \frac{2^r D + R}{G} = \frac{2^r D}{G} + \frac{R}{G} = Q + \frac{R}{G} + \frac{R}{G} = Q$$

CRC的特点

- G(X)：包含两项或多项，含因式(X+1)和1，但不含($X^{i-j}+1$)因子，i-j>1
- 编码效率高
  - G(X)能检出所有1比特错
    - 因G(X)不含X^i因子即含1
  - G(X)能检出所有2比特错
    - 因G(X)不含X^i因子且对任何0<e≤n-1的e除不尽x^e+1
  - G(X)能检出所有奇数位错
    - 因G(X)含因式(X+1)
  - 能检出所有长度小于等于CRC长度的突发错
- 易于硬件实现
  - 移位、异或

G(x) 生成多项式举例

- CRC-12
  - $X^{12} + X^{11} + X^3 + X^2 + X + 1$
- CRC-16
  - $X^{16} + X^{15} + X^2 + 1$
- CRC-CCITT
  - $X^{16} + X^{15} + X^5 + 1$
- CRC-32
  - $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- CRC序列采用移位寄存器实现

小结：差错控制方法

- 差错编码分类
  - 检错码：例如奇偶校验、CRC、交织
  - 纠错码：例如汉明码
- 差错控制方法
  - 前向纠错（FEC：Forward Error Correction）：采用纠错码，接收端发现差错并纠正
  - 自动请求重传（ARQ：Automatic Repeat reQuest）：接收端检测到差错时，通知发送端重传，直到收到正确的数据
  - 混合纠错（HARQ：Hybrid ARQ）检错码+纠错码

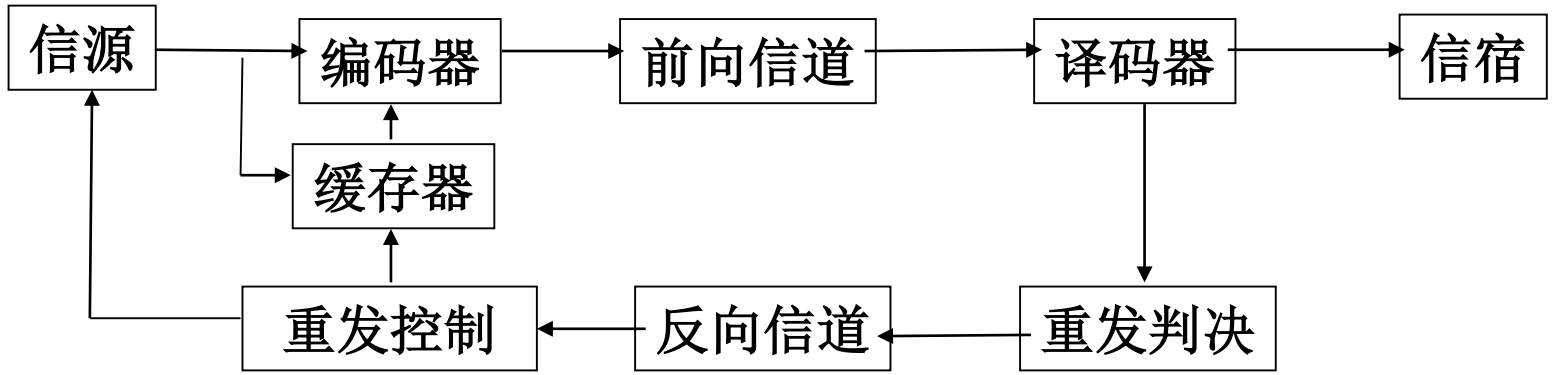
前向纠错FEC

- 采用纠错编码，其模型如下：

- 发送端对数据进行纠错编码，然后送信道传输
- 接收端对接收的数字信号译码，如果检测到错误则自动纠正。
- 优点：不需要反向信道，可用于单工通信，也可用于一点对多点通信
- 缺点：译码设备复杂，为纠正错误需要附加冗余码

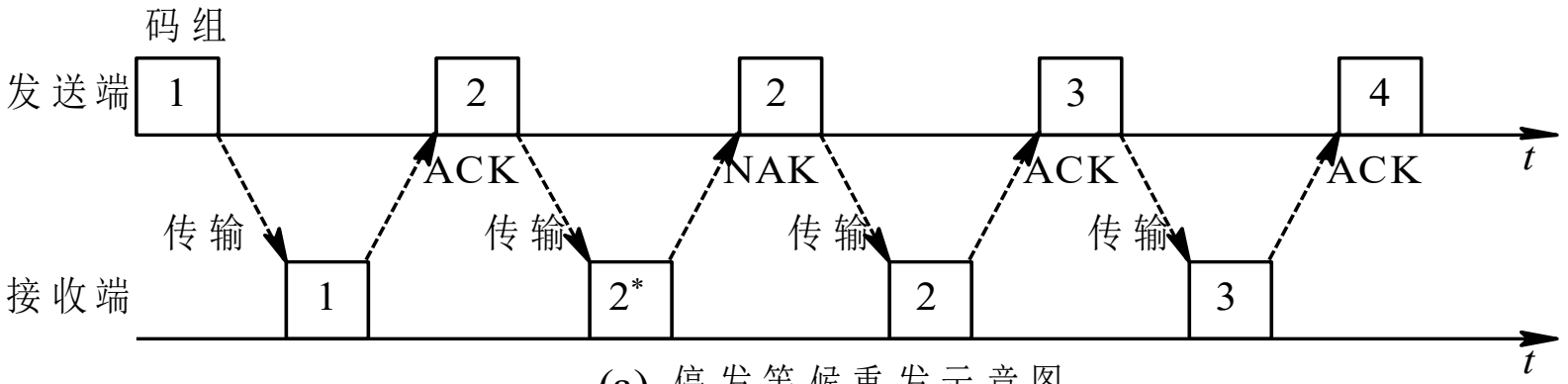
自动请求重传(ARQ)

- 自动请求重传(ARQ)：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发的方式。

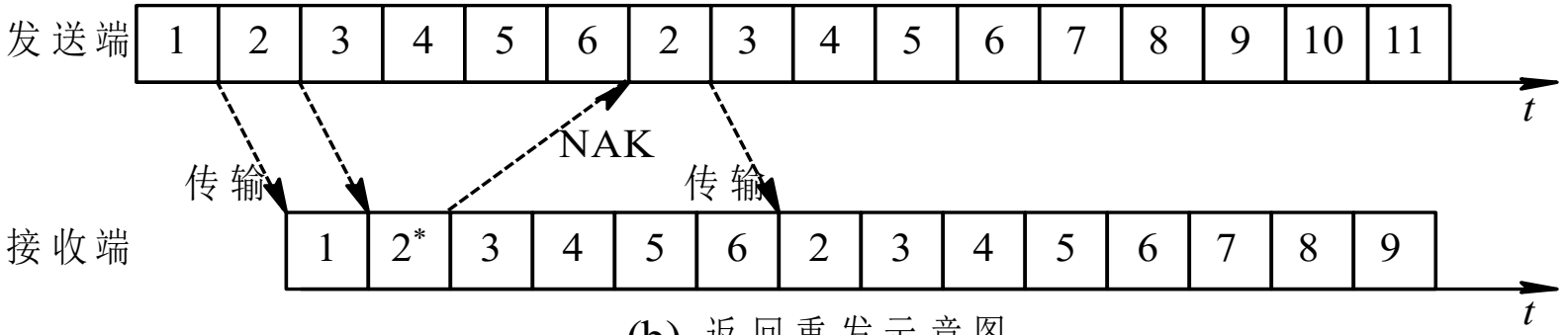


- 发送端将数据发向信道，同时缓存数据。如果接到重发请求，则将放入缓存器的数据重新编码发送。
- 接收端判决，如果接收正确则发送ACK，若检测到错误则发送NACK，要求重发。

ARQ的三种方式



(a) 停发等候重发示意图



(b) 返回重发示意图

(c) 选择重发示意图

小结

- 链路：一条点到点的物理线路，中间没有交换结点。分为有线、无线；
- 数据链路：包括物理线路以及控制数据传输的通信协议，由实现协议的硬件和软件组成。
- 成帧：面向字符的（如异步传输协议），面向比特的（如HDLC同步传输协议）
- 差错检测：奇偶校验，汉明码，CRC，交织码，互联网校验和等

练习题

- 数据链路层的基本功能？有哪些基本协议标准？
- 检错码和纠错码有何不同？
- 若字符编码为 A:01000111 B:11100011 FLAG:01111110 ESC:11100000 为传输4个字符的帧 A B ESC FLAG，请分别给出在字节计数、字节填充、比特填充等成帧方法下所发送帧的比特序列。

- 若采用字节填充的方法发送数据段 A B ESC C ESC FLAG FLAG D，请给出经过填充后的输出。

练习题

- 举例说明：两维奇偶校验能够纠正和检测1比特错，某些2比特错能被检测但不能纠正，某些4比特错无法检测。
- 若收到12位汉明码0XE4F，问该码的原始值是什么，设最多出现1位错，假设首位为最低位，采用偶校验。

- 使用CRC传输10011101，生成多项式为x^3+1（1）给出实际传输的比特序列（2）若第3个比特错，请说明在接收方可以被检测出（3）给出一个传输错误的实例，使得接收方无法检测出错误。