



第三章 数据链路层（一）

袁华, hyuan@scut.edu.cn

华南理工大学计算机科学与工程学院
广东省计算机网络重点实验室

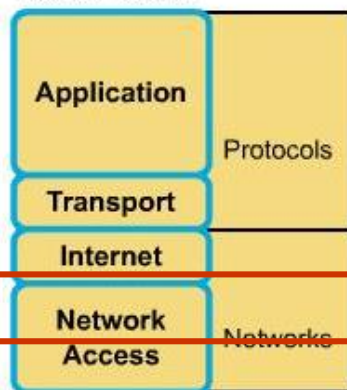
数据链路层的功能P151

- 为网络层提供服务，良好的服务接口P151
- 保证数据传输的**有效、可靠**：
 - 处理传输错误：差错检测和控制P156~157
 - 流量控制P157

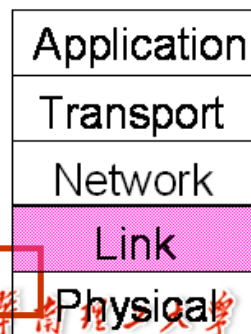
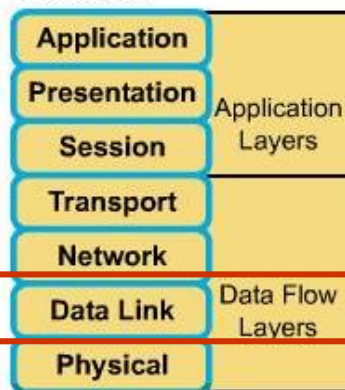
- 基于速率
- 基于反馈

Comparing TCP/IP with OSI

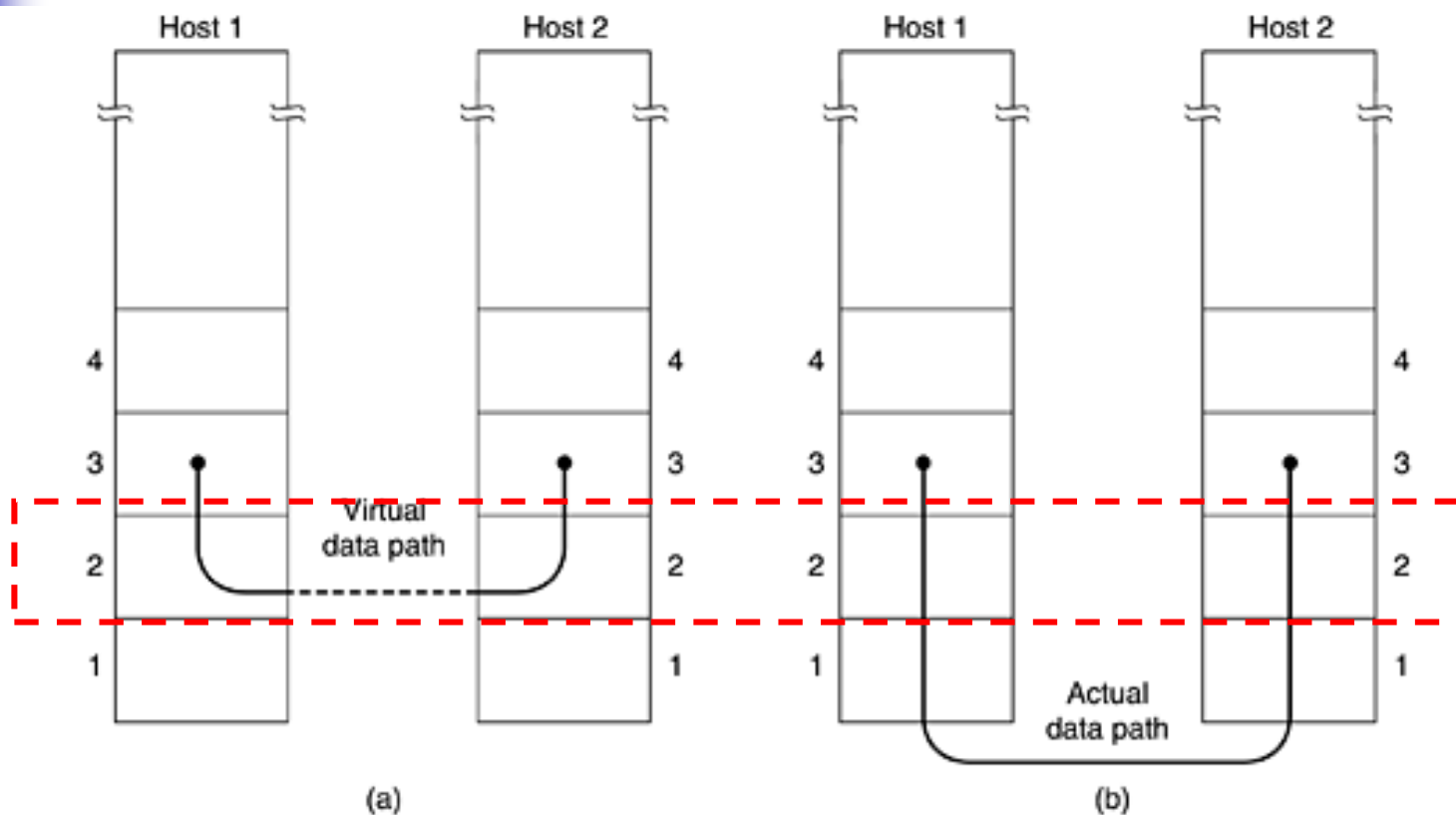
TCP/IP Model



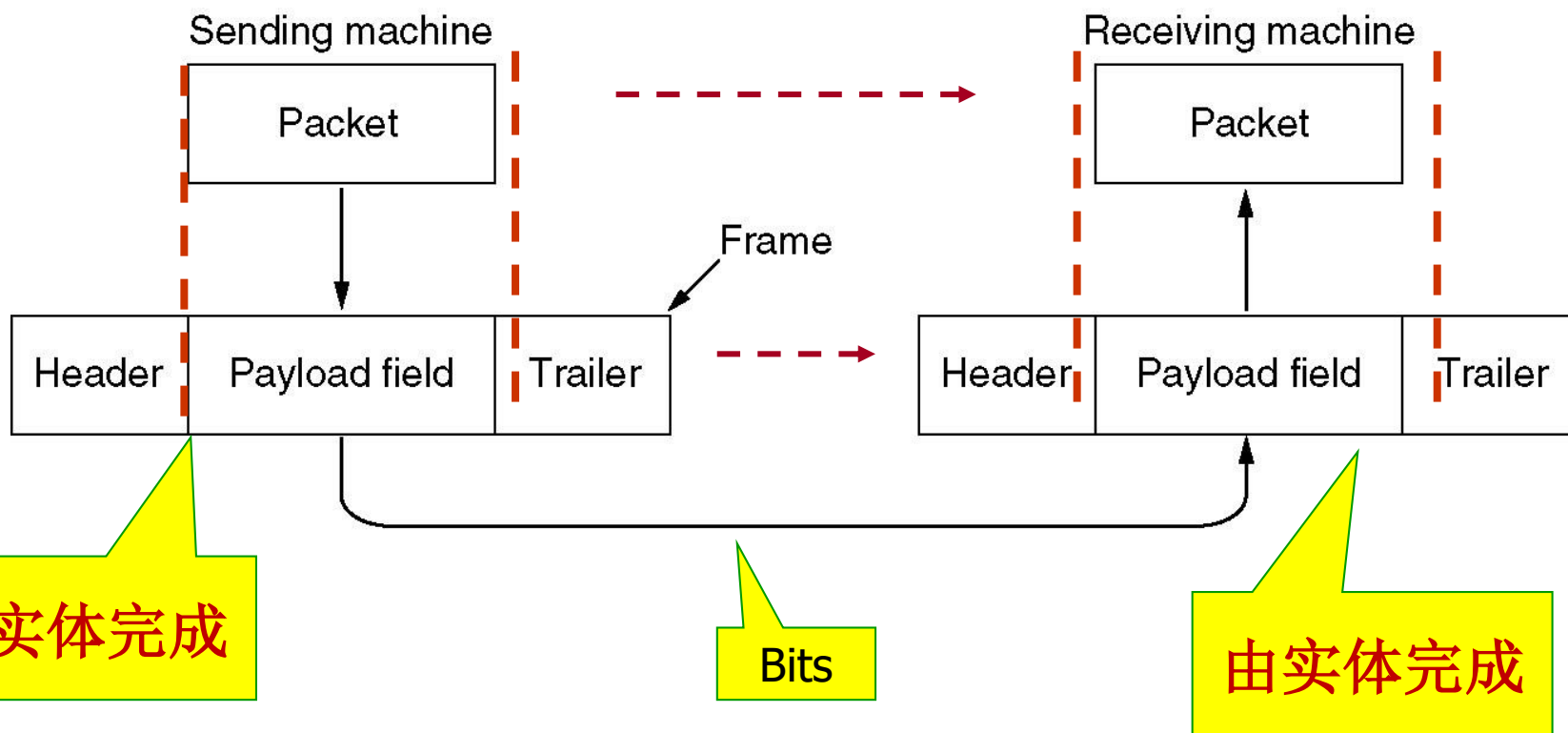
OSI Model



数据链路层的位置



帧和分组的关系 P151





本章的主要内容

- **DLL**层设计问题
- 检错和纠错
- 数据链路层的协议
 - 滑动窗口
- 数据链路层协议实例



本节目的

- 理解数据链路层功能
- 掌握成帧的方法
- 掌握重要的检错和纠错方法
 - 海明码（纠错）
 - 循环冗余码**CRC**（检错）



DLL提供的服务 (1/4) P152

- 确认：接收方在收到数据帧后，必须给发送方发回一个确认
- 面向连接：发送方和接收方在传输数据之前必须建立逻辑连接，传输结束后必须释放连接
- 服务种类P155
 - 无确认的无连接服务
 - 有确认的无连接服务
 - 有确认的面向连接服务



DLL提供的服务 (2/4)

■ 无确认的无连接服务 P152

- 无确认是指接收方在收到数据帧后，**毋需**发回一个确认
- 无连接服务是指在数据传输前**毋需**建立逻辑链路
- 无确认并非不可靠，可靠性由上层协议负责
- 局域网主要采用这种方式，因为：
 - 共享信道毋需建立连接
 - 信道较为理想，数据传输的误码率很低
 - 即使出错或丢失由上层负责恢复
- 适用于实时通信



DLL提供的服务 (3/4)

■ 有确认的无连接服务P153

- 使用前不建立连接，即不建立数据链路，但每帧传输必须得到确认
- 如果没有得到确认，将重传
- 这在信号传播延时较大、线路状态不一定很可靠的情况下是有效的
 - 例如：无线通信，信道使用率很低但数据传输的误码率相对较高，确认是必要的



DLL提供的服务（4/4）

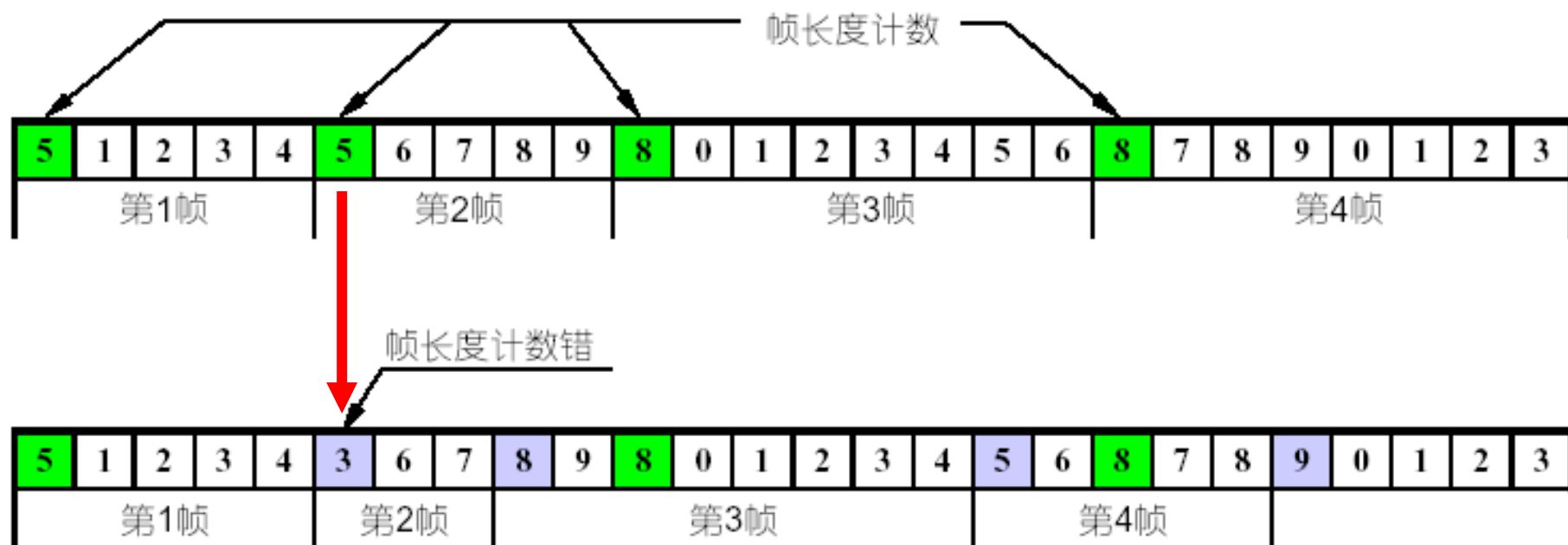
- 有确认的面向连接服务
 - 使用前**先建立**连接，即先建立数据链路，并且每帧的传输**必须得到确认**
 - 有连接的服务必须在使用前先建立连接（即建立逻辑链路），然后使用，最后释放
 - 例如：卫星信道、长途电话



成帧 P153

- 数据链路层使用物理层提供的服务，物理层处理的是位流，数据链路层处理的是帧
- 将原始的位流分散到离散的帧中，叫成帧，成帧的方法有：
 - 字符计数法
 - 带字节/字符填充的标志字节法
 - 比特填充的比特标志法
 - 物理层编码违例法

字符计数法





字符计数法成帧的特点 P158

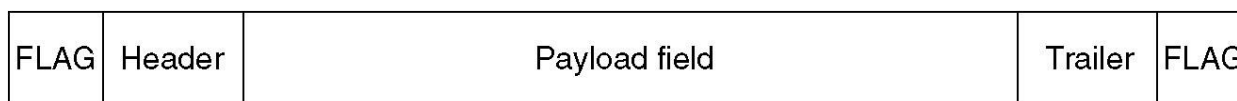
- 简单
- 缺点：无法恢复
- 很少被使用



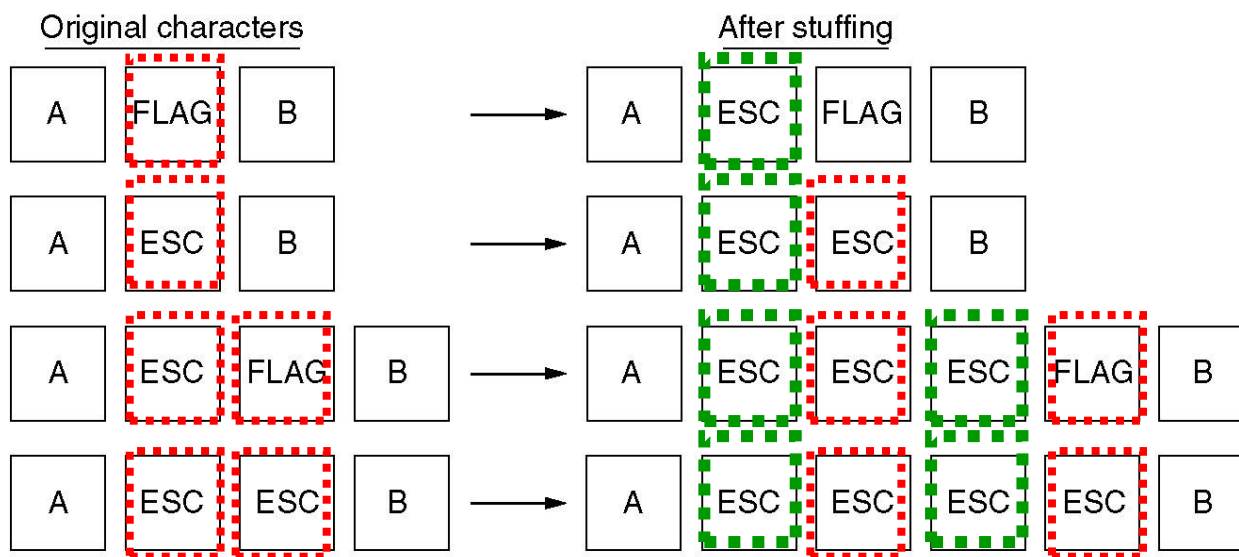
字节填充的标志字节法^{P158}

- 该方法考虑了错误之后重新同步的问题，让每一帧都用一些特殊的字节作为开始和结束
 - 标志/标记 字节 (**flag byte**)

字节填充的标志字节法^{P159}



(a)



(b)



字节填充的标志字节法的特点

- 缺点：
 - 容易造成帧界混淆 **P155**，——增加转义字节
 - 依赖于**8位**字符
- 解决方法
 - 一种方法是在二进制数中偶然出现的标志字节前插入一个转义字节。这就称为字节/字符填充法
 - 新技术的采用——位填充

比特填充的标志比特法 P155

- 这是一种面向二进制位的帧格式，把所有需传输的数据以比特位一字排开，并以特殊的位模式**01111110**作为帧标志，即一个帧的开始（同时标志前一个帧的结束）
- 当帧内容中出现一个与帧标志相同的位串**01111110**，则在**5个1**后插入一个**0**，即变成**01111101**，接收方将自动删除第**5**个**1**后的**0**。这称为**位填充法**，也称为透明传输。
- 如果由于干扰，一个帧没有正确接收，则可扫描接收串，一旦扫描到**01111110**，即新的一帧从此开始。即可以再同步

比特填充的标志比特法 (图3.5)

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Stuffed bits

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0



物理层编码违例法 P156

- 在曼切斯特编码中，连续高电平或连续低电平可用作帧边界
 - 采用冗余编码技术，如曼切斯特编码，即两个脉冲宽来表示一个二进制位
 - 数据**0**：低-高电平对
 - 数据**1**：高-低电平对
 - 高-高电平对和低-低电平对没有使用，可用作帧边界



差错控制和流量控制P156~157

- 帧的校验
- 接收端对帧的确认（确认帧）
- 超时与重发（计时器）
- 帧的序号（解决重复帧的问题）
- 流控
 - 基于反馈的流控
 - 基于速率的流控（发送端确定，在**DLL**中几乎不采用）



差错的类型 P158

- 单个错误：分散在各块中
- 突发错误：集中在某个块中
- 例子：
 - 块大小：**1000 bits**（**100**个块）
 - 出错率是：每个比特 **0.001**（**1‰**）
- 突发错误比单个错误更加难于纠正



差错的处理 P158

- 纠错码 ——前向纠错技术

因其需要太多的冗余位，纠错开销太大，在有线网络中极少使用，主要用于无线网络(Why?)中。

- 检错码 (不能恢复，可重传)

计算机网络中主要采用循环冗余码(CRC)。

- 两种不同的处理方法适用于不同的环境

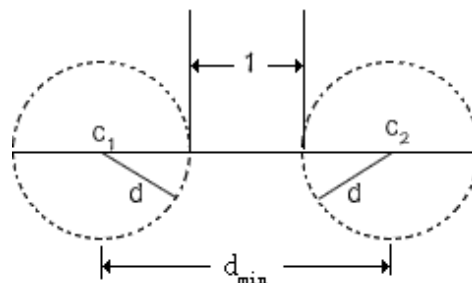
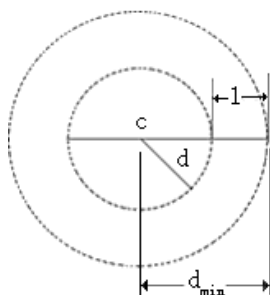
海明距离 (Hamming Distance)

- 码字：包含数据和校验位的n位单元。 P159
- 海明距离 P159
 - 两个码字(codeword)的海明距离：两个码字之间不同位的数目。
如：10001001 和 10110001 的海明距离为3。
 - 异或 的结果
- 全部码字的海明距离
全部码字中任意两个码字之间海明距离的最小值。
- 海明距离的意义：如果海明距离为d，则一个码字需要发生d个1位错误才能变成另外一个码字

海明距离与检错和纠错的关系

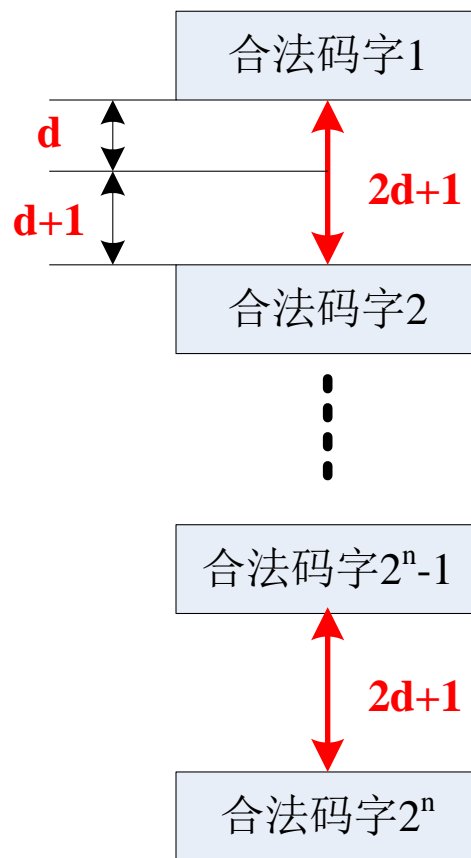
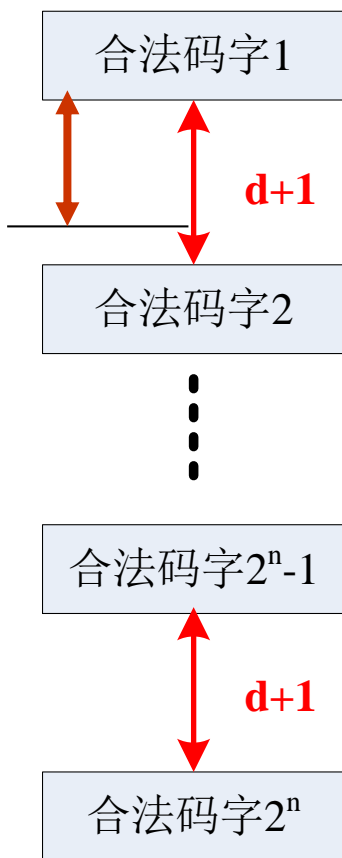
(1/2 P162)

- 海明距离为 $d+1$ 的编码能检测出 d 位差错。
 - 因为在距离为 $d+1$ 的检验码中，只改变 d 位的值，不可能产生另一个合法码。如奇偶校验码，海明距离为 2 ，能查出单个错。
- 海明距离为 $2d+1$ 的编码，能纠正 d 位差错。
 - 因为此时，如果一个码字有 d 位发生差错，它仍然距离原来的码字距离最近，可以直接恢复为该码。



海明距离与检错和纠错的关系

(2/2)



一个检错码的简单例子(奇偶位P183)

- 奇偶校验码
- 一个**校验位 (Parity Bit)**追加到数据后。
- 校验位的值取“0”还是“1”，取决于整个码字的总的“1”的个数。（奇数还是偶数）。
 - Data: 1011010
 - Even: 1011010 **0** （偶校验）
 - Odd: 1011010 **1** （奇校验）
- 海明距离等于 2。
 - 如果1个比特发生了跳变错误，可以检测出来。
 - 如果2个比特发生了跳变错误，接收方无法检测出错误，认为码字正确。

一个简单的具体的偶校验例子

□ 一个系统要传输的原始码字:

■ 00、01、10、11

□ 编码后的偶校验码字是: (海明距离2)

■ 000、011、101、110

□ 如果接收方收到一个码字001, 这是一个非法码字, 所以, 出错了。

□ 如果系统收到000, 这是一个合法码字, 正确。不排除这是110跳变2后的错误码字。

一个纠错码的例子

□ 一个系统有4个合法码字:

■ 0000000000, 0000011111, 1111100000, and 1111111111

□ 海明距离是 $5=2*2+1$, 所以可纠正2位错误

■ 发送: 0000011111

■ 接收: 0000000111

■ 收方纠正后: 0000011111

■ 发送: 0000000000

■ 接收: 0000000111

■ 收方纠正后: 0000011111

?? 发生3个错,
无法纠正!

注意

- ❑ 随着海明距离的增加，纠错的能力也增加；即海明距离越大，纠错能力越强。
- ❑ 海明距离为3，可以纠正1个错误；而海明距离为5，可以纠正2个错误。
- ❑ 当一个系统中的海明距离增加的时候，合法码字就减少了；即传输效率降低

检错: $d+1$
纠错: $2d+1$

矛盾!



纠正单比特错的冗余位下界^{P160}

- 冗余位（校验位）： r ；数据位： m
- 纠正单个错误需要的校验位的下界满足：

m	r	n （码字的总位数）
1	2	3
2~4	3	5~7
5~11	4	9~15
12~26	5	17~31
27~57	6	33~63
58~120	7	65~127

$$(n+1)2^m \leq 2^n$$

$$n = m + r$$



$$(m + r + 1) \leq 2^r$$

海明码纠错码 (1950年)



- 每一个码字从左到右编号，最左边为第**1**位
- 校验位和数据位
 - 凡编号为**2**的乘幂的位是校验位，如**1、2、4、8、16、.....**。
 - 其余是数据位，如**3、5、6、7、9、.....**。
- 每一个校验位设置根据：包括自己在内的一些位的集合的奇偶值(奇数或偶数)。

如何决定每个数据位的校验位

- 将某一位数据位的编号展开成**2**的乘幂的和，那末每一项所对应的位即为该数据位的校验位(收方使用)。

如： **$11 = 1 + 2 + 8$**

$29 = 1 + 4 + 8 + 16$

- 校验位**1**的检验集合为所有奇数位。
- 校验位**2**的检验集合：**2、3、6、7、10、11、...**
- 校验位**4**的检验集合：**4、5、6、7、.....**
- 校验位**8**的检验集合：**8、9、10、11、.....**

海明码实例

Char.

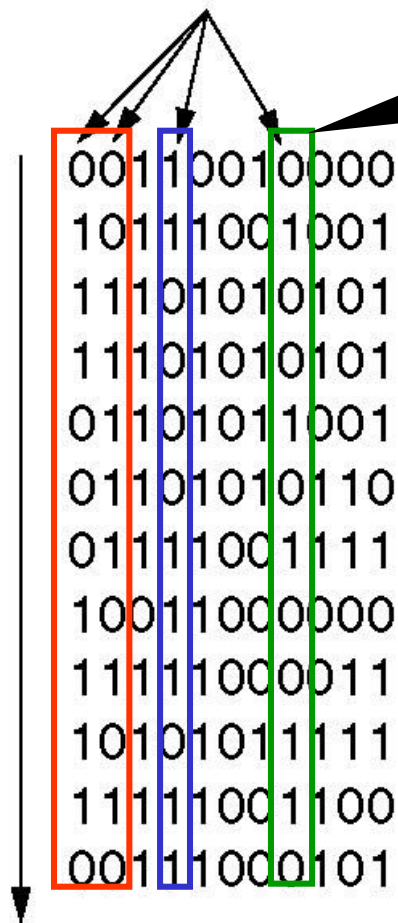
ASCII

**7位
数据位**

H	1001000
a	1100001
m	1101101
m	1101101
i	1101001
n	1101110
g	1100111
	0100000
c	1100011
o	1101111
d	1100100
e	1100101

Check bits

**如何确定
校验位？**



Order of bit transmission

校验位的计算($m=7, r=4$)

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11
	P1	P2	D1	P3	D2	D3	D4	P4	D5	D6	D7
$1=2^0$	√		√		√		√		√		√
$2=2^1$		√	√			√	√			√	√
$4=2^2$				√	√	√	√				
$8=2^3$								√	√	√	√

海明码实例之校验位计算

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11
	P1	P2	D1	P3	D2	D3	D4	P4	D5	D6	D7
信息码	-	-	1	-	0	0	1	-	0	0	0
检验位	0	0	-	1	-	-	-	0	-	-	-
海明码	0	0	1	1	0	0	1	0	0	0	0

使用偶校验，一个校验集合里的1的个数是偶数

课堂练习

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11
	P1	P2	D1	P3	D2	D3	D4	P4	D5	D6	D7
信息码	-	-	1	-	1	0	0	-	0	0	1
检验位	?	?	-	?	-	-	-	?	-	-	-
海明码	?	?	1	?	1	0	0	?	0	0	1

课堂练习（答案）

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11
	P1	P2	D1	P3	D2	D3	D4	P4	D5	D6	D7
信息码	-	-	1	-	1	0	0	-	0	0	1
检验位	1	0	-	1	-	-	-	1	-	-	-
海明码	1	0	1	1	1	0	0	1	0	0	1

海明码纠错过程(接收端)

- 首先将差错计数器置“0”，**counter=0**。
- 当海明码数据到达接收端后，接收端逐个检查各个校验位的奇偶性。
- 如发现某一校验位和它所检测的集合的奇偶性不正确，就将该检验位的编号加到差错计数器中。
 - **Counter=0**，无差错
 - **counter ≠ 0**，出错，该值指明出错的位

接收纠错实例P161

- 例如接收到码字为**00111000100**，校验各位：

- 第一位：00111000100，校验集合有3个1，错
- 第二位：00111000100，校验集合有1个1，错
- 第四位：00111000100，校验集合有2个1，对
- 第八位：00111000100，校验集合有1个1，错

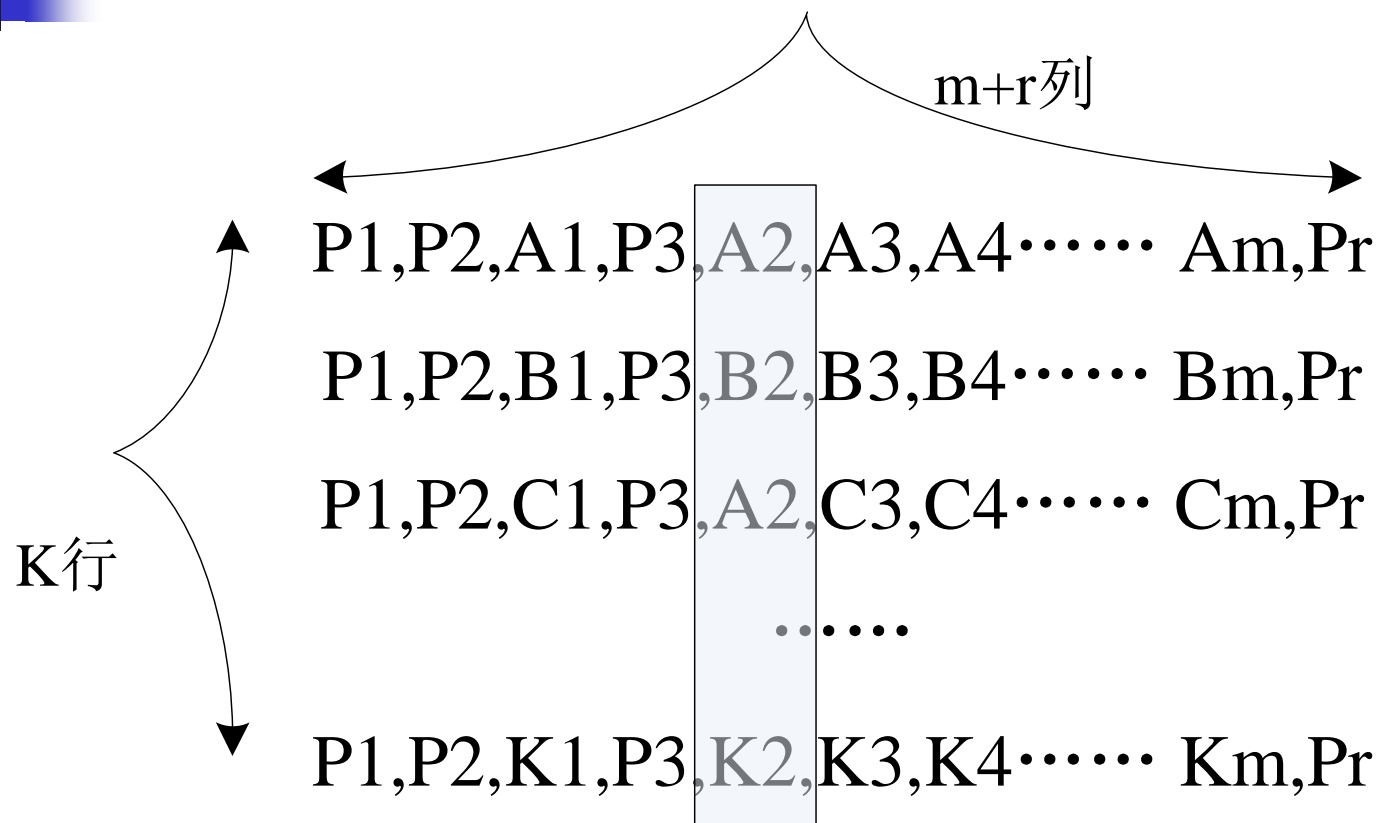
- 累加出错位编号：**1+2+8=11**

- 可计算得其第**11**位出错，将该位由**0**改为**1**，即纠正得到正确结果：**00111000101**

利用海明码纠正突发错误P163

- 将连续的 k 个码字按行排列成矩阵
- 发送数据时，按列发送，每列 k 位
- 如果一个突发性错误长度是 k 位，则在 k 个码字中，至多只有一位受到影响，正好可用海明码纠错改位后恢复

利用海明码纠正突发错误图示



课堂练习 (1/2)

- 原码字为：10101111，采用偶校验海明纠错1位错编码，请问编码后的码字是什么？

- 解答： $m=8$ ，根据 $(m+r+1) \leq 2^r$ 得：

$r=4$ ， ??1?010?1111

编码后码字是： 101001001111

课堂练习 (2/2)

- 采用上面这道题一样的编码，假设接收方收到一个码字：**1 0 0 1 1 0 0 0 1 1 0 0**
($m=8, r=4$)，请问这个码字对还是错？如果错，正确的码字应该是什么？
- 解答：计数器累加：
 $1+2=3$ ，所以，第3位出错，正确码字应该为：
1 0 1 1 1 0 0 0 1 1 0 0

计算题给出解题过程

- 解：将计数器置零，并检查每个校验位的校验集合是否正确：

$$P1=B1\oplus B3\oplus B5\oplus B7\oplus B9\oplus B11=\Sigma(1,0,1,0,1,0)=1$$

$$P2=B2\oplus B3\oplus B6\oplus B7\oplus B10\oplus B11=\Sigma(0,0,0,0,1,0)=1$$

$$P3=B4\oplus B5\oplus B6\oplus B7\oplus B12=\Sigma(1,1,0,0,0)=0$$

$$P4=B8\oplus B9\oplus B10\oplus B11\oplus B12=\Sigma(0,1,1,0,0)=0$$

所以，计数器 **Counter=1+2=3**，数据位第三位出错，正确的应该是码字和原始码字（数据位）分别是：

1 0 1 1 1 0 0 0 1 1 0 0 和 **11001100**



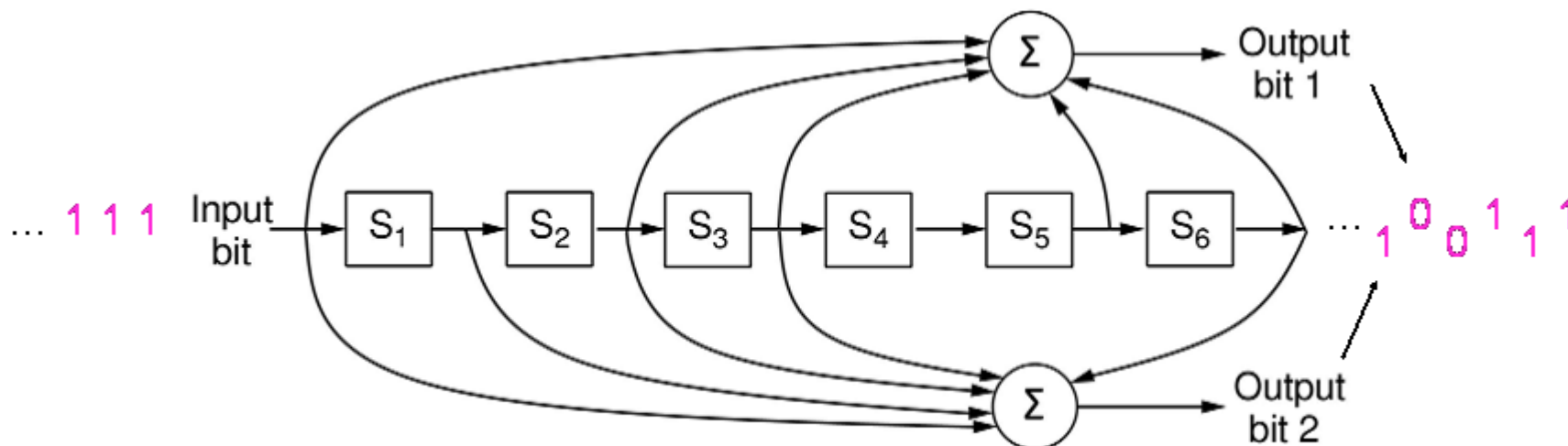
其它纠错码 P161

- **Binary convolutional codes** （二进制卷积码）
- **Reed-Solomon and Low-Density Parity Check codes** （里德所罗门码和低密度奇偶校验码）
 - **Mathematically complex, widely used in real systems**

卷积码 P161

操作位序列

- 输出序列是所有输入位的函数
- 采用**Viterbi**算法解码



Popular NASA binary convolutional code (rate = $1/2$) used in 802.11



检错码P163

- 检错码仅包含让接收方检查出是否有错误的足够信息
- 如果物理层的错误率足够低，采用检错和重传机制更加有效
 - 铜线或光纤

效率如何？ 一个例子 P163

- 考虑一根出错率为 每比特 10^{-6} 的信道，每个数据块包含 **1000** 比特
 - 为了纠单个错误，每块需要**10** 个校验位，传输 **1000** 块，共需要 **10,000** 校验位。
 - 为了检查单个错误，每块只需要一个校验位（奇偶校验）即可，要传输 **1000** 块，仅需要**1000** 位校验位，因为每比特 10^{-6} 的出错率，仅有一个块需要重传，总开销仅 **2001** ($= 1000 * 1 + 1001$) 比特



检错码P163

- 奇偶（海明距离为**2**，检**1**位错）
- 校验和 **P165**
- 循环冗余校验

检错码 – 奇偶校验

□ 奇偶位取值等同于对数据位进行模2和运算

■ 也等同于 XOR运算; (偶校验时)

■ 例如, 采用偶校验: 1110000 1110000**1**

■ 接收方检查是否存在单个比特的错误

□ 查出偶数个错误的简单方法

■ 例如: 1 error, 11100**1**01; 检出, sum is wrong

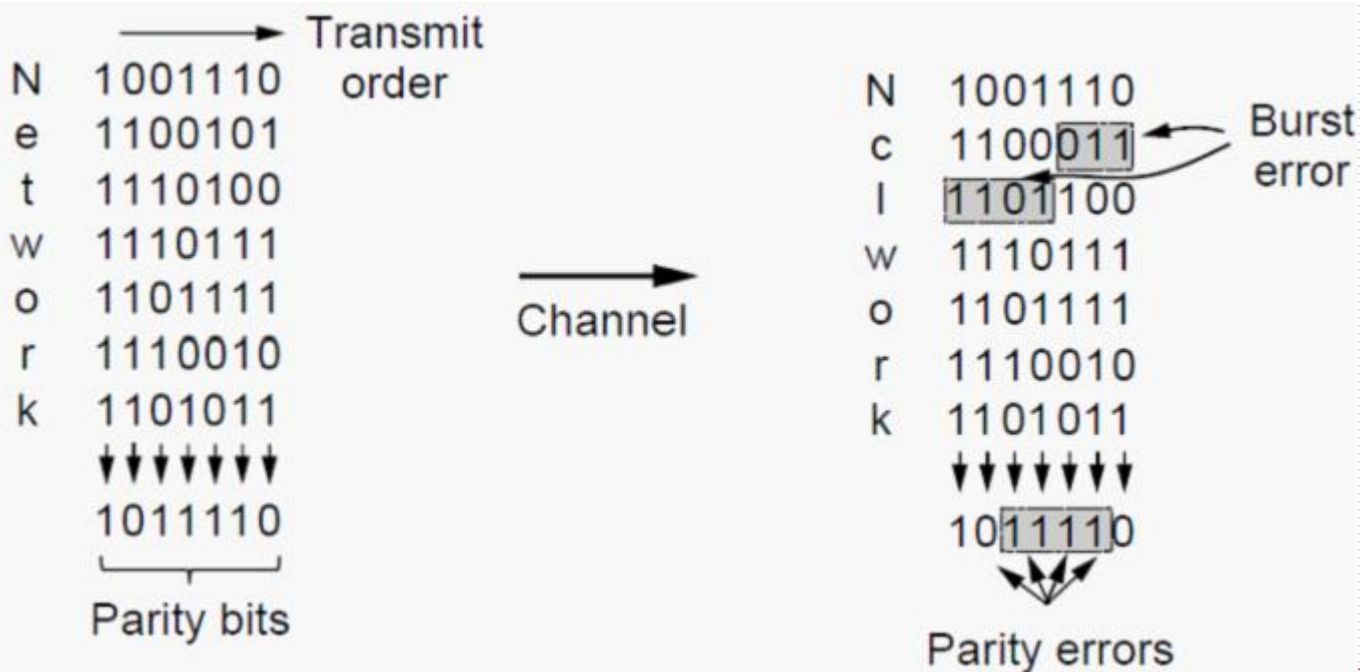
■ 例如: 3 errors, 11**01**1001; 检出, sum is wrong

■ 例如: 2 errors, 1110**11**01; 不能检出, sum is right!

■ 检出错误的概率为 $\frac{1}{2}$

检错码 – 奇偶校验(续)

- 交替的N位校验，可检查出最多N位的突发错误
- N位下的突发都可检出。



检错码 — 校验和

- 校验和通常是按照N位码字来进行模2和运算，发放将运算结果附加在数据报文尾部，作为校验位。
- 例如：16位的互联网补码校验和
- 特点：
 - 比奇偶检验更好的检错性能
 - 能检出高致N位的突发错
 - 检错随机错误率 $1-2^N$
 - 易受系统错误干扰, 比如, 增加的“0”

互联网校验和计算文档

□ RFC1071: computing the internet checksum

- (1) 待校验的相邻字节成对组成16比特的整数一行，按列从低位开始计算其模2和；并将结果按位取反码，作为校验和取值。
- (2) 检查校验和时，将所有字节，包括校验和，进行相加并求二进制反码。接收方：如果结果为全1，无错误
- 注意：如果某列的模2和有溢出，向高位进位，如果高位产生进位，循环向低位进位。

检错码—循环冗余检错码CRC

- 任何一个**k**位的帧看成为一个**k-1**次的多项式
 - 如：**1011001**看成 $x^6+x^4+x^3+x^0$
- 设定一个多项式编码生成多项式**G(x)**，**G(x)**为**r**阶
- 设一个**m**位的帧的多项式为**M(x)**，**m > r**，即**M(x)**比**G(x)**长
- 如 $x^r M(x) / G(x) = Q(x) + R(x) / G(x)$ 其中**Q(x)**为商、**R(x)/G(x)**为余数
- 则($x^r M(x) - R(x)$)一定能被**G(x)**整除，即余数为**0**
- 在二进制运算中，减法和加法都做异或运算即**0+1=1, 1+1=0**，**0-1=1**



一个十进制的仿真例子

- 收发双方约定，被“**3**”整除
- 发送方发送数字“**23**”， $23/3=7+2$ ，所以，发送 **$23-2=21$**
- 收方收到了“**21**”，用“**3**”除，刚好除尽，没出错。如果收到“**22**”，不能被“**3**”除，错了。
 - 但是，如果，收到了“**24**”？会怎样呢？

什么是模2运算？

◆ Modulo 2 addition & subtraction: XOR logic

$$0 \oplus 0 = 0; 0 \oplus 1 = 1;$$

$$1 \oplus 0 = 1; 1 \oplus 1 = 0.$$

– Modulo 2 multiplication:

$$\begin{array}{r} 1010 \\ \times \quad 101 \\ \hline 1010 \\ 0000 \\ 1010 \\ \hline 100010 \end{array}$$

-- Modulo 2 division:

$$\begin{array}{r} 101 \\ 101 \overline{) 10000} \\ \underline{101} \\ 010 \\ \underline{000} \\ 100 \\ \underline{101} \\ 01 \end{array}$$

CRC码计算举例

- 如一帧为**1101011011** (**m=10**)

即 $M(x) = x^9 + x^8 + x^6 + x^4 + x^3 + x + 1$

- $G(x) = x^4 + x + 1$ (**r=4**阶)

- $T(x) = x^4 M(x)$ (相当于在原码字后加**r**个**0**)

$$= x^4 (x^9 + x^8 + x^6 + x^4 + x^3 + x + 1)$$

$$= x^{13} + x^{12} + x^{10} + x^8 + x^7 + x^5 + x^4$$



帧数据

余数

1 0 0 1 1

1 0 0 1 1

1 0 0 1 1

1 0 0 1 1

0 0 0 0 1

0 0 0 0 0

0 0 0 1 0

0 0 0 0 0

0 0 1 0 1

0 0 0 0 0

0 1 0 1 1

0 0 0 0 0

1 0 1 1 0

1 0 0 1 1

0 1 0 1 0

0 0 0 0 0

1 0 1 0 0

1 0 0 1 1

0 1 1 1 0

0 0 0 0 0

1 1 1 0

余数

余数

CRC码计算举例（续）

计算**11010110110000**/**10011** 得余数**1110**

即**CRC码为11010110111110**

- 如**CRC码**在接收端能被**10011**整除则说明接收正确。
- 如发送方发送的 **$T(x)$** ，接收方收到的是 **$T(x)+E(x)$** ，如果不能被整除，则被检测到已出错。

接收端的检查

除数 $G(x)$

10011

XOR

XOR

XOR

XOR

1100001010

) 11010110111110

10011

10011

10011

10111

10011

10011

10011

0

被除数 $x^r M(x)$

余数



生成多项式国际标准

- **CRC-12:** $x^{12} + x^{11} + x^3 + x^2 + x^1 + 1$

用于字符长度为**6**位

- **CRC-16 :** $x^{16} + x^{15} + x^2 + 1$

用于字符长度为**8**位

- **CRC-CCITT :** $x^{16} + x^{12} + x^5 + 1$

用于字符长度为**8**

- **CRC32:** $G(X) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^3 + x + 1$



CRC计算算法P166

- 在数据帧的低端加上 r 个零，对应多项式为 $X^rM(x)$
- 采用模2除法，用 $G(x)$ 去除 $X^rM(x)$ ，得余数
- 采用模2减法，用 $X^rM(x)$ 减去余数，得到带CRC校验和的帧

CRC小结

■ Sender P165

- 在数据帧的低端加上 r 个零，对应多项式为 $X^rM(x)$
- 采用模2除法，用 $G(x)$ 去除 $X^rM(x)$ ，得余数
- 采用模2减法，用 $X^rM(x)$ 减去余数，得到CRC校验和的帧

■ Receiver

- 用收到的帧去除以 $G(x)$
- 为零：无错误产生
- 非零：发生了错误，重传



CRC课堂练习

- 如果生成多项式是 $G(x) = x^3 + x^2 + 1$ ，待传送的原始码字分别是 **1111** 和 **1100**，请计算采用**CRC**编码后的码字分别是多少？

□ 参考答案

■ 1111 111

■ 1100 101



本节小结

- 理解数据链路层功能
- 掌握成帧的方法
- 掌握重要的检错和纠错方法
 - 海明码
 - 循环冗余码**CRC**
 - 可以检测到所有长度小于等于 r 的突发错误
 - 广泛用于各种网络，几乎所有的局域网



谢谢!