第四章 数据链路层

课前思考

为什么建立链路连接？

链路连接与物理连接的联系和区别？

数据链路层上常用的差错控制编码有哪些？

数据链路层上常用的流量控制策略有哪些？

常用的数据链路层协议有哪些？

本章内容

- 4.1 差错控制
  - 4.1.1 差错的产生及特征
  - 4.1.2 编码效率、检错和纠错能力
  - 4.1.3 海明码
  - 4.1.4 循环冗余码
  - 4.1.5 其它差错控制编码
- 4.2 流量控制
  - 4.2.1 停—等协议
  - 4.2.2 滑动窗口协议
- 4.3 数据链路层协议举例
  - 4.3.1 HDLC协议
  - 4.3.2 PPP协议

4.1 差错控制

4.1.1 传输差错的特征

差错产生的主要原因
- 热噪声:传输介质内的分子热运动
- 冲击噪声：外界干扰

特征
- 热噪声:干扰幅度小，持续性　　对模拟通信影响大
- 冲击噪声：干扰幅度大，突发性　对数字通信影响大

计算机网络通信中，差错控制主要针对冲击噪声。

如数据率为9600bps，一次闪电持续时间为10ms，则连续破坏96位。

差错控制

- 差错控制方法
  - 通过特殊的编码（差错控制码），使接收端能够发现甚至自动纠正错误。
- 常用的差错控制编码有两类
  - 检错码
    - 能够发现差错，但无法自动纠正差错，通过发送方重传来获得正确的数据。
  - 纠错码
    - 不但能过发现差错，而且能够知道哪里出错，从而自动纠正差错

4.1.2 编码效率、检错和纠错能力

- 码字
  - 码字有信息位和校验位（冗余位）组成。
  - 设信息位为m位，校验位为r位，则码字长度为n=m+r
- 两个码字的距离
  - 两个码字的不同位数称为这两个码字的距离。
  - 例：10001001和10110001的距离为3。
- 海明距离
  - 给定某种编码算法，就能够造出包含全部合法码字的码字表（编码系统）。该码字表中必存在着两个码字之间的距离最小，这个最小距离称为该码字表（编码系统）的海明距离。
  - 海明距离决定了编码系统的检错和纠错能力

- 编码效率R=m/n=m/(m+r).

  信息位为m位，校验位为r位

  - 若检测d位出错，则海明距离至少为d+1.
  - 若纠正d位出错，则海明距离至少为2d+1.

4.1.3 海明码(纠错码)

1950年海明发明海明码 。设海明码的信息位有4位，记为$a_1a_2a_3a_4$；校验位为3位，记为$a_5a_6a_7$。 编码系统中任何一个合法的码字必须满足线性独立的方程：

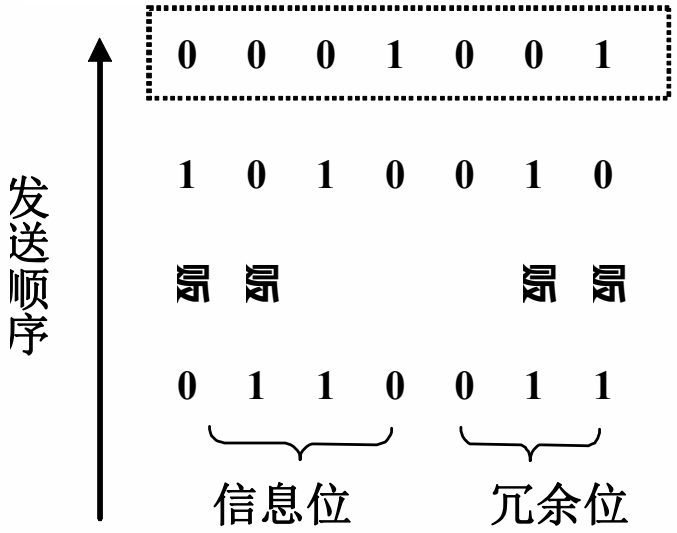
$$\begin{cases} a_5 = a_1 + a_2 + a_3 \\ a_6 = a_2 + a_3 + a_4 \\ a_7 = a_1 + a_3 + a_4 \end{cases} \quad (4\text{-}1)$$

注：+异或， 由式（4-1）构造的编码表如下：

0000000	0100110	1000101	1100001
0001011	0101101	1001110	1101000
0010111	0110001	1010010	1110100
0011100	0111010	1011001	1111111

显然，编码表的海明距离为3，能够自动纠正一位出错的编码

- 海明距离只能纠正一位出错，而实际通信过程中经常发生的是突发性错误（一连串位出错）。
- 要纠正这样的突发性出错，则必须加大海明距离；但加大海明距离势必会增加校验位长度，从而降低了编码效率。同时也会使编码系统过于复杂。
- 只要将发送方式稍做改变，就能利用纠正一位出错的海明码来纠正突发错。



设每次传输的数据块有k个码字组成，将这k个码字排列成一个矩阵，每行一个码字

若要纠正突发错，则按列发送，数据块到达接受端，再重新组成矩阵。

如果突发长度≦K,则每个码字最多出现一位错误，而前述的海明码正好能够纠正这样的一位错。

应用：ATM网对信元的自动纠错。

4.1.4 循环冗余码(检错码)

- 循环冗余码简称为CRC码(Cyclic Redunancy Code)

  是目前计算机网络中使用最广泛的一种检错码

- CRC码又称多项式码，每个码字对应于一个多项式。

  设码字为$a_1a_2a_3....a_n$，则对应的多项式为：

  $A(x)=a_1x^{n-1}+ a_2x^{n-2}+ a_3x^{n-3}+. ...+a_n$

  如 110001 $\rightarrow x^5+x^4+1$

编码原理（发送端）

- 设信息位串为$a_1a_2a_3....a_m$，则信息编码多项式为$M(x)=a_1x^{m-1}+ a_2x^{m-2}+ a_3x^{m-3}+. ...+a_m$
- 选择一个r次多项式G(x)作为生成多项式，按下面步骤生成校验串：
  - 在信息位串后补r个0，对应的多项式为$x^rM(x)$.
  - 用模2不借位除法，计算余数R(x)

    $R(x)= MOD(x^rM(x)/G(x))$
  - 要发送的码字多项式$T(x)=x^rM(x)+ R(x)$

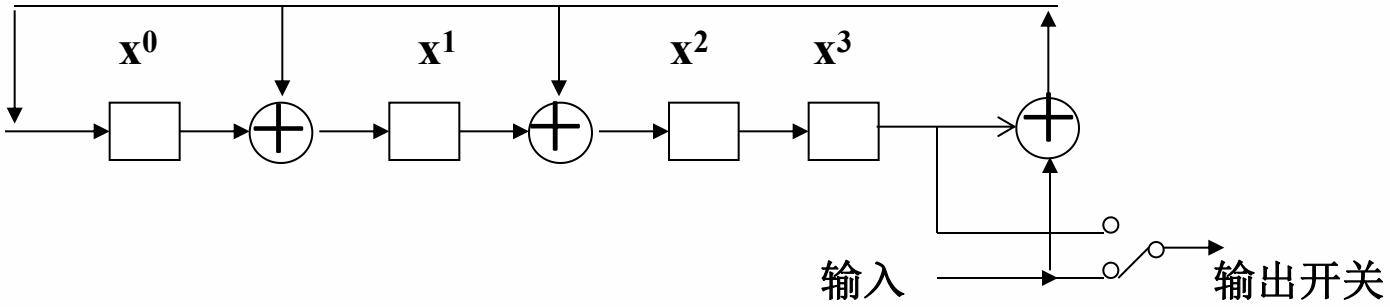
例：信息位串为1010001，若$G(x)= x^4+x^2+x+1$，求CRC码。

解：$M（x）= x^6+x^4+ 1$　　$r=4$

$x^rM(x)= x^{10}+x^8+ x^4$　　$\rightarrow 10100010000$

计算 $R(x)= MOD(x^rM(x)/G(x))$

编码电路



G（x）= x^4+x^2+x+1的编码电路

译码原理 （接收端）

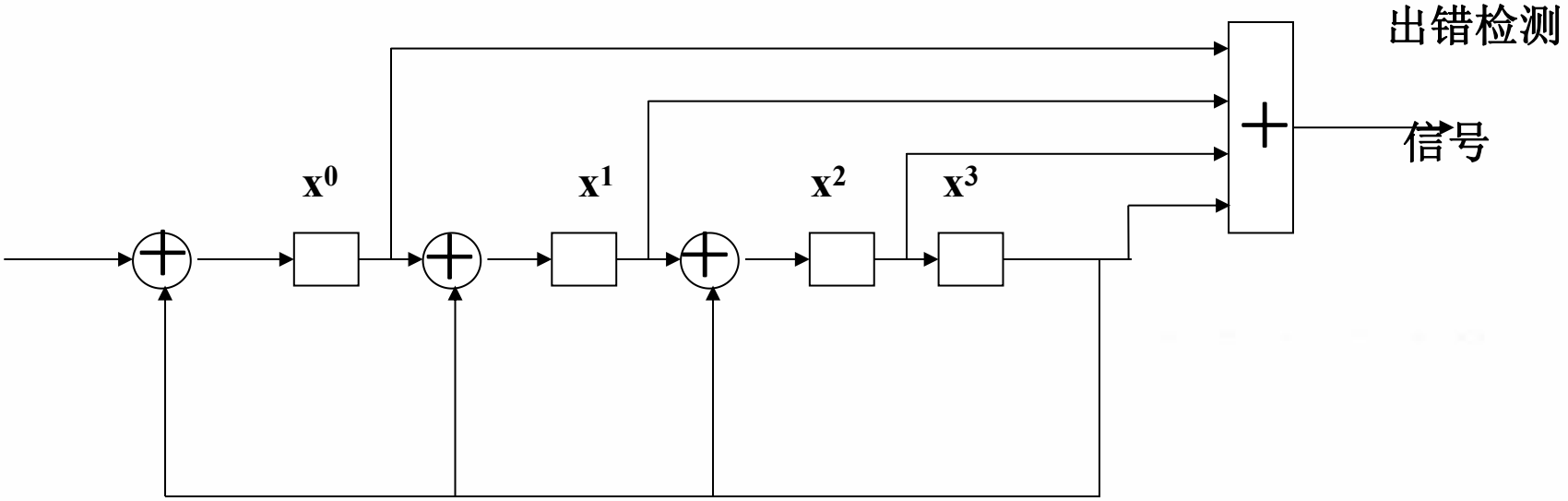
接受方收到一个码字后

用同一生成多项式G(x)除该码字多项式T′(x)

若MOD(T′(x))/G(x))=0则正确

若MOD(T′(x))/G(x)) ≠0则出错，要求重发

译码电路（ G（x）= x^4+x^2+x+1 ）

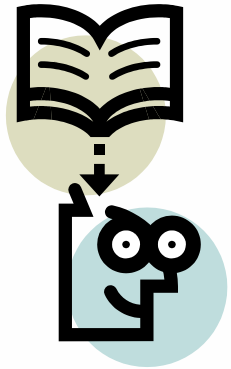


CRC码检错能力

思考：

1.若MOD(T′(x))/G(x))=0，是否一定正确

2.若MOD(T′(x))/G(x)) ≠0，是否一定出错



CRC码不能100%的发现错误，

当余数为“0”时可能发生错误。

CRC检错率取决于生成多项式G(x)

生成多项式性质

- 若G(x)中含有x+1因子，则能检测出所有的奇数位错。
- 若G(x)中不含有x因子，或者说，G(x)含有常数项1，那么能检测出所有突发长度≦r的突发错。
- 若G(x)中不含有x因子，且对任何0<e≦n-1 的e,除不尽x^e+1,则能检测出所有的双位错。
- 若G(x)中不含有x因子，则对于突发长度为r+1的突发错误的漏校率为$2^{-(r-1)}$.
- 若G(x)中不含有x因子, 则对突发长度大于r+1的突发错误的漏校率为2^{-r}.

- 三个标准CRC生成多项式：

$CRC\text{-}12 = x^{12}+x^{11}+x^{3}+x^{2}+x+1$

$CRC\text{-}16 = x^{16}+x^{15}+x^{2}+1$

$CRC\text{-}CCITT = x^{16}+x^{12}+x^{5}+1$

4.1.5 其它差错控制编码

- 奇偶校验码
- 定比码
  - 指定每个码字中均含有相同数目的“1”
  - 编码效率
    - $R=\log_2 C_n^m/n$ (n为码字的长度，m为“1”的数目。)
    - 编码效率较低。
  - 检错能力
    - 除了码字中“1”变为“0”和“0”变为“1”成对出现外，其余所有差错都能被检测出来，
- 正反码

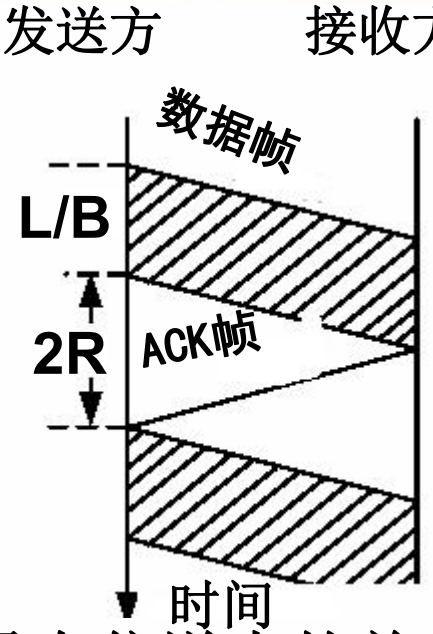
4.2 流量控制

4.2.1 停—等协议

- 发送方发送一帧后，等待对方的应答。
- 接收端收到一帧后，检查校验位串。若出错，返回“否认”信息；若无错，返回“确认”信息。
- 发送端收到“确认”后，立即发送下一帧；收到“否认”则重发该帧。
- 发送端发送一帧后，立即启动超时计时器。若超时中断，重发该帧。
- 接收端应保存最近收到的帧序号，若下一个到达帧的序号与该序号相同，则丢弃并返回“确认”信息。

停—等协议

- 缺点
  - 信道利用率低
- 优点
  - 简单
- 信道最大利用率 $U=\frac{L/B}{L/B+2R}$



B为信道速率，L为帧长，R为信号在信道中的单程传播延时，U为信道的最大利用率。

如考虑由于差错造成的重发，以及帧头、校验和冗余信息，信道实际利用率达不到最大利用率，实际利用率见P

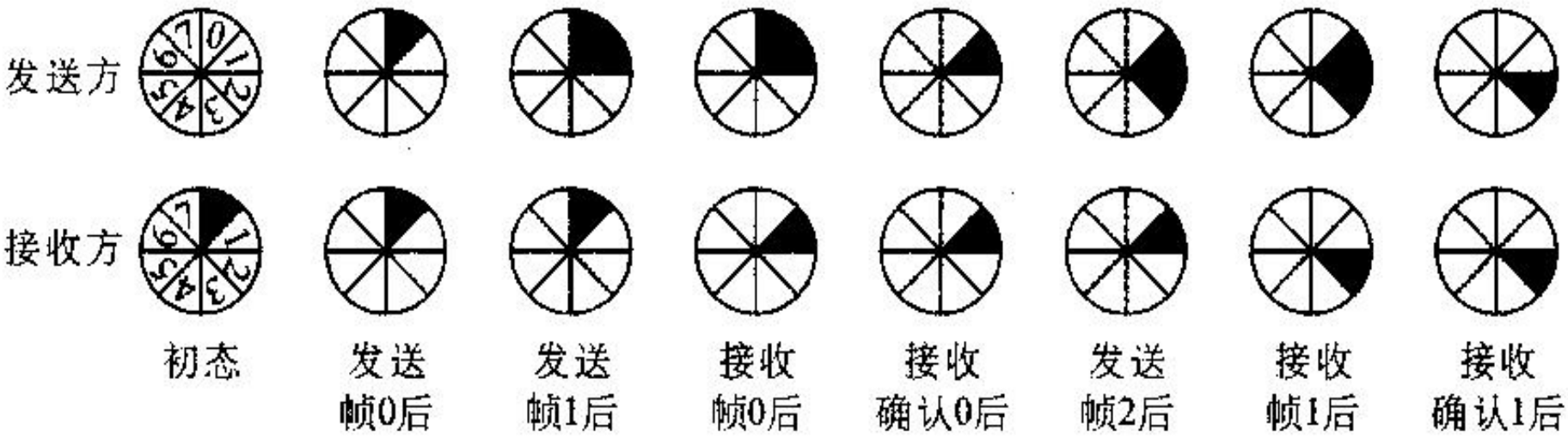
4.2.2 滑动窗口协议

- 基本思想
  - 为提高信道利用率，允许发送方连续发送若干帧后再等待对方应答。
- 基本概念
  - 窗口：可容纳数据帧的缓冲区。
  - 发送窗口：发送方用来保存已发送但尚未经确认的数据帧。
  - 接收窗口：接收方用来保存已正确接受但尚未提交给主机(网络层)数据帧。
  - 窗口尺寸：窗口中可以保存的帧数目称为窗口尺寸。

- 帧序号
  - 为了保证接收方能按正确次序向主机递交数据帧而设立的临时帧序号。
  - 一般在帧控制字段中用若干位来表示帧序号。如果用3位来表示，则帧序号为0—7。当一次通信超过8帧时，则顺序重复使用这8个帧序号。
- 窗口号：对应帧序号。

滑动窗口协议的基本规则

- 基本规则



W_T=2, 接收窗口尺寸W_R=1.

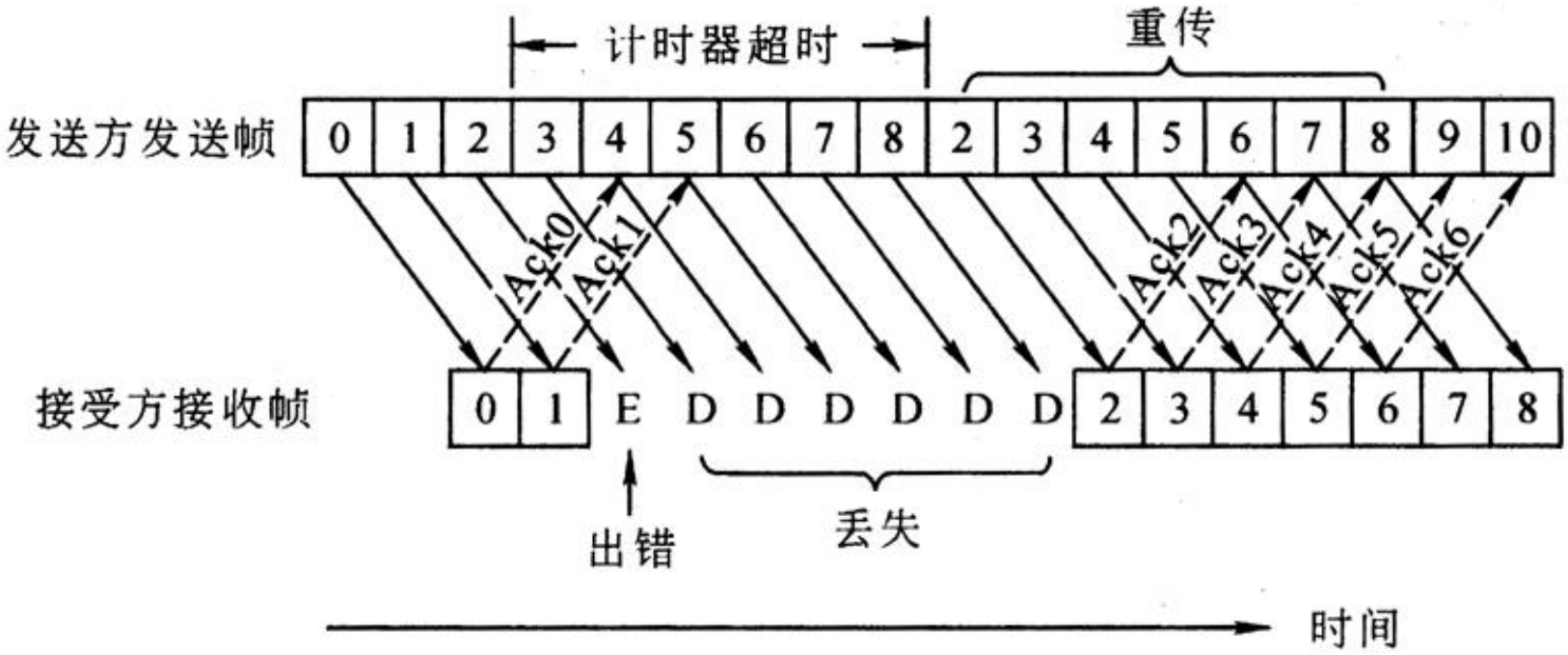
顺序接收管道协议（回退n）

接收窗口尺寸为1的滑动窗口协议，也称回退n协议。

设发送窗口尺寸W_T=n, 接收窗口尺寸W_R=1.

- 发送方可连续发送n帧而无需对方应答，但需要将已发出但尚未收到确认的帧保存在发送窗口中，以备由于出错或丢失而重发。
- 接收方将正确的且帧序号落入当前接收窗口的帧存入接收窗口，同时按序将接收窗口的帧送交给主机（网络层）。出错或帧序号未落入当前窗口的帧全部予以丢弃。
- 当某帧丢失或出错时，则其后到达的帧均丢弃，并返回否认信息，请求对方从出错帧开始重发。
- 发送方设置一个超时计时器，当连续发送n帧后，立即启动超时计时器。当超时计时器满且未收到应答，则重发这n帧。

回退n


计时器超时
重传
发送方发送帧
0 1 2 3 4 5 6 7 8 2 3 4 5 6 7 8 9 10
Ack0
Ack1
Ack2
Ack3
Ack4
Ack5
Ack6
接受方接收帧
0 1 E D D D D D D 2 3 4 5 6 7 8
出错
丢失
时间

选择重传协议

- 顺序接收管道协议
  - 优点：仅需一个接收缓冲区
  - 缺点：当信道误码率较高时，会产生大量重发帧
- 另一种更好的方法：选择重传协议
  - 若某一帧出错，后面正确到达的帧虽然不能立即送网络层，但接收方可将其保存在接收窗口，仅要求发送方重传那个发错帧。其工作原理如P

小结

- 停—等协议 、顺序接收管道协议、选择重传协议都可以看成是滑动窗口协议，其差别仅在窗口的尺寸不同。如下表所示

协议	发送窗口	接收窗口
停—等	1	1
回退n	>1	1
选择重传	>1	>1

窗口尺寸受到的限制

- 帧序号的位数为m，则

  $W_T \geq W_R$

  $W_T + W_R \leq 2^m$

分析：

- 若$W_R > W_T$会有 $W_R - W_T$个窗口永远用不上。
- $W_T + W_R \leq 2^m$保证了上一轮帧序号和下一轮序号在$W_T + W_R$范围内不会出现重复，否则接收端无法判断落入窗口的帧是上轮重发的还是新的帧。

4.3.1 HDLC协议

HDLC(High Level Data Control)协议是一种面向比特的链路层协议。

- 所谓“面向比特”是指以二进制位作为数据帧的基本数据单位。
- HDLC是ISO在IBM的SDLC(Synchronous Data Link Control)的基础上制定的。
- 该协议已成为链路层协议的典型代表。

HDLC帧格式

	标志	地址	控制	数据	帧检验	标志
字节数	1	1	1	任意	2	1

- 标志
  - 固定为**01111110**，标志着一个帧的开始和结束。
  - 具有帧之间的同步作用。
  - 在连续发送多帧时，可用一个标志字段，既表示帧的开始，又表示帧的结束。

插“0”技术

为了避免其它字段中出现“01111110”，产生误解，HDLC采用插“0”技术

发送方：除标志位外，连续发现5个“1”后自动插“0”。

接收方：连续发现5个“1”后

- 其后为“0”，则自动去掉该“0”。
- 其后为“1”，则检查下一位
  - 为“0”则为标志位
  - 为“1”则出错

“0”的插入与删除

数据中某一段比特组合恰好出现标志字段

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

会被误认为是标志 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

在此位置删除填入的 0 比特

- 控制：该字段表示帧类型，帧编号及其他控制信息。

帧类型
- 信息帧：含有要传输的数据
- 监督帧：应答帧
- 无编号帧：不带编号和数据，相当于控制帧
  - SNRM（置正常响应模式）
  - UA（无编号帧确认）
  - FRMR（帧拒绝）
  - DISC（拆除连线）
  - RESET(复位)等。

控制字段的格式

1	2	3	4	5	6	7	8	
0	N(S)			P/F	N(R)			信息帧“0”打头
1	0	S		P/F	N(R)			监督帧“10”打头
1	1	M		P/F	M			无编号帧“11”打头

N(S):表示信息帧的帧序号**0-7**，以便标识信息帧的发送顺序。
N(R):接收端期望接收的下一帧的序号。
P/F：轮询/结束位，用于多点轮询访问方式。

HDLC帧格式

S
- 00:确认以前各帧，准备接受后继帧。
- 10:确认以前各帧，但暂停接收后继帧，用来进行流量控制。
- 01:否认N(R) 起的各帧，请求重发从N(R)开始的各帧。
- 11:仅否认 N(R)帧，请求重发N(R)那一帧。

M：共5位，表示$2^5=32$种控制功能。

数据：要传输的数据，可以是任意二进制位的组合，即高层的报文分组。

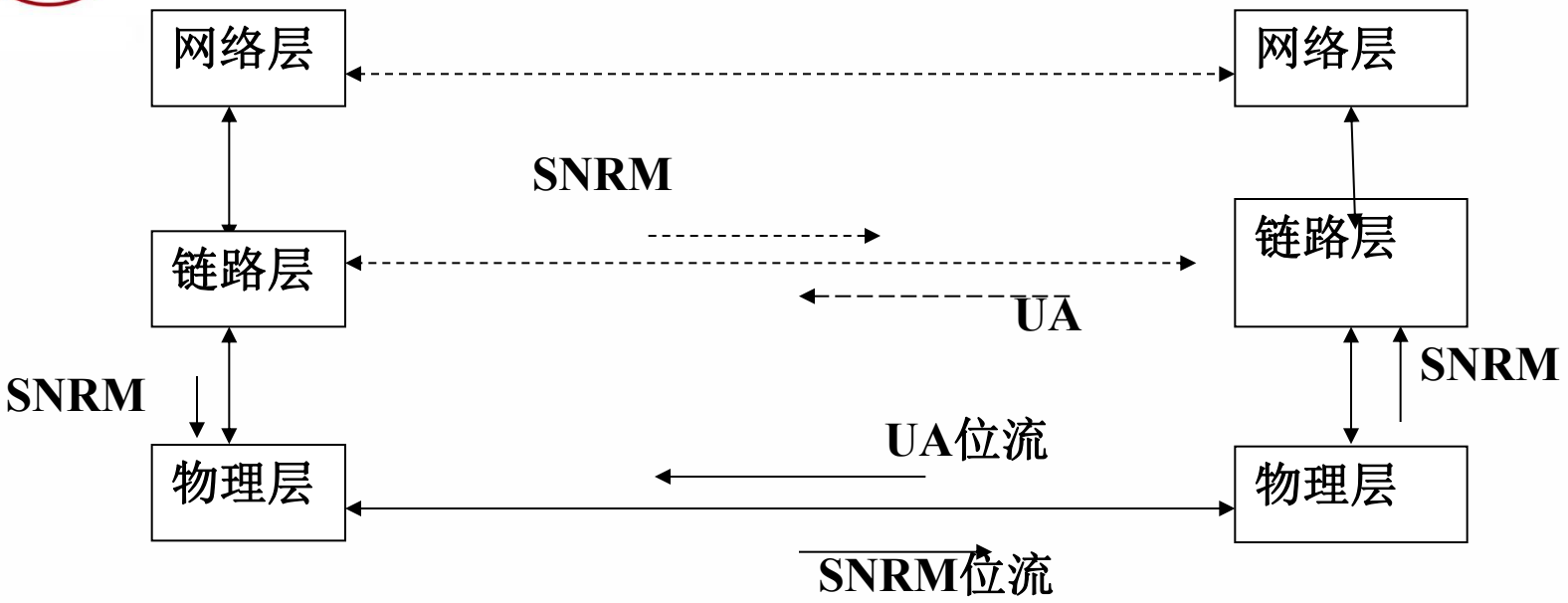
帧校验：16位CRC码，$G(X)=CRC\text{-}CCITT= x^{16}+x^{12}+x^5+1$

HDLC工作原理

分三个阶段:

- 建立数据链路连接
- 传输数据帧
  - 当数据链路建立完毕，发送/接收方按照某种流量控制策略发送和接收数据帧，并允许捎带应答。
- 拆除链路连线
  - 全部数据发送完毕，发送方发出**DISC**（拆除连接）无编号帧，接收方返回**UA**作为响应，此时释放链路层实体占用的资源。

建立数据链路连接



- 请求建立物理连接
- 请求建立链路连接，即发送 **SNRM**
- 接收方同意建立链路连接，即发送 **UA**

本章小结

- 内容
  - 主要介绍数据链路层协议以及相关技术，如数据链路层的功能、差错控制、流量控制、滑动窗口协议，以及经典的数据链路层协议HDLC和PPP的工作原理、数据帧格式及各字段的含义等。
- 重点
  - 差错控制和滑动窗口协议
- 难点
  - 滑动窗口协议