



第四章 数据链路层

课前思考

为什么建立链路连接？

链路连接与物理联系的联系和区别？

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 数据链路层的功能**
- **4.2 差错控制**
 - 4.2.1 差错的产生及特征
 - 4.2.2 编码效率、检错和纠错能力
 - 4.2.3 海明码
 - 4.2.4 循环冗余码
 - 4.2.5 其它差错控制编码
- **4.3 流量控制**
 - 4.3.1 停—等协议
 - 4.3.2 滑动窗口协议
- **4.4 数据链路层协议举例**
 - 4.4.1 HDLC协议
 - 4.4.2 PPP协议
- **4.5 数据链路层设备**



4.1 数据链路层的功能

- 链路是数据传输中两个节点间一条点到点的物理线路段，链路间不存在其他交换节点（如交换机、路由器）。
- 计算机之间的网络连接，通常是由多个链路组成的，把用来实现控制数据传输规程的一些硬件和软件加到链路上就构成了数据链路。
- 为此，数据链路除必须具有一个物理线路外，还需要一些必要的规程来控制数据的传输。



4.1 数据链路层的功能

- 物理层只负责比特流的接收和发送，而不考虑信息本身的意义，同时物理层也不能解决数据传输的控制，而这就是数据链路层所要解决的，即通过制定一些数据链路层协议来解决这些问题。
- 将物理层可能出错的原始比特流传输连接改造成为逻辑上无差错的数据链路。
- 概括来讲，数据链路层的功能：实现系统实体之间二进制数据单元的正确传输，通过必要的同步控制、差错控制、流量控制，为其上层（网络层）提供可靠、无差错的数据信息。



4.1 数据链路层的功能

- 将物理层可能出错的原始比特流传输连接改造成为逻辑上无差错的数据链路
 - 帧同步
 - 将比特组合成帧(**frame**)，即数据链路层的传送单位
 - 差错控制
 - 处理传输中出现的差错
 - 流量控制
 - 调节发送方的发送速率，以免较慢的接收方不能承受
 - 链路管理
 - 数据链路层连接的建立、维持和释放



帧同步

● 帧同步

- 在数据链路层中，数据的传输单位是帧。
- 每个帧除了要传送的数据外，还包括校验码，以使接收方能发现传输中的差错。
- 帧同步即接收端应当能从收到的比特流中准确地识别一个帧的开始和结束，并且让发送方将在传输中出错的帧重新发送（重传），这样可避免重新传输所有的数据。



差错控制

● 差错控制

- 为了将比特流在传输中的差错率控制在一定的范围之内，广泛地采用了各种编码技术。其中，编码技术可分为三类：
 - 向前纠错，当接收方收到出错的数据帧后，能够自动将错误修正过来。这种方法的系统开销大，不适合于计算机网络；
 - 自动请求重发，即接收方检错到错误的帧后，就通知发送方重新发送该帧，直到收到正确的帧为止。这种方法的系统开销较小，通常应用于计算机网络中。
 - 混合纠错（HEC），当码字中的差错个数在纠正能力以内时，直接进行纠正；否则，请求发送方重发来纠正错误。



前向纠错

- 前向纠错**FEC** (**Forward Error Correction**)
 - 发送方将要发送的数据附加上一定的冗余纠错码一并发送
 - 接收方则根据纠错码对数据进行差错检测
 - 如发现差错，由接收方进行纠正



自动请求重发

- 自动请求重发 (**Automatic Repeat request, ARQ**)
 - 发送方将要发送的数据附加上一定的冗余检错码一并发送;
 - 接收方根据检错码对数据进行差错检测;
 - 如发现差错, 则接收方返回请求重发的信息, 发送方在收到请求重发的信息后, 重新传送数据;
 - 如没有发现差错, 则发送下一个数据;
 - 为保证通信正常进行, 还需引入**计时器** (防止整个数据帧或反馈信息丢失) 和**帧编号** (以防止接收方多次收到同一帧并递交给网络层)。



流量控制

●流量控制

- 为防止接收端缓存能力不足而造成的阻塞和数据丢失，当接收方来不及接收时，接收方必须及时控制发送方发送数据的速率，这就是流量控制功能。
- 流量控制就是使发送方和接收方的数据处理速率保持一致。
- 数据链路层常用的流量控制方法有停等协议和滑动窗口机制。



链路管理

● 链路管理

- 当网络中的两个节点间要进行通信时，数据的发送方必须知道接收方是否处于准备接收状态。在传输数据之前通信双方必须事先交换一些必要的信息，让通信双方做好数据发送和接收的准备。
- 在传输数据时要维持数据链路，而在通信完毕时要释放数据。
- 数据链路的建立、维持和释放就叫做链路管理。



4.2 差错控制

4.2.1 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



4.2.2 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 $2d+1$.

例：设有一编码系统的码字表有4个码字组成：

0000000000 0000011111

1111100000 1111111111

- ✓ 海明距离为5，则可检测出4位出错，纠正两位出错。
- ✓ 如果收到一个码字：0000000111，判断出错。
- ✓ 如果出错位不超过两位，则可以断定其正确码字为0000011111
- ✓ 如果出错位超过两位（如3位），则无法断定其正确码字是0000011111，还是0000000000。



4.2.3 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：

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

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

16



设发送方发出一个码字 $a_1a_2a_3a_4a_5a_6a_7$,
接受方收到的码字为 $b_1b_2b_3b_4b_5b_6b_7$ 。

$$\text{再令} \left\{ \begin{array}{l} s_1 = b_1 + b_2 + b_3 + b_5 \\ s_2 = b_2 + b_3 + b_4 + b_6 \\ s_3 = b_1 + b_3 + b_4 + b_7 \end{array} \right. \quad (4-2)$$

其中 s_1 s_2 s_3 称为校验因子。

接受方将根据校验因子来判断哪位出错。



根据式（4-3）判断，
如果7个码位均不出错，则

$$\begin{cases} s1=0 \\ s2=0 \\ s3=0 \end{cases}$$

出错位与校验因子

出错位	s1	s2	s3
无	0	0	0
1	1	0	1
2	1	1	0
3	1	1	1
4	0	1	1
5	1	0	0
6	0	1	0
7	0	0	1

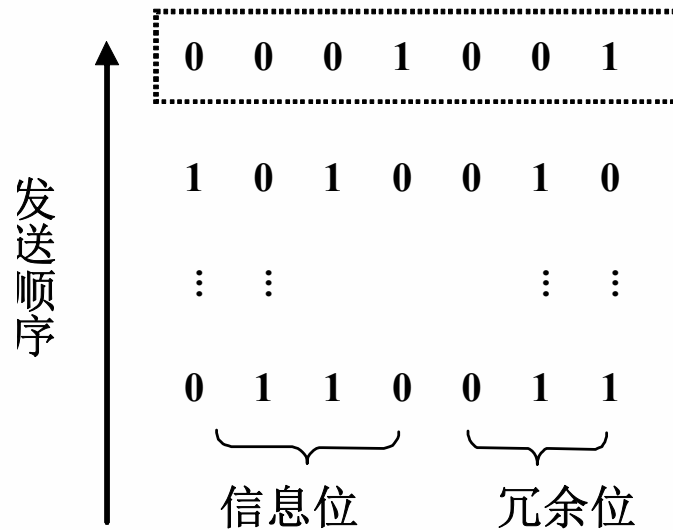
若第一位出错，则

$$\begin{cases} s1=1 \\ s2=0 \\ s3=1 \end{cases}$$

以此类推。得到下表



- 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 $\leq k$ ，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。



4.2.4 循环冗余码(检错码)

- 循环冗余码简称为**CRC码(Cyclic Redundancy 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 $a_1a_2a_3\dots a_n$ ，则对应的多项式为：

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

如 **110001** $\rightarrow x^5+x^4+1$



编码原理（发送端）

- 设信息位串为 $a_1a_2a_3\dots a_m$ ，则信息编码多项式为 $M(x)=a_1x^{m-1}+a_2x^{m-2}+a_3x^{m-3}+\dots+a_m$
- 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 $R(x)$
 $R(x)=\text{MOD}(x^rM(x)/G(x))$
 - 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 $R(x)=\text{MOD}(x^rM(x)/G(x))$



模2运算的原理-加法

- 模2加法运算定义为:

$$0+0=0$$

$$0+1=1$$

$$1+0=1$$

$$1+1=0$$

例如 $0101+0011=0110$ ，列竖式计算：

$$\begin{array}{r} 0101 \\ + 0011 \\ \hline 0110 \end{array}$$



模2运算的原理-减法

- 模2减法运算定义为:

$$0-0=0$$

$$0-1=1$$

$$1-0=1$$

$$1-1=0$$

例如 $0110-0011=0101$ ，列竖式计算：

$$\begin{array}{r} 0110 \\ - 0011 \\ \hline 0101 \end{array}$$



模2运算的原理-乘法

● 模2乘法运算定义为:

$$0 \times 0 = 0, 0 \times 1 = 0, 1 \times 0 = 0, 1 \times 1 = 1$$

多位二进制模2乘法类似于普通意义上的多位二进制乘法，模2乘法对中间结果的处理方式采用的是模2加法。

例如 $1011 \times 101 = 100111$ ，列竖式计算：

$$\begin{array}{r} 1011 \\ \times 101 \\ \hline 1011 \\ 0000 \\ + 1011 \\ \hline 100111 \end{array}$$



模2运算的原理-除法

●模2除法运算定义为：

$$0 \div 1 = 0, 1 \div 1 = 1$$

模2除做法与算术除法类似，但每一位除（减）的结果不影响其它位，即**不向上一位借位**。然后再移位做下一位的模2减。步骤如下：

- （1）用除数对被除数最高几位做模2减，没有借位；
- （2）除数右移一位，若余数最高位为1，商为1，并对余数做模2减。若余数最高位为0，商为0，除数继续右移一位；
- （3）一直做到余数的位数小于除数时，该余数就是最终余数。



1001111

商

除数G(X)

10111

10100010000

被除数

10111

0011010

10111

11010

10111

11010

10111

11010

10111

1101

余数

$T(x) = x^r M(x) + R(x)$

$= x^{10} + x^8 + x^4 + x^3 + x^2 + 1$

\rightarrow

1010001

1101

信息位串

校验位串

$R(x) = x^3 + x^2 + 1$



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 $\text{MOD}(T'(x))/G(x)=0$ 则正确
 - 若 $\text{MOD}(T'(x))/G(x) \neq 0$ 则出错，要求重发



CRC码检错能力

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 $G(x)$ 。



●三个标准**CRC**生成多项式:

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

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

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



4.2.5 其它差错控制编码

● 奇偶校验码

- 通过增加冗余位来使得码字中1的个数（即码重）保持奇数或偶数的编码方法。

● 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

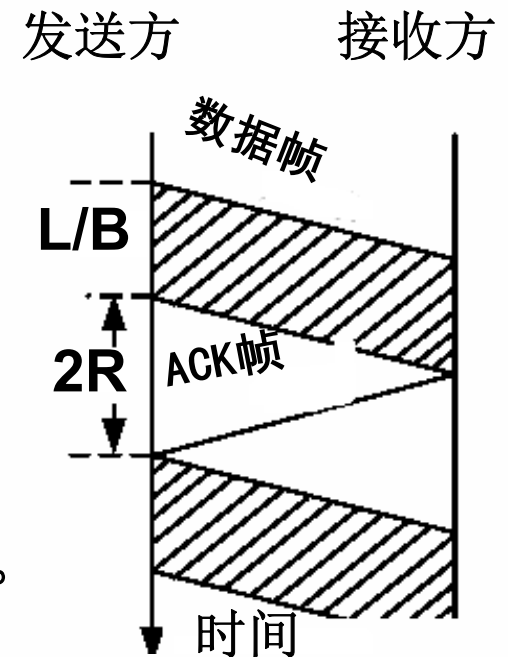
- 冗余位的个数与信息位个数相同。冗余位与信息位完全相同或者完全相反。



4.3 流量控制

4.3.1 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

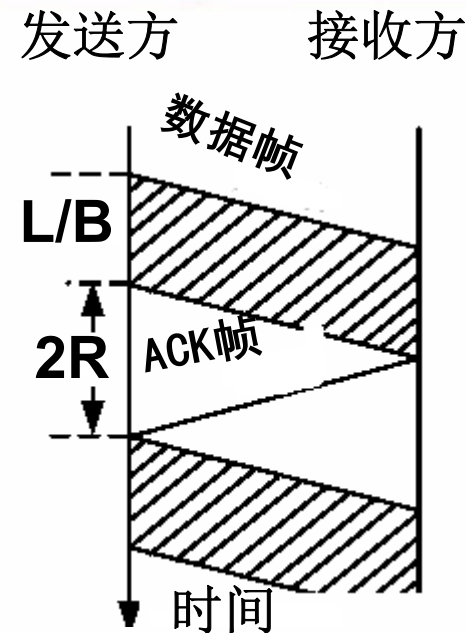
- 缺点

- 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。



4.3.2 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。



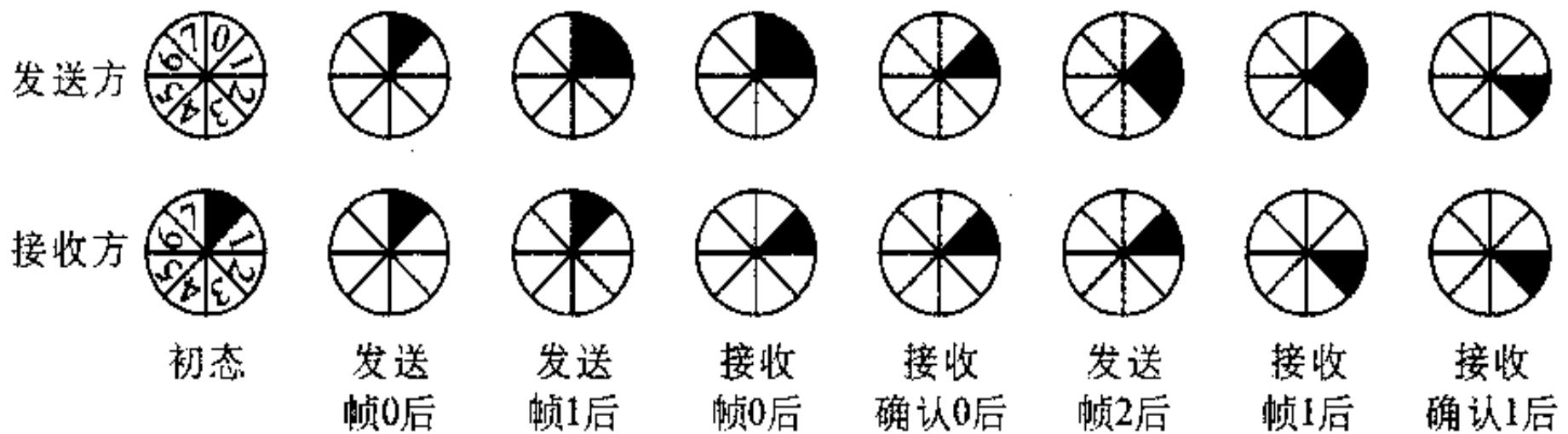
● 帧序号

- 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

- 只有帧序号落入当前窗口的帧才有资格发送，发送方收到对方确认信息后，将发送窗口向前滑动（顺序改变当前窗号）。
- 只有帧序号落入当前窗口的帧才接收，否则丢弃，接收方接收窗口中的帧递交给主机后，接收窗口向前滑动（顺序改变当前窗号）。



$W_T=2$, 接收窗口尺寸 $W_R=1$.



顺序接收管道协议（回退n）

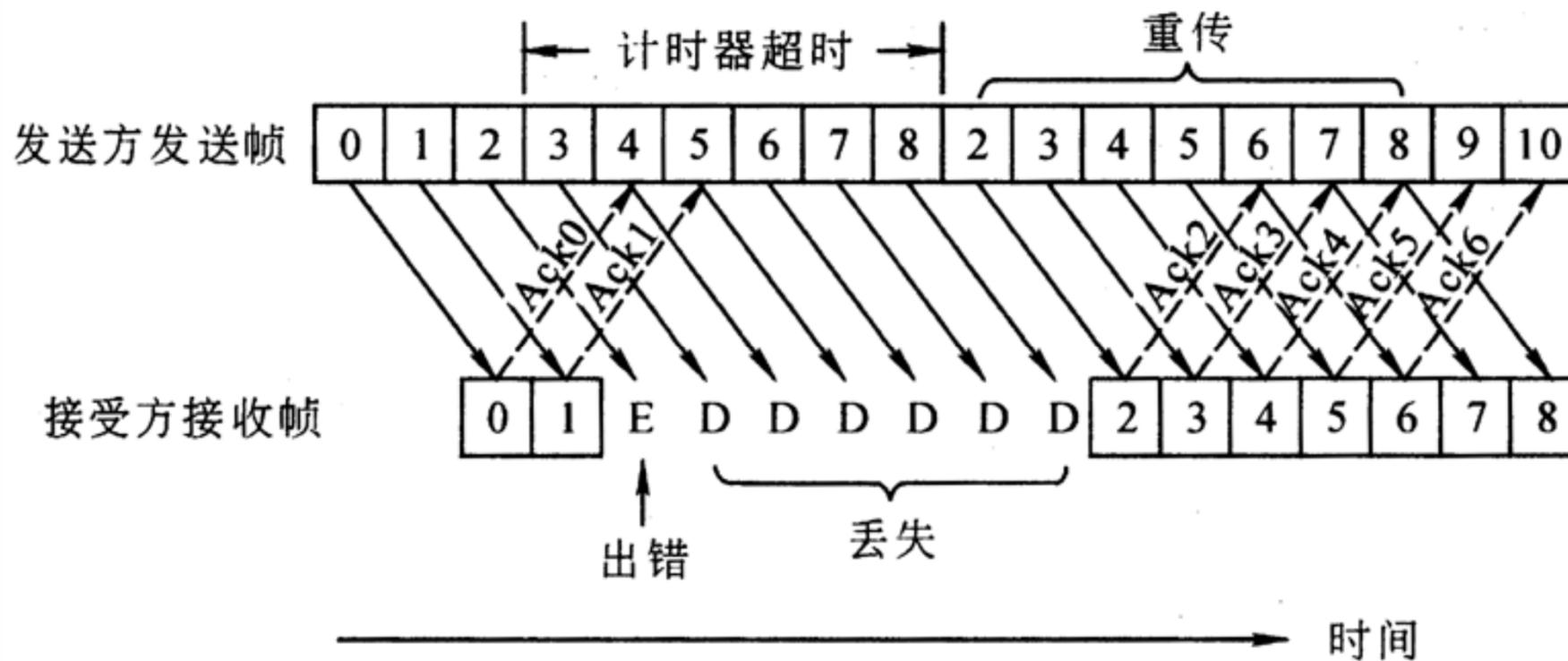
接收窗口尺寸为1的滑动窗口协议，也称回退n协议。

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





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

- 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。



滑动窗口协议小结

- 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。



4.4 数据链路层协议举例

4.4.1 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。



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

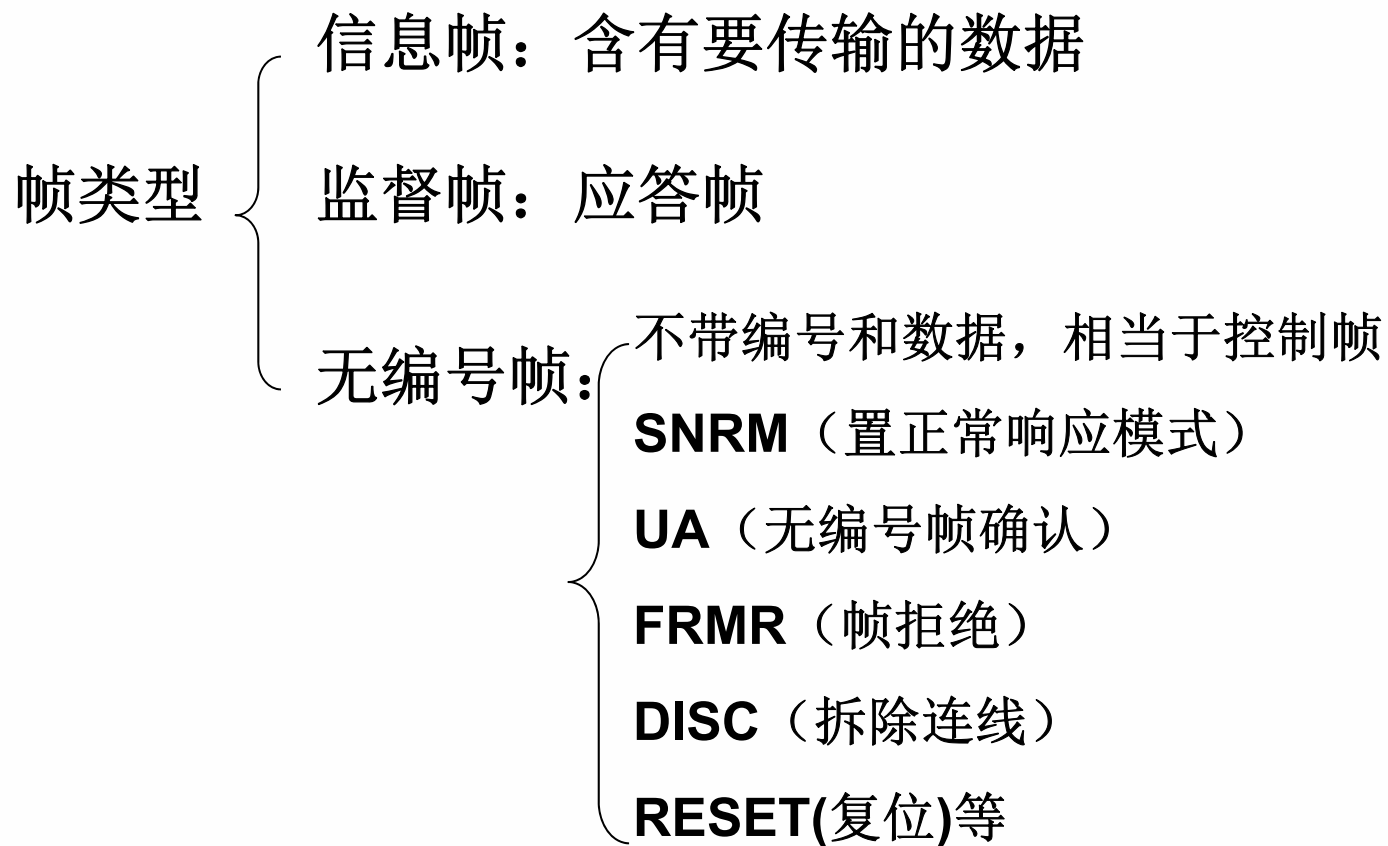
在接收端将 5 个连 1 之后的 0 比特删除，恢复原样

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。





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。当一帧由发出命令的主站发出时，该位起探寻的作用。该位为**1**时，即要求被选择的从站给出响应。从站的响应可以是多个帧，最后一个帧中该位应为**1**，表示“结束”。



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

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, 如要求各种计数器复位。



	标志	地址	控制	数据	帧检 验	标志
字节数	1	1	1	任意	2	1

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， $G(X)=CRC-CCITT = x^{16}+x^{12}+x^5+1$



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。



4.4.2 PPP协议

PPP: Point to Point Protocol

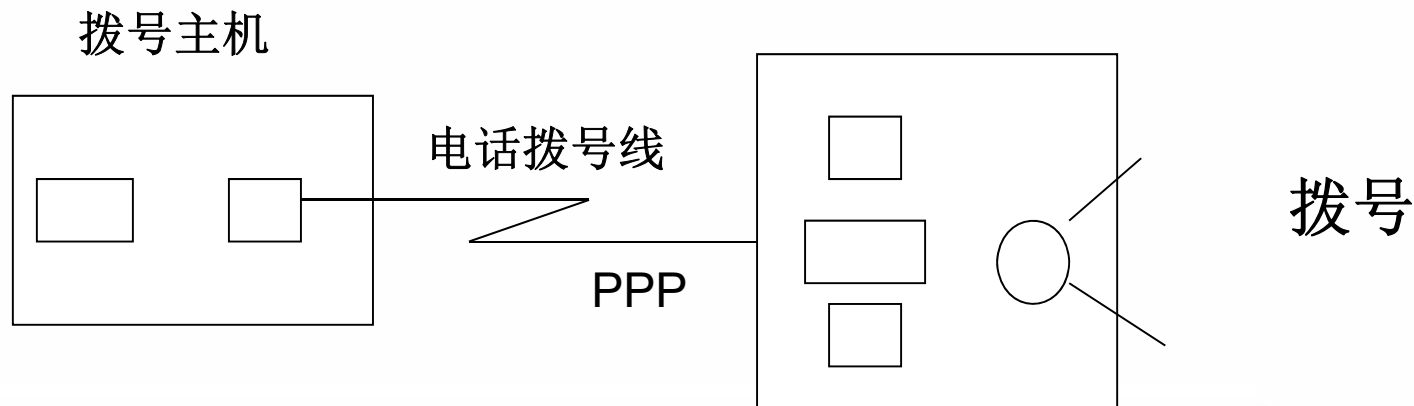
PPP协议是Internet上主流的链路层协议

PPP应用场合

主要用于拨号接入Internet的场合。

PPP协议的对等段是**客户和ISP**，它们通过Modem和电话网络连接。

- 能够传输数据帧：封装帧、建立数据链路
- 能够协商使用的网络层协议





PPP

- PPP是使用串行线路通信的面向字节协议
- PPP没有纠错功能，不进行流量控制，不需要帧序号，不支持多点链路，使用全双工方式传输数据。



PPP协议组成

PPP协议规定了以下3部分内容，即：

- (1) 帧格式及成帧方法 (**HDLC封装**)
- (2) 用于建立、配置和测试PPP链路的LCP (Link Control Protocol, **链路控制协议**)
- (3) 用于建立和配置网络层协议的NCP (Network Control Protocol, **网络控制协议**)。对于IP网络而言，使用IPCP (IP Control Protocol, IP控制协议)
- 此外，当客户拨入ISP时，ISP需要**验证客户的身份**，此时可使用2个认证协议，即**PAP** (Password Authentication Protocol, 口令认证协议) 和**CHAP** (Challenge Handshake Authentication Protocol)。



PPP协议组成

(1) 一种成帧方法(HDLC)

- 定义了将IP数据报封装到串行链路的方法，明确地定界一个帧的**结束**和下一个帧的**开始**，其帧格式允许进行错误检测。
- PPP既支持异步链路（无奇偶检验的8位数据），也支持面向位串的同步链路。
- IP数据报是PPP中信息部分，其长度受最大传送单元MTU的限制。MTU的默认值是1500字节。



PPP协议组成

(2) 一个链路控制协议 (LCP)

- 链路控制协议 (Link Control Protocol, LCP) 负责线路建立、配置、测试和选项协商，并在它们不再被需要时，稳妥地把它们释放。

(3) 一套网络控制协议 (NCP)

- 网络控制协议 (Network Control Protocol, NCP) 可支持不同的网络层协议，如IP、Appletalk等，对于所支持的每一个网络层协议都有一个不同的网络控制协议，用来建立和配置不同的网络层协议。提供了协商网络层选项的方式。



PPP协议流程

- 在建立PPP链路前，发起方必须通过电话网络呼叫回应方；呼叫成功后双方建立了一条物理连接
- 利用LCP创建PPP链路
- 用PAP或CHAP验证客户身份
- 用IPCP配置IP层参数（主要是配置IP地址）
- 通信完成后，双方利用LCP断开PPP链路
- 之后，断开物理连接。



PPP协议流程

上述流程中各个报文的含义如下：

- (1) 发起方发送LCP配置请求报文，其中包括各项配置参数，比如使用的认证协议、最大接收单元和压缩协议等
- (2) 回应方若同意各项配置参数，则返回确认报文
- (3) 发起方提供账号和口令，以便验证自己的身份
- (4) 回应方验证发起方的身份成功后，向其返回确认报文
- (5) 发起方发出IPCP配置请求
- (6) 回应方返回确认，其中包含了分配给发起方的IP地址
- (7) 发起方发送LCP终止链路请求
- (8) 回应方返回确认，链路终止



PPP的帧格式

字节数: 1 1 1 2 0-1500 2—4 1

标志 01111110	地址 11111111	控制 00000011	协议	数据	CRC	标志 01111110
----------------	----------------	----------------	----	----	-----	----------------

- 标志: **01111110**, 标志一帧的开始和结束。
- 地址: 总是固定为**11111111**.
- 控制: 缺省**00000011**.
- 协议: **PPP**的高层协议, 如 **IP**等。
- 数据: 静载荷数据, 如 **IP**分组
- **CRC**: **CRC**校验序码, 同**HDLC**.

在缺省的情况下, **PPP**没有采用序号和确认来进行可靠的传输。**主要用于信道质量较好的场合**, 在信道质量较差的情况下, 也可使用序号和确认机制进行差错控制。



4.5 数据链路层设备

- 网桥（**bridge**）是一种数据链路层互联的设备。
- 网桥是一种对帧进行转发的技术，根据**MAC**分区块，可隔离碰撞。网桥将网络的多个网段在数据链路层连接起来。
- 网桥也叫桥接器，是连接两个局域网的一种存储/转发设备，它能将一个大的**LAN**分割为多个网段，或将两个以上的**LAN**互联为一个逻辑**LAN**，使**LAN**上的所有用户都可访问服务器。



网桥

- 在网络互联中它起到数据接收、地址过滤与数据转发的作用，用来实现多个网络系统之间的数据交换。
- 网桥可根据帧的终点地址处于哪一网段来进行转发和滤除。
 - 使用网桥连接两段LAN时，网桥对来自网段1的MAC帧首先要检查其终点地址。如果该帧是发往网段1上某一站的，网桥则不将帧转发到网段2，而将其滤除；如果该帧是发往网段2上某一站的，网桥则将它转发到网段2。



网桥的基本特征

网桥具有以下基本特征：

- (1) 网桥在数据链路层上实现局域网互联。
- (2) 网桥以接收、存储、地址过滤与转发的方式实现互联网络之间的通信。
- (3) 网桥可以分隔两个网络之间的广播通信量，有利于改善互连网络的性能与安全性。



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