

CHAPTER 3：数据链路层

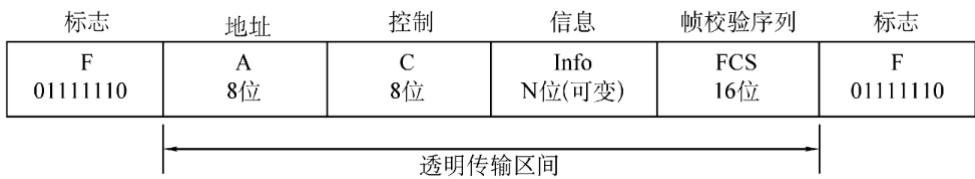
逻辑链路控制子层

笔记源文件：[Markdown](#)，[长图](#)，[PDF](#)，[HTML](#)

1. 数据链路概述

1.1. 基本概念

- 1 结点：链路层中任一设备(主机/路由器/交换机/Wifi接入点)
- 2 链路：相邻结点间的通信信道，无线连接也是链路，不同类型链路运行不同链路协议
- 3 帧：第二层的数据包，由数据报封装(加上首部/尾部控制信息)而来，例如以下



1.2. 链路层的目的/服务/功能

- 1 目的：有差错的物理线路→无差错的逻辑链路
- 2 服务：

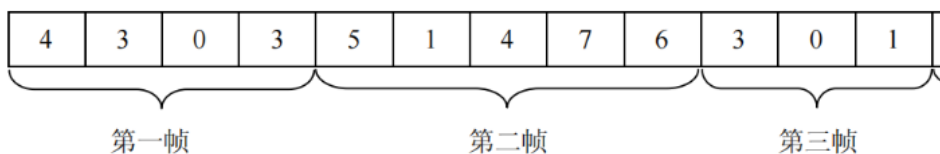
	无确认服务	有确认服务
无连接服务	√	√
有连接服务	×	√

- 3 功能
 - 1. 数据成帧：数据字段+若干首部
 - 2. 链路接入：通过MAC协议实现
 - 3. 可靠交付：通过确认重传实现
 - 4. 差错控制：由硬件实现，包括检错和纠错
 - 5. 半双工/全双工通信
 - 6. 流量控制

2. 组帧/成帧：给数据加上首尾

2.1. 字符计数法

- 1 含义：头部包含一个字段指示帧中字符的数量，尾部不动



❷ 缺点：计数字段在传输中出现差错是很难办

2.2. 首尾界符法

❶ ASCII中有33个不可打印的控制字符

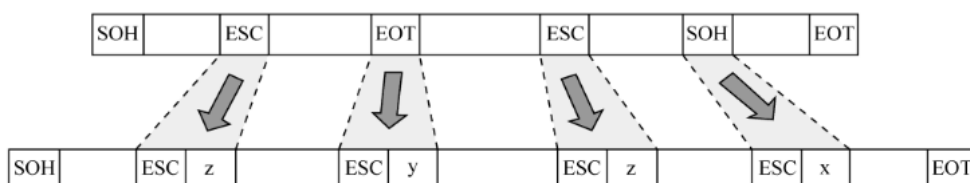
❷ 字符填充：传送帧为文本是，在一帧的首尾加上控制字符(SOH/EOH)

❸ 缺陷：对于非文本帧，中间可能出现控制字符，全乱套了

2.3. 字节填充的首尾界符法

❶ 基于首尾界符，数据中若出现控制字符，就将其转化为另一个非控制字符

❷ 转化方式不唯一，不论哪种方式都在接收端转化回来



2.4. 比特填充的首尾标志法

❶ 方式：用01111110(6个1)作为帧起止标志

❷ 透明传输的解决

1. 发送时：已经发出5个1了，马上接下来强行插入一个0然后发出
2. 接收时：已经收到5个1了，强制删除后面紧跟的0

2.5. 物理编码违例法

❶ 方式：用物理介质上编码的违法标志，来区分帧的起止

❷ 例如：Manchester编码中

1. 一帧会中期跳变，所以只可能是0-1或1-0
2. 可以拿不存在的1-1和0-0来作为起止标示

3. 差错控制

3.1. 水平奇偶校验(检错不纠错)

❶ 奇/偶校验：

1. 发送端：信息后加1bit的校验，使得1的个数是奇数/偶数
2. 接收端：检测到1有奇数/偶数个，则认为没出错(否则重传)

❷ 特点：二比特同时出错，检测不出

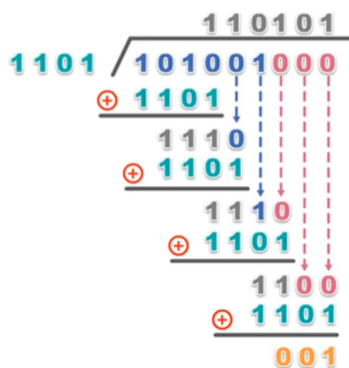
3.2. 循环冗余码CRC(检错不纠错)

3.2.1. 示例

1 前提条件:

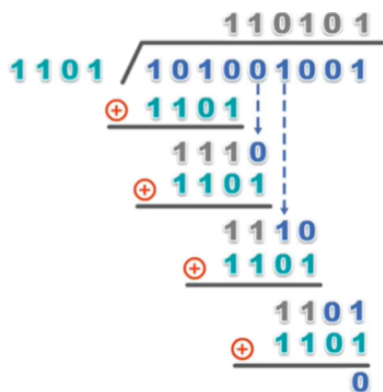
1. 数据单元为101001, 对应 $G(x) = x^5 + x^3 + 1$
2. 强制规定生成码为1101, 规定生成多项式为 $G(x) = x^3 + x^2 + 1$
3. 令 r = 生成码位数-1, 为检测位位数

2 发送端行为



1. 构造被除数: 原始数据+0(个数等于生成码位数-1)→<101001><000(检测位)>
2. 除数: 就是生成码, 即1101
3. 执行除法: 注意图中的⊕表示异或, 余数就是校验码, 这里是001
4. 发出数据+校验码, 为<101001><001>

3 接收端行为:



1. 收到<101001><001>
2. 执行除法, 被除数是收到数据, 除数是生成码
3. 余数为0, 则没出错

3.2.2. 检错性能

1 有 r 检测位的多项式, 能检出所有小于等于 r 位的突发

2 长度大于 $r+1$ 的错误逃脱的概率是 $\frac{1}{2^r}$

3.3. 海明码(可纠错)

3.3.1. 求海明码：D=101101为例

1 确定校验码位数 r ：满足 $k \leq 2^r - r - 1$ ， k 为数据位数，此处 $k = 6$ 所以至少有 $r = 4$

2 确定校验码放哪：第 i 位校验码 P_i 放在总的 2^{i-1} 位，此处如下(数据按顺序填满剩下的位)

	M ₁	M ₂	M ₃	M ₄	M ₅	M ₆	M ₇	M ₈	M ₉	M ₁₀
甲	P ₁	P ₂	D ₁	P ₃	D ₂	D ₃	D ₄	P ₄	D ₅	D ₆
乙			1		0	1	1		0	1

3 求出校验值

- 假设出错位为 e_1, e_2, \dots, e_r ，此处是 e_1, e_2, e_3, e_4
- 用 r 位二进制数，给 M_1, M_2, \dots, M_{k+r} 标号，此处是 M_1, M_2, \dots, M_{10}
- 找出 M 与 r 的对应关系

	标号	e_4	e_3	e_2	e_1
M_1	0001	无关	无关	无关	有关
M_2	0010	无关	无关	有关	无关
M_3	0011	无关	无关	有关	有关
M_4	0100	无关	有关	无关	无关
M_5	0101	无关	有关	无关	有关
M_6	0110	无关	有关	有关	无关
M_7	0111	无关	有关	有关	有关
M_8	1000	有关	无关	无关	无关
M_9	1001	有关	无关	无关	有关
M_{10}	1010	有关	无关	有关	无关

4. 整理有关项，然后异或

出错位	有关项	异或操作
e_1	$M_{1,3,5,7,9}$	$e_1 = M_1 \oplus M_3 \oplus M_5 \oplus M_7 \oplus M_9$
e_2	$M_{2,3,6,7,10}$	$e_2 = M_2 \oplus M_3 \oplus M_6 \oplus M_7 \oplus M_{10}$
e_3	$M_{4,5,6,7}$	$e_3 = M_4 \oplus M_5 \oplus M_6 \oplus M_7$
e_4	$M_{8,9,10}$	$e_4 = M_8 \oplus M_9 \oplus M_{10}$

5. 代入 2 中数据，求出 e 表达式

出错位	表达式
e_1	$P_1 \oplus D_1 \oplus D_2 \oplus D_4 \oplus D_5$

出错位	表达式
e_2	$P_2 \oplus D_1 \oplus D_3 \oplus D_4 \oplus D_6$
e_3	$P_3 \oplus D_2 \oplus D_3 \oplus D_6$
e_4	$P_4 \oplus D_5 \oplus D_6$

6. 如果没有错误信息，则必须由 $e_1, e_2, e_3, e_4 = 0$ ，注意 $A \oplus B = 0 \iff A = B$ ，所以有

校验码	表达式	取值
P_1	$D_1 \oplus D_2 \oplus D_4 \oplus D_5$	0
P_2	$D_1 \oplus D_3 \oplus D_4 \oplus D_6$	0
P_3	$D_2 \oplus D_3 \oplus D_6$	0
P_4	$D_5 \oplus D_6$	1

7. 填充，最后得到海明码

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
甲	P_1	P_2	D_1	P_3	D_2	D_3	D_4	P_4	D_5	D_6
丙	0	0	1	0	0	1	1	1	0	1

3.3.2. 通过海明码校验

1 先通过这张表求出 e

出错位	有关项	异或操作
e_1	$M_{1,3,5,7,9}$	$e_1 = M_1 \oplus M_3 \oplus M_5 \oplus M_7 \oplus M_9$
e_2	$M_{2,3,6,7,10}$	$e_2 = M_2 \oplus M_3 \oplus M_6 \oplus M_7 \oplus M_{10}$
e_3	$M_{4,5,6,7}$	$e_3 = M_4 \oplus M_5 \oplus M_6 \oplus M_7$
e_4	$M_{8,9,10}$	$e_4 = M_8 \oplus M_9 \oplus M_{10}$

2 结果按照 e_4, e_3, e_2, e_1 排列，排列结果化为十进制 N ，那么出错的就是第 N 位

3.3.3. 海明距离

1 含义：把两个码字对齐后，不相同的位的个数

2 海明距离和编码方案

目的	编码方案的海明距离
检测出 d 位错误	$d + 1$
纠正 d 位错误	$2d + 1$

3 $L - 1 \geq D + C$ ： LDC 分别为编码长度/检错位数/纠错位数

4. 基本数据链路协议

4.1. 乌托邦协议

1 单工传输

2 假设是一个理想环境/没有错误/不丢包等，所以也不会有差错控制/流量控制

4.2. 停-等协议

1 含义：发完一帧后，必须等对方回送ACK(所以是半双工)，才能发送下一帧

2 问题与Solution

1. 一直收不到ACK：发送端超时重传，**停止-等待+重传机制=PAR协议**
2. 接收端无法确定收到数据是否重复：给每帧加上序列号，序列号有限所以可以是循环的

2 协议中的差错

1. 数据帧传丢了
2. 数据帧损毁：被接收方直接扔掉，也不发ACK
3. 确认帧传丢了/损坏：发送方确认不了，总是超时重传

4.3. 滑动窗口协议

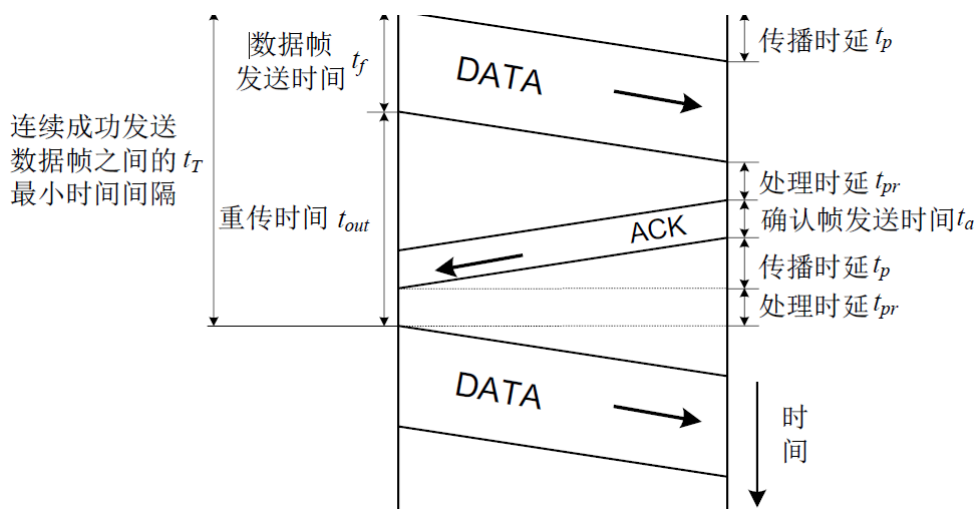
4.3.1. 单比特滑动窗口

1 一对一：发送窗口大小=1，接收窗口大小=1

2 含义：实质上是有错信道下的双工停-等协议

3 特点：发送端最多缓存一个未确认的帧，接收端收到一个帧后就不再接收帧了

4 发送效率： t_f/t_T



4.3.2. 后退N帧(GBN)协议

1 多对一： n 比特给窗口编号，发送窗口尺寸 $1 < W_T \leq 2^n - 1$ ，接收窗口还是1

2 原理

1. 发送端一股脑发很多帧，若收到ACK，则继续一股脑又发很多帧

2. 如果某一帧出错了，则接收端丢掉该帧及以后所有帧，发送端重发被丢掉的帧

4.3.3. 选择重传

1 多对多，二者窗口都可以是 $W_T \leq 2^{n-1}$ ，但不一定相等

2 原理

1. 若一帧出错，其后续帧先缓存在接收窗口，同时要求发送方重传出错帧
2. 收到重传帧后，和缓存一起按顺序排好