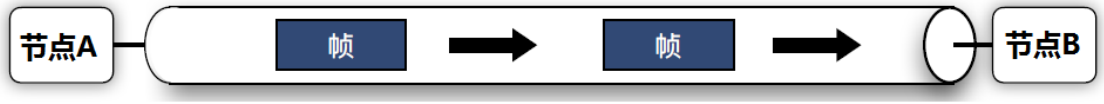
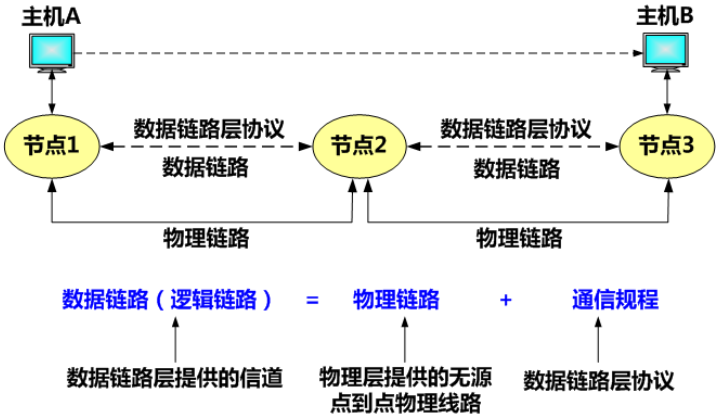
课程回顾

- **数据链路层**是实现设备之间通信的非常重要的一层
  - 广域网中的**主机**、**路由器**等都必须实现数据链路层。
  - 局域网中的**主机**、**交换机**等都必须实现数据链路层。
  - 实现**相邻节点**之间的通信，为网络层实现的主机之间的通信提供支持。



- **物理链路**和**数据链路**



- 数据链路层需要解决三个问题
  - **封装成帧**（在一段数据的前后分别添加首部和尾部）
  - **透明传输**（如**字节填充**）
  - **差错检测**（如**循环冗余检验CRC**）问题。

差错控制编码是差错控制的核心，其基本思想是在待传送的信息码元序列中按照一定的规则插入一定数量的新码元（即监督码元），使原来彼此独立、不相关的信息码元序列经过变换后产生某种规律性或相关性，在接收端可以根据这种规律性来检查传输序列中的差错，乃至纠正差错。

码的检错和纠错能力是用信息量的冗余度来换取的，一般信源发出的任何消息都可以用二进制信号“0”和“1”来表示。

- 1位二进制编码可以表示2种不同的状态
- 2位二进制编码可以表示4种不同的状态
- 3位二进制编码可以表示8种不同的状态
- n位二进制编码可以表示2^n种不同的状态

几个常用的差错控制编码

- 奇偶校验码(奇偶监督码)
- 矩阵校验码（行列监督码、二维奇偶校验码）
- 循环冗余校验码

- 奇偶校验码(奇偶监督码)——最简单的检错码

  编码规则：

  待传送的信息为$n-1$个码元$C_{n-1}C_{n-2}\cdots C_2C_1$

  在分组的数据后面附加1位校验位C_0

  偶校验：$C_{n-1}\oplus C_{n-2}\oplus\cdots\oplus C_2\oplus C_1\oplus C_0=0$

  奇校验：$C_{n-1}\oplus C_{n-2}\oplus\cdots\oplus C_2\oplus C_1\oplus C_0=1$

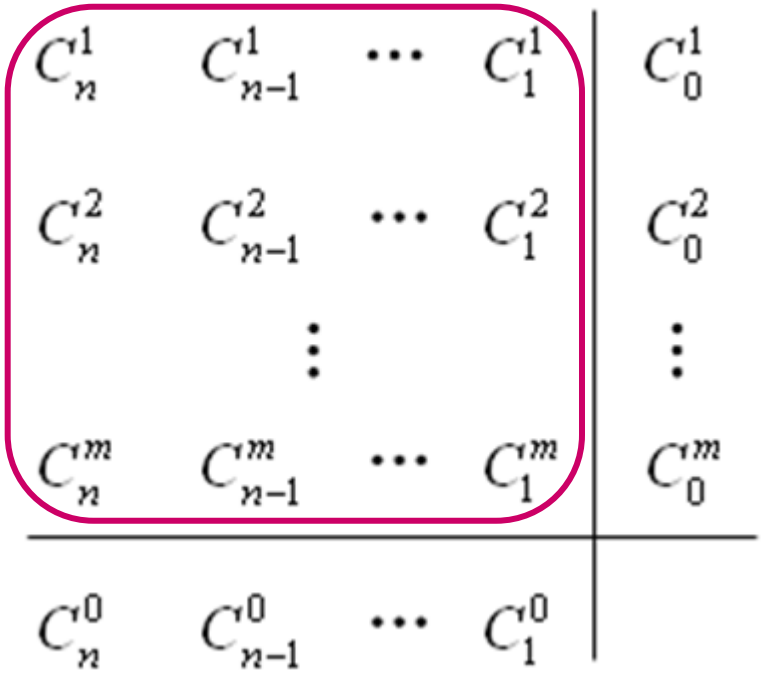
优点：冗余度低，编码效率高。
缺点：只能发现单个或奇数个错误，检测能力不高。

- 矩阵校验码（行列监督码、二维奇偶校验码）

信息位

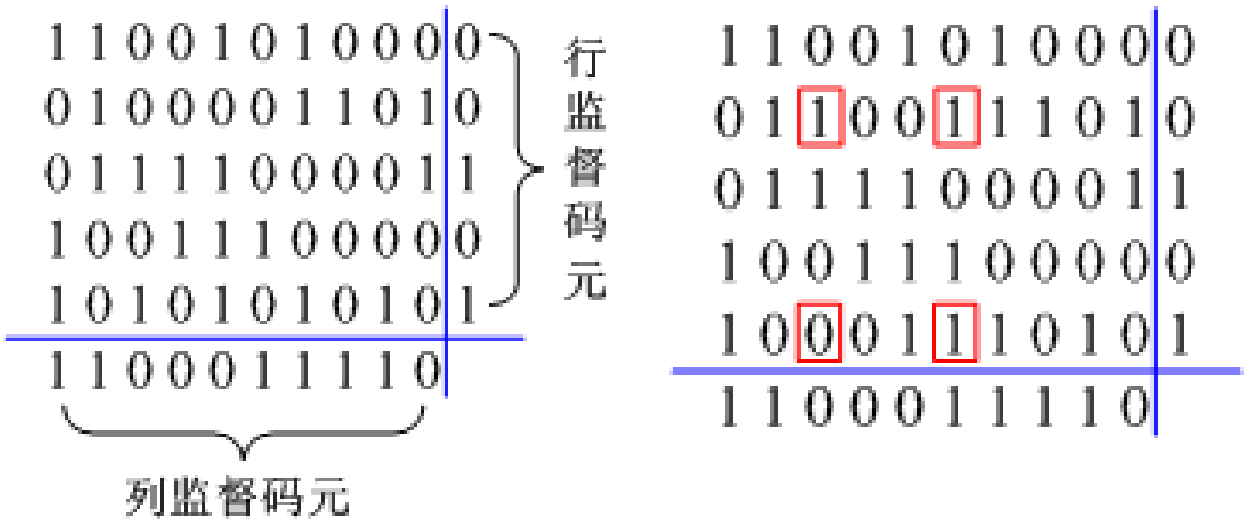
*m*个码字

每个码字*n*位



例 用户要发送的信息序列为(1100101000010001101011110000110011100001010101010)，现将每10个信息码元分为一组，然后编成方阵，按行和列分别加上监督码元，则在发送端构成的方阵为

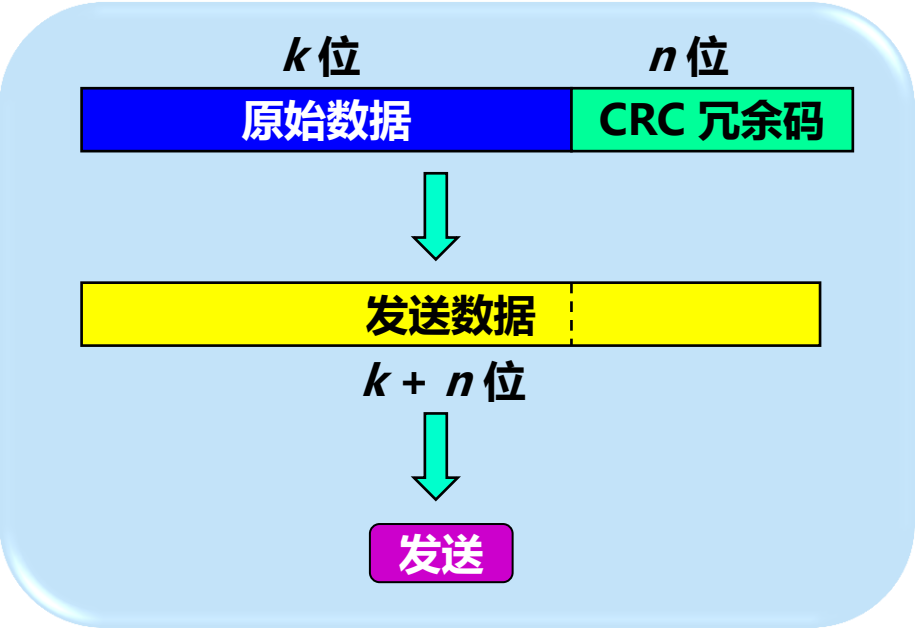
例 用户要发送的信息序列为(1100101000010001101011110000110011100001010101010)，现将每10个信息码元分为一组，然后编成方阵，按行和列分别加上监督码元，则在发送端构成的方阵为



差错检测

循环冗余校验（CRC）原理

- 在发送端，先把数据划分为组。假定每组 *k* 个比特（即数据 *M*）。
- 在每组后面再添加供差错检测用的 *n* 位冗余码，然后一起发送出去。

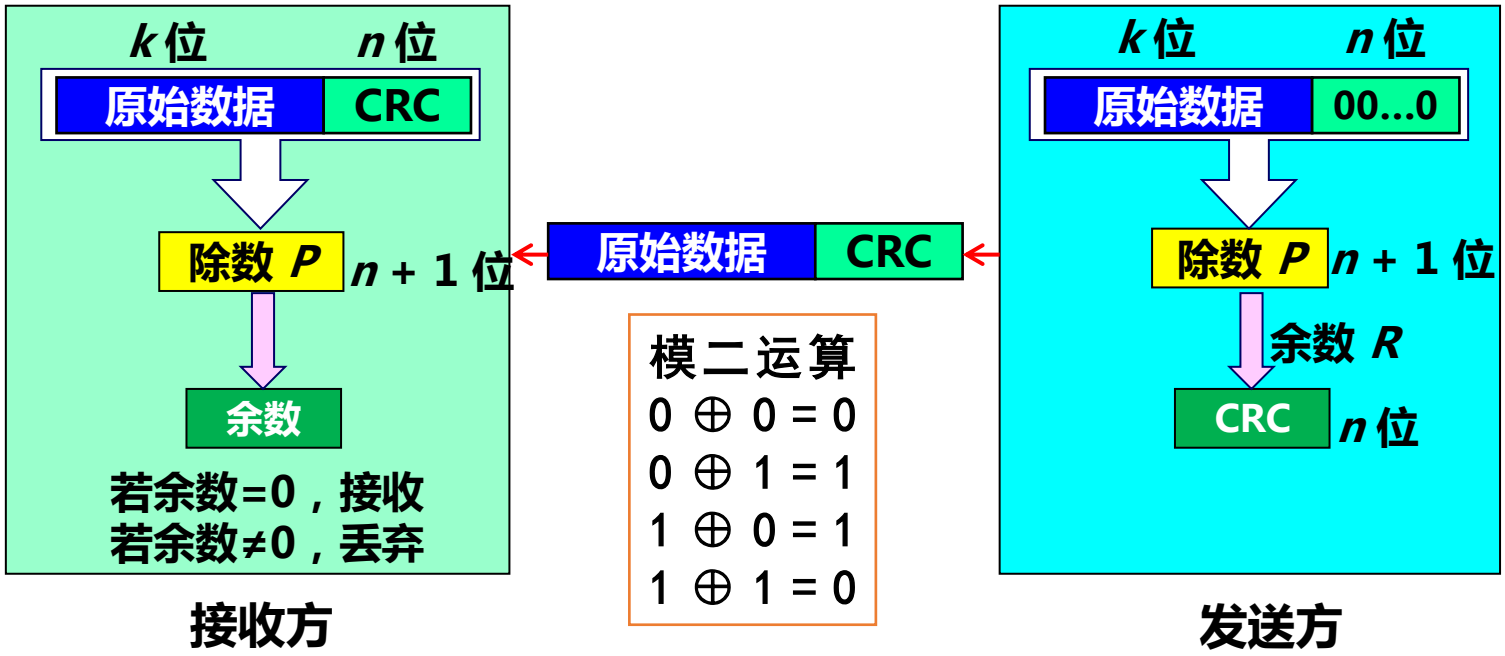


（7，3）码	
信息组	码　字
0 0 0	0 0 0 0 0 0 0
0 0 1	0 0 1 1 1 0 1
0 1 0	0 1 0 0 1 1 1
0 1 1	0 1 1 1 0 1 0
1 0 0	1 0 0 1 1 1 0
1 0 1	1 0 1 0 0 1 1
1 1 0	1 1 0 1 0 0 1
1 1 1	1 1 1 0 1 0 0

差错检测

冗余码的计算

- 在 *k* 比特数据 *M* 的末尾加上 *n* 个0，*n* 是一个比生成多项式 *P* 的位数少1的二进制数；
- (*k* + *n*) 位的数除以(*n* + 1)位的生成多项式 *P*，得出商是 *Q* 而余数是 *R*，余数 *R* 比除数 *P* 少1位，即 *R* 是 *n* 位，*R* 就是得到的冗余码。
- 将余数 *R* 作为冗余码拼接在数据 *M* 后面，一起发送出去。



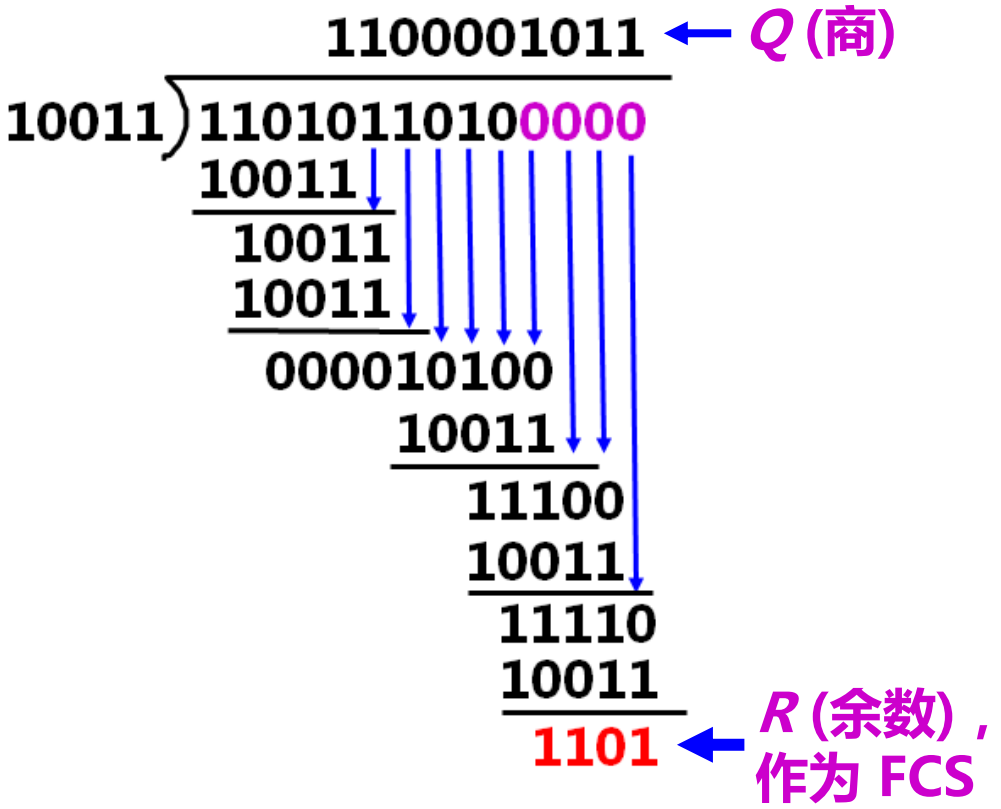
差错检测

冗余码的计算举例

假设要发送的数据为 1101011010，已知 CRC 的生成多项式 $G(X)=X^4+X+1$，试求添加在数据后面的余数。

用二进制数表示为 G = 10011，这就是模2运算中的除数

在发送的数据后面添加4个0，得到被除数11010110100000。



差错检测

注意：

- 在数据后面添加上的冗余码称为**帧检验序列** FCS (Frame Check Sequence)。
- 循环冗余检验 CRC 和帧检验序列 FCS **并不等同**。
  - CRC 是一种常用的**检错方法**，而 FCS 是添加在数据后面的冗余码。
  - FCS 可以用 CRC 这种方法得出，但 CRC 并非用来获得 FCS 的唯一方法。
- 仅用循环冗余检验 CRC 差错检测技术只能做到**无差错接收**。
- **“无差错接收”是指**：“凡是接收的帧（即不包括丢弃的帧），我们都能以非常接近于 1 的概率认为这些帧在传输过程中没有产生差错”。也就是说：“凡是接收端数据链路层接收的帧都没有传输差错”（有差错的帧就丢弃而不接收）。

差错检测

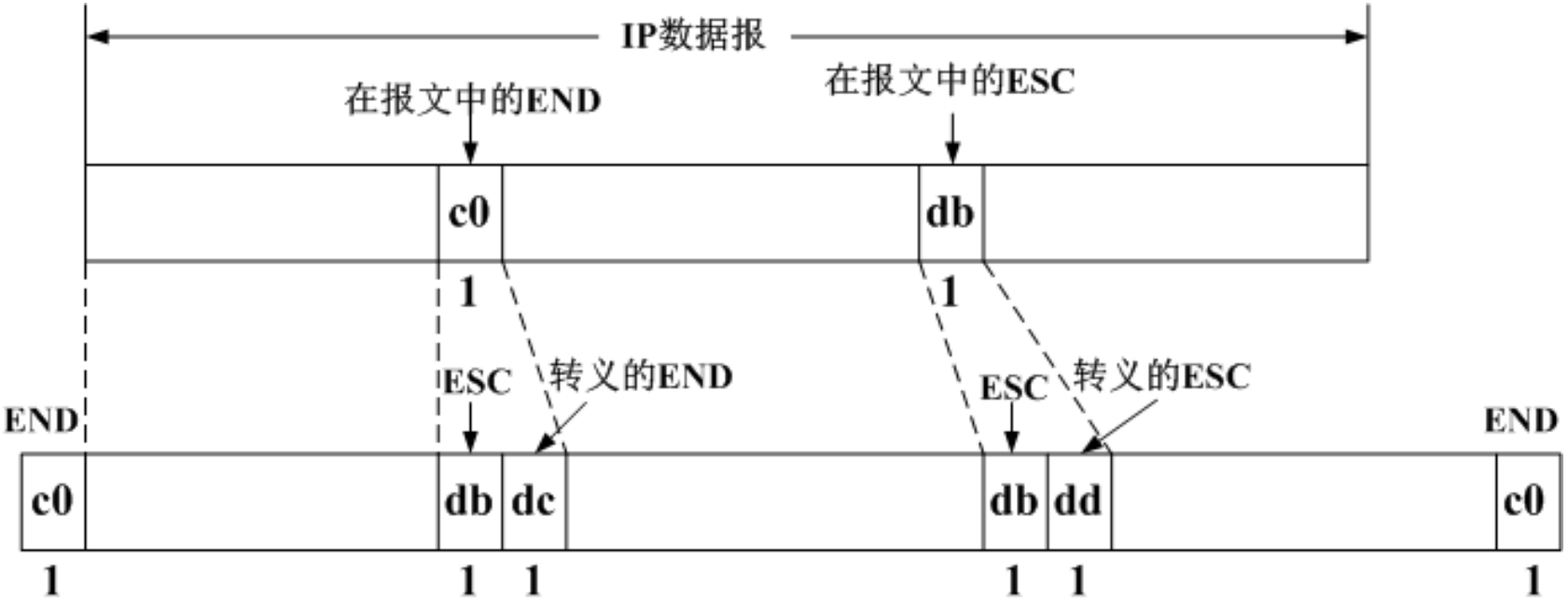
注意：

- 单纯使用 CRC 差错检测技术不能实现“无差错传输”或“可靠传输”。
- 应当明确，“无比特差错”与“无传输差错”是不同的概念。
- 在数据链路层使用 CRC 检验，能够实现无比特差错的传输，但这还不是可靠传输。
- 要做到“无差错传输”（即发送什么就收到什么）就必须再加上确认和重传机制。

串行线路IP（SLIP）

- 一种在串行通信线路上支持 TCP/IP 协议的点对点的链路层通信协议，不但能发送和接收 IP 数据报，还提供了 TCP/IP 的各种网络应用服务，如Telnet、FTP等。
- 允许主机和路由器混合连接，即主机-主机、主机-路由器、路由器-路由器都是SLIP网络通用的配置。
- 线路速率非常低，一般介于1.2kbit/s~19.2kbit/s之间，允许用户传送低速的交互业务。
- 为个人用户上网提供拨号IP模式，为行业用户通过串行媒介传输业务数据提供了专线IP模式。

串行线路IP（SLIP）



SLIP存在的问题：

- **没有寻址功能；**
- **数据帧中没有类型标记；**
- **没有差错检测和校正功能。**

点对点协议（PPP）

- 对于点对点的链路，目前使用最广泛的数据链路层协议是**点对点协议** PPP (Point-to-Point Protocol)。
- PPP 协议在 1994 年成为互联网的正式标准。
- 能够在多种链路上运行
  - 串行、并行
  - 同步、异步
  - 低速、高速
  - 交换（动态）、非交换（静态）
  - 电、光

PPP 协议——应满足的需求

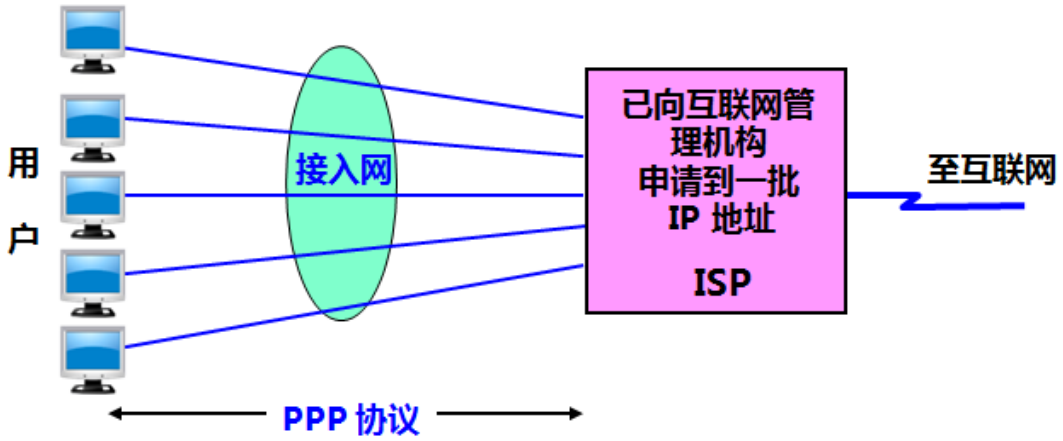
- **简单**——**这是首要的要求**。
- **封装成帧**——必须规定特殊的字符作为帧定界符。
- **透明性**——必须保证数据传输的透明性。
- **多种网络层协议**——能够在同一条物理链路上同时支持多种网络层协议。
- **多种类型链路**——能够在多种类型的链路上运行。
- **差错检测**——能够对接收端收到的帧进行检测，并立即丢弃有差错的帧。
- **检测连接状态**——能够及时自动检测出链路是否处于正常工作状态。
- **最大传送单元**——必须对每一种类型的点对点链路设置最大传送单元 MTU 的标准默认值，促进各种实现之间的互操作性。
- **网络层地址协商**——必须提供一种机制使通信的两个网络层实体能够通过协商知道或能够配置彼此的网络层地址。
- **数据压缩协商**——必须提供一种方法来协商使用数据压缩算法。

PPP 协议——不需要的功能

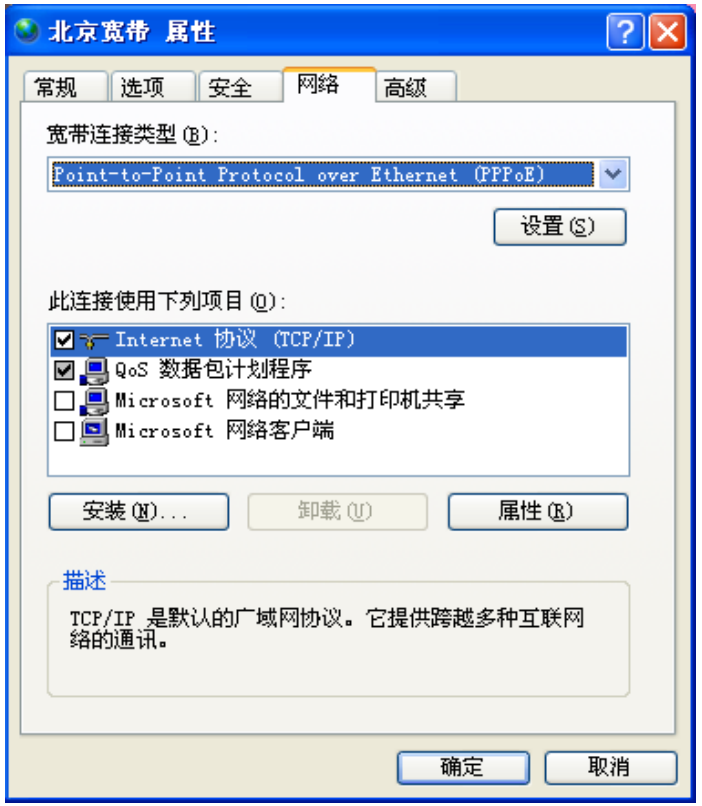
- 纠错
- 流量控制
- 设置序号
- 多点线路
- 半双工或单工链路

PPP非常简单：每收到一个帧，进行CRC检验，如果CRC检验正确，就接收这个帧；反之，就丢弃这个帧，其他什么也不做。

PPP 用在哪里？



用户到 ISP 的链路使用 PPP 协议



PPPoE实现了传统以太网没有身份验证、加密以及压缩等功能。

PPP 的组成

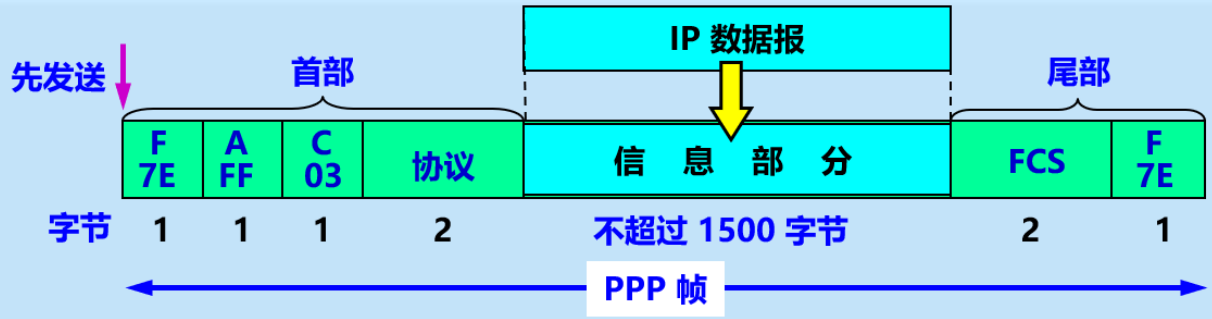
- PPP 提供了在点对点链路上传输多种不同类型协议数据的标准方法
- 由三个部分组成：
  - 一个将 IP 数据报封装到串行链路的方法。
  - 链路控制协议 LCP (Link Control Protocol)，用来建立、配置和测试链路，其最重要的功能之一是身份验证 (PAP，CHAP)。
    - 配置确认帧：所有选项都能接受；
    - 配置否认帧：所有选项都理解但不接受；
    - 配置拒绝帧：选项无法识别或不能接受内容，需要协商。
  - 网络控制协议 NCP (Network Control Protocol)，支持不同的网络层协议 (IP、OSI网络层协议、AppleTalk等)。

PPP 的组成

- PPP 提供了在点对点链路上传输多种不同类型协议数据的标准方法
- 由三个部分组成：
  - 一个将 IP 数据报封装到串行链路的方法。
  - 链路 立、配置
    和测 CHAP)。
    - i
    - i
    - i
  - 网络 协商。
    的网 支持不同

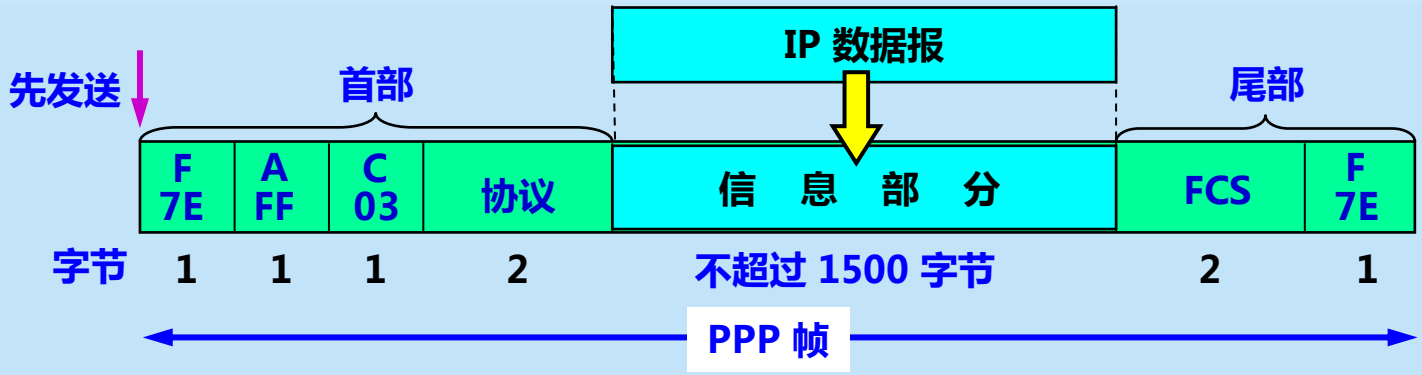
PPP 的帧格式

- PPP 帧的首部和尾部分别为 4 个字段和 2 个字段。
- 标志字段 F = 0x7E（符号“0x”表示后面的字符是用十六进制表示。十六进制的 7E 的二进制表示是 01111110）。
- 地址字段 A 只置为 0xFF(即11111111)，实际上并不起作用。
- 控制字段 C 通常置为 0x03(00000011)。
- PPP 是面向字节的，所有的 PPP 帧的长度都是整数字节。



PPP 有一个 2 个字节的协议字段。其值
- 若为 0x0021，则信息字段就是 IP 数据报。
- 若为 0x8021，则信息字段是网络控制数据。
- 若为 0xC021，则信息字段是 PPP 链路控制数据。
- 若为 0xC023，则信息字段是鉴别数据。

PPP 的帧格式



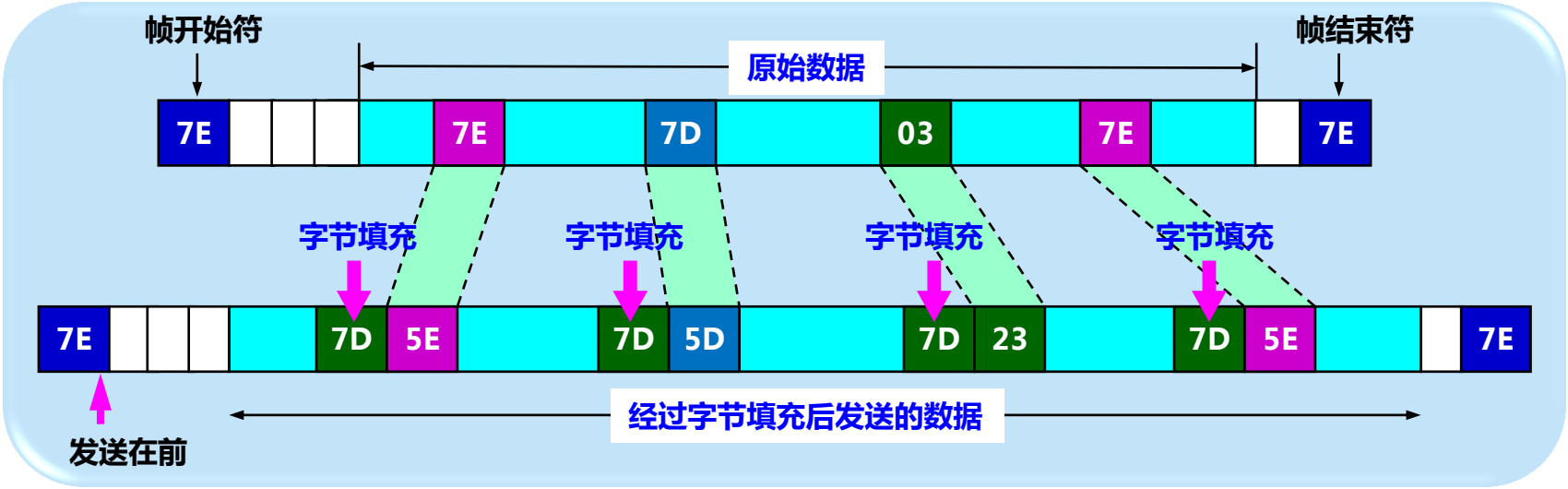
PPP 有一个 2 个字节的协议字段。其值

- 若为 0x0021，则信息字段就是 IP 数据报。
- 若为 0x8021，则信息字段是网络控制数据。
- 若为 0xC021，则信息字段是 PPP 链路控制数据。
- 若为 0xC023，则信息字段是鉴别数据。

透明传输问题

- 当 PPP 用在异步传输时，就使用一种特殊的字节填充法。
- 当 PPP 用在同步传输时，协议规定采用硬件来完成比特填充（和高级数据链路控制（HDLC）的做法一样）。

PPP 提供的透明传输方法——字节填充法



- 将信息字段中出现的 0x7E 字节均转变成为 2 字节序列 (0x7D, 0x5E)。
- 若信息字段中出现 0x7D 字节, 将其转变成为 2 字节序列 (0x7D, 0x5D)。
- 若信息字段中出现 ASCII 码的控制字符（即数值小于 0x20 的字符），则在该字符前面要加入一个 0x7D 字节，同时将该字符的编码加以改变。

(0x7D, 控制字符+0x20)

PPP 提供的透明传输方法——零比特填充

- PPP 用在 SONET/SDH 链路时，使用同步传输（一连串的比特连续传送），这时 PPP 采用零比特填充方法来实现透明传输。
- 在发送端，只要发现有 5 个连续 1，则立即填入一个 0。
- 接收端对帧中的比特流进行扫描，每当发现 5 个连续1时，就把这 5 个连续 1 后的一个 0 删除。

信息字段中出现了和标志字段 F 完全一样的 8 比特组合 0x7E	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 F
发送端在 5 个连 1 之后填入比特 0 再发送出去	0 1 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 发送端填入 0 比特
接收端把 5 个连 1 之后的比特 0 删除	0 1 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0 接收端删除填入的 0 比特

数据部分恰好出现与 0x7E 一样的二进制位串