

链路层三个基本问题，PPP

苏铅坤

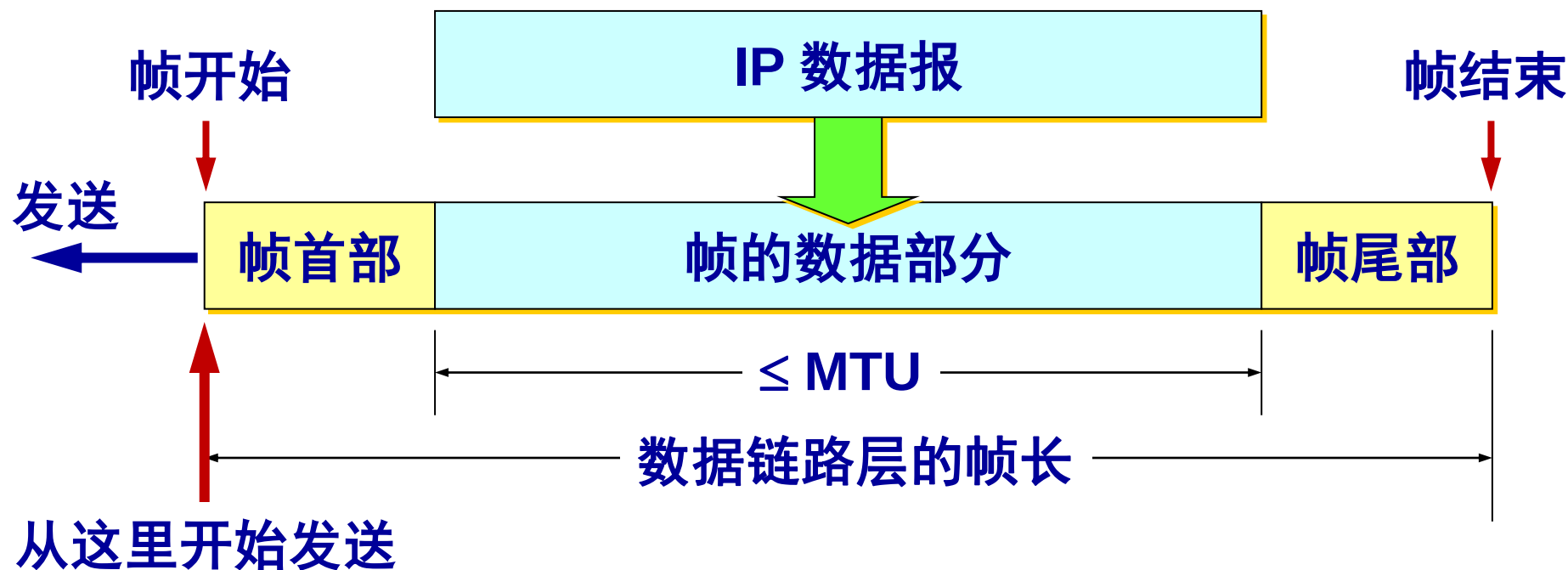
1. 数据链路层协议三个基本问题

数据链路层协议三个基本问题

- 数据链路层协议有许多种，但有三个基本问题则是共同的。这三个基本问题是：
 - 封装成帧
 - 透明传输
 - 差错检测

(1). 封装成帧

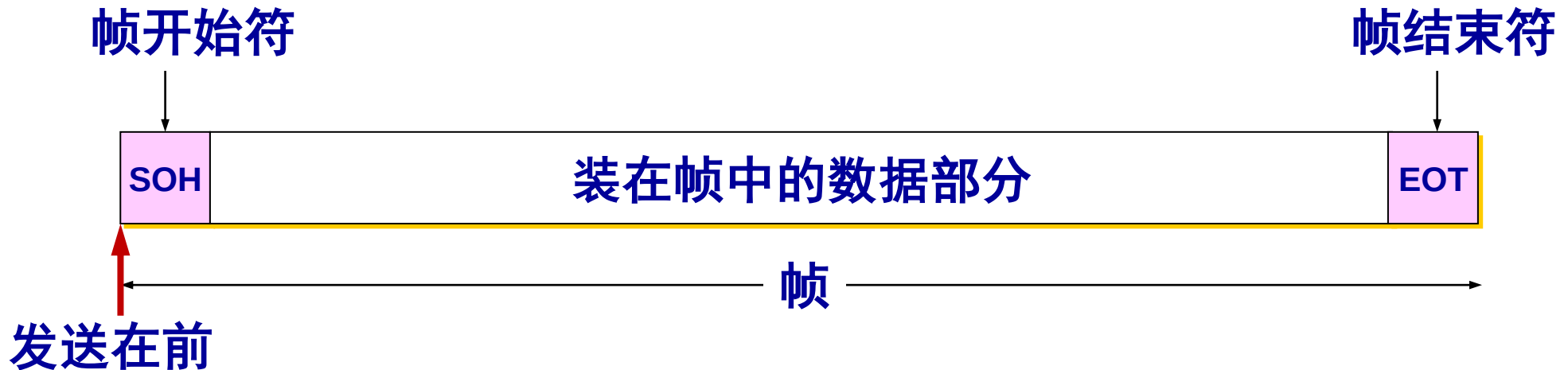
- **封装成帧** (framing) 就是在一段数据的前后分别添加首部和尾部，然后就构成了一个帧。确定帧的界限。
- 首部和尾部的一个重要作用就是进行**帧定界**。



用帧首部和帧尾部封装成帧

用控制字符进行帧定界的方法举例

- 当数据是由可打印的 ASCII 码组成的文本文件时，帧定界可以使用特殊的**帧定界符**。
- 控制字符 SOH (Start Of Header) 放在一帧的最前面，表示帧的首部开始。另一个控制字符 EOT (End Of Transmission) 表示帧的结束。



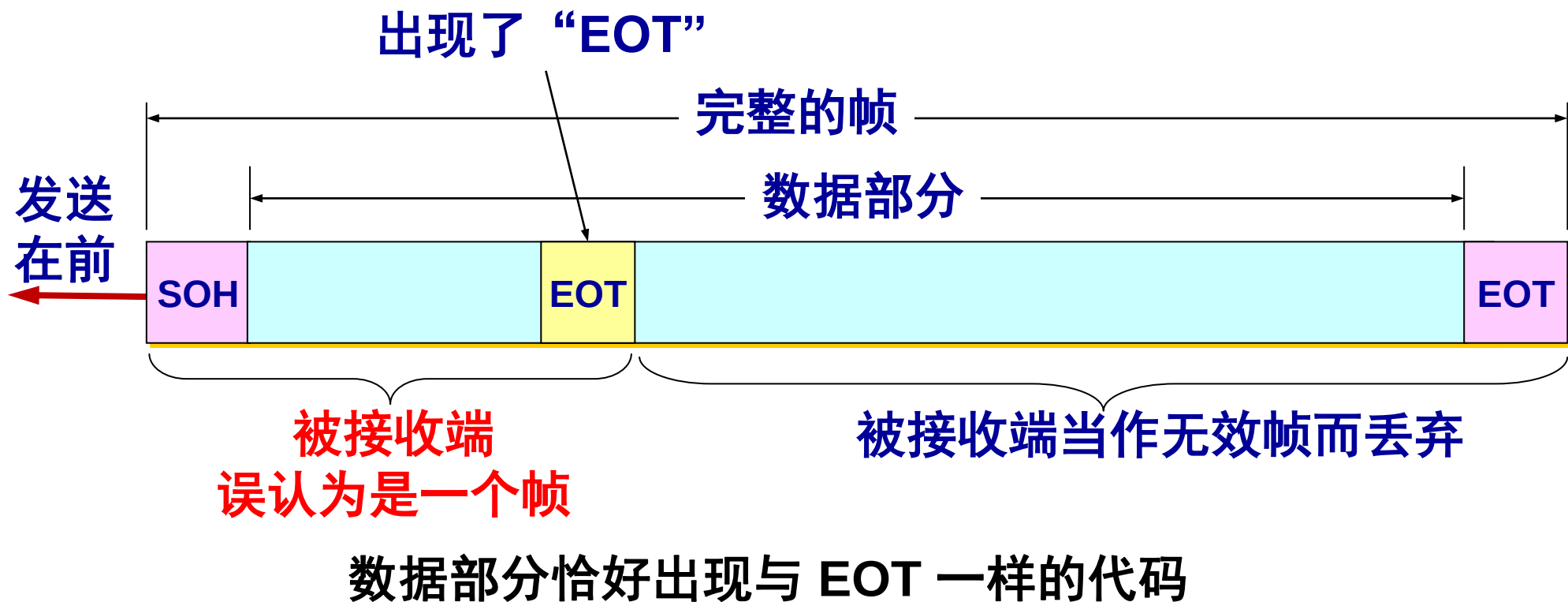
用控制字符进行帧定界的方法举例

思考

数据中包含SOH或者EOT，怎么办？

(2). 透明传输

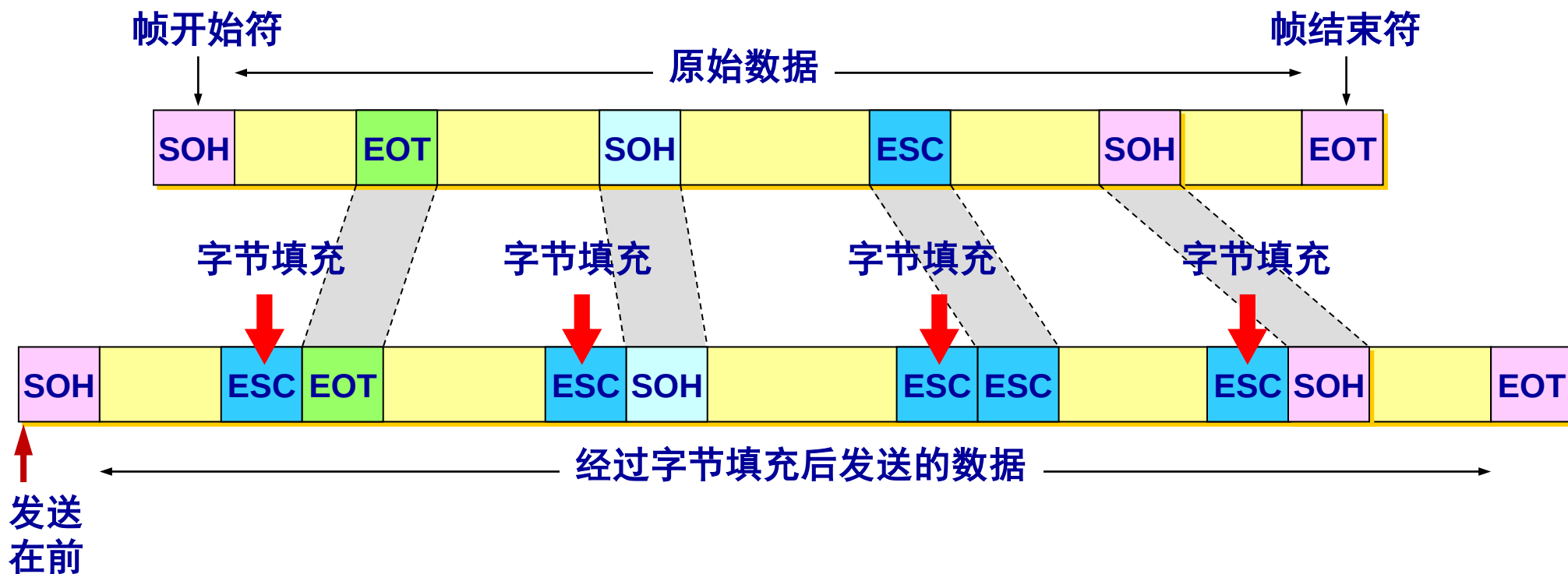
- 如果数据中的某个字节的二进制代码恰好和 SOH 或 EOT 一样，数据链路层就会错误地“找到帧的边界”。



解决透明传输问题

- **解决方法：字节填充 (byte stuffing) 或字符填充 (character stuffing)。**
 - 发送端的数据链路层在数据中出现控制字符“SOH”或“EOT”的前面**插入一个转义字符“ESC”** (其十六进制编码是 1B)。
 - 接收端的数据链路层在将数据送往网络层之前删除插入的转义字符。
- **如果转义字符也出现在数据当中**，那么应在转义字符前面插入一个转义字符 ESC。当接收端收到连续的两个转义字符时，就删除其中前面的一个。

用字节填充法解决透明传输的问题



用字节填充法解决透明传输的问题

(3). 差错检测

- 在传输过程中可能会产生**比特差错**：1 可能会变成 0 而 0 也可能变成 1。
- 在一段时间内，传输错误的比特占所传输比特总数的比率称为**误码率** BER (Bit Error Rate)。
- 误码率与信噪比有很大的关系。
 - 信噪比(Signal-Noise Ratio)： 一个电子设备或者电子系统中信号与噪声的比例
- 为了保证数据传输的可靠性，在计算机网络传输数据时，必须采用各种差错检测措施。

帧检验序列 FCS

- 在数据后面添加上的冗余码称为帧检验序列 FCS (Frame Check Sequence)。
- CRC 是一种常用的检错方法，FCS 可以用 CRC 这种方法得出

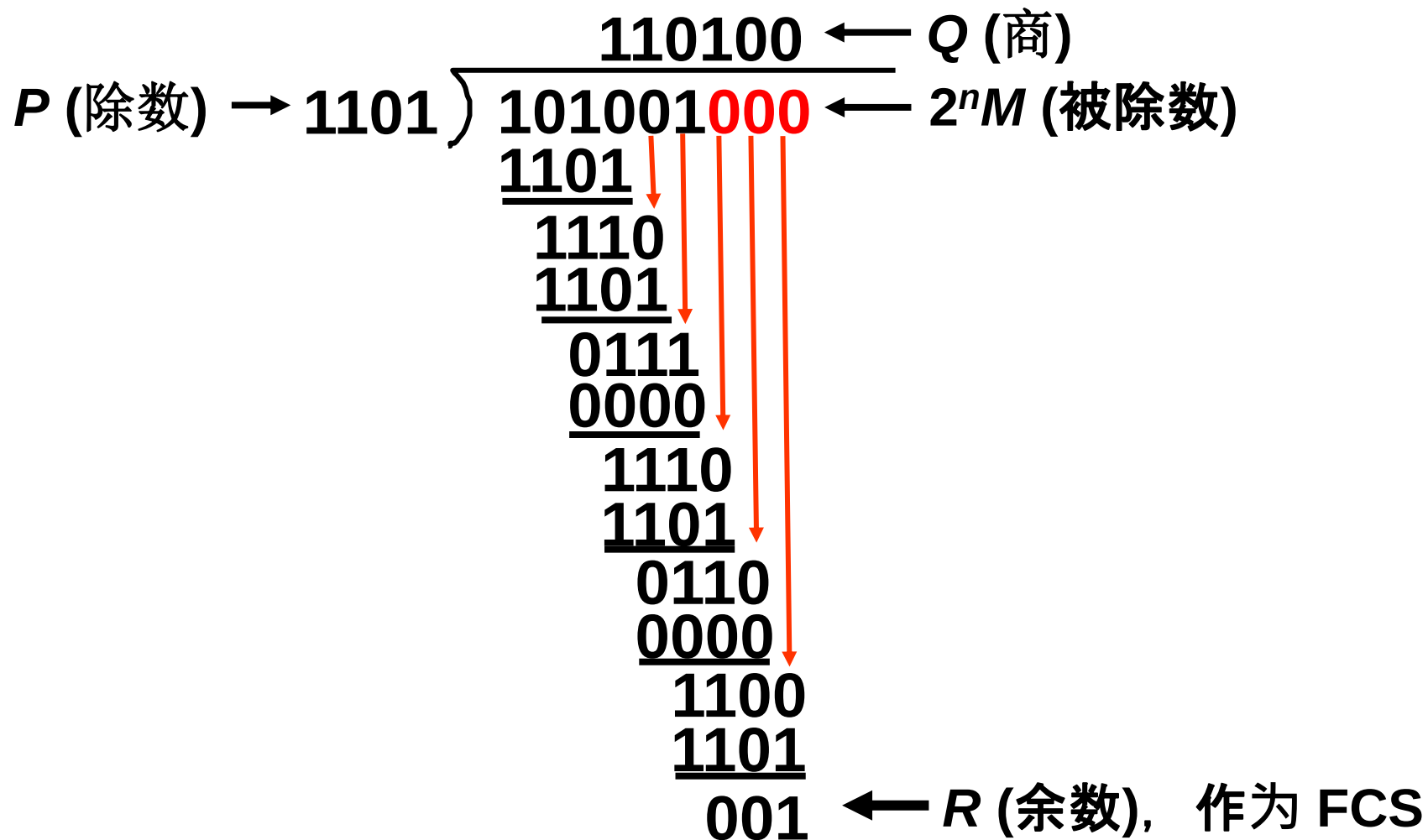
循环冗余检验的原理

- 在数据链路层传送的帧中，广泛使用了**循环冗余检验** CRC (Cyclic Redundancy Check) 的检错技术。
- 在数据的后面添加冗余码一起发送
 - 101001
 - 101001**001**

冗余码的计算举例

- 在发送端，先把数据划分为组。假定每组 k 个比特。
 - 假设待传送的一组数据 $M = 101001$ （现在 $k = 6$ ）
- 用二进制的模 2 运算进行 2^n 乘 M 的运算，这相当于在 M 后面添加 n 个 0
 - $n = 3$
 - $2^n M = 101001000$
- 得到的 $(k + n)$ 位的数除以事先选定好的长度为 $(n + 1)$ 位的除数 P
 - 除数 $P = 1101$
 - 商 $Q = 110101$, 余数 $R = 001$
- 将余数 R 作为冗余码拼接在数据 M 后面发送出去
 - 101001001

循环冗余检验的原理说明



接收端对收到的每一帧进行 CRC 检验

- 除以同样的除数 P
 - (1) 若得出的余数 $R = 0$ ，则判定这个帧没有差错，就**接受** (accept)。
 - (2) 若余数 $R \neq 0$ ，则判定这个帧有差错，就**丢弃**。
- 但这种检测方法并不能确定究竟是哪一个或哪几个比特出现了差错。
- 只要经过严格的挑选，并使用位数足够多的**除数** P ，那么出现检测不到的差错的概率就很小很小。
 - 生成 多项式

无差错接受 不等于 可靠传输

- 仅用循环冗余检验 CRC 差错检测技术只能做到**无差错接受** (accept)。
 - **无差错接受**：凡是接受的帧（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
- 要做到“可靠传输”（即发送什么就收到什么）就必须再加上确认和重传机制。
 - 帧编号、确认、重传机制

在哪里做可靠传输

■ 通信线路质量差

- 在链路层实现可靠传输：高级数据链路控制HDLC (High-level Data Link Control)

■ 通信线路质量好

- 将可靠传输交给上一层来做
- 点对点协议PPP (Point-to-Point Protocol)

数据链路层使用的信道

数据链路层使用的信道主要有以下两种类型：

- 点对点信道。
 - 一对一的点对点通信方式。
- 广播信道。
 - 一对多的广播通信方式，因此过程比较复杂。广播信道上连接的主机很多，因此必须使用专用的共享信道协议来协调这些主机的数据发送。

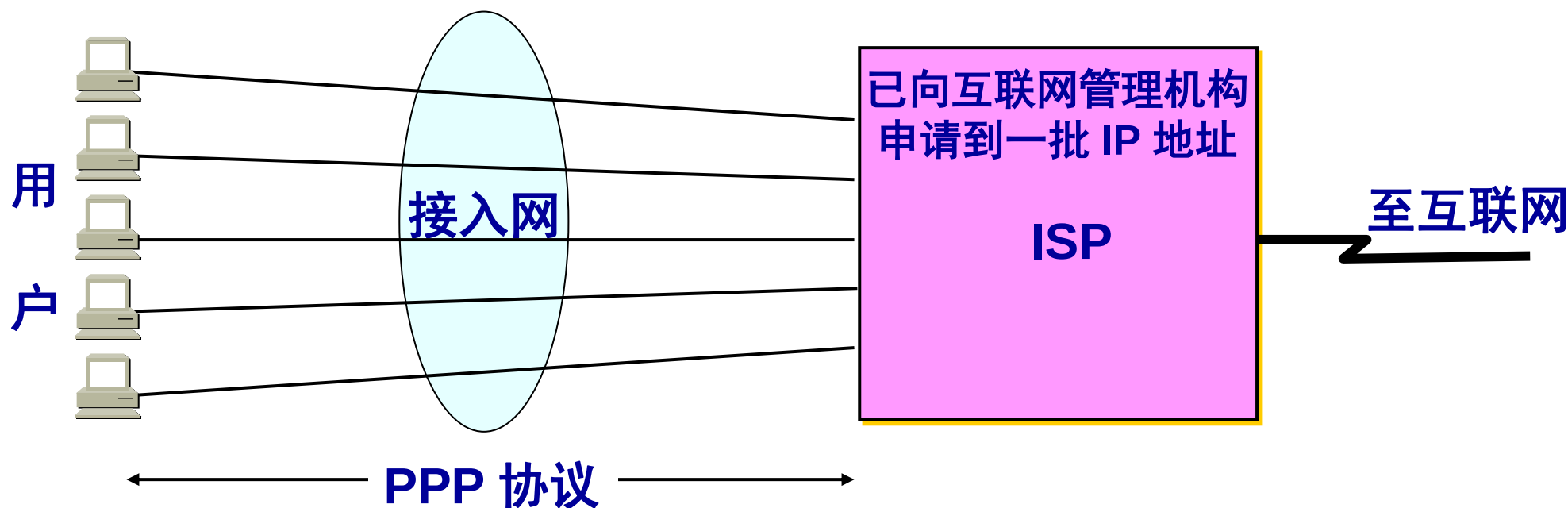
2. 点对点协议PPP

3.2.1 PPP 协议的特点

- 对于点对点的链路，目前使用得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年就已成为互联网的正式标准。

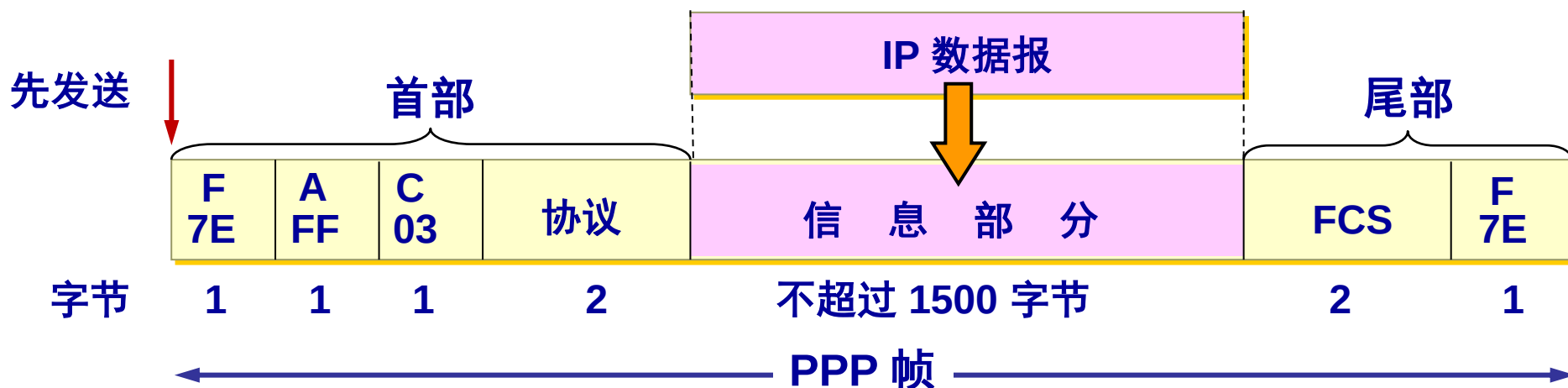
用户到 ISP 的链路使用 PPP 协议

- 用户使用拨号电话线接入互联网时，用户计算机和 ISP 进行通信时所使用的数据链路层协议就是 PPP 协议。

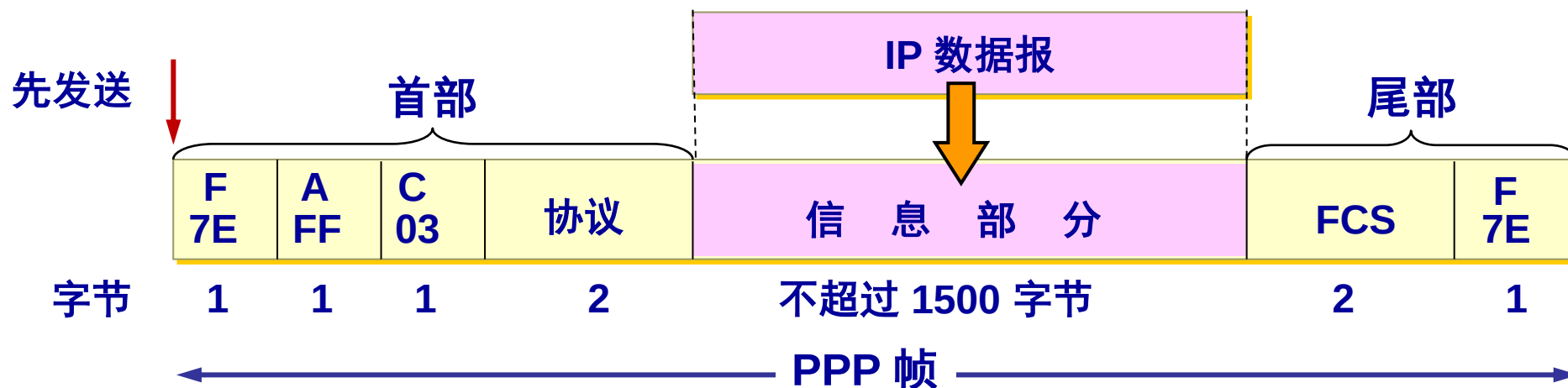


(1) 封装成帧

- 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
 - 地址字段 A 只置为 0xFF。地址字段实际上并不起作用。
 - 控制字段 C 通常置为 0x03。
- PPP 是面向字节的，所有的 PPP 帧的长度都是整数字节。



PPP 协议的帧格式



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。

(2) 透明传输

- 当 PPP 用在同步传输链路时，协议规定采用硬件来完成比特填充（和 HDLC 的做法一样）
- 当 PPP 用在异步传输时，就使用一种特殊的字符填充法。

字符填充

■ 转义字符：0x7D

- 将信息字段中出现的每一个 0x7E 字节转变成为 2 字节序列 (0x7D, 0x5E)。
- 若信息字段中出现一个 0x7D 的字节, 则将其转变成为 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。

零比特填充

- PPP 协议用在 SONET/SDH 链路时，使用同步传输（一连串的比特连续传送）。这时 PPP 协议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。

零比特填充

信息字段中出现了和
标志字段 F 完全一样
的 8 比特组合

0 1 0 0 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 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

发送端填入 0 比特

接收端把 5 个连 1
之后的 0 比特删除

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

零比特的填充与删除

(3) 差错检测

- 帧检验序列FCS
 - 使用循环冗余检验

不提供使用序号和确认的可靠传输

- **PPP 协议之所以不使用序号和确认机制是出于以下的考虑：**
 - 在数据链路层出现差错的概率不大时，使用比较简单的 PPP 协议较为合理。
 - 在因特网环境下，PPP 的信息字段放入的数据是 IP 数据报。数据链路层的可靠传输并不能够保证网络层的传输也是可靠的。
 - 帧检验序列 FCS 字段可保证无差错接受。