

Computer Networks-Data Link Layer

Dr. Jiang RuoBing

jrb@ouc.edu.cn

信息学院南楼B411房间



中国海洋大学
OCEAN UNIVERSITY OF CHINA

第 3 章 数据链路层

3.1 使用点对点信道的数据链路层

3.1.1 数据链路和帧

3.1.2 三个基本问题

3.2 点对点协议 PPP

3.2.1 PPP 协议的特点

3.2.2 PPP 协议的帧格式

3.2.3 PPP 协议的工作状态

第3章 数据链路层（续）

3.3 使用广播信道的数据链路层

补充：媒体接入控制典型问题及协议

3.3.1 局域网的数据链路层

3.3.2 CSMA/CD 协议

3.4 使用广播信道的以太网

3.4.1 使用集线器的星形拓扑

3.4.2 以太网的信道利用率

3.4.3 以太网的 MAC 层

3.5 扩展的以太网

3.5.1 在物理层扩展以太网

3.5.2 在数据链路层扩展以太网

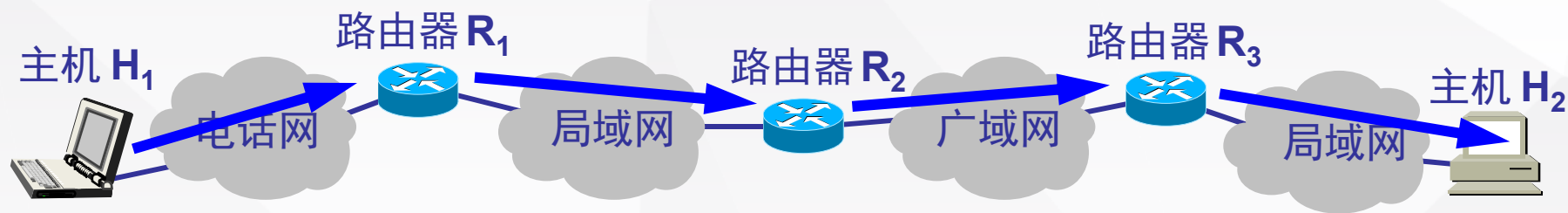
数据链路层

数据链路层使用的信道主要有：

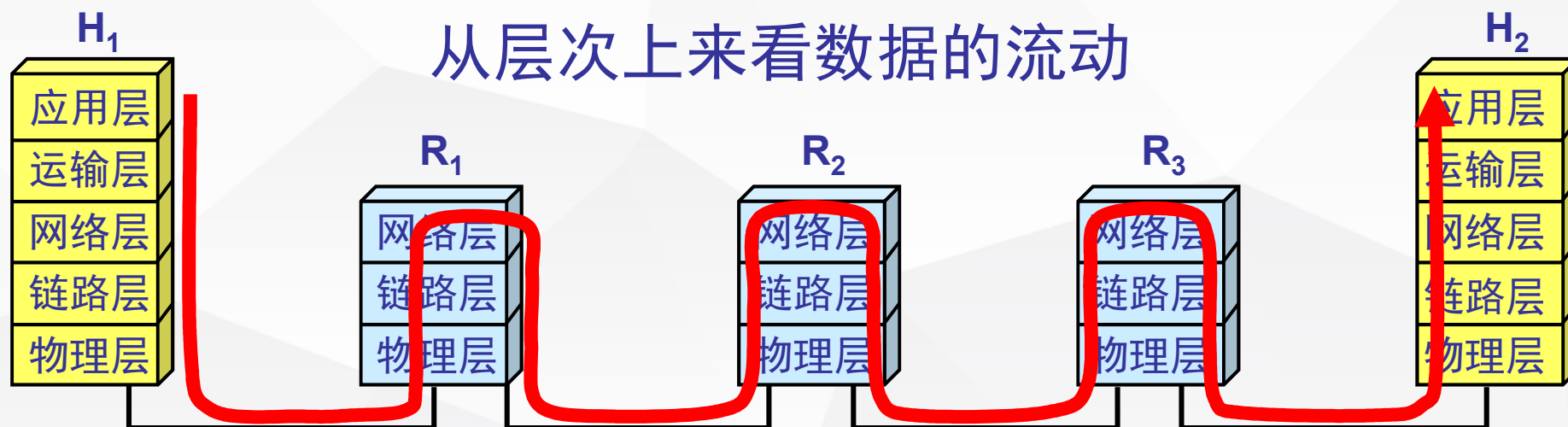
- 点对点信道：一对一的点对点通信方式。
- 广播信道：一对多的广播通信方式；广播信道上连接的主机很多，共享信道协议来协调这些主机的数据发送。

数据链路层的简单模型

主机 H_1 向 H_2 发送数据

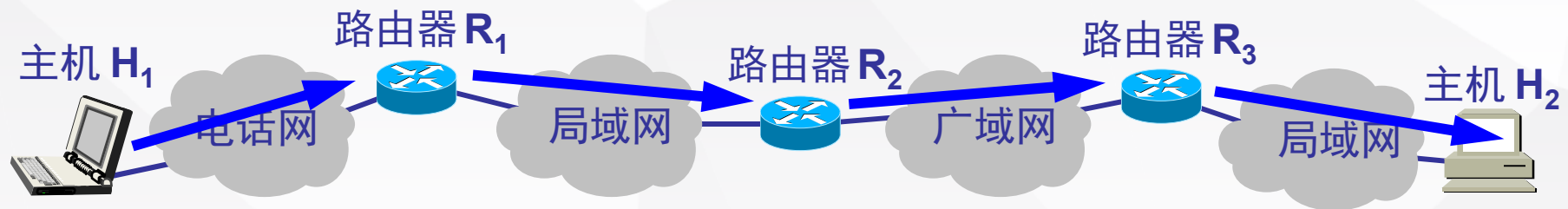


从层次上来看数据的流动

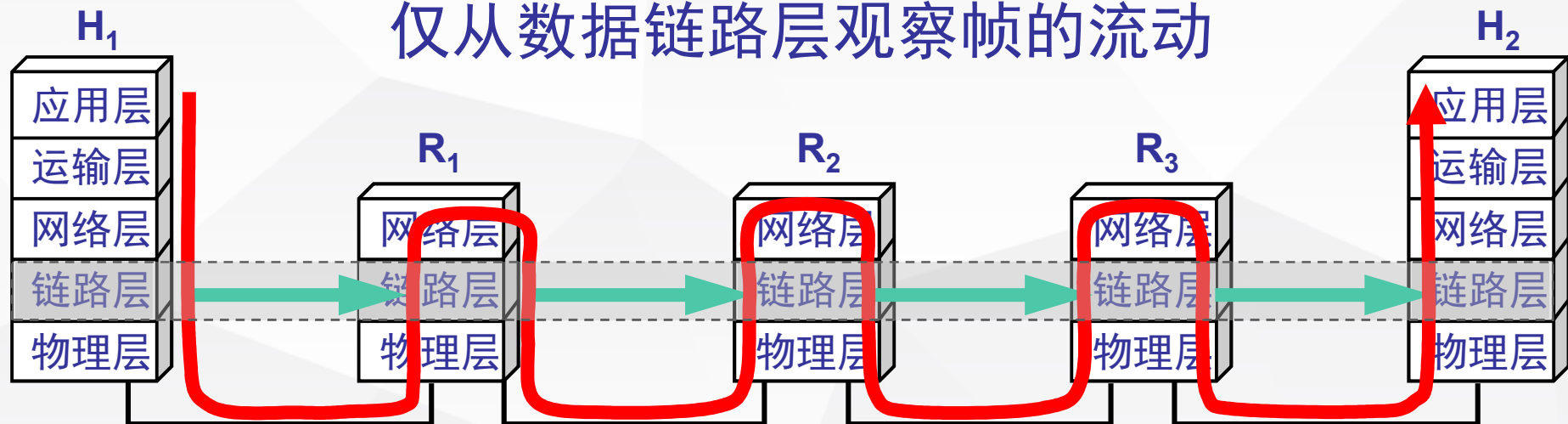


数据链路层的简单模型（续）

主机 H_1 向 H_2 发送数据



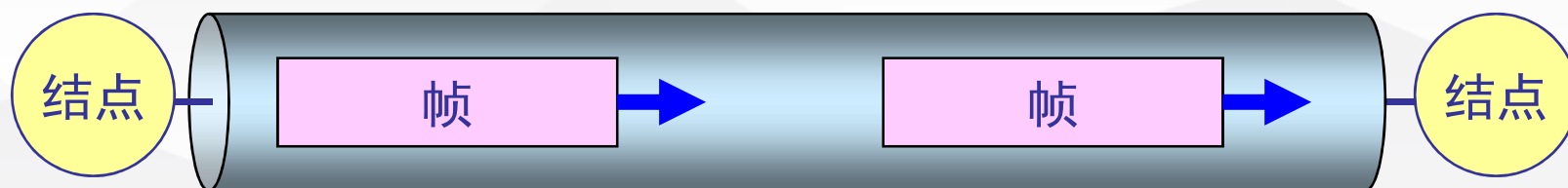
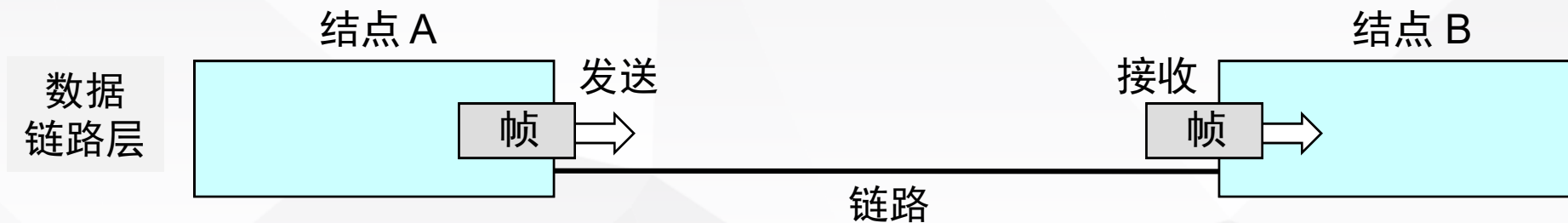
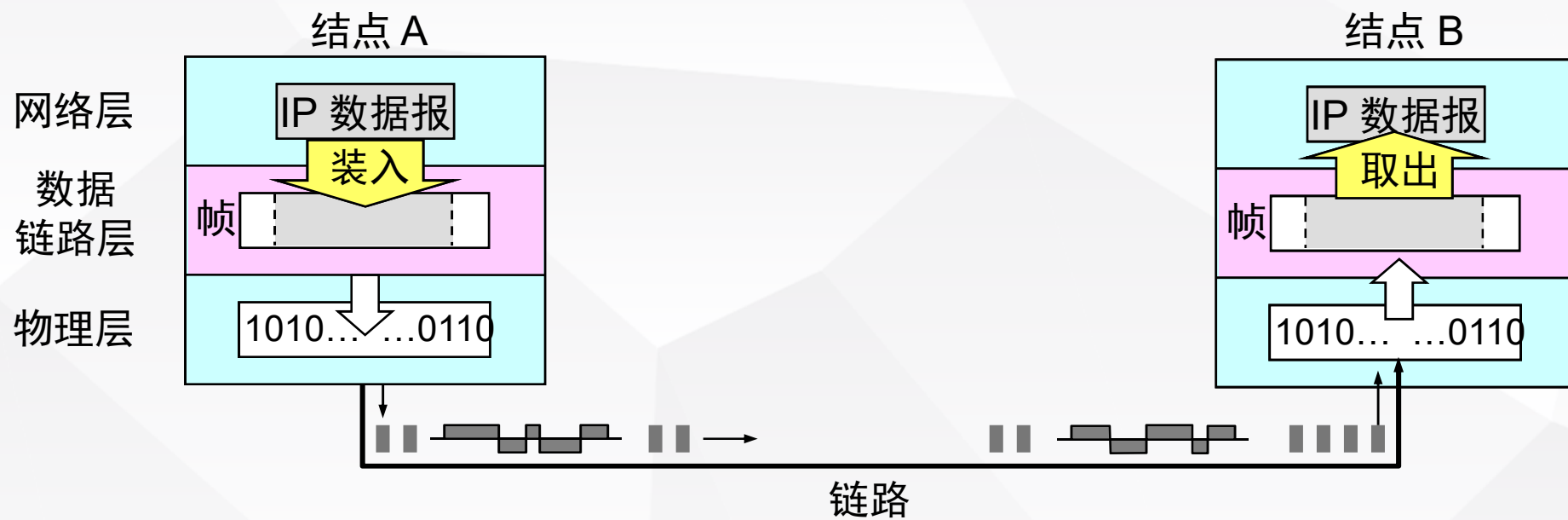
仅从数据链路层观察帧的流动



3.1 使用点对点信道的数据链路层

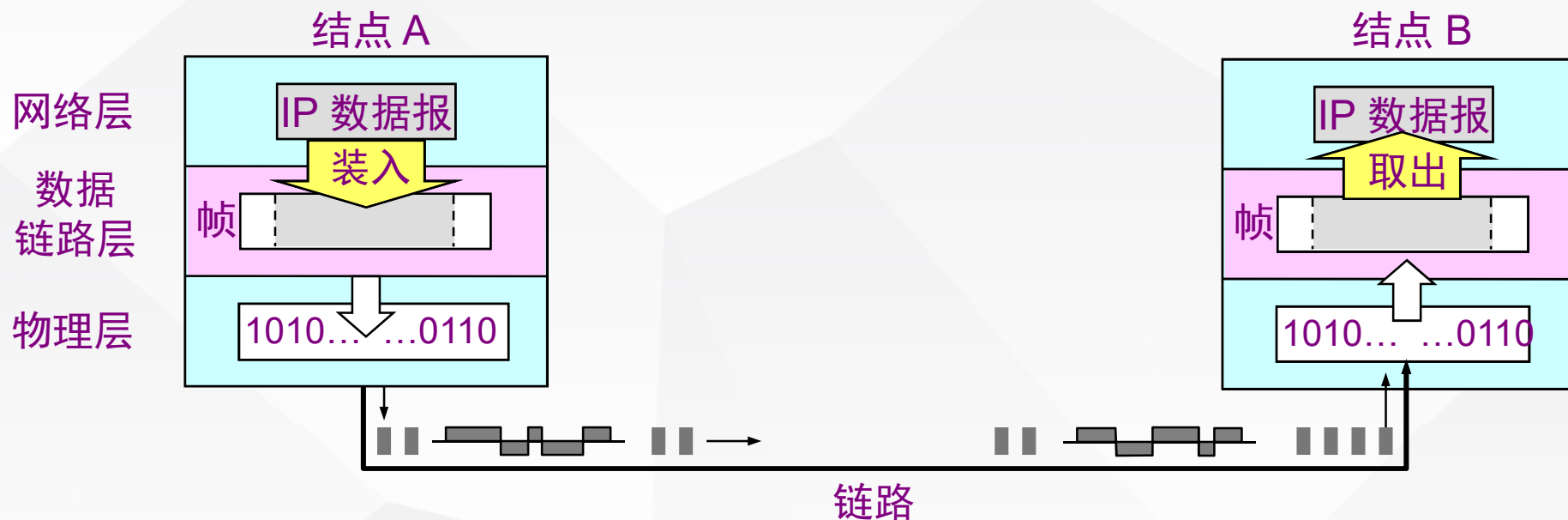
3.1.1 数据链路和帧

- **链路(link)**是一条无源的点到点的物理线路段，中间没有任何其他的交换结点。
 - 一条链路只是一条通路的一个组成部分。
- **数据链路(data link)** 物理线路 + 通信协议（硬件和软件）
 - 使用适配器（即网卡）来实现这些协议的硬件和软件。
 - 网卡都包括了数据链路层和物理层这两层的功能。



数据链路层传送的是帧

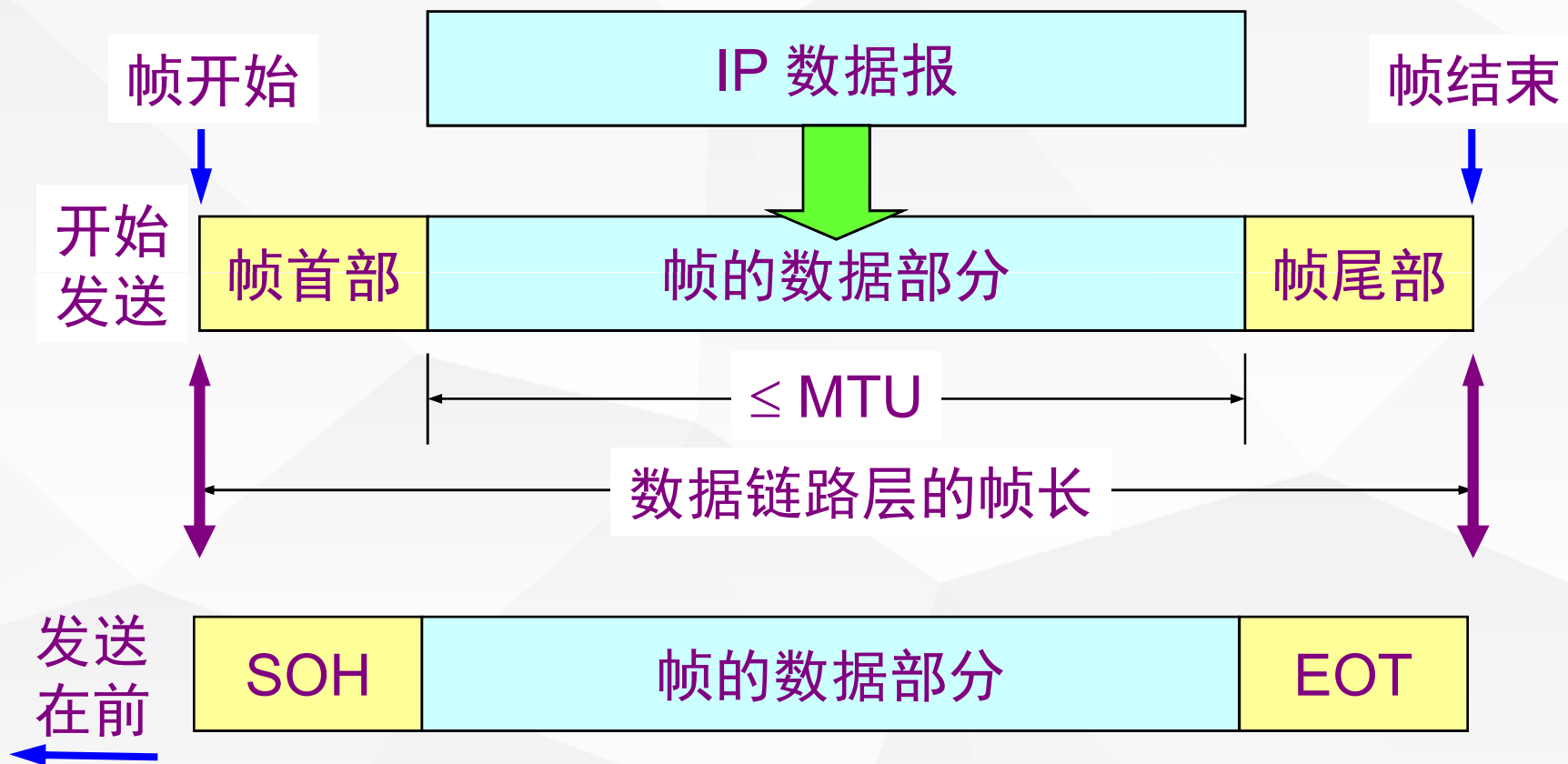
3.1.2 三个基本问题



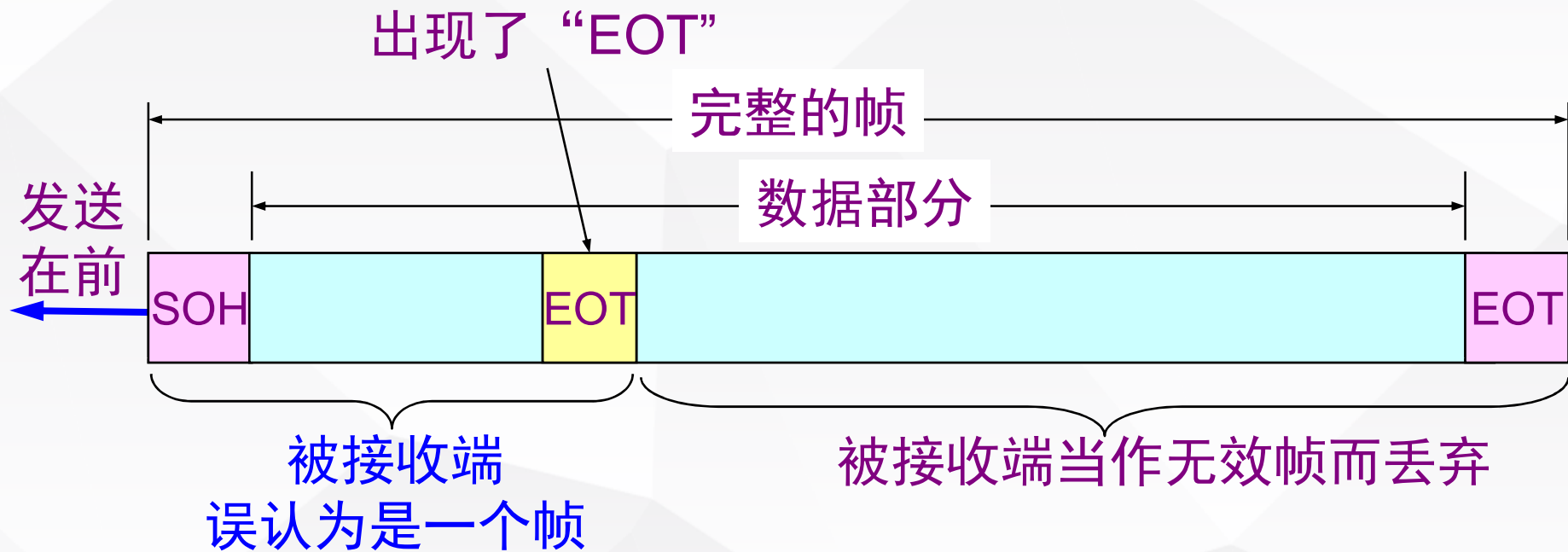
(1) 封装成帧 (2) 透明传输 (3) 差错控制

1. 封装成帧

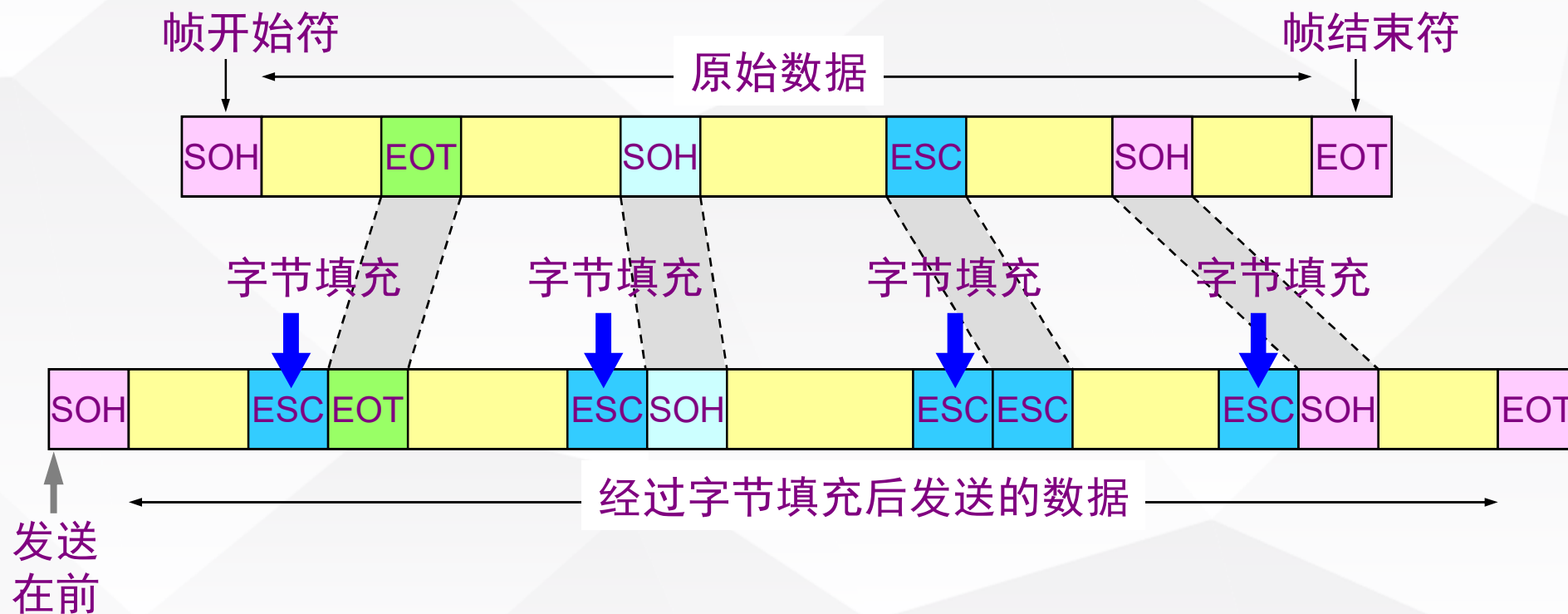
- 封装成帧(framing)——帧定界
 - 数据的前后添加首部和尾部，确定帧的界限。



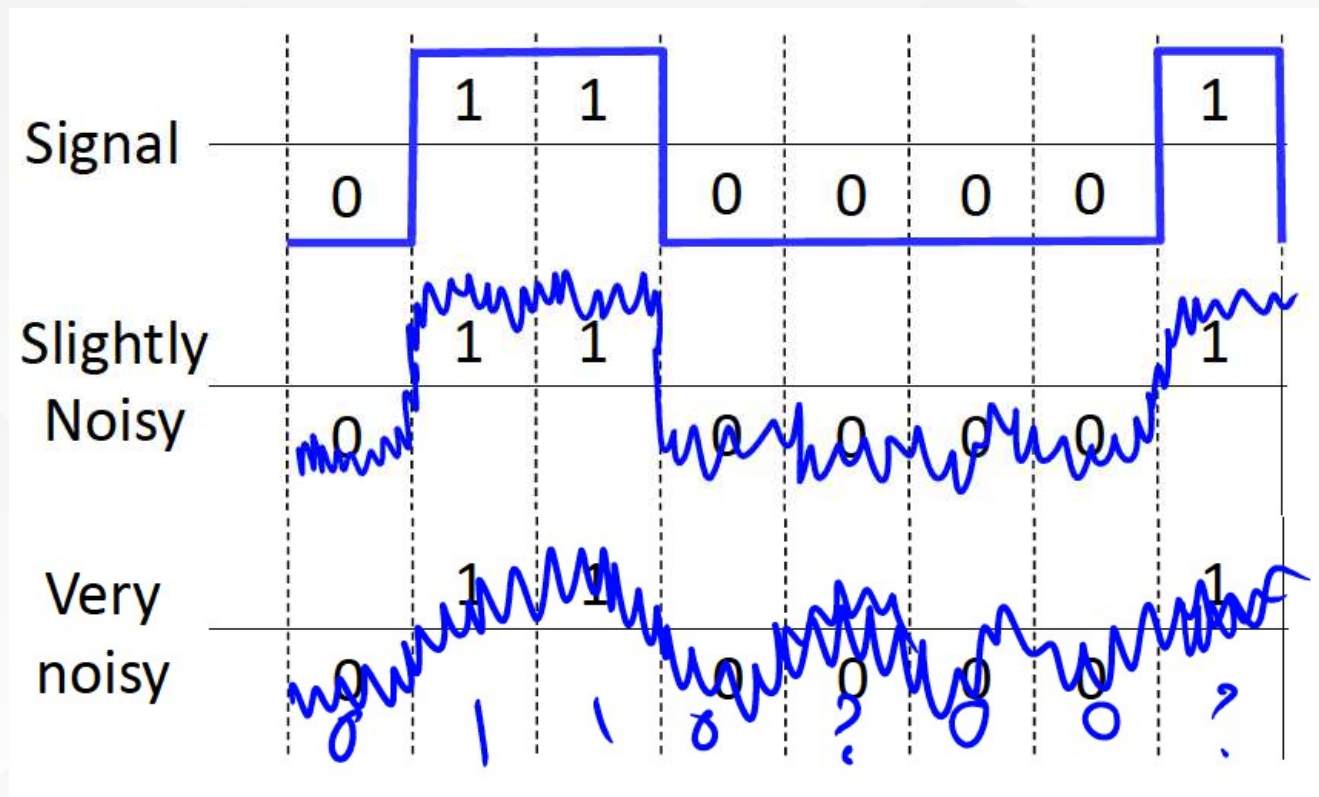
2. 透明传输



用字节填充法解决透明传输的问题



3. 差错检测



- 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。

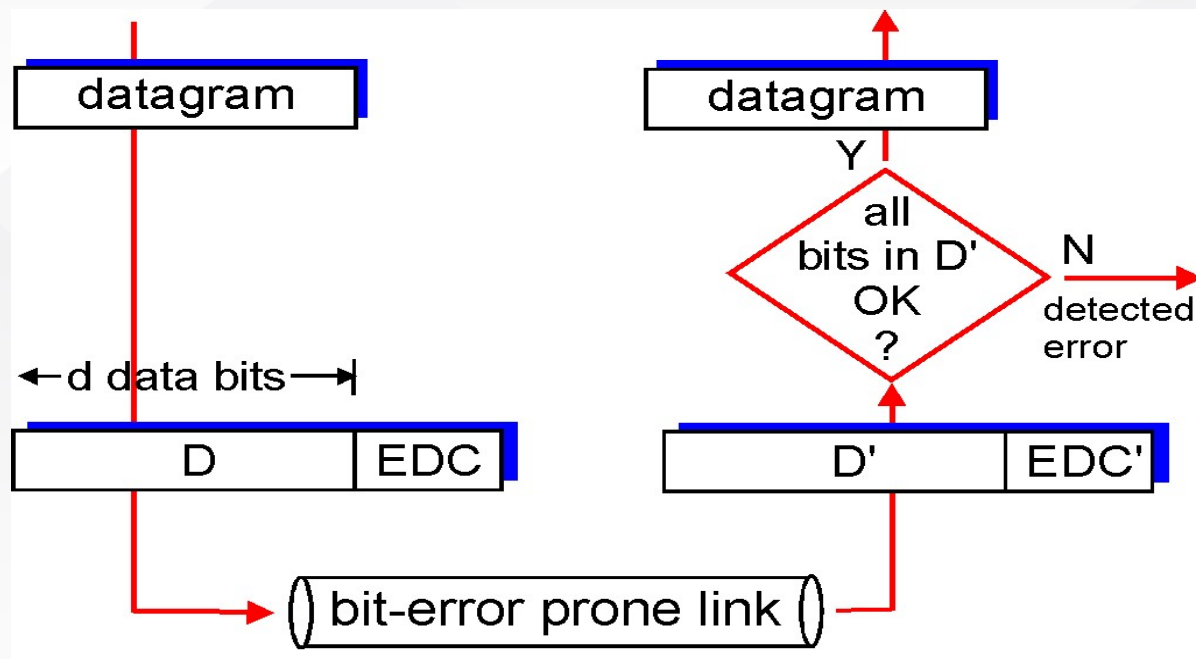
帧检验序列 FCS

- 帧检验序列 FCS (Frame Check Sequence) : 在数据后面添加上的冗余码(EDC)。
- 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。

最直接的校验方法

- 传输两遍
 - 有区别
- 效果：
 - 检错：1
 - 校错：0
 - 失效：2
 - 开销：n

错误检测



- 奇偶校验

- 效果：

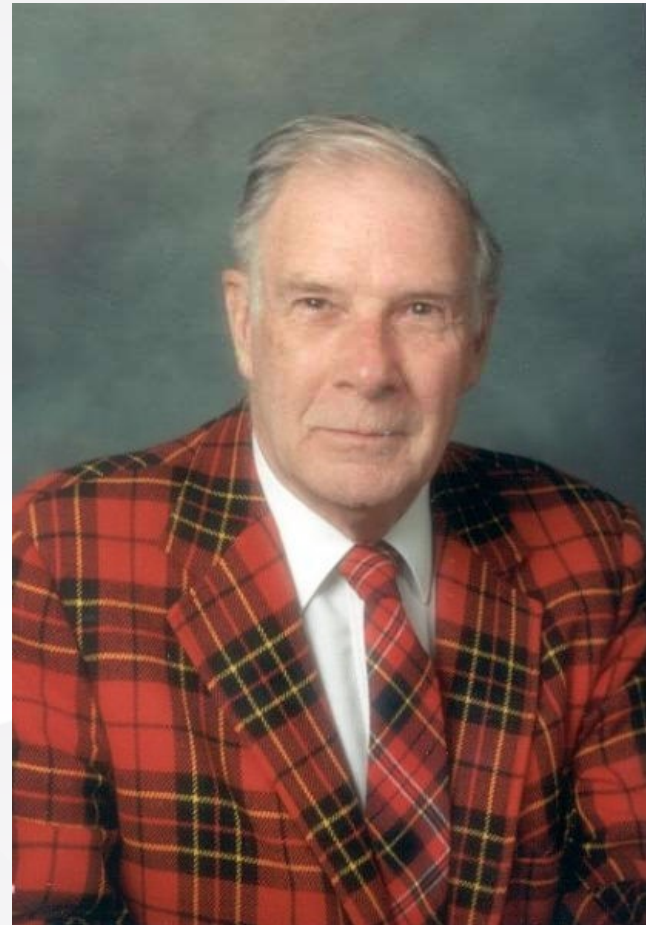
- 检错：1
- 纠错：0
- 失效：2
- 开销：1

EDC= 帧检验序列 (冗余码, r bits)

D = 待传送数据 (数据码, d bits)

R.W. Hamming (1915-1998)

- Error Detecting and Error Correcting Codes", BSTJ, 1950
- "You and Your Research", 1986



海明距离

- 距离:
 - 从一个序列到另一个序列需要反转的bit数
 - 1:111, 0: 000
 - Distance:3
- Hamming码距
 - Hamming码中任意一对码之间的最短距离
 - 1:111, 0: 000, 码距: 3
- 检错
 - 码距为 d 的Hamming码, 检错: $d-1$ 位。
 - 001,010,100
 - 011,101,110
- 校错:
 - 码距为 $2d+1$ 的Hamming码, 校错: d 位

循环冗余检验的原理

目标:

$$D \cdot 2^r \text{ XOR } R = mG$$

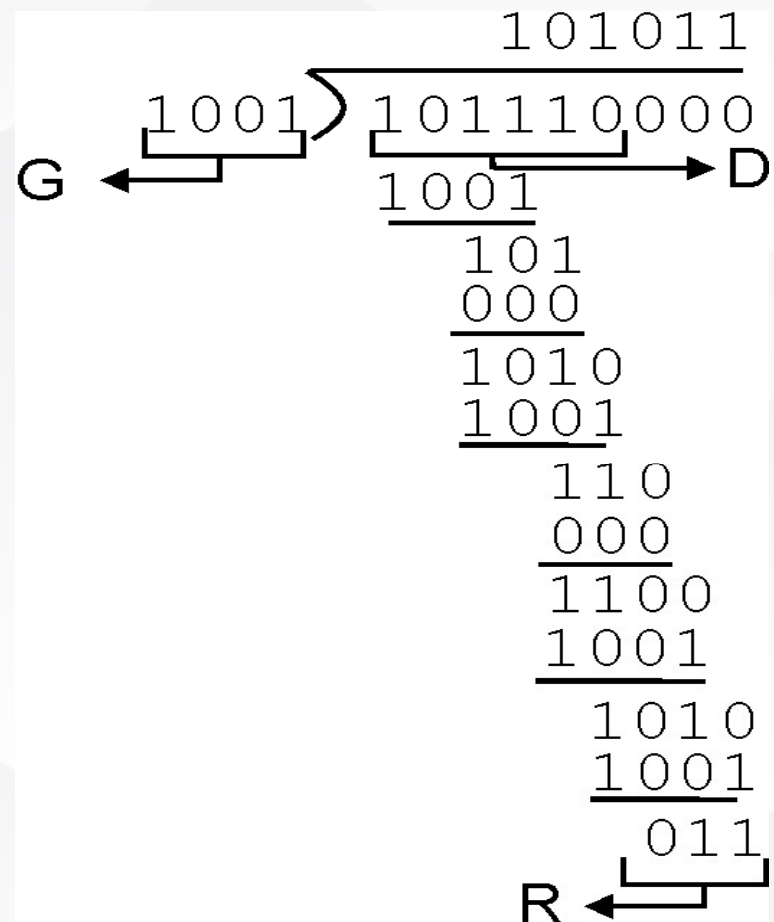
$$r = G \text{ 的长度} - 1$$

相当于:

$$D \cdot 2^r = mG \text{ XOR } R$$

即:

$$R = \text{remainder}\left[\frac{D \cdot 2^r}{G}\right]$$



冗余码的计算 --CRC检验

- $k = 6, D = 101001$ 。
- 设 $n = 3$, 除数 $G = 1101$,
 - 被除数是 $2^n M = 101001000$ 。
 - 模 2 运算的结果是：
 - 商 $Q = 110101$, 余数 $R = 001$ 。
 - 把余数 R 作为冗余码添加在数据 M 的后面发送出去。发送的数据是: $2^n M + R$
即: 101001001 , 共 $(k + n)$ 位。

循环冗余检验举例

$$\begin{array}{r} 110101 \leftarrow Q \text{ (商)} \\ G(\text{除数}) \rightarrow 1101 \overline{) 101001000} \leftarrow 2^n M \text{ (被除数)} \\ \underline{1101} \\ 1110 \\ \underline{1101} \\ 0111 \\ \underline{0000} \\ 1110 \\ \underline{1101} \\ 0110 \\ \underline{0000} \\ 1100 \\ \underline{1101} \\ 001 \leftarrow R \text{ (余数), 作为EDC} \end{array}$$

接收端对收到的每一帧进行 CRC 检验

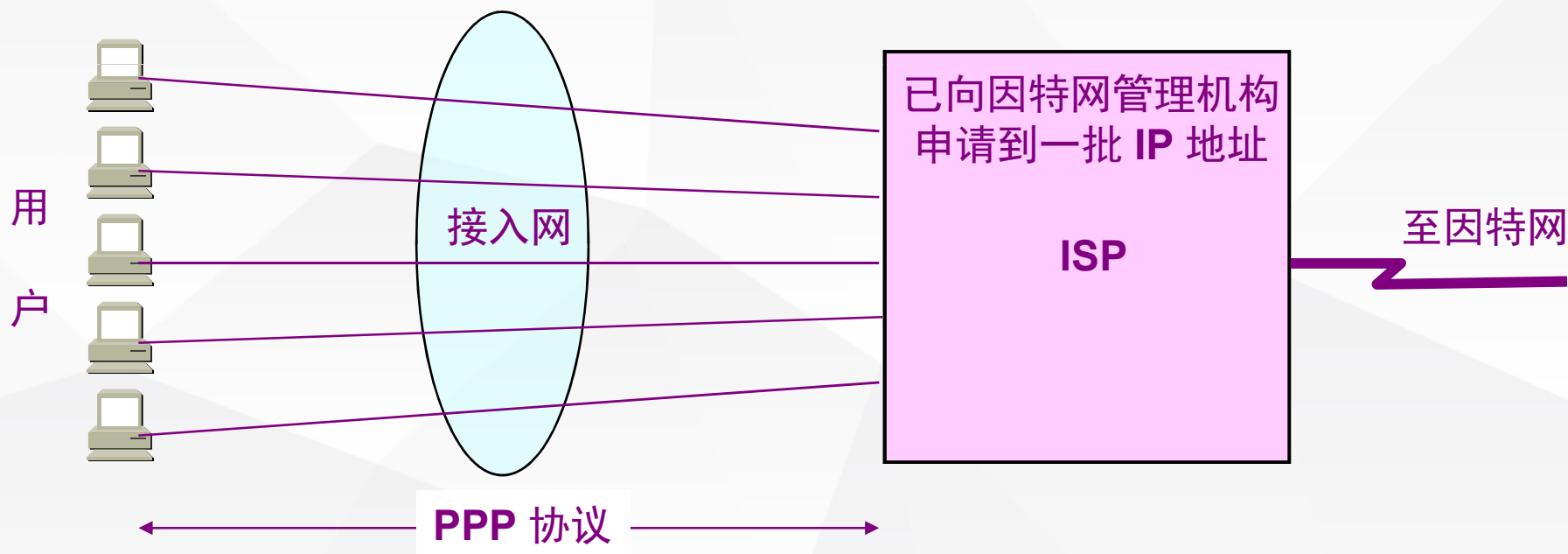
- (1) 若得出的余数 $R = 0$, 则判定这个帧没有差错 , 就接受(accept)。
- (2) 若余数 $R \neq 0$, 则判定这个帧有差错 , 就丢弃。
- 但这种检测方法并不能确定究竟是哪一个或哪几个比特出现了差错。
- 只要经过严格的挑选 , 并使用位数足够多的除数 G , 那么出现检测不到的差错的概率就很小很小。

应当注意

- 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)。
 - 凡是接受的帧（即**非丢弃的帧**，以非常接近于 1 的概率认为在传输过程中没有产生差错
 - 有差错的帧就丢弃而不接受
 - 要做到“**可靠传输**”，即发送什么就收到什么，必须做到**无比特差错&无传输差错**，还需再加上**确认和重传**机制。

3.2 点对点协议 PPP

- PPP (Point-to-Point Protocol)
 - 使用得最多的数据链路层协议
 - 用户使用拨号电话线接入因特网时，一般都是使用 PPP 协议。



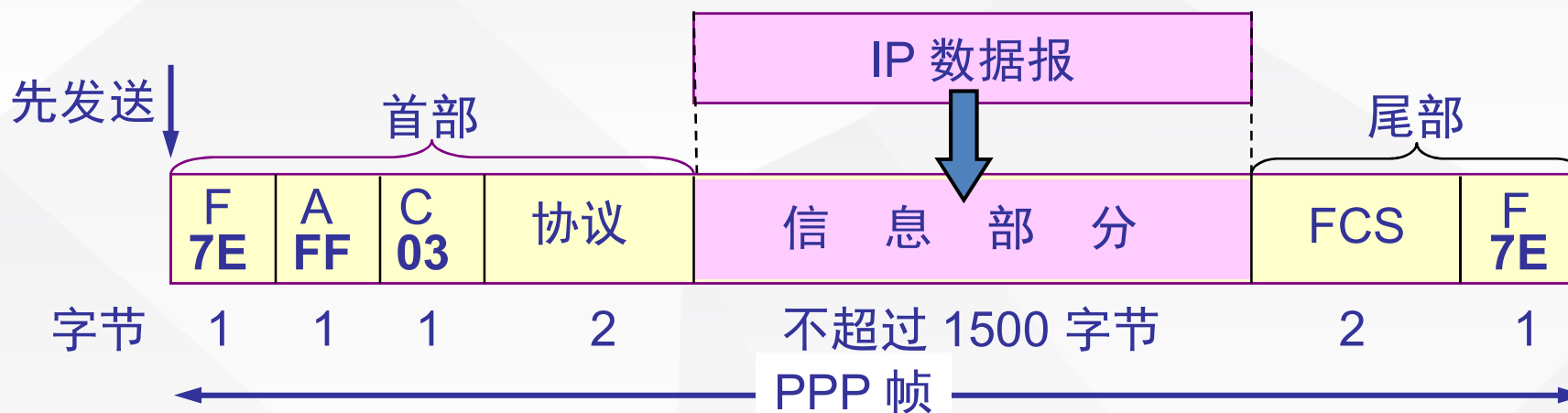
PPP 协议功能

- 简单——首要
- 基本功能
 - 封装成帧
 - 透明性
 - 差错检测
- 兼容
 - 多种网络层协议
 - 多种类型链路
 - 最大传送单元
 - 网络层地址协商
 - 数据压缩协商
- 检测连接状态
- 不具备的功能
 - 纠错
 - 流量控制
 - 序号
 - 多点线路
 - 半双工或单工链路

PPP 协议的组成

- 制订：1992；修订：1993 年、1994 年
- RFC 1661。
- PPP 组成
 - 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)**。
 - 网络控制协议 **NCP (Network Control Protocol)**。

PPP 协议的帧格式



- PPP 有一个 2 个字节的协议字段。
 - 当协议字段为 **0x0021** 时，PPP 帧的信息字段就是 **IP 数据报**。
 - 若为 **0xC021**，则信息字段是 **PPP 链路控制数据**。
 - 若为 **0x8021**，则表示这是网络控制数据。

3.2.2 PPP 协议的帧格式

- 头部字段：
 - 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。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。
- 帧结束：F = 0x7E

透明传输

- 异步传输
 - 字符填充法
- 同步传输
 - 硬件完成比特填充
- 字符填充
 - 标志字段0x7E → 2 字节序列(0x7D, 0x5E)。
 - 转义符 0x7D → 2 字节序列(0x7D, 0x5D)。
 - ASCII 码控制字符（小于 0x20 的字符）
 - → 2 字节序列(0x7D, 原字符加上0x20)。


零比特填充

- 同步传输
 - SONET/SDH 链路（光纤）
 - 一连串的比特连续传送，以帧为单位
- 帧定届：01111110
- 零比特填充
 - 发送端，发现5 个连续 1，立即填入一个 0。
 - 接收端，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

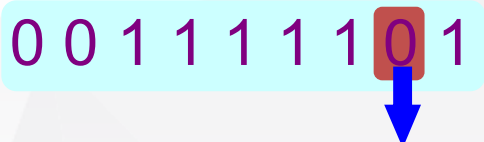
0 1 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0



发送端填入 0 比特

在接收端把 5 个连 1
之后的 0 比特删除

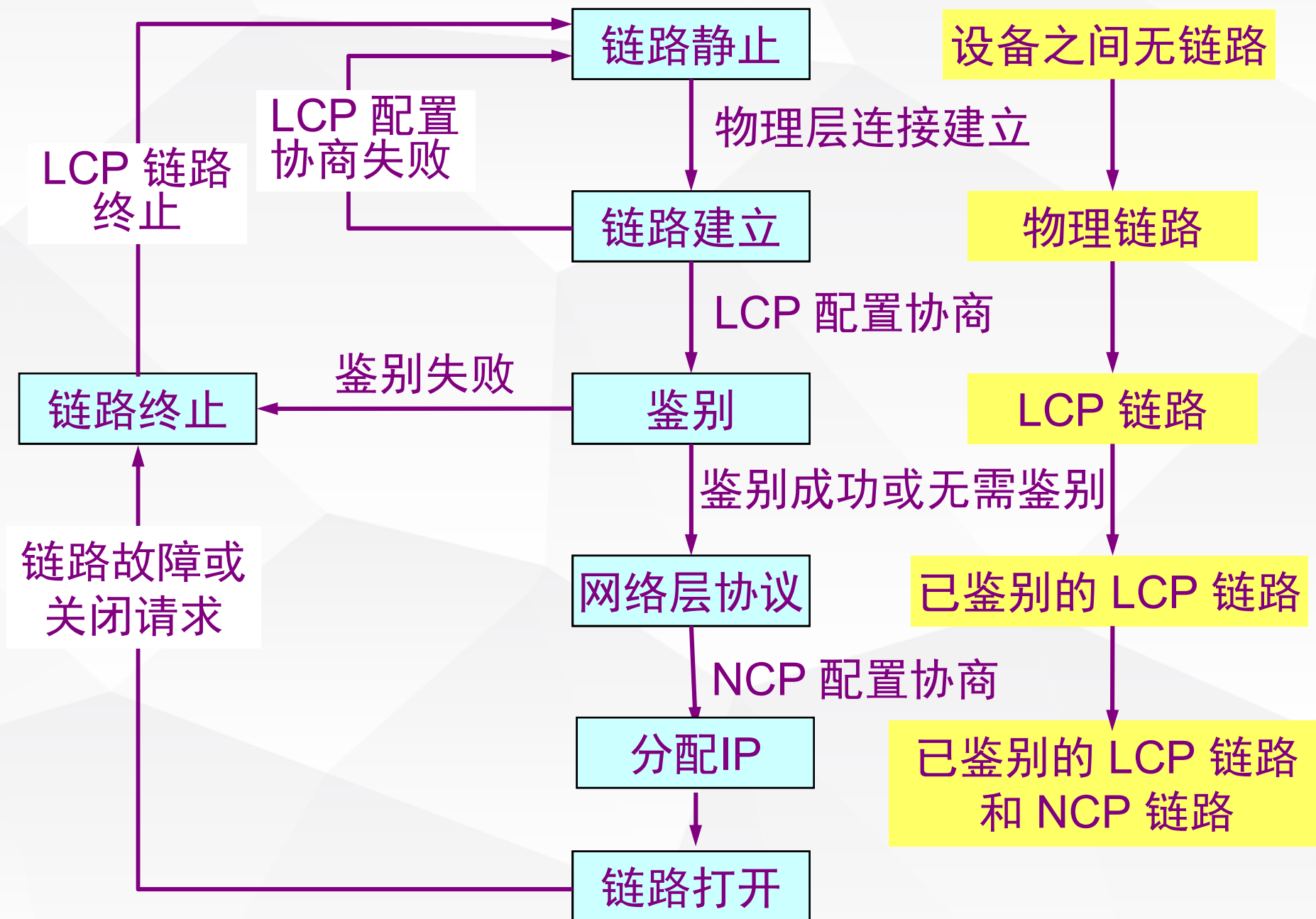
0 1 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0



接收端删除填入的 0 比特

不使用序号和确认

- PPP 协议不使用序号和确认机制
 - 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。



- 思考
- CSMA/CD 和频分、时分、码分复用的本质区别在哪里？哪种复用方式使用的范围最广？
- 为什么发现冲突还要进行人工干扰？