



石家庄铁道大学  
SHIJIAZHUANG TIEDAO UNIVERSITY

# 计算机网络

## 第 5 讲 数据链路层概述及PPP协议



# 上讲内容回顾

- ◆ 传输介质
- ◆ 网络设备
- ◆ Internet接入方式



# 本讲内容

- ◆ 数据链路和帧
- ◆ 数据链路层三个基本问题
- ◆ 点对点协议 PPP





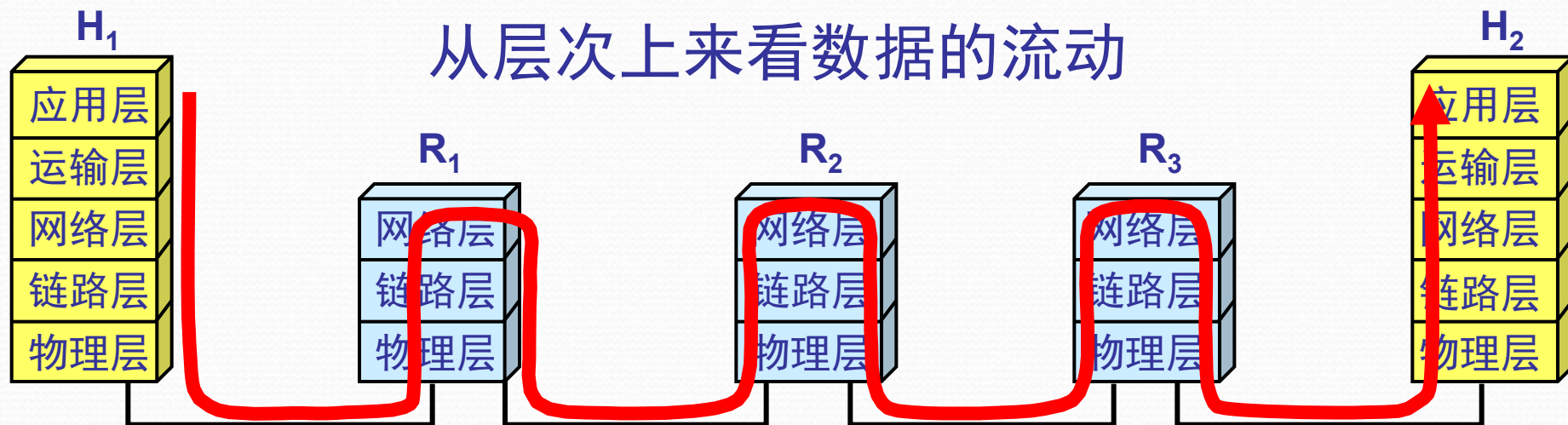
# 数据链路层

数据链路层使用的信道类型：

- **点对点信道**。这种信道使用一对一的点对点通信方式。
- **广播信道**。这种信道使用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。

# 数据链路层的简单模型

主机  $H_1$  向  $H_2$  发送数据



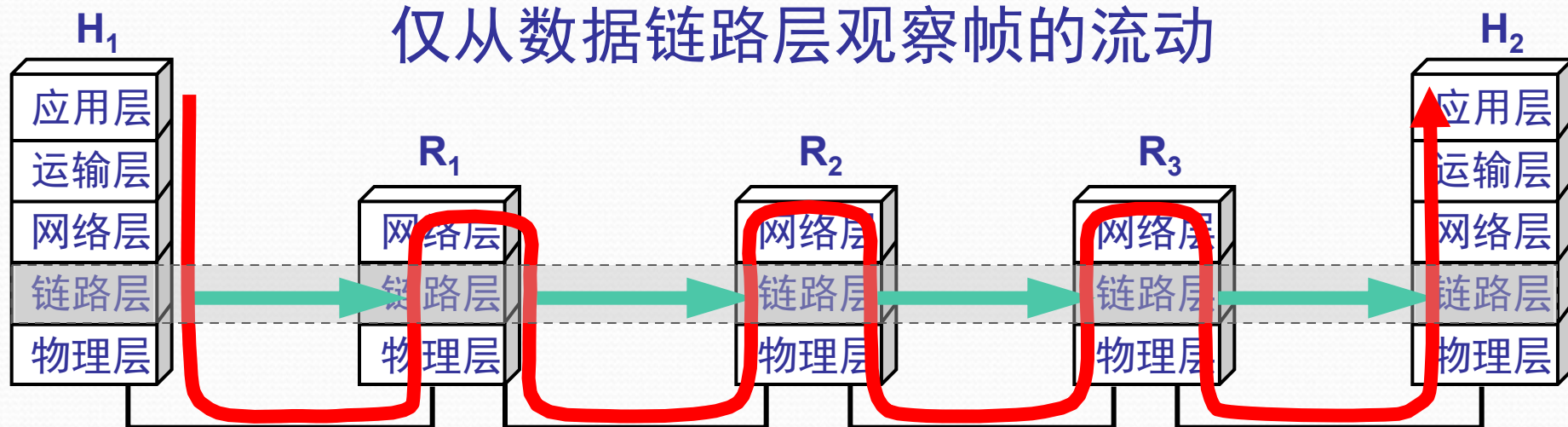


# 数据链路层的简单模型

主机  $H_1$  向  $H_2$  发送数据



仅从数据链路层观察帧的流动



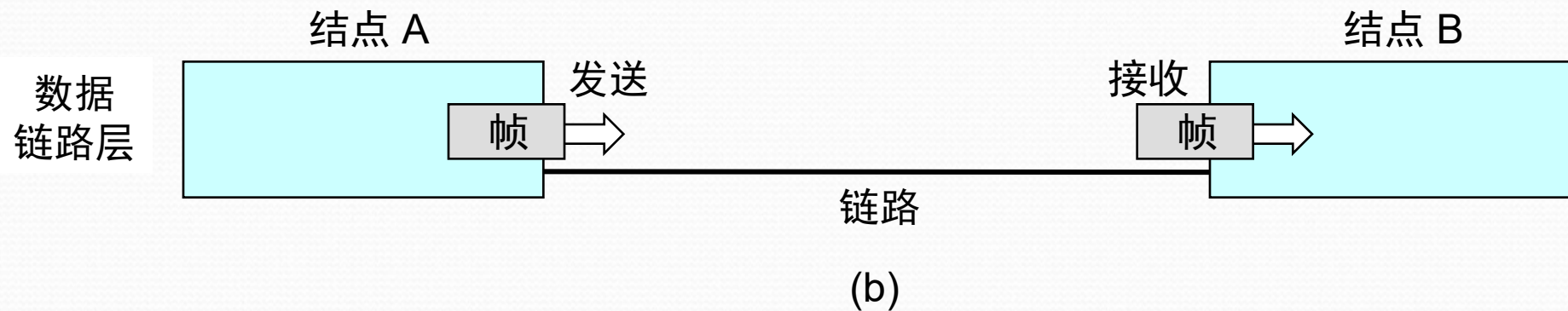
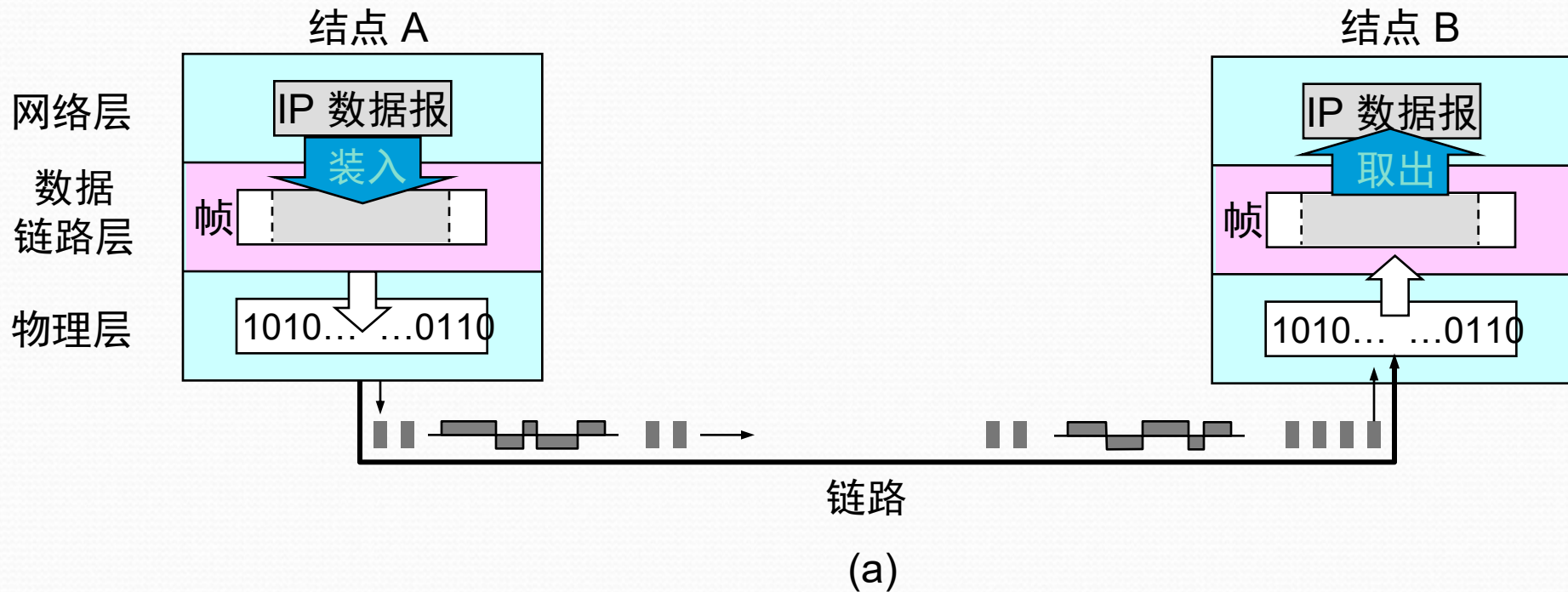


# 使用点对点信道的数据链路层

## 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)**除了物理线路外，还必须要有通信协议来控制这些数据的传输。若把实现这些协议的硬件和软件加到链路上，就构成了数据链路。
  - 现在最常用的方法是使用适配器（即网卡）来实现这些协议的硬件和软件。
  - 一般的适配器都包括了数据链路层和物理层这两层的功能。

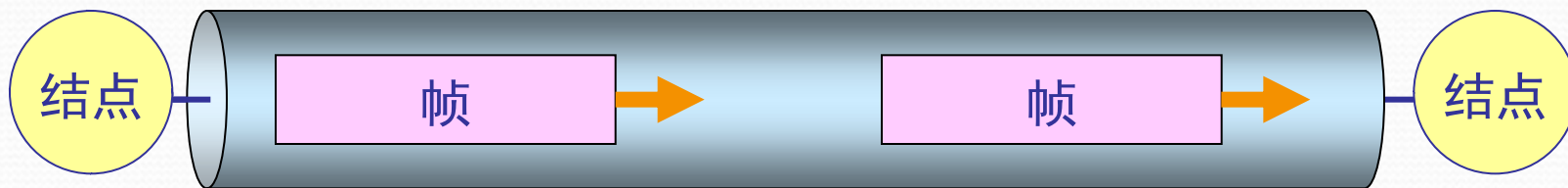
# 数据链路层传送的是帧





# 数据链路层像个数字管道

- 常常在两个对等的的数据链路层之间画出一个数字管道，而在这条数字管道上传输的数据单位是帧。



- 早期的数据通信协议曾叫作通信规程(procedure)。因此在数据链路层，规程和协议是同义语。

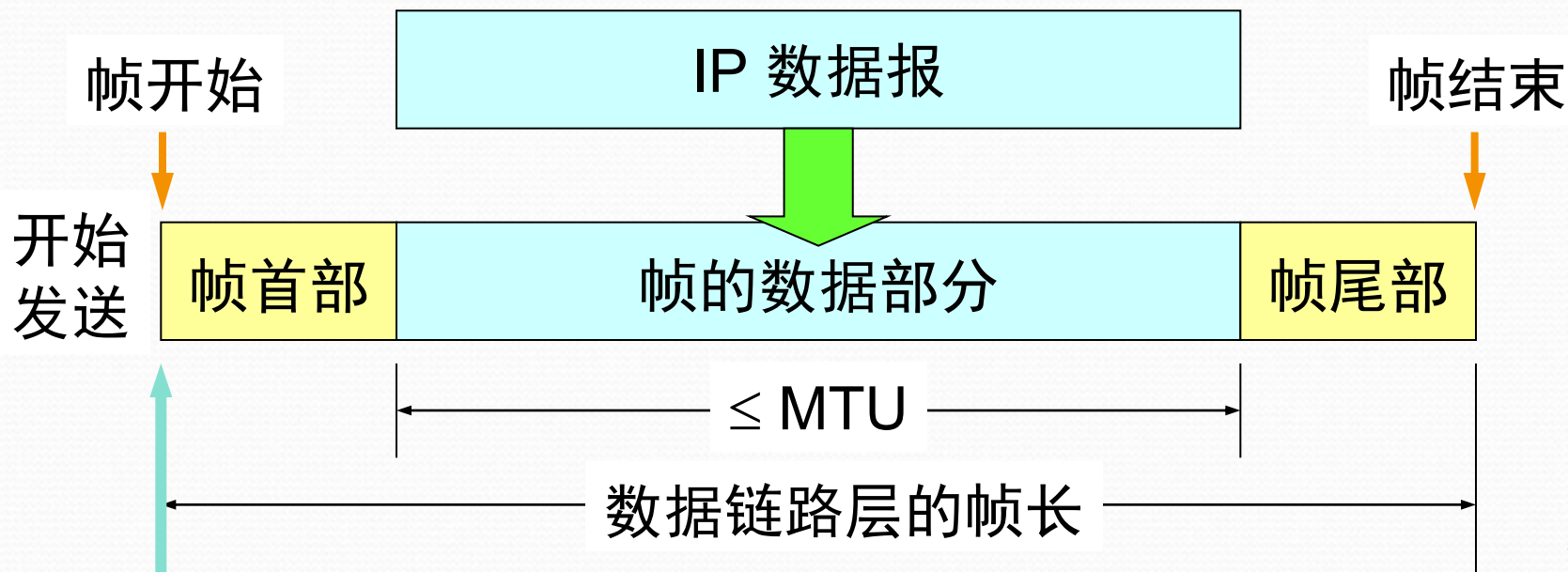


# 三个基本问题

- (1) 封装成帧
- (2) 透明传输
- (3) 差错控制

# 1. 封装成帧

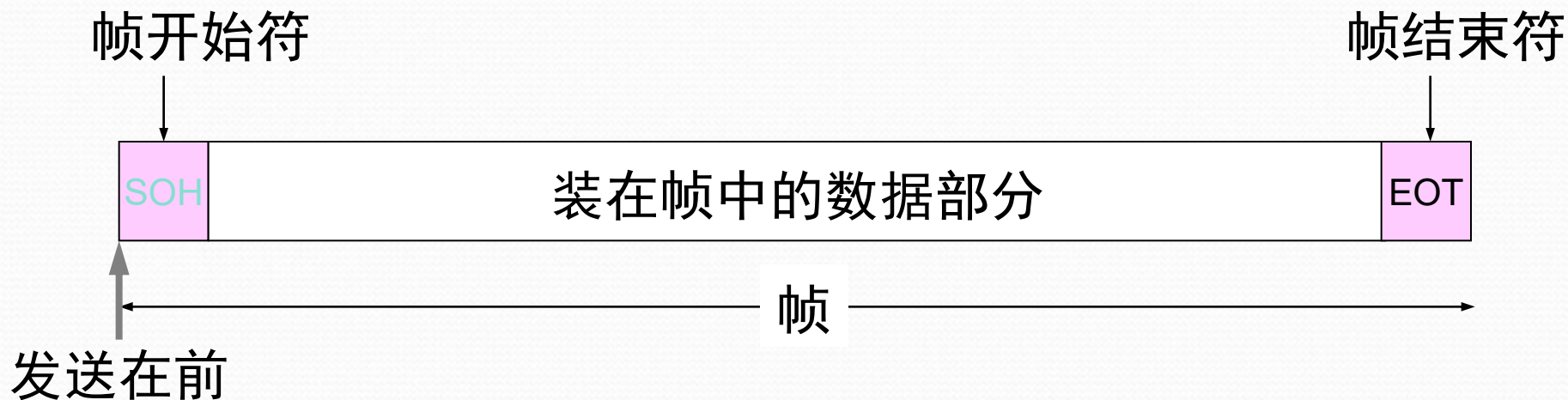
- 封装成帧(framing)就是在一段数据的前后分别添加首部和尾部，然后就构成了一个帧。确定帧的界限。
- 首部和尾部的一个重要作用就是进行帧定界。



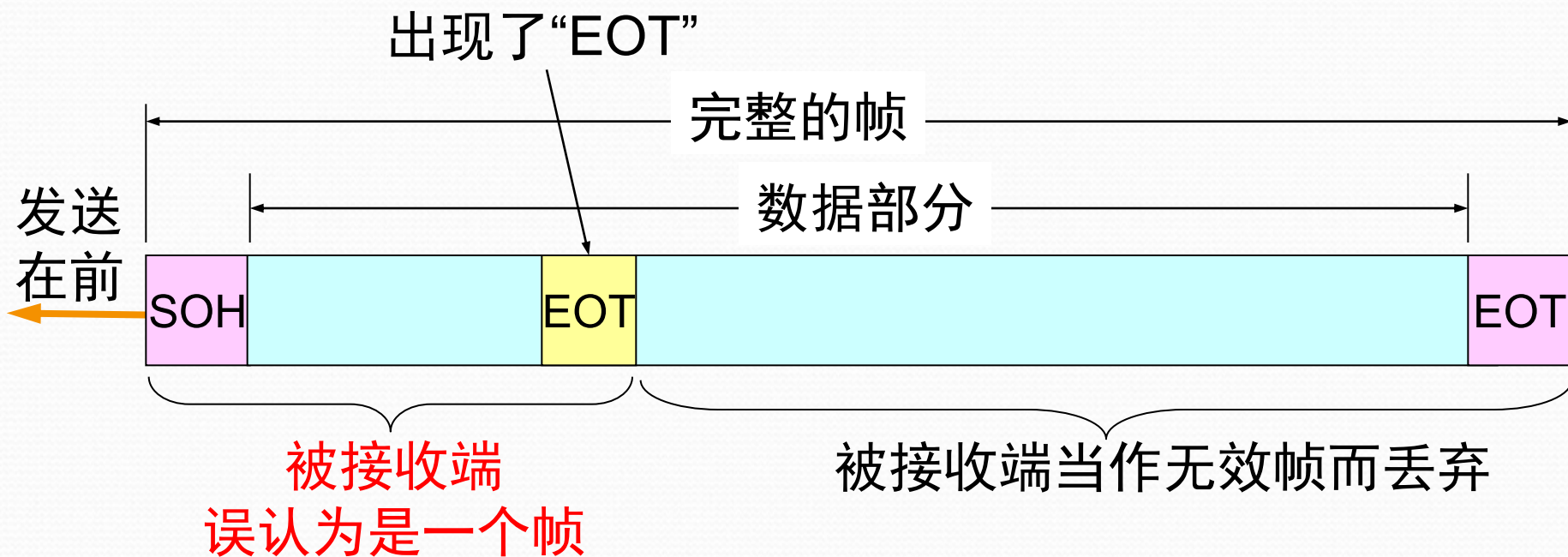




## 用控制字符进行帧定界的方法举例



## 2. 透明传输



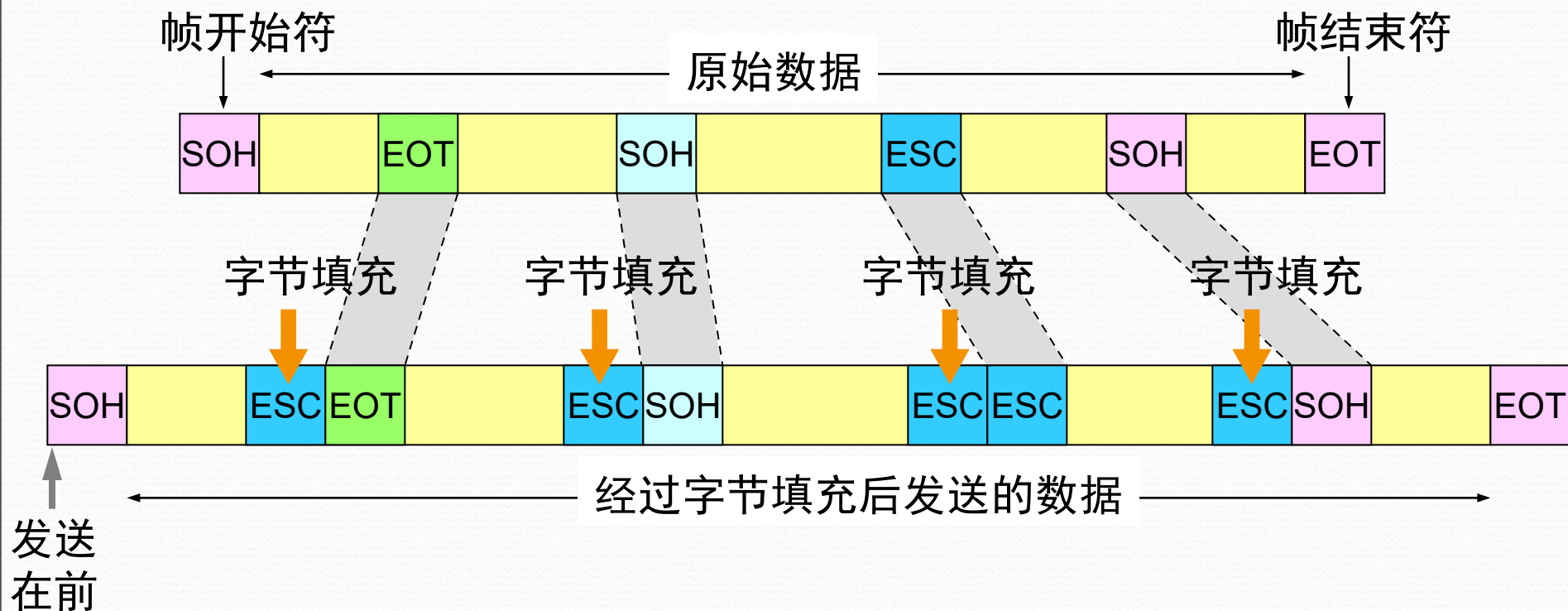


## 解决透明传输问题

- 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)。
- **字节填充**(byte stuffing)或**字符填充**(character stuffing)——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数据当中，那么应在转义字符前面插入一个转义字符。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





### 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。
- 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。



# 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 的检错技术。
- 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  $k = 6$ ）。我们在  $M$  的后面再添加供差错检测用的  $n$  位冗余码一起发送。





# 冗余码的计算

- 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。
- 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  $P$ ，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。



# 冗余码的计算举例

- 现在  $k = 6$ ,  $M = 101001$ 。
- 设  $n = 3$ , 除数  $P = 1101$ ,
- 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。





# 循环冗余检验的原理说明

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

$001 \leftarrow R \text{ (余数)}, \text{ 作为 FCS}$





# 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)。
- 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。



## 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  $P$ ，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)。
- “无差错接受”是指：“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”。
- 也就是说：“凡是接收端数据链路层接受的帧都没有传输差错”（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）就必须再加上**确认**和**重传**机制。

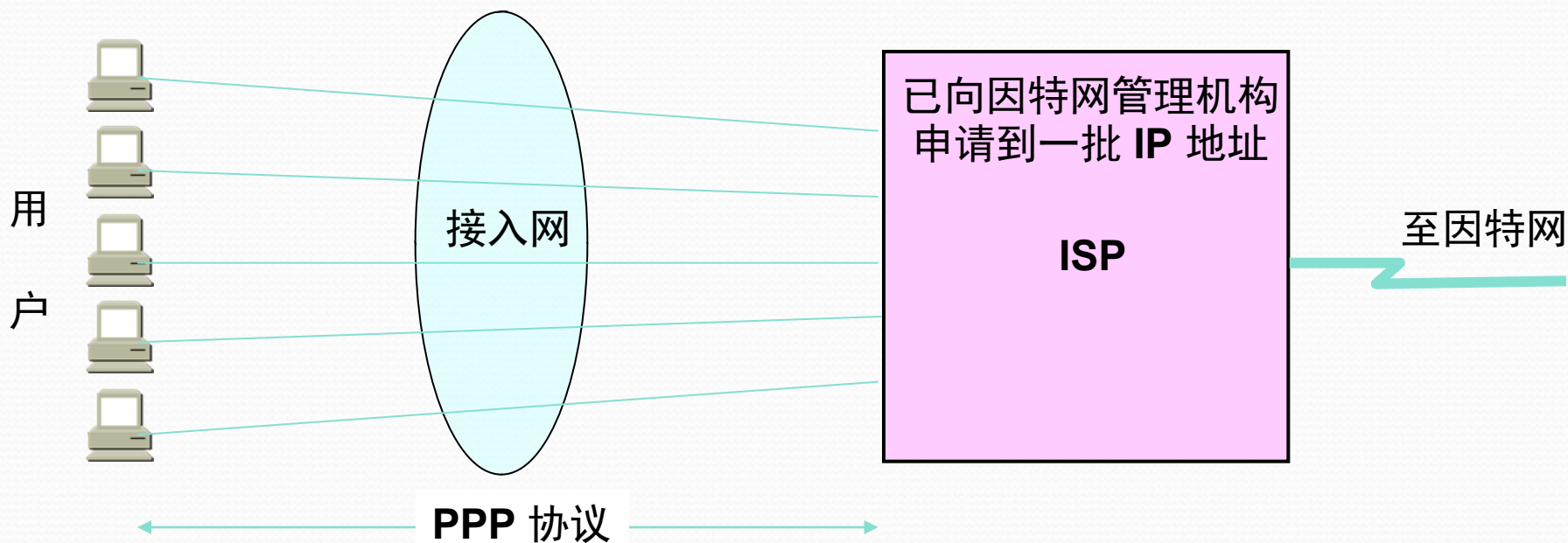




# 点对点协议 PPP

- 现在全世界使用得最多的数据链路层协议是**点对点协议** PPP (Point-to-Point Protocol)。
- 用户使用拨号电话线接入因特网时，一般都是使用 PPP 协议。

## 用户到 ISP 的链路使用 PPP 协议





# PPP 协议应满足的需求

- 简单——这是首要的要求
- 封装成帧
- 透明性
- 多种网络层协议
- 多种类型链路 (PPPoE)
- 差错检测
- 检测连接状态
- 最大传送单元
- 网络层地址协商
- 数据压缩协商





# PPP 协议不需要的功能

- 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

- 1992 年制订了 PPP 协议。经过 1993 年和 1994 年的修订，现在的 PPP 协议已成为因特网的正式标准[RFC 1661]。
- PPP 协议有三个组成部分
  - 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)。
  - 网络控制协议 NCP (Network Control Protocol)。



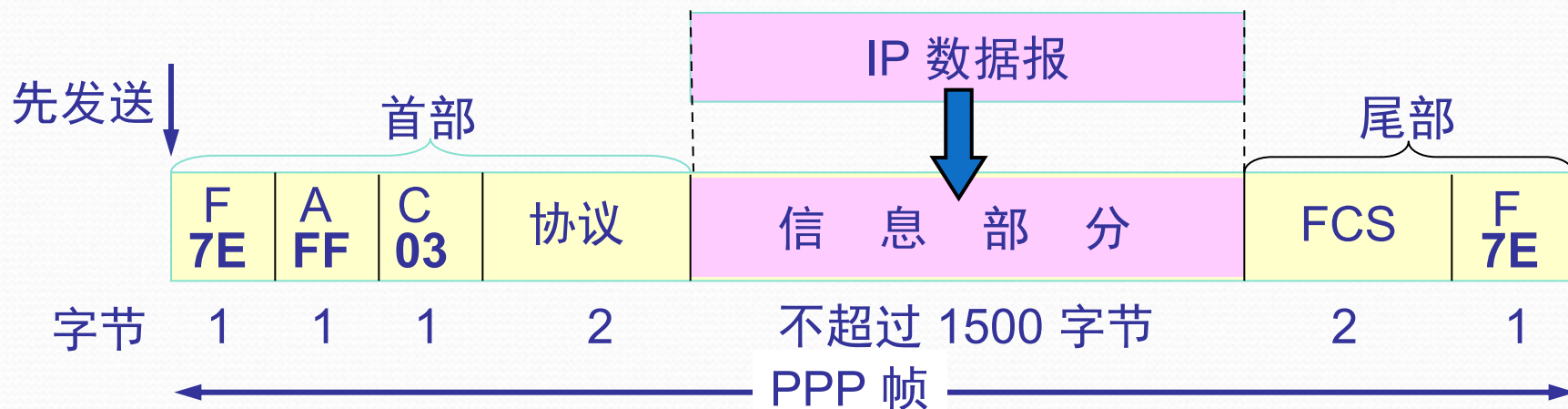


# PPP 协议的帧格式

- 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。



# 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。



# 透明传输问题

- 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

- 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，是使用同步传输（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

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

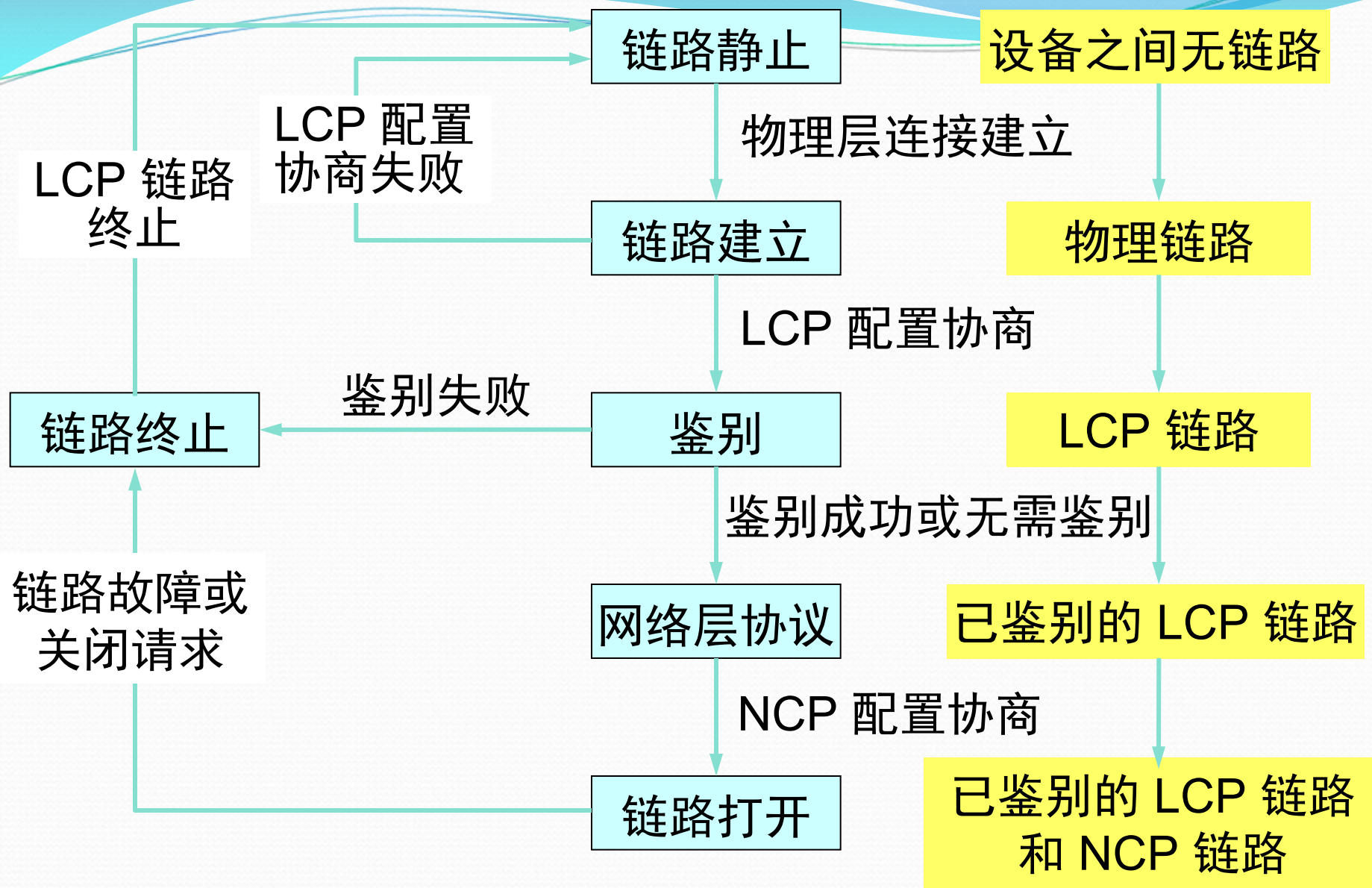
- PPP 协议之所以不使用序号和确认机制是出于以下的考虑：
  - 在数据链路层出现差错的概率不大时，使用比较简单的 PPP 协议较为合理。
  - 在因特网环境下，PPP 的信息字段放入的数据是 IP 数据报。数据链路层的可靠传输并不能够保证网络层的传输也是可靠的。
  - 帧检验序列 FCS 字段可保证无差错接受。





# PPP 协议的工作状态

- 当用户拨号接入 ISP 时，路由器的调制解调器对拨号做出确认，并建立一条物理连接。
- PC 机向路由器发送一系列的 LCP 分组（封装成多个 PPP 帧）。
- 这些分组及其响应选择一些 PPP 参数，和进行网络层配置，NCP 给新接入的 PC 机分配一个临时的 IP 地址，使 PC 机成为因特网上的一个主机。
- 通信完毕时，NCP 释放网络层连接，收回原来分配出去的 IP 地址。接着，LCP 释放数据链路层连接。最后释放的是物理层的连接。



# 本讲总结

- ◆ 数据链路和帧
- ◆ 数据链路层三个基本问题
- ◆ 点对点协议 PPP





# 作业

- 3-07、3-10