



# 第 3 章 数据链路层

---



# 数据链路层

---

- 3.1 使用点对点信道的数据链路层
- 3.2 点对点协议 PPP
- 3.3 使用广播信道的数据链路层
- 3.4 使用广播信道的以太网
- 3.5 扩展的以太网
- 3.6 高速以太网



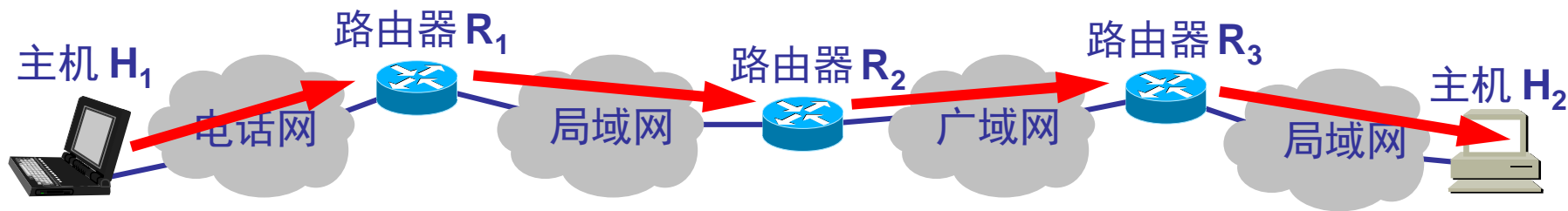
# 数据链路层

数据链路层使用的信道主要有以下两种类型：

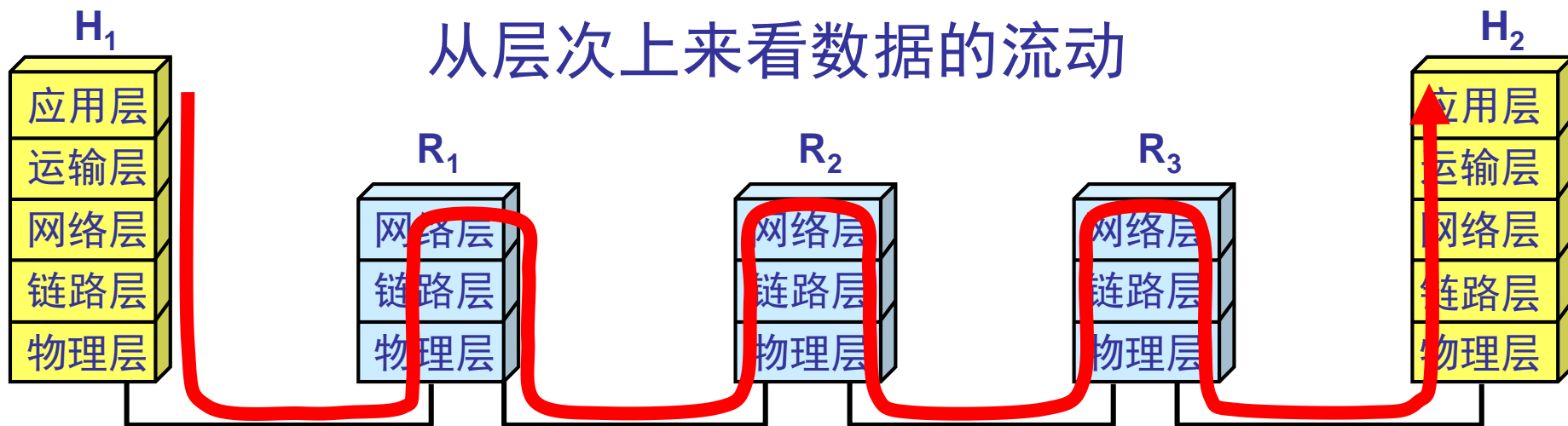
- **点对点信道**。这种信道使用一对一的点对点通信方式。
- **广播信道**。这种信道使用一对多的广播通信方式，因此过程比较复杂。广播信道上连接的主机很多，因此必须使用专用的共享信道协议来协调这些主机的数据发送。

# 数据链路层的简单模型

主机  $H_1$  向  $H_2$  发送数据

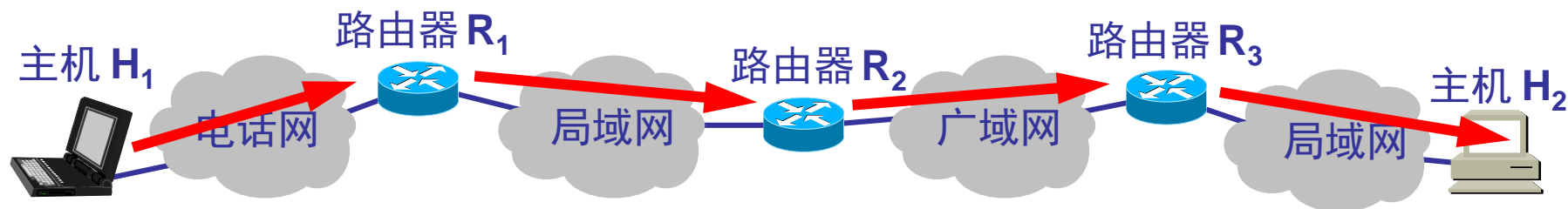


从层次上来看数据的流动

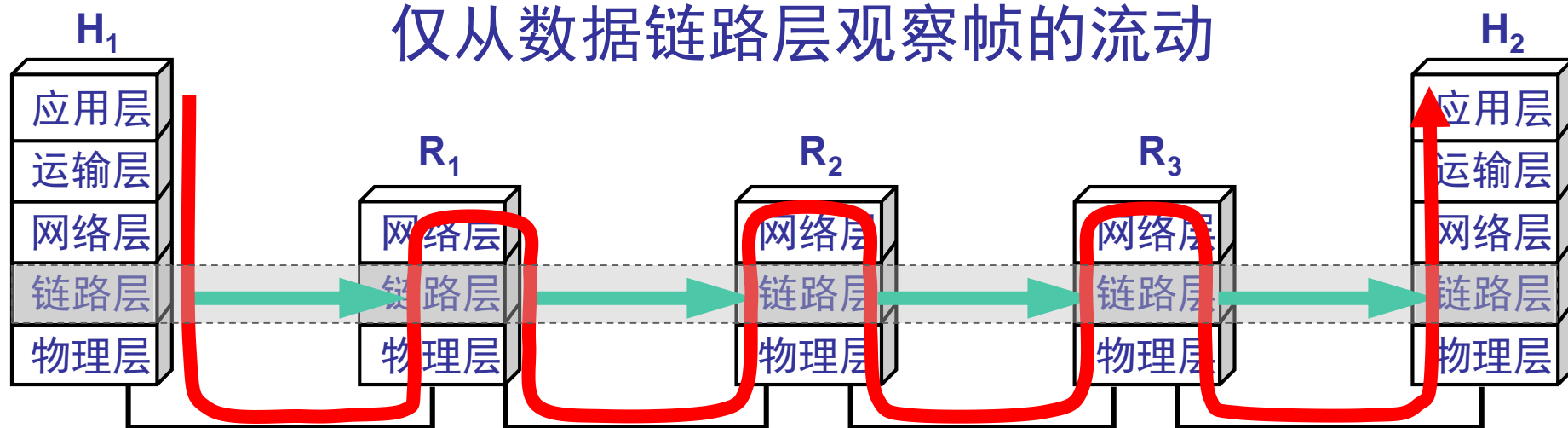


# 数据链路层的简单模型（续）

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)是从一个结点到相邻结点的一段物理线（有线或无线），而中间没有任何其他的交换结点。
  - 一条链路只是一条路径的一个组成部分。

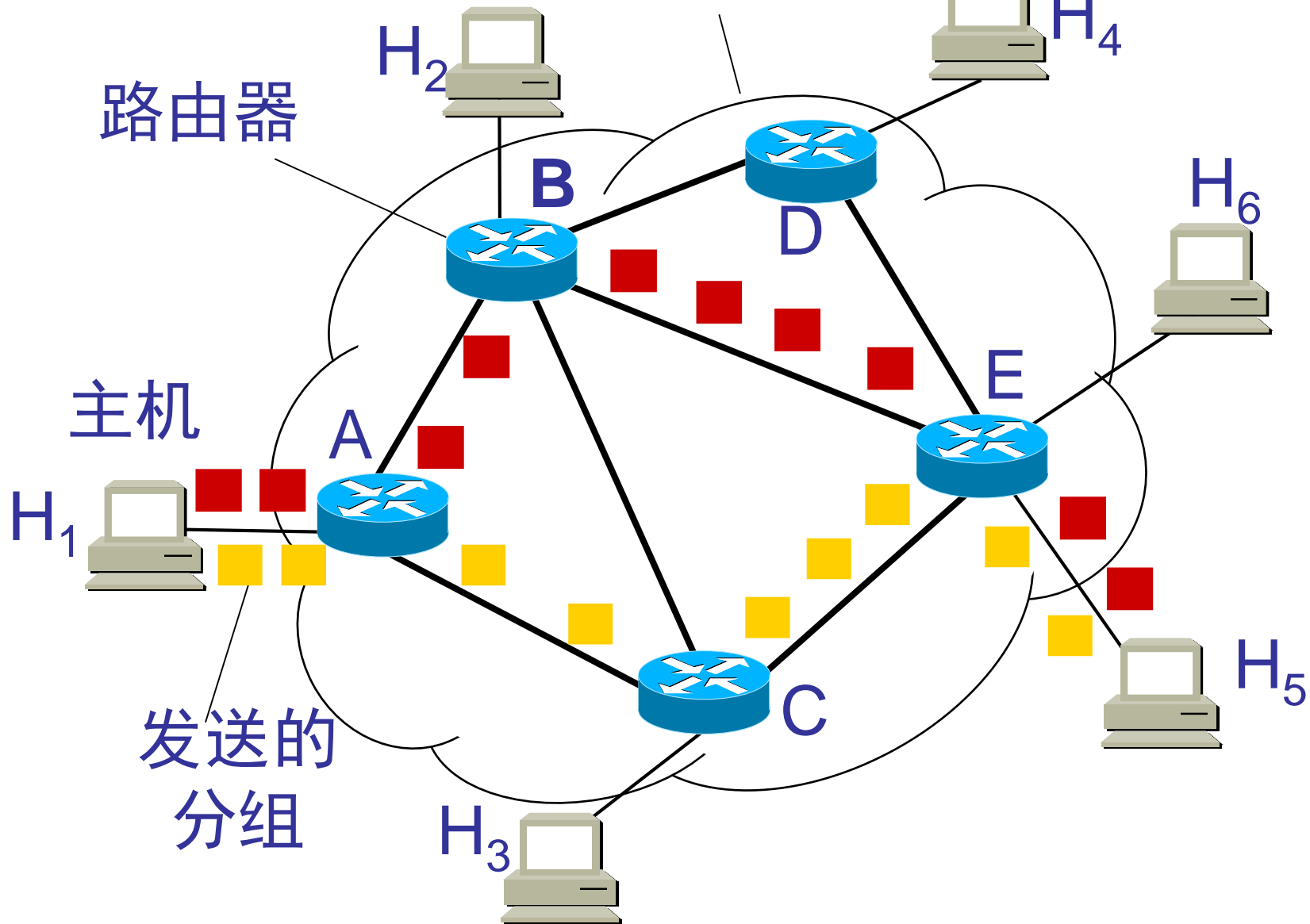


# 数据链路和帧

---

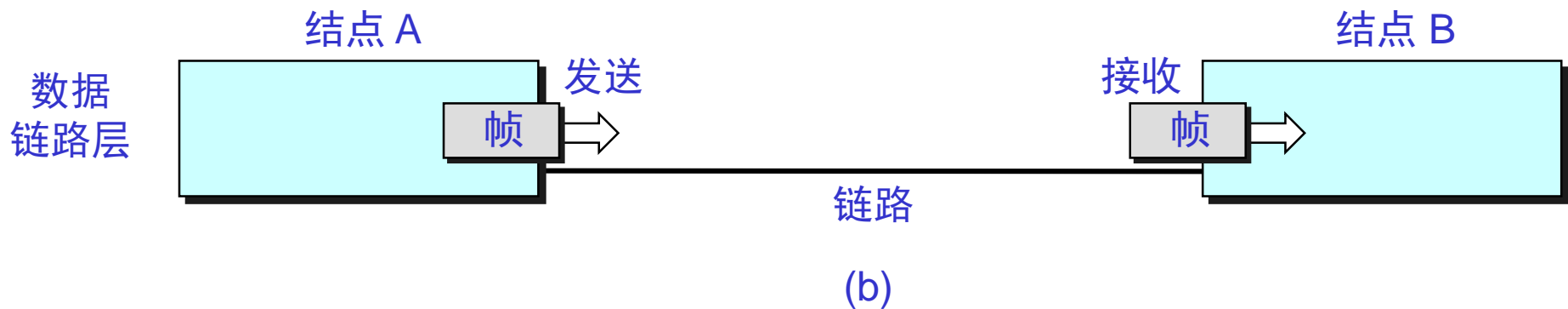
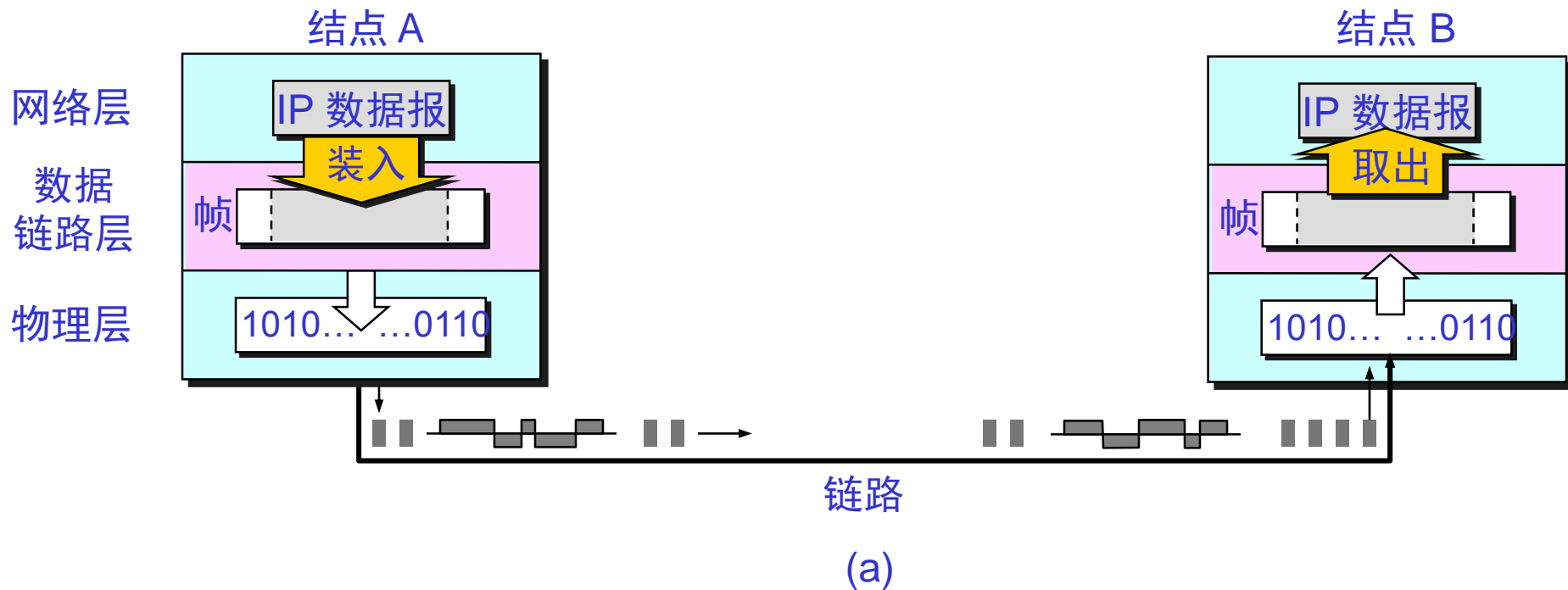
- **数据链路**(data link) 除了物理线路外，还必须有通信协议来控制这些数据的传输。若把实现这些协议的硬件和软件加到链路上，就构成了数据链路。

# 网络核心部分





# 数据链路层传送的是帧





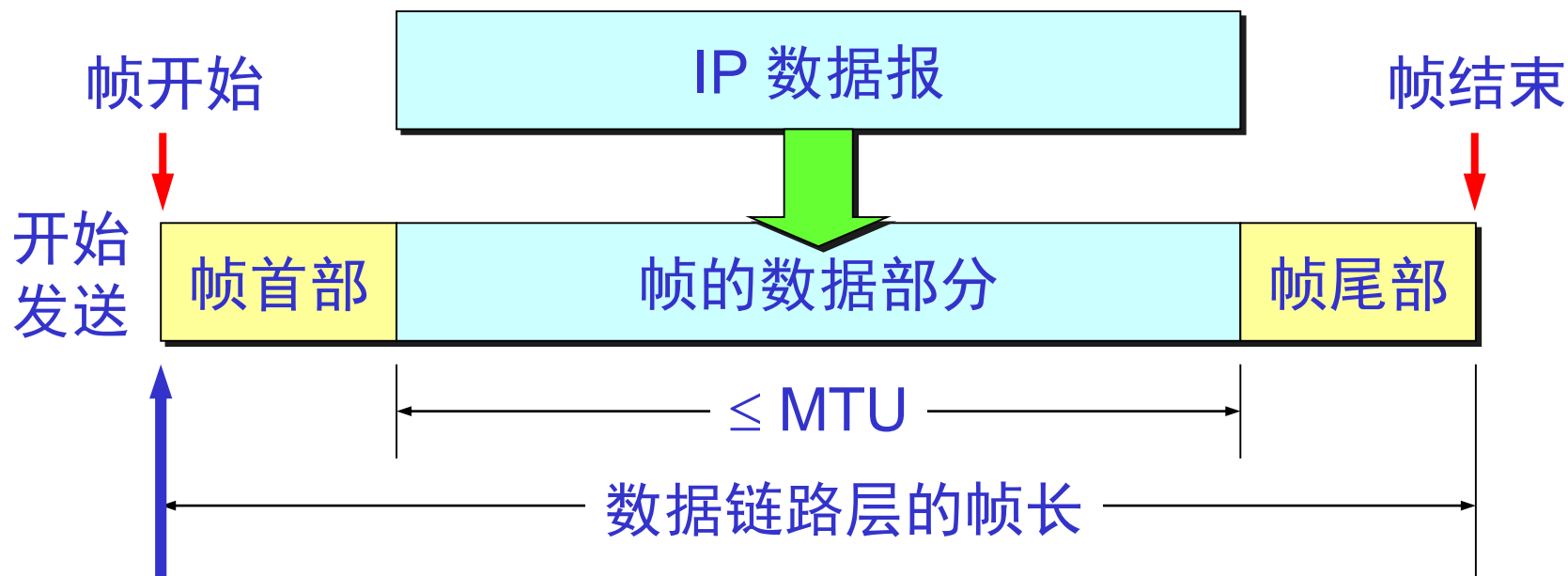
## 3.1.2 三个基本问题

---

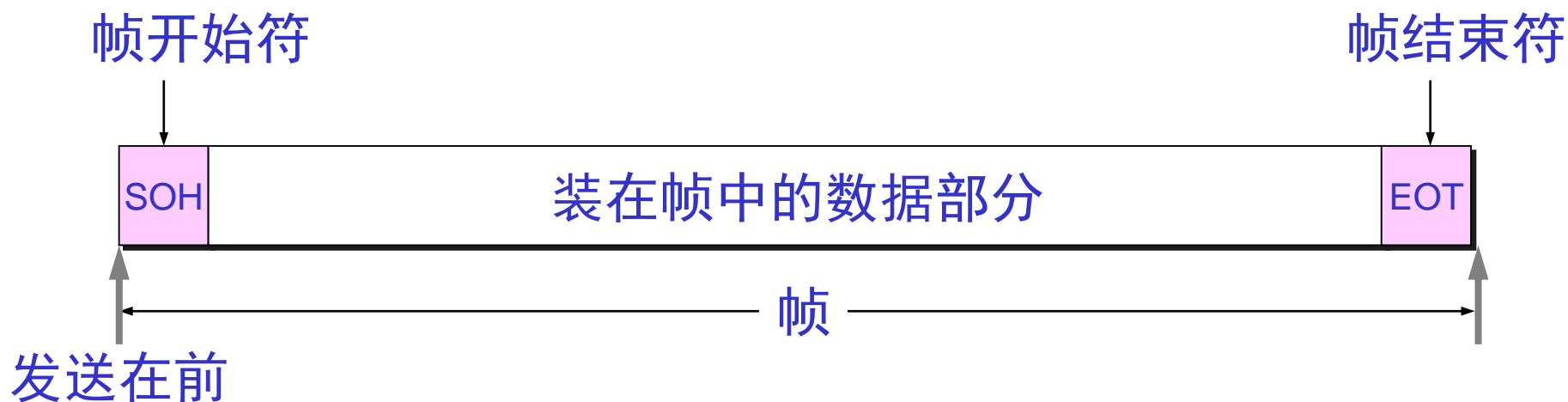
- (1) 封装成帧
- (2) 透明传输
- (3) 差错控制

# 1. 封装成帧

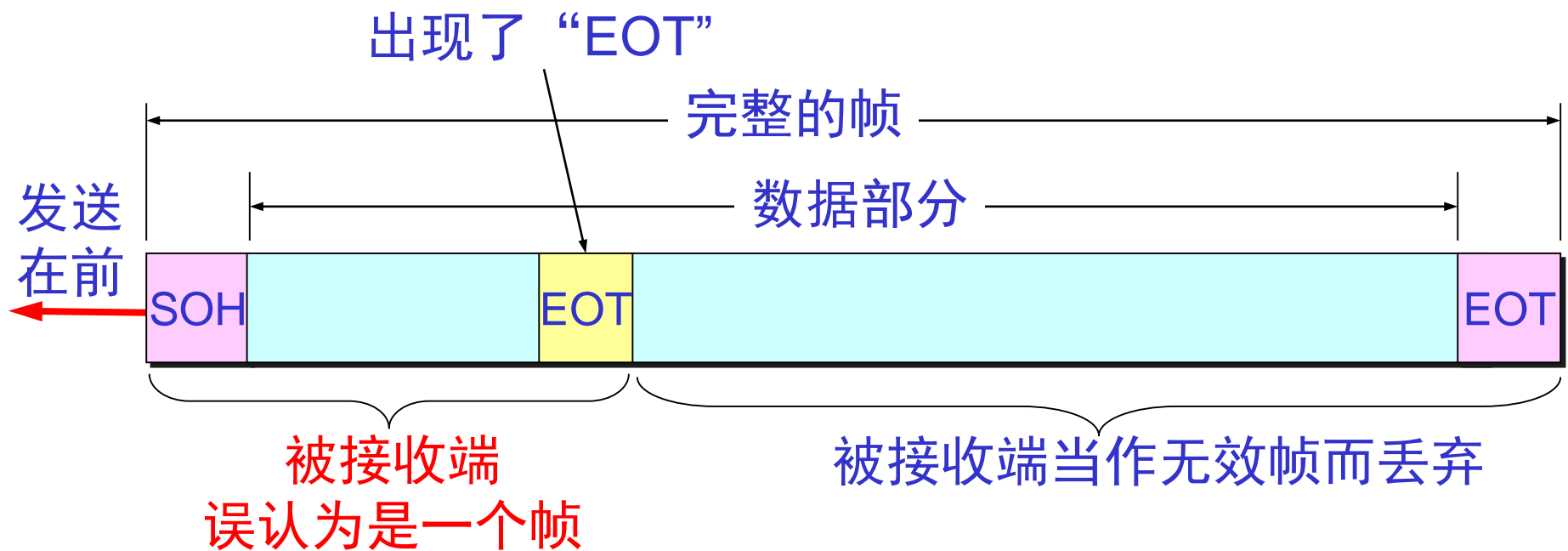
- 封装成帧(framing)就是在一段数据的前后分别添加首部和尾部，然后就构成了一个帧。确定帧的界限。
- 首部和尾部的一个重要作用就是进行帧定界。



# 用控制字符进行帧定界的方法举例



## 2. 透明传输

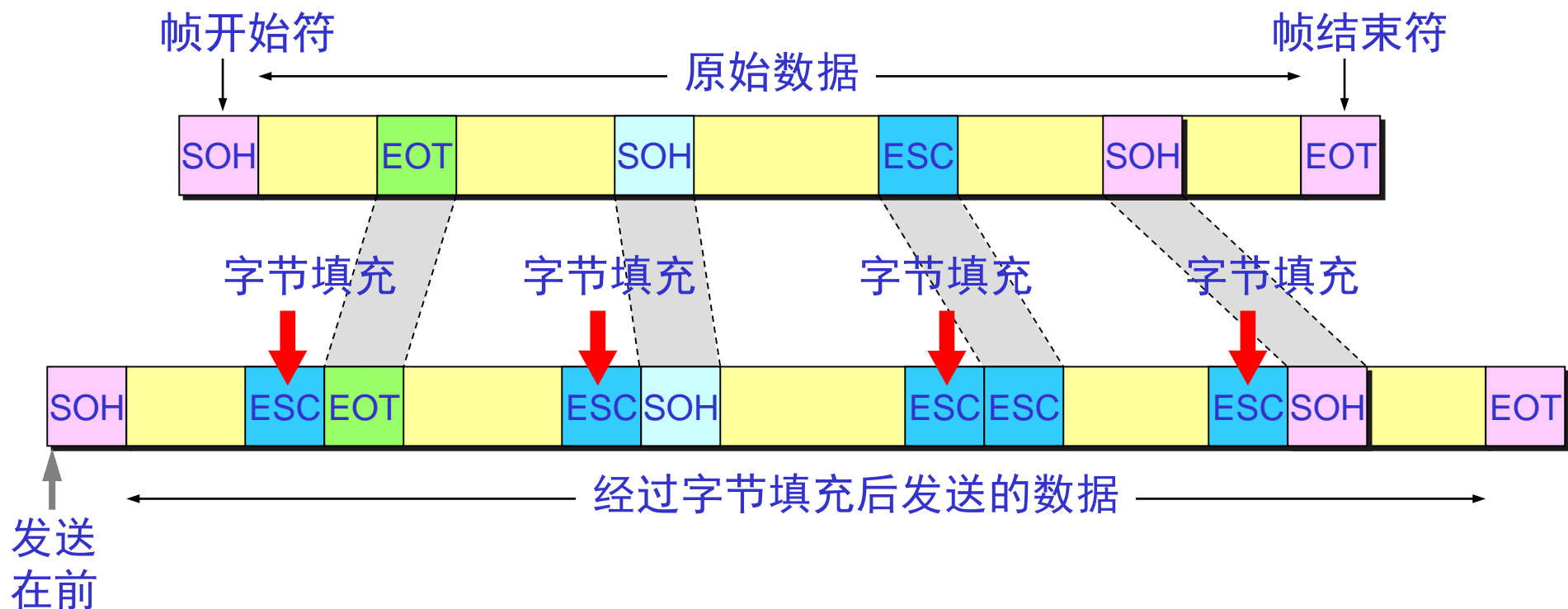




# 解决透明传输问题

- 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)。被称为**字节填充**(byte stuffing)或**字符填充**(character stuffing)。
- 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



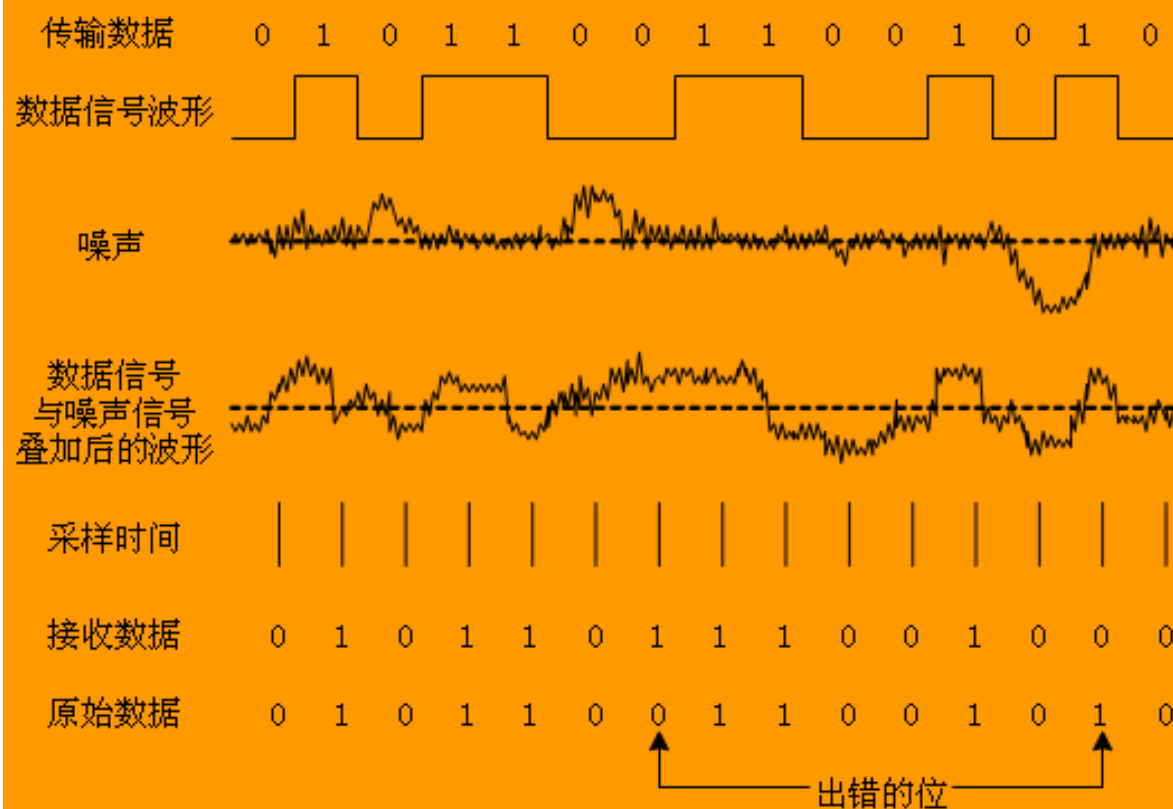
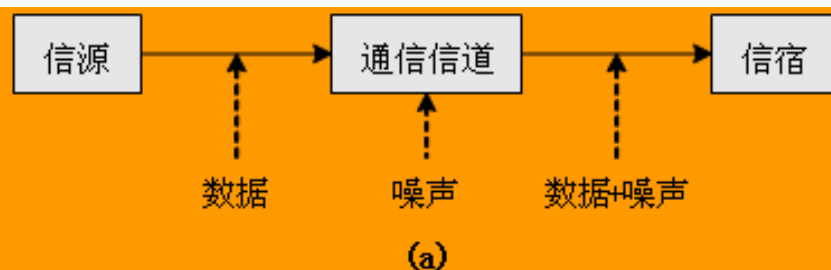


### 3. 差错检测

---

- **比特差错**:比特在传输过程中可能出现差错。
- **误码率** BER (Bit Error Rate):在一段时间内,传输错误的比特占所传输比特总数的比率。
- 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。







# 差错检测

---

- 纠错码：海明码

每个传输的分组带上足够的冗余信息；  
接收端能发现并自动纠正传输差错。

- 检错码：奇偶校验码、循环冗余码（CRC）

分组仅包含足以使接收端发现差错的冗余信息；

接收端能够发现出错，但不能确定哪位是错的，并且自己不能纠正传输差错。



# 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**(称为**帧检验序列 FCS**)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

$$\begin{array}{r} 110101 \leftarrow Q \text{ (商)} \\ P \text{ (除数)} \rightarrow 1101 \overline{) 101001000} \leftarrow 2^n M \text{ (被除数)} \\ \underline{1101} \phantom{0000000} \\ 1110 \phantom{000000} \\ \underline{1101} \phantom{00000} \\ 0111 \phantom{0000} \\ \underline{0000} \phantom{000} \\ 1110 \phantom{00} \\ \underline{1101} \phantom{0} \\ 0110 \\ \underline{0000} \\ 1100 \\ \underline{1101} \\ 001 \leftarrow R \text{ (余数), 作为 FCS} \end{array}$$



# 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  $P$ ，那么出现检测不到的差错的概率就很小很小。



# CRC练习

---

- 传送的数据 $M=110011$
- $P=10011$
- 计算FCS并写出CRC的计算过程。





# 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”。



# 注意

- 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。
- 数据链路层的工作方式：
  - 无连接无确认：误码率很低的信道。
  - 无连接有确认：误码率很高的信道，如无线通信。
  - 有连接有确认：通信要求比较高（可靠性、实时性）的情况。



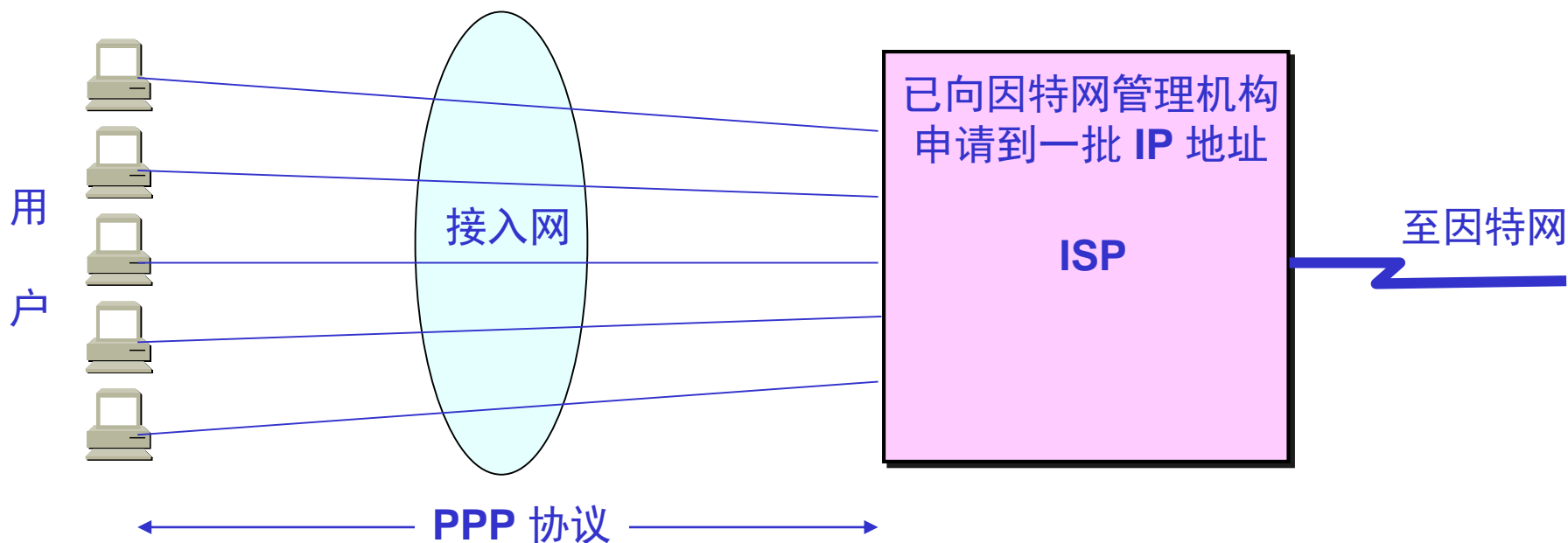
## 3.2 点对点协议 PPP

### 3.2.1 PPP 协议的特点

---

- 链路层协议:HDLC,SLIP
- 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





# 1. 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
- 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



## 2. 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



# 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。



### 3. 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]。



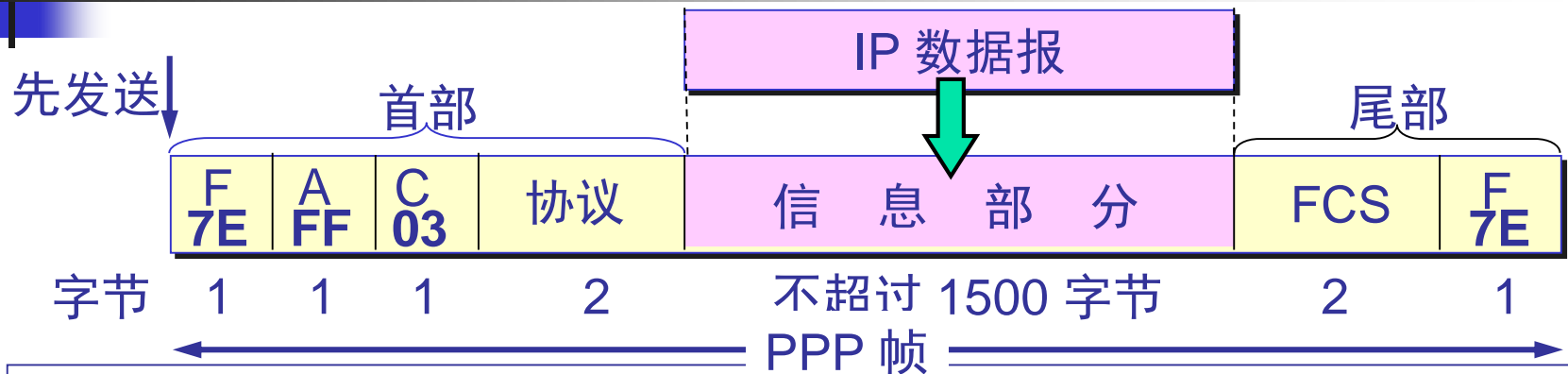


# PPP 协议的组成

---

- 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



- 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, 则表示这是网络层控制数据。



# 透明传输问题

---

- 当 PPP 用在异步传输时，就使用一种特殊的**字符填充法**。
- 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）。



# 字符填充

---

- 将信息字段中出现的每一个 0x7E 字节转变成成为 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



## 3.2.3 PPP 协议的工作状态

---

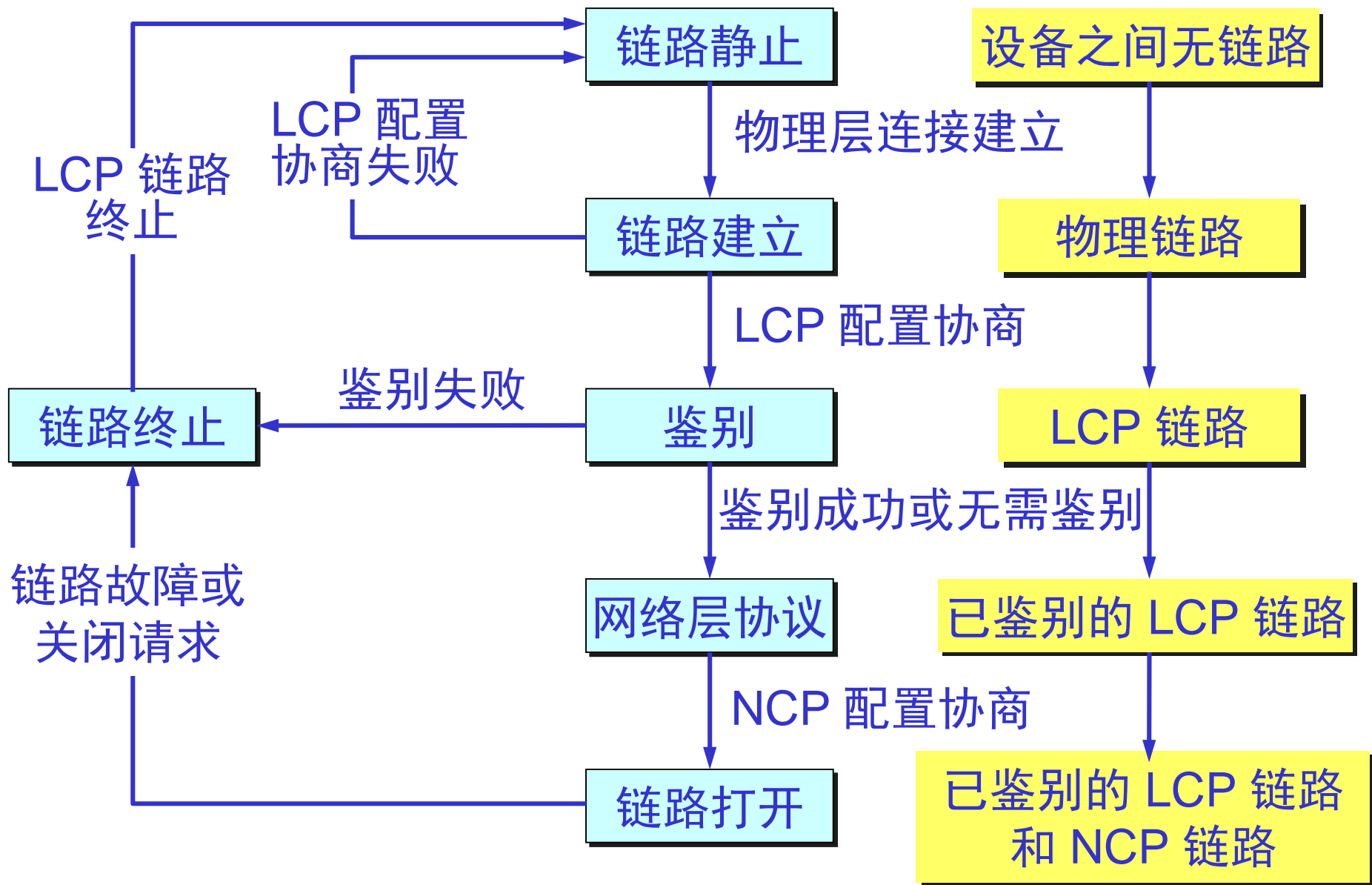
- 当用户拨号接入 ISP 时，路由器的调制解调器对拨号做出确认，并建立一条物理连接。
- PC 机向路由器发送一系列的 LCP 分组（封装成多个 PPP 帧）。



# PPP 协议的工作状态

- 这些分组及其响应选择一些 PPP 参数
- 进入鉴别状态:PAP,CHAP
- 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。







# 课后作业

---

- 习题:3-03,3-04,3-06, 3-07,3-09, 3-10
- 用抓包工具wireshark分析PPPOE协议,准备PPT.

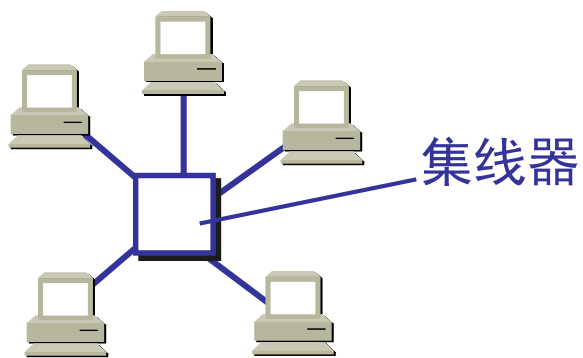
## 3.3 使用广播信道的数据链路层

### 3.3.1 局域网的数据链路层

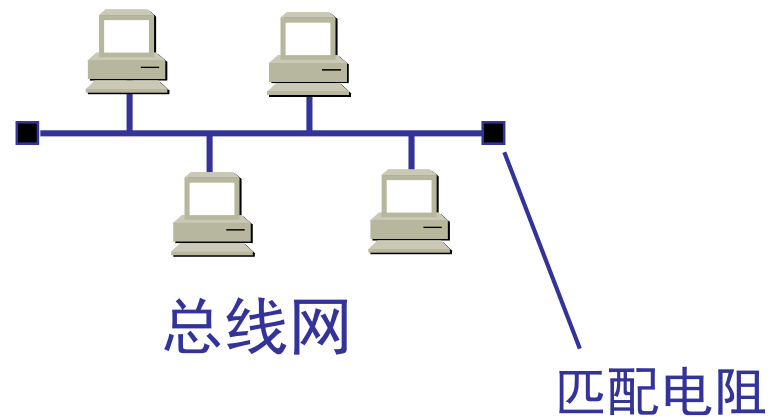
---

- 局域网最主要的特点是：网络为一个单位所拥有，且地理范围和站点数目均有限。
- 局域网使用的就是广播信道。

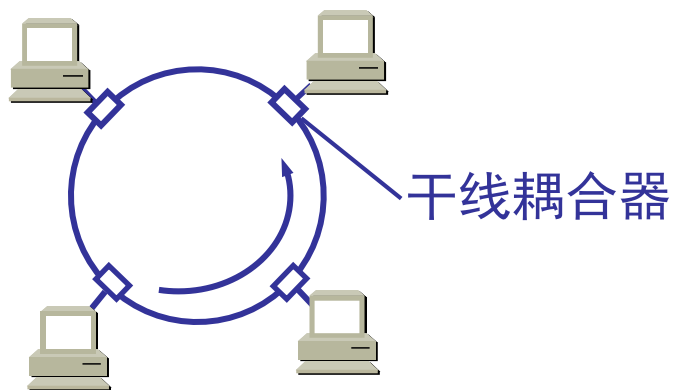
# 局域网的拓扑



星形网



总线网



环形网



# 媒体共享技术

---

- 静态划分信道
  - 频分复用
  - 时分复用
  - 波分复用
  - 码分复用
- 动态媒体接入控制（多点接入）
  - 随机接入
  - 受控接入:令牌局域网或多点线路探询 (polling)，或称轮询。



# 以太网的标准

- DIX Ethernet V2 是世界上第一个局域网产品（以太网）的规约。
- IEEE 的 802.3 标准。
- DIX Ethernet V2 标准与 IEEE 的 802.3 标准只有很小的差别，因此可以将 802.3 局域网简称为“以太网”。
- 严格说来，“以太网”应当是指符合 DIX Ethernet V2 标准的局域网

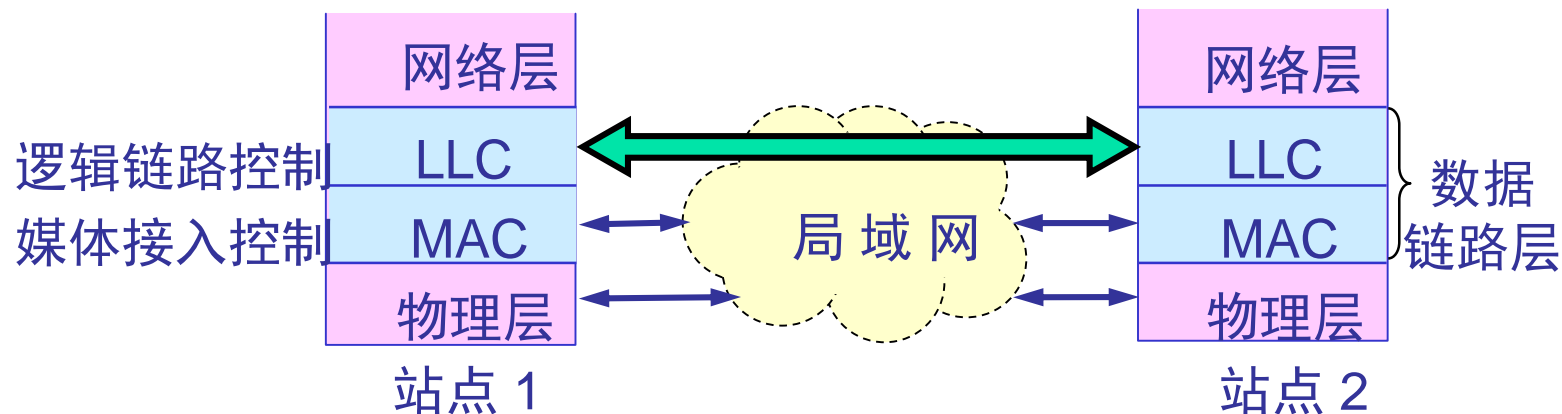


# 数据链路层的两个子层

---

- 为了使数据链路层能更好地适应多种局域网标准，802 委员会就将局域网的数据链路层拆成两个子层：
  - 逻辑链路控制 LLC (Logical Link Control) 子层
  - 媒体接入控制 MAC (Medium Access Control) 子层。

# 局域网对 LLC 子层是透明的



- 与接入到传输媒体有关的内容都放在 MAC子层
- LLC 子层则与传输媒体无关





## 以后一般不考虑 LLC 子层

---

- 由于 TCP/IP 体系经常使用的局域网是 DIX Ethernet V2 而不是 802.3 标准中的几种局域网，因此现在 802 委员会制定的逻辑链路控制子层 LLC（即 802.2 标准）的作用已经不大了。

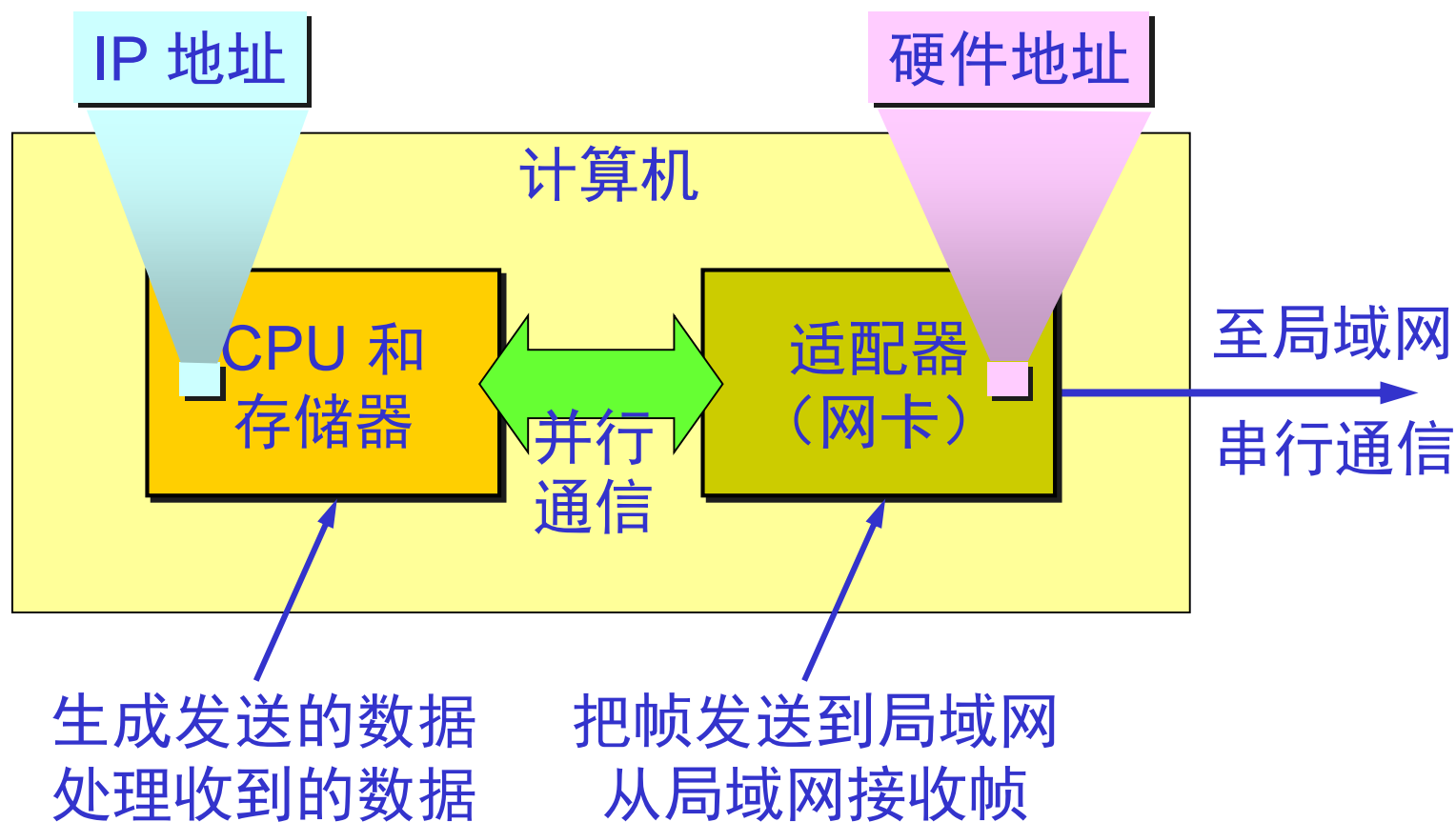


## 2. 适配器的作用

---

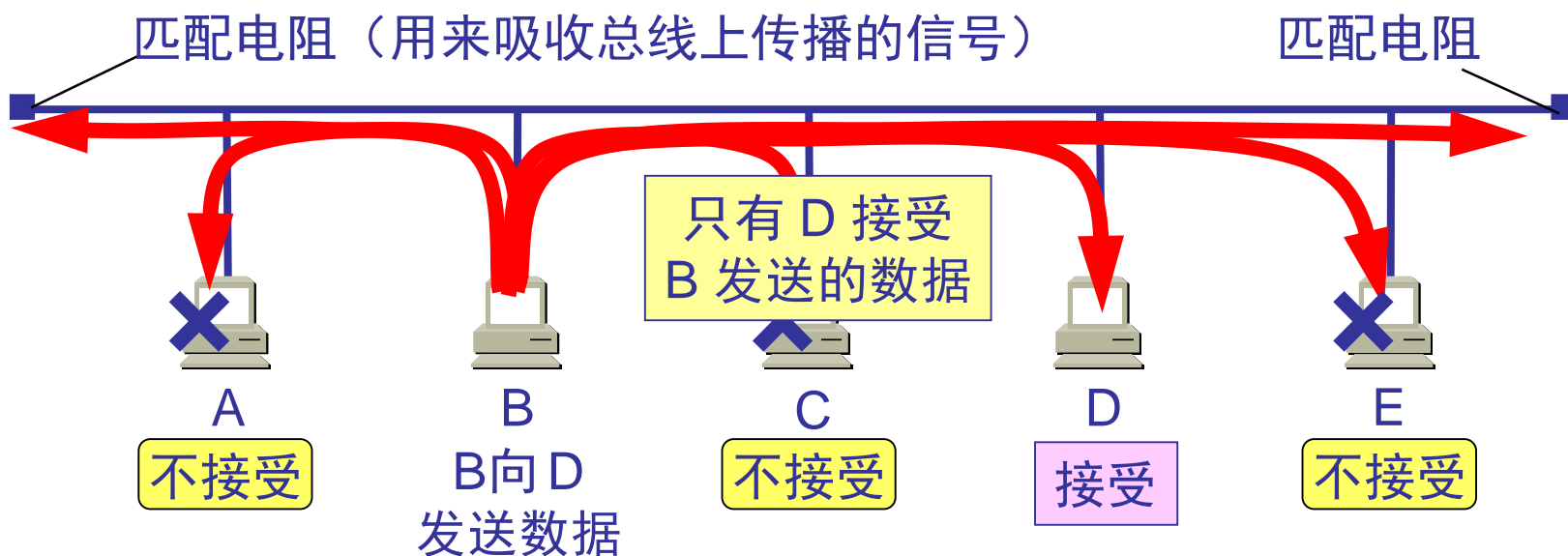
- 网络接口板又称为**通信适配器** (adapter)或**网络接口卡** NIC (Network Interface Card), 或 “**网卡**”。
- 适配器的主要功能：
  - 进行串行/并行转换。
  - 对数据进行缓存。
  - 在计算机的操作系统安装设备驱动程序。
  - 实现以太网协议。

# 计算机通过适配器和局域网进行通信



## 3.3.2 CSMA/CD 协议

- 最初的以太网是将许多计算机都连接到一根总线上。当初认为这样的连接方法既简单又可靠，因为总线上没有有源器件。





# 以太网的广播方式发送

---

- 总线上的每一个工作的计算机都能检测到 B 发送的数据信号。
- 由于只有计算机 D 的地址与数据帧首部写入的地址一致，因此只有 D 才接收这个数据帧。
- 其他所有的计算机（A, C 和 E）都检测到不是发送给它们的数据帧，因此就丢弃这个数据帧而不能够收下来。

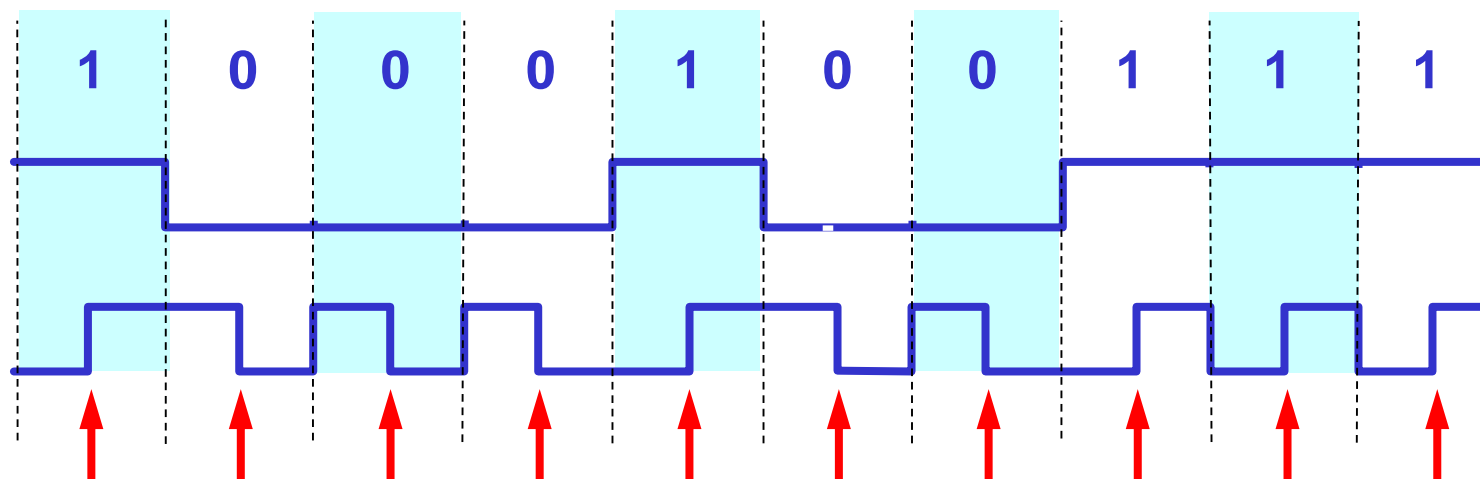


# 为了通信的简便 以太网采取了两种重要的措施

- 以太网采用较为灵活的无连接的工作方式,对发送的数据帧不进行编号,也不要求对方发回确认。
- 当目的站收到有差错的数据帧时就丢弃此帧,其他什么也不做。差错的纠正由高层来决定。
- 如果高层发现丢失了一些数据而进行重传。

# 以太网发送的数据都使用曼彻斯特(Manchester)编码

码元  
基带数字信号  
曼彻斯特编码  
出现电平转换





# 载波监听多点接入/碰撞检测 CSMA/CD

---

- CSMA/CD 表示 Carrier Sense Multiple Access with Collision Detection。
- “**多点接入**”表示许多计算机以多点接入的方式连接在一根总线上。
- “**载波监听**”是指每一个站在发送数据之前先要检测一下总线上是否有其他计算机在发送数据，如果有，则暂时不要发送数据，以免发生碰撞。





# 碰撞检测

---

- “碰撞检测”就是计算机边发送数据边检测信道上的信号电压大小。
- 当几个站同时在总线上发送数据时，总线上的信号电压摆动值将会增大（互相叠加）。
- 当一个站检测到的信号电压摆动值超过一定的门限值时，就认为总线上至少有两个站同时在发送数据，表明产生了碰撞。



## 检测到碰撞后

---

- 在发生碰撞时，总线上传输的信号产生了严重的失真，无法从中恢复出有用的信息来。立即停止发送，免得继续浪费网络资源，然后等待一段随机时间后再次发送。

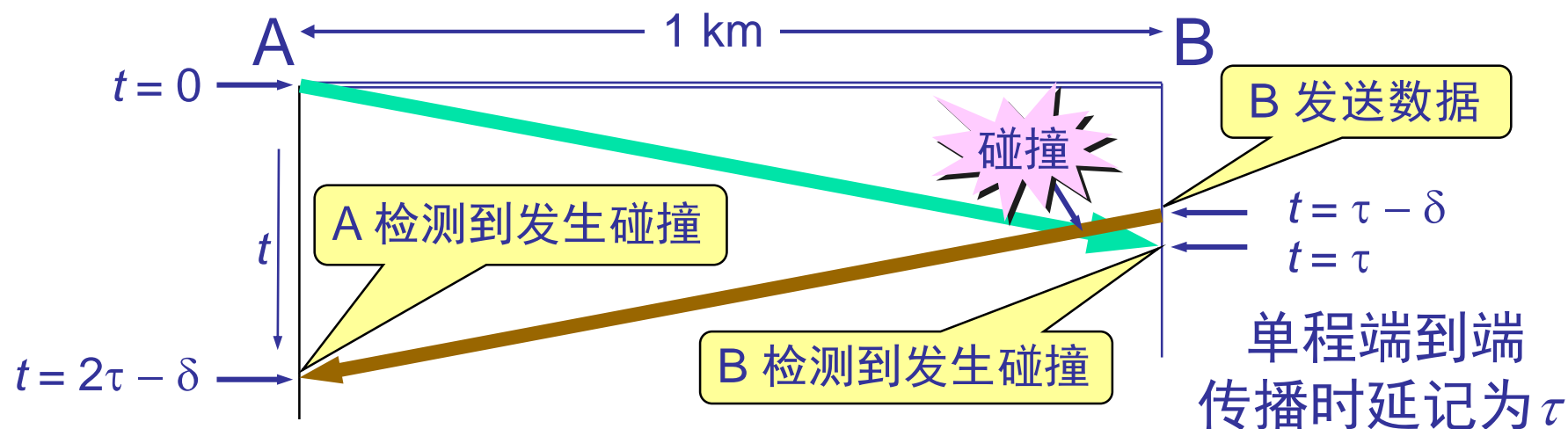


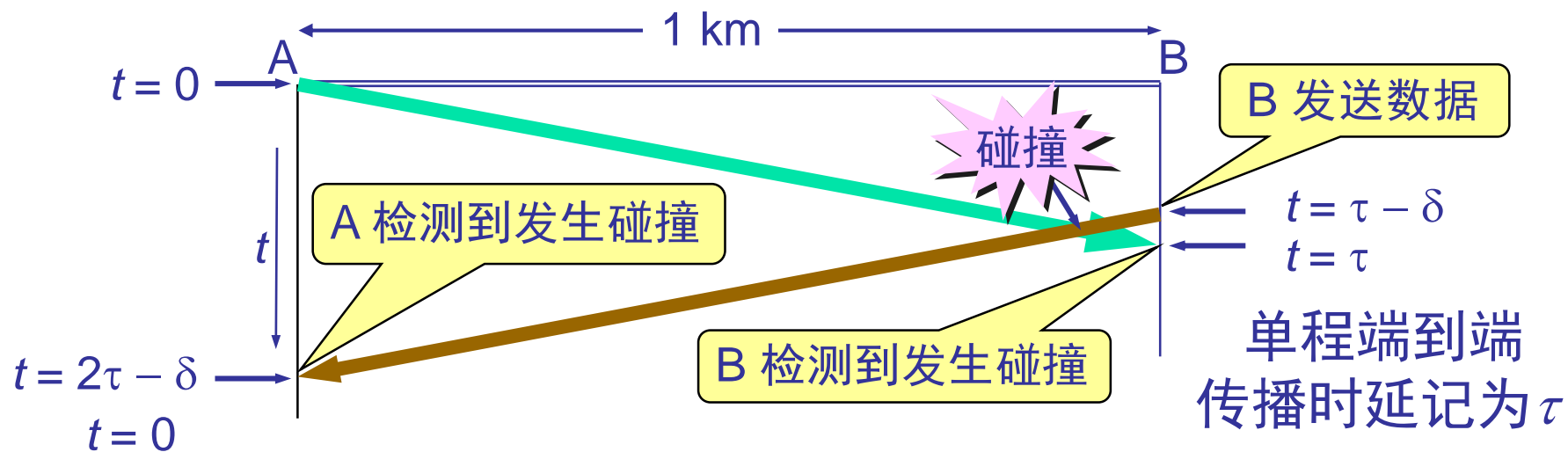
# 电磁波在总线上的 有限传播速率的影响

---

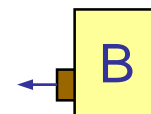
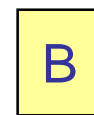
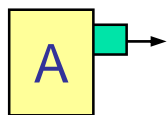
- 当某个站监听到总线是空闲时，也可能总线并非真正是空闲的。
- A 向 B 发出的信息，要经过一定的时间后才能传送到 B。
- B 若在 A 发送的信息到达 B 之前发送自己的帧(因为这时 B 的载波监听检测不到 A 所发送的信息)，则必然要在某个时间和 A 发送的帧发生碰撞。
- 碰撞的结果是两个帧都变得无用。

# 传播时延对载波监听的影晌

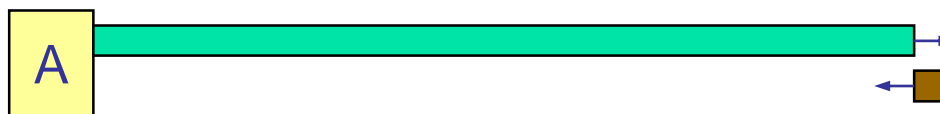




A 检测到  
信道空闲  
发送数据



$t = \tau - \delta$   
B 检测到信道空闲  
发送数据



$t = \tau - \delta / 2$   
发生碰撞



$t = \tau$   
B 检测到发生碰撞  
停止发送

$t = 2\tau - \delta$   
A 检测到  
发生碰撞





# 重要特性

---

- 使用 CSMA/CD 协议的以太网不能进行全双工通信而只能进行双向交替通信（半双工通信）。
- 每个站在发送数据之后的一小段时间内，存在着遭遇碰撞的可能性。



# 争用期

- 最先发送数据帧的站，在发送数据帧后至多经过时间  $2\tau$ （两倍的端到端往返时延）就可知道发送的数据帧是否遭受了碰撞。  $2\tau$  称为争用期，或碰撞窗口
- 经过争用期这段时间还没有检测到碰撞，才能肯定这次发送不会发生碰撞。



# 截断二进制指数退避算法

(truncated binary exponential type)

- 发生碰撞的站在停止发送数据后，要退避一个随机时间才能再发送数据。
  - 确定基本退避时间，一般是取为争用期  $2\tau$ 。
  - 定义重传次数  $k$ ， $k \leq 10$ ，即
$$k = \text{Min}[\text{重传次数}, 10]$$
  - 从整数集合  $[0, 1, \dots, (2^k - 1)]$  中随机地取出一个数，记为  $r$ 。重传所需的时延就是  $r$  倍的基本退避时间。
  - 当重传达 16 次仍不能成功时即丢弃该帧，并向高层报告。





# 争用期的长度

---

- 以太网取  $51.2\ \mu\text{s}$  为争用期的长度。
- 对于  $10\ \text{Mb/s}$  以太网，在争用期内可发送  $512\ \text{bit}$ ，即  $64$  字节。
- 以太网在发送数据时，若前  $64$  字节没有发生冲突，则后续的数据就不会发生冲突。
- 以太网规定了最短有效帧长为  $64$  字节，凡长度小于  $64$  字节的帧都是由于冲突而异常中止的**无效帧**。

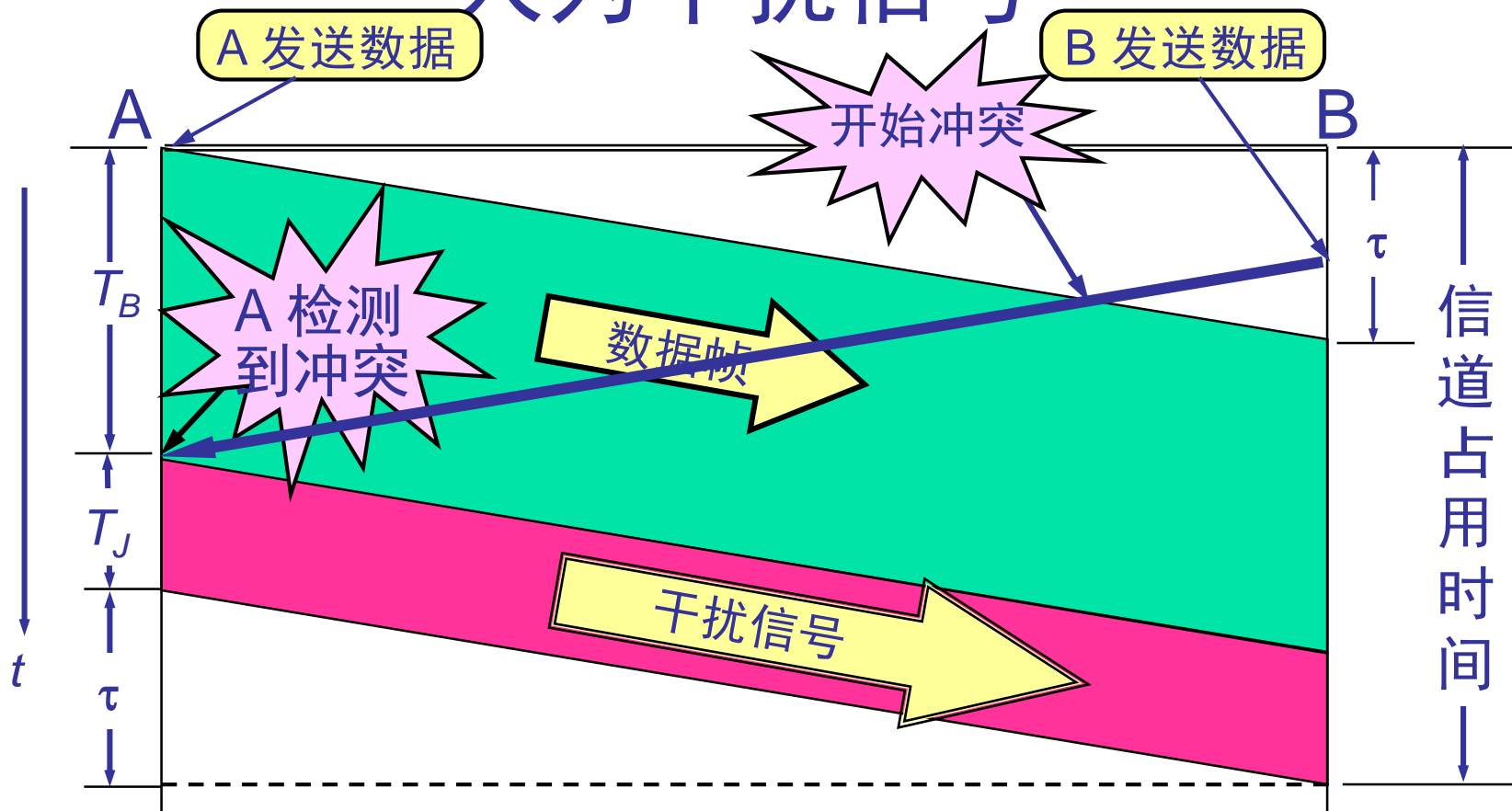


# 强化碰撞

---

- 当发送数据的站一旦发现发生了碰撞时：
  - 立即停止发送数据；
  - 再继续发送若干比特的人为干扰信号 (jamming signal)，以便让所有用户都知道现在已经发生了碰撞。

# 人为干扰信号



B 也能够检测到冲突，并立即停止发送数据帧，接着就发送干扰信号。这里为了简单起见，只画出 A 发送干扰信号的情况。



# 帧间最小间隔

---

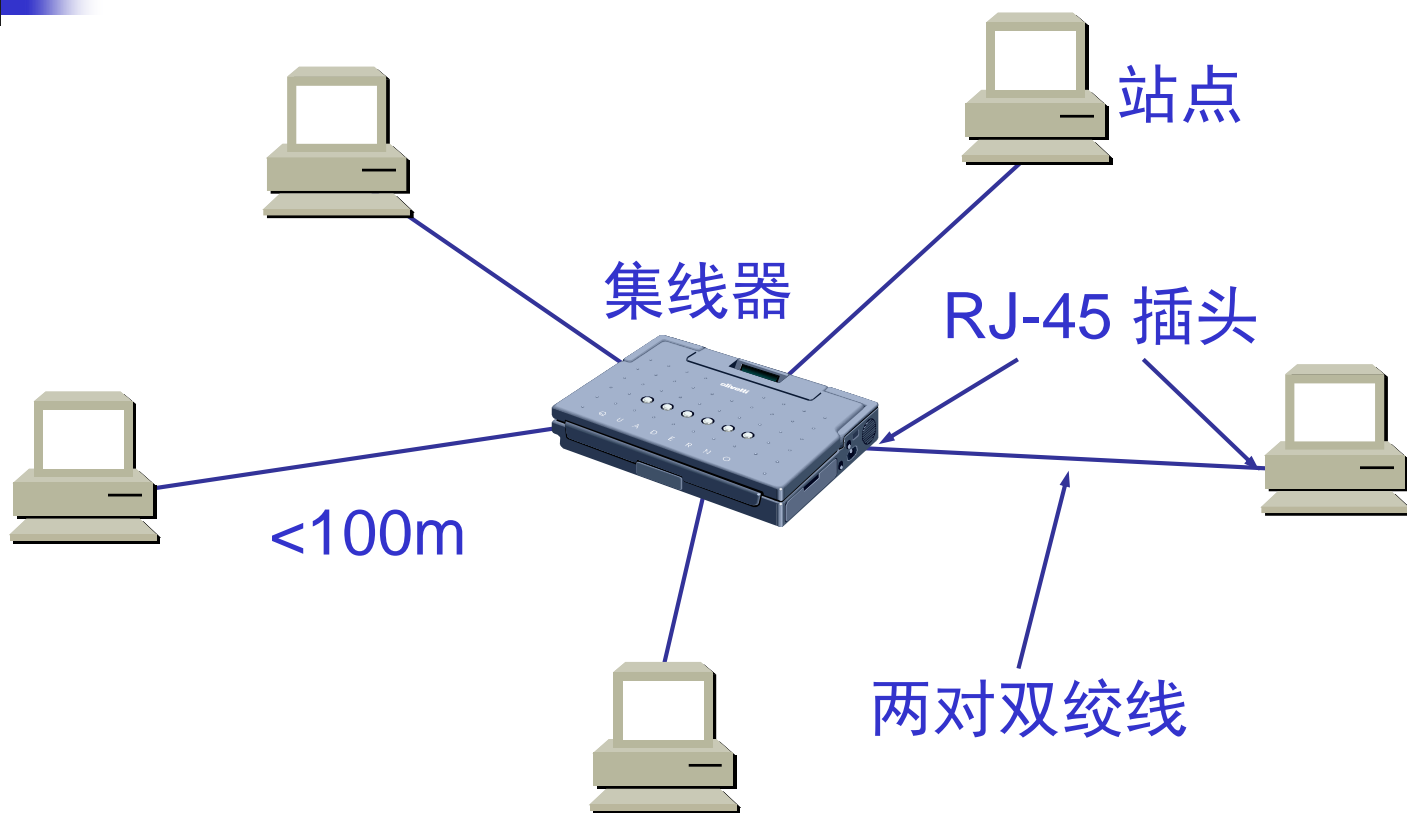
- 帧间最小间隔为  $9.6\ \mu\text{s}$ ，相当于 96 bit 的发送时间。
- 一个站在检测到总线开始空闲后，还要等待  $9.6\ \mu\text{s}$  才能再次发送数据。
- 这样做是为了使刚刚收到数据帧的站的接收缓存来得及清理，做好接收下一帧的准备。



### 3.3.3 使用集线器的星形拓扑

- 传统以太网最初是使用粗同轴电缆，后来演进到使用比较便宜的细同轴电缆，最后发展为使用更便宜和更灵活的双绞线。
- 这种以太网采用星形拓扑，在星形的中心则增加了一种可靠性非常高的设备，叫做**集线器**(hub)

# 使用集线器的双绞线以太网





# 星形网 10BASE-T

---

- 集线器使用了大规模集成电路芯片，因此这样的硬件设备的可靠性已大大提高了。
- 10BASE-T 双绞线以太网的出现，是局域网发展史上的一个非常重要的里程碑，它为以太网在局域网中的统治地位奠定了牢固的基础。



# 集线器的一些特点

---

- 集线器是使用电子器件来模拟实际电缆线的工作，因此整个系统仍然像一个传统的以太网那样运行，物理上是星型。
- 使用集线器的以太网在逻辑上仍是一个总线网，各工作站使用的还是 *CSMA/CD* 协议，并共享逻辑上的总线。
- 集线器工作在物理层。



## 3.3.5 以太网的 MAC 层

### 1. MAC 层的硬件地址

---

- 在局域网中，硬件地址又称为物理地址，或 MAC 地址。
- IEEE 的注册管理机构 RA 是全球 MAC 地址的管理机构。



## 48 位的 MAC 地址

- 生产适配器的厂家向IEEE购买6字节的前3个字节(高24位),称为**组织唯一标识符OUI**,又称为**公司标识符**。
- 地址字段中的后三个字节(即低位 24 位)由厂家自行指派,称为**扩展标识符**,必须保证生产出的适配器没有重复地址。



# 适配器检查 MAC 地址

---

- “发往本站的帧” 包括以下三种帧：
  - 单播(unicast)帧（一对一）
  - 广播(broadcast)帧（一对全体）
  - 多播(multicast)帧（一对多）
- 适配器可以工作在一种特殊的混杂方式(promiscuous mode)。

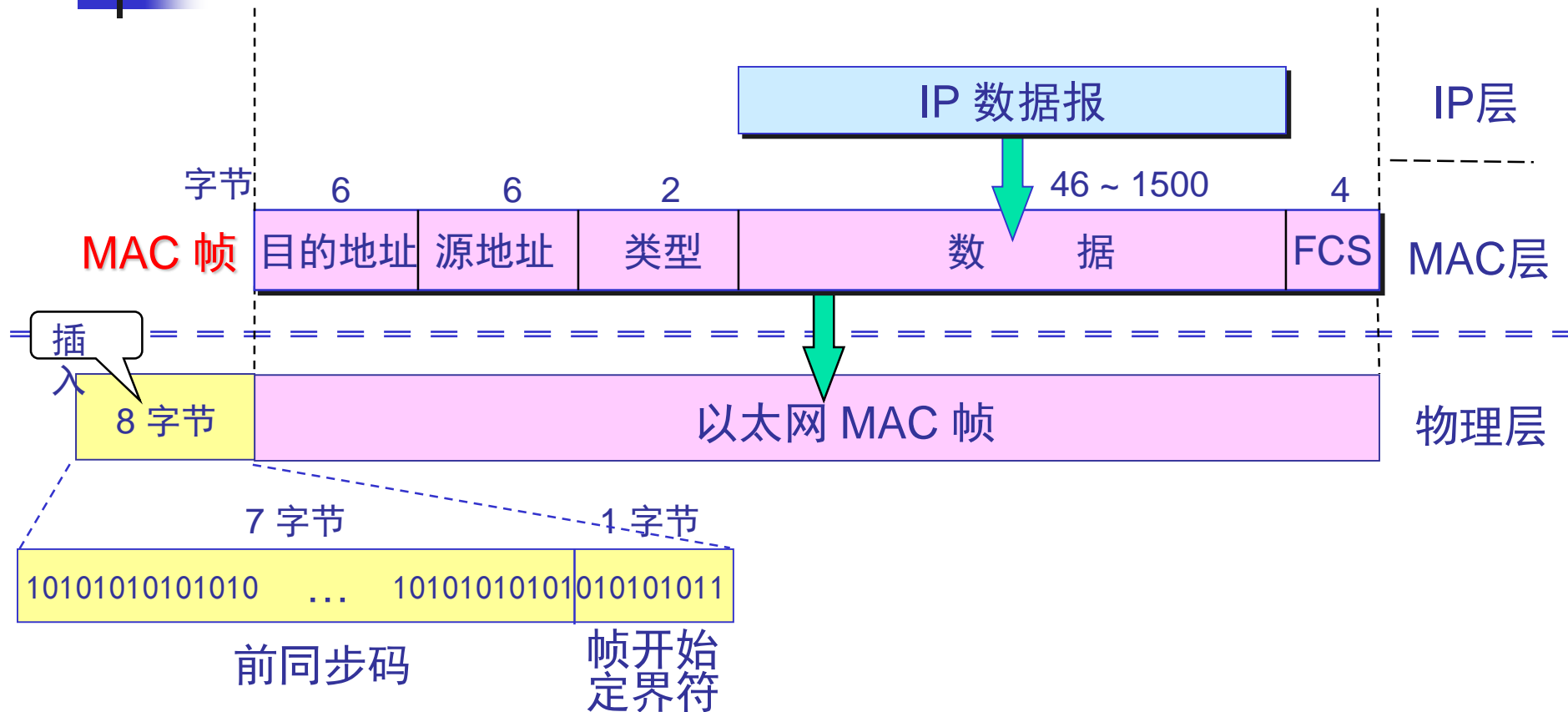


## 2. MAC 帧的格式

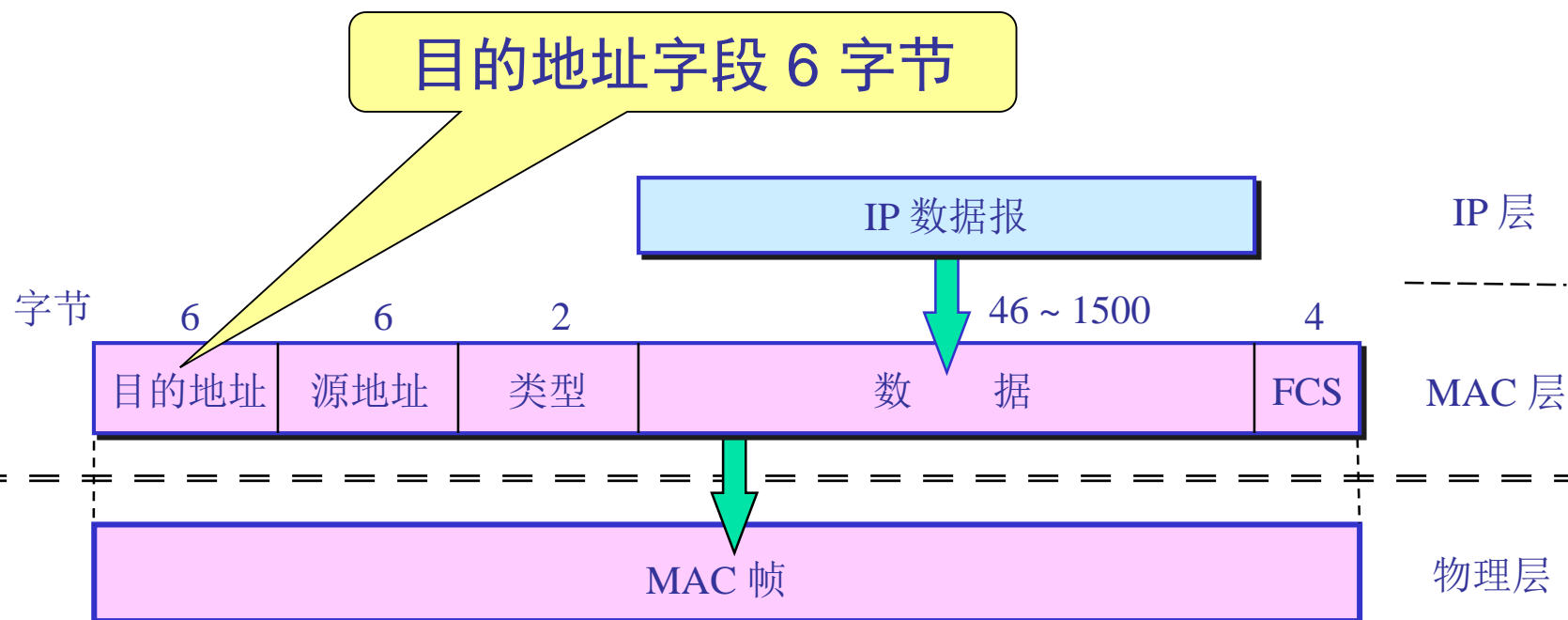
---

- 常用的以太网MAC帧格式有两种标准：
  - DIX Ethernet V2 标准
  - IEEE 的 802.3 标准
- 最常用的 MAC 帧是以太网 V2 的格式。

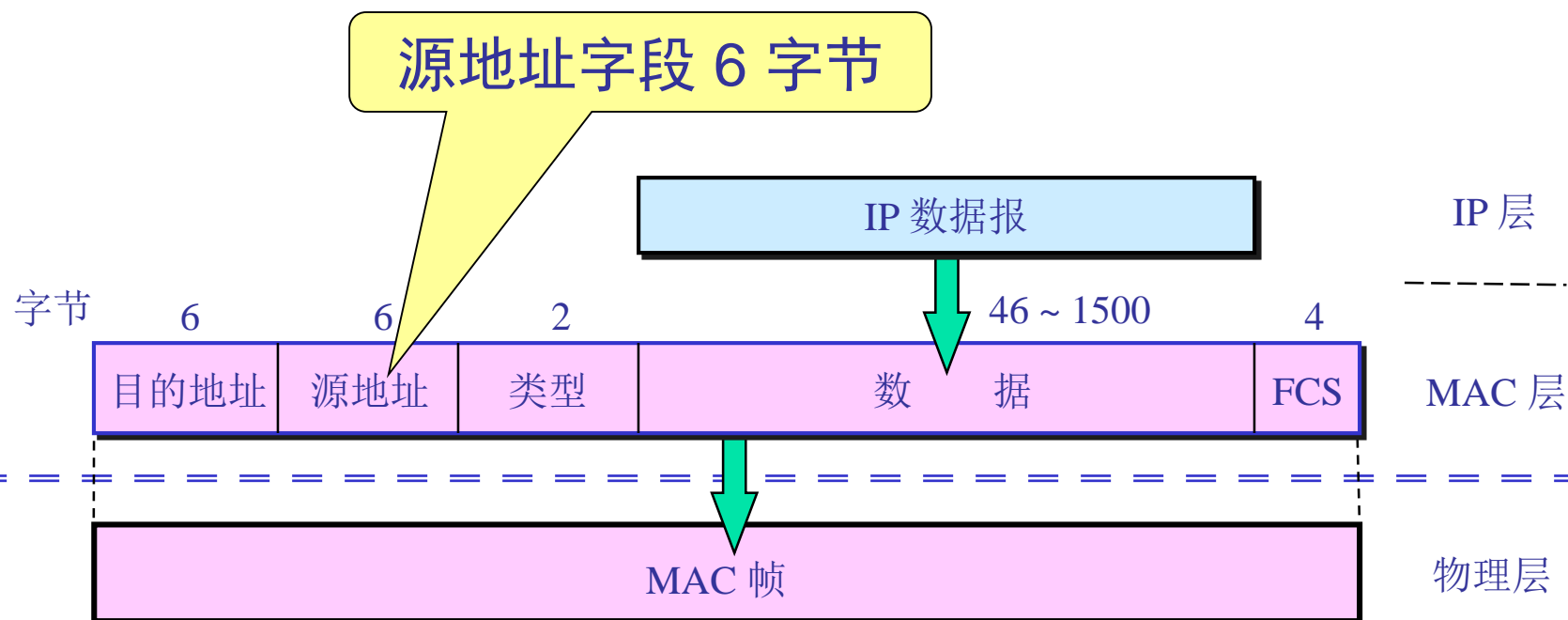
# 以太网V2 的 MAC 帧格式



# 以太网 V2 的 MAC 帧格式

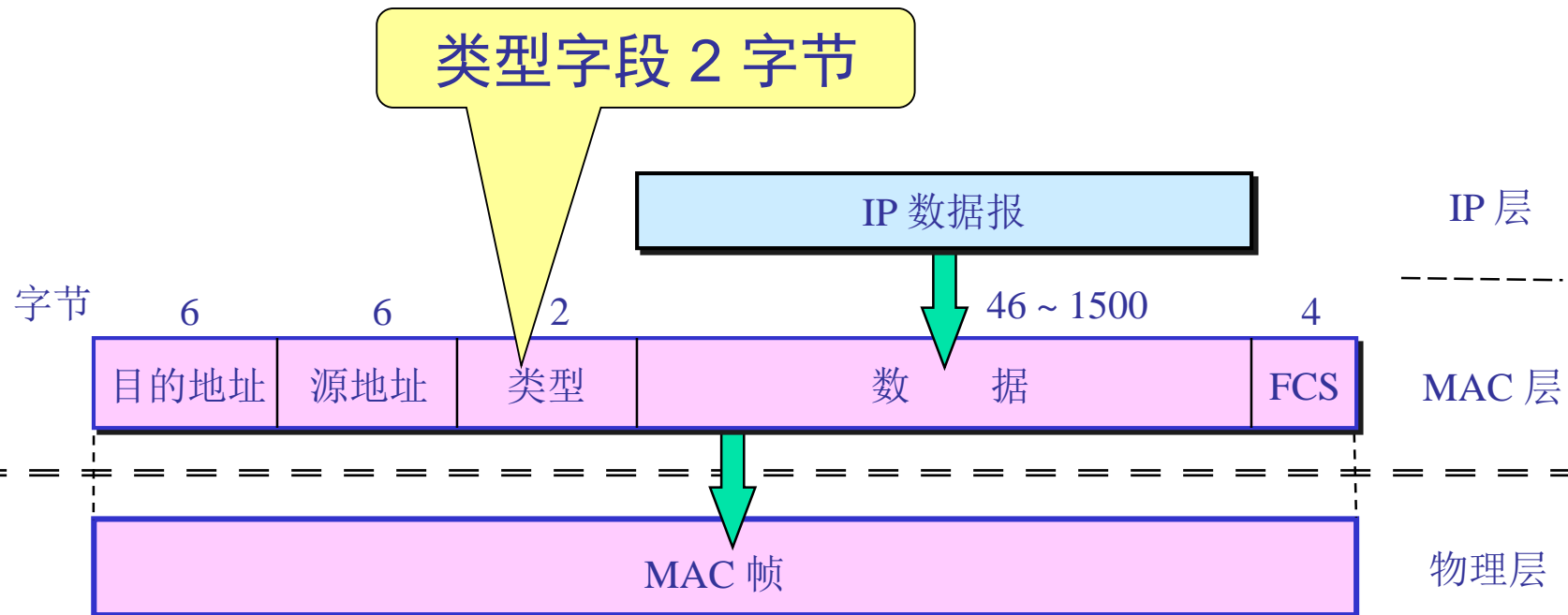


# 以太网 V2 的 MAC 帧格式



# 以太网 V2 的 MAC 帧格式

类型字段用来标志<sup>上</sup>一层使用的是什么协议，以便把收到的 MAC 帧的数据上交给上一层的这个协议。



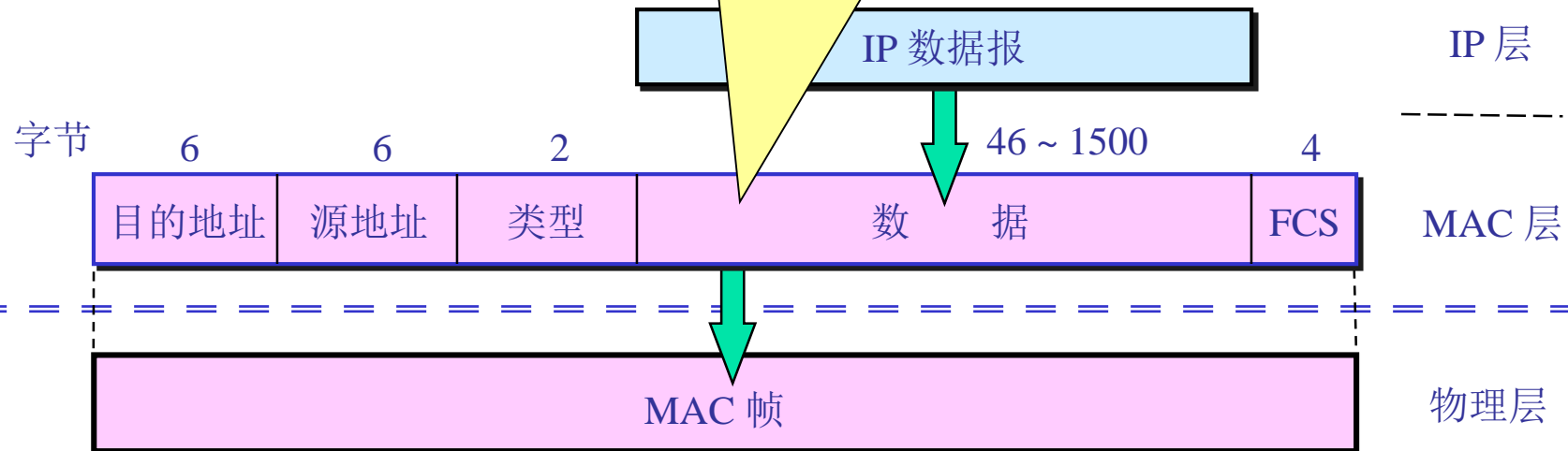


# 以太网 V2 的 MAC 帧格式

数据字段的正式名称是 **MAC 客户数据字段**

最小长度 64 字节 – 18 字节的首部和尾部 = 数据字段的最小长度

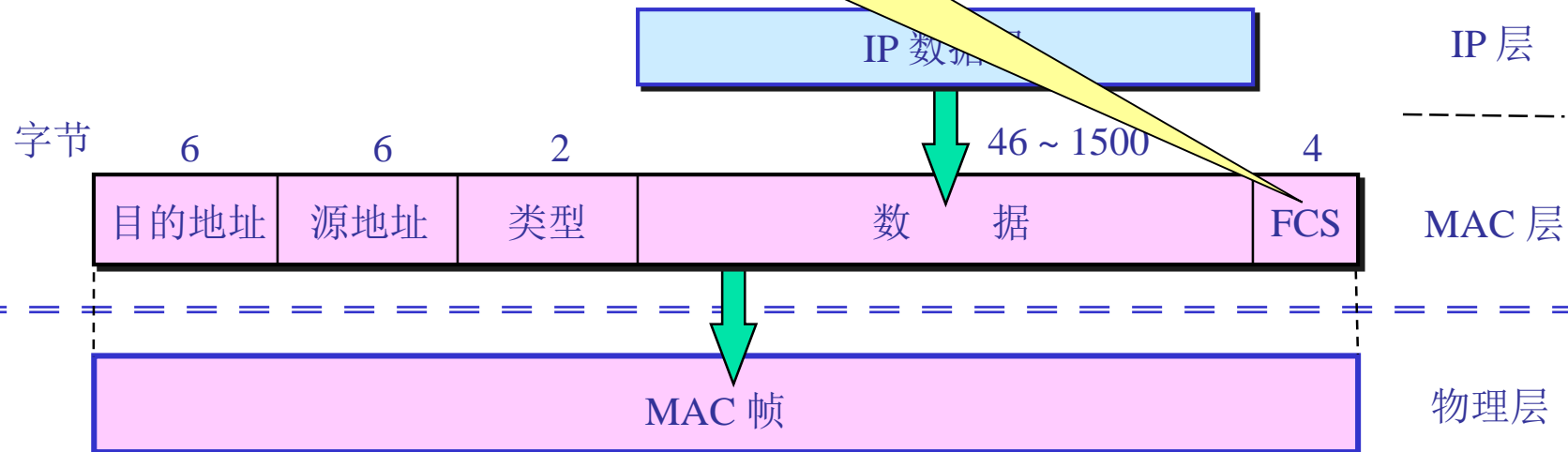
数据字段 46 ~ 1500 字节



# 以太网 V2 的 MAC 帧格式

当传输媒体的误码率为  $1 \times 10^{-8}$  时，  
MAC 子层可使未检测到的差错小于  $1 \times 10^{-14}$ 。

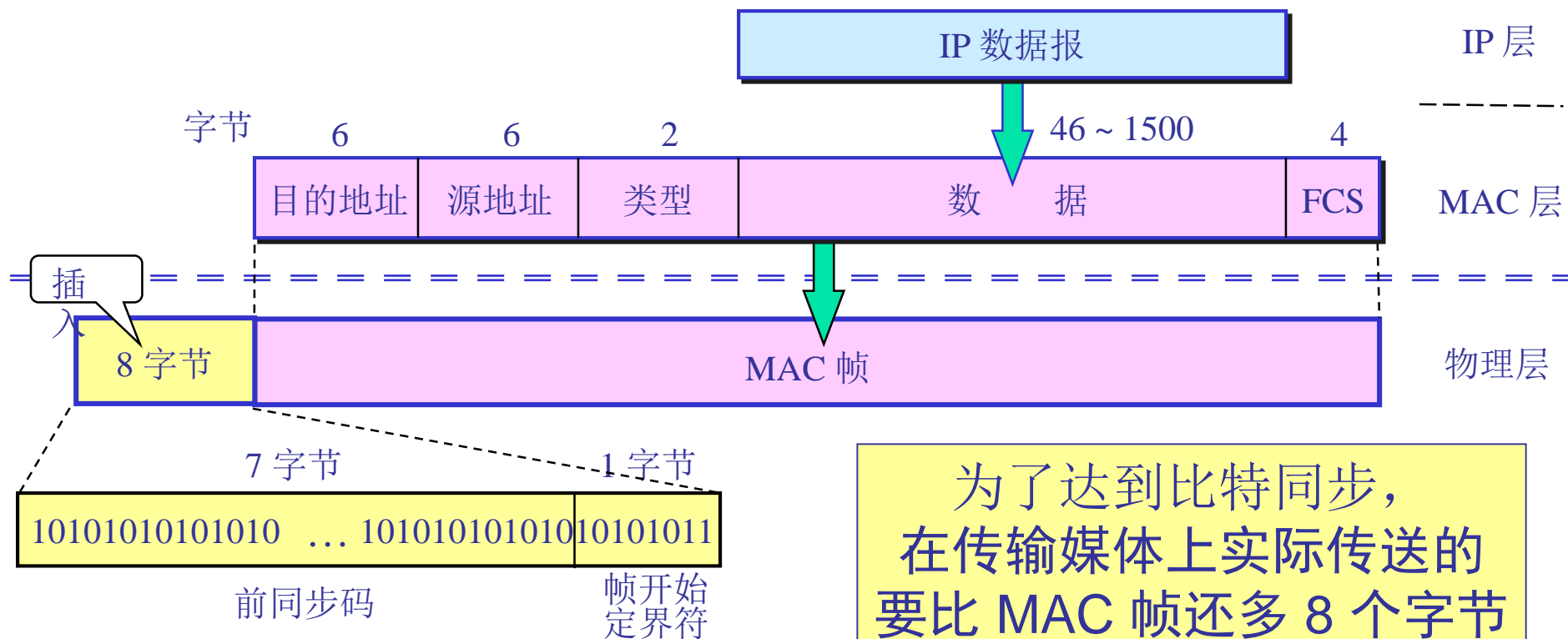
FCS 字段 4 字节



当数据字段的长度小于 46 字节时，  
应在数据字段的后面加入整数字节的填充字段，  
以保证以太网的 MAC 帧长不小于 64 字节。

# 以太网 V2 的 MAC 帧格式

在帧的前面插入的 8 字节中的第一个字段共 7 个字节，是前同步码，用来迅速实现 MAC 帧的比特同步。  
第二个字段是帧开始定界符，表示后面的信息就是 MAC 帧。



为了达到比特同步，  
在传输媒体上实际传送的  
要比 MAC 帧还多 8 个字节



# 作业

---

- 习题: 3-15,3-18,3-20,3-24, 3-25

## 3.4 扩展的局域网

### 3.4.1 在物理层扩展局域网

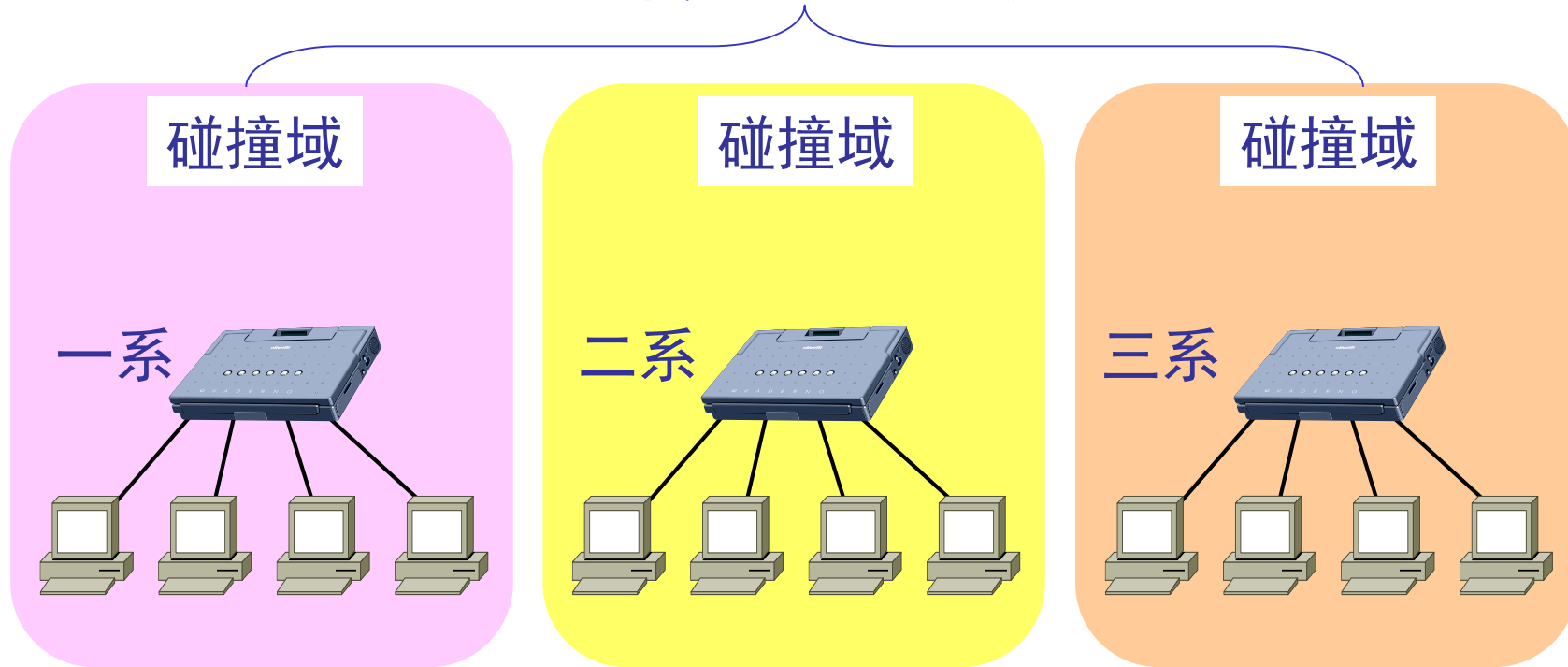
- 主机使用光纤和一对光纤调制解调器连接到集线器。



# 多个集线器可连成更大的局域网

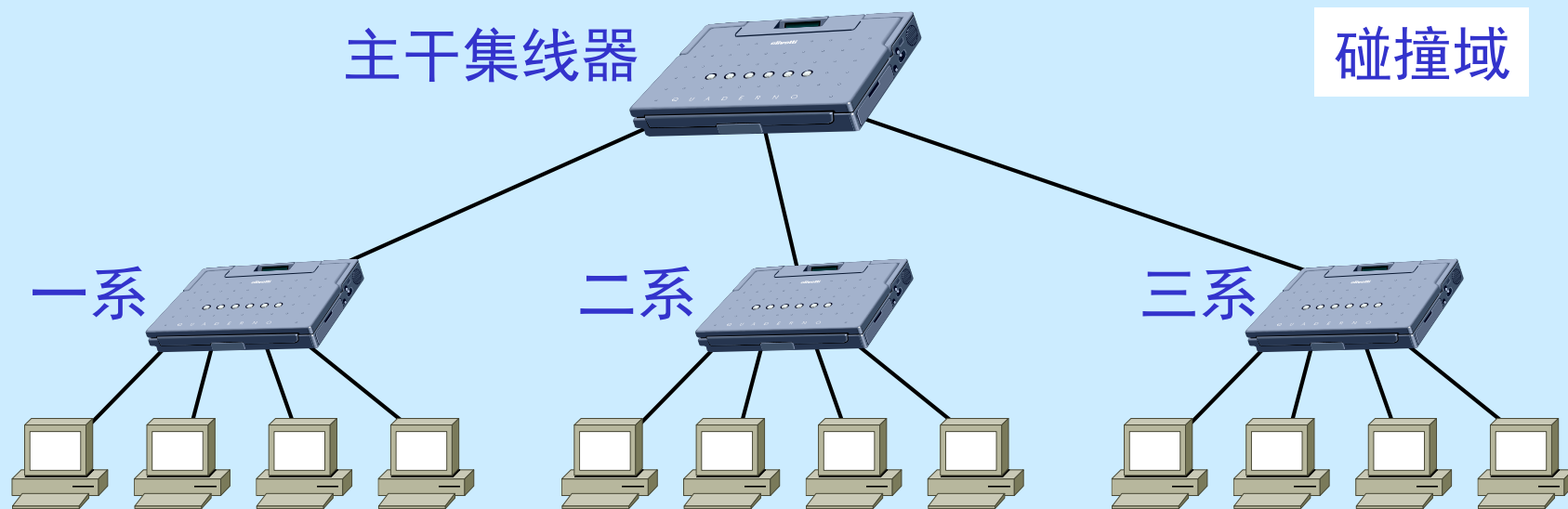
- 某大学有三个系，各自有一个局域网

三个独立的碰撞域



# 用集线器组成更大的局域网 都在一个碰撞域中

一个更大的碰撞域





# 用集线器扩展局域网

---

## ■ 优点

- 使原来属于不同碰撞域的局域网上的计算机能够进行跨碰撞域的通信。
- 扩大了局域网覆盖的地理范围。

## ■ 缺点

- 碰撞域增大了，但总的吞吐量并未提高。
- 如果不同的碰撞域使用不同的数据率，那么就不能用集线器将它们互连起来。





## 3.4.2 在数据链路层扩展局域网

■ 早期使用网桥，现在使用以太网交换机。

- 网桥工作在数据链路层。
- 它根据 MAC 帧的目的地址对收到的帧进行转发和过滤。
- 当网桥收到一个帧时，并不是向所有的接口转发此帧，而是先检查此帧的目的 MAC 地址，然后再确定将该帧转发到哪一个接口，或把它丢弃。

- 1990 年问世的交换式集线器 (switching hub) 可明显地提高以太网的性能。
- 交换式集线器常称为以太网交换机 (switch) 或第二层交换机 (L2 switch)，强调这种交换机工作在数据链路层。



# 以太网交换机的特点

- 以太网交换机实质上就是一个多接口的网桥。
  - 通常都有十几个或更多的接口。
- 每个接口都直接与一个单台主机或另一个以太网交换机相连，并且一般都工作在全双工方式。



# 以太网交换机的特点

---

- 以太网交换机具有并行性。
  - 能同时连通多对接口，使多对主机能同时通信。
- 相互通信的主机都是独占传输媒体，无碰撞地传输数据。



# 独占传输媒体的带宽

- 对于普通 10 Mb/s 的共享式以太网，若共有  $N$  个用户，则每个用户占有的平均带宽只有总带宽(10 Mb/s)的  $N$  分之一。
- 使用以太网交换机时，虽然在每个接口到主机的带宽还是 10 Mb/s，但由于一个用户在通信时是独占而不是和其他网络用户共享传输媒体的带宽，因此对于拥有  $N$  对接口的交换机的总容量为  $N \times 10$  Mb/s。这正是交换机的最大优点。



# 以太网交换机的交换方式

---

## ■ 存储转发方式

- 把整个数据帧先缓存后再进行处理。

## ■ 直通 (cut-through) 方式

- 接收数据帧的同时就立即按数据帧的目的 MAC 地址决定该帧的转发接口，因而提高了帧的转发速度。
- 缺点是它不检查差错就直接将帧转发出去，因此有可能也将一些无效帧转发给其他的站。

## 2. 以太网交换机的自学习功能

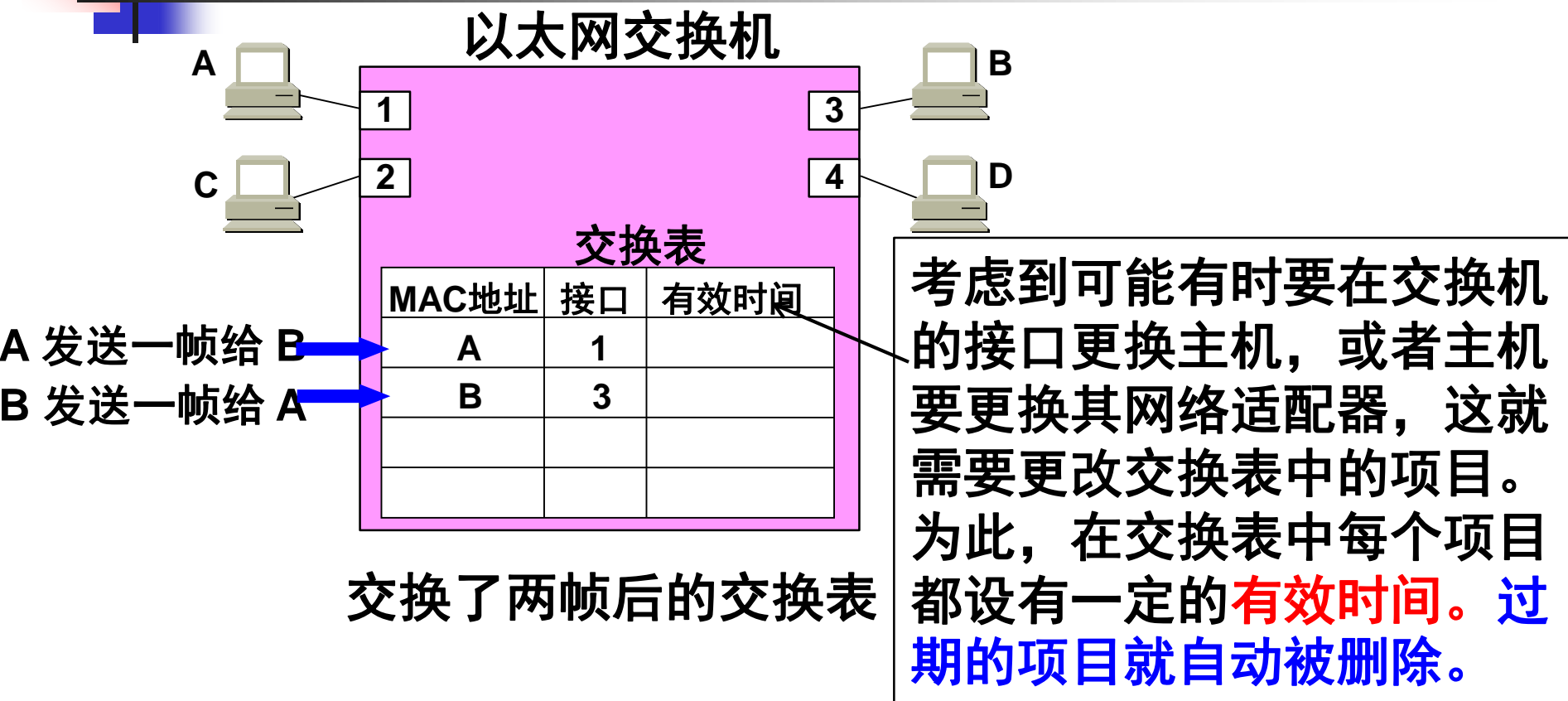
- 以太网交换机运行自学习算法自动维护**交换表**。

- 开始时，以太网交换机里面的交换表是空的。



交换表一开始是空的

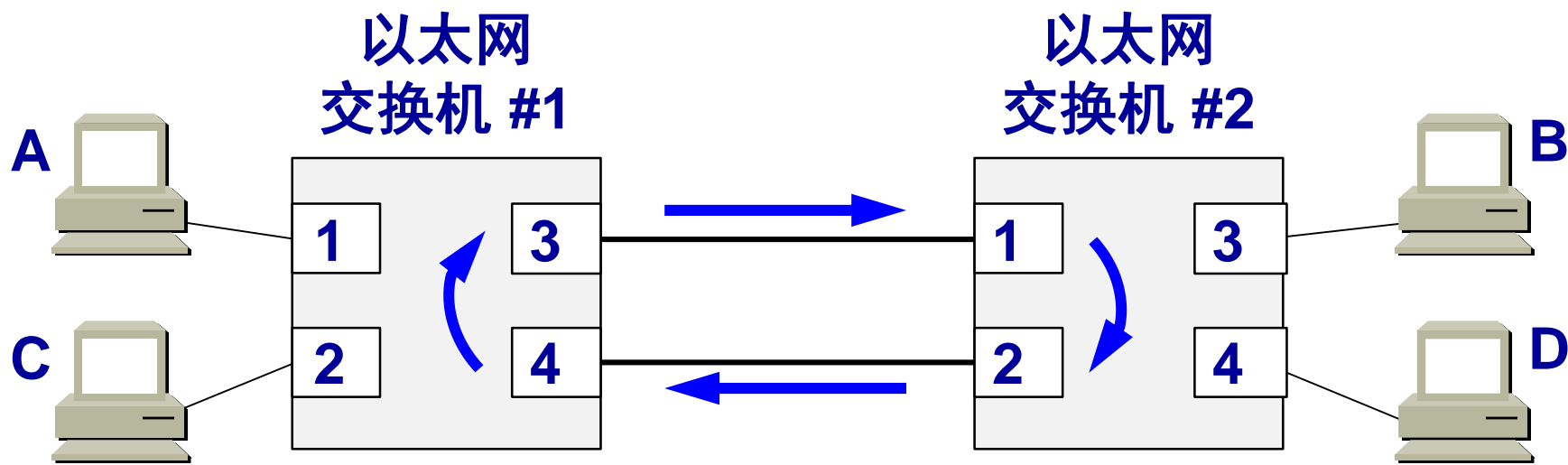
# 按照以下自学习算法 处理收到的帧和建立交换表



以太网交换机的这种自学习方法使得以太网交换机能够即插即用，不必人工进行配置，因此非常方便。

# 交换机使用了生成树协议STP

- 增加冗余链路时，自学习的过程就可能导致以太网帧在网络的某个环路中无限制地兜圈子。



在两个交换机之间兜圈子的帧





### 3. 从总线以太网到星形以太网

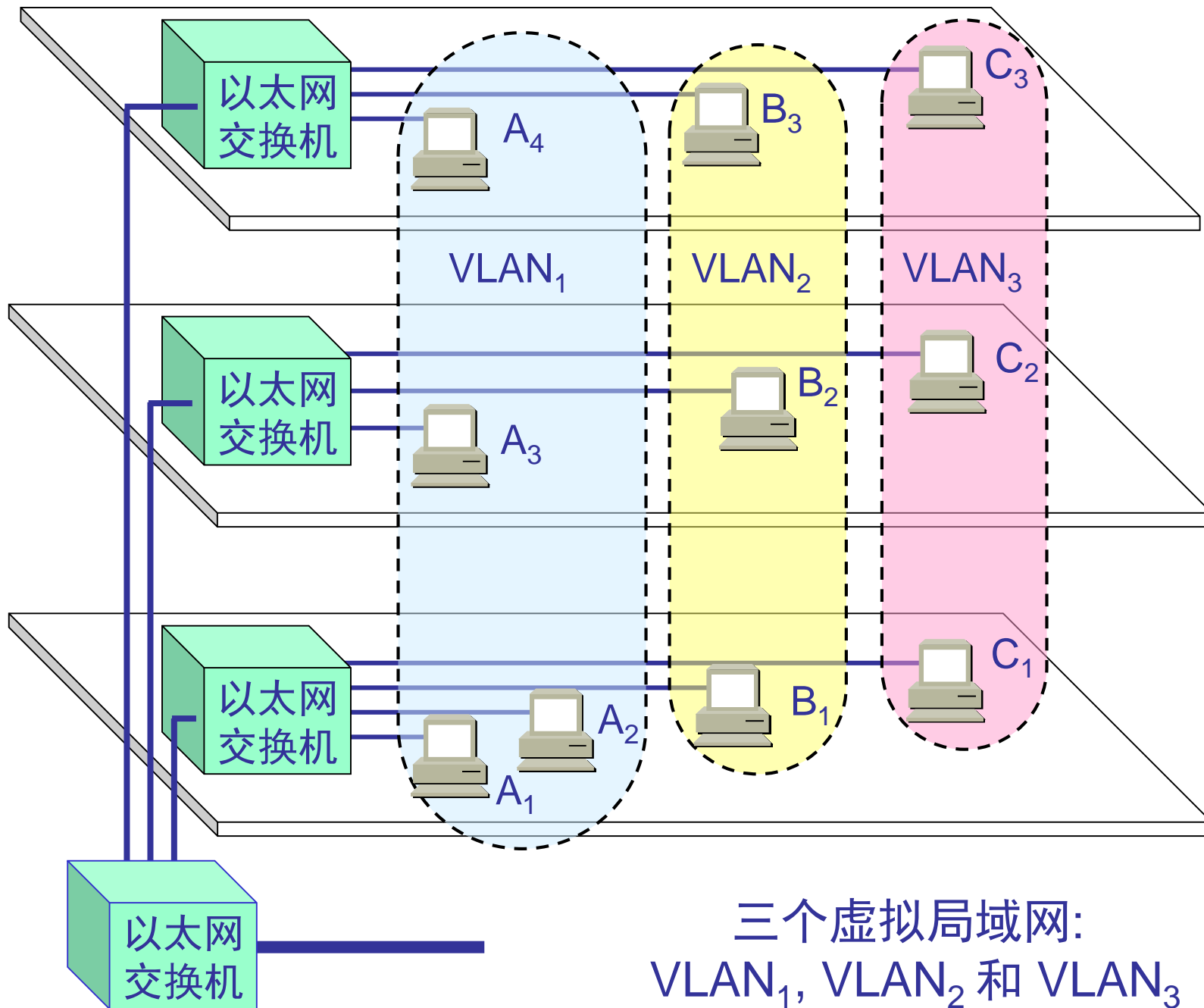
---

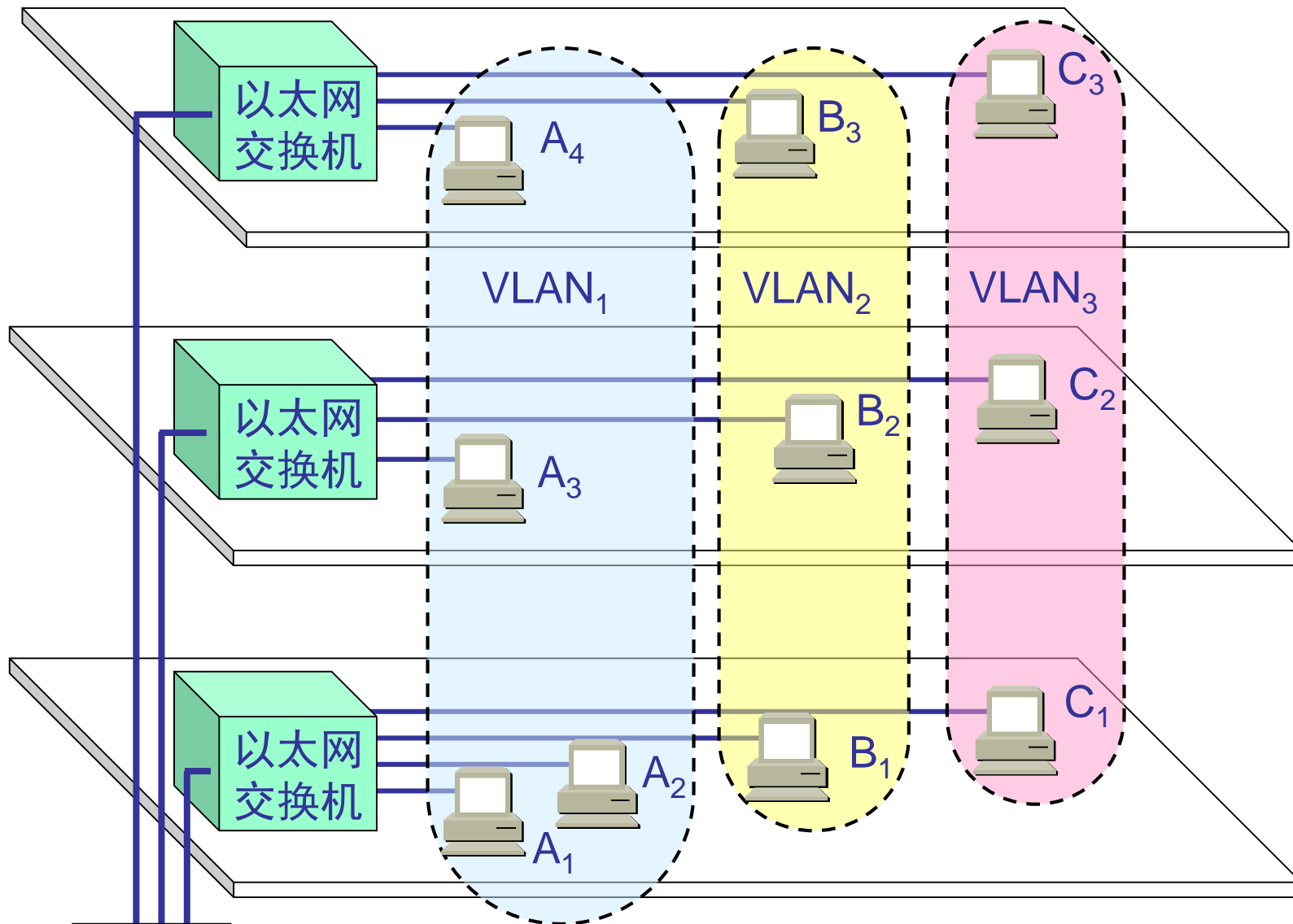
- 早期，以太网采用无源的总线结构。现在，采用以太网交换机的星形结构成为以太网的首选拓扑。
- 总线以太网使用 CSMA/CD 协议，以半双工方式工作。
- 以太网交换机不使用共享总线，没有碰撞问题，因此不使用 CSMA/CD 协议，而是以全双工方式工作。但仍然采用以太网的帧结构。



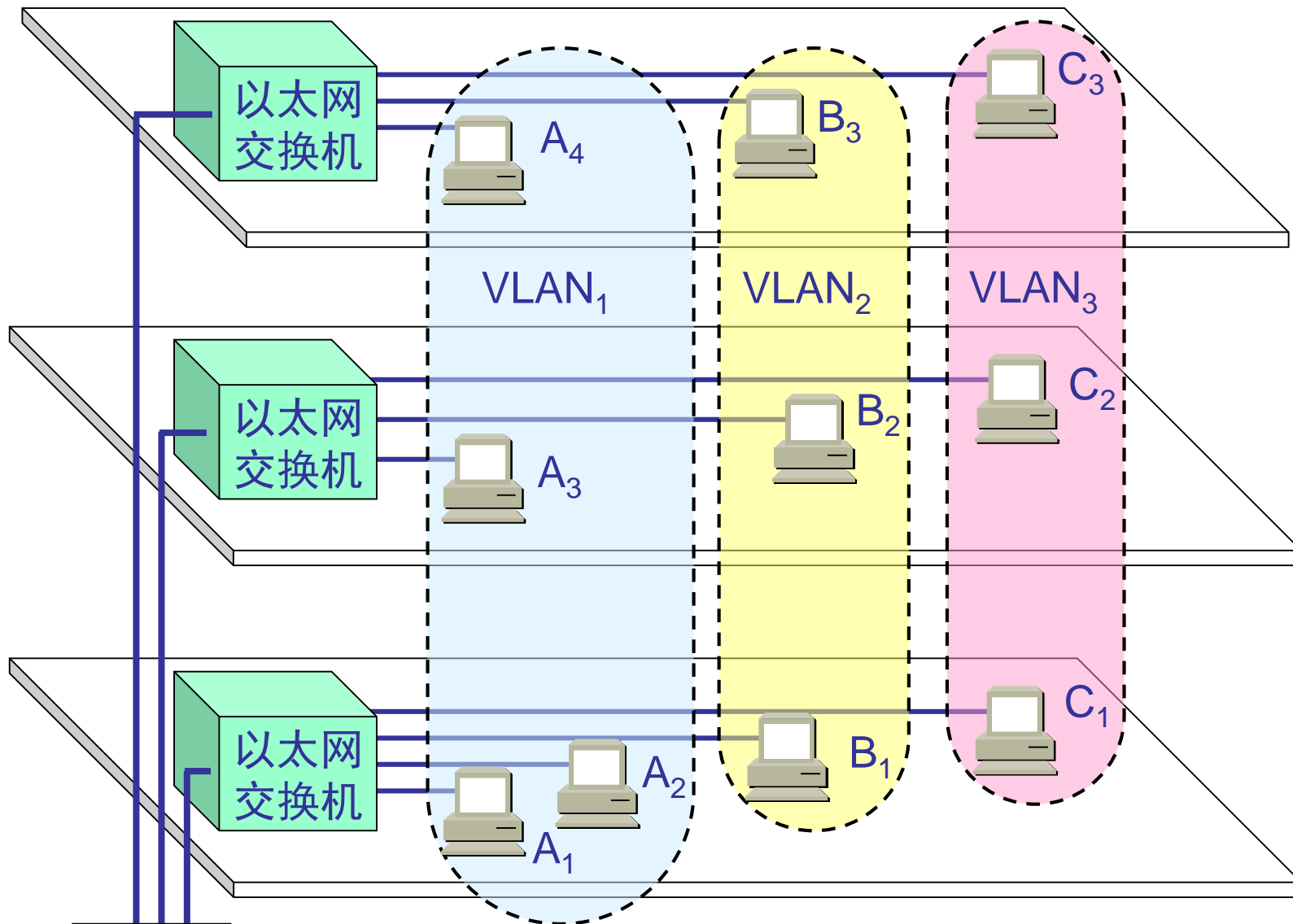
# 利用以太网交换机可以很方便地实现虚拟局域网

- **虚拟局域网** VLAN 是由一些局域网网段构成的与物理位置无关的逻辑组。
  - 这些网段具有某些共同的需求。
  - 每一个 VLAN 的帧都有一个明确的标识符，指明发送这个帧的工作站是属于哪一个 VLAN。
- 虚拟局域网其实只是局域网给用户提供服务的一种服务，而并不是一种新型局域网。

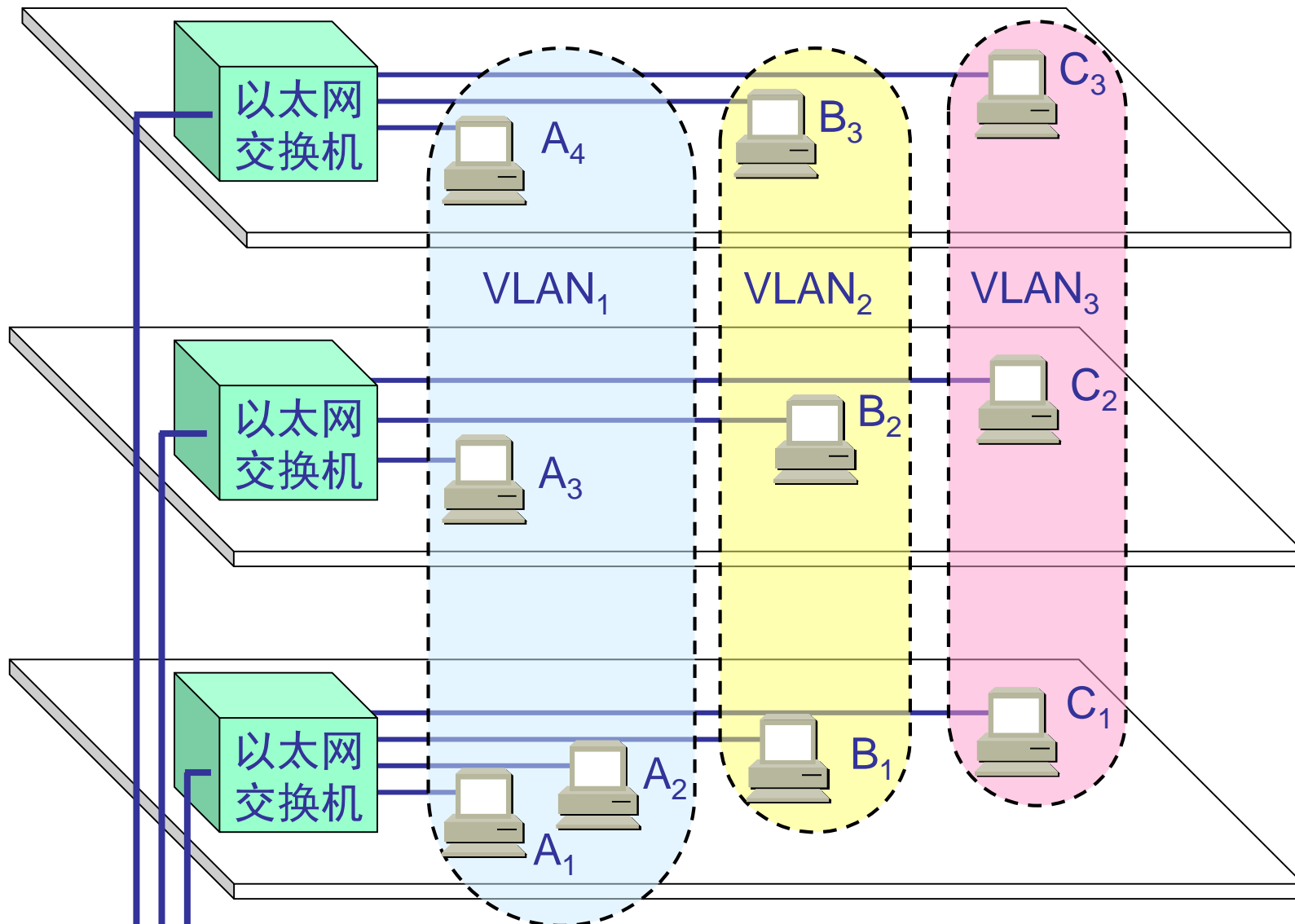




当 B<sub>1</sub> 向 VLAN<sub>2</sub> 工作组内成员发送数据时，  
工作站 B<sub>2</sub> 和 B<sub>3</sub> 将会收到广播的信息。



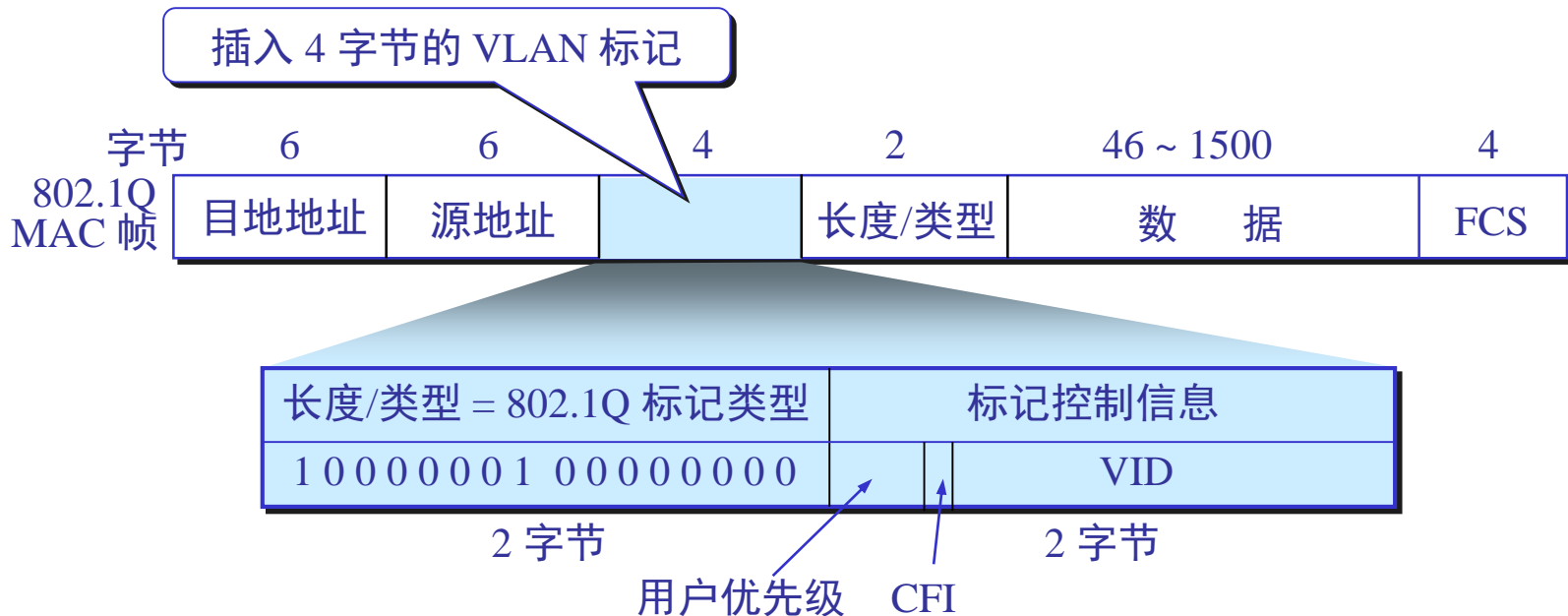
B<sub>1</sub> 发送数据时，工作站 A<sub>1</sub>, A<sub>2</sub> 和 C<sub>1</sub> 都不会收到 B<sub>1</sub> 发出的广播信息。



虚拟局域网限制了接收广播信息的工作站数，使得网络不会因传播过多的广播信息（即“广播风暴”）而引起性能恶化。

# 虚拟局域网使用的以太网帧格式

- 虚拟局域网协议允许在以太网的帧格式中插入一个 4 字节的标识符，称为 VLAN 标记(tag)，用来指明发送该帧的工作站属于哪一个虚拟局域网。



## 3.5 高速以太网

### 3.5.1 100BASE-T 以太网

- 速率达到或超过 100 Mb/s 的以太网称为**高速以太网**。
- 在双绞线上传送 100 Mb/s 基带信号的星型拓扑以太网(集线器或交换机), 仍使用 IEEE 802.3 的CSMA/CD 协议
- 100BASE-T 以太网又称为**快速以太网**(Fast Ethernet)。





# 100BASE-T 以太网的特点

- 可在全双工方式下(交换机)工作而无冲突发生。因此, 不使用 CSMA/CD 协议。
- MAC 帧格式仍然是 802.3 标准规定的。
- 保持最短帧长不变, 但将一个网段的最大电缆长度减小到 100 m。
- 帧间时间间隔从原来的  $9.6\ \mu\text{s}$  改为现在的  $0.96\ \mu\text{s}$ 。

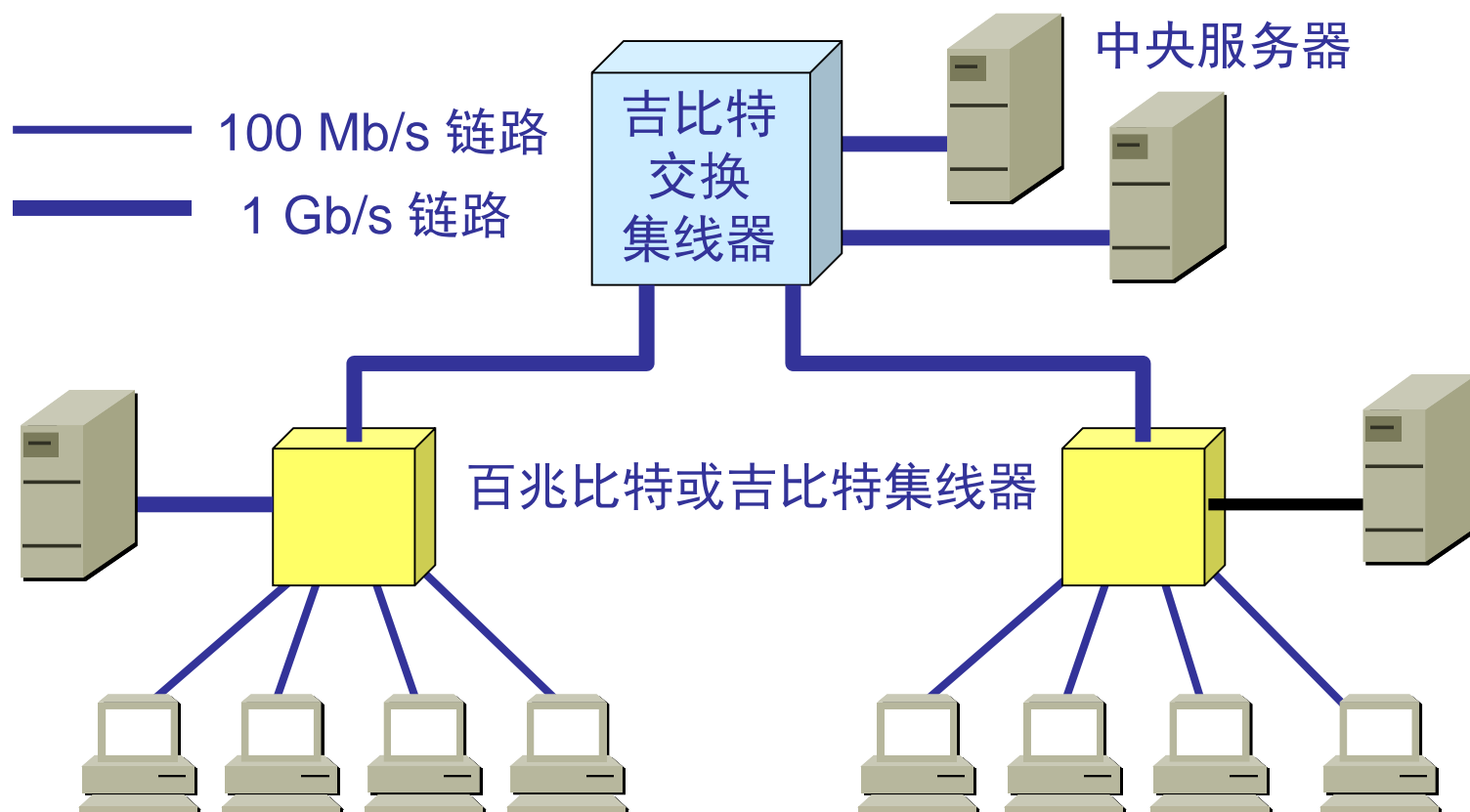


## 3.5.2 吉比特以太网

---

- 允许在 1 Gb/s 下全双工和半双工两种方式工作。
- 使用 802.3 协议规定的帧格式。
- 在半双工方式下使用 CSMA/CD 协议（全双工方式不需要使用 CSMA/CD 协议）。
- 与 10BASE-T 和 100BASE-T 技术向后兼容。

# 吉比特以太网的配置举例



### 3.5.3 10 吉比特以太网和更快的以太网

- 10 吉比特以太网与 10 Mb/s, 100 Mb/s 和 1 Gb/s 以太网的帧格式完全相同。
- 10 吉比特以太网还保留了 802.3 标准规定的以太网最小和最大帧长, 便于升级。
- 10 吉比特以太网只工作在全双工方式, 因此没有争用问题, 也不使用 CSMA/CD 协议。



# 端到端的以太网传输

---

- 10 吉比特以太网的出现，以太网的工作范围已经从局域网（校园网、企业网）扩大到城域网和广域网，从而实现了端到端的以太网传输。



## 3.5.4 使用高速以太网 进行宽带接入

---

- 以太网已成功地把速率提高到  $1 \sim 10 \text{ Gb/s}$ ，所覆盖的地理范围也扩展到了城域网和广域网，因此现在人们正在尝试使用以太网进行宽带接入。
- 以太网接入的重要特点是它可提供双向的宽带通信，并且可根据用户对带宽的需求灵活地进行带宽升级。
- 采用以太网接入可实现端到端的以太网传输，中间不需要再进行帧格式的转换。这就提高了数据的传输效率和降低了传输的成本。



# 课后作业

---

- 习题: 3-27 (理由), 3-28, 3-29, 3-33,  
3-34