第 3 章 数据链路层

南京信息工程大学 黄群 副教授

第 3 章 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 扩展的以太网
- 3.5 高速以太网

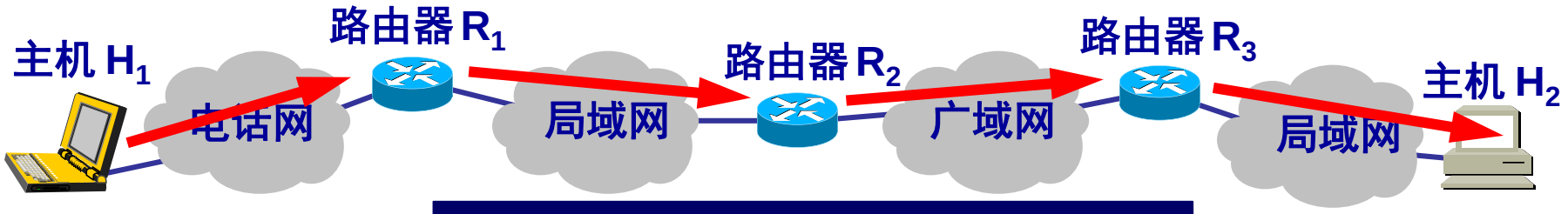
数据链路层使用的信道

数据链路层使用的信道主要有以下两种类型：

- 点对点信道。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。

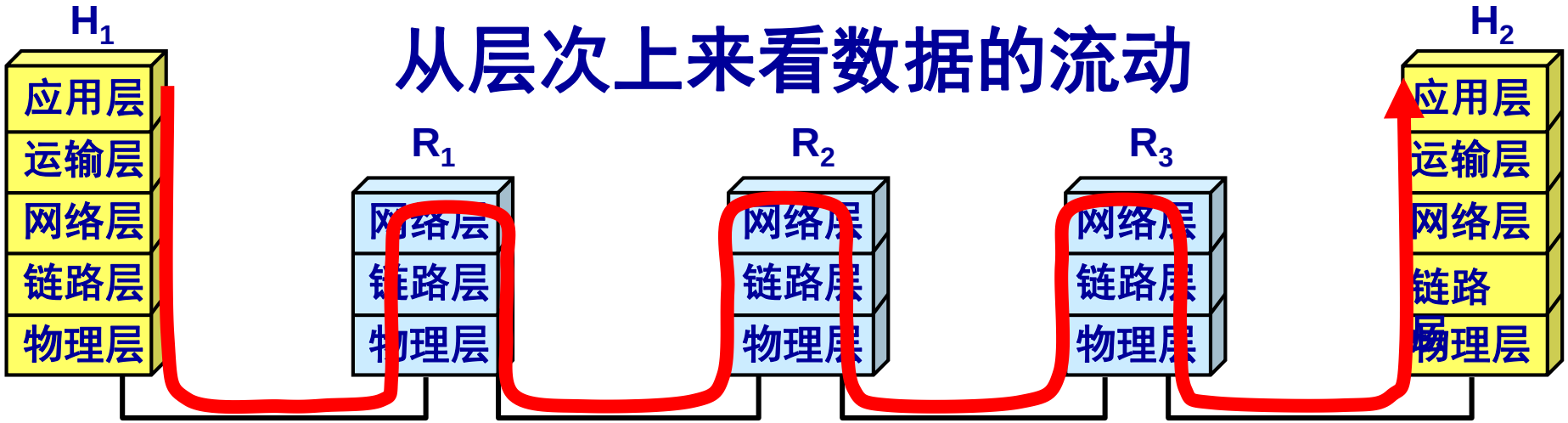
数据链路层的简单模型

主机 H_1 向 H_2 发送数据



H_1 到 H_2 所经过的网络可以是多种的

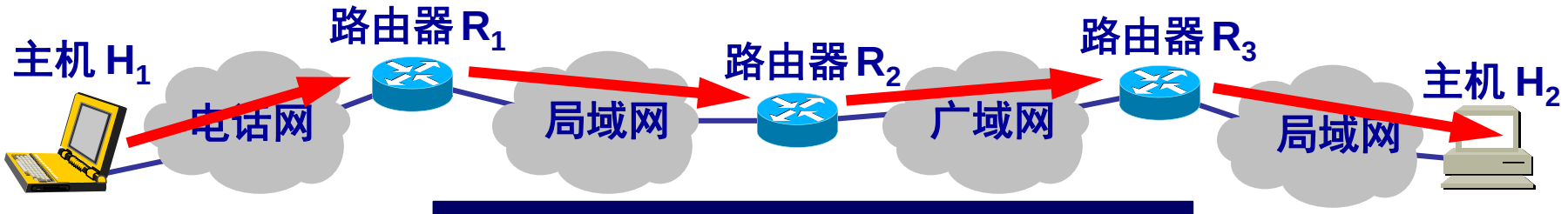
从层次上来看数据的流动



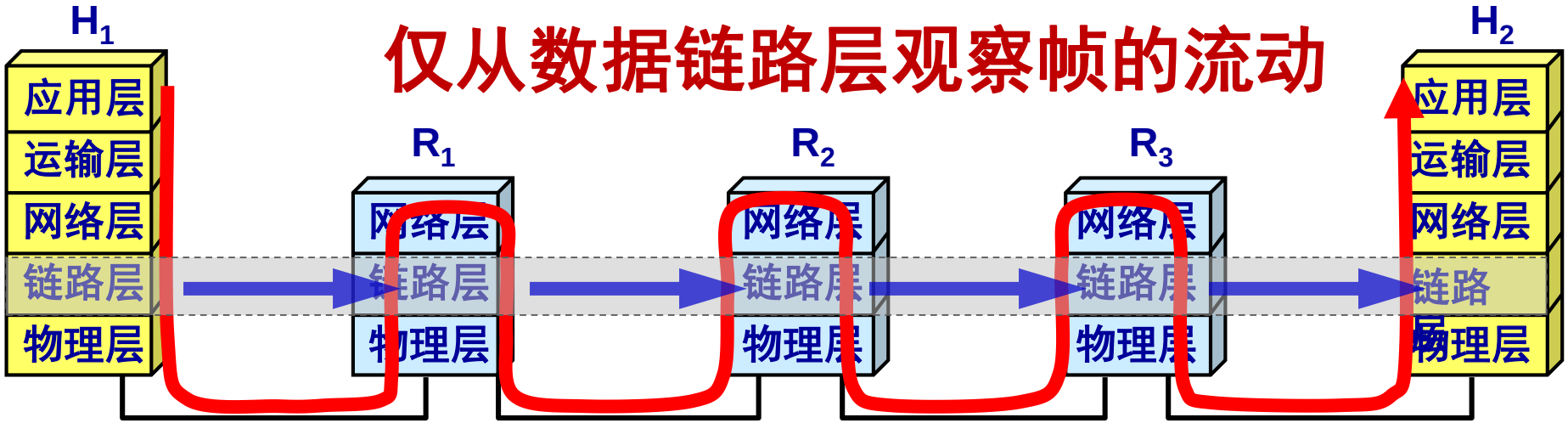
数据链路层的地位

数据链路层的简单模型（续）

主机 H_1 向 H_2 发送数据



H_1 到 H_2 所经过的网络可以是多种的



仅从数据链路层观察帧的流动

不同的链路层可能采用不同的数据链路层协议

只考虑数据在数据链路层的流动

3.1 使用点对点信道的数据链路层

- 3.1.1 数据链路和帧
- 3.1.2 三个基本问题

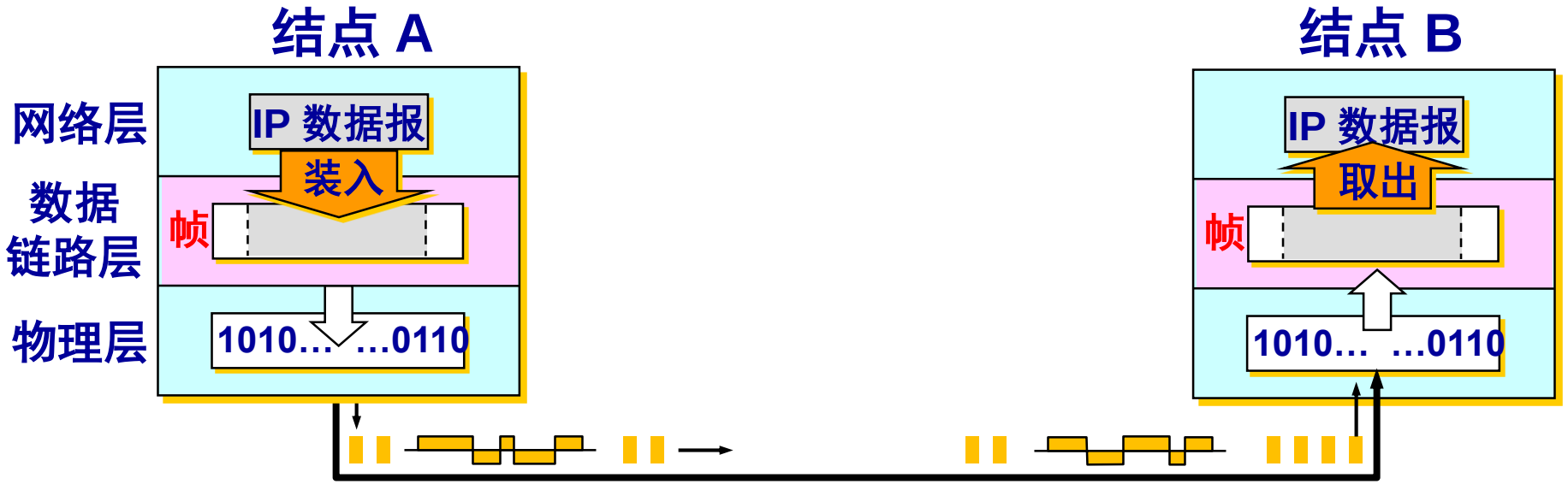
3.1.1 数据链路和帧

- **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) 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。
  - 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。

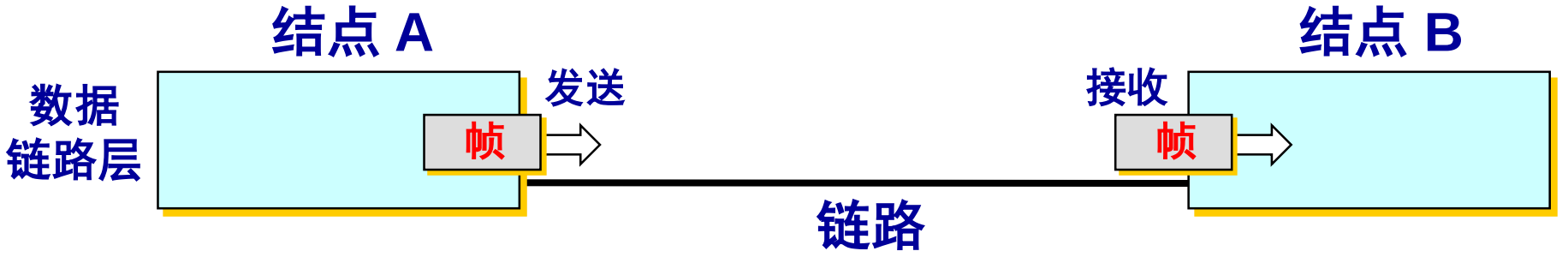
3.1.1 数据链路和帧

- 也有人采用另外的术语。这就是把链路分为物理链路和逻辑链路。
- **物理链路**就是上面所说的链路。
- **逻辑链路**就是上面的数据链路，是物理链路加上必要的通信协议。
- 早期的数据通信协议曾叫做**通信规程**(procedure)。因此在数据链路层，规程和协议是同义语。

数据链路层传送的是帧



(a) 三层的简化模型

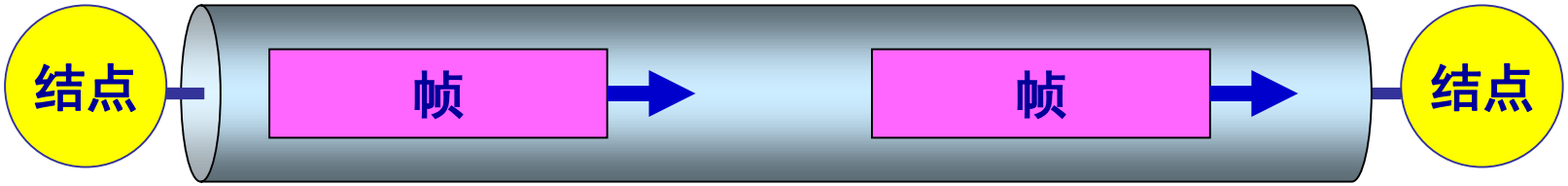


(b) 只考虑数据链路层

使用点对点信道的数据链路层

数据链路层像个数字管道

- 常常在两个对等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。



- 数据链路层不必考虑物理层如何实现比特传输的细节。甚至还可以更简单地设想好像是沿着两个数据链路层之间的水平方向把帧直接发送到对方。

3.1.2 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：

1. 封装成帧
2. 透明传输
3. 差错控制

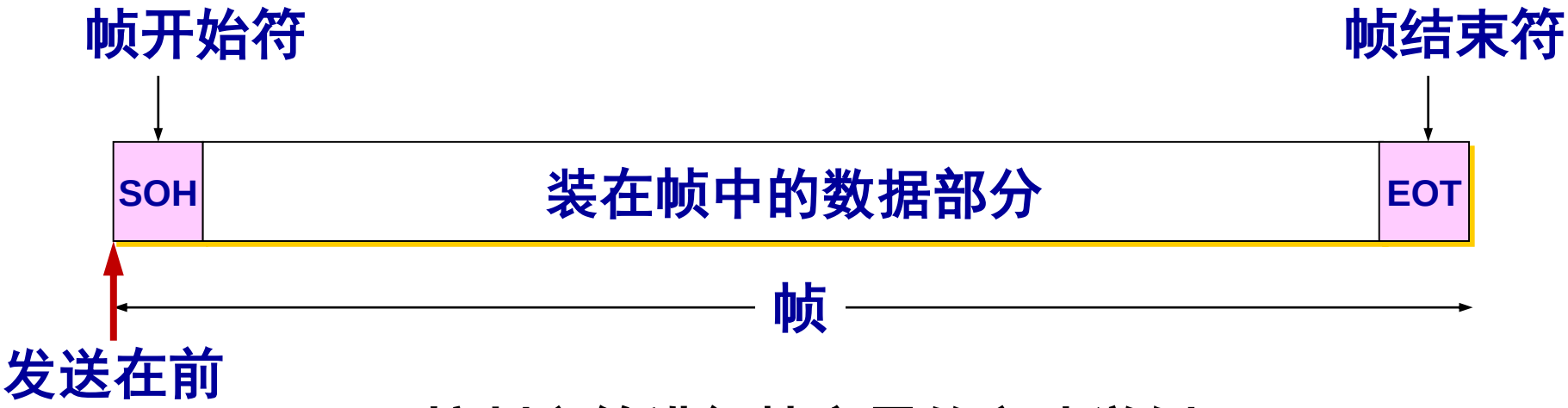
1. 封装成帧

- **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

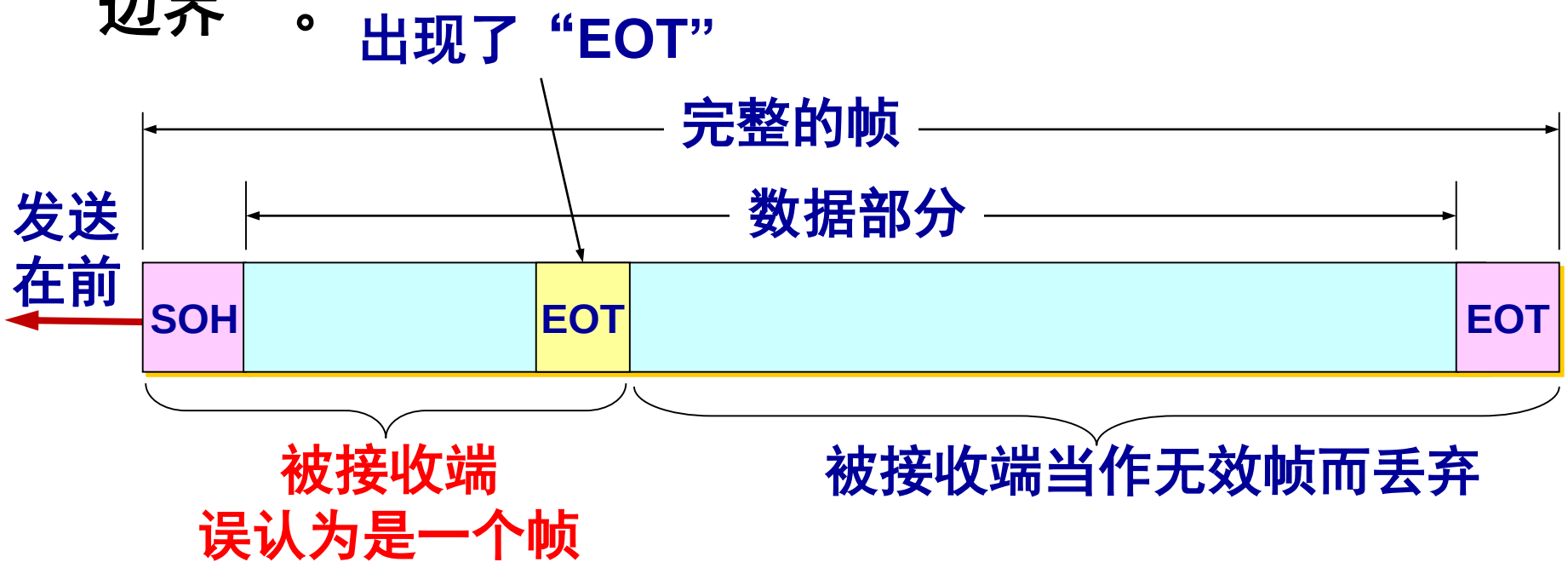
- 当数据是由可打印的 ASCII 码组成的文本文件时，帧定界可以使用特殊的帧定界符。
- 控制字符 SOH (Start Of Header) 放在一帧的最前面，表示帧的首部开始。另一个控制字符 EOT (End Of Transmission) 表示帧的结束。



用控制字符进行帧定界的方法举例

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”。

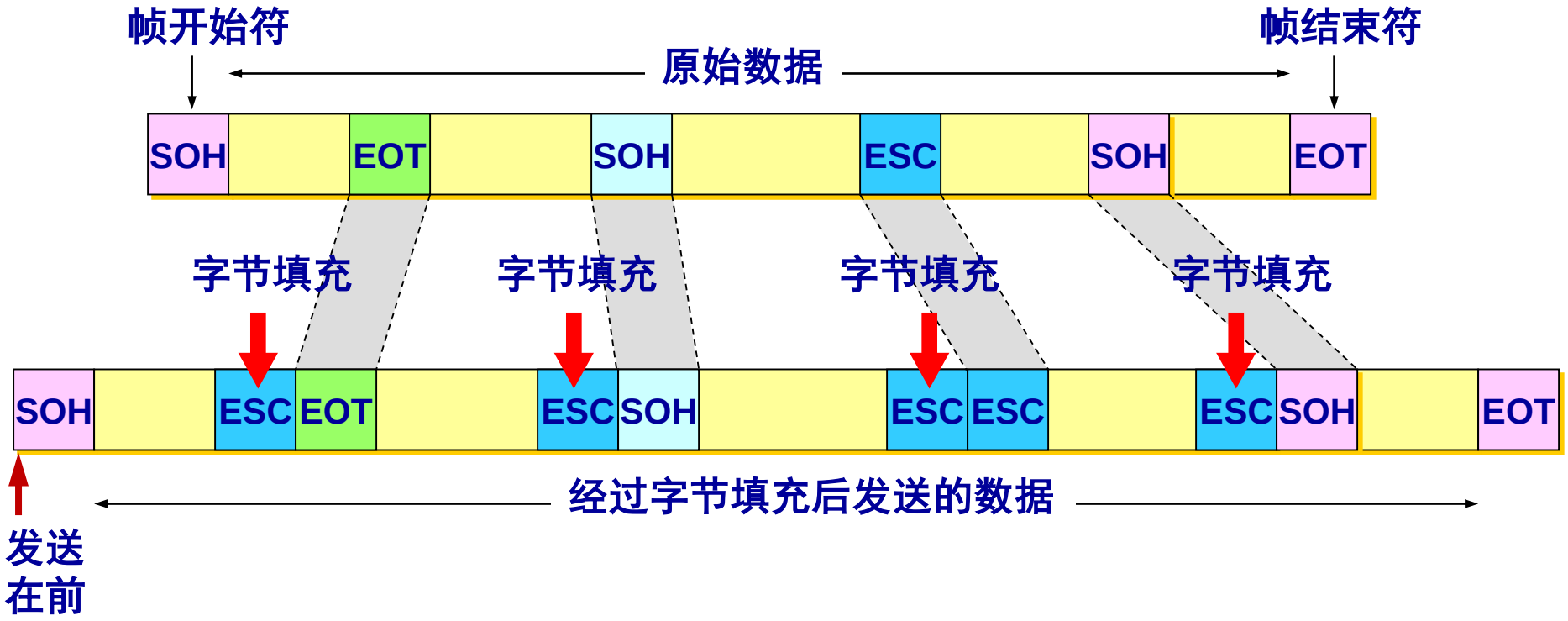


数据部分恰好出现与 EOT 一样的代码

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。
- 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。

循环冗余检验的原理

- 任何一个K位的帧看成是一个K+1次的多项式M(X)
- 如 101001 看成 是 $x^5+x^3+x^0$
- 设定一个生成多项式G(X),为n阶，k>n
- 如设 $n = 3$，$G(x)=x^3+x^2+x^0$ 除数 $P = 1101$
- $X^nM(X) /G(X) =Q(X) +R(X)/G(X)$,其中 Q(X)为商， R(X)为校验码。

将 CRC码接在帧后一起发送，即发送的数据为

- $X^nM(X)+R(X)$

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，即余数 R 是 n 位。
- 将余数 R 作为冗余码拼接在数据 M 后面发送出去。

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 $2^nM = 101001000$。
- 模 2 运算的结果是：商 $Q = 110101$,

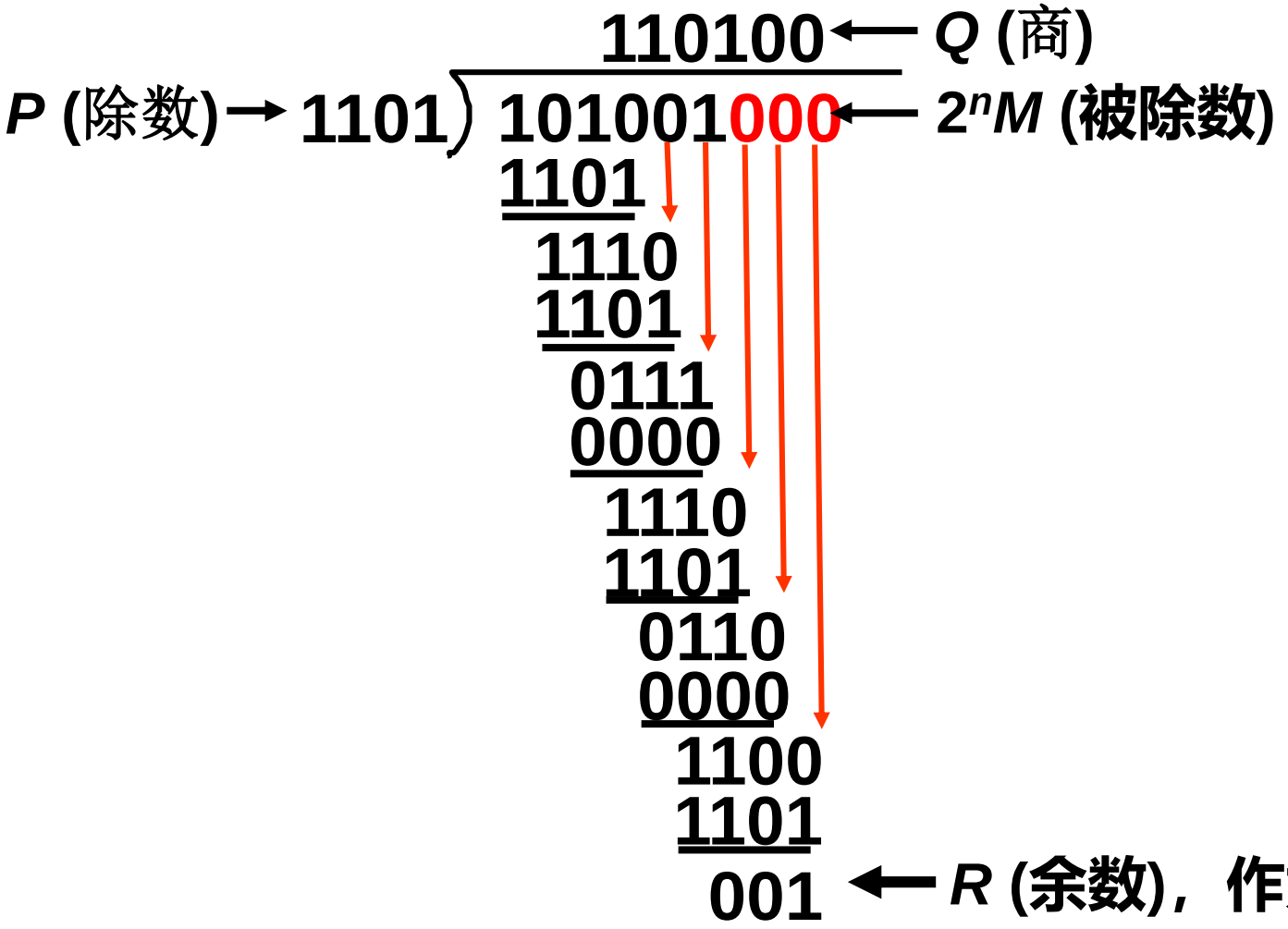
  余数 $R = 001$。
- 把余数 R 作为冗余码添加在数据 M 的后面发送出去。发送的数据是：$2^nM + R$

  即：101001001，共 $(k + n)$ 位。

模2运算

- 模2运算即是异或运算
- 加法不进位，减法不借位
- 0+0=0，1+0=1，0+1=1，1+1=0
- 0-0=0 ，0-1=1；1-0=1；1-1=0

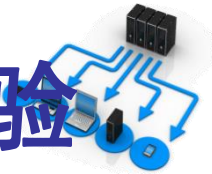
循环冗余检验的原理说明



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。

接收方 检查 CRC码

- 上例中 接收方收到 101001 001， 为被除数
- 除数 *P* = 1101 ，
- 进行除法运算， 结果 =？ (练习）
- 结论？

练习

- 设数据为 11100110，
- 生成多项式G(X)为 $x^4+x^3+x^0$,
- 求校验码。
- 被除数 =
- 除数=

CRC示例2：发送端

例：设数据$M(x)$ =11100110， $G(x)$= 11001, r=4

求CRC校验码$R(x)$:

$G(x)$ →11001) 11100110**0000**

10110110 ← 商 $Q(x)$

11001

10111

11001

11100

11001

101**00**

11001

1101**0**

11001

0110 ← 余数 $R(x)$

发送码：

$M(x)\cdot x^r+R(x)$

=11100110 0110

数据位 校验位

应当注意

- 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“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。
- 本章介绍的数据链路层协议都不是可靠传输的协议。

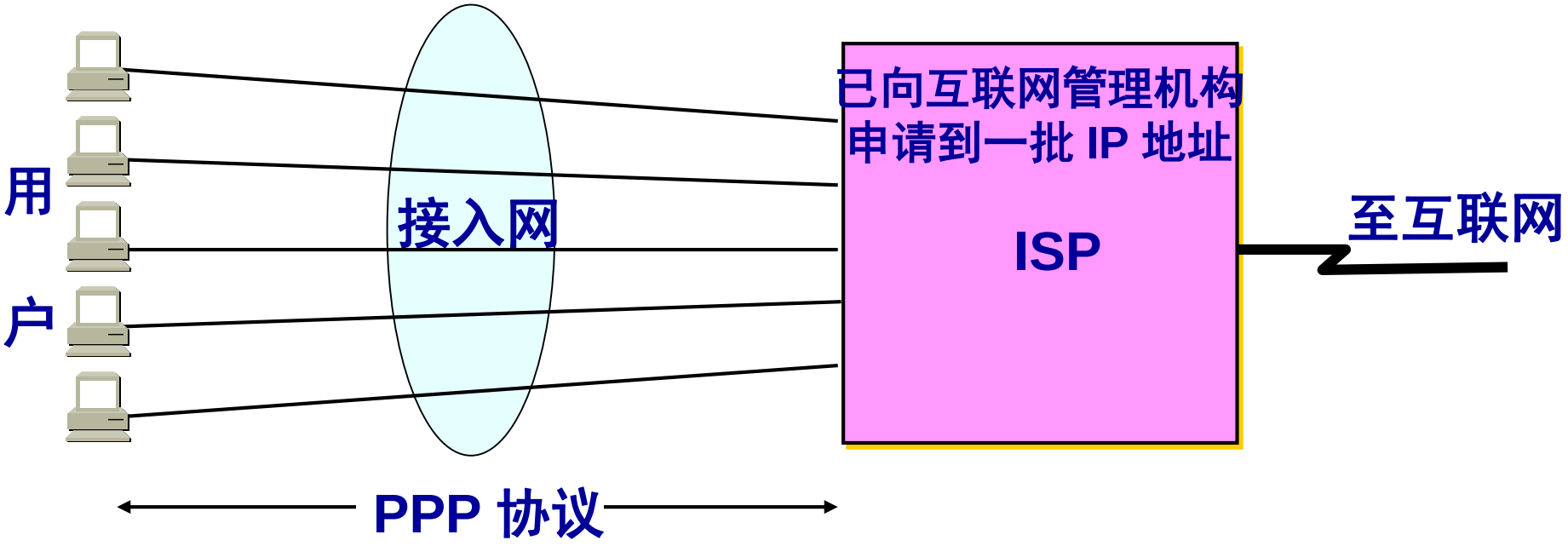
3.2 点对点协议 PPP

- 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

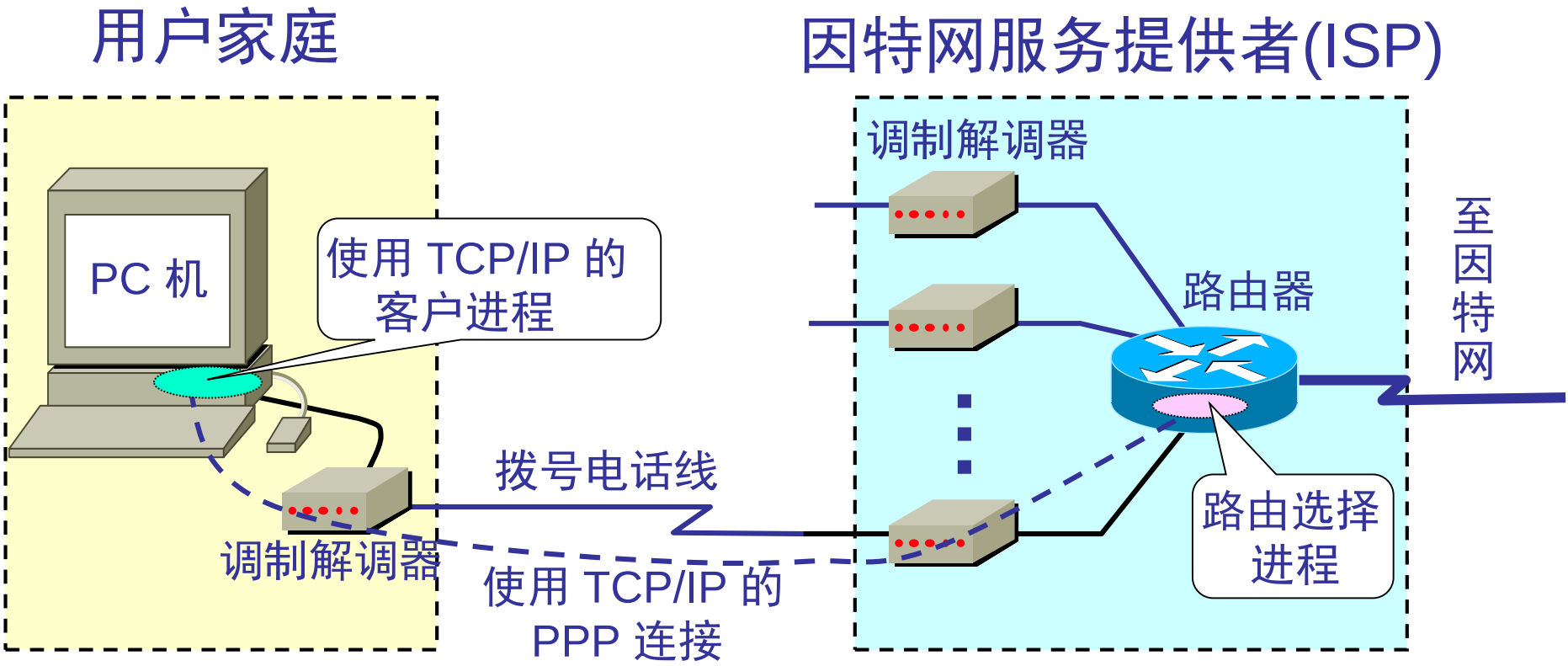
3.2.1 PPP 协议的特点

- 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)。
- 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。
- 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



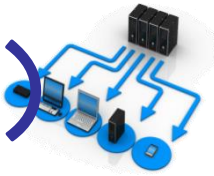
用户拨号入网的示意图


用户家庭
因特网服务提供者(ISP)
PC 机
使用 TCP/IP 的
客户进程
调制解调器
拨号电话线
调制解调器
使用 TCP/IP 的
PPP 连接
调制解调器
路由器
至
因
特
网
路由选择
进程

1. 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。

1. PPP 协议应满足的需求（续）



- 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。

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

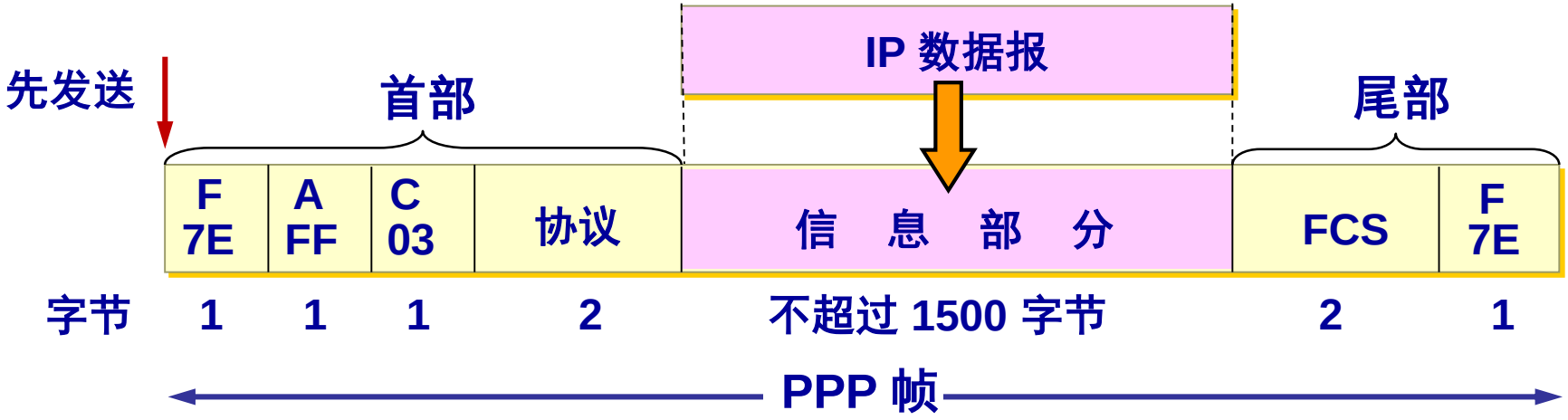
3. PPP 协议的组成

- PPP 协议有三个组成部分：
  - (1) 一个将 IP 数据报封装到串行链路的方法。
  - (2) 链路控制协议 LCP (Link Control Protocol)。
  - (3) 网络控制协议 NCP (Network Control Protocol)。

3.2.2 PPP 协议的帧格式

- 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。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。

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。

习题

- 如果接受方收到 一个PPP帧数据部分 ：
- 7D 5E FE 27 7D 5D 7D 5D 65 7D 5E

- 发送方发送的数据为

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 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

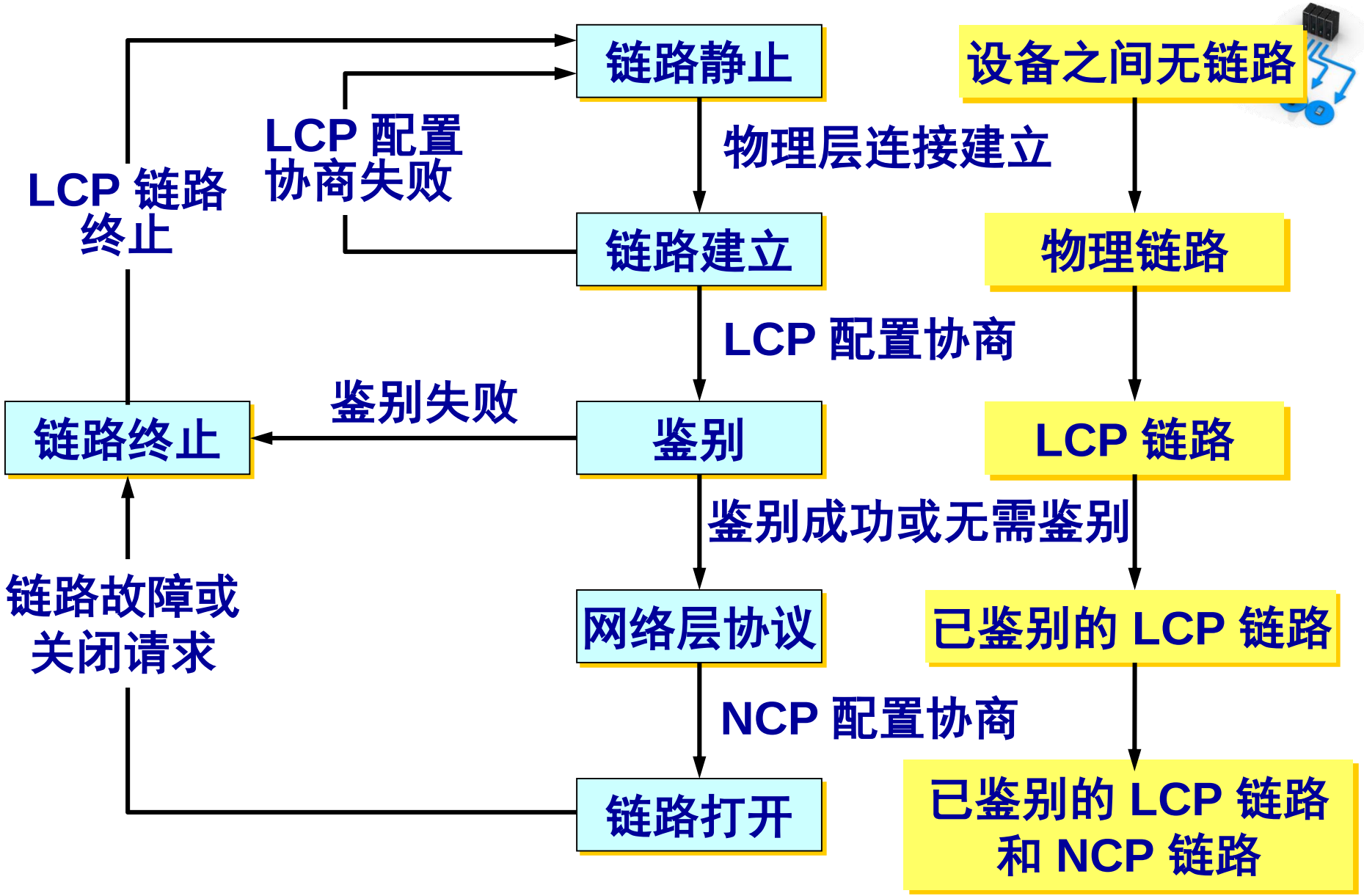
0 1 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0

接收端删除填入的 0 比特

零比特的填充与删除

3.2.3 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）。
- 这些分组及其响应选择一些 PPP 参数，并进行网络层配置，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。
- 可见，PPP 协议已不是纯粹的数据链路层的协议，它还包含了物理层和网络层的内容。


链路静止
设备之间无链路
LCP 配置
协商失败
物理层连接建立
LCP 链路
终止
链路建立
物理链路
LCP 配置协商
鉴别失败
链路终止
鉴别
LCP 链路
鉴别成功或无需鉴别
链路故障或
关闭请求
网络层协议
已鉴别的 LCP 链路
NCP 配置协商
链路打开
已鉴别的 LCP 链路
和 NCP 链路

PPP 协议的状态图

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
- 3.3.2 CSMA/CD 协议
- 3.3.3 使用集线器的星形拓扑
- 3.3.4 以太网的信道利用率
- 3.3.5 以太网的 MAC 层

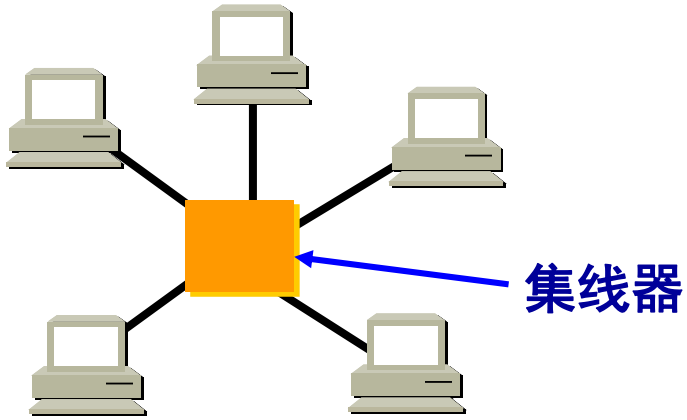
3.3.1 局域网的数据链路层

- 局域网最主要的特点是：
  - 网络为一个单位所拥有；
  - 地理范围和站点数目均有限。
- 局域网具有如下主要优点：
  - 具有广播功能，从一个站点可很方便地访问全网。局域网上的主机可共享连接在局域网上的各种硬件和软件资源。
  - 便于系统的扩展和逐渐地演变，各设备的位置可灵活调整和改变。
  - 提高了系统的可靠性、可用性和残存性。

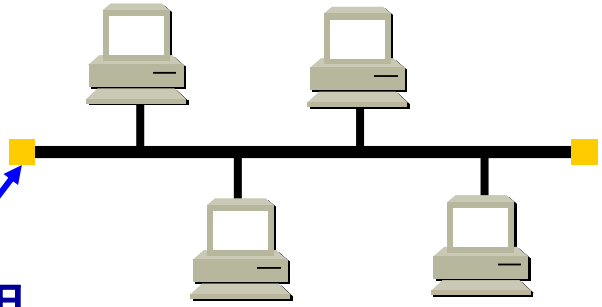
局域网的主要技术要素

- 网络拓扑结构
- 传输介质
- 介质访问控制方法

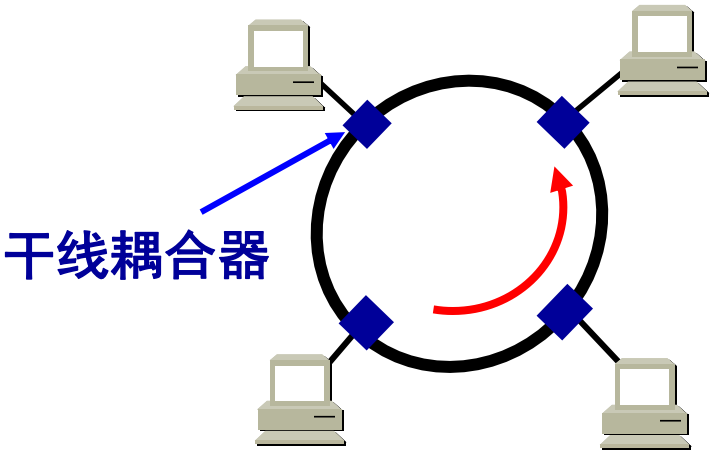
局域网拓扑结构



星形网



总线网



环形网

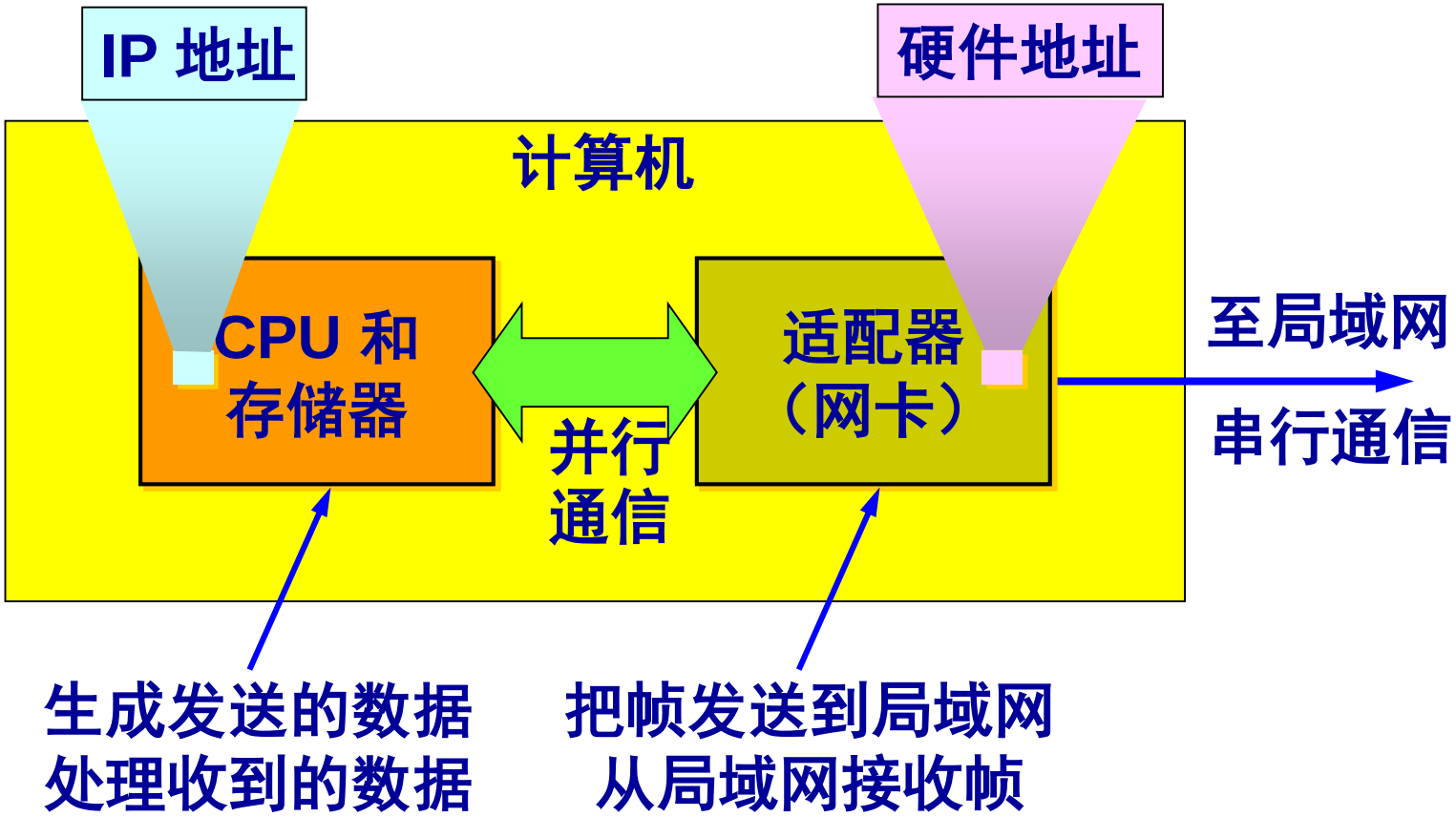
1. 以太网的两个标准

- DIX Ethernet V2 是世界上第一个局域网产品（以太网）的规约。
- IEEE 802.3 是第一个 IEEE 的以太网标准。
- 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 。

2. 适配器的作用

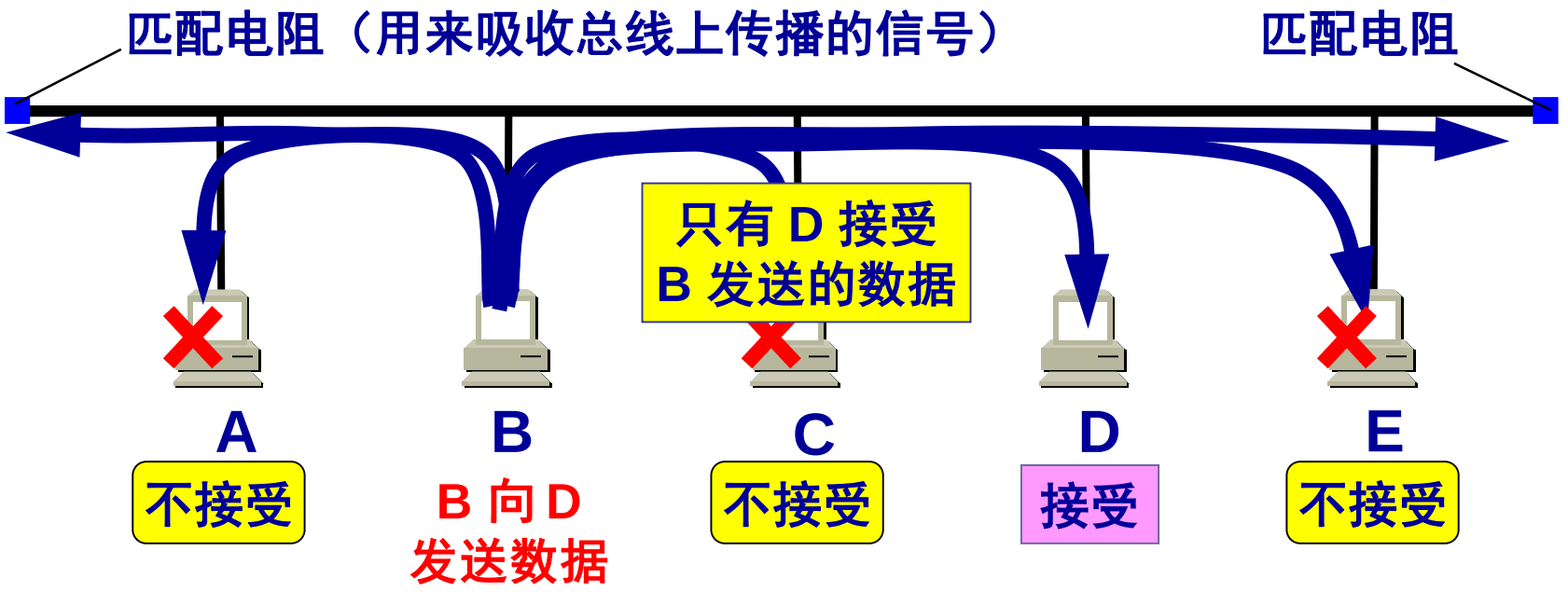
- 网络接口板又称为通信适配器 (adapter) 或网络接口卡 NIC (Network Interface Card), 或“网卡”。
- 适配器的重要功能：
  - 进行串行/并行转换。
  - 对数据进行缓存。
  - 在计算机的操作系统安装设备驱动程序。
  - 实现以太网协议。

计算机通过适配器和局域网进行通信



3.3.2 以太网

- 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。



以太网采用广播方式发送

- 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。
- 在具有广播特性的总线上实现了一对一的通信。

以太网采取了两种重要的措施

为了通信的简便，以太网采取了两种重要的措施：

(1) 采用较为灵活的无连接的工作方式

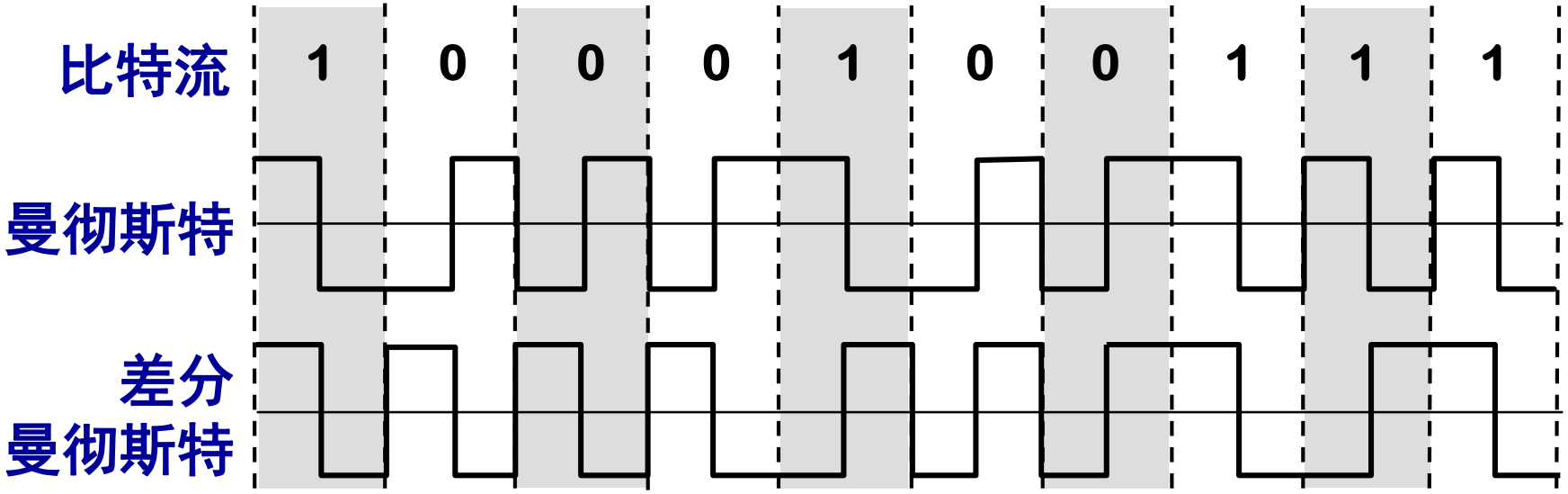
- 不必先建立连接就可以直接发送数据。
- 对发送的数据帧不进行编号，也不要求对方发回确认。

以太网提供的服务

- 以太网提供的服务是不可靠的交付，即尽最大努力的交付。
- 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。

以太网采取了两种重要的措施

(2) 以太网发送的数据都使用曼彻斯特 (Manchester) 编码



曼彻斯特编码缺点是：它所占的频带宽度比原始的基带信号增加了一倍。

CSMA/CD协议

- CSMA/CD 含义：载波监听多点接入 / 碰撞检测 (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。
- “载波监听”就是用电子技术检测总线上有没有其他计算机发送的数据信号。

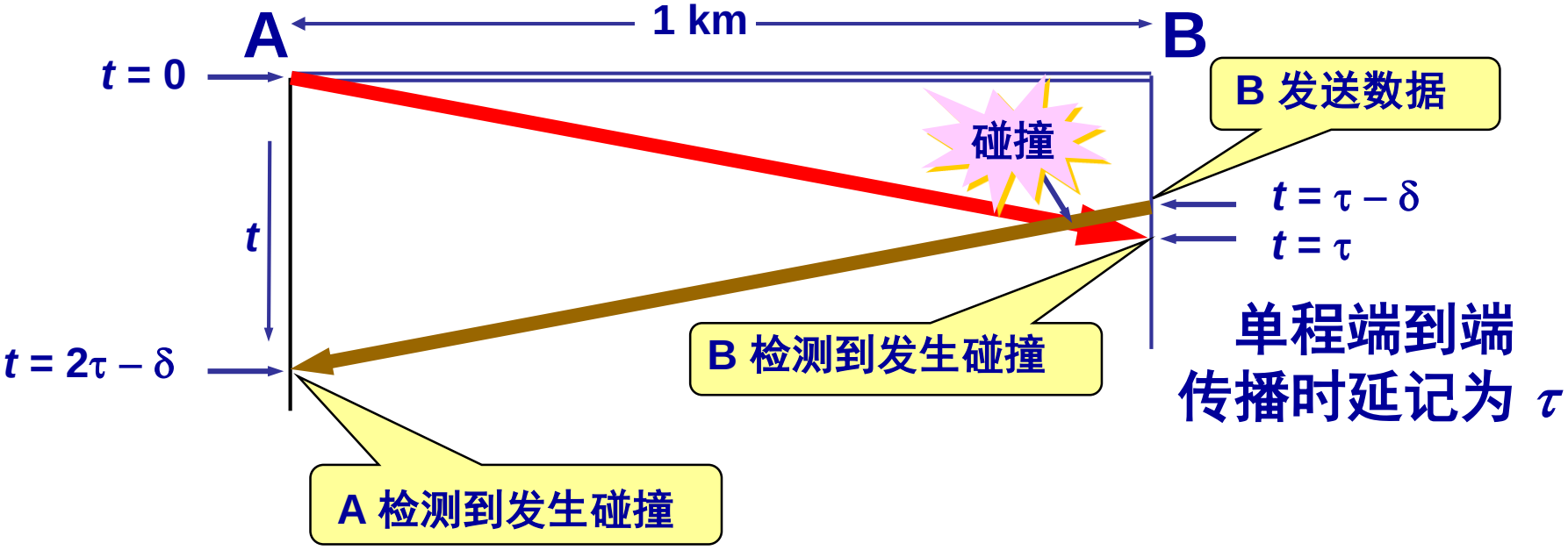
碰撞检测

- “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）。
- 当一个站收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。
- 所谓“碰撞”就是发生了冲突。因此“碰撞检测”也称为“冲突检测”。

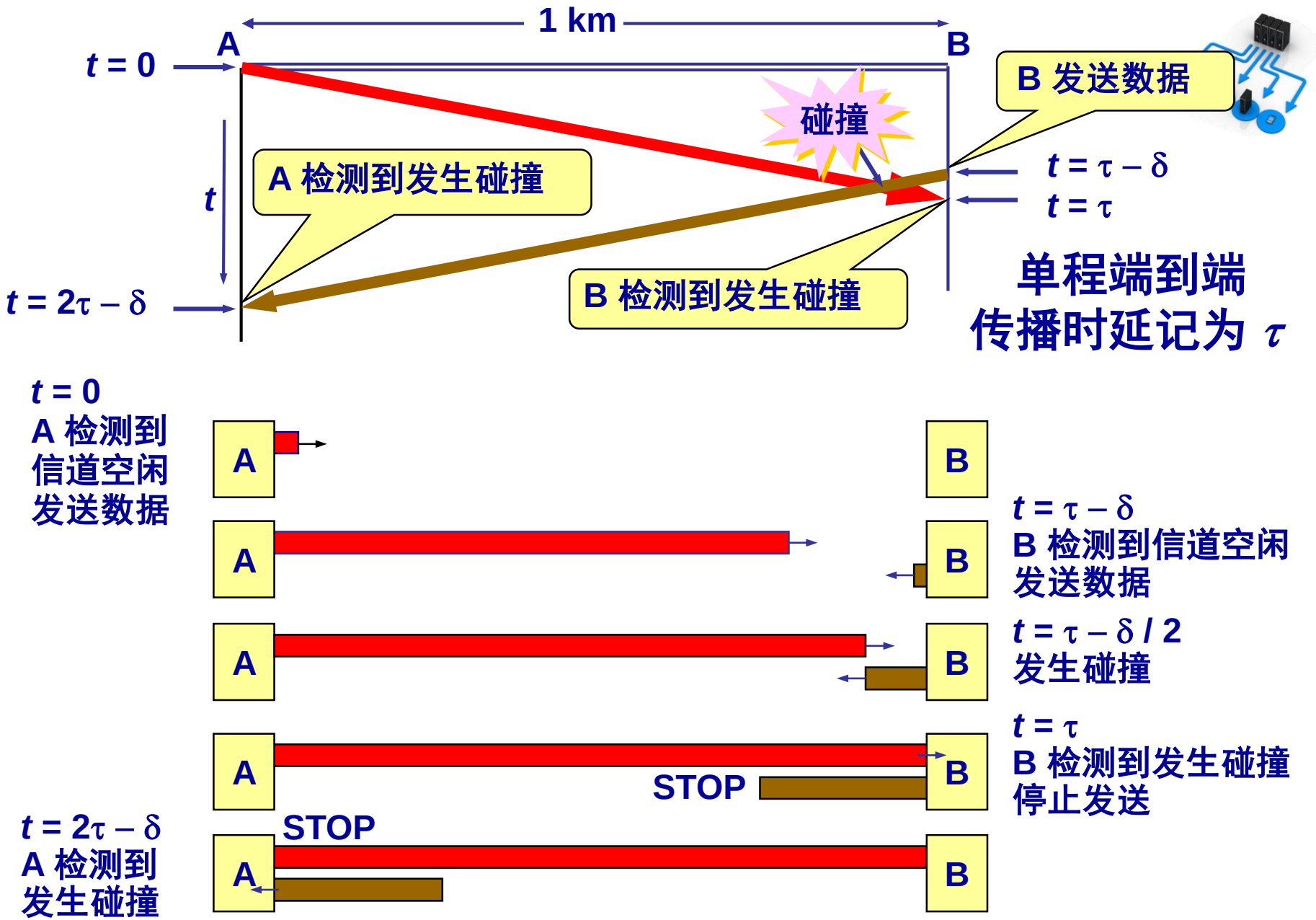
为什么要进行碰撞检测？

- 由于电磁波在总线上的传播速率是有限的，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。（传输有时延）
- 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。
- 所以需要在发送期间进行碰撞检测，以检测冲突。

信号传播时延对载波监听的影响



A需要单程传播时延的 2 倍的时间，
才能检测到与 B 的发送产生了冲突


1 km
A
B
t = 0
t
t = 2τ – δ
碰撞
B 发送数据
A 检测到发生碰撞
B 检测到发生碰撞
t = τ – δ
t = τ
单程端到端
传播时延记为 τ
t = 0
A 检测到
信道空闲
发送数据
A
B
t = τ – δ
B 检测到信道空闲
发送数据
t = τ – δ / 2
发生碰撞
t = τ
B 检测到发生碰撞
停止发送
STOP
STOP
t = 2τ – δ
A 检测到
发生碰撞

CSMA/CD 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。
- 这种发送的不确定性使整个以太网的平均通信量远小于以太网的最高数据率。

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 2τ（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。
- 以太网的端到端往返时延 2τ 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。

争用期的长度

- 10 Mbit/s 以太网取 51.2 μs 为争用期的长度。
- 对于 10 Mbit/s 以太网，在争用期内可发送 512 bit，即 64 字节。

这意味着：
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。

最短有效帧长

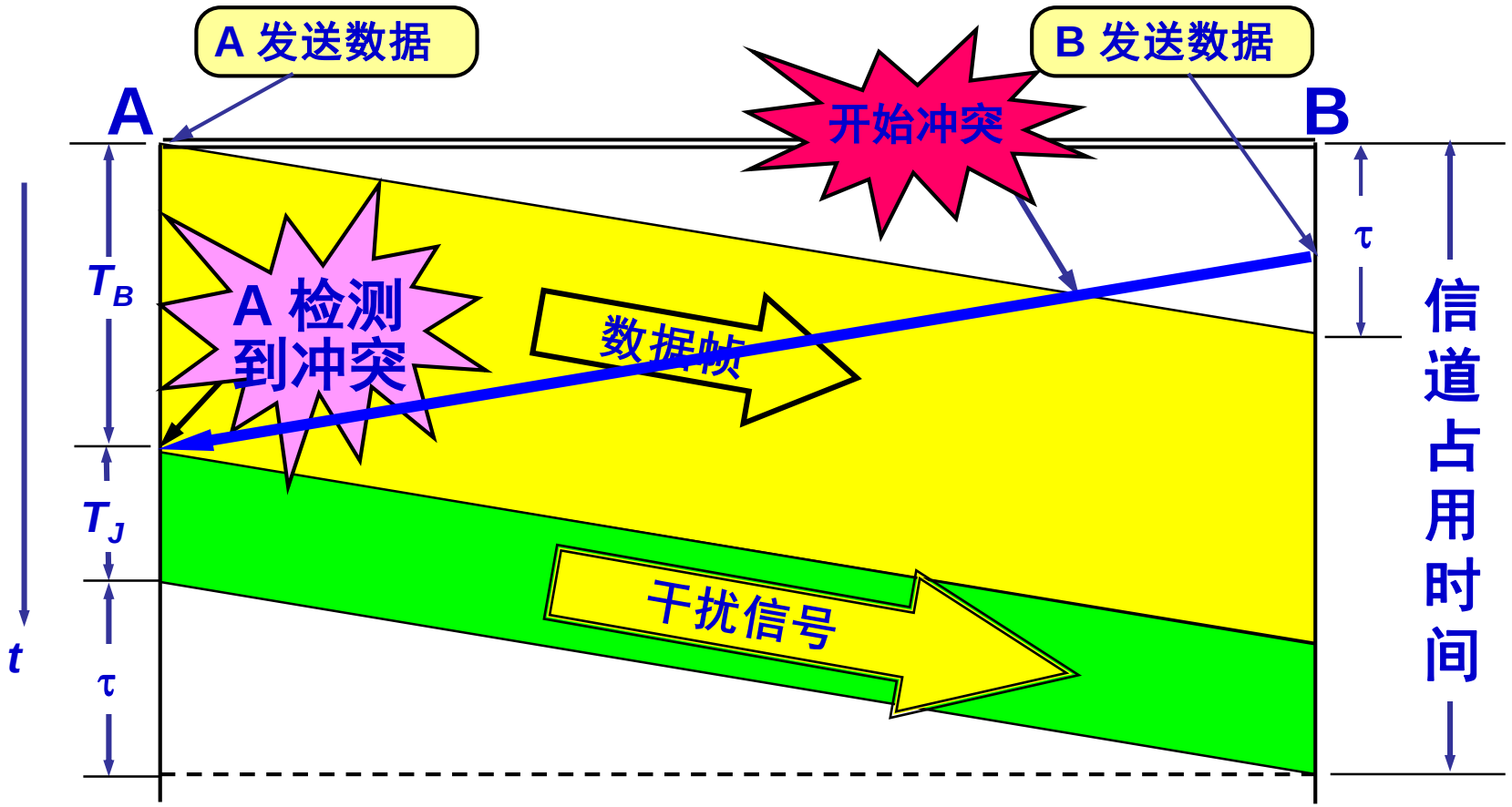
- 如果发生冲突，就一定是在发送的前 64 字节之内。
- 由于一检测到冲突就立即中止发送，这时已经发送出去的数据一定小于 64 字节。
- 以太网规定了最短有效帧长为 64 字节，凡长度小于 64 字节的帧都是由于冲突而异常中止的无效帧。

强化碰撞

当发送数据的站一旦发现发生了碰撞时：

- (1) 立即停止发送数据；
- (2) 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。（32比特或48比特)

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。

二进制指数类型退避算法

- 发生碰撞的站在停止发送数据后，要推迟（退避）一个随机时间才能再发送数据。
  - 基本退避时间取为争用期 2τ。
  - 从整数集合 [0, 1, … , $(2^k - 1)$] 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（$r^* 2\tau$）。
  - 参数 k 按下面的公式计算：

$$k = \text{Min}[\text{重传次数}, 10]$$

  - 当 $k \le 10$ 时，参数 k 等于重传次数。
  - 当重传达 16 次仍不能成功时即丢弃该帧，并向高层报告。

冲突后的等待时间

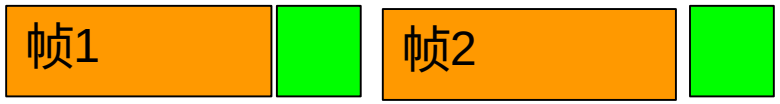
- 二进制指数后退算法
  - 定基本退避时间取为争用期 2τ，为51.2μs
  - 退避时间　r* 2τ
  - 如果第一次冲突，则r在0，1两个值中随机选择。
  - 如果第二次冲突，则r在0～3 中随机选择；
  - （如r选2 ，则等待 2* 2τ 后重传 ）
  - 如果第i次冲突，则r 在0～2^i-1 中随机选择
  - 次数如果大于10次，则r在 0~1023 中随机选择
  - 次数如果大于16次，放弃传输并向上层提交错误报告

CSMA/CD协议的要点

- (1) 准备发送。但在发送之前，必须先检测信道。
- (2) 检测信道。若检测到信道忙，则应不停地检测，一直等待信道转为空闲。若检测到信道空闲，并在 96 比特时间内信道保持空闲（保证了帧间最小间隔），就发送这个帧。
- (3) 检查碰撞。在发送过程中仍不停地检测信道，即网络适配器要边发送边监听。这里只有两种可能性：
  - ①发送成功：在争用期内一直未检测到碰撞。这个帧肯定能够发送成功。发送完毕后，其他什么也不做。然后回到 (1)。
  - ②发送失败：在争用期内检测到碰撞。这时立即停止发送数据，并按规定发送人为干扰信号。适配器接着就执行指数退避算法，等待 *r* 倍 512 比特时间后，返回到步骤 (2)，继续检测信道。但若重传达 16 次仍不能成功，则停止重传而向上报错。

帧间最小间隔

- 帧间最小间隔为 9.6 μs，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 μs 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。



练习题

1.在二进制退避算法中，如果发生了11次的碰撞，那么站点会在0至（　　）中选择一个随机数r。

2.如果发生了16次碰撞，如何处理？

3. 10Mbit/s 的发送速率，发送1比特 需要多长时间？

The end