第四章 数据链路层

4.1 线路规程

4.2 流量控制与差错控制

4.3 HDLC通信协议

4.4 IEEE局域网通信协议

4.5 以太网

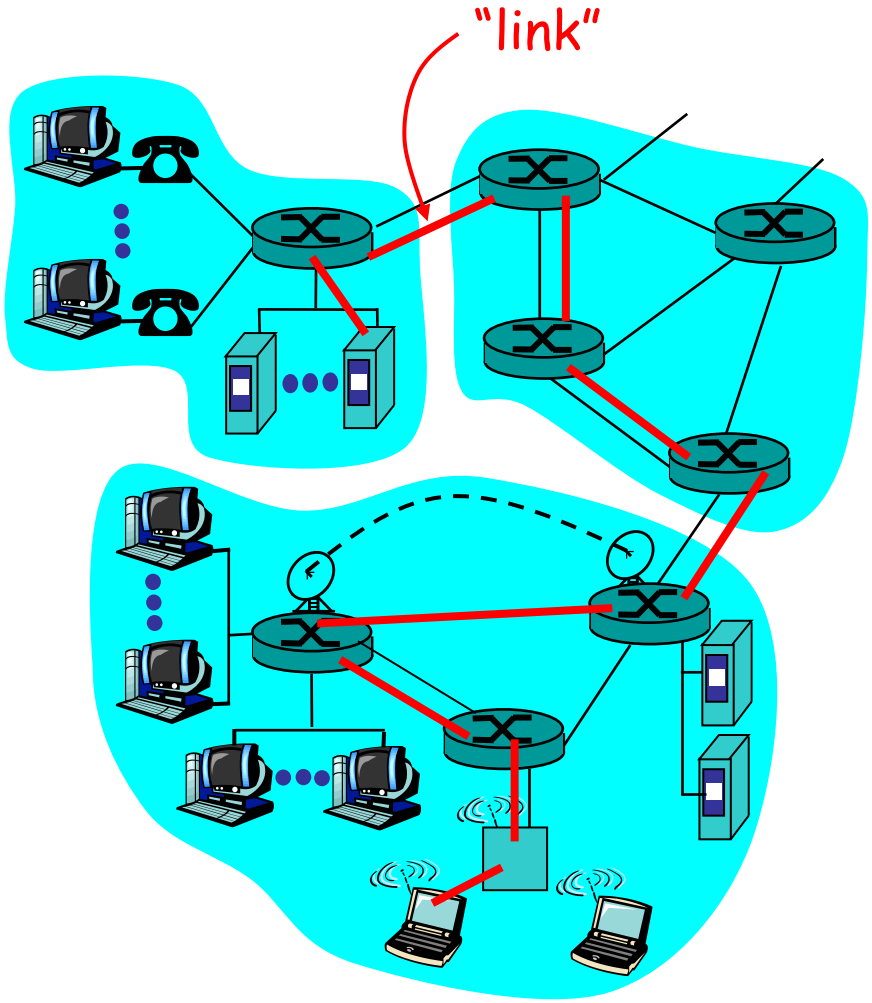
4.6 无线局域网

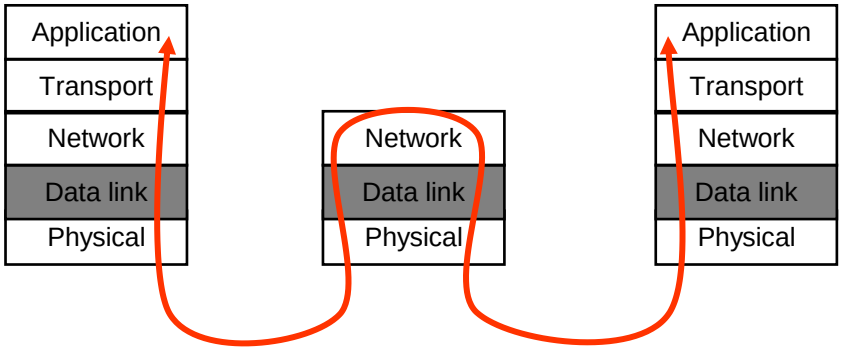
4.7 数据链路层网络互连

数据链路层(Data-Link)

- 数据链路层处理相邻节点的数据传输
- 将不可靠的物理链路变为可靠的数据链路
- 传输的数据单元是帧（Frame）
- 数据链路层的主要工作：
  - 线路规程： 分帧、排序
  - 差错控制： 为上层提供可靠链路
  - 流量控制： 处理输入数据的速率
  - 链路管理： 链路的建立， 维持， 拆除

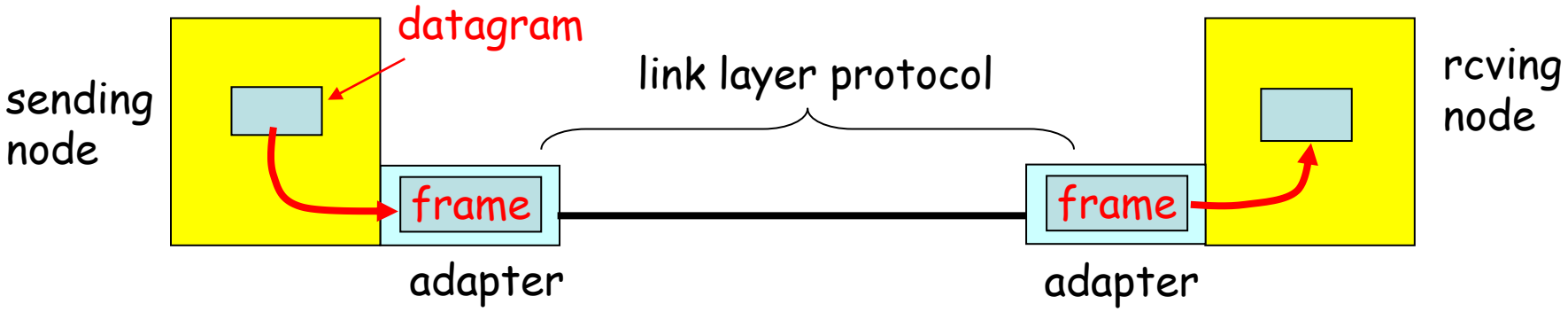
Link Layer





- Datagram transferred by different link protocols over different links:
  - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- Each link protocol provides different services
  - e.g., may or may not provide rdt over link

Adaptors Communicating



- link layer implemented in "adaptor" (aka NIC)
  - Ethernet card, PCMCIA card, 802.11 card
- adapter is semi-autonomous

- sending side:
  - encapsulates datagram in a frame
  - adds error checking bits, rdt, flow control, etc.
- receiving side
  - looks for errors, flow control, etc
  - extracts datagram, passes to rcving node

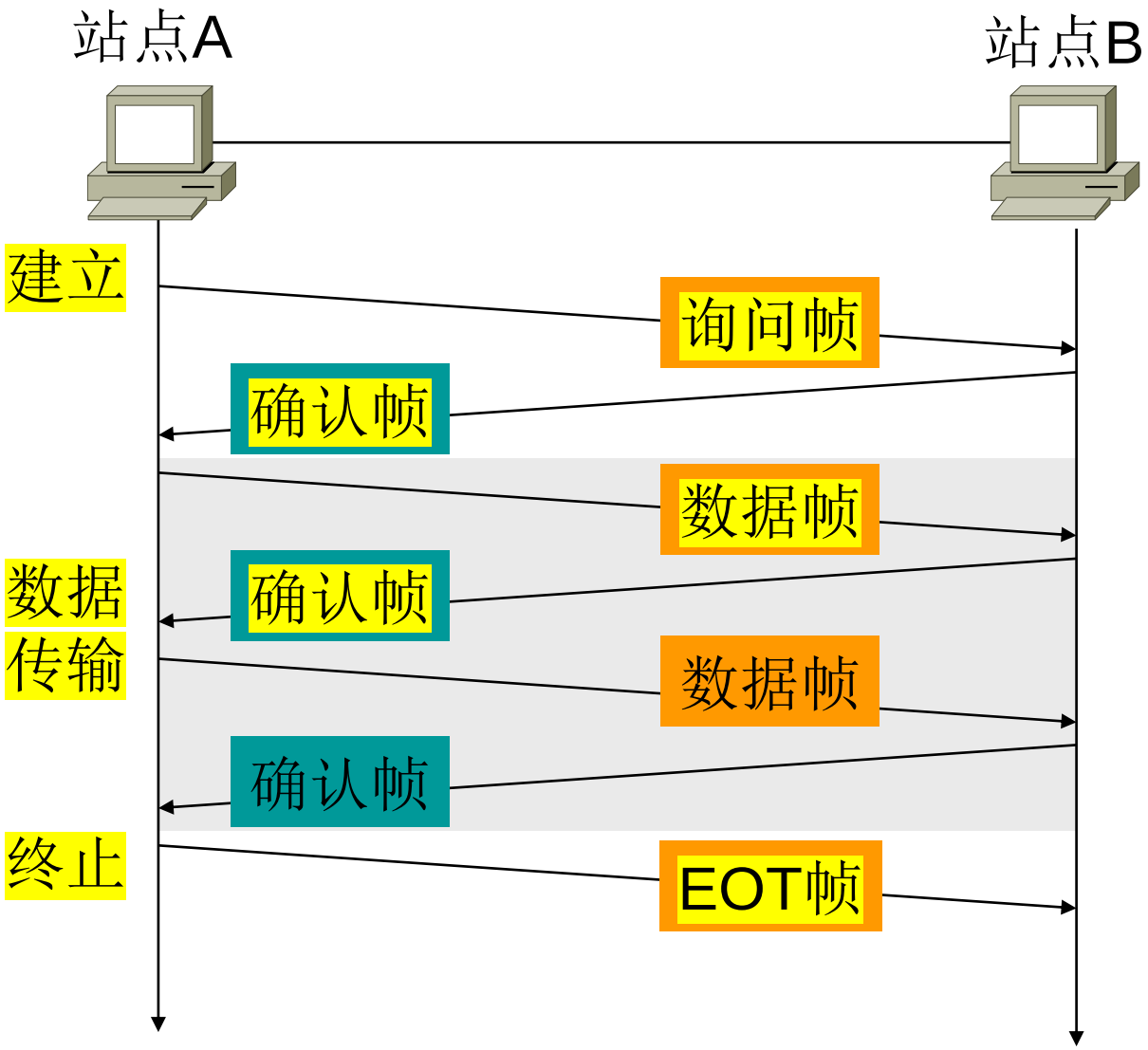
4.1 线路规程

- 线路规程是监视链路的建立，以及在给定时刻分配一个具体设备进行数据传送的权利。
- 线路规程可以两种方式实现：
  - 询问/确认方式（ENQ/ACK）
  - 轮询/选择方式（Poll/Select）

4.1.1 询问/确认模式

- 使用场合
  - 当两个设备之间存在一条专用链路时，即点对点方式，采用询问/确认模式。只要一条链路两头的设备级别相同，任意一个设备都可以启动一个会话过程。
- 工作方式
  - 启动方首先发送一个询问帧（ENQ）询问接收方是否可以接收数据
    - 接收方如果已经准备好接收，回答一个确认帧（ACK）
    - 如果没有准备好接收，回答一个否认帧（NAK）

ENQ/ACK(询问/确认)线路规程

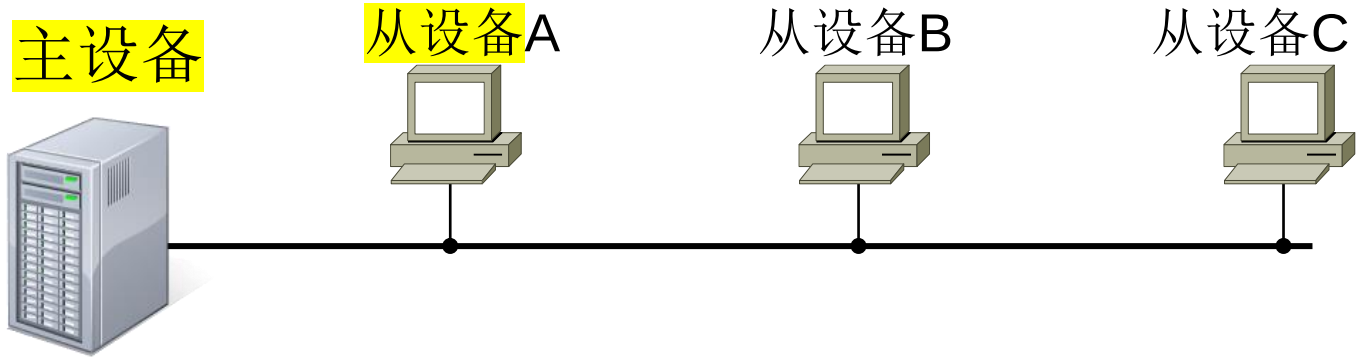


3种可能情况：

- 没有应答
- 回答是否定的
- 回答是肯定的

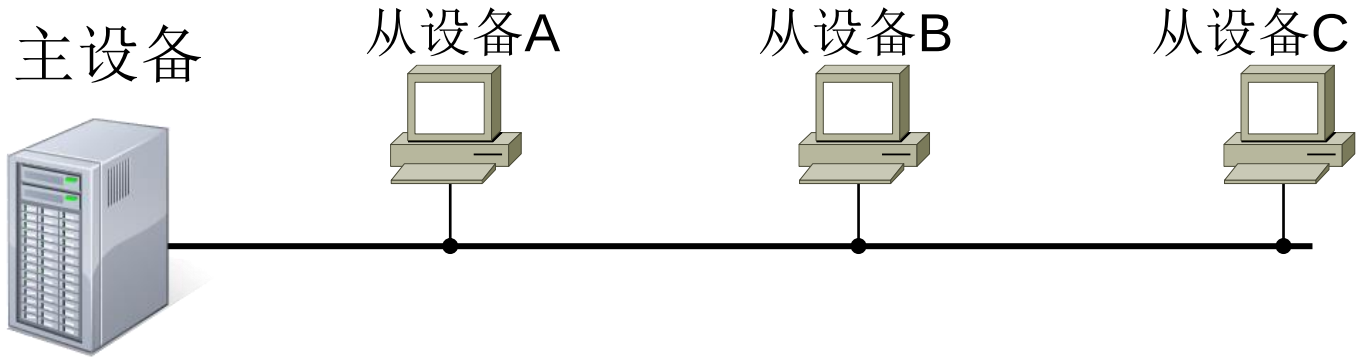
4.1.2 轮询/选择(Poll/Select)模式

- 使用场合：
  - 应用在多点连接系统中。
  - 在这种环境下，不仅仅要确定设备是否就绪，还要确定哪一个站点有权使用信道。



轮询/选择模式

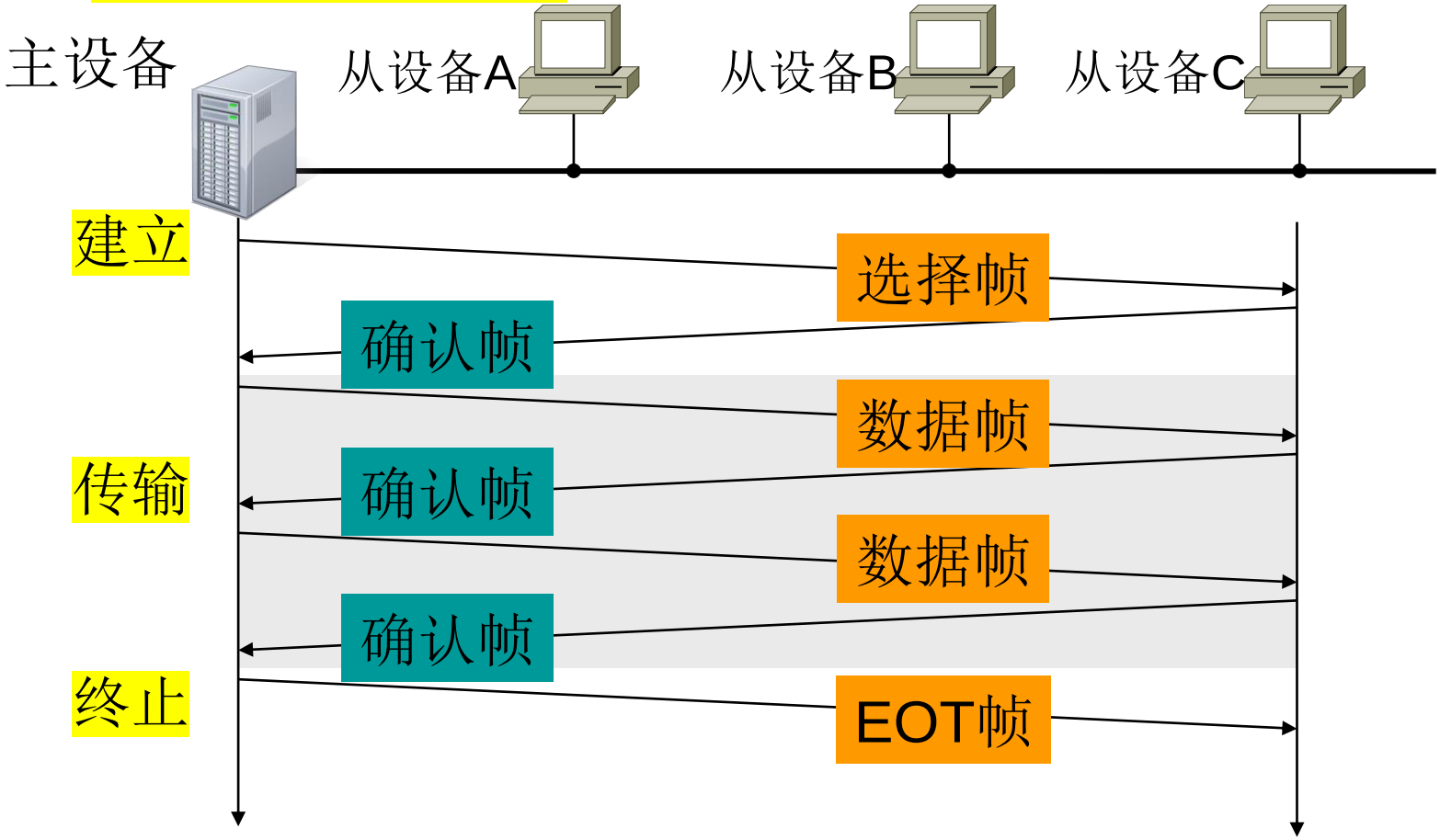
- 工作方式：
  - 主设备控制链路，主设备发命令，从设备响应。
  - 轮询：如果主设备希望接收数据
  - 选择：如果主设备希望发送数据



- 地址问题：
  - 在链路上的每个设备都有一个地址来标识自己。

选择

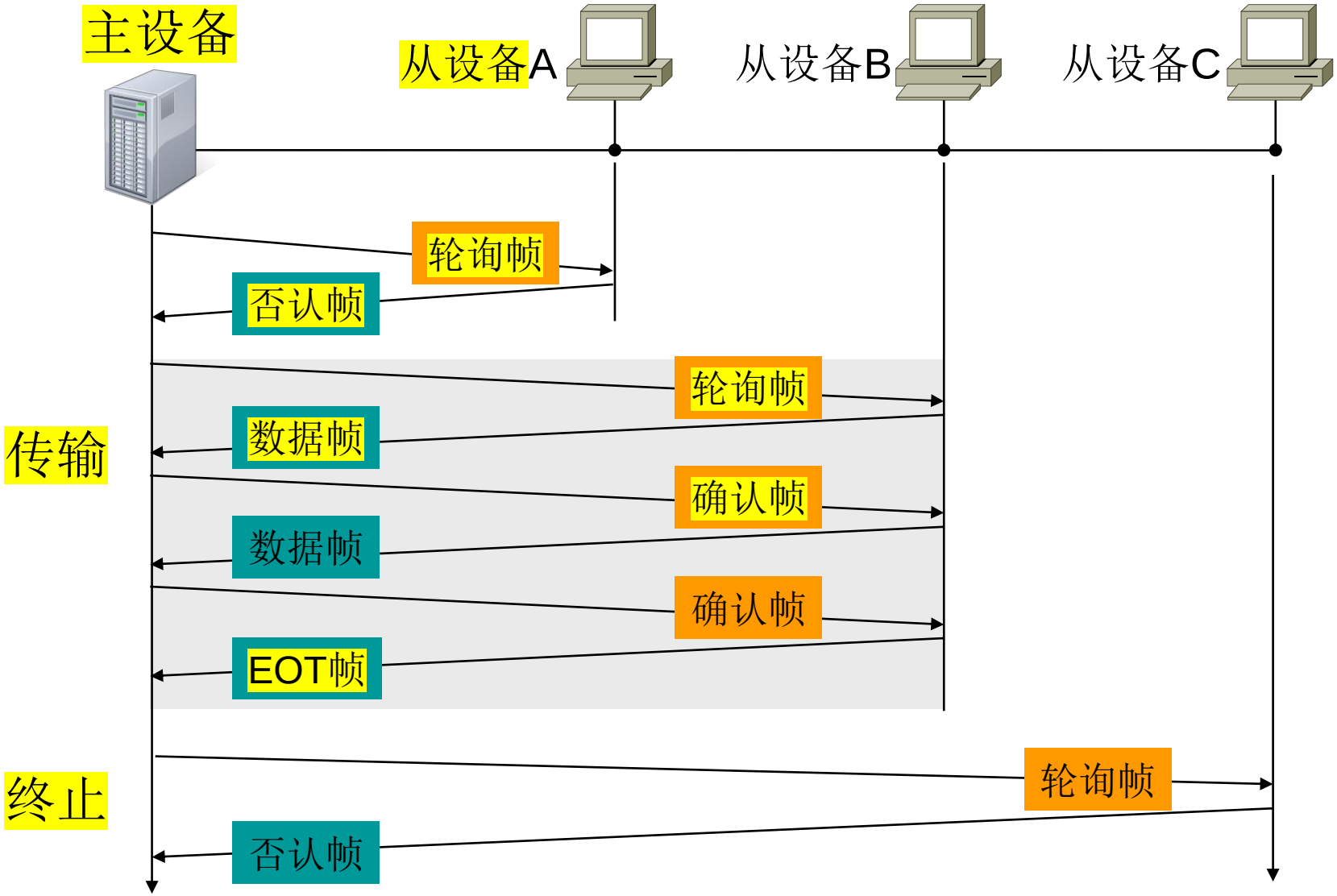
- 选择：主设备希望发送数据，用SEL告诉从设备准备接收数据。从设备用ACK同意接收，用NAK拒绝接收。



轮询

- 轮询：主设备希望接收数据，逐个询问设备是否有数据发送。
  - 如果没有数据发送，必须回答否定应答帧(NAK)。
  - 如果有数据，直接用数据帧应答。

轮询


主设备
从设备A
从设备B
从设备C
轮询帧
否认帧
传输
轮询帧
数据帧
确认帧
数据帧
确认帧
EOT帧
终止
轮询帧
否认帧

轮询/选择模式

- 两种终止信息交互的方法：
  - 从设备将所有的数据发送完毕，并以一个传输结束帧(EOT)结束传输。
  - 主设备发出“时间到”消息。

4.2 流量控制与差错控制

- 流量控制是一组过程，这组过程是用来告诉发送方在等待接收方的确认信号之前最多可以传送多少数据。
- 流量控制的两个要点：
  - 数据流不能使接收方过载
  - 接收方对数据进行确认

差错控制

- 差错控制主要指错误检测和重传方法。
- 自动重复请求(ARQ)：数据帧在传输中出现错误，接收方就返回一个否定应答帧(NAK)，出错的帧就会被发送方重新传送。这个过程叫自动重复请求ARQ。
- 数据被重传的情况有三种：
  - 帧破坏
  - 帧丢失
  - 应答帧丢失

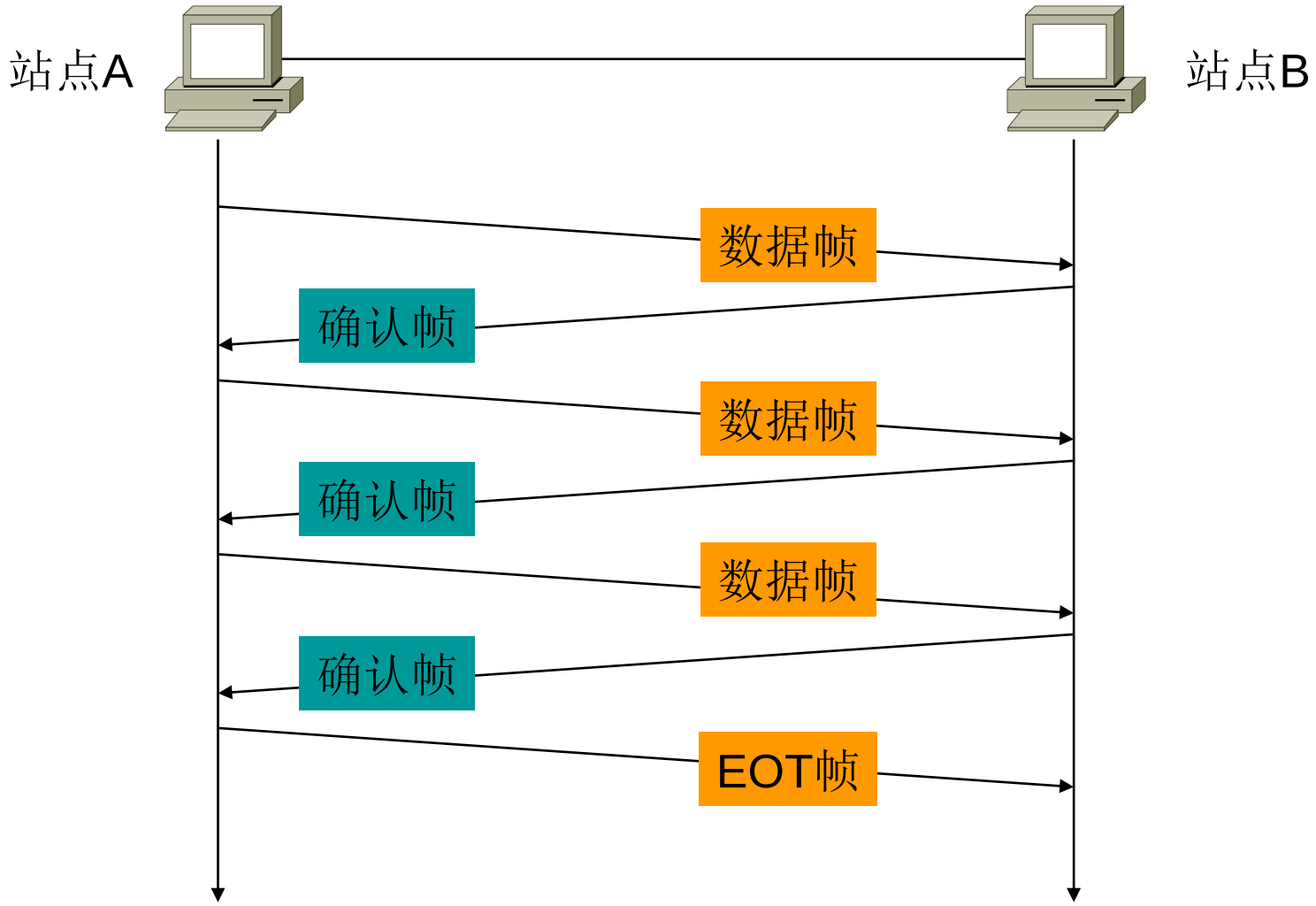
实现技术

- 流量控制和差错控制是结合在一起实现的
- 两种实现流量控制和差错控制的技术：
  - 停止等待协议
  - 滑动窗口协议

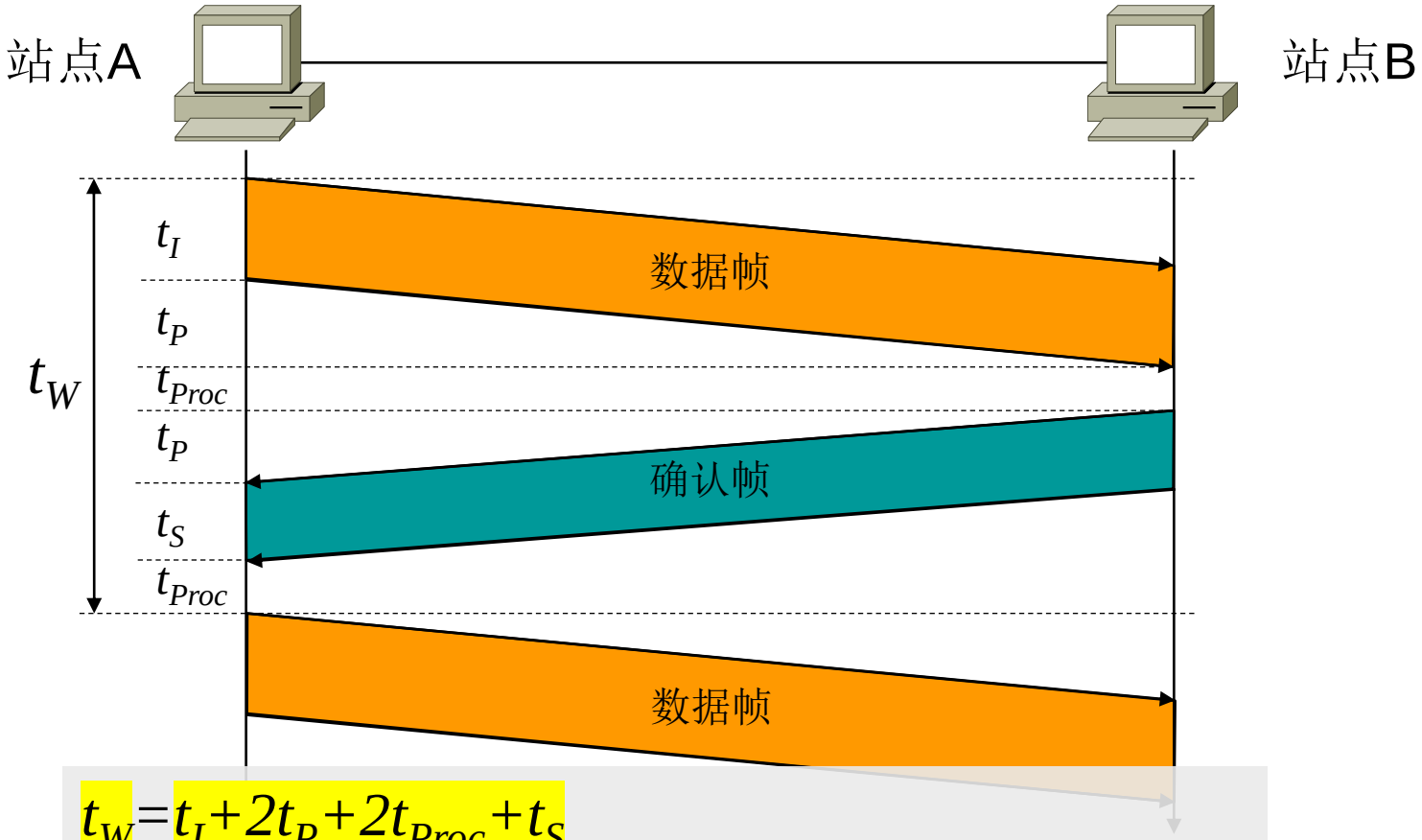
4.2.1 停止等待协议

- 停止等待协议的流量控制
  - 发送方每发送一帧后就等待应答。只有收到一个应答(ACK)后，才发送下一个帧。直到发送方发送一个传输结束帧。
  - 优点：协议简单。
  - 缺点：效率低，在线路上只有一帧。如果设备之间的距离很长，在每帧之间等待ACK帧所花费的时间很长。

停止等待协议



完成一帧发送所需的最短时间



$t_W = t_I + 2t_P + 2t_{Proc} + t_S$

t_I: 发送数据帧时间=帧长/数据传输率

t_S: 发送确认帧时间=确认帧长/数据传输率

t_P: 信号传输延时=距离/信号传输速度

t_{Proc}: 收到一个帧的处理时间和一个帧的形成时间

停等协议的定量分析(无差错)

- 信道利用率：信道被占用的时间和总时间之比。
- 有效数据传输率：单位时间内传输的有效数据位数。
- 无差错情况的信道利用率P：

$$P=t_I/t_W$$

t_I：发送数据的时间

t_W：发送一帧的时间；

- 无差错情况的有效数据传输率

$$S=N/t_W$$

N：有效数据位数；

t_W：发送一帧的时间；

例：

C = 传输速率（10Mbps或10bit/μs）

S = 信号速度（200m/μs）d

D = 发送方与接收方的距离（200m）

t_{Proc} = 生成一帧的时间（1μs）

L_f = 一帧的比特数（200bit）

N = 一帧的数据比特数（160bit）

L_S = 一确认帧的比特数（40bit）

解：

$t_W = t_I + 2t_P + 2t_{Proc} + t_S$

$t_I = L_f / C = 200/10 = 20(\mu s)$

$t_S = L_S / C = 40/10 = 4(\mu s)$

$t_P = D/S = 200/200 = 1(\mu s)$

$t_W = 20 + 2 \times 1 + 2 \times 1 + 4 = 28(\mu s)$

在 t_W 时间内，发送信道被占用的时间为 $t_I = 20(\mu s)$

所以：

信道利用率：$P = t_I / t_W = 21/28 = 75\%$

有效数据传送速率：$S = N / t_W = 160/28 = 5.7Mbps$

停等协议的定量分析(有差错)

- 有差错时正确传送一帧的平均时间
  - 无差错情况下，发送一帧的最小时间间隔为t_W
  - 当出错率为p时，正确发送一帧的平均时间间隔t_V为（根据概率统计学）：

$$t_V=t_W/(1-p)$$

p：出错率

t_V：发送一帧的平均时间

t_W：发送一帧的时间(无差错情况下)

停等协议的定量分析(有差错)

- 系统最大吞吐量λ_{max}（每秒成功发送的帧数）

$$\lambda_{max} = 1/t_V = (1-p)/t_W$$

- 极限吞吐量$M = 1/t_I$（t_I：发送数据的时间）
- 系统传输效率：最大吞吐量/极限吞吐量

$$\rho = \lambda_{max}/M$$
$$=[(1-p)/t_W]/(1/t_I)$$
$$=(1-p)/(t_W/t_I)$$

令$a \equiv t_W/t_I$，则：

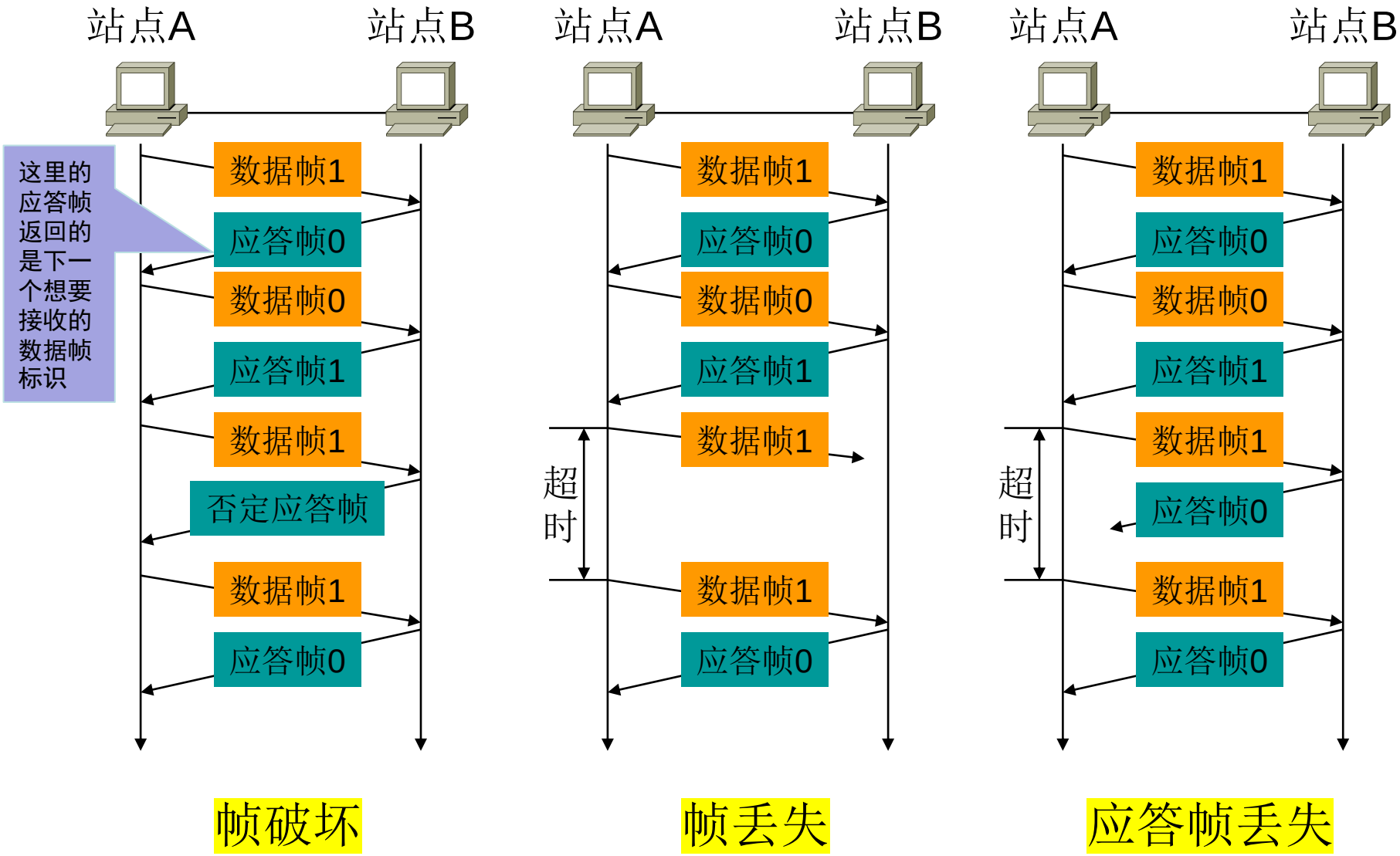
$$\rho = (1-p)/a$$

- ρ与a成反比，a越小效率越高。

停止等待协议中的差错控制

- 为了实现差错控制，停止等待协议采用ARQ(Auto Repeat reQuest)技术， ARQ处理错误的三种情况：
  - 帧破坏
  - 帧丢失
  - 应答帧丢失

ARQ处理错误的三种情况



差错控制的要求

- 发送站要保留数据帧的备份。
- 数据帧和应答帧必须交替的标识为0和1。如果接收方收到了两个相邻的数据帧且标号相同，说明接收方收到了一个重复帧。应当丢弃一个重复帧。
- 否定应答帧(NAK)，通知发送方重新发送最近的一帧。
- 定时器，判断数据帧在传输中丢失。

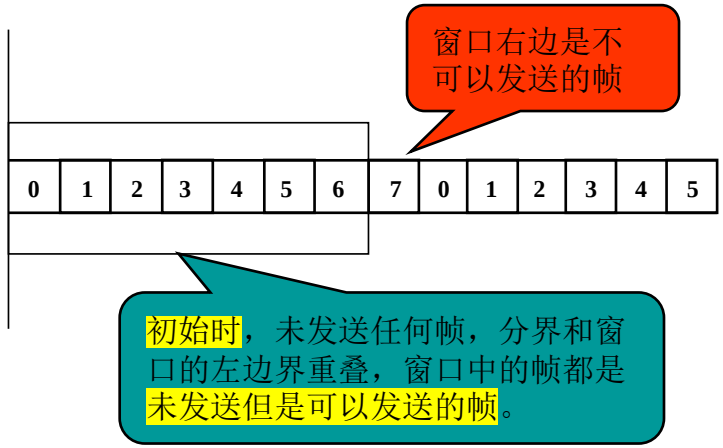
4.2.2 滑动窗口协议

- 流量控制
  - 发送方在收到应答消息前可以发送若干帧。
  - 接收方使用一个ACK帧来对多个数据帧的接收进行确认。具体可以接受多少帧视窗口大小而定。
  - 帧编号：
    - 在滑动窗口协议中，数据帧以模n方式编号，也就是说，编号从0到n-1。
    - 窗口的大小是n-1。
    - 接收方发送的应答帧(ACK)编号是接收方希望收到的下一帧的编号。

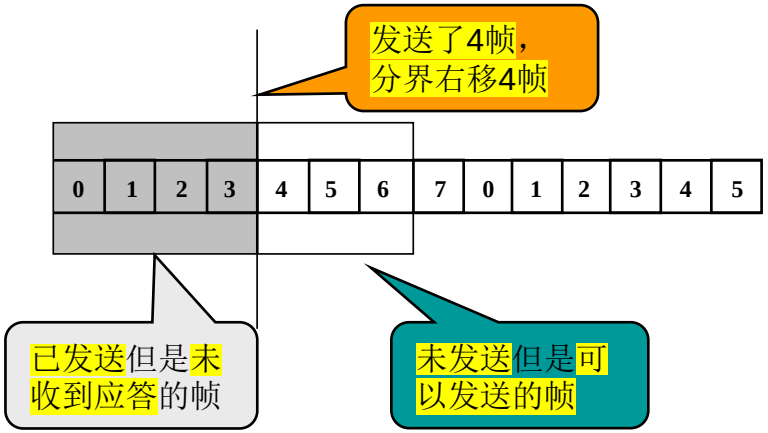
滑动窗口协议中的流量控制

- 窗口是发送方和接收方存放数据帧的缓冲区
- 发送方窗口用于存放已经发送但未收到应答的数据帧和在收到应答帧之前可以发送的数据帧。
- 接收方窗口用于存放已经被接收但未给应答的数据帧。
- 在接收方，只要窗口未填满就可以在未发送应答帧的情况下继续接收数据帧。

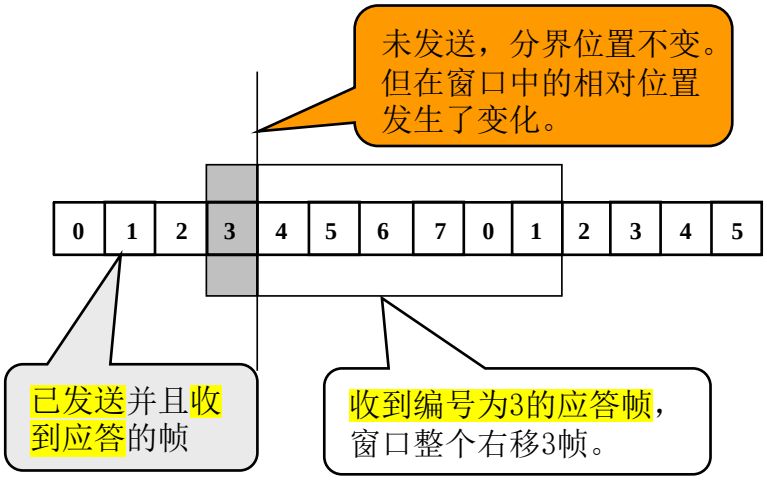
发送方发送窗口



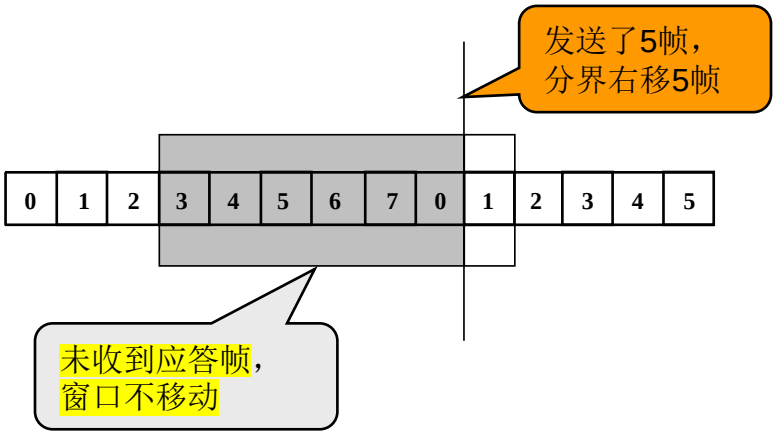
(a)



(b)

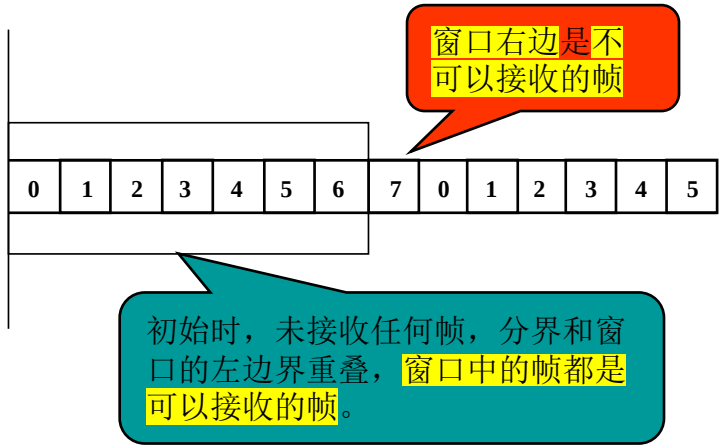


(c)

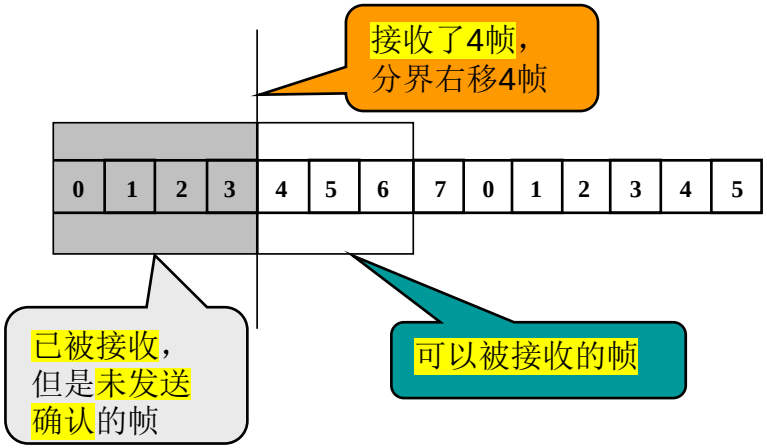


(d)

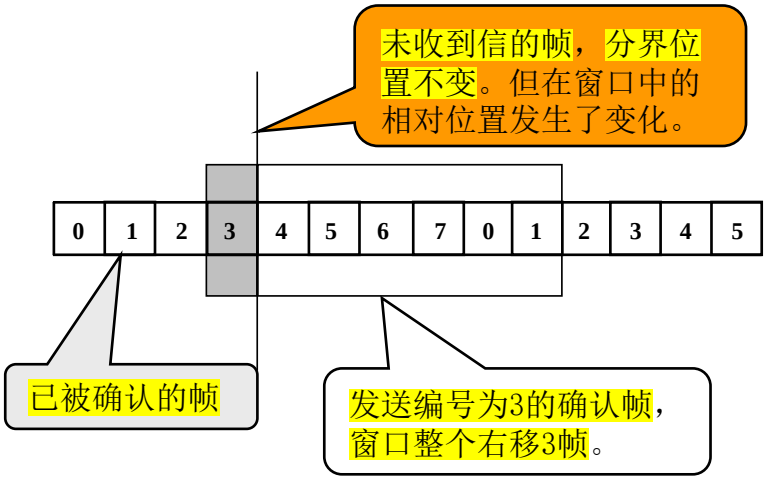
接收方发送窗口



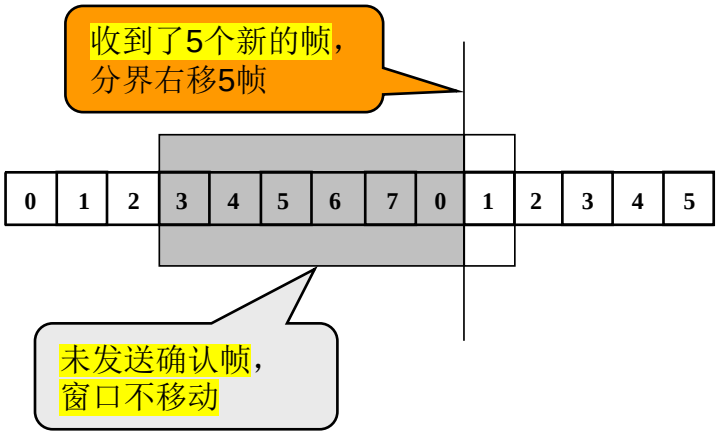
(a)



(b)



(c)



(d)

滑动窗口协议中的差错控制

- 滑动窗口协议中，有两种实现自动重传请求(ARQ)技术：
  - 回退N自动重传请求(Go-back-N)
  - 选择拒绝自动重传请求(Select-Rej)
- 要求：
  - 发送站要保留数据帧的备份；
  - 除确认帧外，接收方可以发送否认帧，告诉发送方重新发送一个损坏的帧；
  - 确认帧、否认帧必须有编号；
  - 定时器，判断数据帧在传输中丢失。

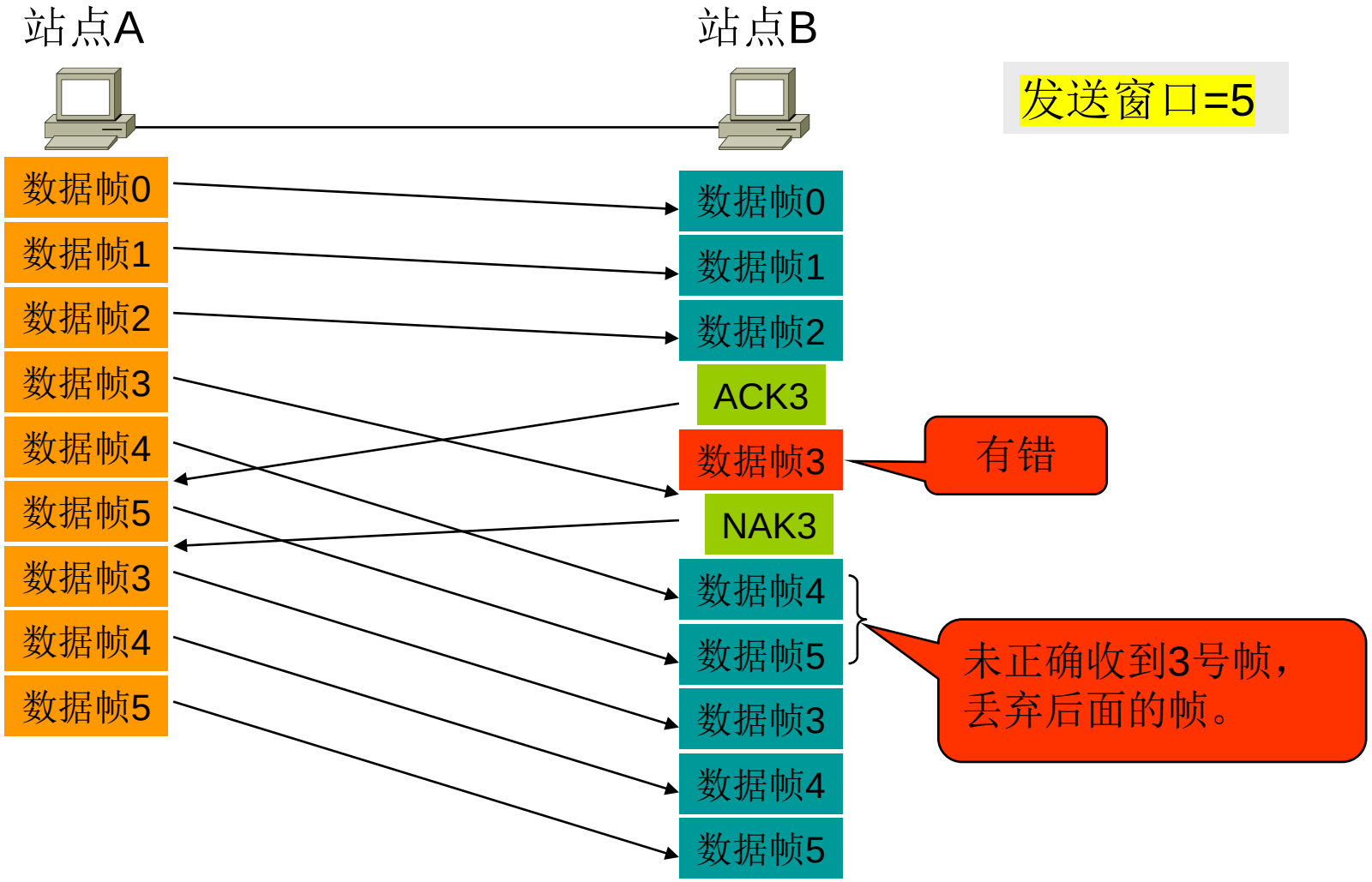
应答帧(ACK)帧的编号是下一次希望收到的帧的编号

应答帧(ACK)帧的编号是正确接收的帧的编号

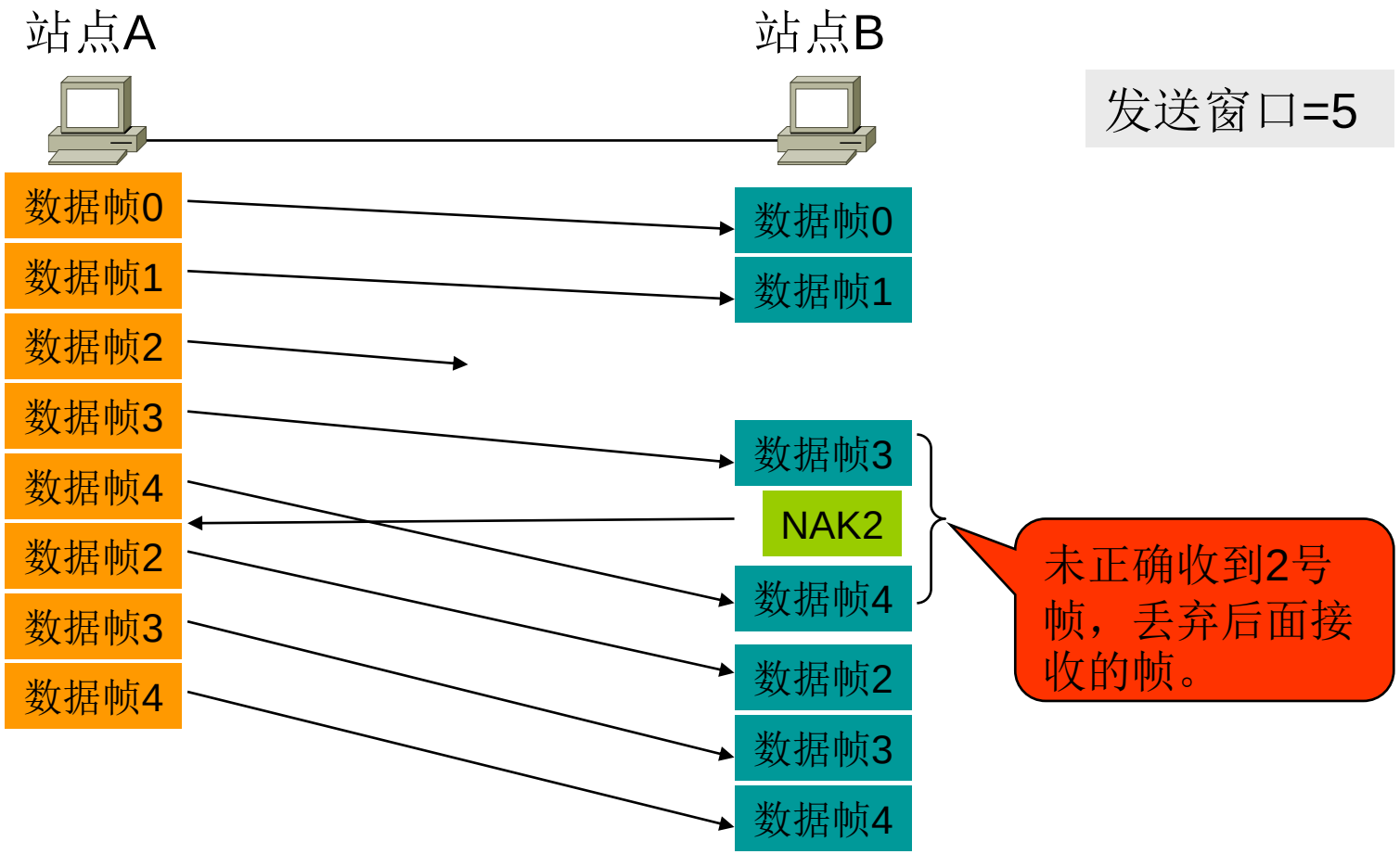
回退n自动重传请求

- 在滑动窗口的回退 n 自动重传请求中，如果一帧丢失或损坏了，从最近一次得到应答的数据帧开始，未被应答的所有帧都进行重传。
- 差错情况有三种：
  - 帧破坏
  - 数据帧丢失
  - 确认帧丢失

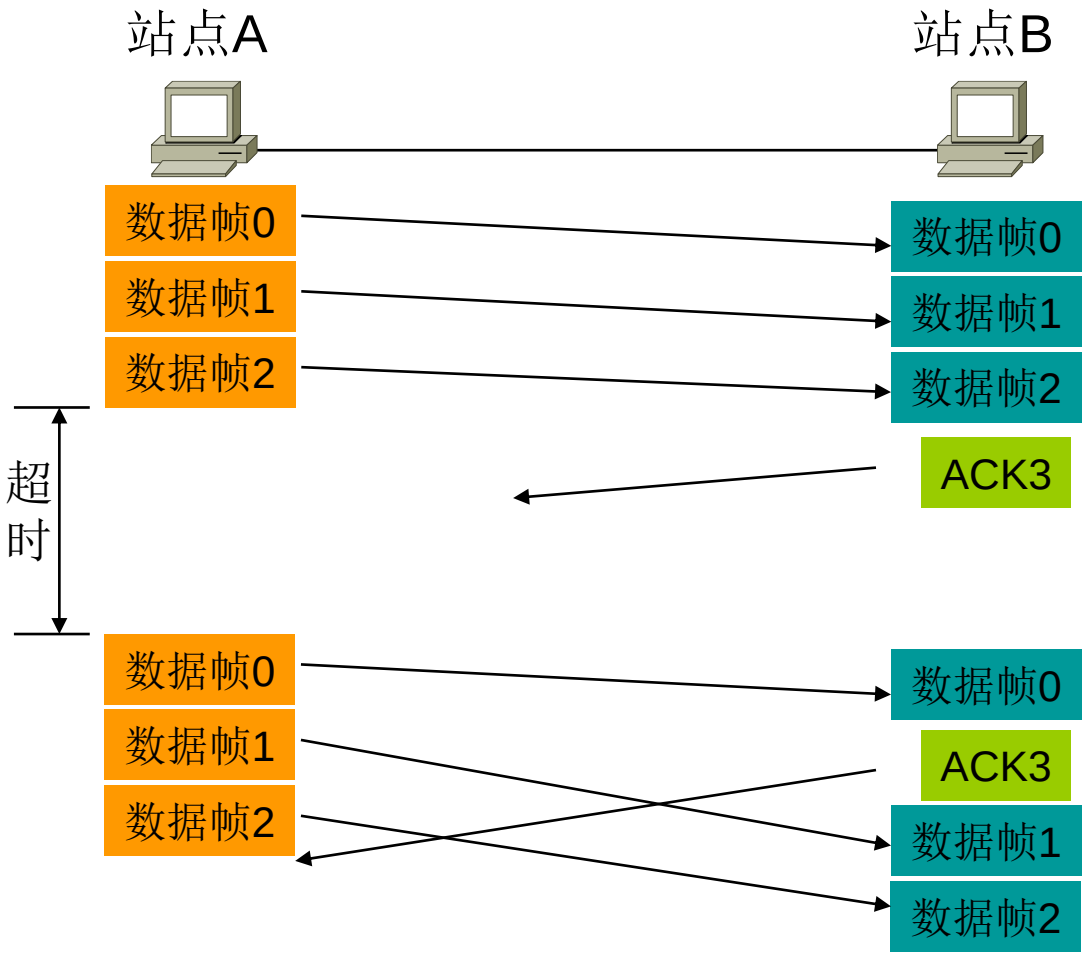
帧破坏


站点A
站点B
发送窗口=5
数据帧0
数据帧1
数据帧2
数据帧3
数据帧4
数据帧5
数据帧3
数据帧4
数据帧5
数据帧0
数据帧1
数据帧2
ACK3
数据帧3
NAK3
数据帧4
数据帧5
数据帧3
数据帧4
数据帧5
有错
未正确收到3号帧，
丢弃后面的帧。

数据帧丢失


站点A
站点B
发送窗口=5
数据帧0
数据帧1
数据帧2
数据帧3
数据帧4
数据帧2
数据帧3
数据帧4
数据帧0
数据帧1
数据帧3
NAK2
数据帧4
数据帧2
数据帧3
数据帧4
未正确收到2号帧，丢弃后面接收的帧。

确认帧丢失

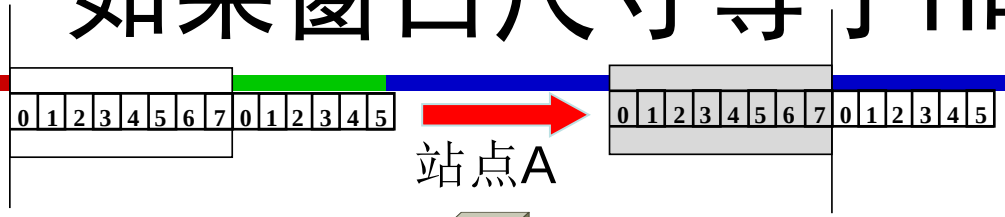


定时器启动：当发送窗口满，或者没有数据要发送了。

窗口的大小与编号范围的关系

- 在回退N协议中，如果帧的编号范围是0～n-1，则窗口的尺寸为n-1。为什么？
- 4种可能的情况：
  - 如果窗口的尺寸>n
  - 如果窗口的尺寸=n
  - 如果窗口的尺寸=n-1
  - 如果窗口的尺寸<n-1

如果窗口尺寸等于n时，协议失败



站点A　　站点B

发送数据帧0~7　t_1

t_2

按顺序收到数据帧0~7，发送ACK0

超时

ACK0丢失

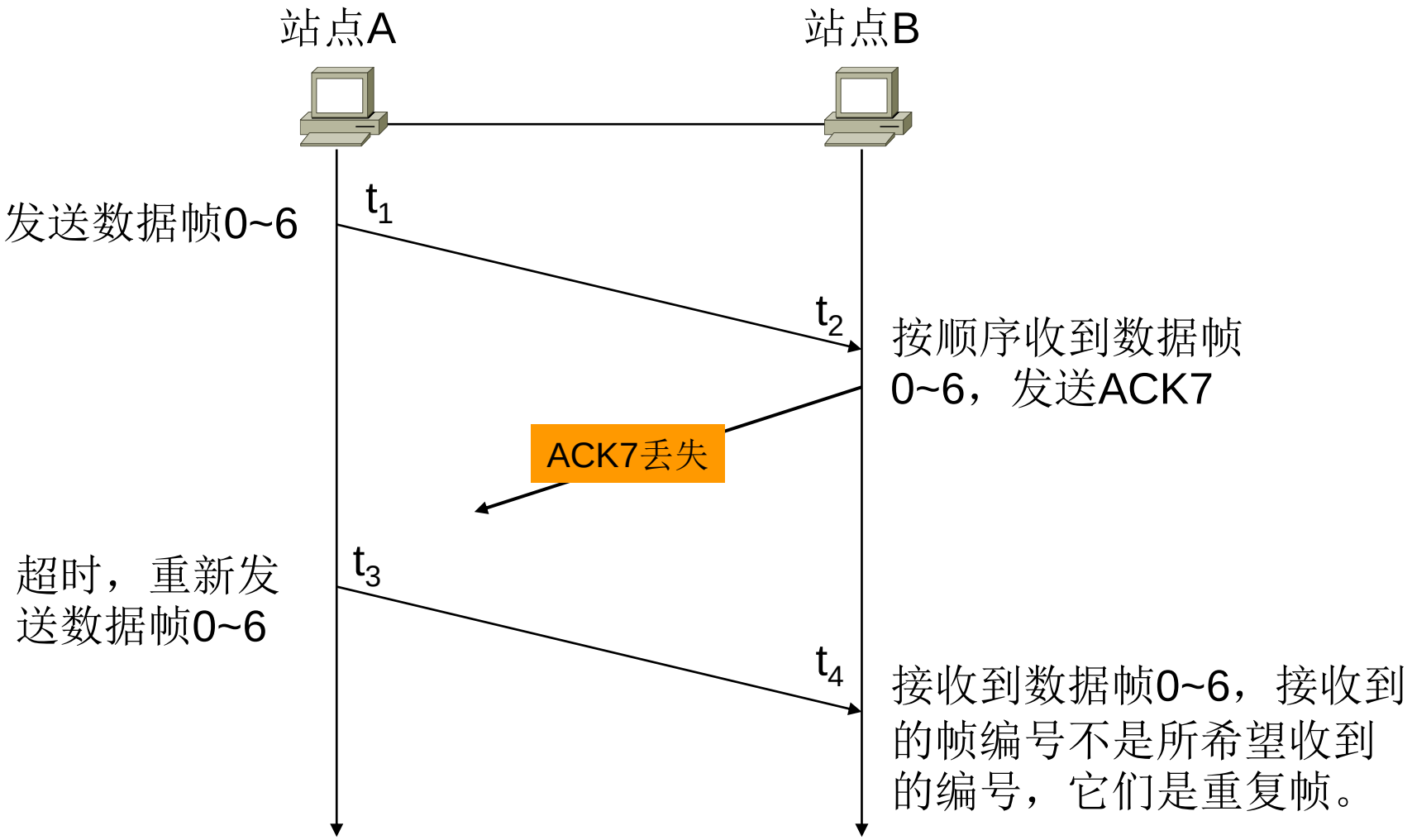
接收方收到了0~7帧，接下来想收到新的数据帧0，但ACK0这个确认帧丢失。
发送方因为没有收到确认帧ACK0，所以窗口不划动并开始从0开始重传旧帧，但旧的帧又被接收方误认成了新帧，发生错误

重新发送数据帧0~7　t_3

t_4

接收到数据帧0~7，接收到的帧编号正是所希望收到的编号，认为它们是新帧，但是它们是重复帧。

如果窗口尺寸等于n-1时，协议成功



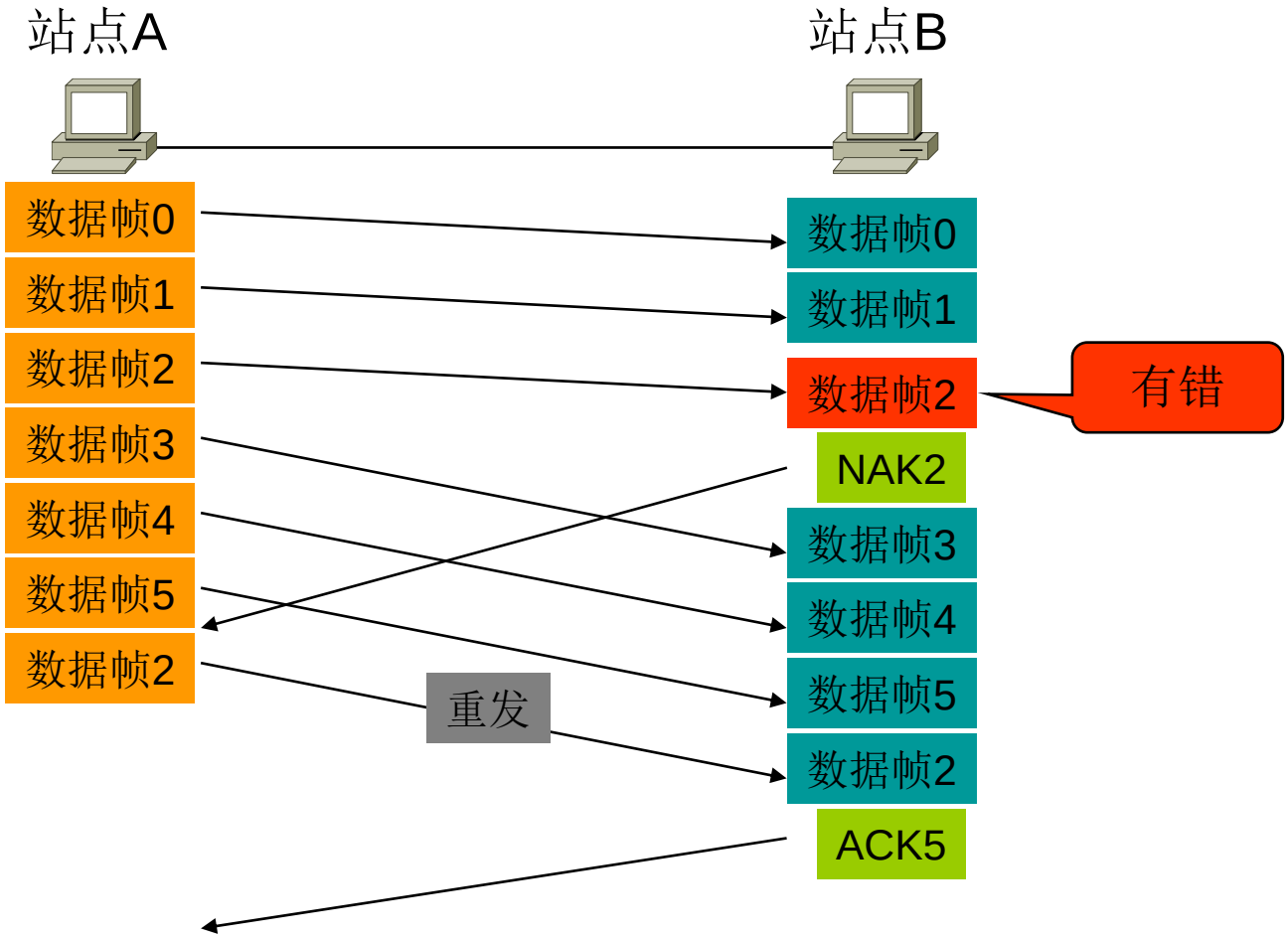
选择拒绝自动重传请求

- 只有特定的丢失或损坏帧被重发。
- 接收方收到的数据帧可以是不按顺序到达的。
- 选择拒绝自动重传请求与回退n自动重传请求有4个不同点：
  - 接收设备必须具有排序功能。在发送了NAK帧之后，必须存储所收到的所有帧，直到损坏的帧被重新收到为止。
  - 发送设备须具有查找机制，以便发现和选择需要重传的帧。
  - 所有的重传帧被排序和所有重复帧被辨别出来并删除之前，所收到的所有帧都必须保存。
  - ACK帧的编号指的是被正确接收的帧编号，不是指期望接收的帧编号，而NAK帧的编号指的是错误或丢失的帧。

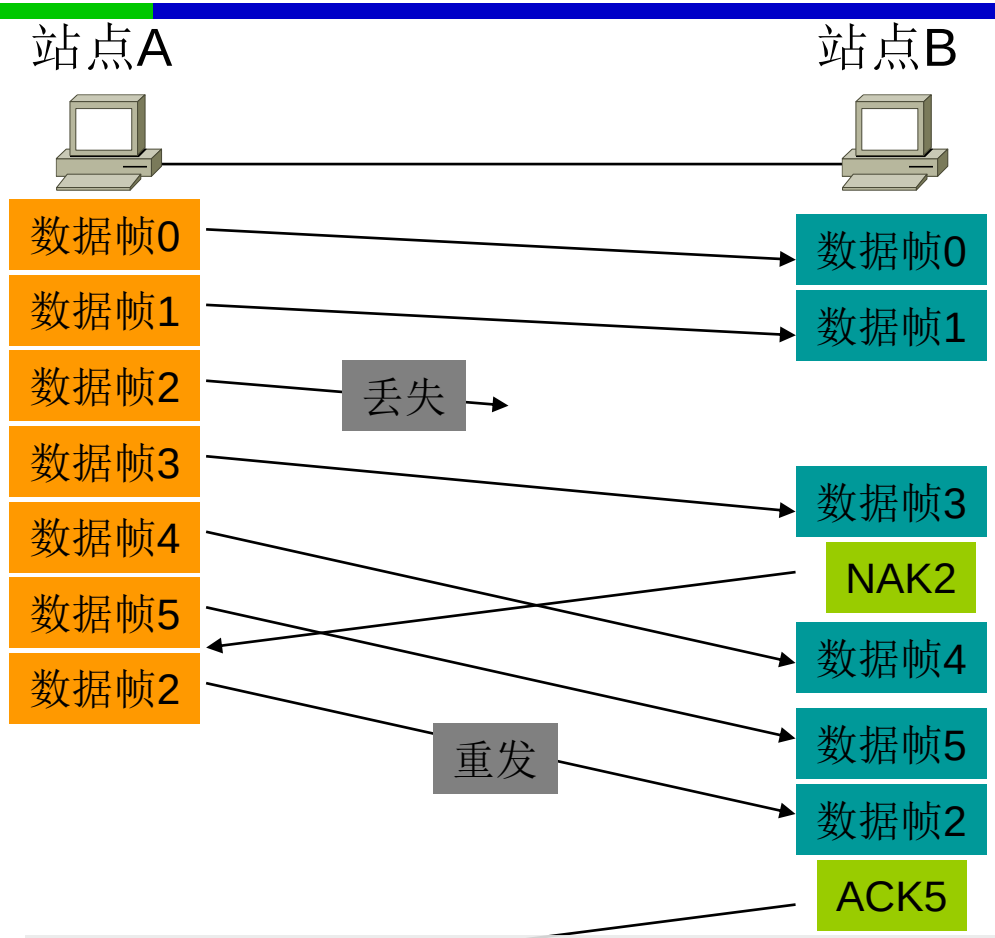
选择拒绝自动重传请求错误处理

- 差错情况有三种：
  - 帧破坏
  - 数据帧丢失
  - 确认帧丢失

帧破坏


站点A
站点B
数据帧0
数据帧1
数据帧2
数据帧3
数据帧4
数据帧5
数据帧2
数据帧0
数据帧1
数据帧2
有错
NAK2
数据帧3
数据帧4
数据帧5
数据帧2
重发
ACK5

数据帧丢失



如果丢失的是最后一帧？

- 接收方不做任何反应
- 发送方按丢失确认帧进行处理。

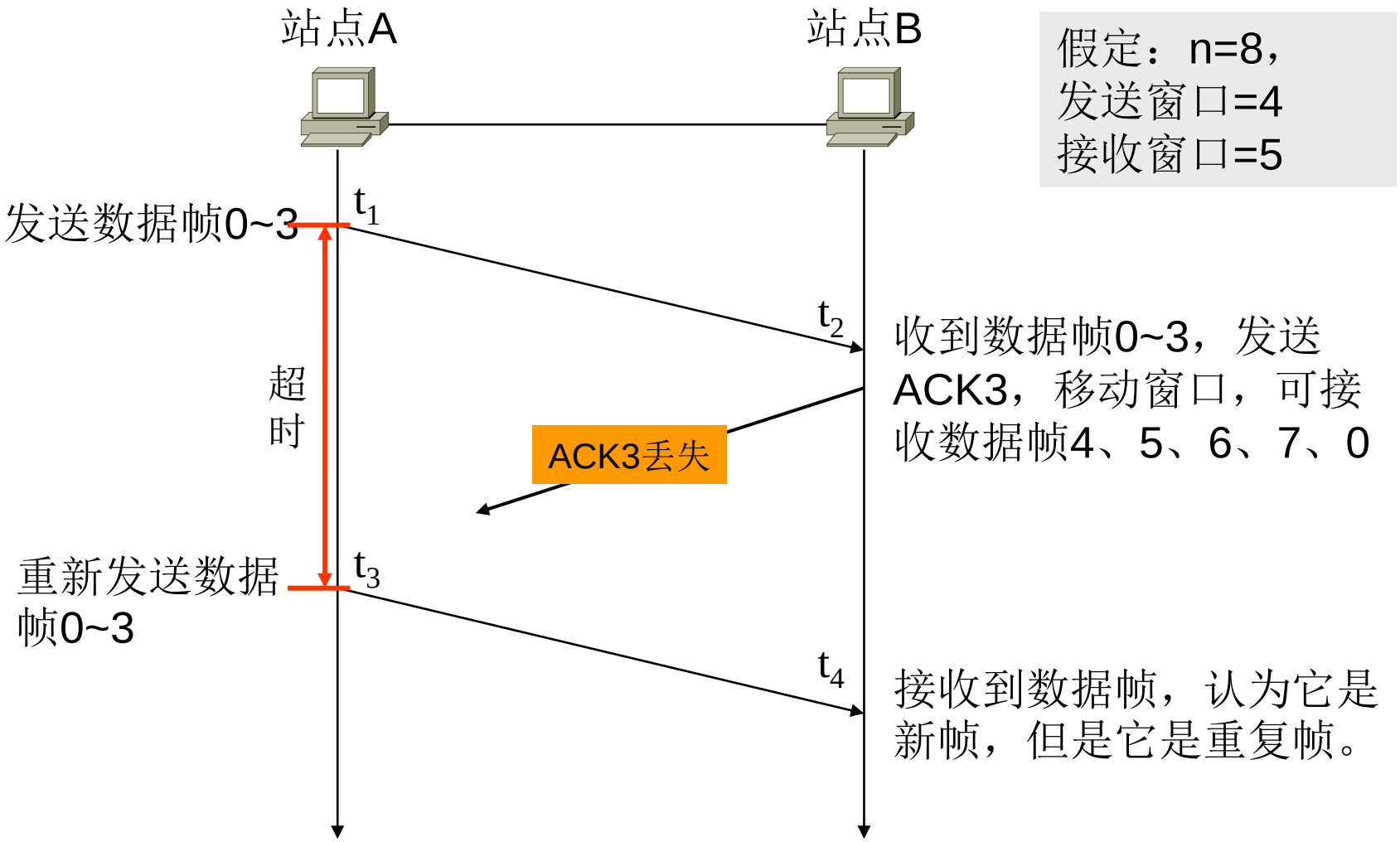
确认帧丢失

- 当发送窗口满时，或传输完毕时，启动定时时钟。
- 如果在预定时间段内没有应答到来，发送方将尚未应答的所有帧都重传一遍。

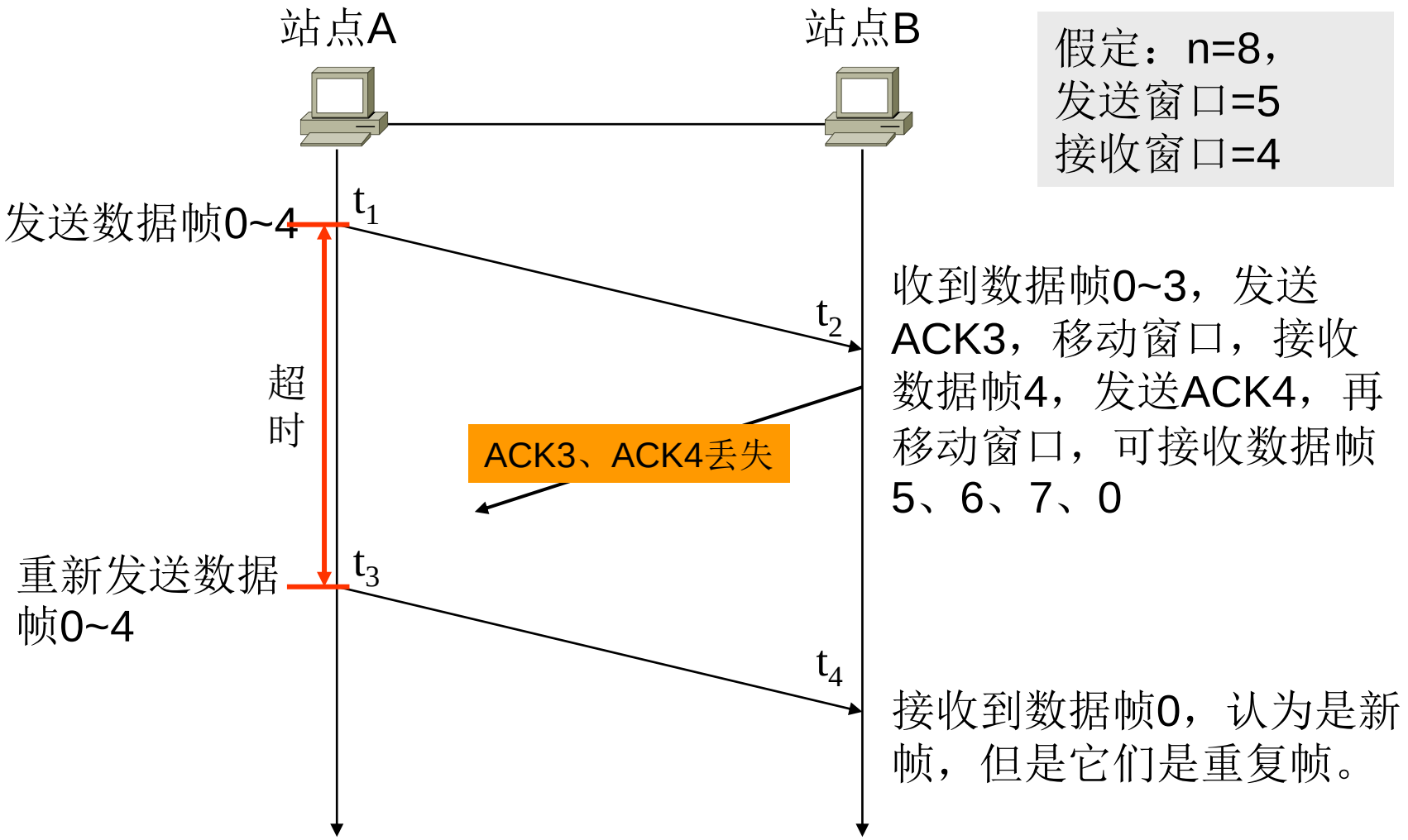
窗口的大小与编号范围的关系

- 如果帧的编号范围是0到n-1(即模n编号)，则发送窗口尺寸和接收窗口尺寸之和应小于或等于n。如果要求发送窗口和接收窗口大小相等，则窗口尺寸应该小于或等于n/2。

发送窗口尺寸太大，协议失败



接收窗口尺寸太大，协议失败



窗口大小的选择

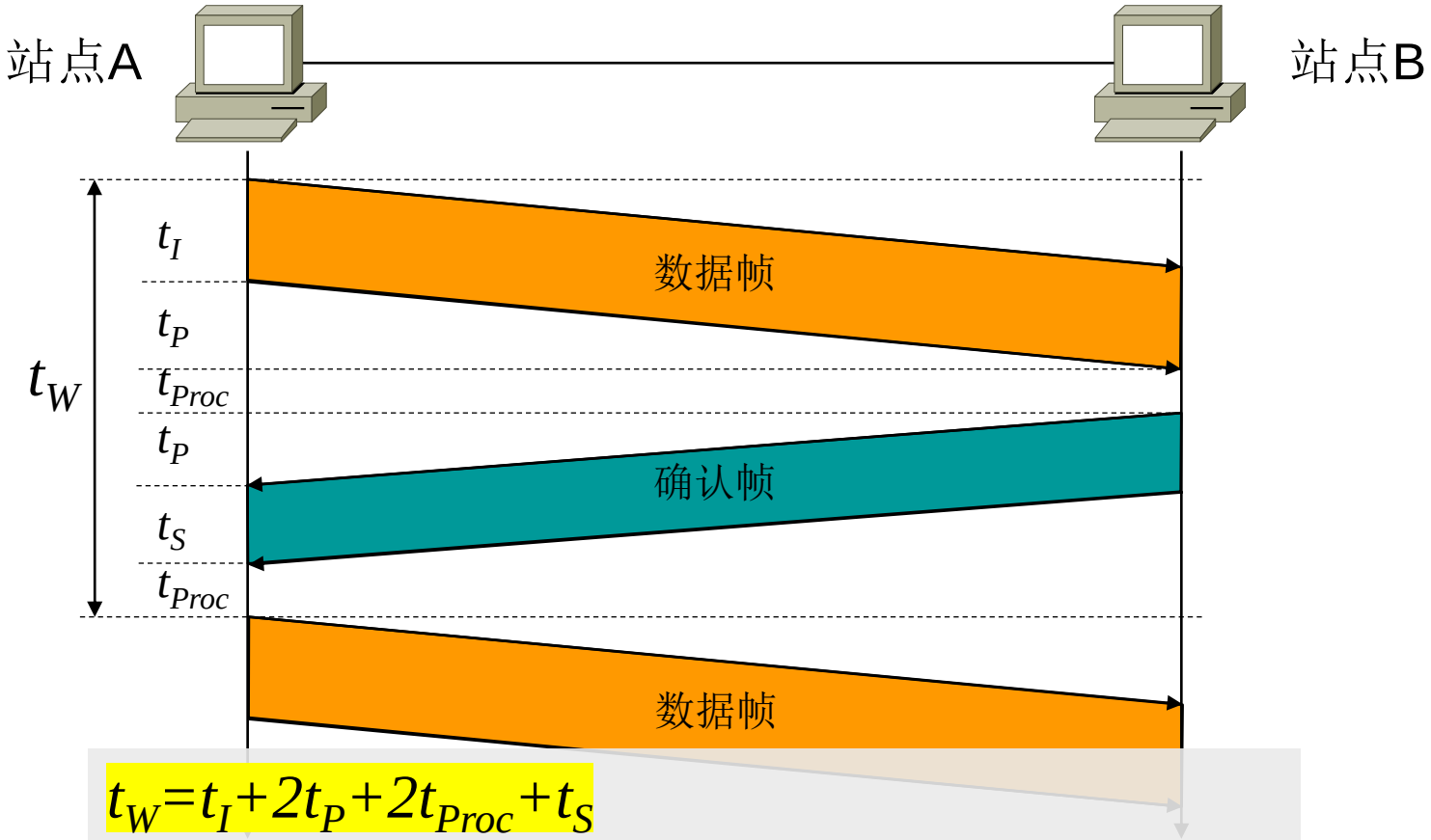
t_I：发送数据帧时间=帧长/数据传输率
t_S：发送确认帧时间=确认帧长/数据传输率
t_P：信号传输延时=距离/信号传输速度

- 窗口选得太大，会要求有足够大的缓存空间
- 如果选得太小，由于传播和发送延迟，第一个应答帧返回到发送方之前，发送方发送窗口中的帧已经全部发送出去，但是没有得到应答，发送方必须等待，从而影响了传输速度和传输效率。
- 假设一个帧的发送时间为t_I，传播时间为t_P，则窗口的大小n应该满足如下条件

$$nt_I > 2(t_I + t_P)$$

$$n > 2 + 2 \times t_P / t_I$$

滑动窗口协议完成一帧发送所需的最短时间



$t_W=t_I+2t_P+2t_{Proc}+t_S$

t_I：发送数据帧时间=帧长/数据传输率

t_S：发送确认帧时间=确认帧长/数据传输率

t_P：信号传输延时=距离/信号传输速度

t_{Proc}：节点处理数据时间

例

- 在卫星通信中，设帧的长度为1200bit，信道速率为4.8kbit/s，传播延迟为t_P=250ms，在全双工的数据通信中，窗口至少应为多大？
- 解：

$t_I=1200\div(4.8\times10^3)=0.25s=250ms$

所以：

$n>2\times(250+250)\div250$，即$n>4$

滑动窗口协议回退N的效率

- 不考虑应答帧的丢失，正确传送一帧所需的平均时间为：

$$t_V=t_I+pt_W/(1-p)$$

- 系统最大吞吐量：

$$\lambda_{max}=1/\ t_V=(1-p)/(t_I(1+\ p\ (a-1)))$$

- 系统的传输效率：最大吞吐量/极限吞吐量

$$\rho=(1-p)/(1+\ p\ (a-1))$$

其中$a=t_W/t_I$

例1

- 若数据帧的差错率为p=0.01，而a=4，则对于停止等待协议，有：

$$\rho = (1-p)/a$$

$$= 0.99/4 \approx 0.25$$

而对于滑动窗口协议，则有：

$$\rho = (1-p)/(1+p\,(a-1))$$

$$= 0.99/(1+0.01*3)$$

$$\approx 0.96。$$

- 在此情况下，即使p高达0.01，滑动窗口协议也比停止等待协议好。

例2

- 在一个广域网上，设p=0.01，数据帧长度为1200bit，线路速率为9.6kbps，线路长度为160km，应答帧长为120bit，则：

$$t_I=125\text{ms}，\ t_P=1\text{ms}，\ t_S=12.5\text{ms}$$

所以：

$$a=(t_I+2t_P+t_S)/t_I=139.5/125=1.12$$

对于停止等待协议，有：

$$\rho=(1-p)/a=0.99/1.12=0.89$$

而对于滑动窗口协议，则有：

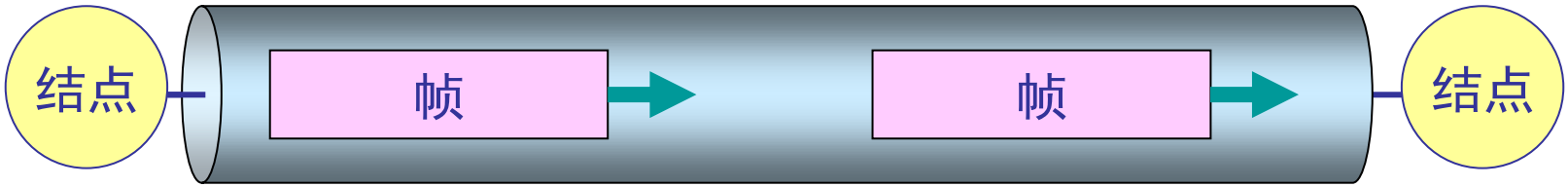
$$\rho=(1-p)/(1+(a-1)p)=0.989$$

两者相差不明显

3种协议方法总结

- 停止等待协议：
  - 发送窗口=1，接收窗口=1
- 滑动窗口中的回退N：
  - 发送窗口>1，接收窗口=1
- 滑动窗口中的选择拒绝：
  - 发送窗口>1，接收窗口>1
- WT+WR>=2的n次方
- WR发送窗口大小，WR接收窗口大小。N为窗口编号

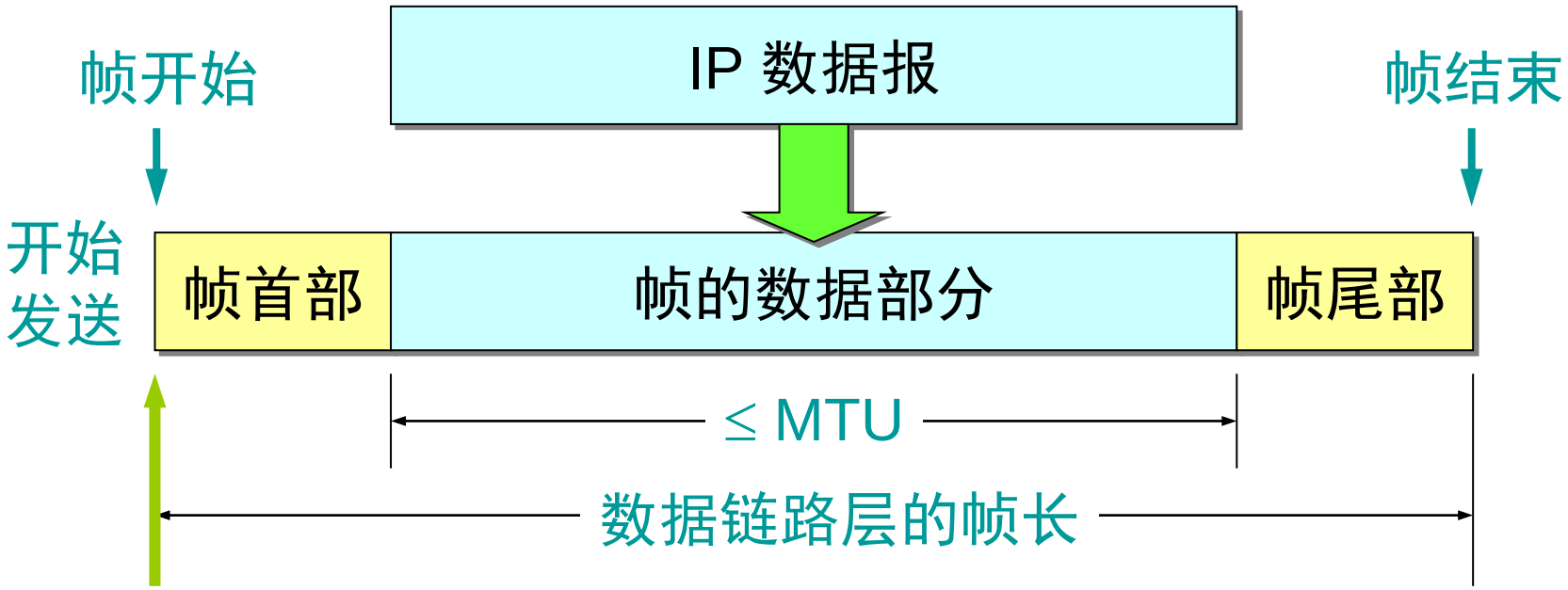
成帧的方法



- 帧的组成必须保证能识别一个完整的帧，并保证一旦出现传输错误导致前一个帧丢失，也必须能识别下一个帧（帧同步）
- 4种方法
  - 字符计数法
  - 含字节填充的分界符法
  - 含位填充的分界标志法
  - 物理层编码违例法

封装成帧

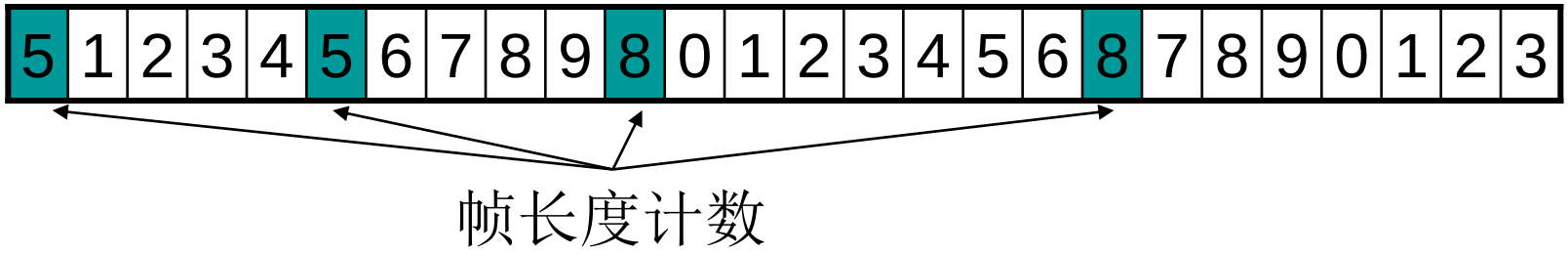
- 封装成帧(framing)就是在一段数据的前后分别添加首部和尾部，然后就构成了一个帧。确定帧的界限。
- 首部和尾部的一个重要作用就是进行帧定界。

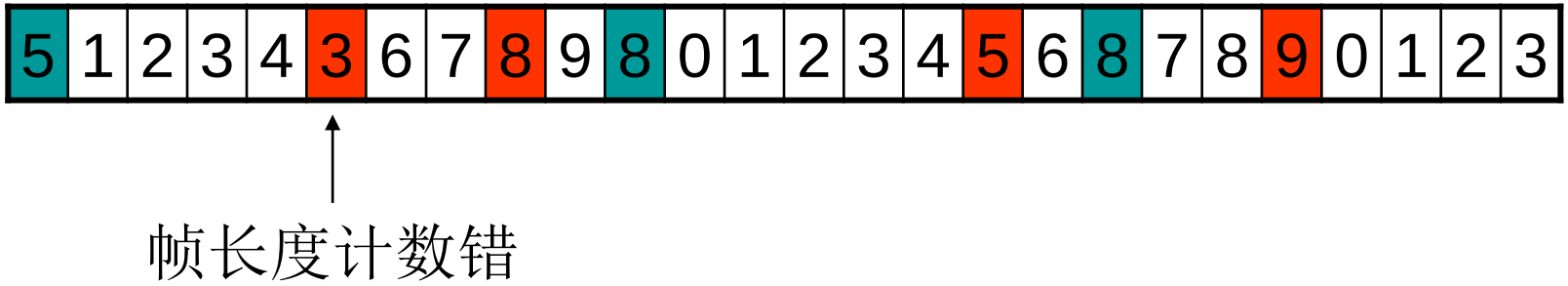


MTU(最大传送单元Maximum Transfer Unit)

字符记数法

- 帧的长度用一个字节表示，作为帧的头部的一个域

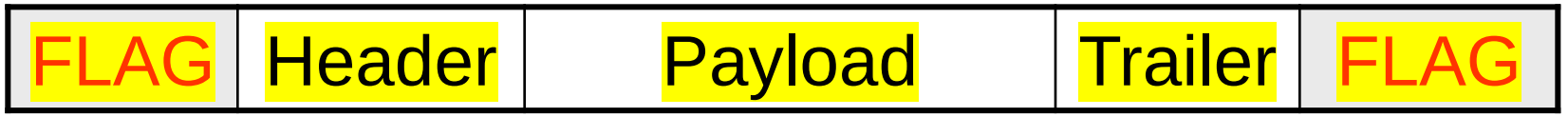




- 问题：一旦帧长度计数有误，将无法再同步

字节填充分界符法

- 用特殊的字符作为帧头和帧尾



- 这是一种面向字符的帧格式，所传输的数据都是字符，但在帧中不允许出现帧界符标志。
- 常用于面向字符的串行通信中
- 所用的特定字符依赖于所采用的字符编码集,兼容性比较差。
- 不适合传输数据中包含二进制数的帧
  - 对于在数据中偶尔出现的帧界符标志，可以在前面插入一个转义字节

字节填充分界符法例

- 通常FLAG用ASCII字符7EH定义
- 对于二进制数中偶尔出现的FLAG前面插入一个ESC(ASCII字符1BH)

原始数据	线路上的数据
41 33 7E 9C 4B 0C	41 33 1B 7E 9C 4B 0C
41 33 1B 9C 4B 0C	41 33 1B 1B 9C 4B 0C
41 33 1B 7E 9C 4B 0C	41 33 1B 1B 1B 7E 9C 4B 0C
41 33 1B 1B 9C 4B 0C	41 33 1B 1B 1B 1B 9C 4B 0C

位填充分界标志法

- 在面向二进制位的通信中常使用带位填充的首尾标志格式，如HDLC
- 以特殊的位模式01111110作为帧标志，即一个帧的开始和结束
- 当帧中出现一个与帧标志相同的位串01111110时，在5个1后插入1个0，变成011111010。接收方将自动删除第5个1后的0
- 含位填充的分界标志法也称为位插入法

原始数据　01101111111111111110010

线路上的数据　01101111101111101111010010

物理层编码违例法

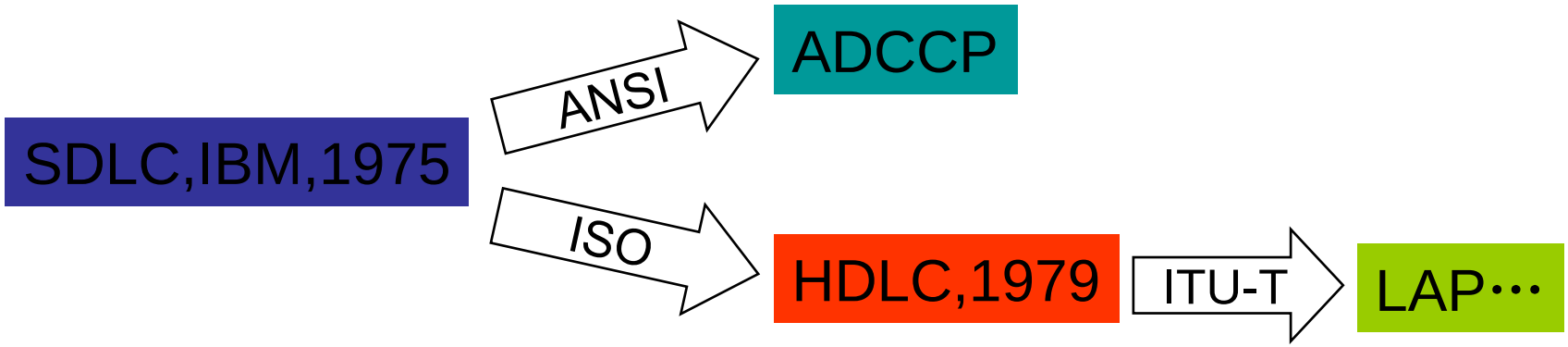
- 例如在双相位编码中
  - 数据0用低高电平对
  - 数据1用高低电平对
- 连续高电平或连续低电平可用作帧边界

4.3 HDLC通信协议 不支持多点平衡配置

- 面向字符型数据链路层协议是以字符为控制传输信息的基本单元，由于与特定字符集有关，其缺点表现在：
  - 兼容性差；
  - 传输透明性不好；
  - 等待发送方式，传输效率低。
- 面向比特型协议的设计目标：
  - 以比特作为传输控制信息的基本单元；
  - 数据帧与控制帧格式相同；
  - 传输透明性好；
  - 连续发送，传输效率高。

面向比特型协议

- 在面向比特的协议中，帧被看作一系列比特。
- 面向比特协议的控制信息可以是一个或多个比特，可以用较短的比特位表示较丰富的控制信息。
- 面向比特的协议的另一个重要的优点：不受任何编码系统的制约。

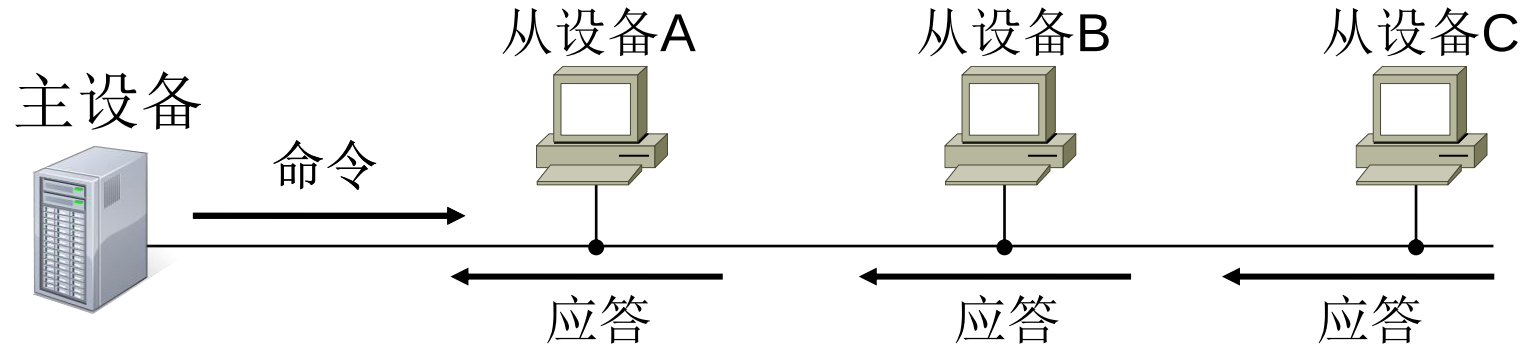


4.3.1 站点类型、链路配置和通信方式

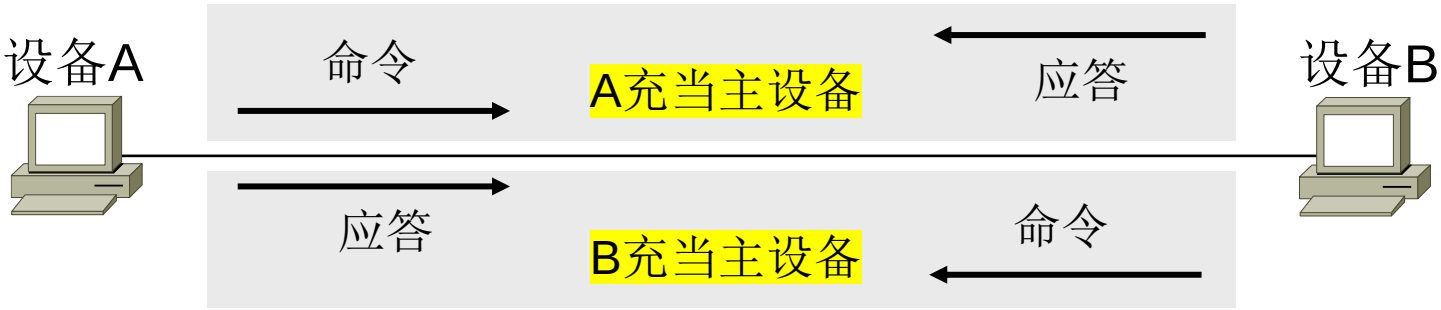
- 站点类型： HDLC协议中站点有三种类型
  - 主站点：具有控制权的一方，主站发出命令
  - 从站点：接受命令，发出响应，配合主站工作
  - 复合站点：由传输的属性、方向决定工作方式
    - 复合站同时具有主站与从站的功能
    - 每个复合站都可以发出命令与响应
- 链路配置： HDLC协议有2种配置方式
  - 非平衡式
    - 点—点方式（对称式）
    - 多点方式（非平衡式）
  - 平衡式

链路配置

非平衡式



对称式



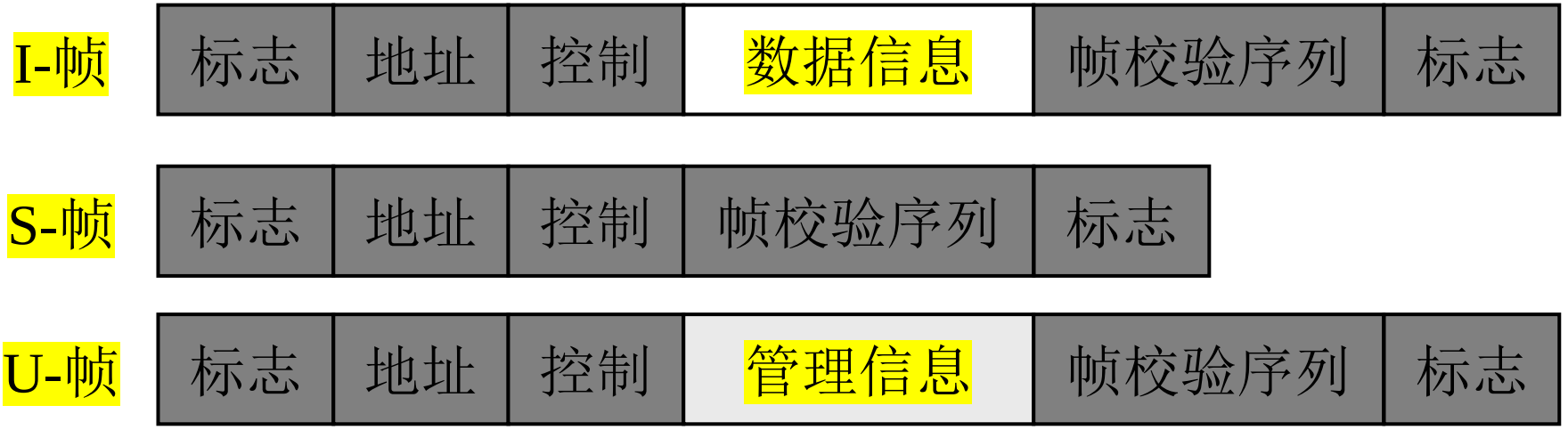
平衡式

通信方式

- 通信方式：在一次交互中所涉及到的两个设备之间的关系。这种方式描述了由谁控制链路。支持3种不同的工作方式：
  - 正常应答方式(NRM)、异步应答方式(ARM)和异步平衡方式(ABM)
- 非平衡式—采用正常应答方式和异步应答方式
  - 正常应答方式：
    - 主站可以随时向从站传输数据帧；
    - 从站只有在主站向它发送命令帧进行探询、响应后才可以向主站发送数据帧。
  - 异步应答方式：
    - 主站负责数据链路的初始化、链路的建立、释放与差错恢复等功能。
    - 主站和从站可以随时相互传输数据帧；
    - 从站可以不需要等待主站发出探询就可以发送数据。
- 平衡式—采用异步平衡方式
  - 每个复合站都可以平等地发起数据传输，不需要得到对方复合站的许可。

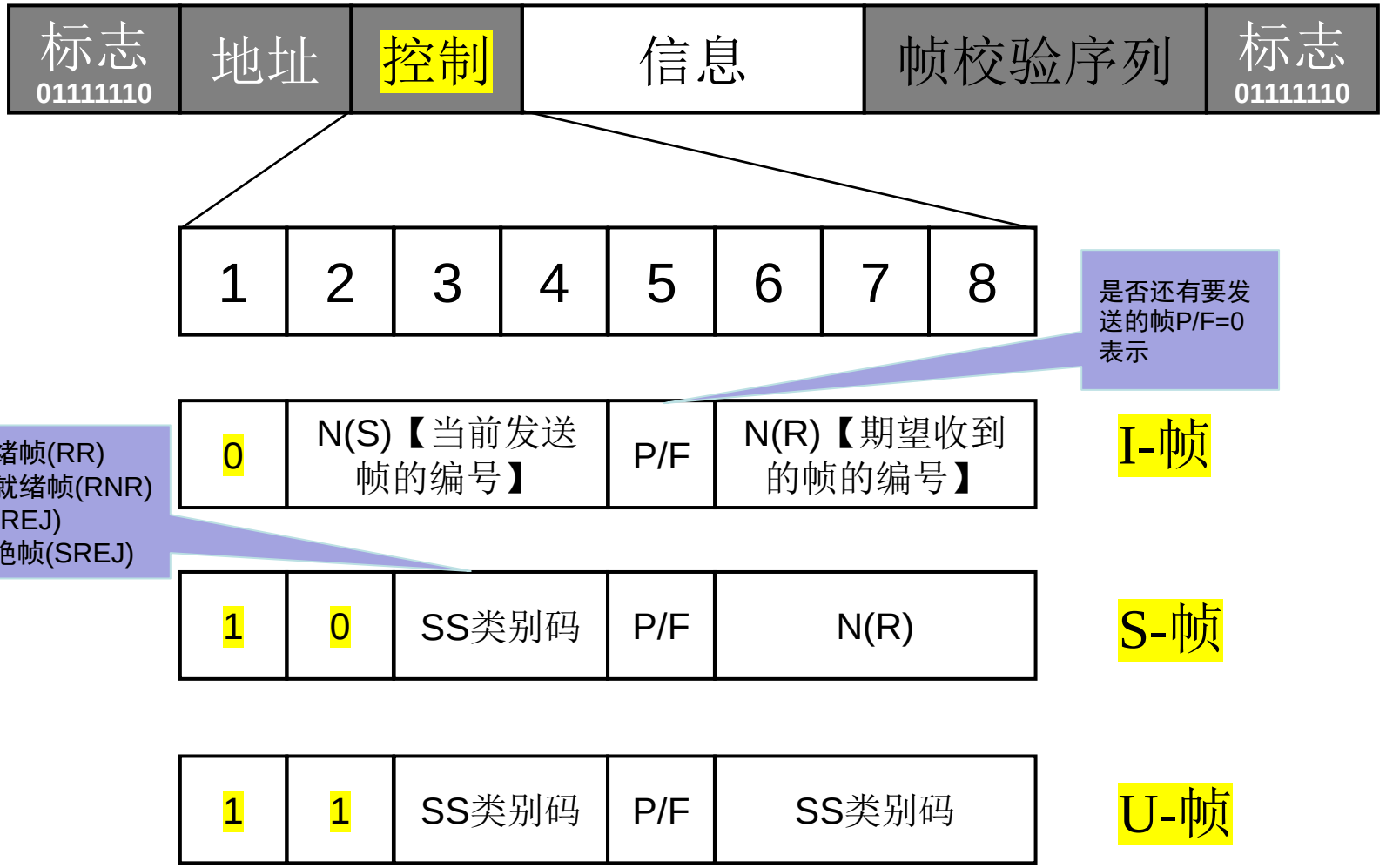
4.3.2 HDLC帧格式

- 为了支持3种通信方式，定义了三种类型的帧：
  - 信息帧（I-帧）：数据及与数据有关的控制信息
  - 监控帧（S-帧）：流量和错误控制信息
  - 无编号帧（U-帧）：链路管理服务



标志 地址 控制 （数据信息/管理信息）帧校验序列 标志

HDLC帧格式 格式不记



标志字段

- 一个字节(8位)，其比特模式为01111110；表示一个帧的开始和结束，并且为接收方提供同步手段。
- 位填充法
  - 发送方发送一个含有五个以上连续1的数据时，它总是在第五个1后面插入一个冗余的0。不管第六个比特是0还是1
  - 接收方接收时作相反的动作(去掉5个1后面的零)。
  - 例如：

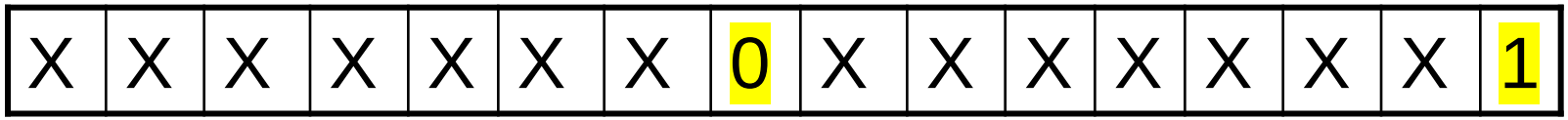
    要发送的序列是01111101111110，发送时变成01111100111110 10

地址字段

- 该字段是指从站地址，或者是以从站方式运行的复合站地址。
- 如果帧是由主站发送的，则地址表示接收该帧的从站地址。
- 如果帧是由从站发送的，则地址表示发送该帧的从站地址。

地址字段的扩展

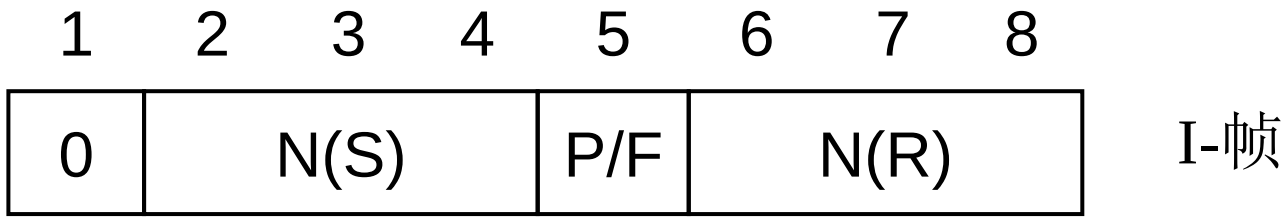
- 根据网络的规模，地址字段可以有1个或几个字节的长度，如果地址字段只有一个字节，该字节最后一比特总是1。如果地址字段有多个字节，除最后一个字节外其他所有字节都要以0结尾，最后一个字节要以1结尾。



控制字段

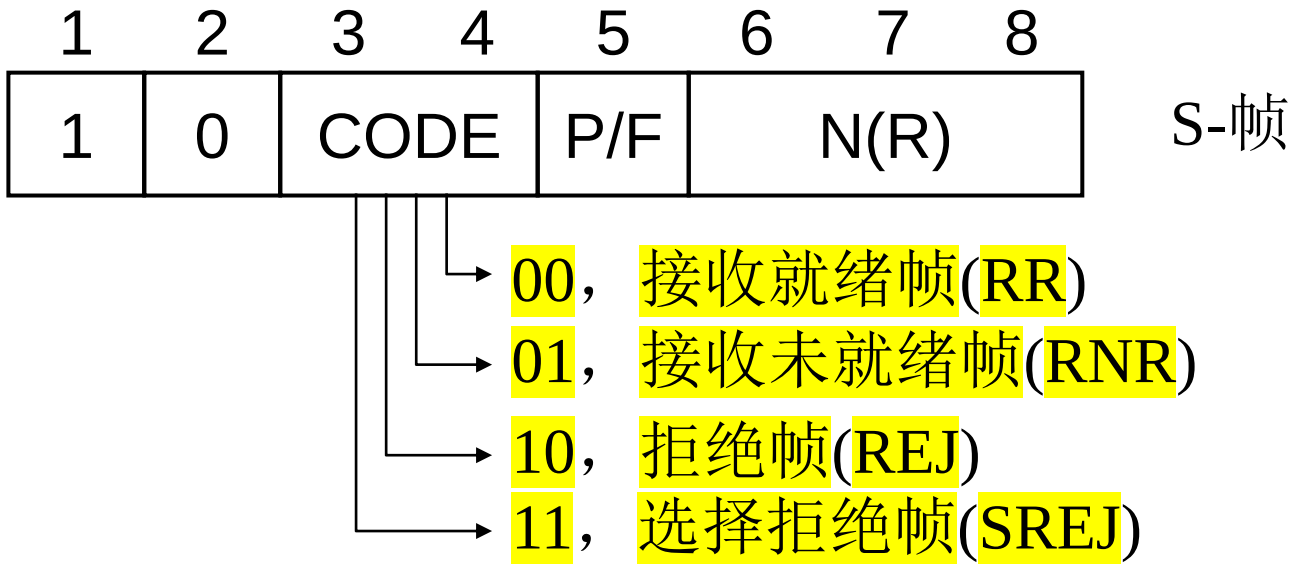
- 用于流量管理。根据控制字段的内容，可以知道一个帧的类型。
  - 如果控制字段的第一个比特是0，该帧就是一个信息帧I。
  - 如果控制字段的前两个比特是10，该帧就是一个监管帧S。
  - 如果控制字段的前两个比特是11，该帧就是一个无编号帧U。

I-帧的控制字段 记



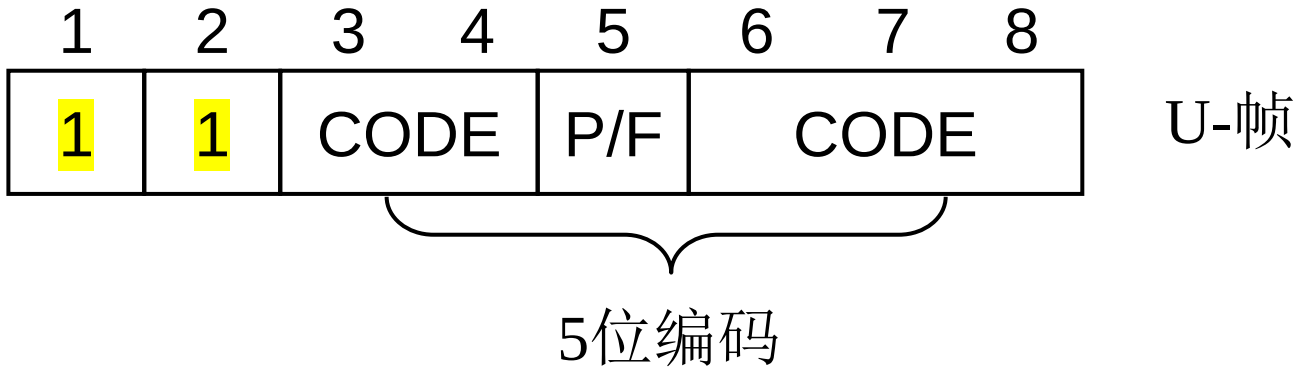
- 第一位为0，它是I帧的标志。
- 第2、3、4位：N(S)位，当前发送帧的编号。
- 第5位：P/F位，表示是否还有要发送的帧
  - P/F=0，表示还有要发送的帧；
  - P/F=1，表示没有要发送的帧，发送结束。
- 第6、7、8位：N(R)位，期望收到的帧编号。

S-帧的控制字段 记



- 当不能在一个I-帧上捎带确认信息时，例如它没有数据信息可发时，用S-帧来对收到的数据帧进行应答。N(R)就是期望收到的帧的编号，它是一个应答域。
- 如果最近一帧是正确的，N(R)域将是序列中下一帧的序号；如果最近一帧是错误的，N(R)域是这个损坏帧的序号。
- S-帧中的P/F位一般都应置1

U-帧的控制字段



- U-帧中的P/F位一般都应置1。
- 无论是I-帧、U-帧、S-帧，P/F位置1的数据帧和命令帧都是要立即进行响应的帧。

信息字段

- S-帧中没有信息字段。
- I-帧的信息字段是用户数据信息。
- U-帧中的信息字段是链路管理信息。
- 把发送信息和控制信息结合到一帧中的技术称为捎带确认。

帧校验序列字段

- 帧校验序列是HDLC协议的错误检测字段。
- 它含有一个两字节或一个四字节的循环冗余校验(CRC)码。
  - 常用CRC-CCITT：$X^{16}+X^{12}+X^{5}+1$

4.3.3 监控帧(S-帧)的使用方式 记

- S-帧没有信息字段，但是每一帧都给接收方带去了某种信息。
- 信息的含义需要通过S-帧的类型和传输上下文来获得。 4种类型
  - 接收就绪(RR)帧
  - 接收未就绪(RNR)帧
  - 拒绝(REJ)帧
  - 选择拒绝(SREJ)帧

接收就绪(RR)帧

- 接收就绪帧有四种使用方式，各有不同意义：
  - 应答(ACK)：接收站本身没有数据信息发送时，用一个接收就绪帧作为应答帧来对所接收的数据帧进行应答。
  - 查询(POLL)：当主站点询问从站点是否有数据发送时，它向从站点发送一个P/F位置1的RR帧。
  - 对查询的否定应答(POLL.NAK)：从站点用一个P/F位置1的RR帧回答主站点的查询，通知主站点它没有数据发送。如果从站点有数据发送，从站点用I-帧来响应查询。
  - 对选择的肯定应答(SEL.ACK)：如果从站点收到了主站点的选择帧，并且从站点准备好从主站接收数据，它用一个P/F位置1的RR帧回答主站点的选择。

接收未就绪(RNR)帧

- 接收未就绪帧有三种使用方式：
  - 应答(ACK)： 接收方向发送方返回的RNR帧有两个意思。
    - 应答，表示接收方收到了编号在N(R)以前的所有帧。
    - 要求发送方暂停发送，直到发送方收到一个RR帧为止。
  - 选择(SEL)：当主站点想要向某个从站点发送数据时，它通过发送一个P/F位置1的RNR帧来通知从站点。
  - 对选择的否定应答(SEL.NAK)：当选择的从设备不能接收数据时，它回答一个P/F置1的RNR帧。

拒绝(REJ)帧和选择拒绝(SREJ)帧

- REJ：在回退N自动重传请求中，当接收方没有要发送的数据用来捎带应答信息时，返回的一个否定应答帧。在REJ帧中，N(R)域指明了损坏帧的序号，损坏帧及其以后所有帧必须重发。
- SREJ：在选择拒绝自动重传请求中，当接收方收到一个损坏帧时，它用一个选择拒绝帧告诉发送方哪一帧被损坏。N(R)指明了被损坏帧的编号。被损坏的帧需要重发。

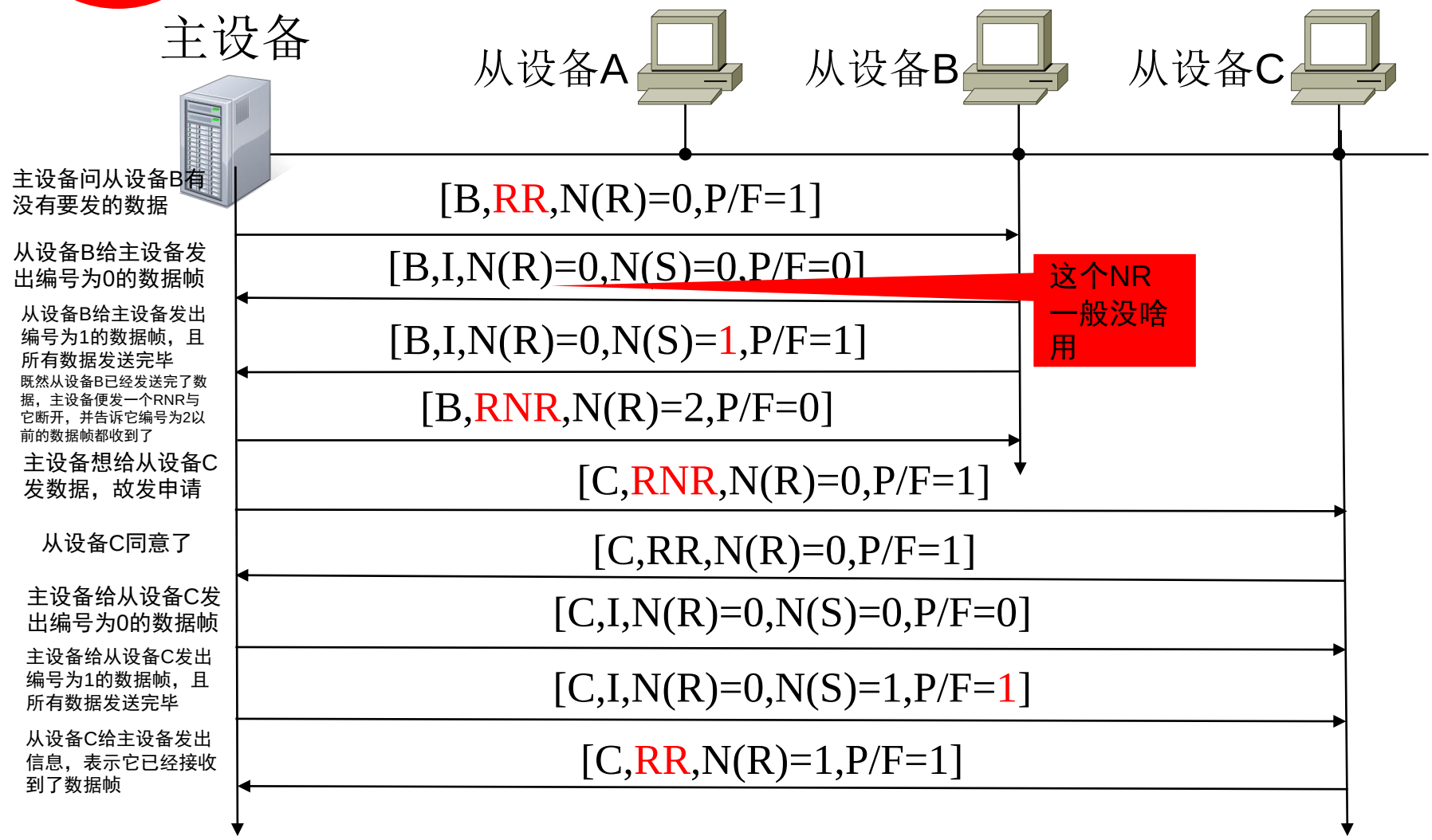
4.3.4 无编号帧的种类及意义

- 无编号帧是用来在互连设备之间交换会话管理信息和控制信息的。
- 无编号帧的控制字段中有5位编码位，这5个编码位可用来表示32种不同类型的无编号帧。
- 包含5个基本功能类：
  - 方式设置
  - 无序号交互
  - 断开连接
  - 启动模式
  - 混杂形式

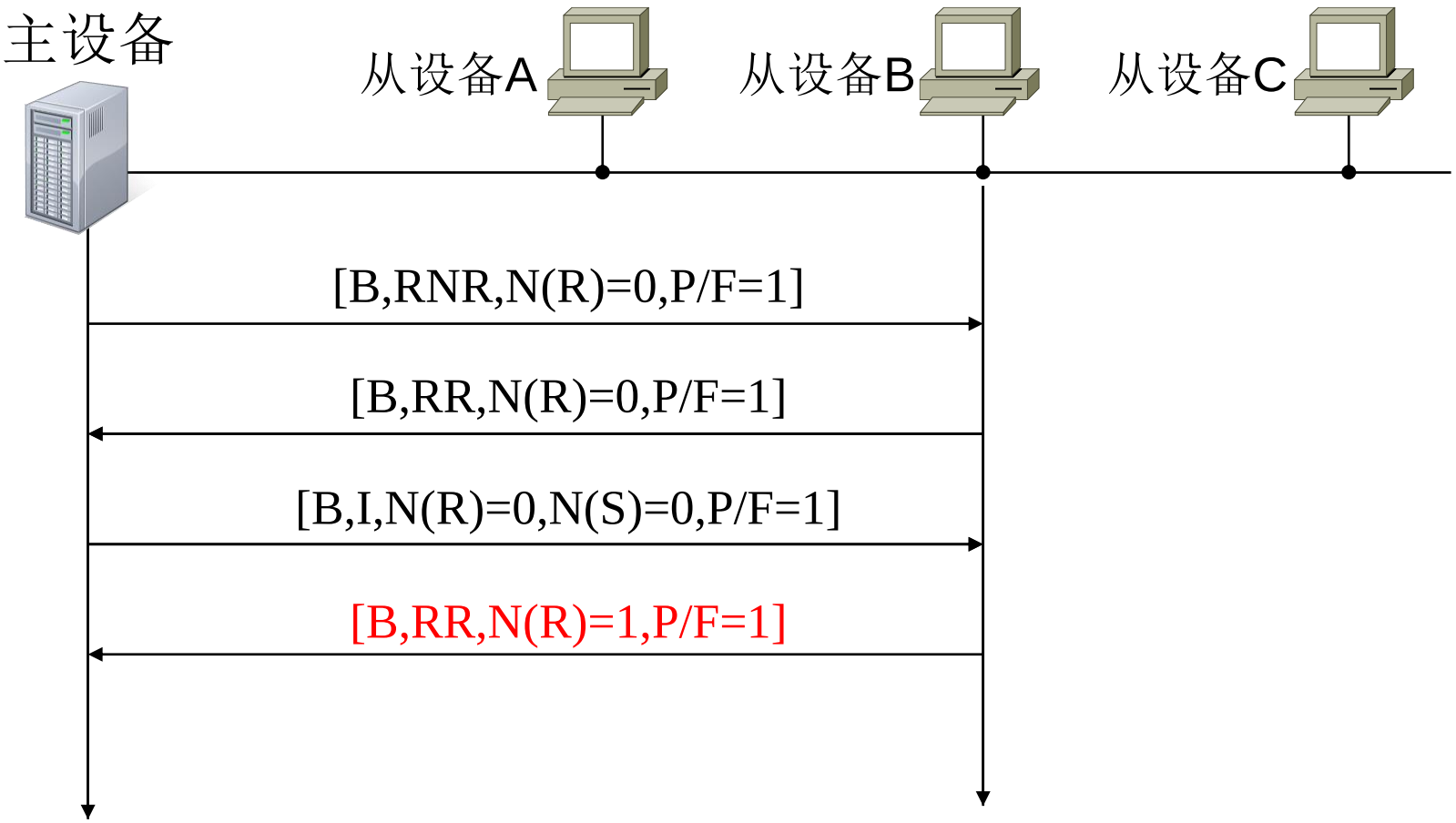
各种类型的无编号帧 标黄的要记住

	编码	名称	性质	意义
方式设置	00 001	SNRM	命令	设置正常响应模式。
	11 011	SNRME	命令	设置扩展正常响应模式。
	11 000	SARM	命令	设置异步响应模式。
	11 010	SARME	命令	设置扩展异步响应模式。
	11 100	SABM	命令	设置异步平衡模式。
	11 110	SABME	命令	设置扩展异步平衡模式。
无序号交互	00 100	UP	命令	无序号轮询。从指定站发来的关于对状态信息的轮询。
	00 000	UI	命令/响应	无序号信息。通常用来发送状态信息，一般是在UP或SIM信号后发送。
	00 110	UA	响应	无序号确认。通常用来确认刚才发送的命令，如设置模式和断开连接。
断开连接	00 010	RD	响应	请求断开连接。
	00 011	DISC	命令	断开连接。初始化两个站之间的断连。当另外一个站用一个UA响应时，断连结束。
	11 000	DM	响应	断开连接方式。告诉主站，从站处于断连状态。
启动模式	10 000	RIM	响应	请求初始化模式。从站请求主站发送一个SIM。
	10 000	SIM	命令	设置初始化模式。命令其它的站初始化它们的数据链路控制功能。
	11 001	RSET	命令	重启动。
混杂模式	11 101	XID	命令/响应	交换标示。允许两个站交换它们的标示和状态信息。
	10 001	FRMR	响应	帧拒绝。通常被用于一个U-帧出现了同步错误。

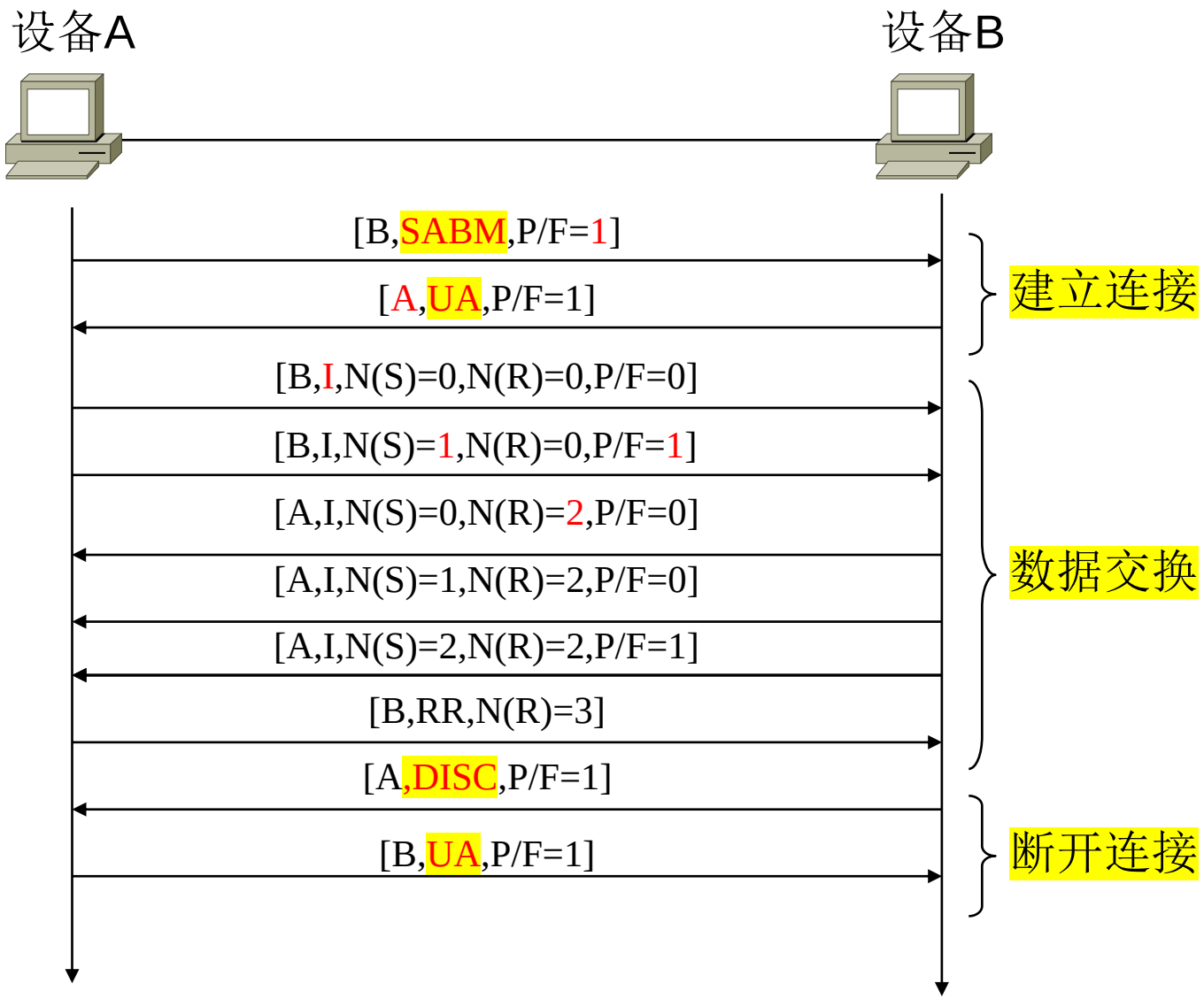
查询/响应


主设备
从设备A
从设备B
从设备C
主设备问从设备B有没有要发的数据
[B,RR,N(R)=0,P/F=1]
从设备B给主设备发出编号为0的数据帧
[B,I,N(R)=0,N(S)=0,P/F=0]
这个NR一般没啥用
从设备B给主设备发出编号为1的数据帧，且所有数据发送完毕
[B,I,N(R)=0,N(S)=1,P/F=1]
既然从设备B已经发送完了数据，主设备便发一个RNR与它断开，并告诉它编号为2以前的数据帧都收到了
[B,RNR,N(R)=2,P/F=0]
主设备想给从设备C发数据，故发申请
[C,RNR,N(R)=0,P/F=1]
从设备C同意了
[C,RR,N(R)=0,P/F=1]
主设备给从设备C发出编号为0的数据帧
[C,I,N(R)=0,N(S)=0,P/F=0]
主设备给从设备C发出编号为1的数据帧，且所有数据发送完毕
[C,I,N(R)=0,N(S)=1,P/F=1]
从设备C给主设备发出信息，表示它已经接收到了数据帧
[C,RR,N(R)=1,P/F=1]

选择/应答


主设备
从设备A
从设备B
从设备C
[B,RNR,N(R)=0,P/F=1]
[B,RR,N(R)=0,P/F=1]
[B,I,N(R)=0,N(S)=0,P/F=1]
[B,RR,N(R)=1,P/F=1]

对等通信


设备A
设备B
[B,SABM,P/F=1]
[A,UA,P/F=1]
建立连接
[B,I,N(S)=0,N(R)=0,P/F=0]
[B,I,N(S)=1,N(R)=0,P/F=1]
[A,I,N(S)=0,N(R)=2,P/F=0]
[A,I,N(S)=1,N(R)=2,P/F=0]
[A,I,N(S)=2,N(R)=2,P/F=1]
[B,RR,N(R)=3]
数据交换
[A,DISC,P/F=1]
[B,UA,P/F=1]
断开连接

4.4 IEEE局域网通信协议

- 局域网(LAN) 是在有限的地理范围内连接许多独立设备，使它们相互之间直接进行通信的系统。
- 局域网出现不久，其产品的数量和品种迅速增加，迫切要求局域网标准化，以便用户设备和局域网互连。

局域网体系结构

- IEEE标准：(Institute of Electrical and Electronics Engineers电气和电子工程师协会)
  - 以太网
  - 令牌总线
  - 令牌环
- ANSI标准：(American National Standards Institute美国国家标准学会)
  - 光纤分布式数据接口(FDDI)

局域网 vs 广域网

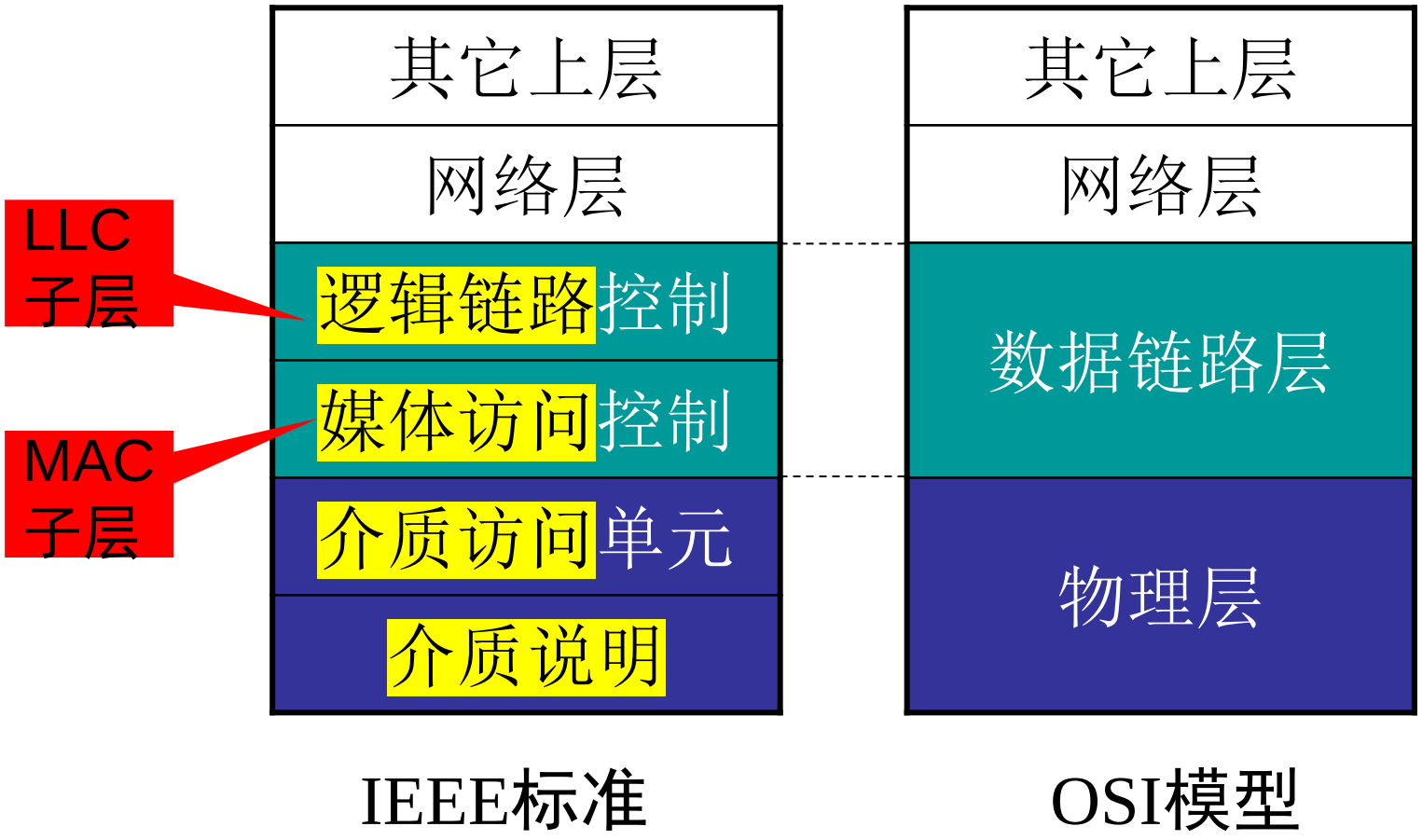
- 网络协议
- 路由技术
- 管理维护

4.4.1 IEEE局域网标准

- 1980年2月IEEE成立了802局域网标准委员会，IEEE制订的有关局域网的标准，主要指的是IEEE 802标准。
  - 802.1：LAN中的网络互连标准
  - 802.2：LLC逻辑链路控制协议标准
  - 802.3：CSMA/CD媒体访问方法
  - 802.4：令牌总线访问方法
  - 802.5：令牌环访问方法
  - 802.11：无线局域网协议

4.4.2 IEEE局域网参考模型

- LAN与OSI的关系



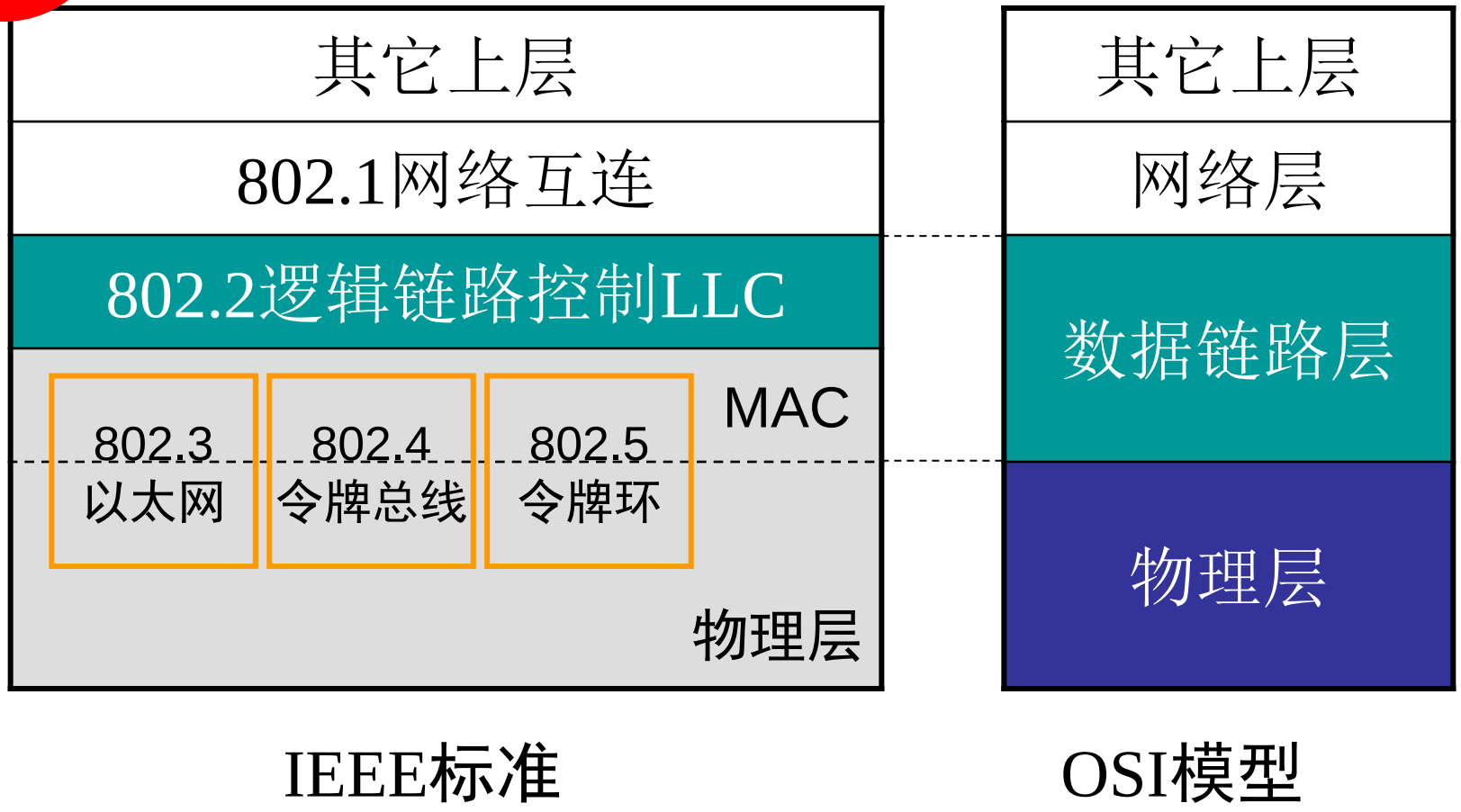
IEEE第一层

- 物理层以透明方式为上层提供一个物理连接，分为两个子层：
  - 上子层—介质访问单元(MAU)，其作用是信息编码、信号发送和介质处理等
  - 下子层—介质说明，电缆可以是各种介质

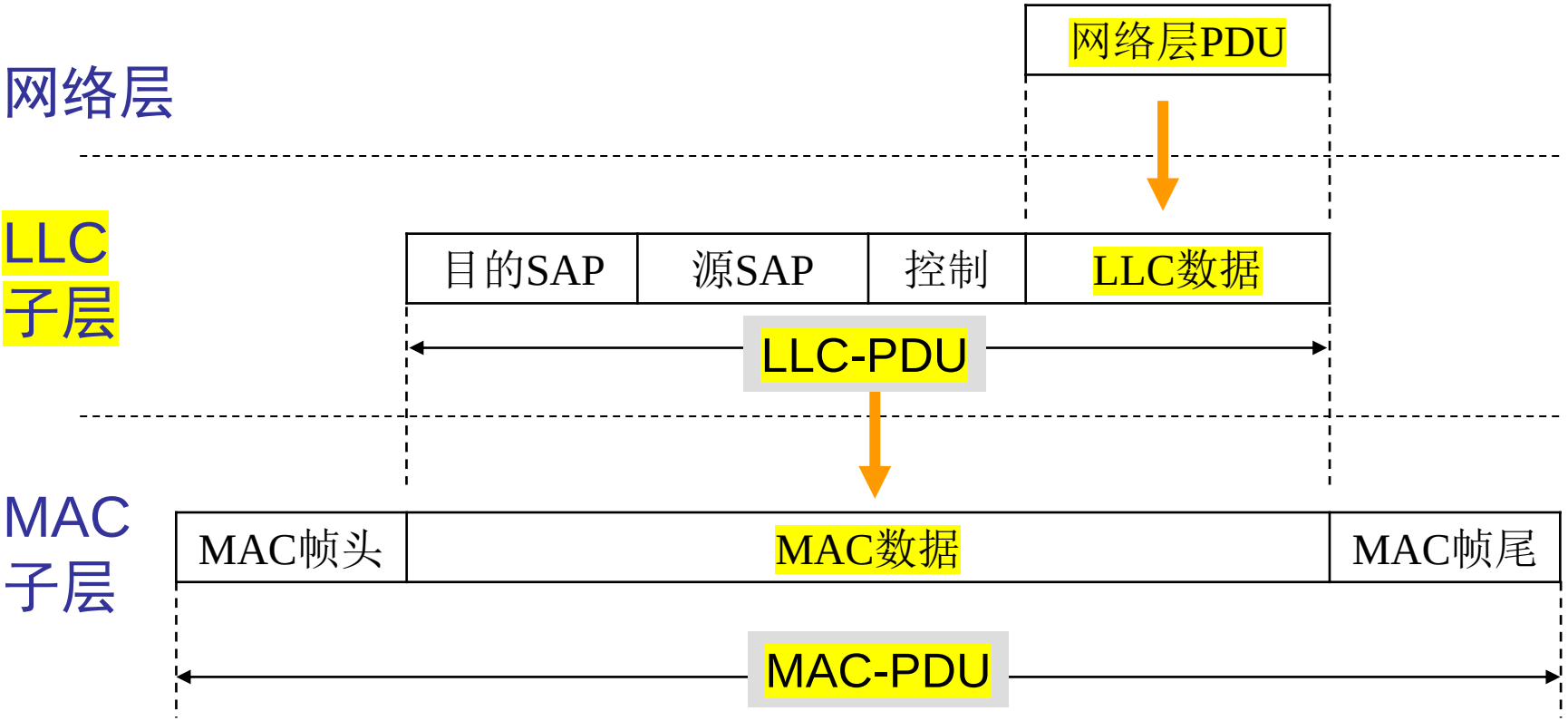
IEEE第二层

- 数据链路层包含2个子层：
  - 逻辑链路控制(LLC)子层：IEEE 802数据链路层的上子层，它对于所有的LAN协议来说都是相同的。
    - 该层中包含了数据帧中和用户相关的部分：逻辑地址、控制信息和数据
  - 媒体访问控制(MAC)子层：解决共享介质的竞争使用问题。它包含了将数据从一个地方传送到另一个地方所必需的同步、标记、流量和差错控制的规范，同时也包括下一个站点的物理地址。
    - 不同局域网具有不同的MAC协议
    - 保证物理功能和逻辑功能的连续性

802标准与OSI模型的层次对应关系

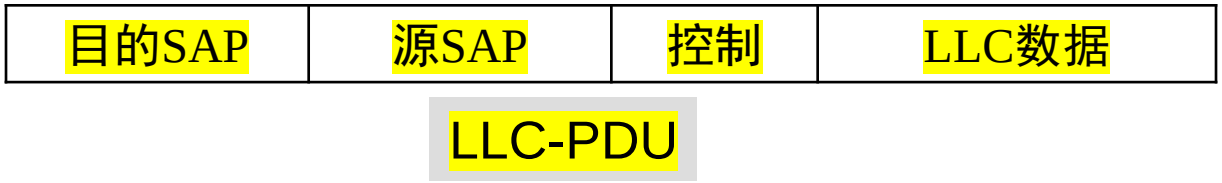


4.4.3 逻辑链路控制LLC



LLC-PDU与相邻层PDU之间的关系

LLC子层格式



- LLC-PDU与HDLC类似，包含四个域：
  - 目的服务访问点(DSAP)
  - 源服务访问点(SSAP)
  - 控制域
  - 信息域
- DSAP和SSAP是LLC所使用的地址，SAP是服务访问点。用来标明接收和发送数据的计算机上的协议栈。

LLC帧的控制字段

- 分为三种：信息帧、监控帧和无编号帧

1b	7b	1b	7b
0	N(S)	P/F	N(R)

信息帧

1b	1b	2b	4b	1b	7b
1	0	SS	(保留位)	P/F	N(R)

监控帧

1	1	M	M	P/F	M	M	M

无编号帧

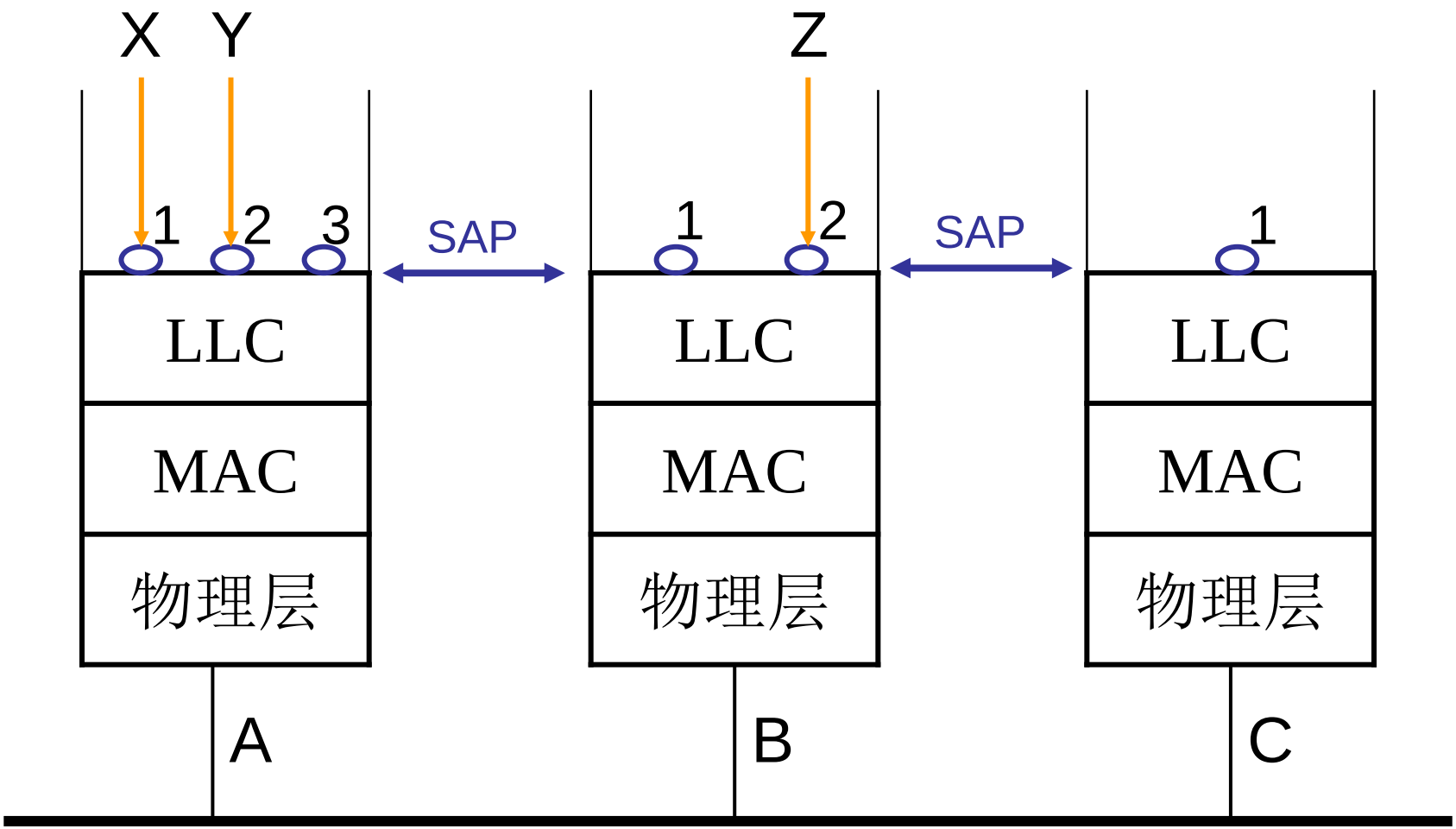
监控帧

- 监控帧用于应答和流量控制
- 共有三种形式的监控帧：
  - 接收就绪帧（RR）
  - 拒绝帧（REJ）
  - 接收未就绪帧（RNR）

LLC地址与MAC地址

- 在MAC 帧的帧首中，有目的站地址和源站地址，它们是6字节长。MAC帧中的地址是站点的物理地址。
- 在 LLC帧的帧首中，使用 DSAP和 SSAP，该地址是逻辑地址，是数据链路层的不同服务访问点。
  - DSAP：首位是0表示个体，首位是1表示组
  - SSAP：首位是0表示命令，首位是1表示响应

LLC与MAC地址、链路复用



1: (A,1) --- (C,1)　　2: (A,2)---(B,1)　　3: (B,2)---(A,3)

LLC子层所提供的服务

- LLC子层向上层可提供3种服务
  - 非确认的无连接服务：不需要确认，质量保证由高层完成
  - 面向连接服务：有3个阶段，连接建立、数据传输、连接释放
  - 确认的无连接服务：对数据帧提供确认，但传输前无连接建立过程

4.5 以太网 无确认无连接

无线局域网 有确认无连接

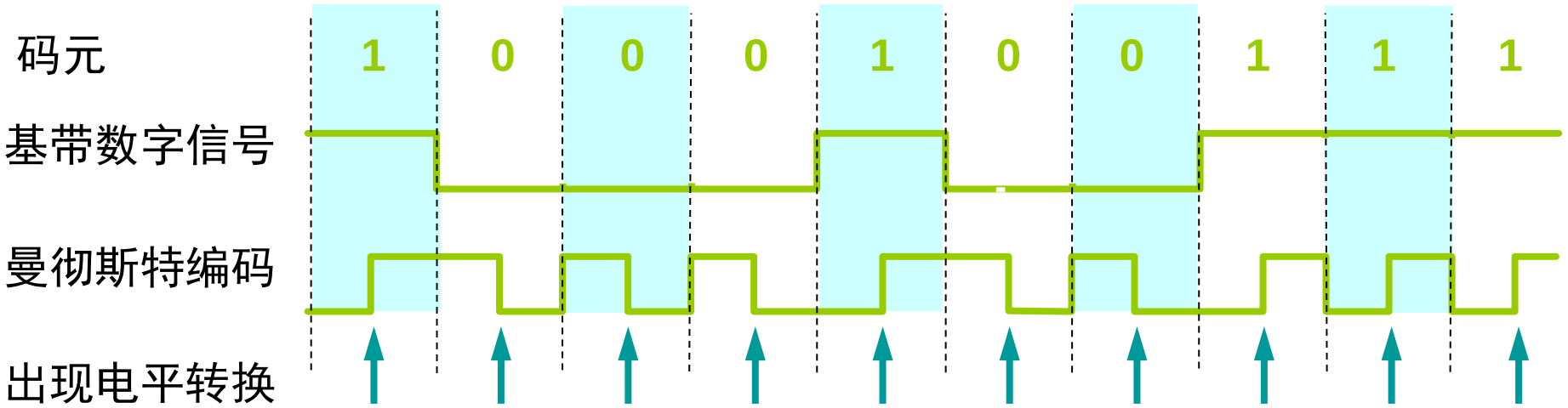
- 以太网由 Digital公司、Intel公司和Xerox公司联合开发(DIX)
- IEEE 802.3(1989年)定义了两个类别： 基带和宽带
  - 基带类使用数字信号传输数据
  - 宽带类使用模拟信号传输数据
- IEEE将基带类划分为5个不同的标准： 10Base5、10Base2、10Base-T、1Base5和100base-T。
  - 开头数字指明了数据传输速率； 最后的数字或字母(5，2，T)指明了最大电缆长度或电缆的类别； base指明的是基带传输。
- IEEE定义了一个宽带类标准：10Broad36，用于小区网络连接。

IEEE 802.3标准的发展

- 1982年　802.3　10Base5
- 1985年　802.3a　10Base2
- 1990年　802.3i　10BaseT
- 1993年　802.3j　10BaseF
- 1995年　802.3u　100BaseT
- 1997年　802.3x　全双工以太网
- 1998年　802.3z　1000BaseX
- 2000年　802.3ab　1000BaseT
- 2002年　802.3ae　10G

以太网数据编码

- 以太网发送的数据使用曼彻斯特(Manchester)编码



4.5.1 以太网访问模式

- 冲突：在LAN中，多个用户在没有任何控制的情况下同时访问一条线路时，会存在由于不同信号叠加而相互破坏的情况。
- 为了使冲突发生的可能性最小，需要有一种机制来协调通信。
- 以太网的媒体访问控制机制—带有冲突检测的载波侦听多路访问(CSMA/CD—Carrier Sense Multiple Access with Collision Detection)
- CSMA/CD的发展：
  - MA →CSMA → CSMA/CD

多路访问(MA)

- 多路访问(MA)：“不听就说”
  - 无通信管制
  - 依赖于响应，判断数据帧是否被链路上的其它通信破坏

载波帧听多路访问(CSMA)

- 载波帧听多路访问(CSMA)：“先听后说”
  - 首先监听链路上是否已经存在通信。
  - 由于存在传输延迟，还是会出现冲突
- CSMA可分为三种：依据避让时间的选择方法
  - 非坚持CSMA
  - 坚持CSMA
  - P-坚持CSMA

CSMA算法

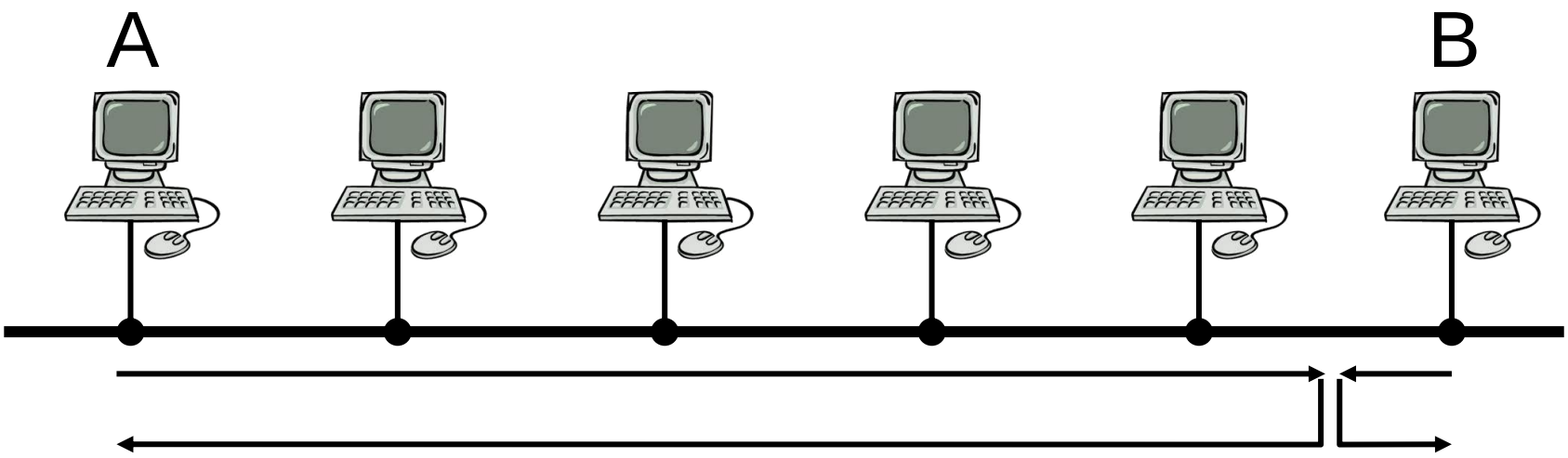
- 非坚持CSMA的算法如下：
  - 如果链路是空闲的，则可以发送。
  - 如果链路是忙的，则延迟一段时间。重发延迟时间在一个时间范围内是随机的。
- 坚持CSMA的算法如下：
  - 如果链路是空闲的，则可以发送。
  - 如果链路是忙的，则继续侦听，直到检测到链路空闲，立即发送。
  - 如果有冲突则等待一个随机的时间。
- *P*-坚持CSMA的算法如下：
  - 如果链路是空闲的，则以*P*的概率发送，而以(1-*P*)的概率延迟一个时间单位。
  - 如果链路是忙的，继续侦听直至链路空闲。

带有冲突检测的载波监听多路访问(CSMA/CD)

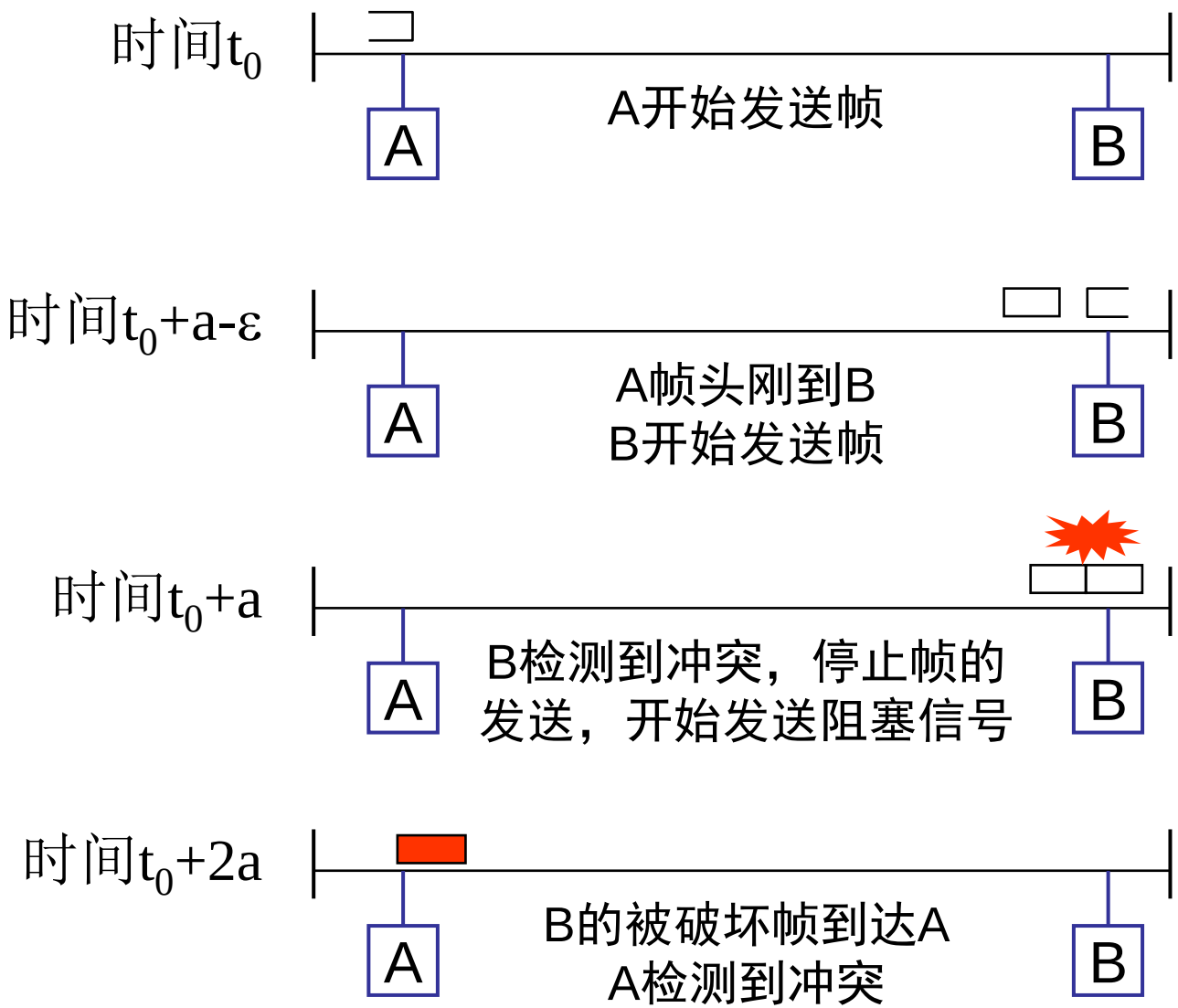
- CSMA/CD：“边听边说”
  - 带有冲突检测的载波侦听多路访问
  - 在传输的时候继续监听链路是否发生冲突
- CSMA/CD的算法描述：
  - 如果链路是空闲的，则可以发送并同时检测冲突。
  - 如果链路是忙的，则继续侦听，直到检测到链路空闲。
  - 如果在发送过程中检测到冲突，则停止当前帧的发送，发阻塞信号，等待一段选定的时间(由退避算法决定)。
- 检测冲突所需的时间称为冲突检测窗口

冲突示意图

- 冲突检测窗口等于任意两个站点之间最大的传播延迟的两倍。



冲突检测窗口


时间t_0
A开始发送帧
A
B
时间t_0+a-ε
A帧头刚到B
B开始发送帧
A
B
时间t_0+a
B检测到冲突，停止帧的
发送，开始发送阻塞信号
A
B
时间t_0+2a
B的被破坏帧到达A
A检测到冲突
A
B

最短帧长

- 最短帧长的计算公式：

$$F_{min}=2R\times D/V$$

  - F_{min}：最短数据帧长（bit）
  - R：数据传输速率（bps）
  - D：任意两站点间的最大距离（m）
  - V：电子传播速度（m/s）

例题

- 电缆长2km，数据传输率10Mbps，信号传播速度200m/μs，求允许的最短帧长？
- 答：

$$F_{min}=2\times 10b/\mu s\times 2000m\div 200m/\mu s=200bit$$

理想情况下10Mbps以太网的
最大传输距离(系统跨距)是多少？

10Mbps以太网规定参数 记

- 冲突窗口为51.2μs
  - 51.2μs正好是10Mbps以太网发送64字节(即512位)的时间
- 2个站点最多可以经过4个中继器
- 每段长度最长为500m

⇒这表明在2500m电缆的传播时延(信号在电缆上的传播速度是200m/μs)加上4个中继器的双向时延要小于51.2μs

CSMA/CD退避算法

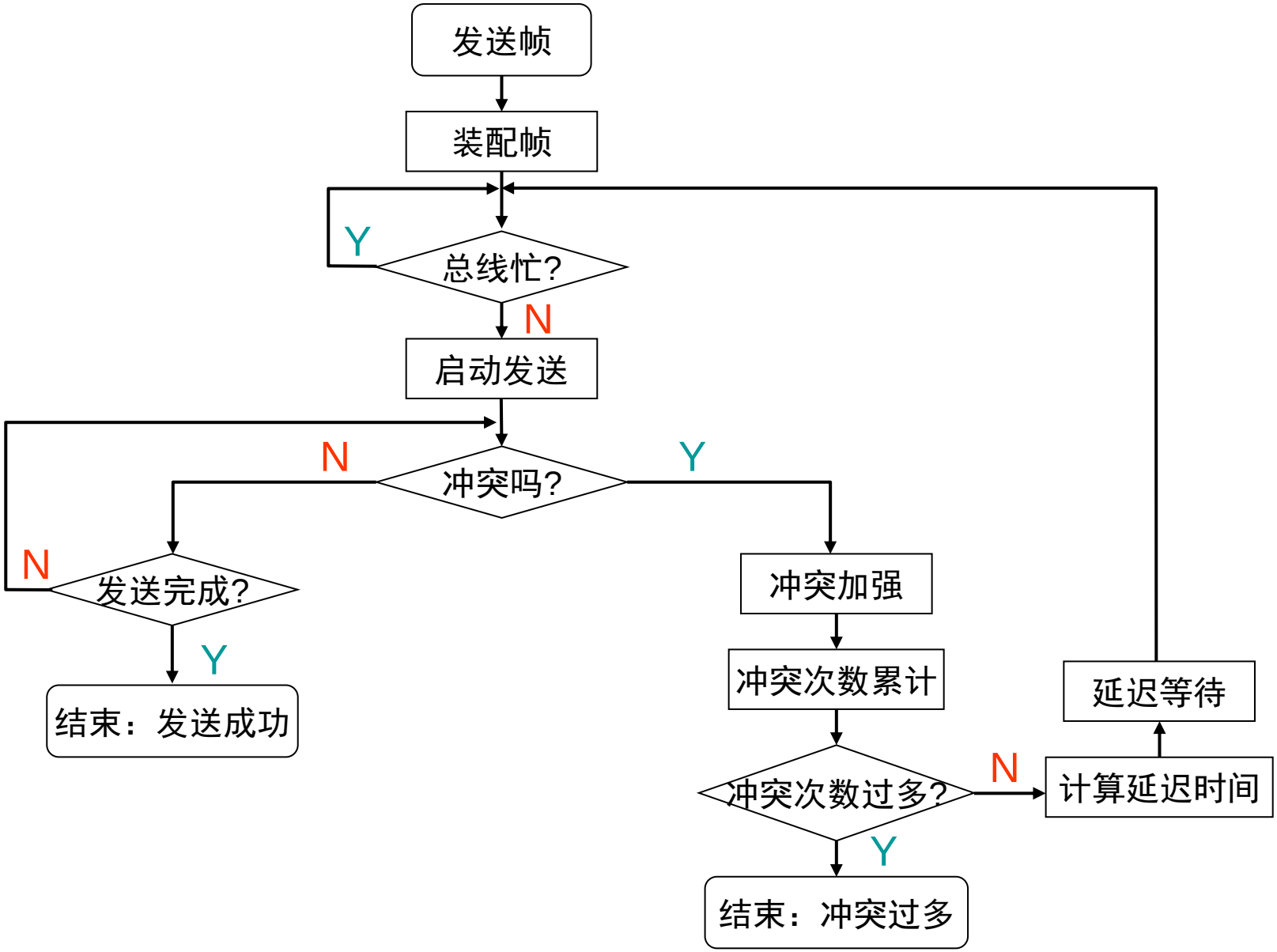
- 对每一个帧，当第一次发生冲突时，设置参数L=2
- 退避间隔取1到L个时间片中的一个随机数。
  - 一个时间片等于链路上最大传输延迟的2倍
- 当帧重复发生一次冲突时，则将参数L加倍。L的最大值为1024。即当L增加到1024时，L不再增加
- 帧的最大重传次数为16，超过这个次数，则该帧不再重传，并报告出错
- 发送站等待的时间：

$$t=R\times 2T$$

其中：

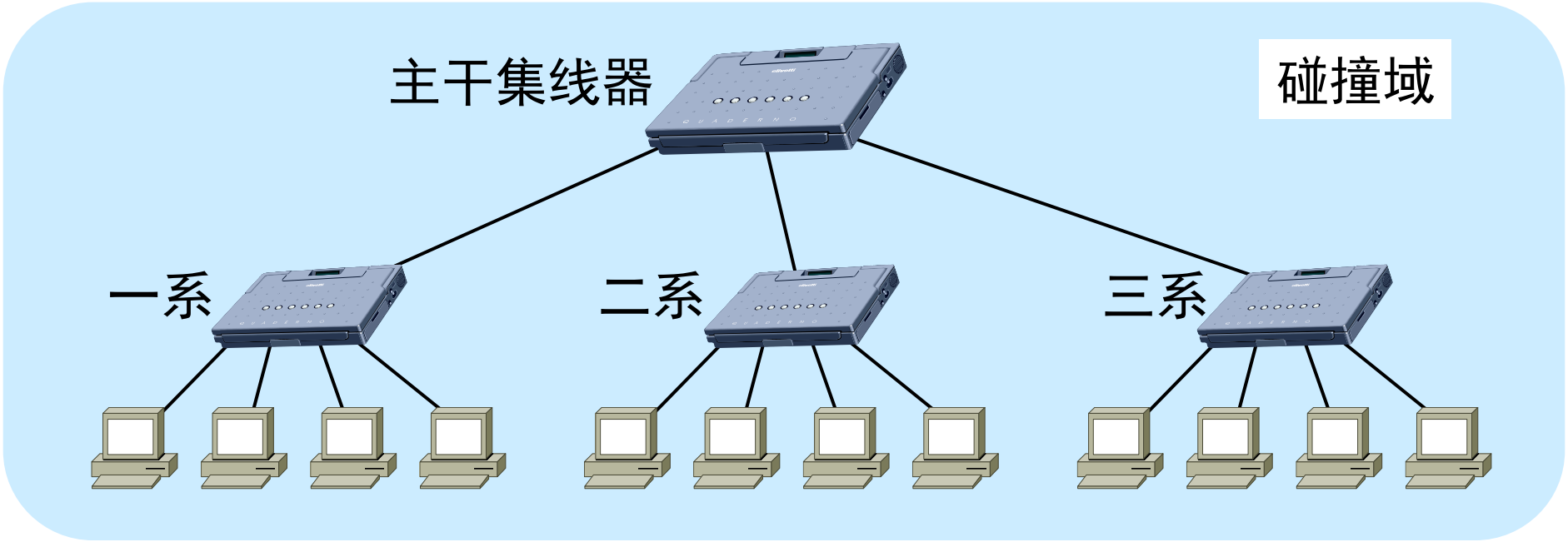
- R是0～$2^{\min(k,10)}$之间的随机数
- k是冲突次数，最大为16
- T为任意两点之间最大的传播延迟

以太网站点发送帧过程


发送帧
装配帧
总线忙?
Y
N
启动发送
冲突吗?
N
Y
发送完成?
N
Y
结束：发送成功
冲突加强
冲突次数累计
冲突次数过多?
N
Y
计算延迟时间
延迟等待
结束：冲突过多

用集线器互连网络

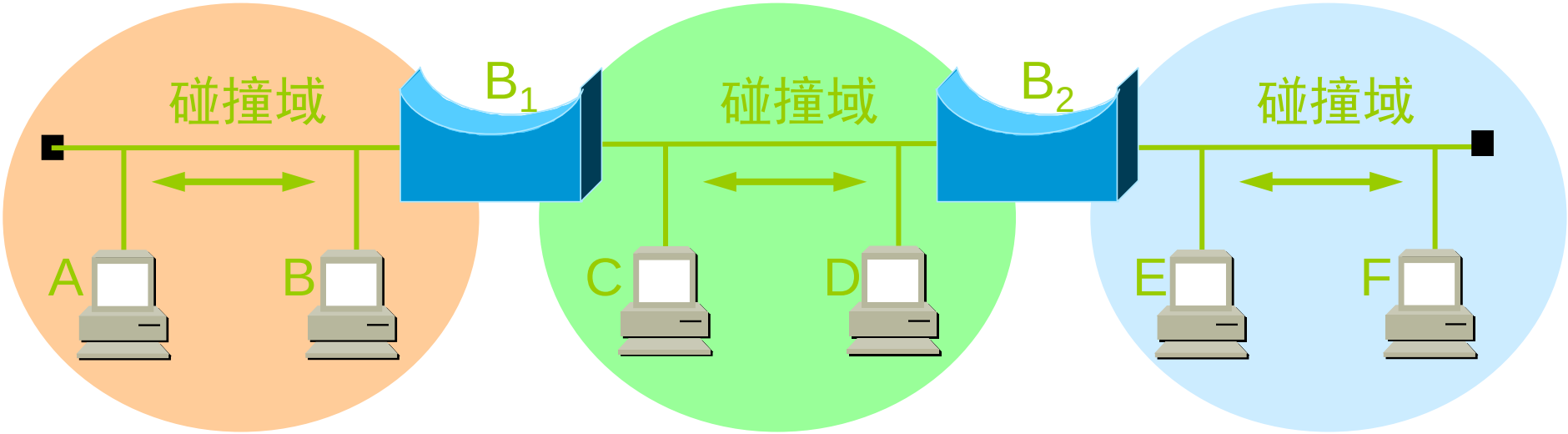
- 用集线器组成更大的局域网都在一个碰撞域中



一个更大的碰撞域

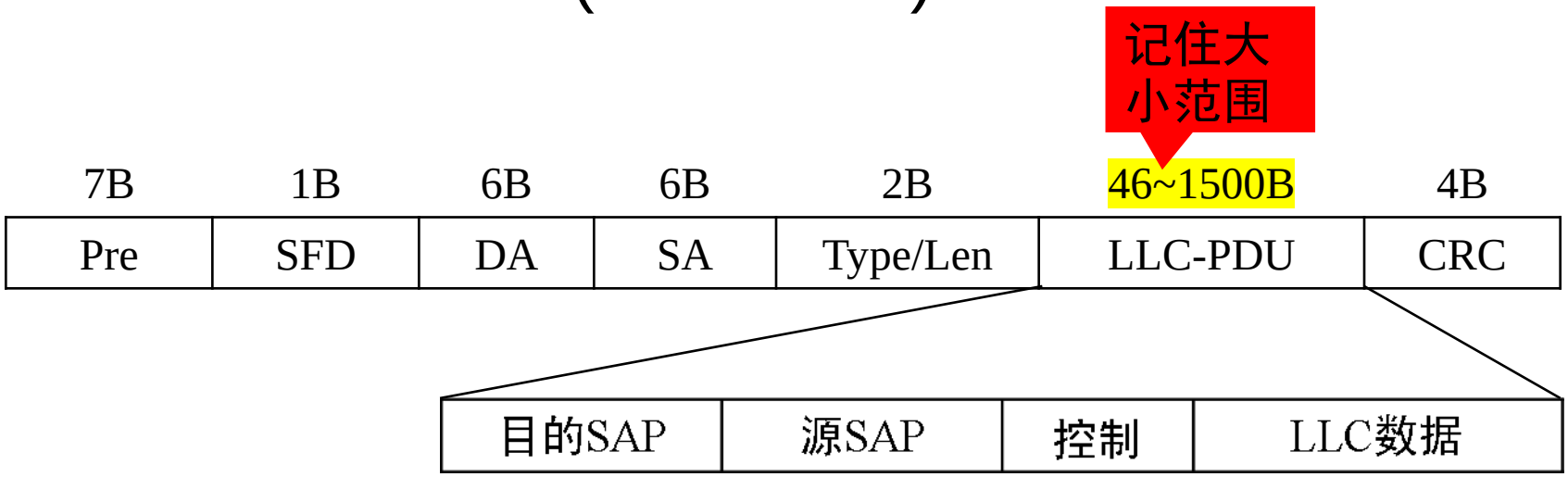
用交换机互连网络

- 交换机使各网段成为隔离开的碰撞域



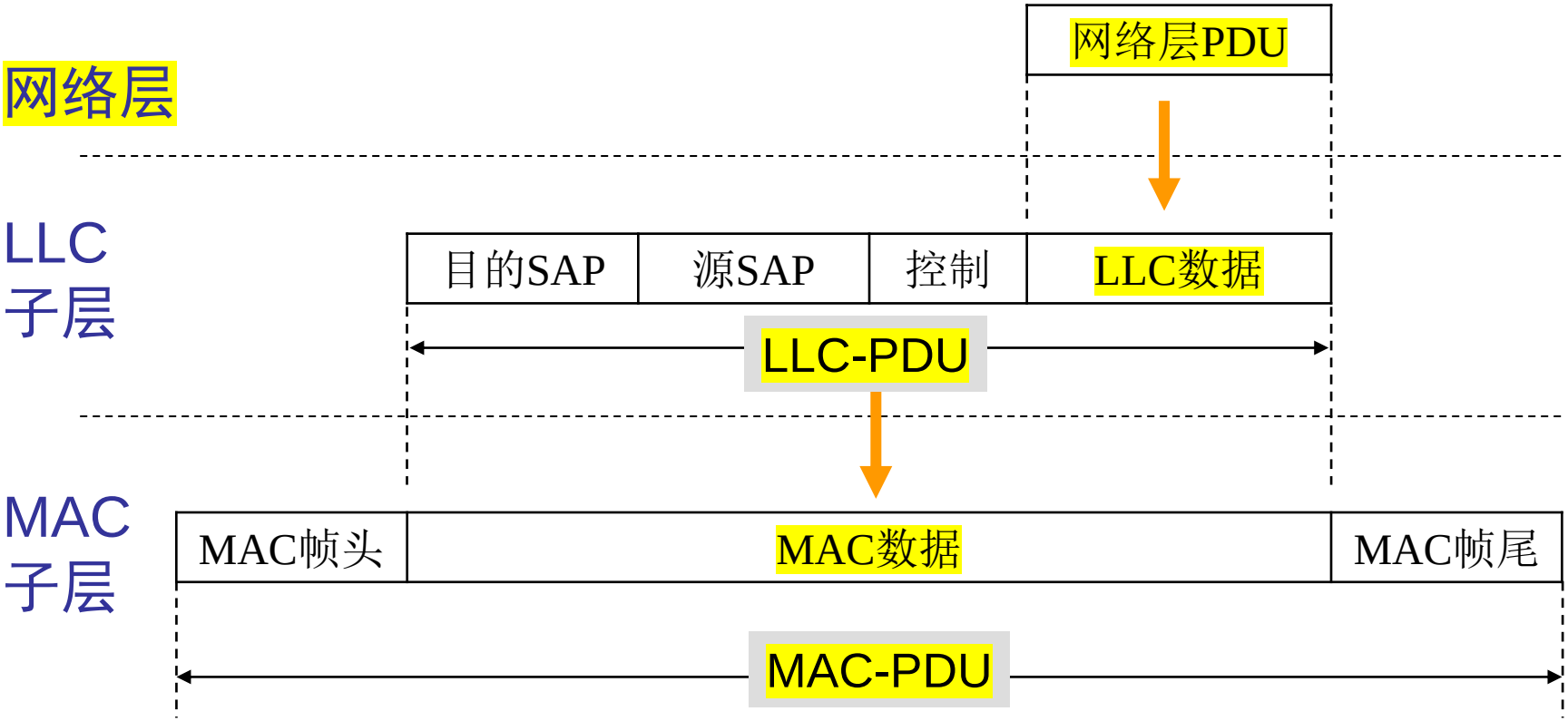
4.5.2 以太网MAC帧格式

- 以太网有多种帧格式，最常用的有2种
  - Ethernet II (RFC 894)
  - Ethernet 802.3 (RFC 1042)



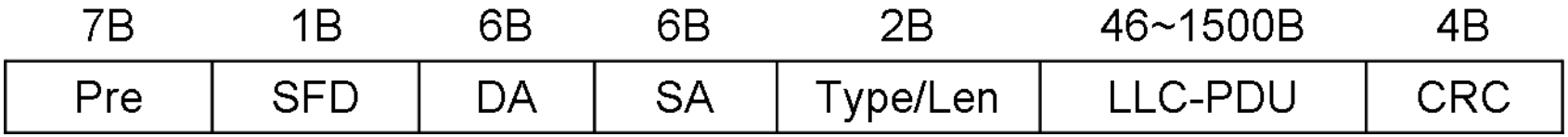
MAC帧大小范围64~1518B

逻辑链路控制LLC



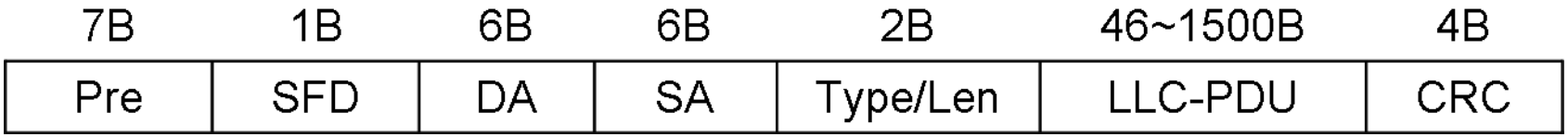
LLC-PDU与相邻层PDU之间的关系

帧格式(1)



- Pre(preamble 前导码)：7个字节的10101010，警告系统接收即将到来的数据帧，同时使系统能够调整同步输入时钟。
- SFD(start-of-frame delimiter 帧起始分界符)：1个字节，10101011，帧起始分界符标记了帧的开始。SFD通知接收方后面所有的内容都是数据。

帧格式(2)

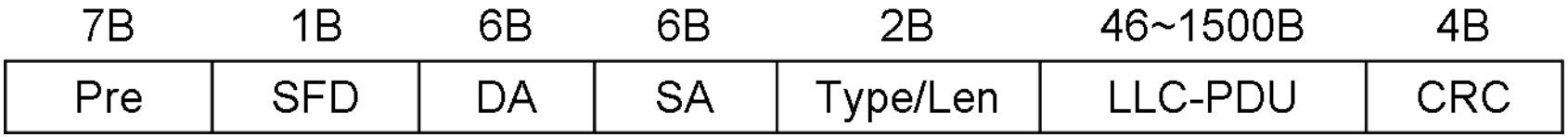


- DA(destination address目的地址)：6个字节，标记了数据帧下一个节点的物理地址。有三种类型：
  - 单播地址(unicast address)
  - 组播地址(multicast address)
  - 广播地址(broadcast address)

地址字节数：
MAC：6字节
网络：4字节
传输：2字节

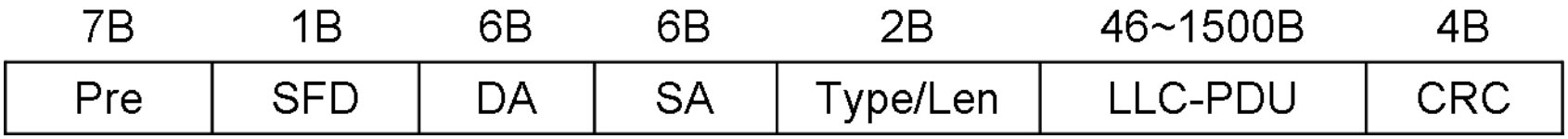
- SA(source address源地址)：6个字节。它包含了最后一个转发此帧的设备的物理地址。也是上个节点的物理地址。必须为单播地址。

帧格式(3)



- Type/Len(type/length LLC-PDU的长度/类型)：2个字节。0～1500保留为长度域值，1536～65535保留为类型域值(0x0600～0xFFFF)
  - Ethernet II作为类型字段，用于表明数据是哪种类型协议交下来的，0x0800—IP，0x0806—ARP，0X8137-Novell IPX,。
  - 802.3作为长度字段，数据段中数据的字节数46~1500。

帧格式(4)



- LLC-PDU(数据)：用户数据，最长1500字节
  - PAD：0~46字节。为了使CSMA/CD协议的正常操作，需要一个最小帧长度64字节。当帧长度小于64字节时，必须进行填充。
- CRC：4字节。校验范围为DA~LLC-PDU。生成多项式为CRC-32：

$$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$$

MAC地址

- 以太网地址长度6个字节
  - 前3个字节标识厂商，由IEEE RA负责分配
    - CISCO：00-00-0C
    - IBM：08-00-5A
  - 后3个字节为系列号，由厂商分配
- IANA规定，将01:00:5E:00:00:00~01:00:5E:7F:FF:FF用于IP组播地址到以太网组播地址的映射。

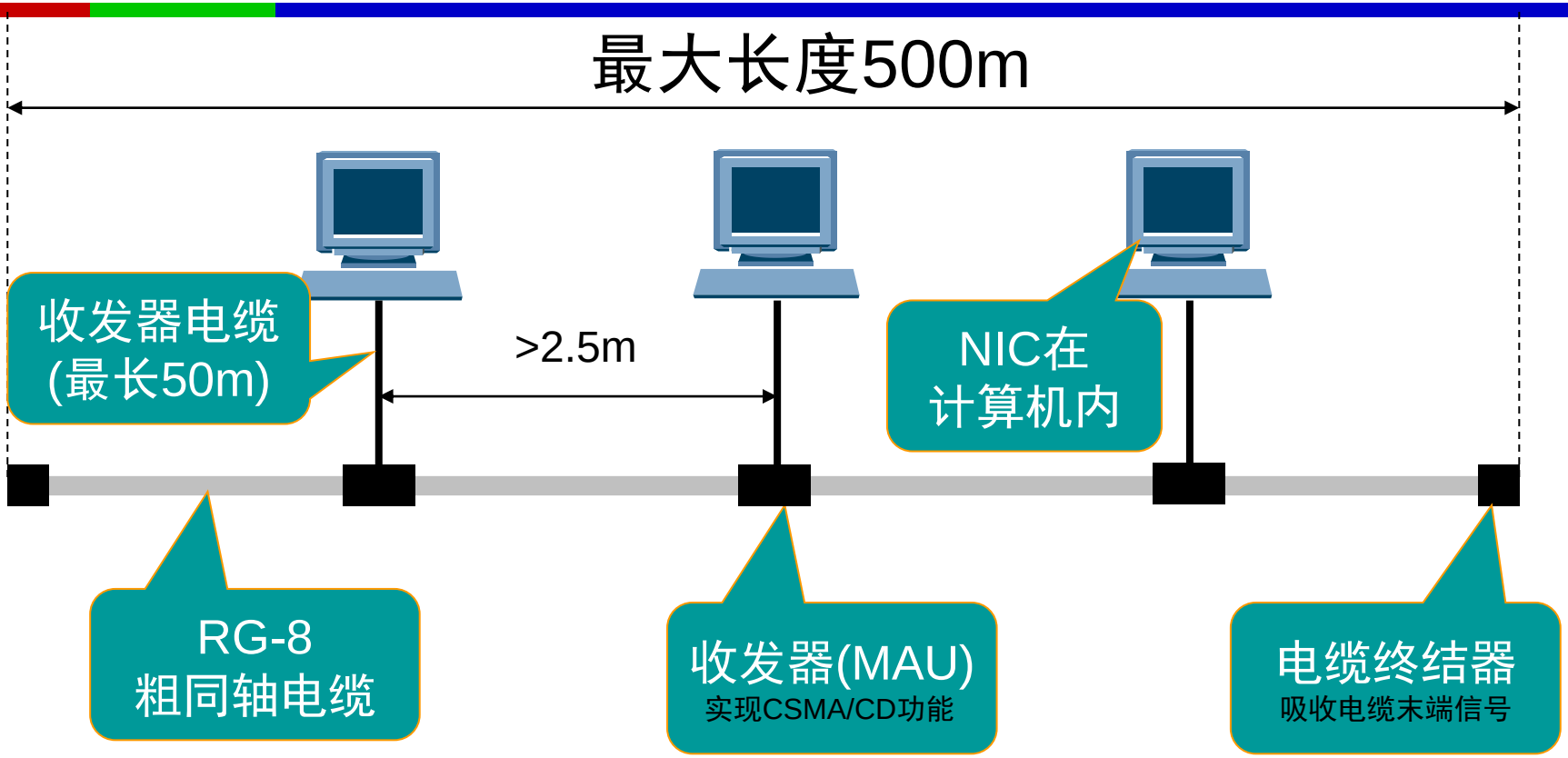
以太网运行参数

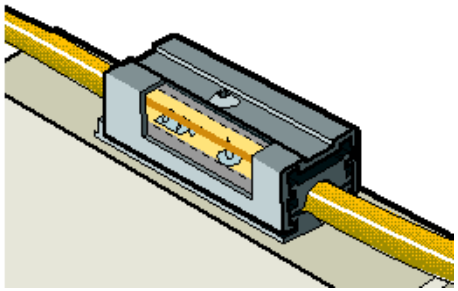
参数	值	参数	值
比特时间	100ns	冲突后退限制	10
冲突窗口	51.2μs	阻塞帧	4字节
帧间间隔	9.6μs	最大帧长度	1518字节
冲突重发次数	16	最小帧长度	64字节

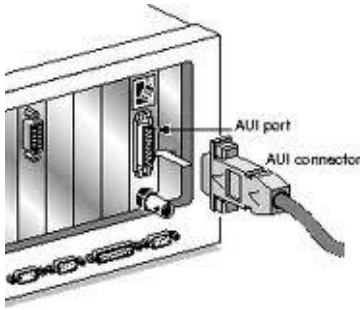
4.5.3 以太网种类

Specification	Cable Type
10BaseT	Unshielded Twisted Pair
10Base2	Thin Coaxial
10Base5	Thick Coaxial
100BaseT	Unshielded Twisted Pair
100BaseFX	Fiber Optic
100BaseBX	Single mode Fiber
100BaseSX	Multimode Fiber
1000BaseT	Unshielded Twisted Pair
1000BaseFX	Fiber Optic
1000BaseBX	Single mode Fiber
1000BaseSX	Multimode Fiber

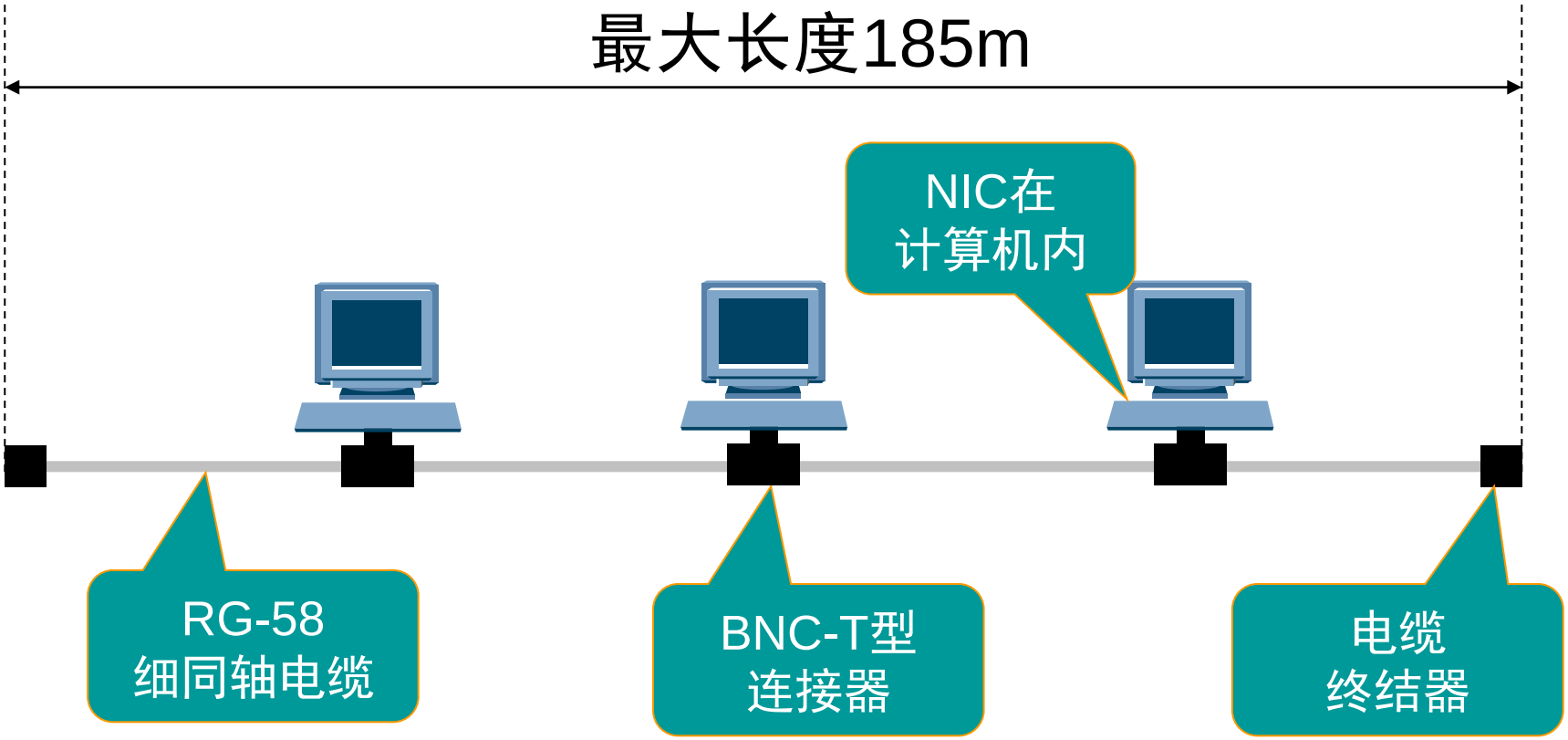
10Base5：粗缆以太网，10Mbps



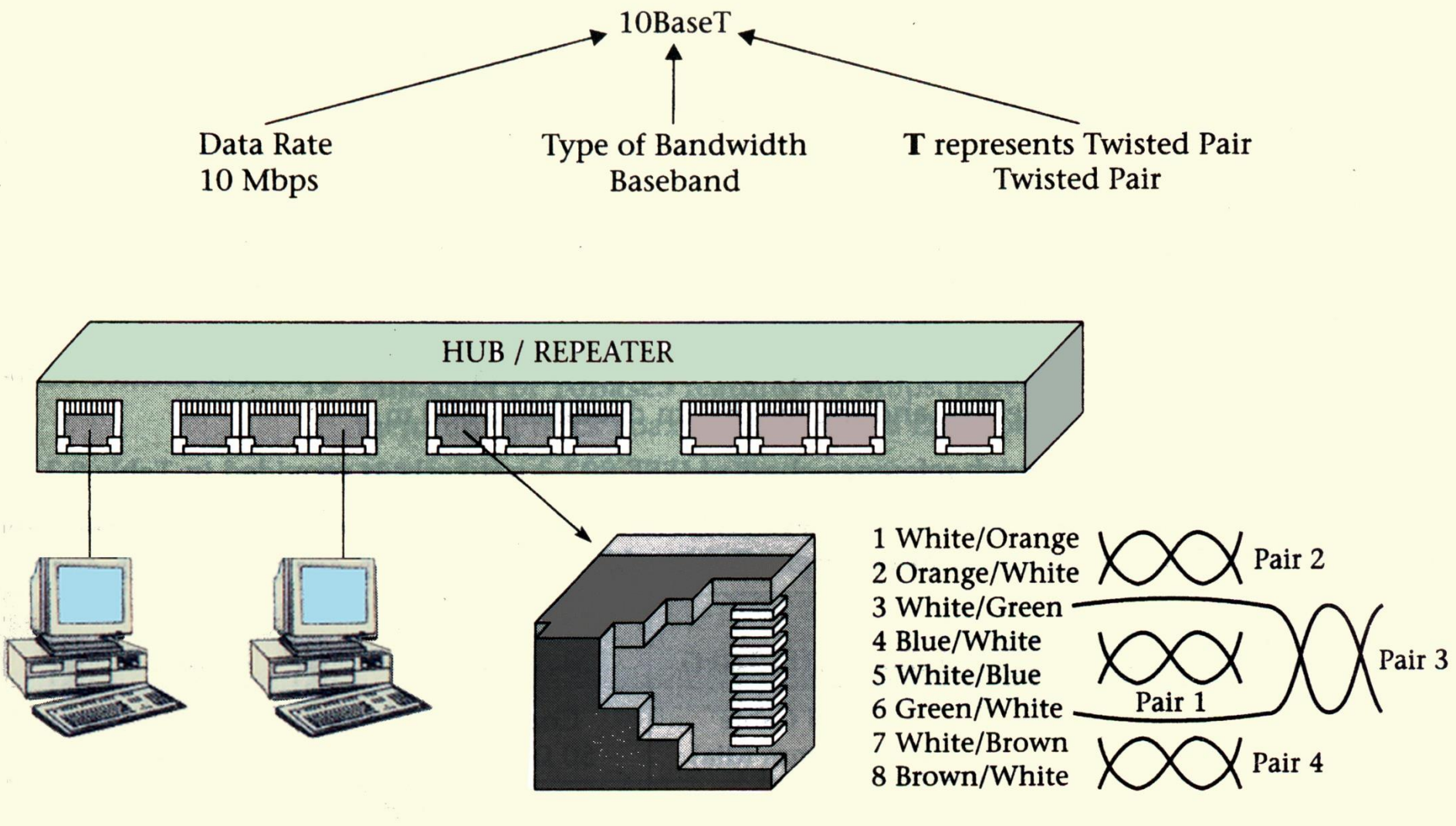




10Base2： 细缆以太网



10BaseT：双绞线以太网->星型拓扑



100Base-T：快速以太网

- 与10Base-T相同点：（1）均为星型拓扑（2）相同的MAC协议和帧格式。
- 与10Base-T的不同点：（1）是编码系统，为了适应100Mbps的速率要求改为4B/5B编码与MLT-3编码组合方式；（2）计算机和集线器间定义了发送和接收两条链路。
- 保持最短帧长不变（最短64字节），一个网段的最大电缆长度100m，因此争用期从原来的51.2 μs改为5.12 μs,帧间时间间隔从原来的9.6μs改为现在的 0.96μs。

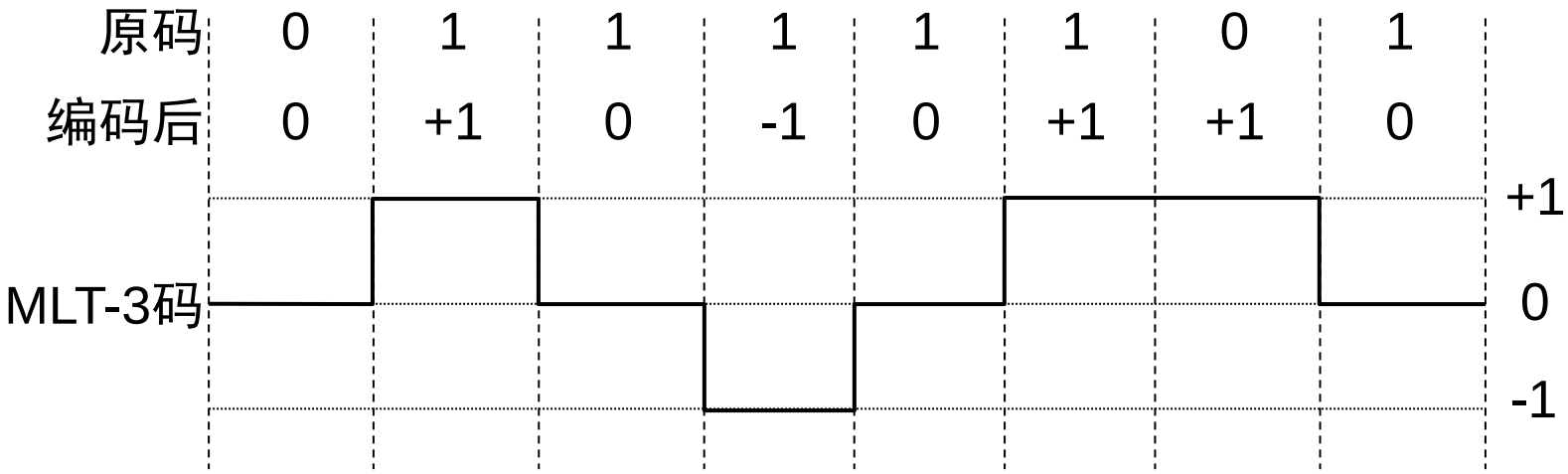
4B/5B编码

- 用5bit的二进制数来表示4bit二进制数，映射方式如右表所示
- 4位二进制共有16种组合，5位二进制共有32种组合，如何从32种组合种选取16种来使用呢？这里需要满足两个规则：
  - 每个5比特码组中不含多于3个“0”
  - 或者5比特码组中包含不少于2个“1”

十进制数	4位二进制数	4B/5B码
0	0000	11110
1	0001	01001
2	0010	10100
3	0011	10101
4	0100	01010
5	0101	01011
6	0110	01110
7	0111	01111
8	1000	10010
9	1001	10011
10	1010	10110
11	1011	10111
12	1100	11010
13	1101	11011
14	1110	11100
15	1111	11101

MLT-3编码规则（属于双极性编码）

- MLT-3：Multi-Level Transmit-3，多电平传输编码
- 规则：有3种电平，-1、0、+1
  - 如果输入为“0”，则电平保持不变
  - 如果输入为“1”，则产生跳变，又分2种情况：
    - 如果前一个输出是“+1”或“-1”，则下一输出为“0”
    - 如果前一个输出为“0”，则下一个输出值与最近一个非“0”相反



快速以太网运行参数

参数	值	参数	值
比特时间	10ns	冲突后退限制	10
冲突窗口	5.12μs	阻塞帧	4字节
帧间间隔	0.96μs	最大帧长度	1518字节
冲突重发次数	16	最小帧长度	64字节

千兆以太网运行参数

参数	值	参数	值
比特时间	0.1ns	冲突后退限制	10
冲突窗口	4.096μs	阻塞帧	4字节
帧间间隔	0.096μs	最大帧长度	1518字节
冲突重发次数	16	最小帧长度	416/520字节

4.6 无线局域网 有确认无连接

- 1990年，IEEE 802委员会成立了一个新工作组—IEEE 802.11，致力于制定无线局域网标准。
- 无线局域网(WLAN)是指以无线信道作为传输介质的计算机局域网。
- 无线局域网必须满足于无线局域网相同的要求：
  - 通信容量
  - 短距离覆盖
  - 站点互连

4.6.1 无线局域网标准

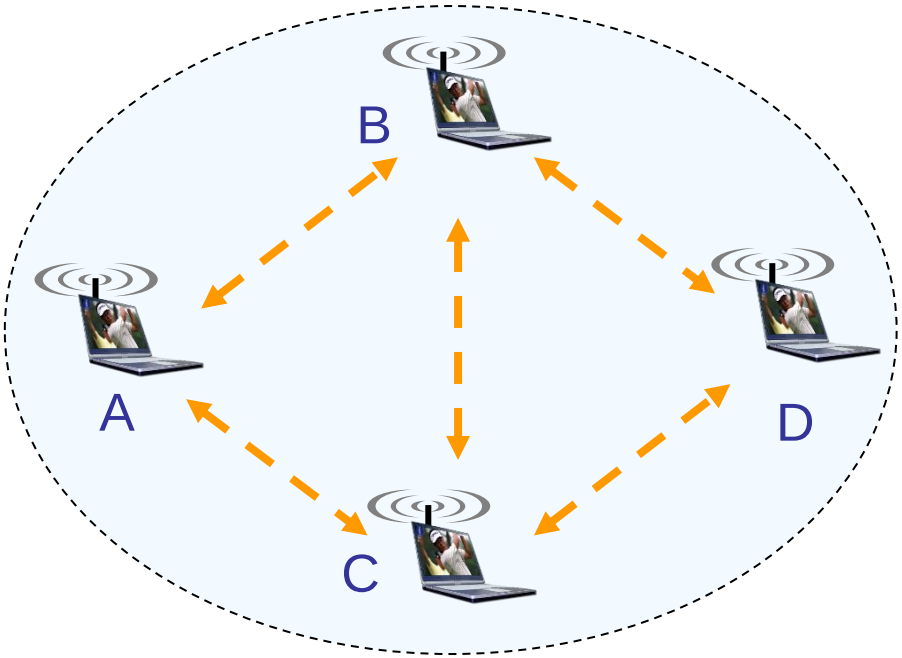
- IEEE 802.11标准定义了3种物理层介质：
  - 调频扩展频谱FHSS
  - 直接序列扩展频谱DSSS
  - 红外线IR
- 正交频分复用OFDM用于802.11a/g
- 目前，已经标准化的有：
  - 802.11b
  - 802.11a
  - 802.11g
  - 802.11n

各种标准比较

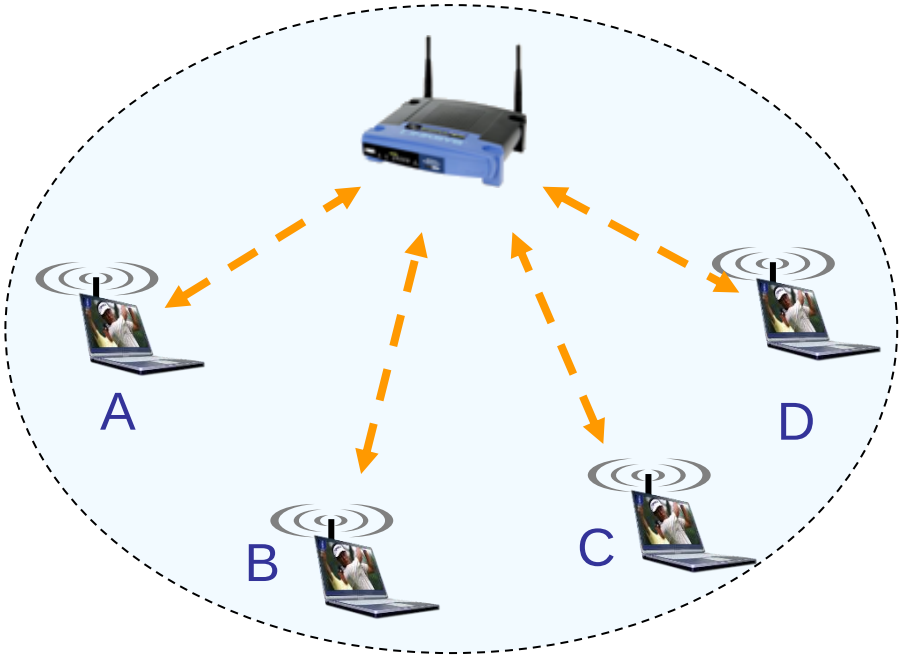
标准	802.11b	802.11g	802.11a
MAC协议	CSMA/CA		
安全机制	WEP/WPA		
工作频率	2.4GHz		5GHz
物理编码	DSSS	DSSS 或OFDM	OFDM (正交频分复用)
最高速率	11Mbps	54Mbps	54Mbps

拓扑结构

- WLAN的最小构成单位是基本服务集(Basic Service Set, BSS)，由运行相同MAC协议并竞争使用同一无线链路的站点组成

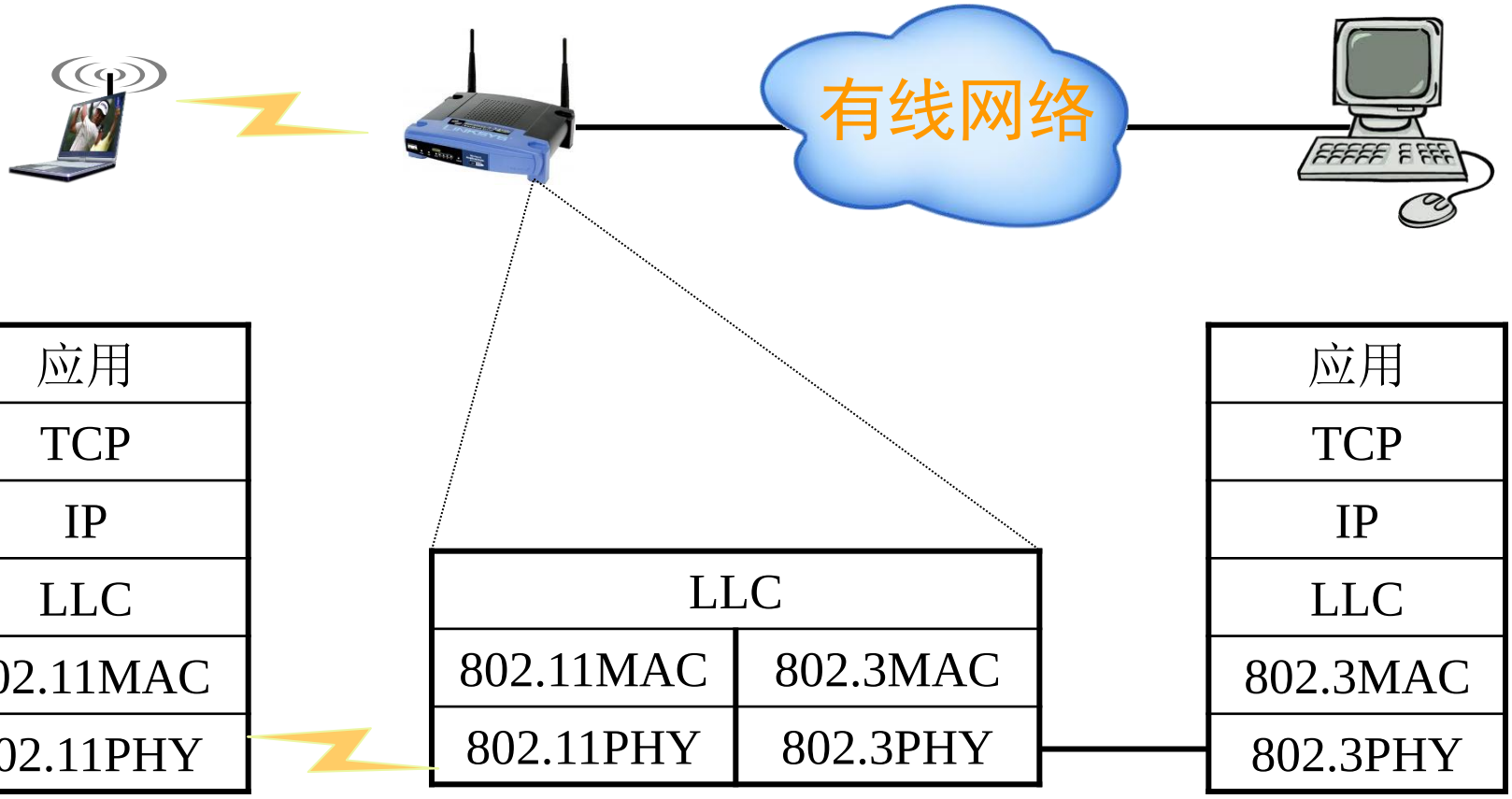


独立BSS

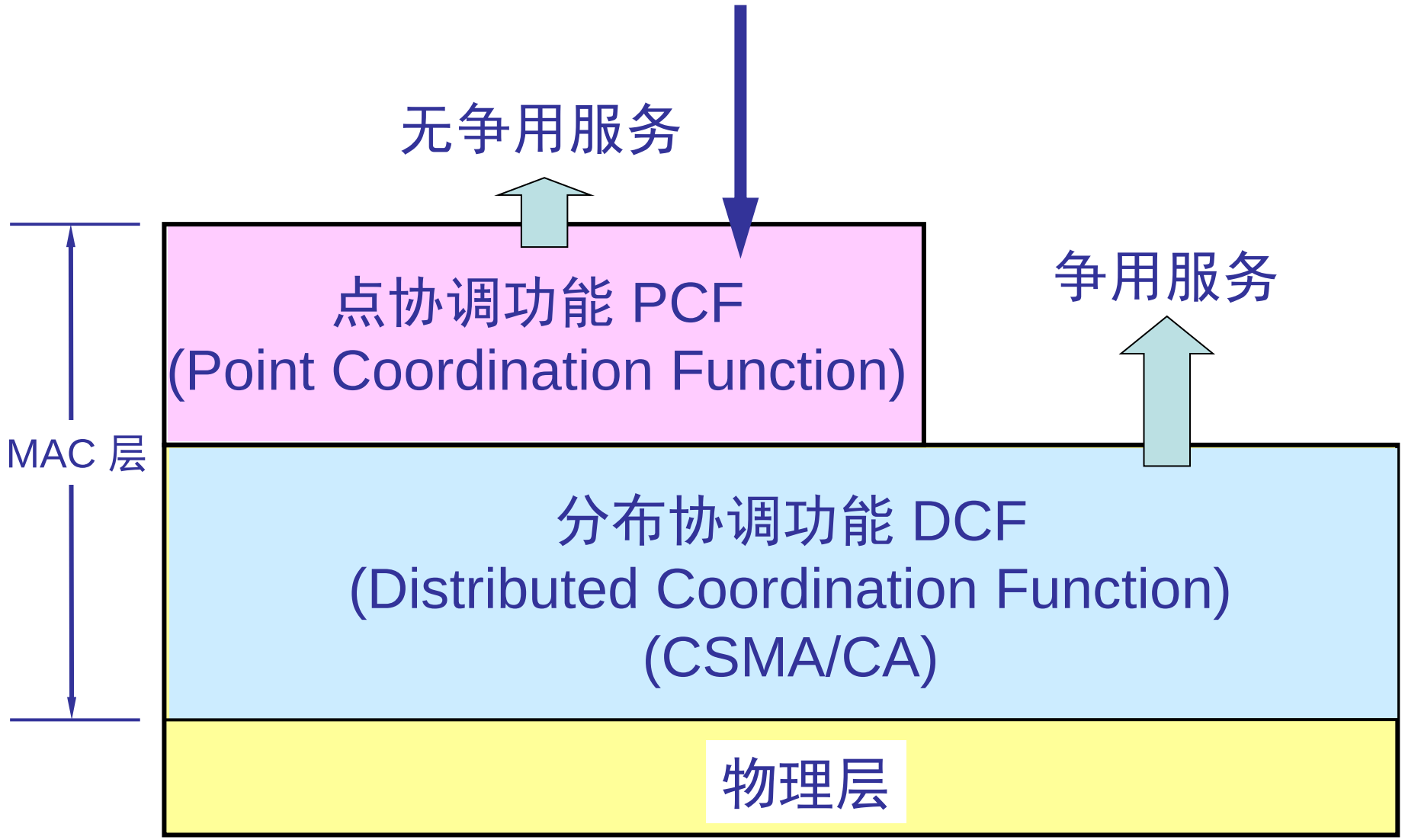


有AP的BSS

协议栈


有线网络
应用
TCP
IP
LLC
802.11MAC
802.11PHY
LLC
802.11MAC
802.3MAC
802.11PHY
802.3PHY
应用
TCP
IP
LLC
802.3MAC
802.3PHY

PCF 子层使用集中控制的接入算法把发送数据权轮流交给各个站从而避免了碰撞的产生



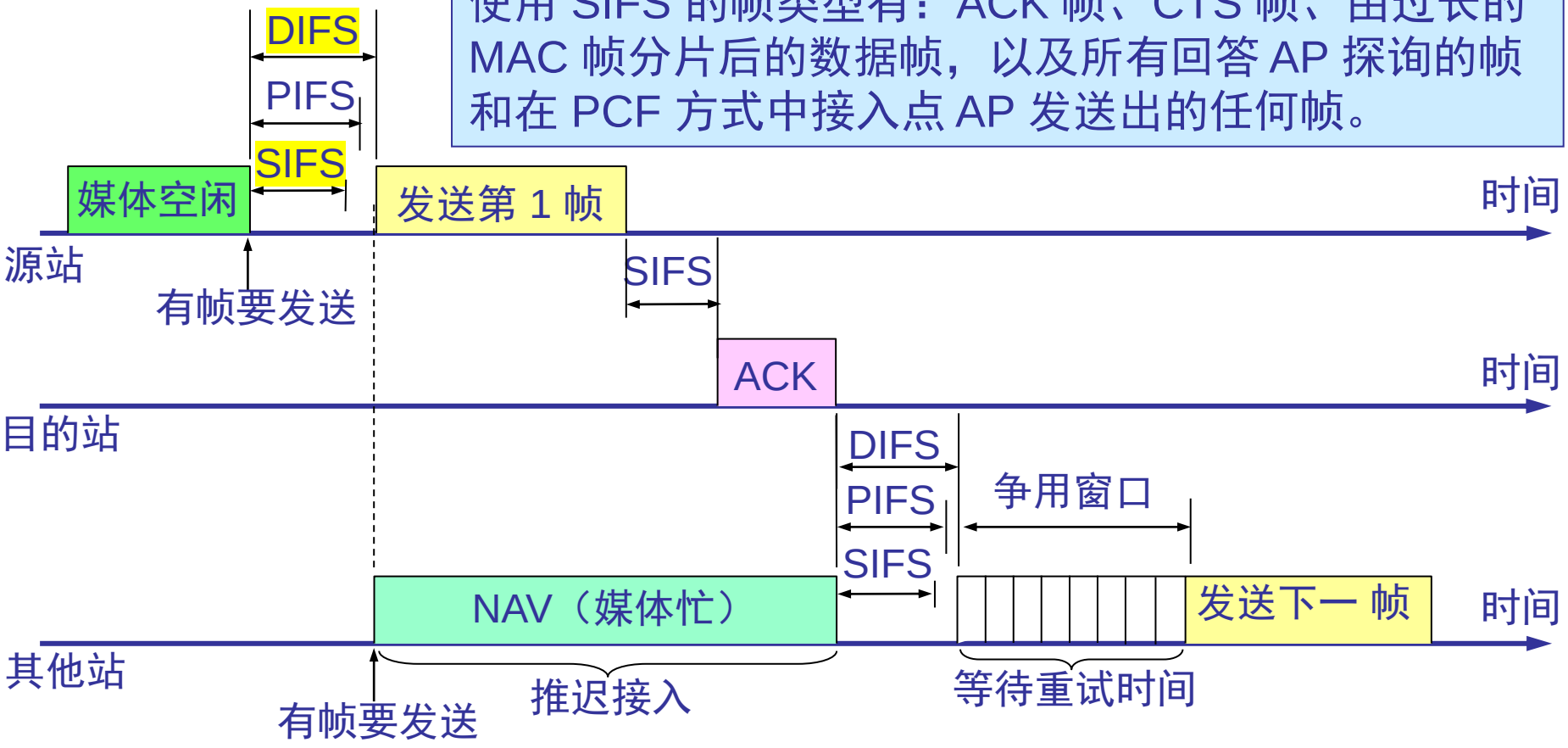
帧间间隔 IFS

- 所有的站在完成发送后，必须再等待一段很短的时间（继续监听）才能发送下一帧。这段时间的通称是帧间间隔 IFS (InterFrame Space)。
- 帧间间隔长度取决于该站欲发送的帧的类型。高优先级帧需要等待的时间较短，因此可优先获得发送权。
- 若低优先级帧还没来得及发送而其他站的高优先级帧已发送到媒体，则媒体变为忙态因而低优先级帧就只能再推迟发送了。这样就减少了发生碰撞的机会。

三种帧间间隔

SIFS，即短(Short)帧间间隔，是最短的帧间间隔，用来分隔开属于一次对话的各帧。一个站应当能够在这段时间内从发送方式切换到接收方式。

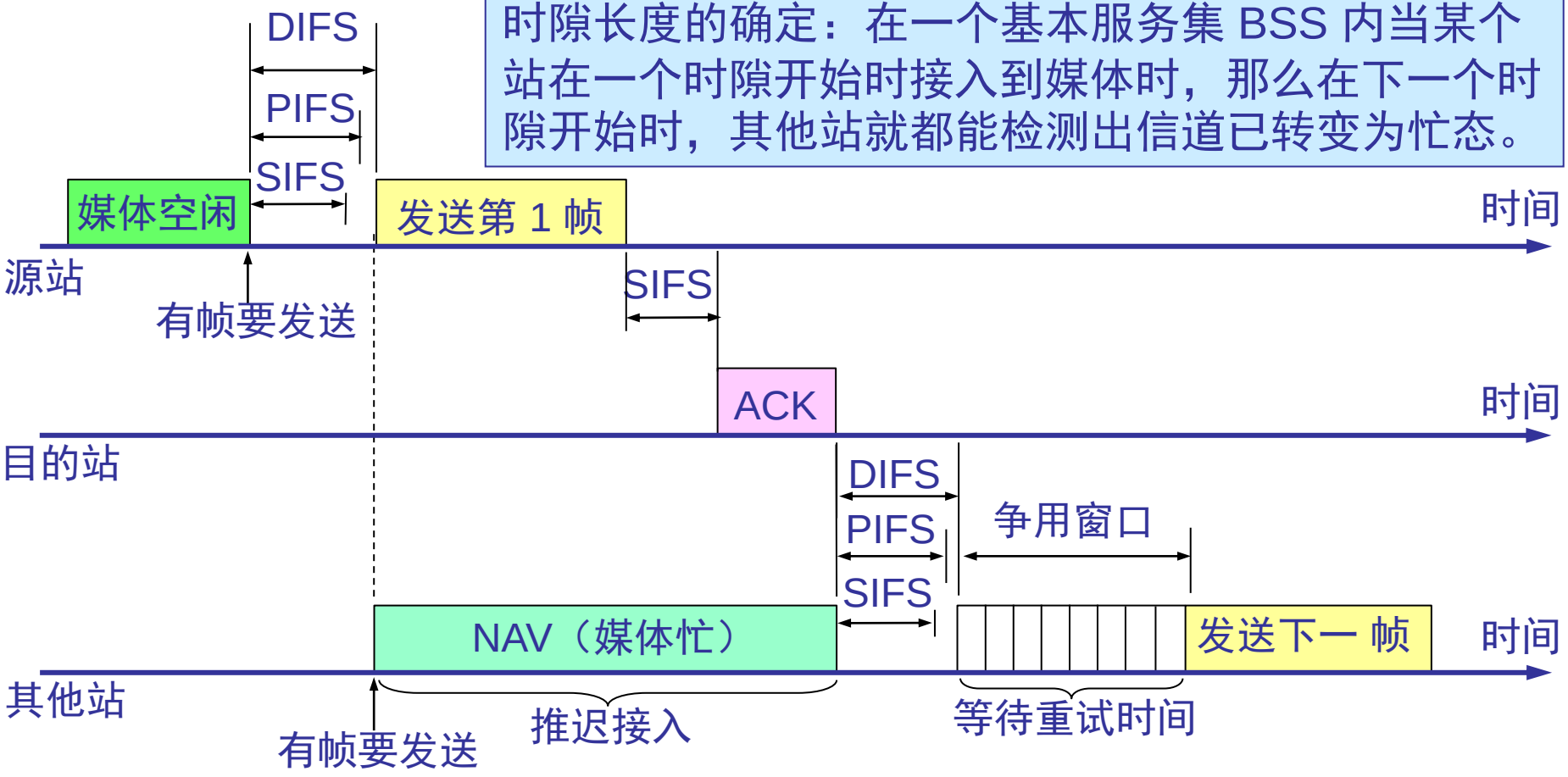
使用 SIFS 的帧类型有：ACK 帧、CTS 帧、由过长的 MAC 帧分片后的数据帧，以及所有回答 AP 探询的帧和在 PCF 方式中接入点 AP 发送出的任何帧。



三种帧间间隔

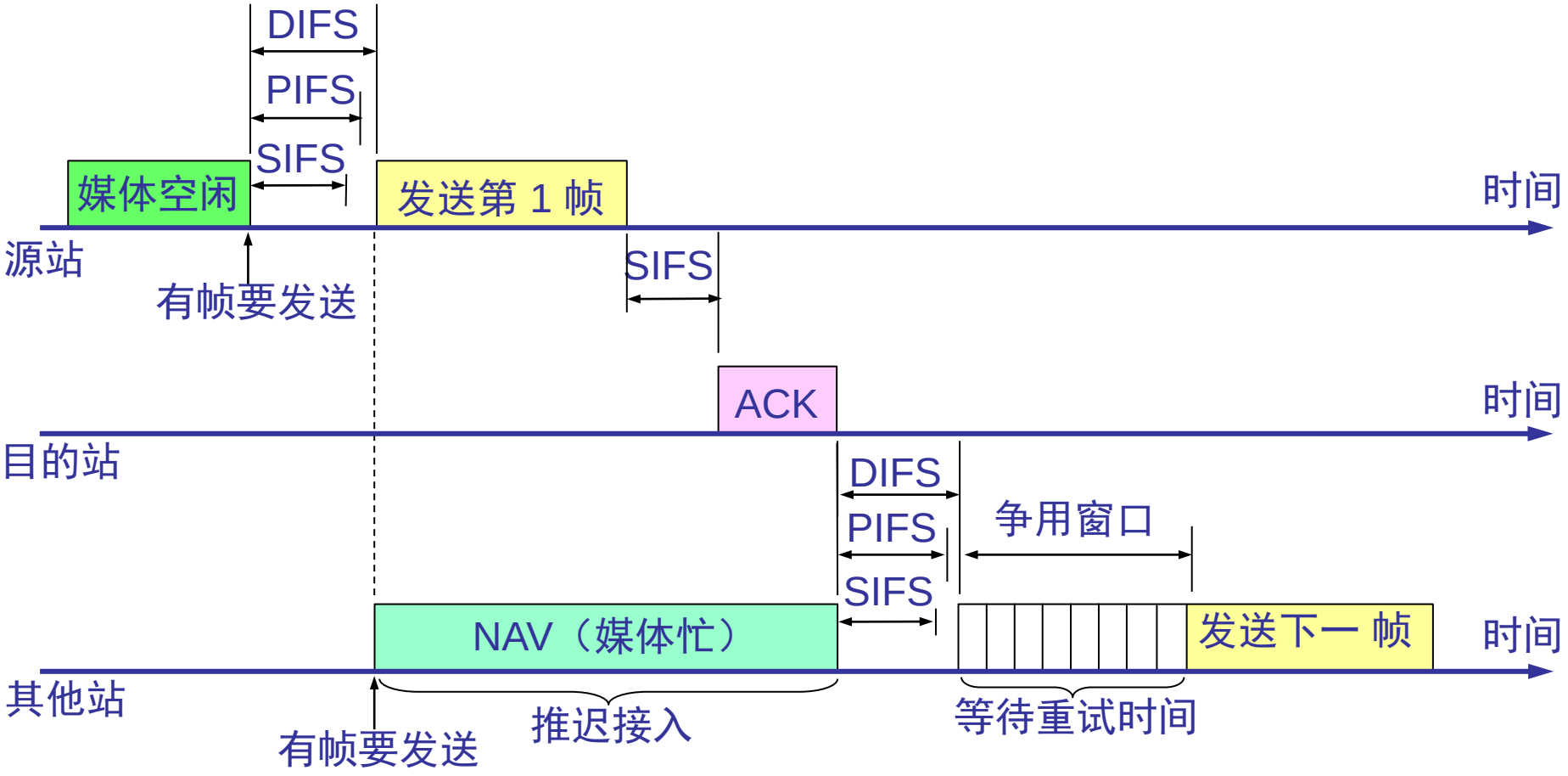
PIFS，即点协调功能帧间间隔，它比 SIFS 长，是为了在开始使用 PCF 方式时（在 PCF 方式下使用，没有争用）优先获得接入到媒体中。PIFS 的长度是 SIFS 加一个时隙(slot)长度。

时隙长度的确定：在一个基本服务集 BSS 内当某个站在一个时隙开始时接入到媒体时，那么在下一个时隙开始时，其他站就都能检测出信道已转变为忙态。

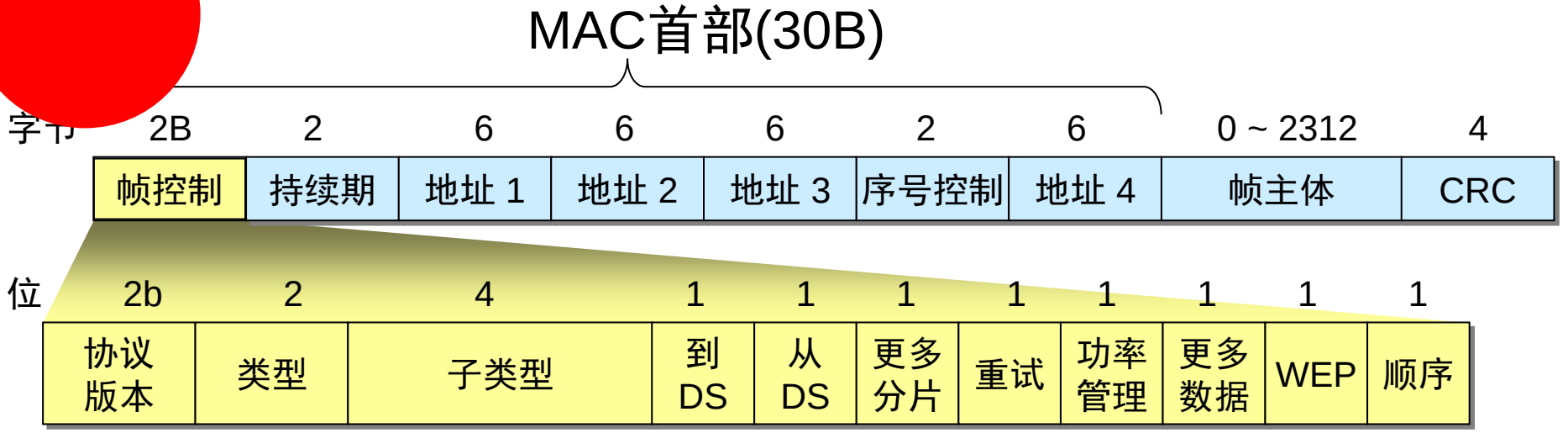


三种帧间间隔

DIFS，即分布协调功能帧间间隔（最长的 IFS），在 DCF 方式中用来发送数据帧和管理帧。DIFS 的长度比 PIFS 再增加一个时隙长度。



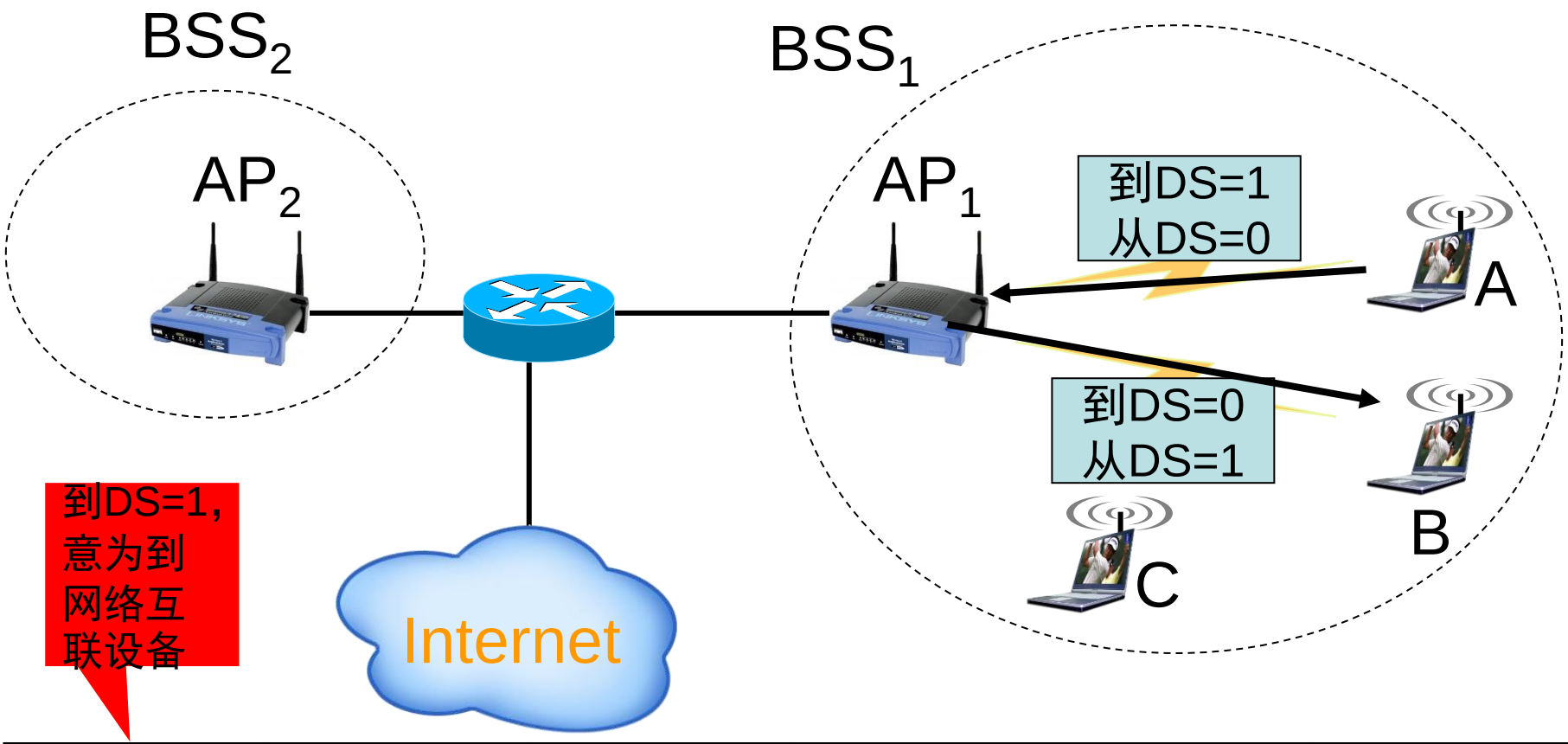
4.6.2 802.11协议族MAC帧结构



- 协议版本：0
- 类型：管理、控制、数据帧
- 子类型：认证、连接、探测帧…
- 持续期：高位0时表示持续期，单位微秒
- 序号控制：序号子字段12位，分片子字段4位

到DS	从DS	地址1	地址2	地址3	地址4
0	1	目的地址	AP地址	源地址	--
1	0	AP地址	源地址	目的地址	--

示例：A向B发送数据



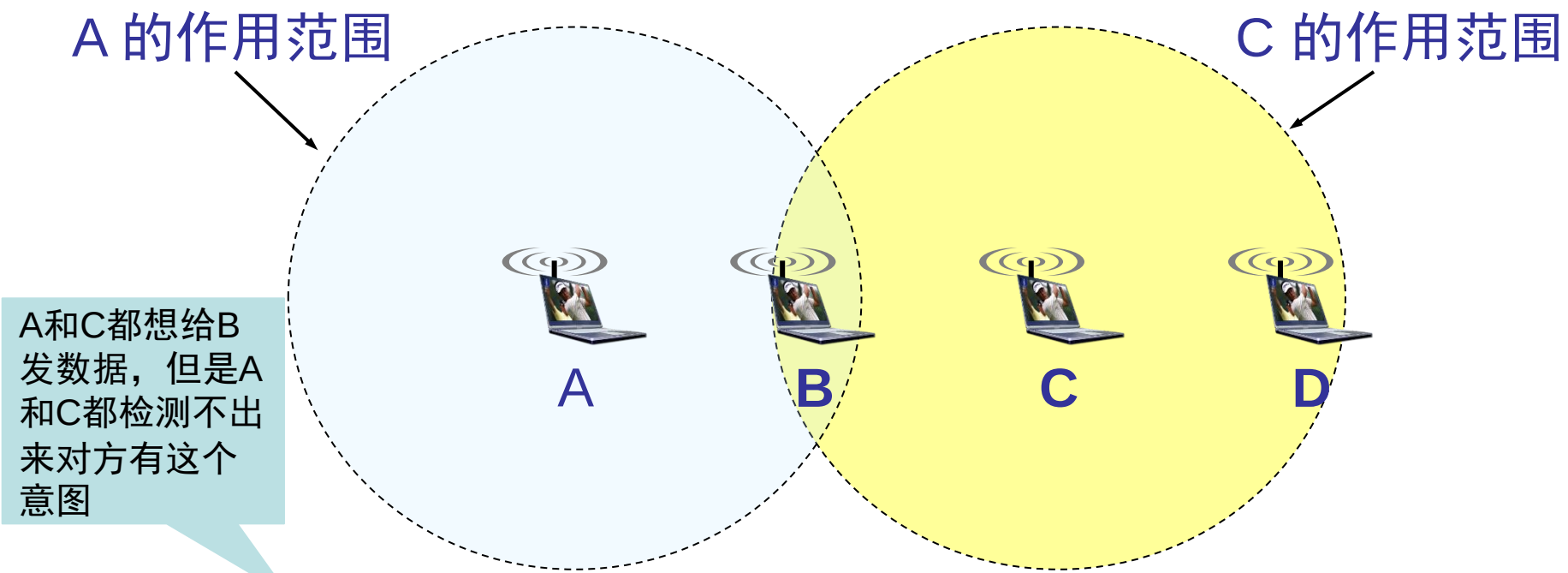
到DS	从DS	地址1	地址2	地址3	地址4
0	1	目的地址	AP地址	源地址	--
1	0	AP地址	源地址	目的地址	--

为何不采用CSMA/CD

- 要检测冲突，设备必须在发送数据时能够接受数据，对无线设备来说难以实现
- 碰撞检测方式不同
  - CSMA/CD：检测电压的变化
  - CSMA/CA：采用能量检测、载波检测和能量载波混合检测
- 即使能够实现碰撞检测的功能，并且当发送数据时检测到信道是空闲的，在接收端仍然有可能发生碰撞—隐蔽站问题

无线设备一般都采用半双工的方式

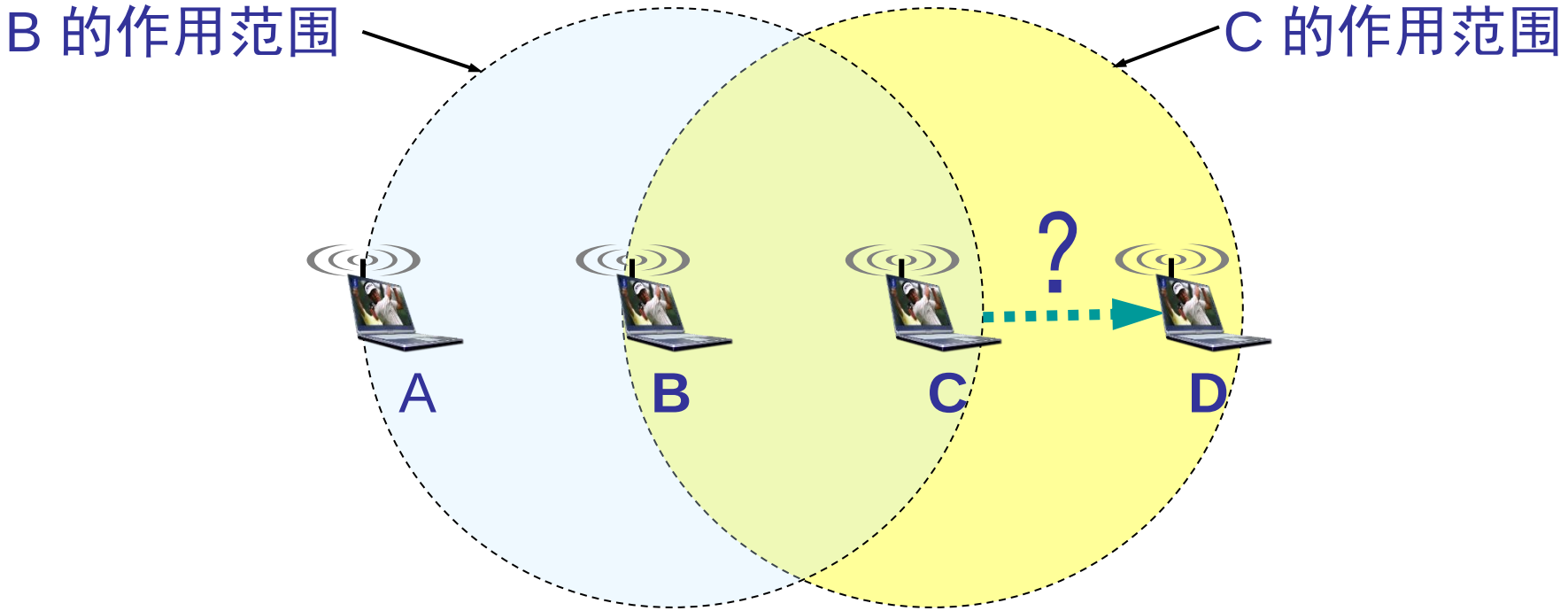
隐蔽站问题



当 A 和 C 检测不到无线信号时，都以为 B 是空闲的，因而都向 B 发送数据，结果发生碰撞。

这种未能检测出媒体上已存在的信号的问题叫做隐蔽站问题(hidden station problem)

暴露站问题



B 向 A 发送数据，而 C 又想和 D 通信。
C 检测到媒体上有信号，于是就不敢向 D 发送数据。

B 向 A 发送数据并不影响 C 向 D 发送数据
这就是暴露站问题(exposed station problem)
不影响协议的正常工作，但导致信道利用率下降

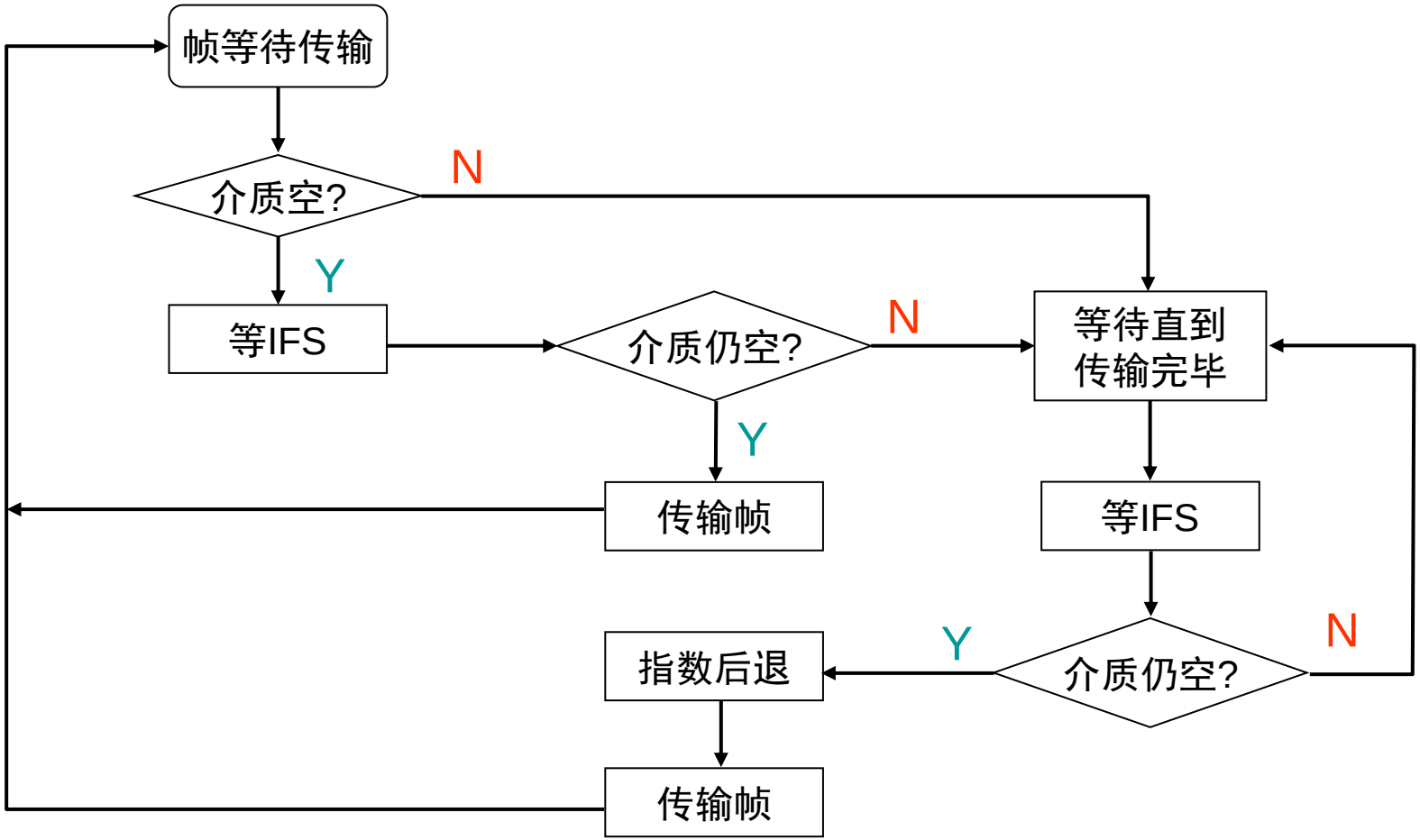
4.6.3 载波监听多址接入/碰撞避免(CSMA/CA)协议原理

- 核心
  - 物理侦听和虚拟侦听
- 物理侦听
  - 检查介质，在发送前侦听信道和检测到冲突后指数后退
- 虚拟侦听
  - 每个站可以保留一个信道何时要用的逻辑记录，通过跟踪网络分配向量（NAV）获得的

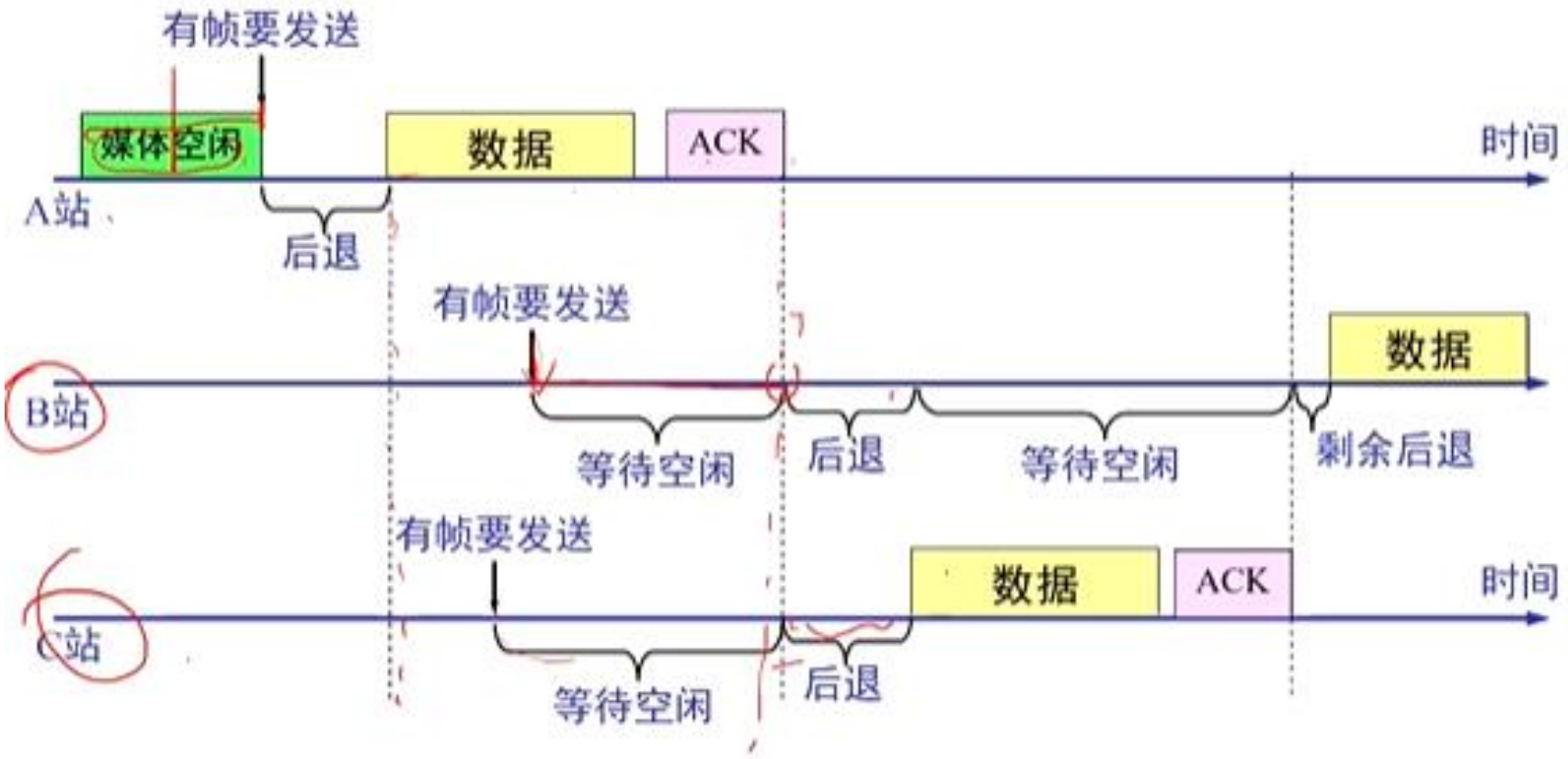
4.6.3 物理侦听

- CSMA/CA工作原理：如果某站点有数据要发送，它首先侦听信道，并根据下列不同的情形进行相应的处理：
  - 如果信道空闲，继续等待IFS(帧间隔)时间，然后再侦听信道；如果信道仍然空闲，立即发送数据
  - 如果信道忙，该站点继续侦听信道，直到当前传输完全结束
  - 一旦当前传输结束，站点继续等待IFS时间，然后再侦听信道，如果信道仍然保持空闲，站点按指数后退一个随机长的时间后，发送数据。

在CSMA/CA工作流程



CSMA/CA机制下发送帧



二进制指数退避算法

- 第 *i* 次退避就在 2^{2+i} 个时隙中随机地选择一个，即：

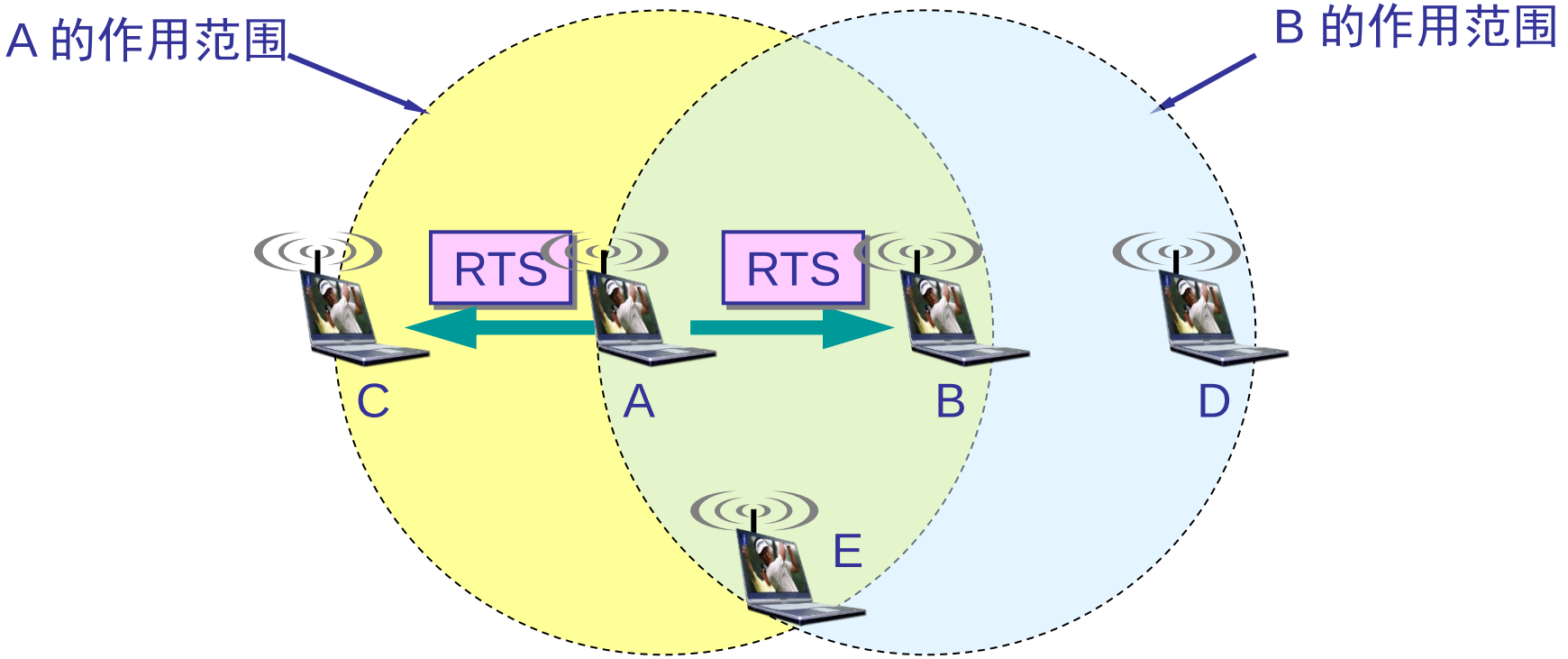
  第 *i* 次退避是在时隙 $\{0, 1, \ldots, 2^{2+i} - 1\}$ 中随机地选择一个。

- 第 1 次退避是在 8 个时隙（而不是 2 个）中随机选择一个。
- 第 2 次退避是在 16 个时隙（而不是 4 个）中随机选择一个。
- 当时隙编号达到255时（6次避退）不再增加

冲突避免技术

- CSMA/CA的关键在于冲突避免。采用了3种机制来实现：
  - 预约信道： 使用RTS/CTS机制(请求发送/允许发送)，向其它站发出通告， 本站将占用多长时间，其它站不要发送数据并调整其网络分配向量NAV，以免冲突。
    - NAV：信道处于忙状态的时间
    - 可以解决隐藏站问题
  - 正向确认：接收站正确收到数据要发确认帧， 否则源站点重复发送数据帧。

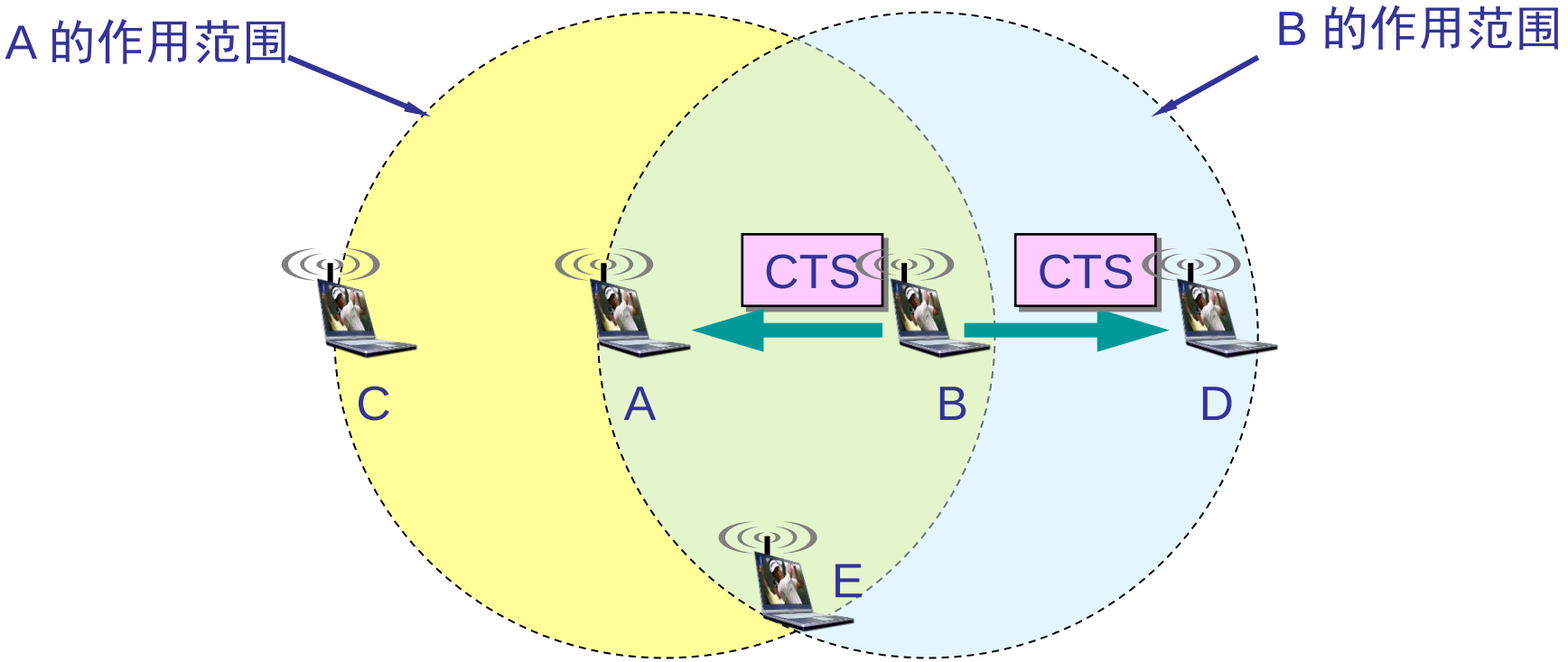
对信道进行预约(RTS)



源站 A 在发送数据帧之前先发送一个短的控制帧，叫做请求发送 RTS，它包括源地址、目的地址和这次通信（包括相应的确认帧）所需的持续时间。

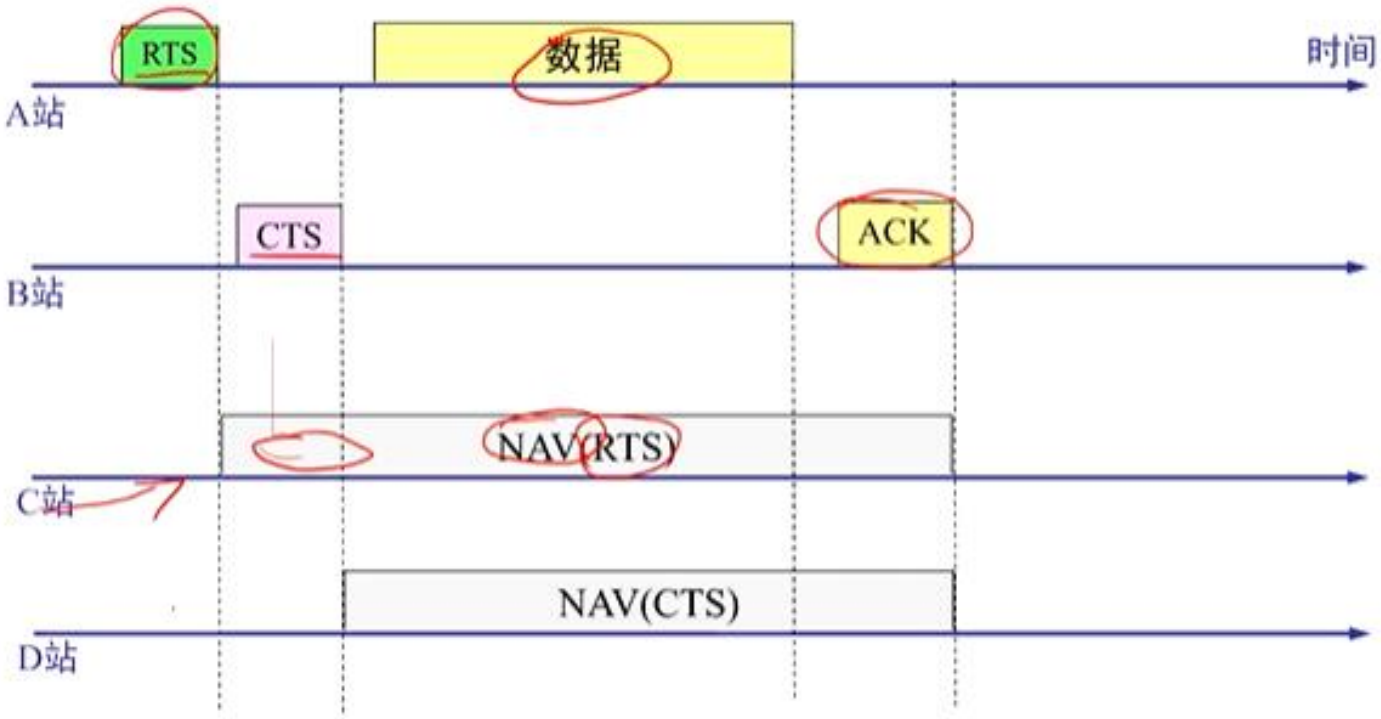
对信道进行预约(CTS)

A 收到 CTS 帧后就可发送其数据帧。



若媒体空闲，则目的站 B 就发送一个响应控制帧，叫做允许发送 CTS，它包括这次通信所需的持续时间（从 RTS 帧中将此持续时间复制到 CTS 帧中）。

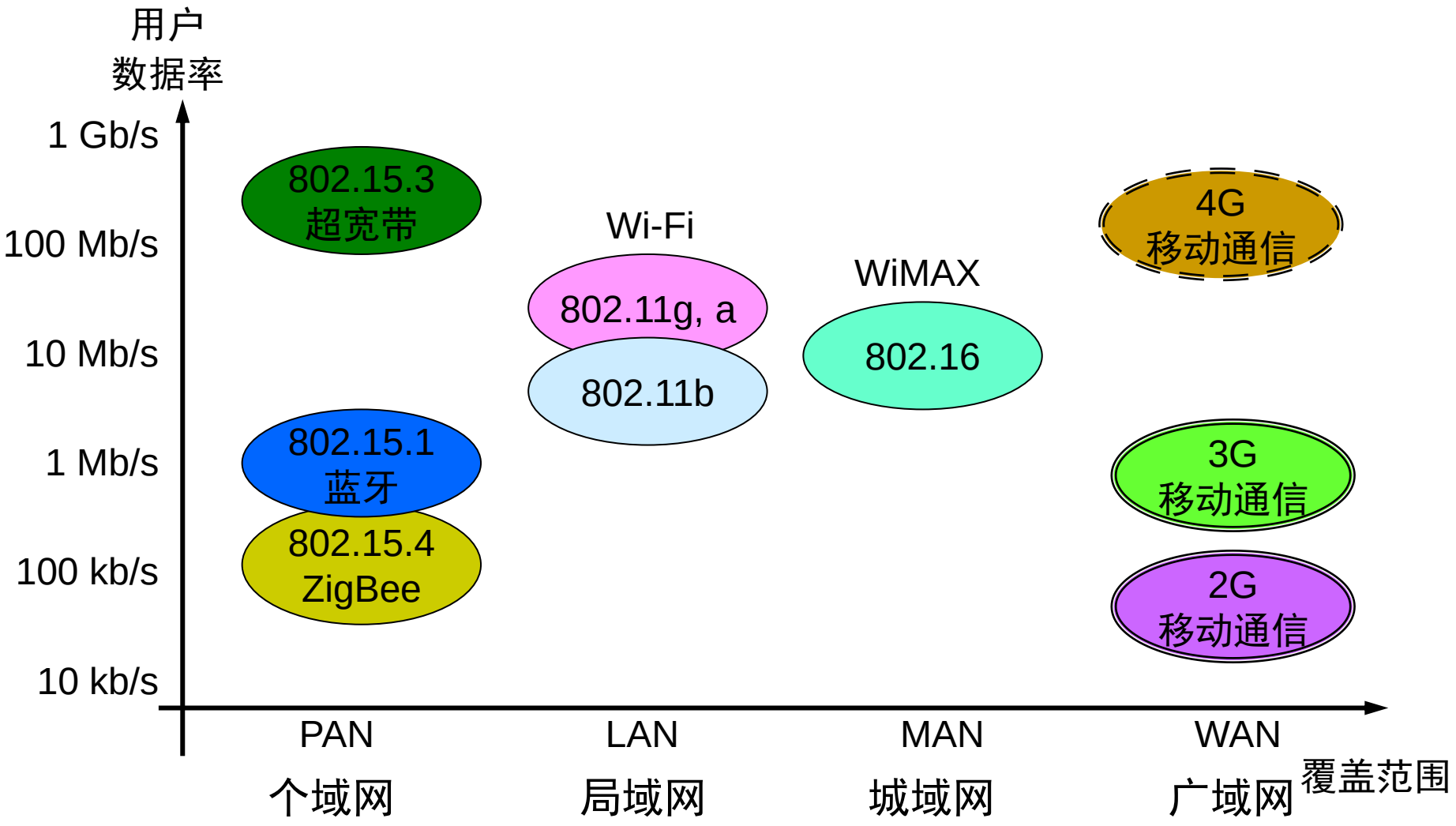
虚拟信道侦听



安全

- 802.11b标准中定义了有线等效保密协议WEP (Wired Equivalent Privacy)
- WEP提供认证和加密服务，但安全性不高
- WPA (Wi-Fi Protected Access)克服了WEP的缺点，提供了认证、加密以及消息完整性验证等服务
- 2003年5月，中国提出了无线局域网国家标准GB15629.11，引入了无线局域网认证与保密基础结构WAPI(WLAN Authentication and Privacy Infrastructure)

几种无线网络的比较


用户
数据率
1 Gb/s
100 Mb/s
10 Mb/s
1 Mb/s
100 kb/s
10 kb/s
802.15.3
超宽带
802.15.1
蓝牙
802.15.4
ZigBee
Wi-Fi
802.11g, a
802.11b
WiMAX
802.16
4G
移动通信
3G
移动通信
2G
移动通信
PAN
个域网
LAN
局域网
MAN
城域网
WAN
广域网
覆盖范围

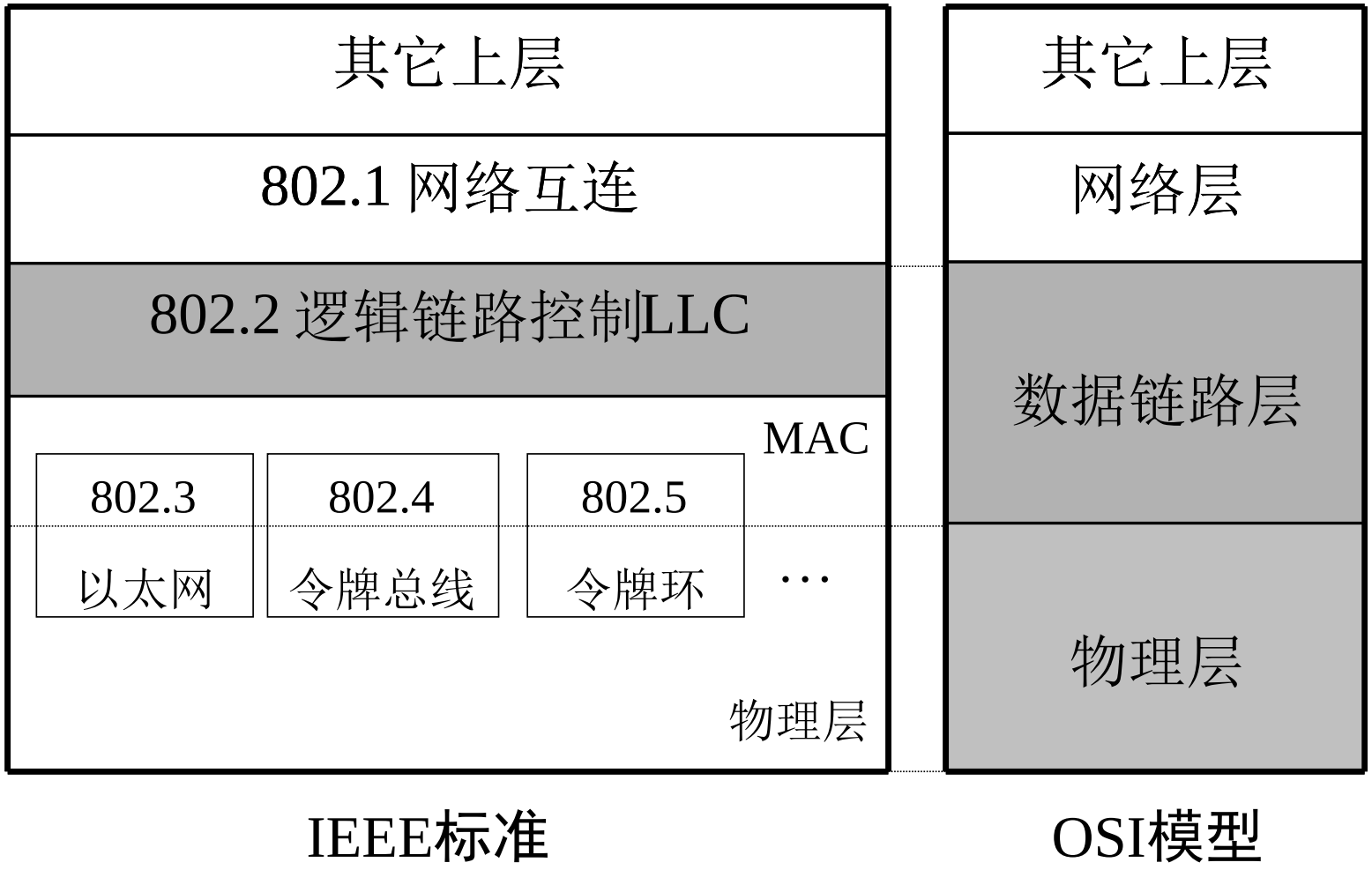
4.7 数据链路层网络互连

- 网桥是数据链路层上的互连设备。
- 从互连网络的结构上看，网桥属于DCE级的端到端的连接；从协议的层次上看，网桥同时作用在OSI的物理层和数据链路层。

4.7.1 网桥

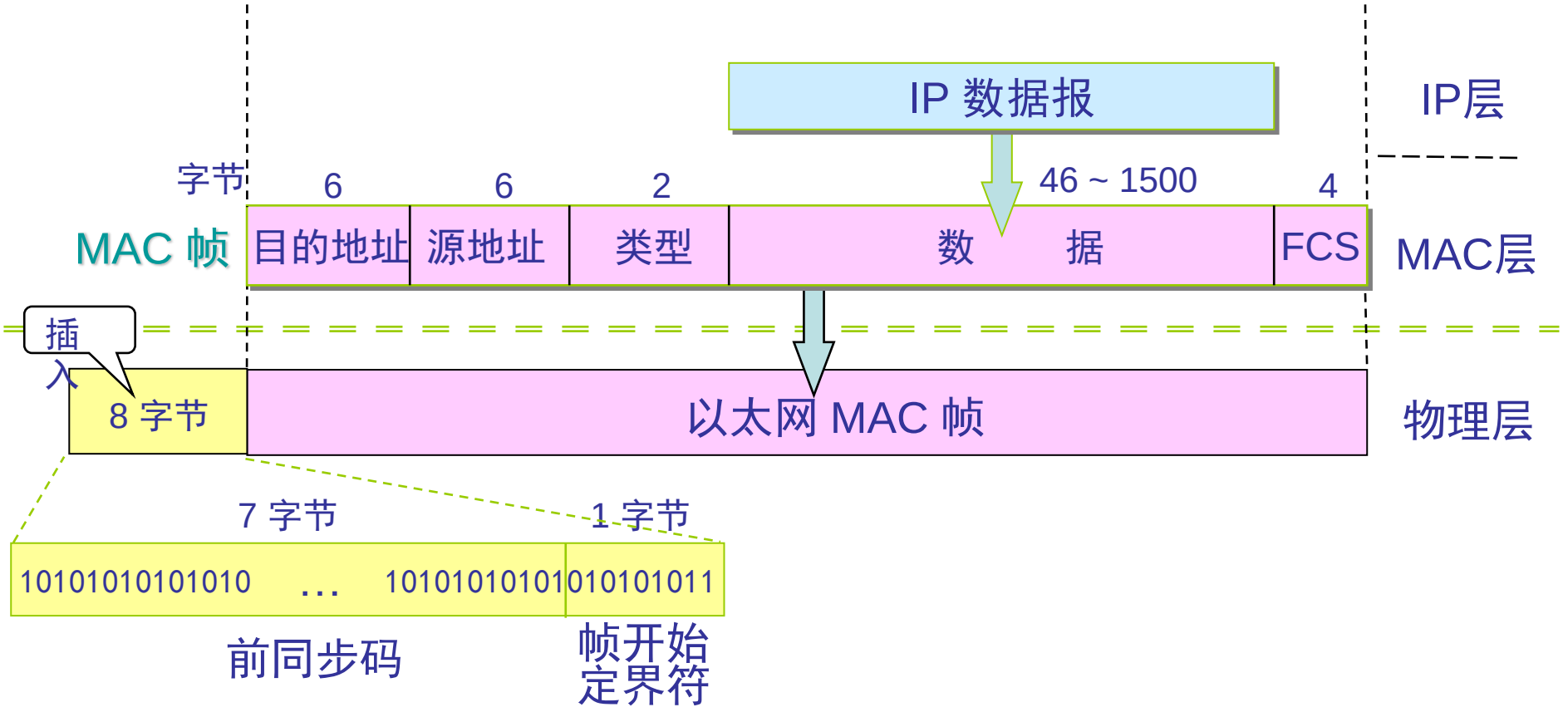
- 网桥在数据链路层上进行数据帧的存贮和转发
- 网桥常用于局域网的互连
- 局域网常用的链路层协议：
  - 802.1：LAN中的网络互连标准
  - 802.2：LLC逻辑链路控制协议标准
  - 802.3：CSMA/CD媒体访问方法
  - 802.4：令牌总线访问方法
  - 802.5：令牌环访问方法
  - 802.11：无线局域网协议

802与OSI的层次对应关系

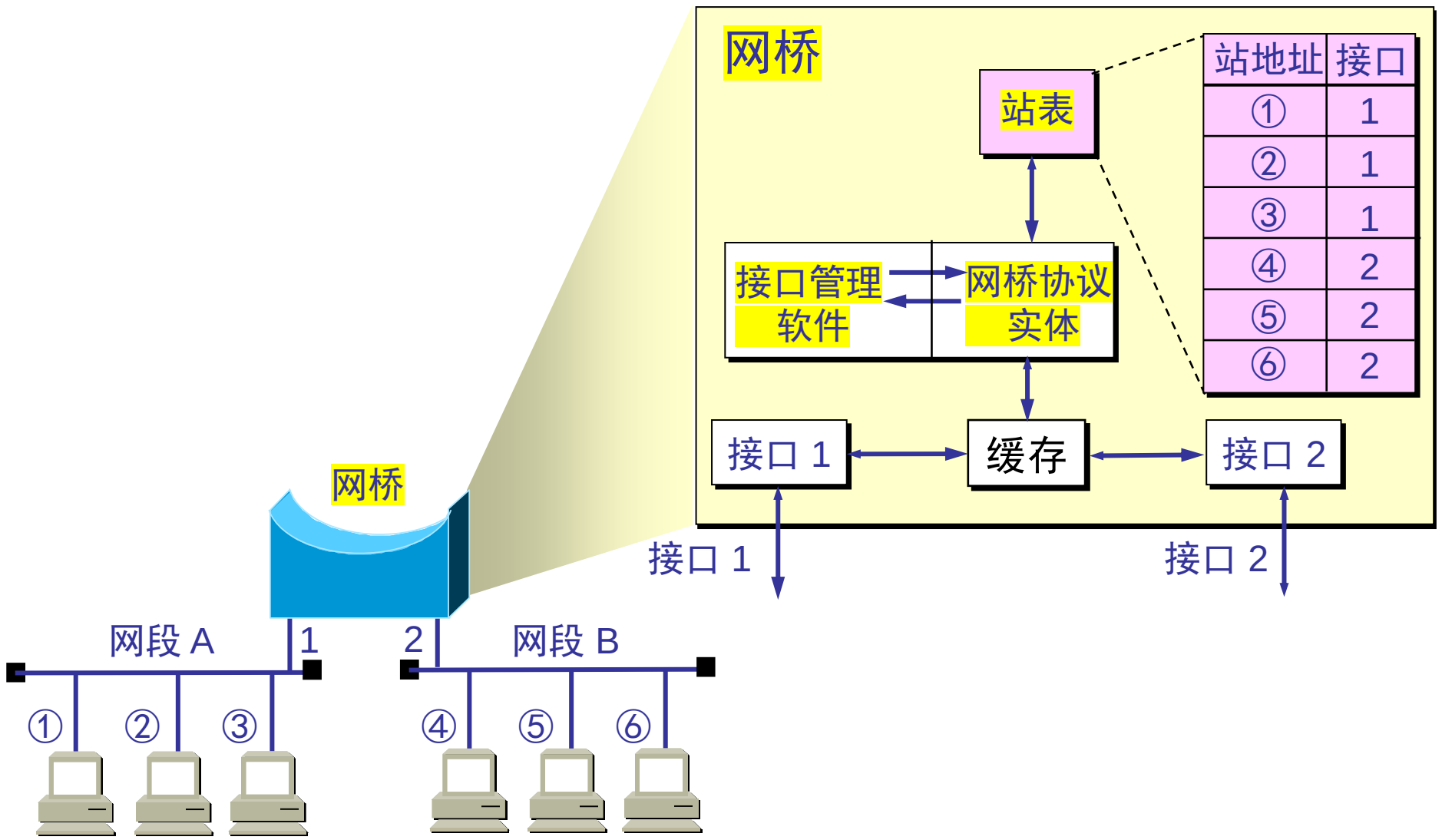


以太网MAC帧

- 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



网桥的内部结构

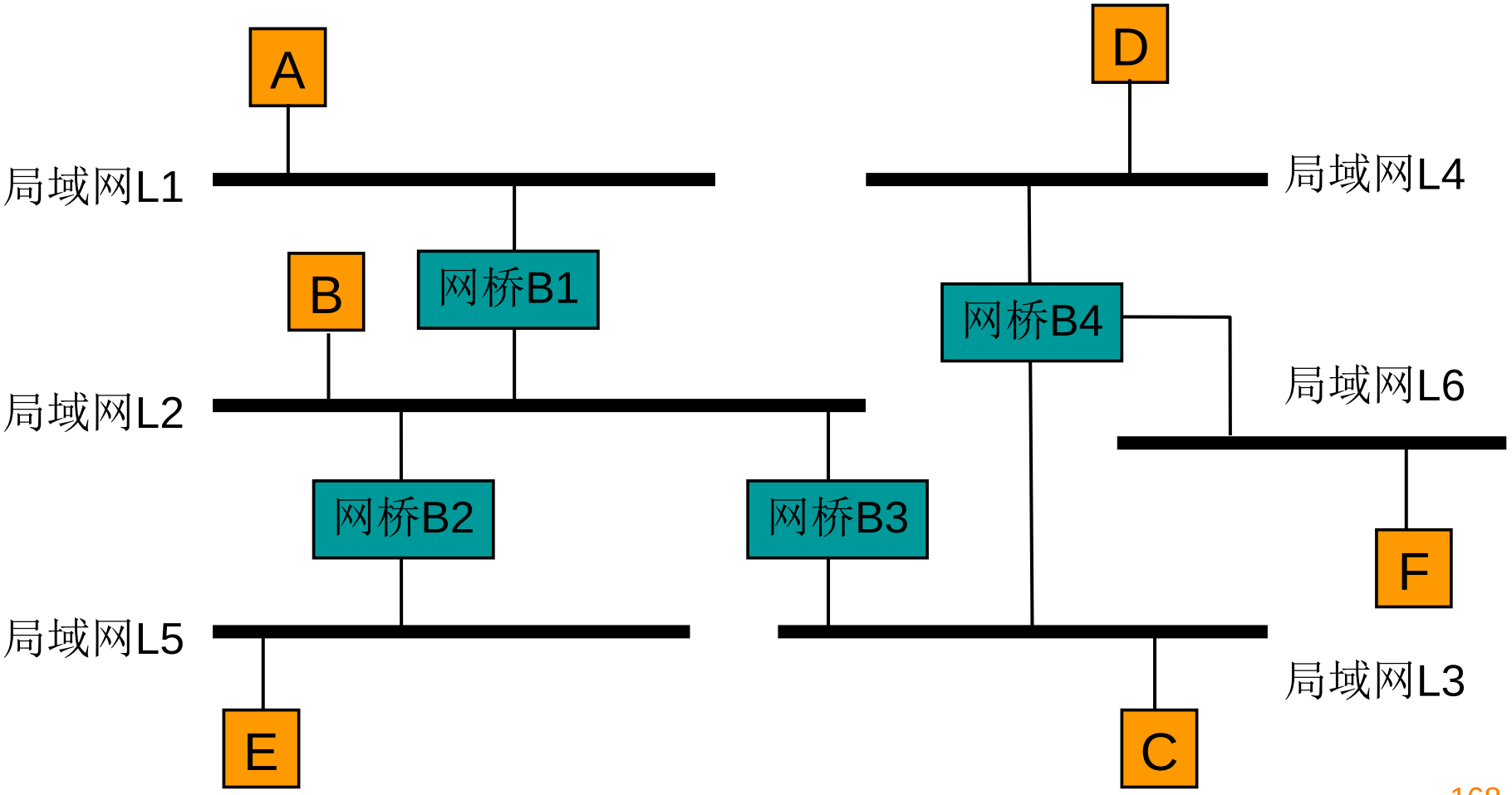

网桥
站表
站地址
接口
①
1
②
1
③
1
④
2
⑤
2
⑥
2
接口管理
软件
网桥协议
实体
接口 1
缓存
接口 2
接口 1
接口 2
网桥
网段 A
1
2
网段 B
①
②
③
④
⑤
⑥

使用网桥带来的好处

- 过滤通信量。
- 扩大了物理范围。
- 可互连不同物理层、不同 MAC 子层和不同速率（如10 Mb/s 和 100 Mb/s 以太网）的局域网。

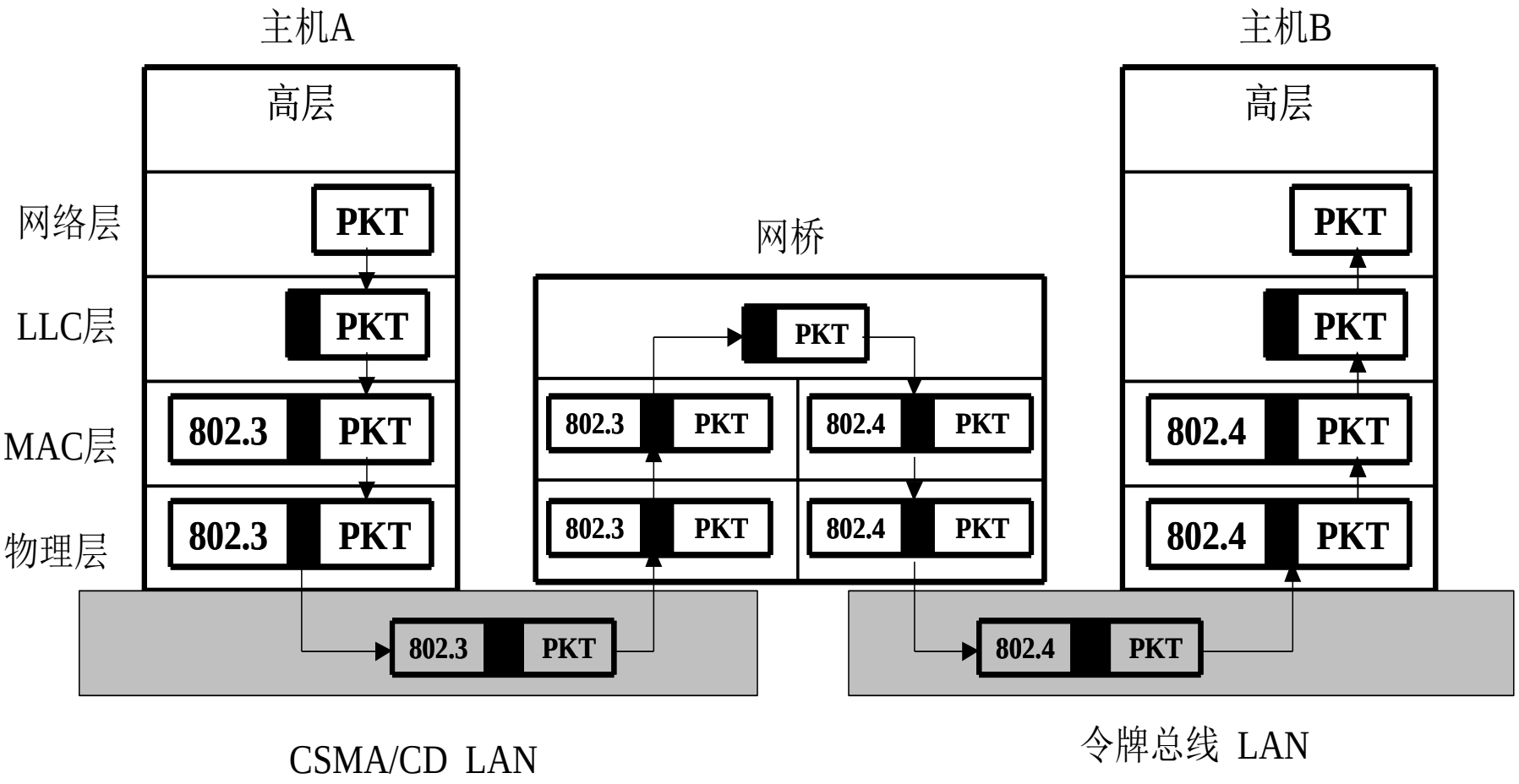
网桥的路由功能

- 网桥具有根据帧的目的地址决定是否接受该帧的功能，也就是具有路由的功能。



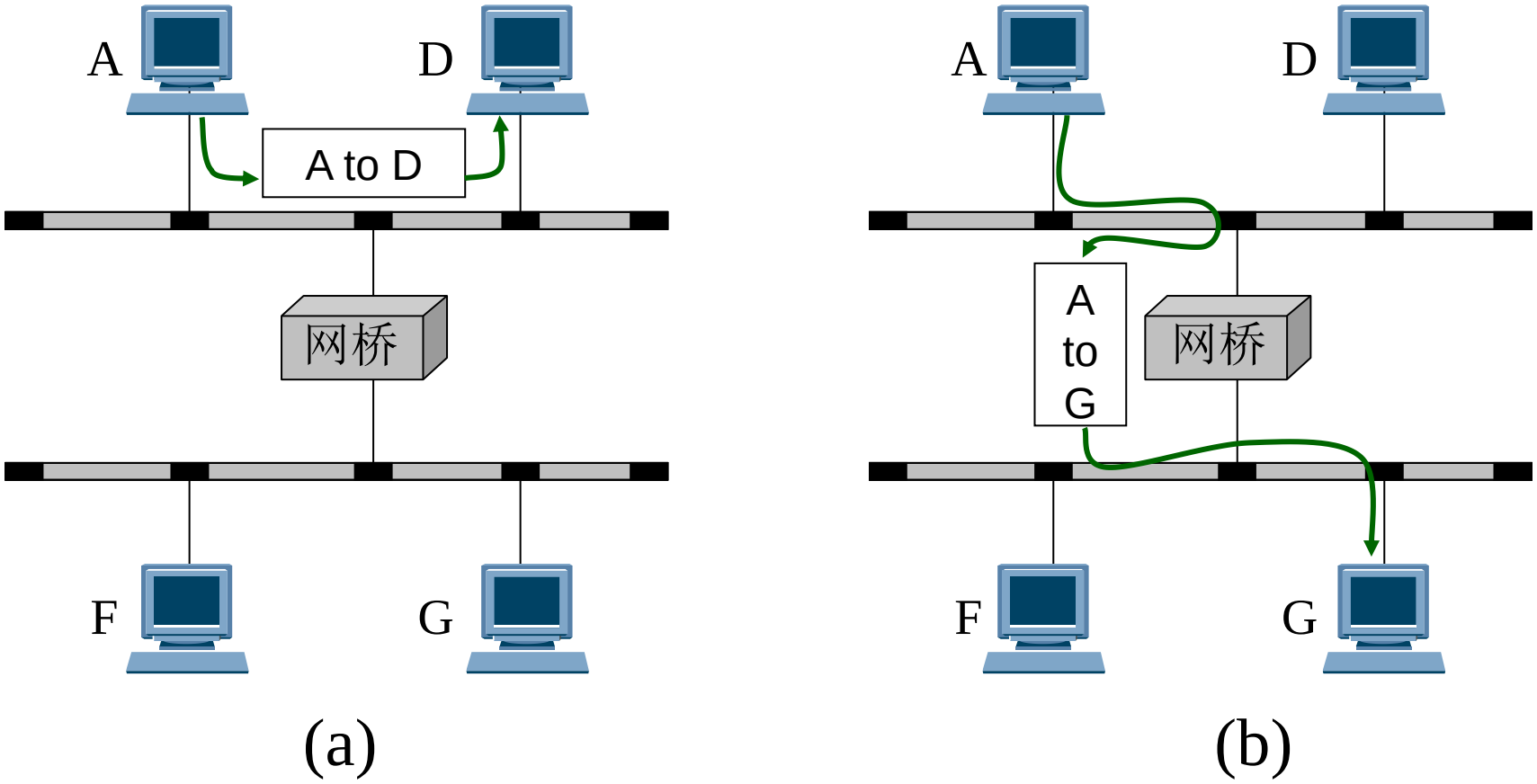
网桥的错误检测和帧格式转换功能

- 由于网桥工作在数据链路层，因此可以将数据链路层不一致的帧转换。

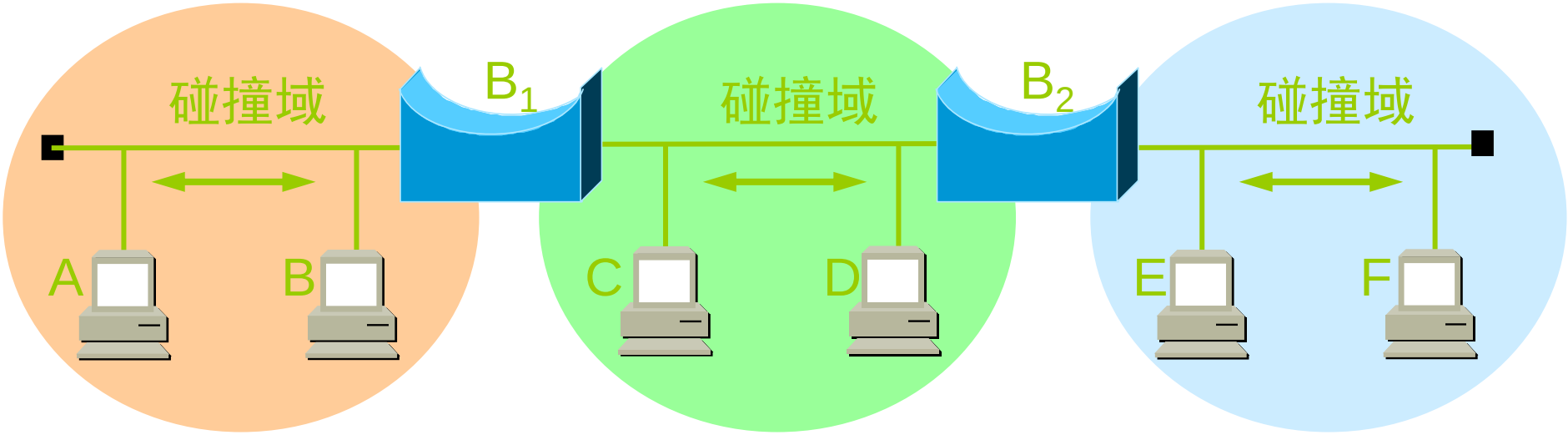


隔离通信功能

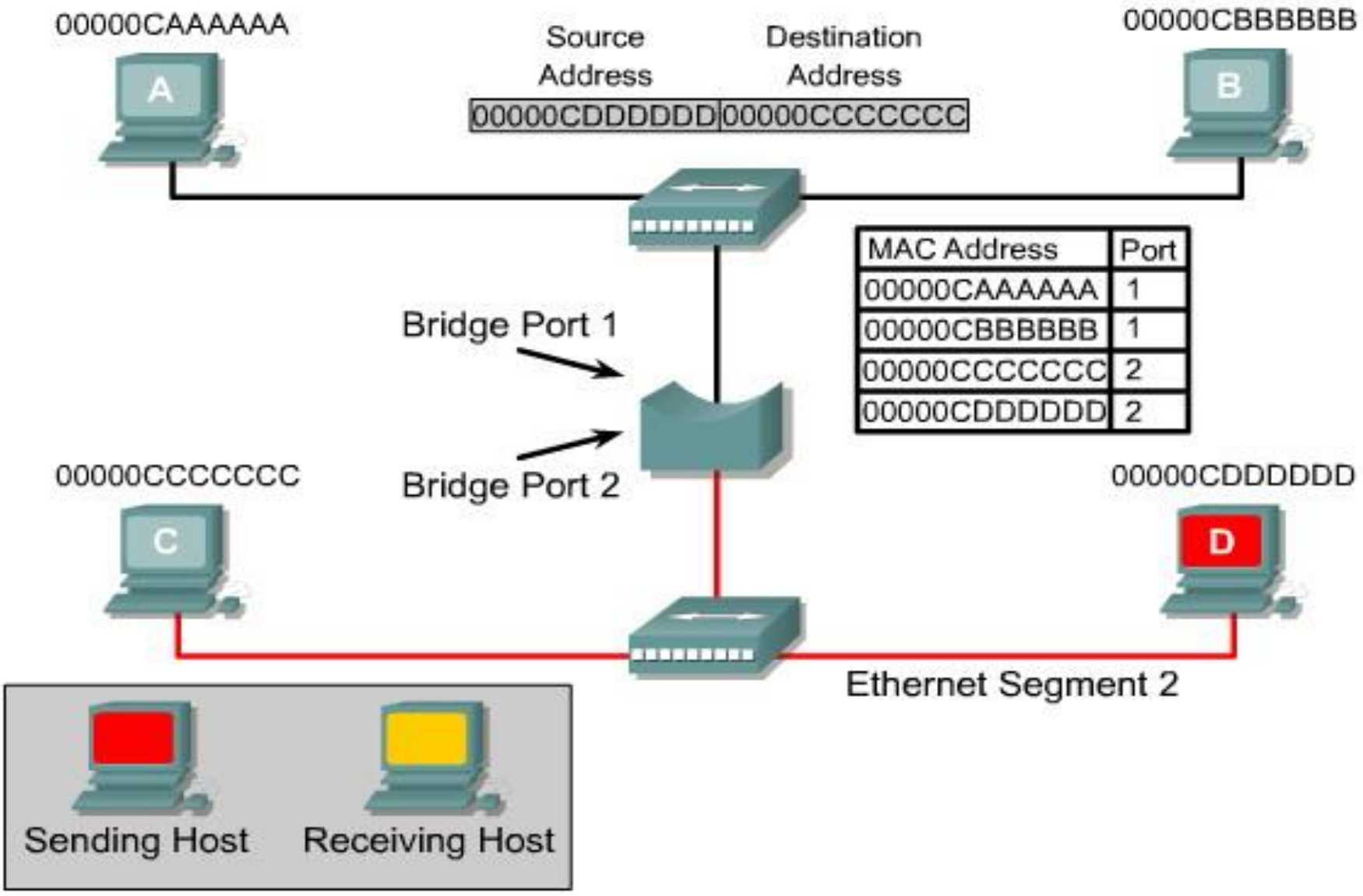
- 网桥和中继器的不同之处：网桥具有隔离通信的功能。



网桥使各网段成为隔离开的碰撞域



示例

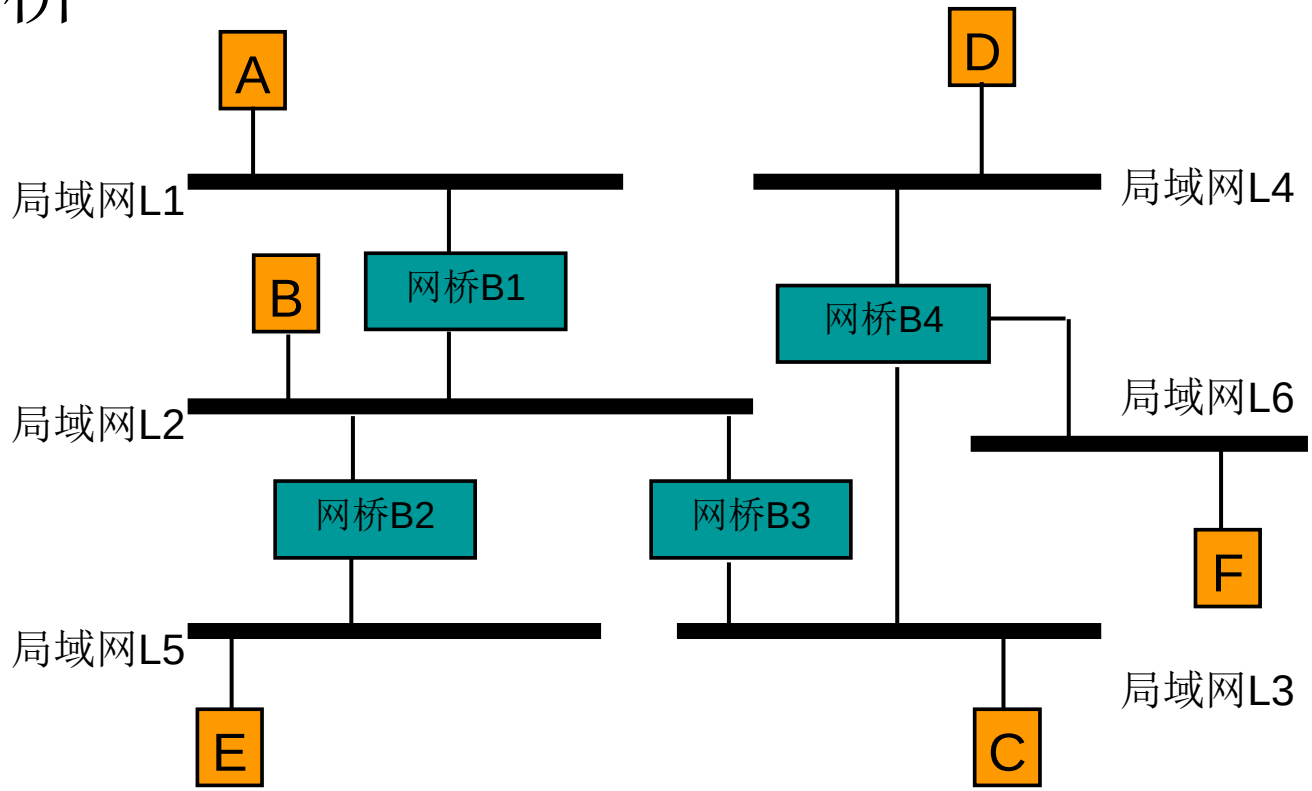

00000CAAAAAA
A
Source Address
Destination Address
00000CDDDDDD
00000CCCCCCC
00000CBBBBBB
B
MAC Address
Port
00000CAAAAAA
1
00000CBBBBBB
1
00000CCCCCCC
2
00000CDDDDDD
2
Bridge Port 1
Bridge Port 2
00000CCCCCCC
C
00000CDDDDDD
D
Ethernet Segment 2
Sending Host
Receiving Host

使用网桥带来的缺点

- 存储转发增加了时延。
- 在MAC 子层并没有流量控制功能。
- 具有不同 MAC 子层的网段桥接在一起时时延更大。
- 网桥只适合于用户数不太多(不超过几百个)和通信量不太大的局域网。

4.7.2 网桥路由算法

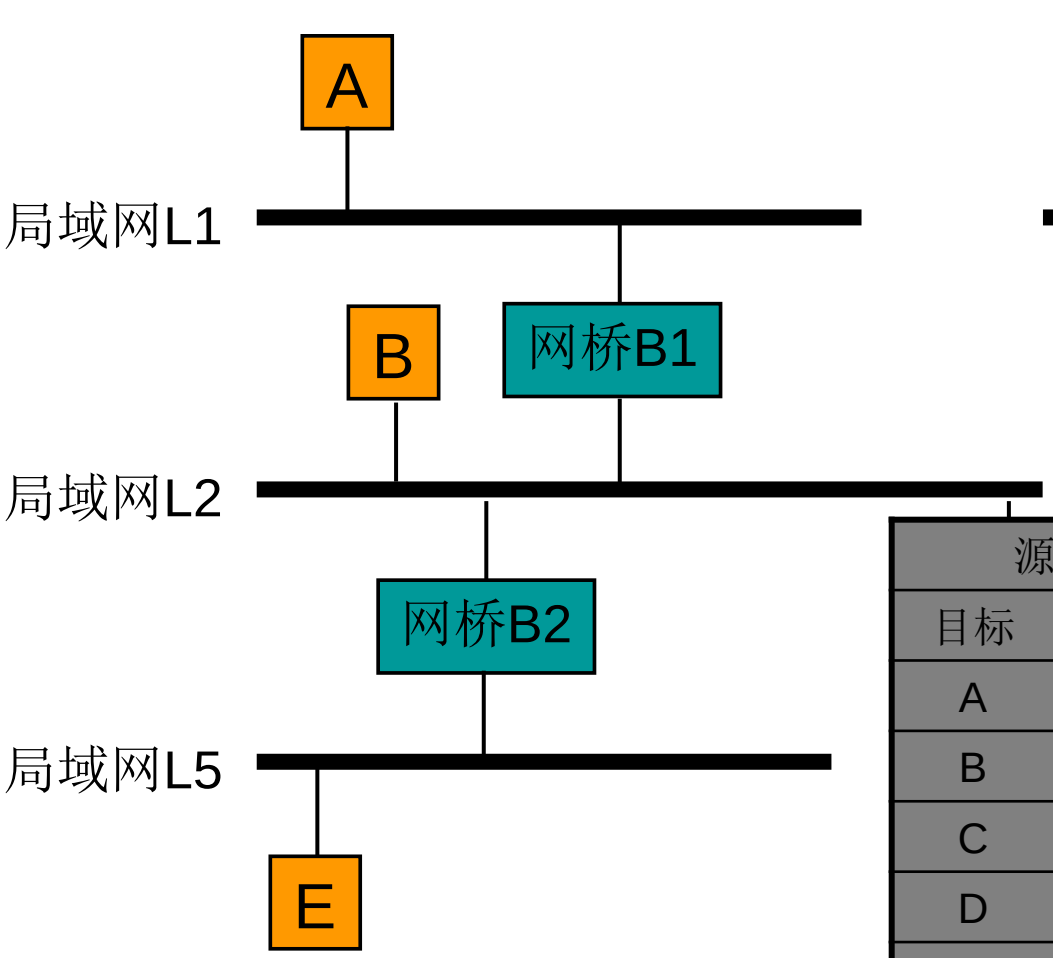
- 固定路由网桥
- 透明网桥
- 源路由网桥

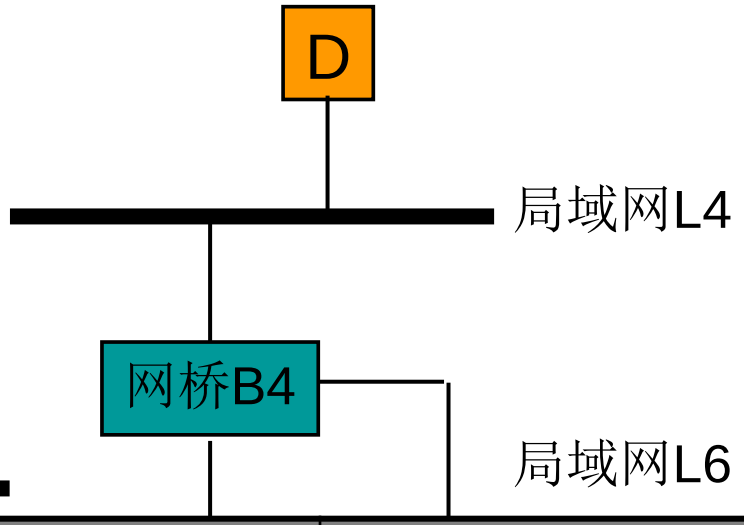


固定路由网桥

- 每个网桥中都有一张表，这张表中记录了到某个特定站点的帧应该转发到那个哪个局域网中去的信息，这个表称为路由表。
- 路由表的生成是由手工配置的，一旦配置完成，路由表不会变动。

固定路由网桥示例





源局域网L1		源局域网L2	
目标	下一局域网	目标	下一局域网
A	--	A	L1
B	L2	B	--
C	L2	C	--
D	L2	D	--
E	L2	E	--
F	L2	F	--

各网桥路由表

源局域网L1		源局域网L2	
目标	下一局域网	目标	下一局域网
A	--	A	L1
B	L2	B	--
C	L2	C	--
D	L2	D	--
E	L2	E	--
F	L2	F	--

网桥B1

源局域网L2		源局域网L5	
目标	下一局域网	目标	下一局域网
A	--	A	L2
B	--	B	L2
C	--	C	L2
D	--	D	L2
E	L5	E	--
F	--	F	L2

网桥B2

源局域网L2		源局域网L3	
目标	下一局域网	目标	下一局域网
A	--	A	L2
B	--	B	L2
C	L3	C	--
D	L3	D	--
E	--	E	L2
F	L3	F	--

网桥B3

源局域网L3		源局域网L4		源局域网L6	
目标	下一局域网	目标	下一局域网	目标	下一局域网
A	--	A	L3	A	L3
B	--	B	L3	B	L3
C	--	C	L3	C	L3
D	L4	D	--	D	L4
E	--	E	L3	E	L3
F	L6	F	L6	F	--

网桥B4

透明网桥(Transparent Bridge)

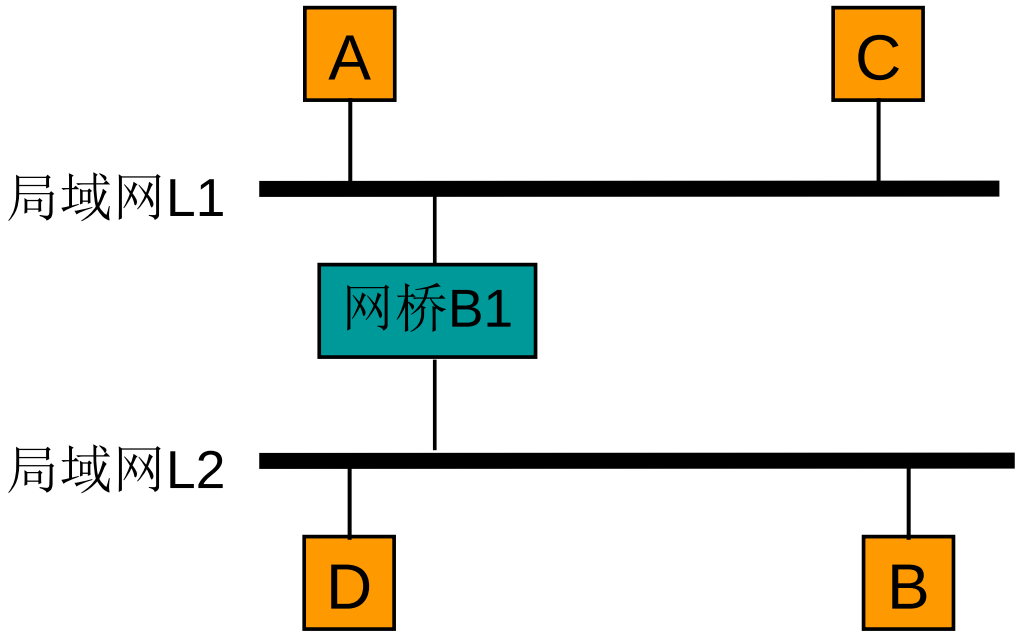
- “透明”是指局域网上的站点并不知道所发送的帧将经过哪几个网桥，因为网桥对各站来说是看不见的。
- 透明网桥是一种即插即用设备，其标准是IEEE 802.1D。

透明网桥

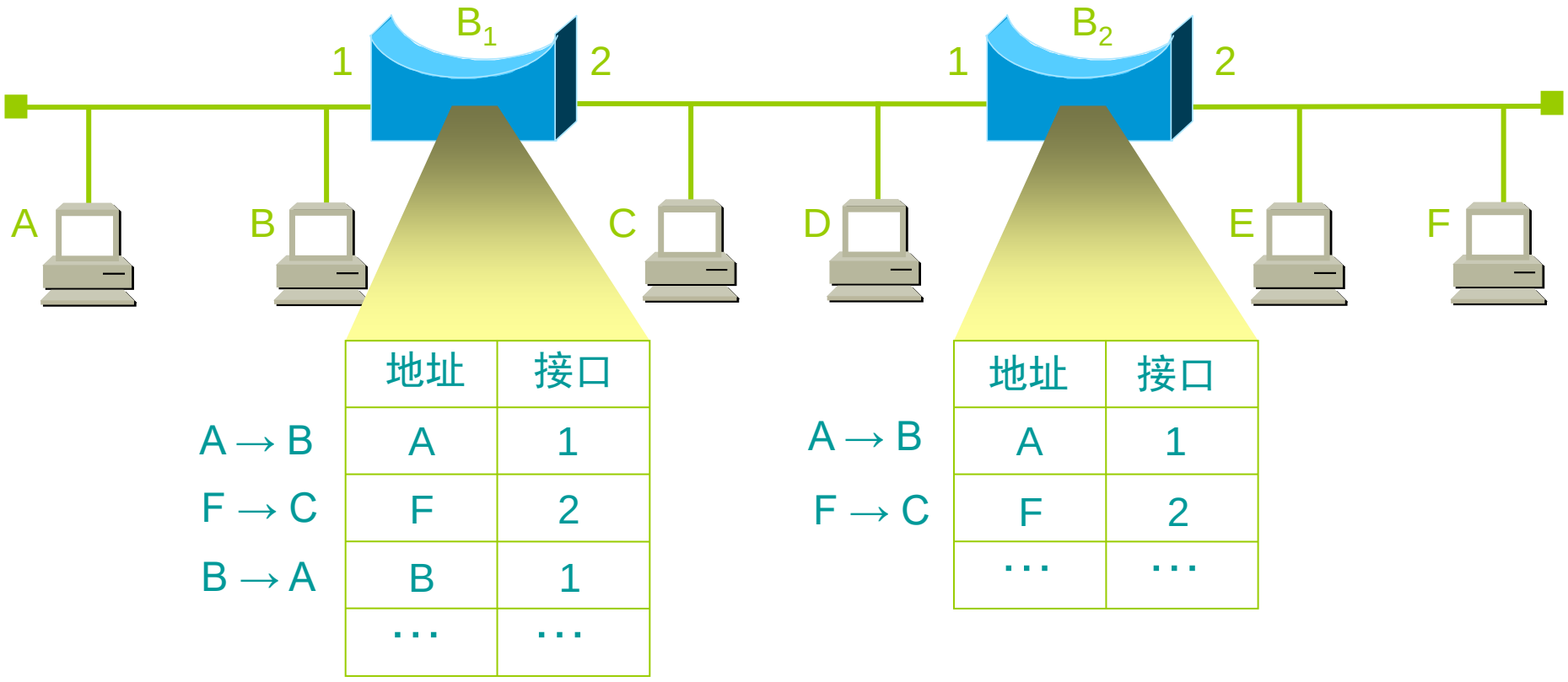
- 采用逆向学习算法能够根据网络信息自动生成和修改自己的路由表
- 自动修改和生成路由表的能力称为路由学习(Route Learning)或地址学习(Address Learning)。
- 要解决以下问题：
  - 路由表的初始化
  - 路由表的自动修改
  - 帧循环--生成树算法

路由表的初始化

- 当网桥收到一个发往某站点的帧，而在路由表中没有该站点的路由信息时，网桥使用一个扩散算法向它所连接的所有局域网发送这个帧。



转发表的建立过程举例



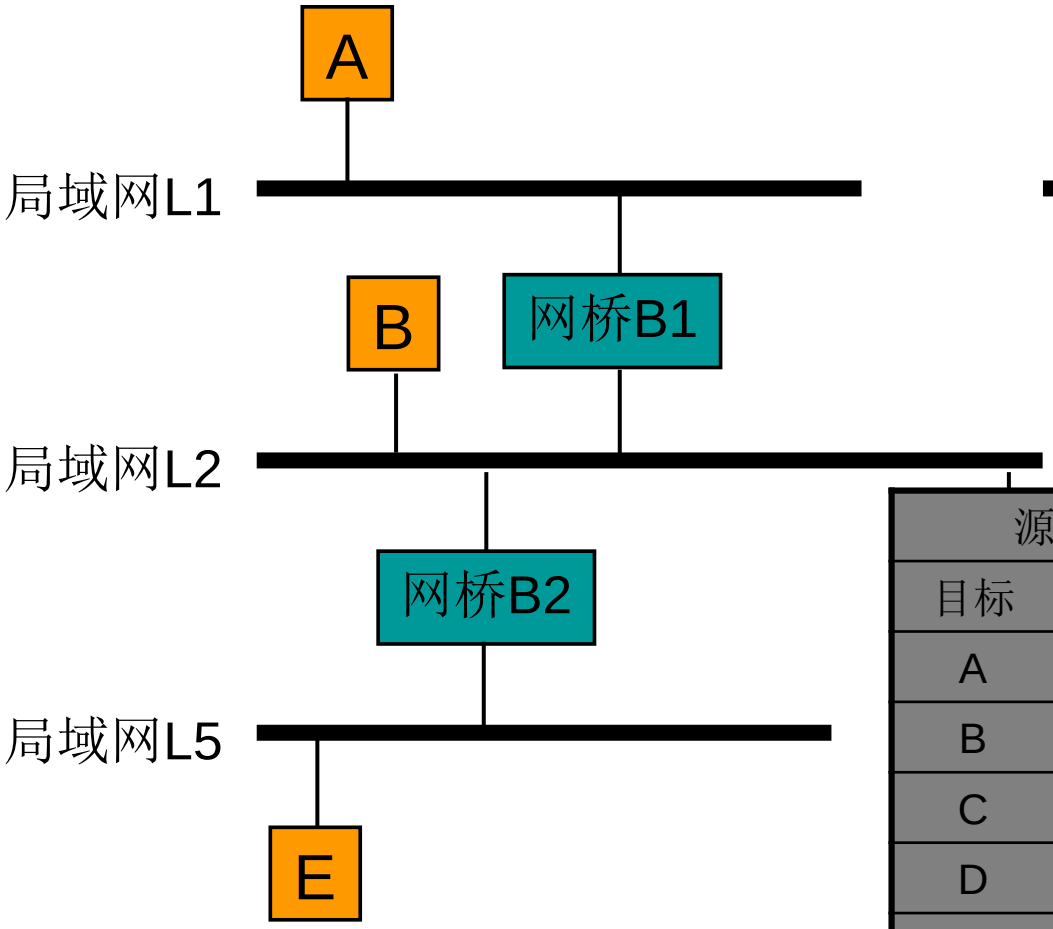
	地址	接口
A → B	A	1
F → C	F	2
B → A	B	1
	…	…

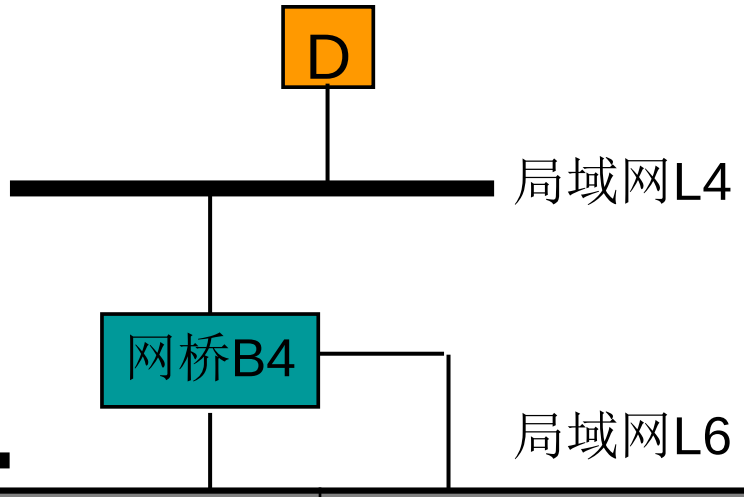
	地址	接口
A → B	A	1
F → C	F	2
	…	…

路由表的自动修改

- 任何时候当它接收到一个帧时，它检查帧的源地址。然后就知道发送这个帧的站点可以通过这个帧刚到达的局域网来访问。
- 网桥的MAC地址表空间有限，当局域网中站点数目大于MAC地址表表项是，要采取某种算法替换一些表项。
- 定时器：为了处理动态拓扑问题，每当增加散列表项时，均在该项中注明帧的到达时间。

路由表的自动修改示例

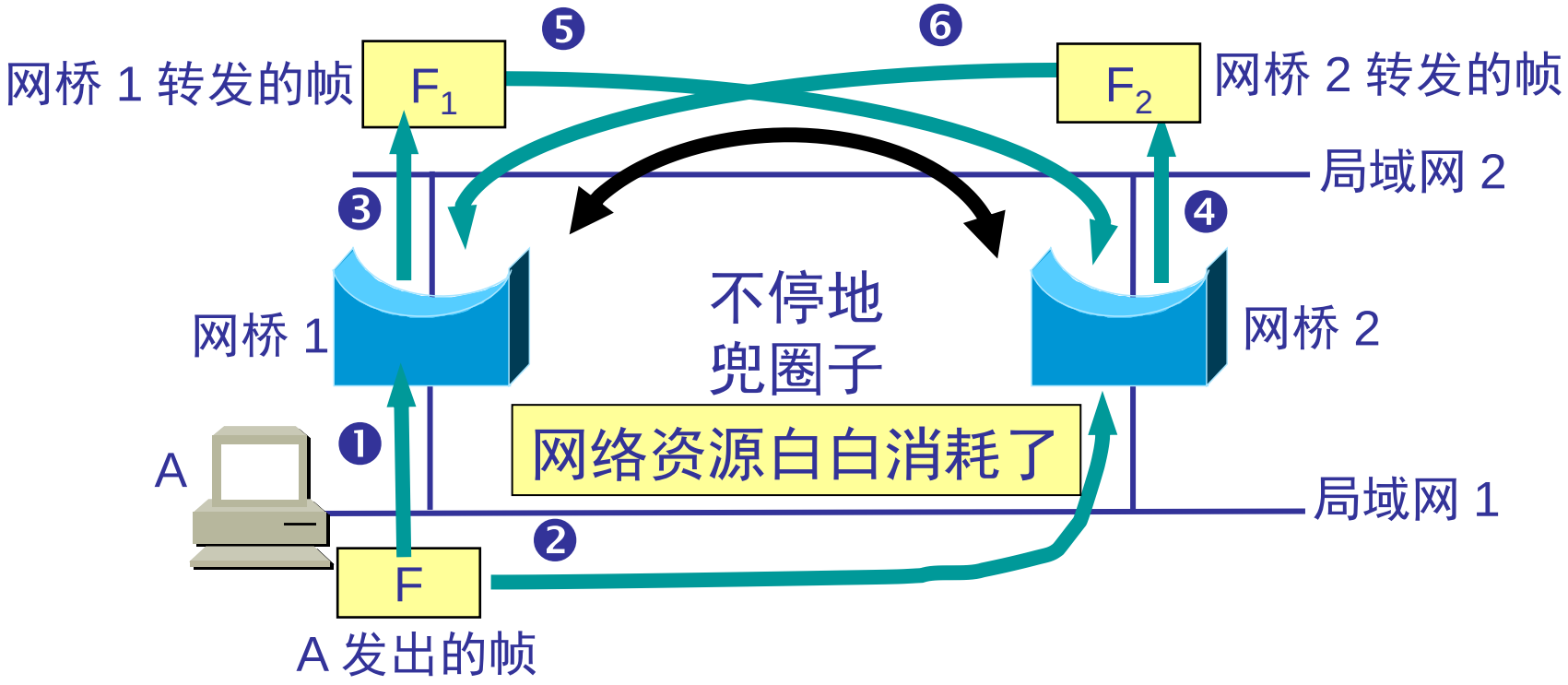




源局域网L1		源局域网L2	
目标	下一局域网	目标	下一局域网
A	--	A	L1
B	L2	B	--
C	L2	C	--
D	--	D	L1
E	L2	E	--
F	L2	F	--

帧循环问题

- 当一个互连局域网有回路时，就可能会产生帧的循环传递问题。这种过程继续下去，将导致帧的爆炸，最终会阻塞整个系统，使通信停止。



生成树算法(Spanning Tree)

- 对于带有回路的互连局域网，必须停用某些网桥来消除循环。即不允许某些网桥转发帧，把它们当作别的网桥失效时的备份。
- 网桥执行一种称为生成树的算法来完成这项工作。算法规则：
  - 先确定根网桥
  - 查找网桥到根的最短费用，选举根端口
  - 确定每个网络连接的网桥，选举指定端口
- 为了能够反映网络拓扑发生变化时的生成树，在生成树上的根网桥每隔一段时间还要对生成树的拓扑进行更新。

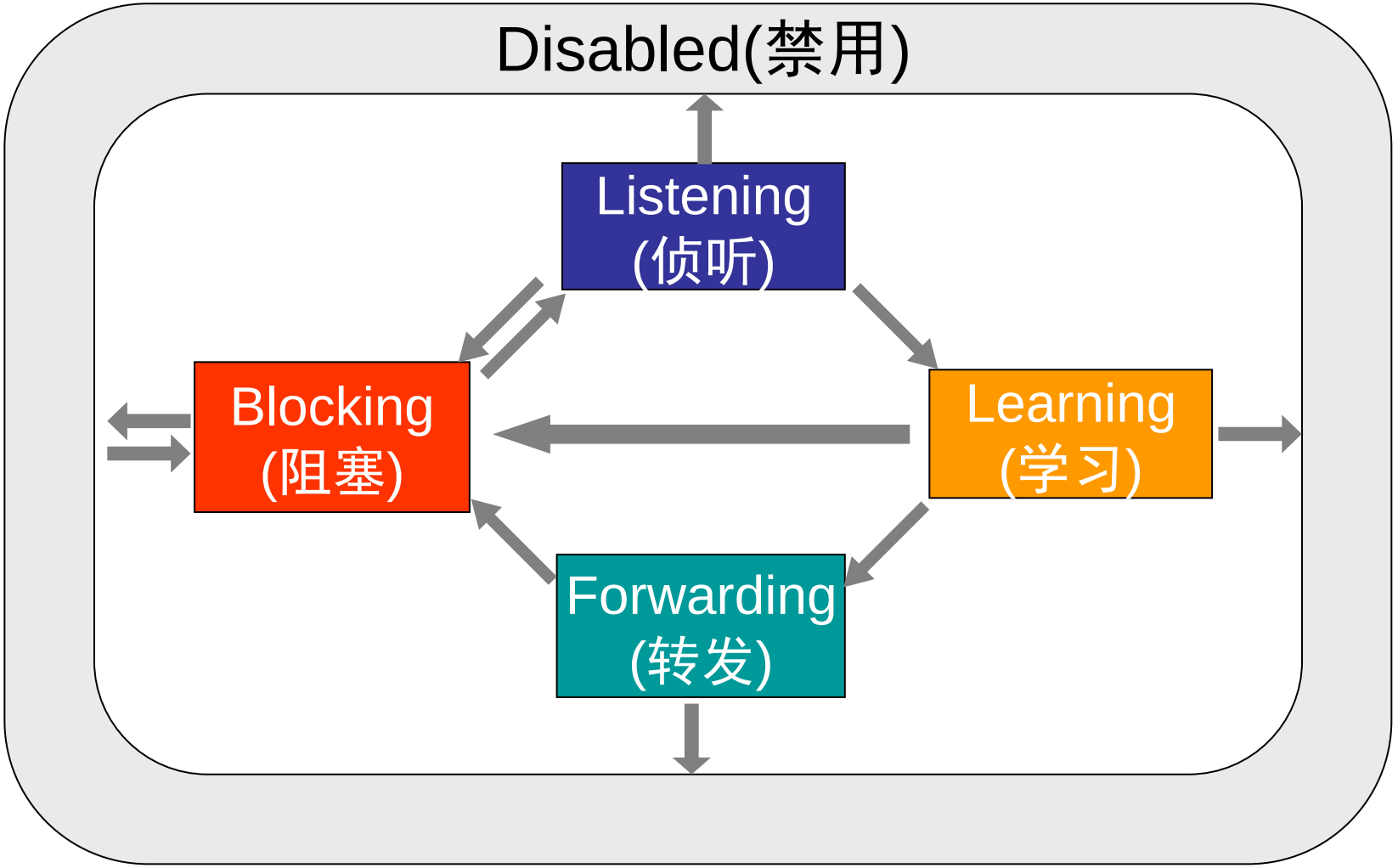
相关名词

- BPDU：STP的数据单元，在网桥局域网内传递信息。
- 根网桥：具有最小网桥ID的网桥被选作根网桥，网桥ID应为唯一的。
- 根端口：在指定网桥上面，到根网桥路径花费最小的端口为根端口，如果指定网桥上面有几个端口，到根网桥路径花费一样小，那么选择端口id 最小的端口为根端口。
- 指定网桥：到根网桥路径花费最少的那个网桥为指定网桥，如果，有几个网桥到到根网桥路径花费一样，那么比较网桥id，id最小的被选作为指定网桥。
- 指定端口：指定网桥上面和局域网相连的端口叫做指定端口，如果指定网桥上面有几个端口同时和局域网相连，那么选择端口id 最小的端口为所在局域网的指定端口。

STP中端口状态

- 禁用(Disabled)状态
- 阻塞(Blocking)状态—不转发数据帧，接收BPDU
- 侦听(Listening)状态—不转发数据帧，侦听BPDU，并进入生成树构造过程
- 学习(Learning)状态—不转发数据帧，学习地址
- 转发(Forwarding)状态—转发数据帧，学习地址

状态转换


Disabled(禁用)
Listening
(侦听)
Blocking
(阻塞)
Learning
(学习)
Forwarding
(转发)

生成树算法

- 首先选择一个网桥作为根网桥。根网桥是具有最小ID的那个网桥，根网桥是生成树的根节点。
  - 根网桥的选择是通过发送网桥协议数据单元BPDU—这样的特殊帧来完成的。每个网桥协议数据单元包含一个网桥的ID，帧首次被发送时的端口ID和接收该帧的端口的累计费用。
  - 网桥开始时假定它自己是根网桥，向它的所有端口发送配置信息，把自己的ID作为根网桥和发送网桥，费用设为0。
  - 当一个网桥接收到一个BPDU时，它将源网桥的ID和自己的ID比较，若自己的ID大，自己就不是根网桥。它记录下源网桥的ID和源网桥到达自己这儿路经费用，将路经费用加上接收端口的费用，然后通过所有的其它端口转发这个BPDU。

生成树算法

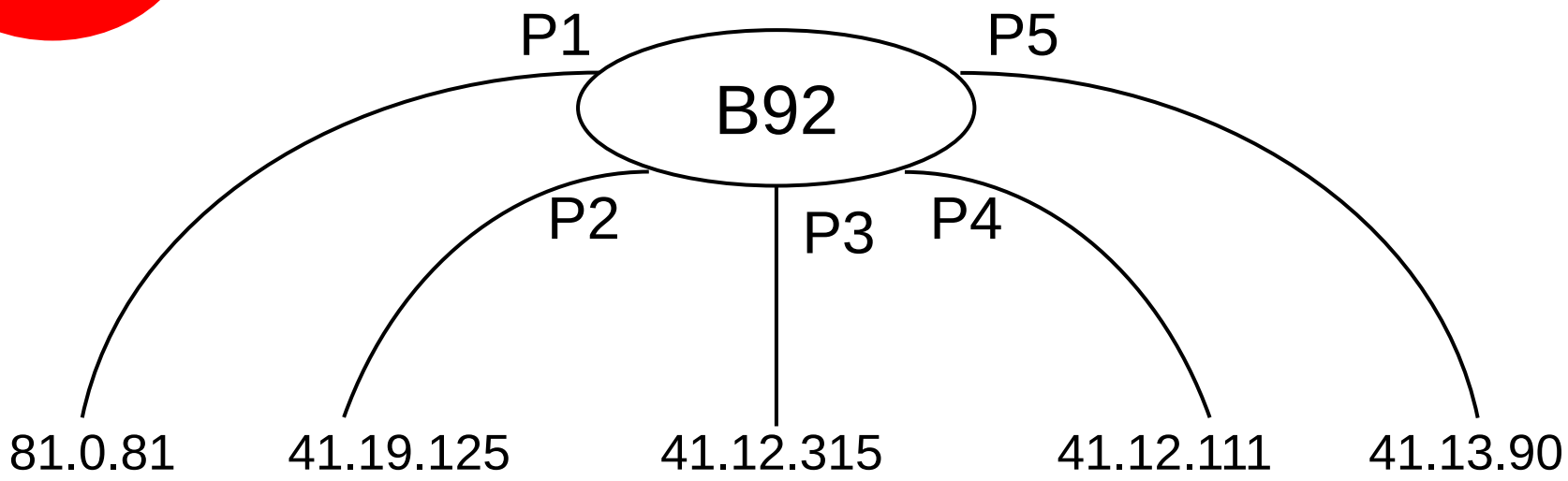
- 每个网桥确定其根端口，依据是：
  - 到根网桥的最低路径成本。
    - 一条链路的带宽越大，它的传输成本就越低。
  - 网桥ID最小
  - 端口ID最小
- 为每个局域网指定一个网桥，依据是：
  - 根路径成本较低
  - 所在的网桥ID值最小
  - 端口ID值最小

Best path to the root

- The sequence of events to determine the best received BPDU is:
  - lowest root bridge id
  - lowest root path cost
  - lowest sender bridge id
  - lowest sender port id

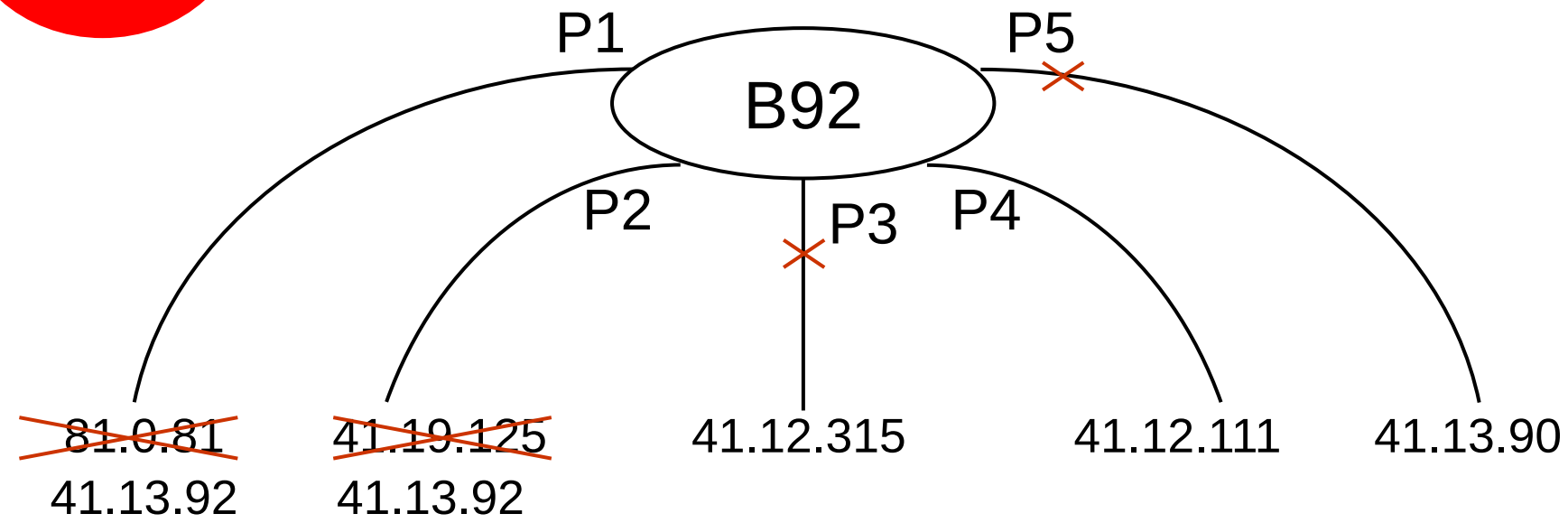
示例

BPDU：根网桥ID.到根费用.发送信息网桥ID



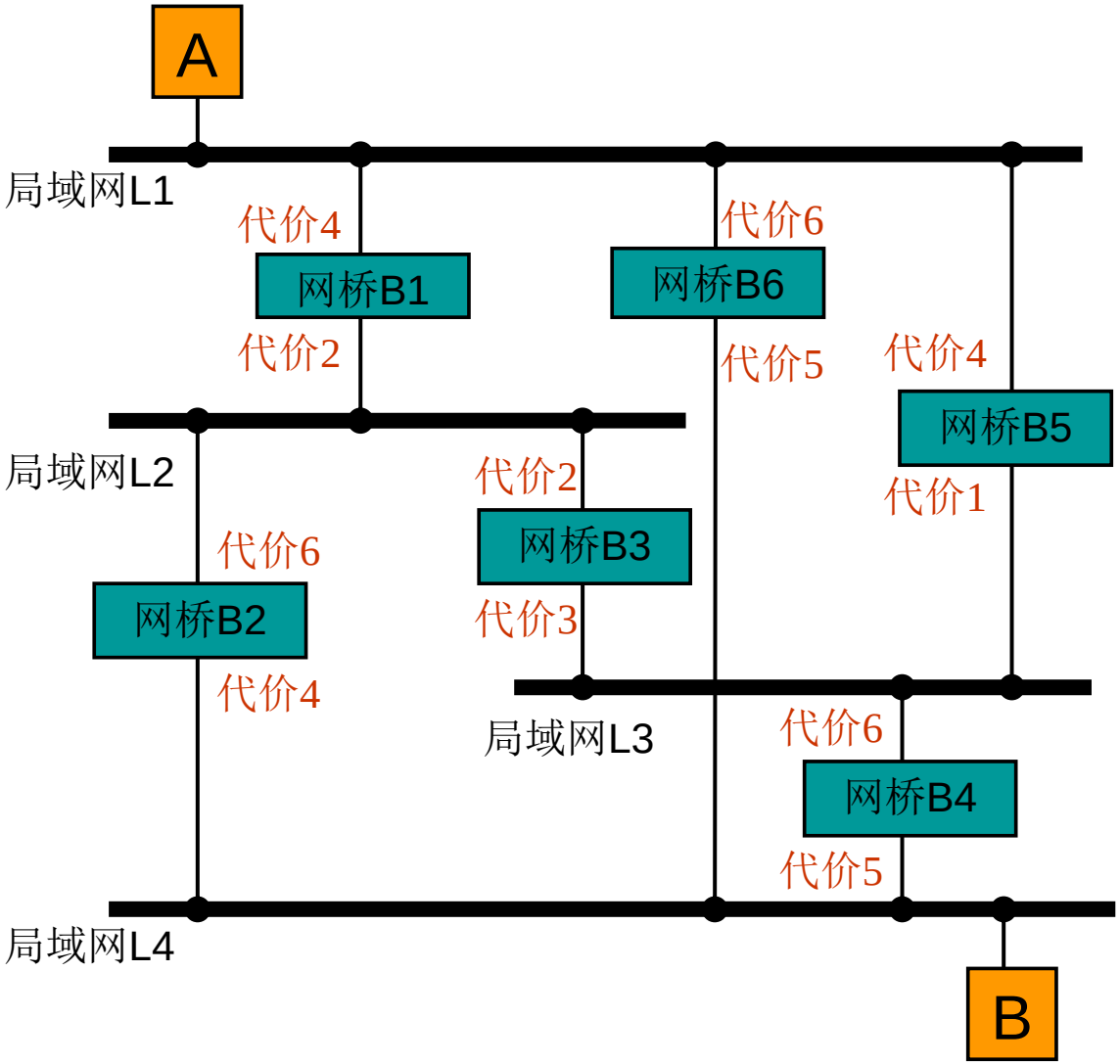
示例(续)

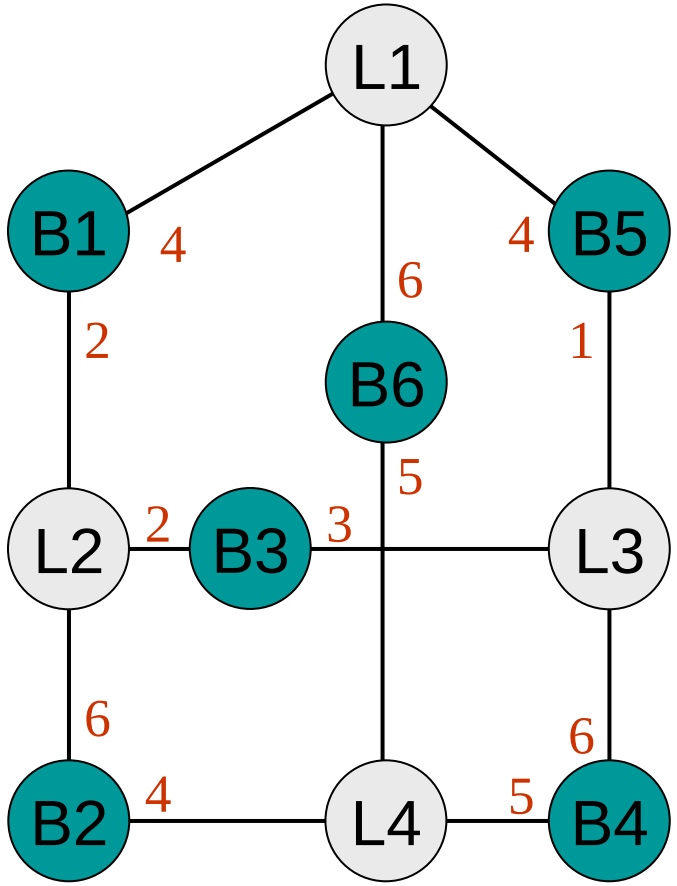
BPDU：根网桥ID.到根费用.发送信息网桥ID



- P4：根端口
- P1、P2：指定端口
- P3、P5：阻塞状态

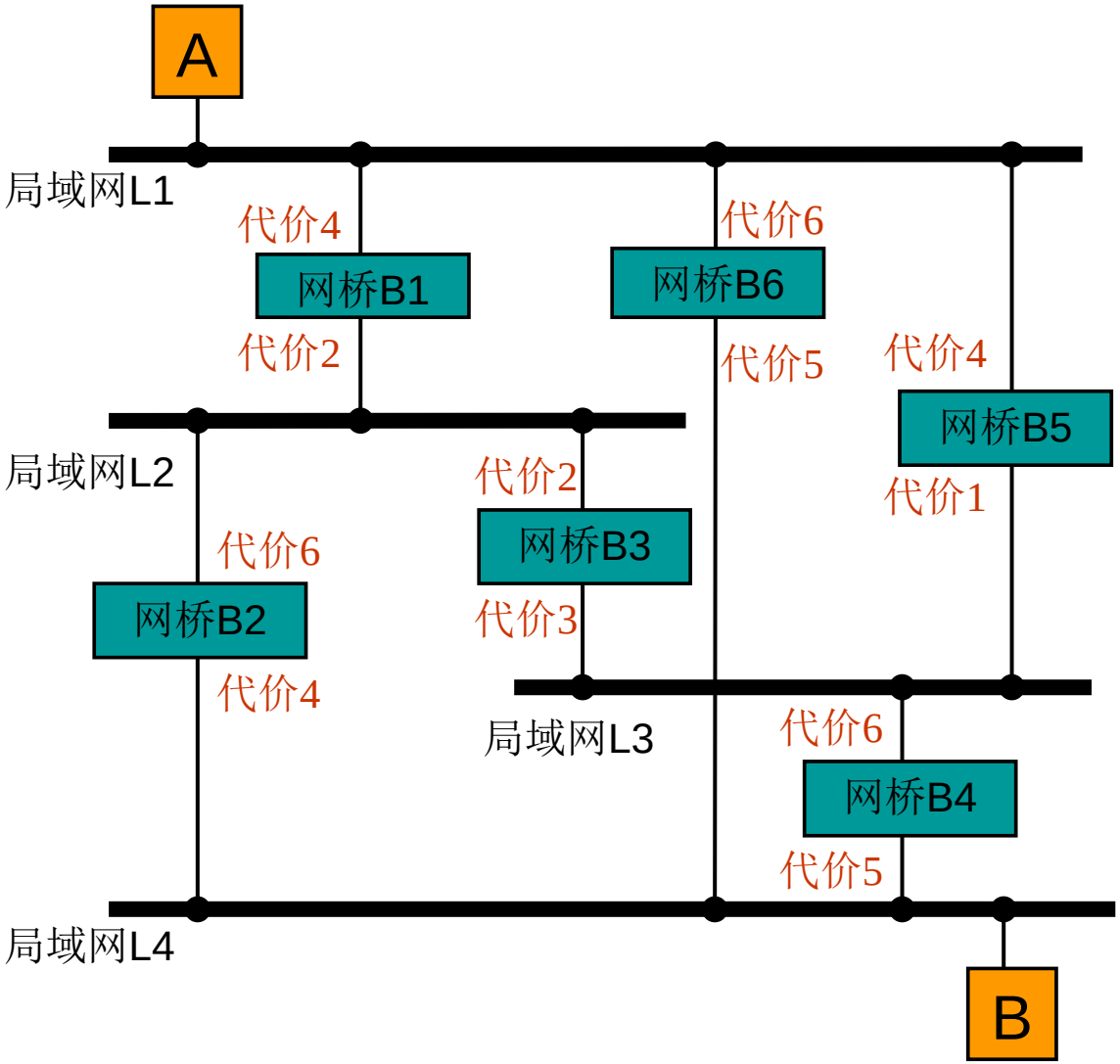
互连局域网及其图表示

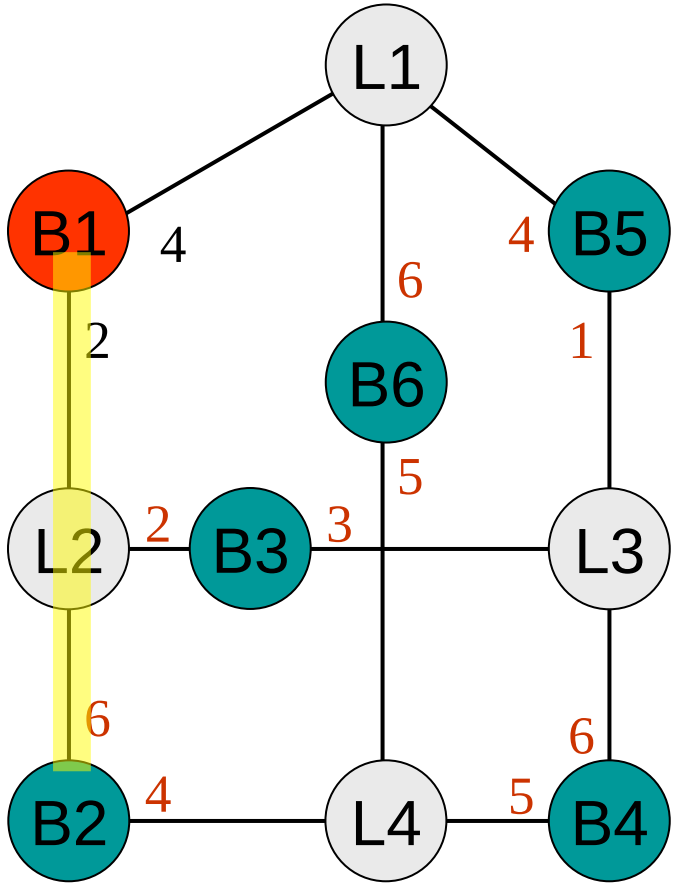

A
局域网L1
代价4
网桥B1
代价2
代价6
网桥B6
代价5
代价4
网桥B5
代价1
局域网L2
代价2
网桥B3
代价3
代价6
网桥B2
代价4
局域网L3
代价6
网桥B4
代价5
局域网L4
B


L1
B1
4
4
B5
6
2
1
B6
5
L2
2
B3
3
L3
6
6
B2
4
L4
5
B4

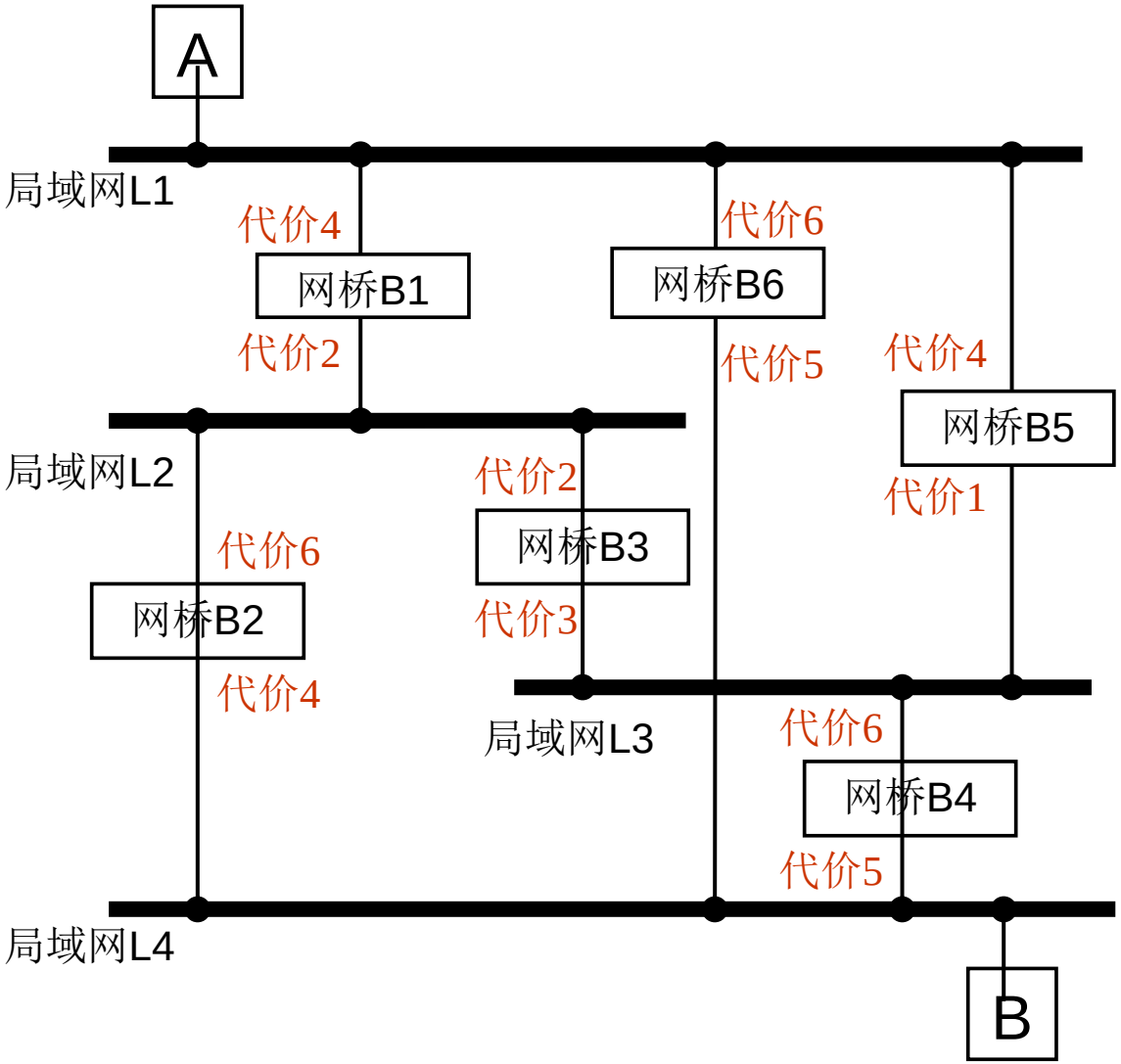
确定根网桥

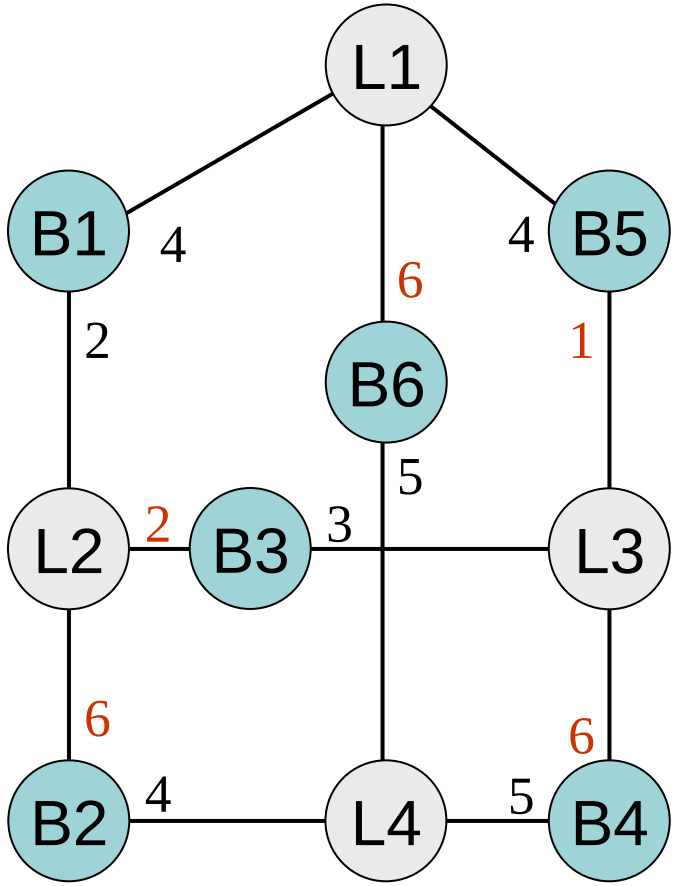
只考虑网桥到网络的费用，不考虑网络到网桥的费用。因此如果走途中标黄的路线，B2到B1的费用为6。



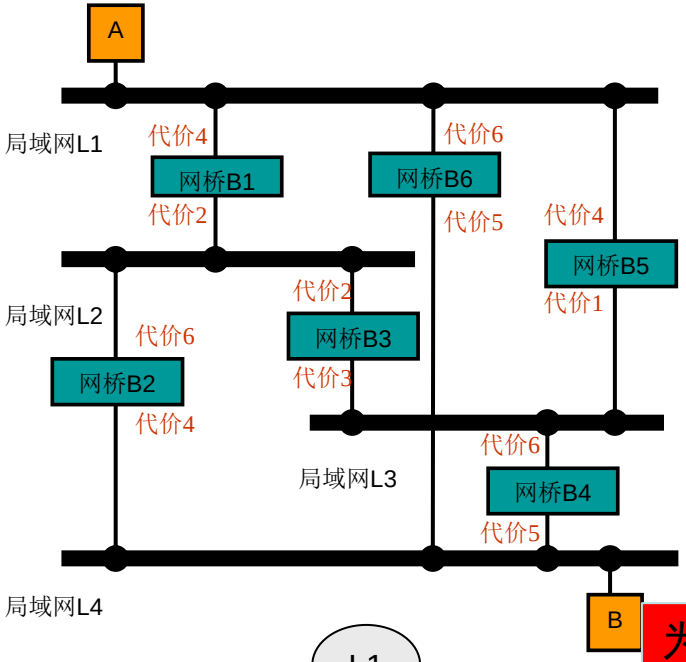


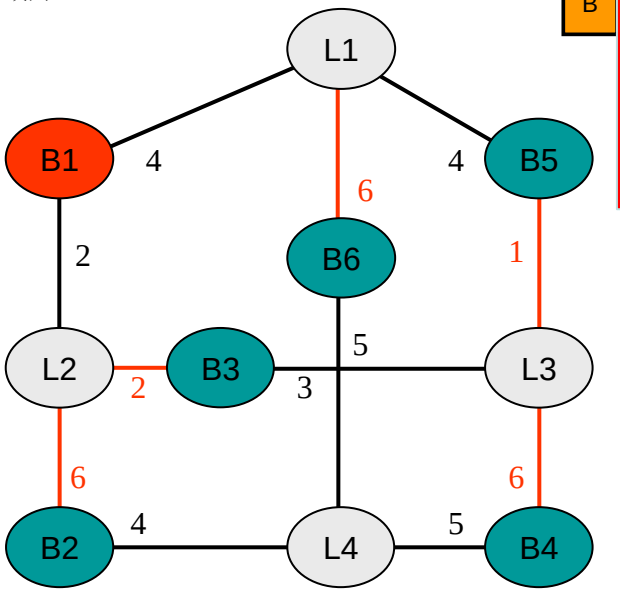
确定根端口


A
局域网L1
代价4
网桥B1
代价2
代价6
网桥B6
代价5
代价4
网桥B5
代价1
局域网L2
代价2
网桥B3
代价3
代价6
网桥B2
代价4
局域网L3
代价6
网桥B4
代价5
局域网L4
B


L1
B1
4
4
B5
6
2
1
B6
5
L2
2
B3
3
L3
6
6
B2
4
L4
5
B4

确定根端口



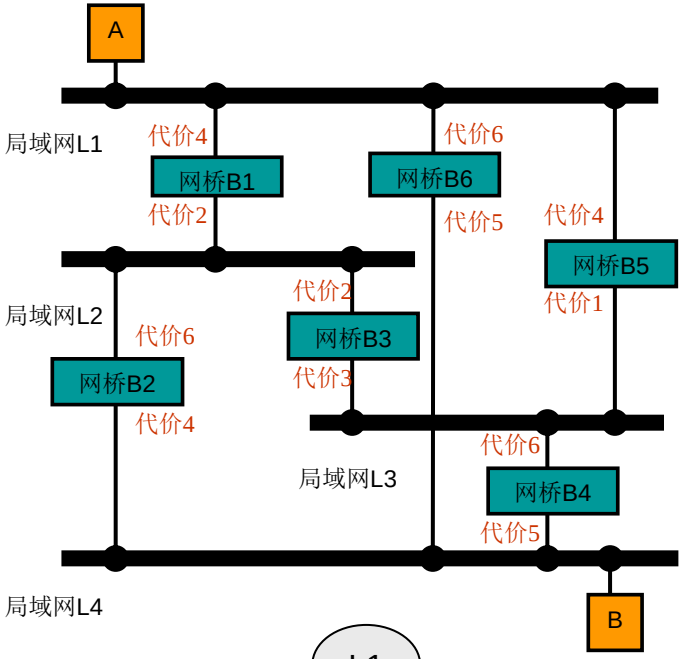


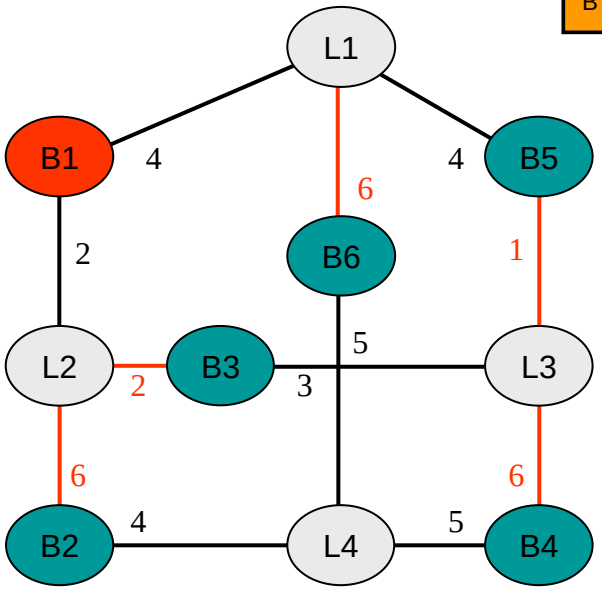
为啥费用是8，3？？？只计算红线的费用

- 到根网桥的最低路径成本。
- 网桥ID最小
- 端口ID最小

网桥	根端口	费用
B2	B2 → L2	6
B3	B3 → L2	2
B4	B4 → L3	8
B5	B5 → L3	3
B6	B6 → L1	6

为每个网络指定网桥

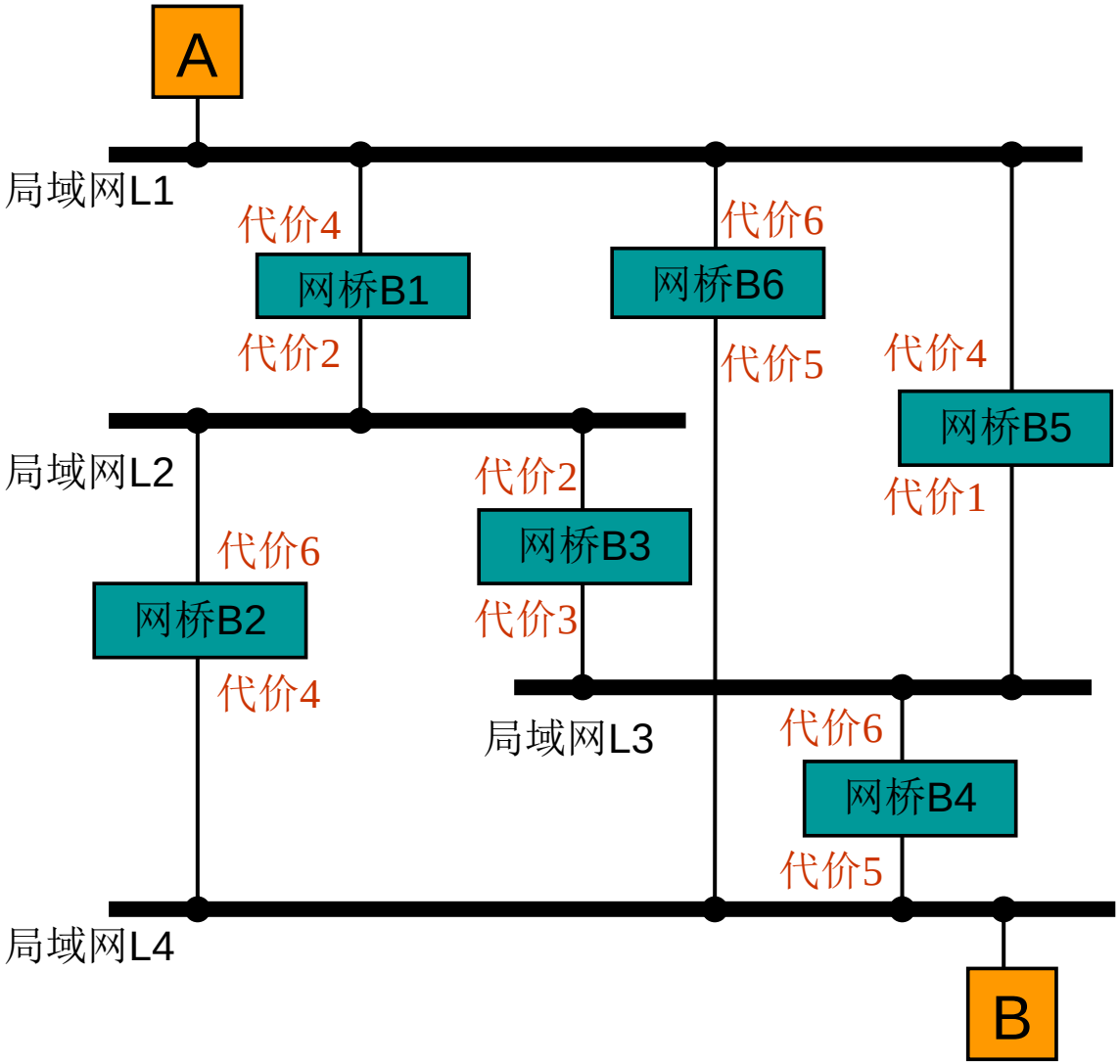


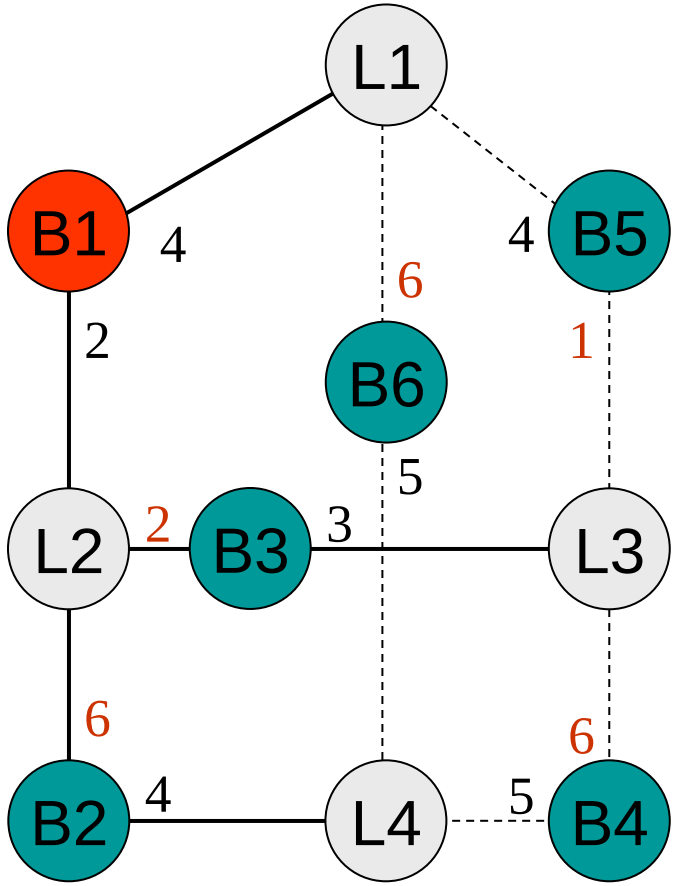


- 通过某网桥至根路径成本较低
- 所在的网桥ID值最小
- 端口ID值最小

网络	指定网桥	费用
L1	L1 → B1	
L2	L2 → B1	
L3	L3 → B3	2
L4	L4 → B2	6

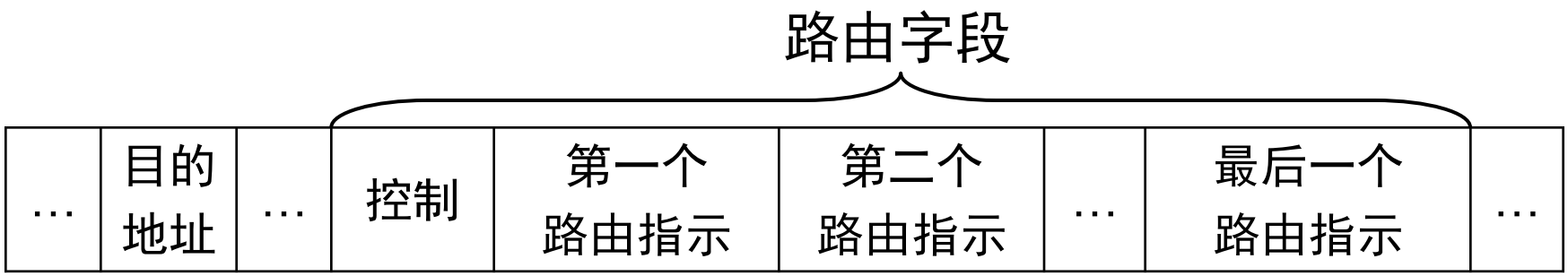
生成树的拓扑及图形表示





源路由网桥

- 源路由网桥：让发送帧的源站点决定转发路由，而不是由网桥来决定。
- 发送站点的网络软件确定到目的站点的路由，并将它存储在帧中，这个路由由路由指示（Route Designator）的序列组成，每个路由指示由一个局域网和一个网桥ID组成。



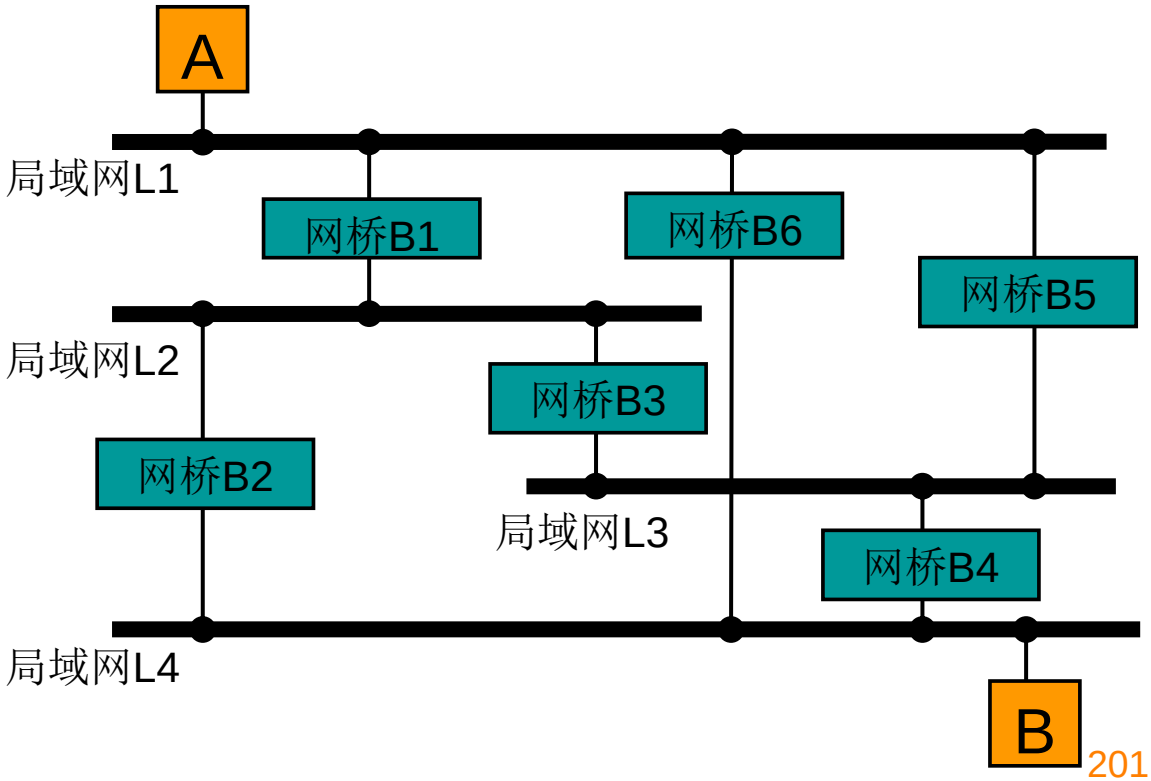
控制：路由字段长度、路由类型、路由方向

路由指示：局域网号、网桥号

源路由网桥示例

- 当一个网桥接收到一个帧时，确定是否有一个路由指示。若有，该路由指示包含该网桥的ID和传输该帧的局域网的ID，网桥接收该帧并将它转发到下一个路由指示指定的局域网上。

例如：假定A发送一个帧给B，并指定一个路由为L1:B5→L3:B4 → L4。



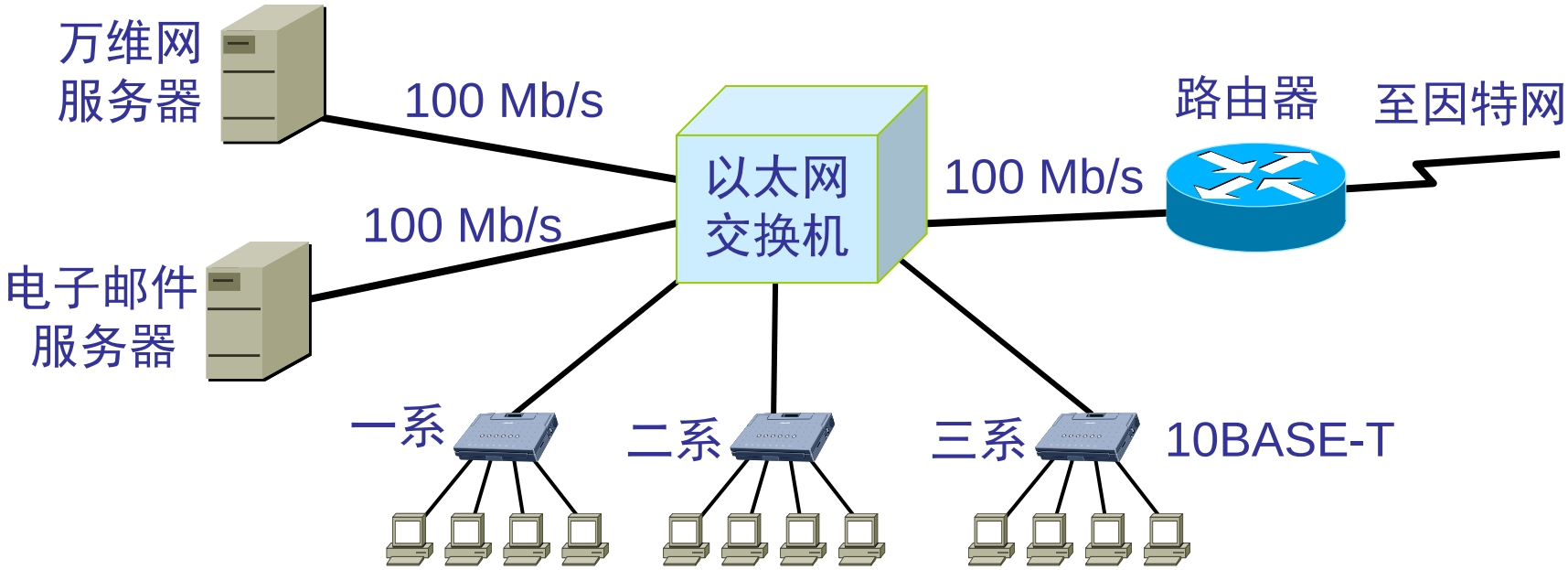
路由的生成

- 路由探索(route discovery)
  - 决定到目的站点的一个路由
- 方法：
  - 源站点发送一个全路由广播帧，目的站点返回一个包含路由的非广播帧
  - 源站点沿生成树给目的发一个单路由广播帧，目的站点返回一个全路由广播帧作为应答

4.7.3 第二层交换机

- 第二层交换机是多接口网桥
- 交换机从某一节点收到一个帧后，立即在其内存中的地址表（端口号－MAC地址）进行查找，以确认该目的MAC的网卡连接在哪一个节点上，然后将该帧转发至该节点。
- 每个端口有各自的带宽，各端口之间并行工作，可以提高网络吞吐量。

用以太网交换机扩展局域网



两种典型的交换技术

- 直通式（Cut Through）
  - 它在输入端口检测到一个数据包时，检查该包的包头，获取包的目的地址，启动内部的动态查找表转换成相应的输出端口，在输入与输出交叉处接通，把数据包直通到相应的端口，实现交换功能。
  - 优点：由于不需要存储，延迟非常小、交换非常快。
  - 缺点：数据包内容交换机不保存，不能提供错误检测能力。由于没有缓存，不能将具有不同速率的输入/输出端口直接接通，而且容易丢包。
- 存储转发（Store & Forward）
  - 它把输入端口的数据包检查，在对错误包处理后才取出数据包的目的地址，通过查找表转换成输出端口送出包。
  - 缺点：在数据处理时延时大
  - 优点：它可以对进入交换机的数据包进行错误检测，有效地改善网络性能。可以支持不同速度的端口间的转换，保持高速端口与低速端口间的协同工作。

虚拟局域网(VLAN)

- 增加局域网可扩展性
- 优点主要有三个：
  - 端口的分隔。即便在同一个交换机上，处于不同VLAN的端口也是不能通信的。这样一个物理的交换机可以当作多个逻辑的交换机使用。
  - 网络的安全。不同VLAN不能直接通信，杜绝了广播信息的不安全性。
  - 灵活的管理。更改用户所属的网络不必换端口和连线，只更改软件配置就可以了。

VLAN示例--三个虚拟局域网