计算机网络

胡亮 等编著

第4章 数据链路层

4.1 数据链路层基本原理

4.2 数据链路控制协议

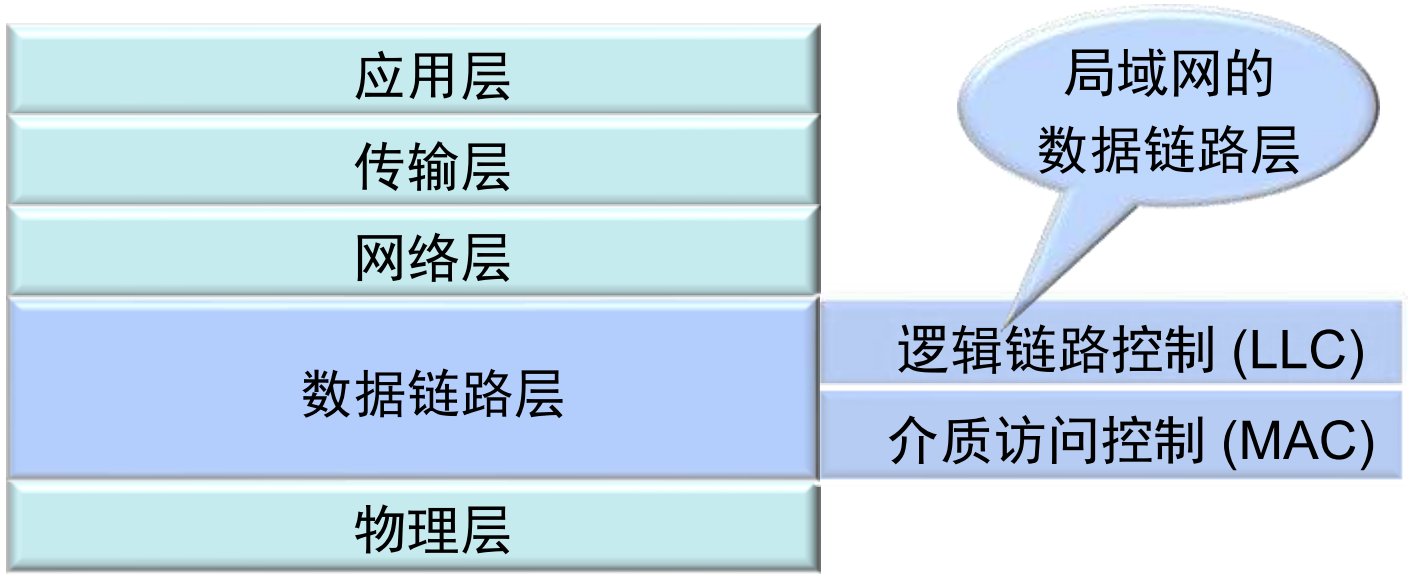
4.3 介质访问控制协议

4.4 数据链路层网络互连

4.5 本章总结

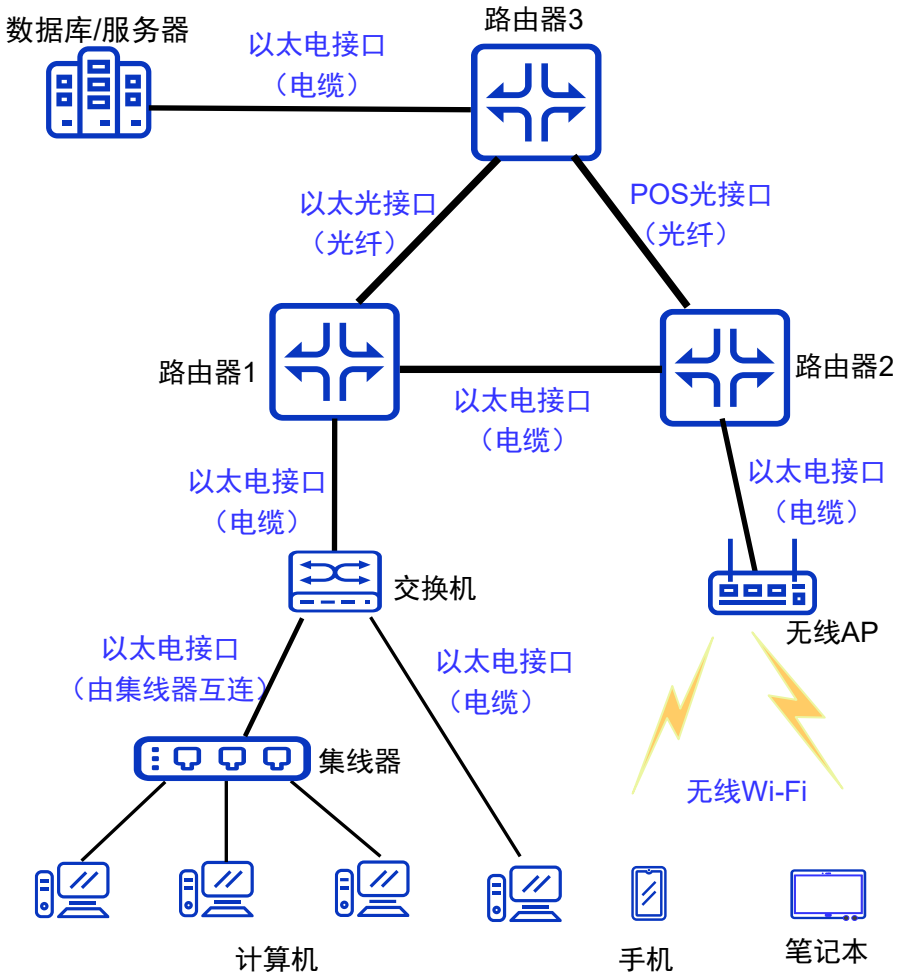
4.1 数据链路层基本原理

- 在 TCP/IP 分层模型中，数据链路层介于物理层和网络层之间：
  - 向下：利用物理层提供的比特流服务（服务可能有差错）
  - 向上：向网络层提供数据包的收发服务（无差错的数据包传送服务）



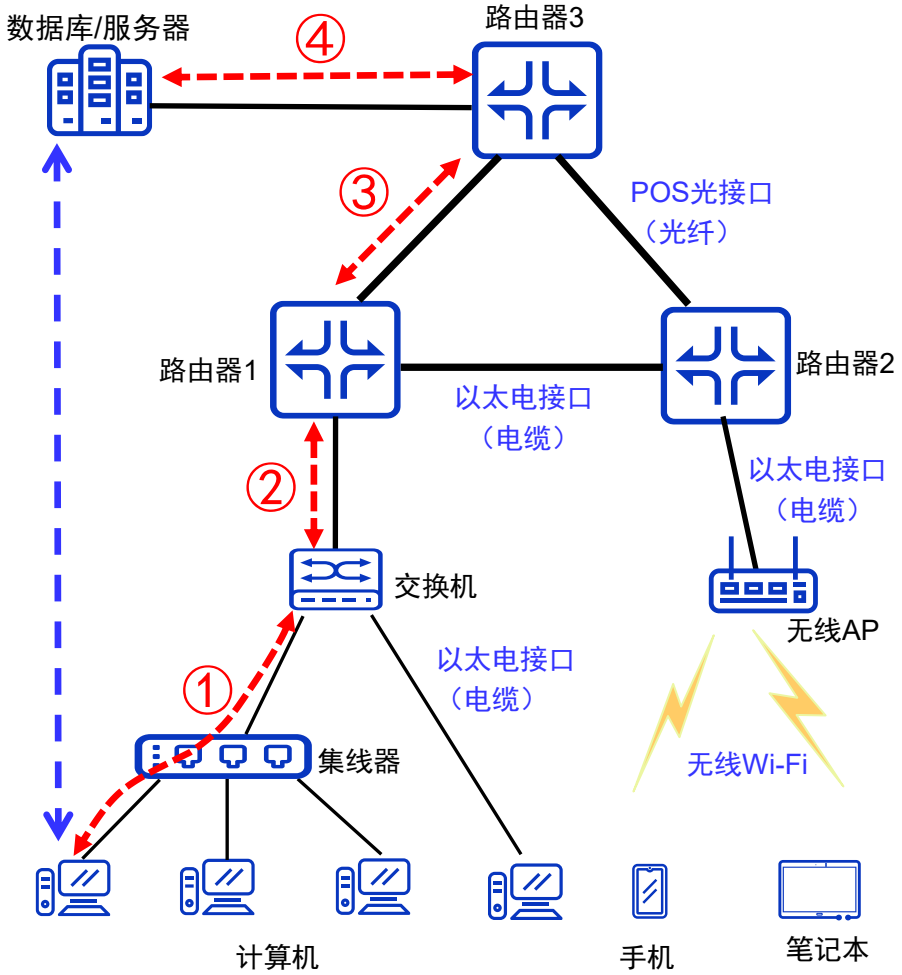
TCP/IP 协议栈

网络是如何构建的?



- 构建一个可连通的网络：
  - 网络设备互连成网
    - 路由器、交换机、无线 AP、集线器等网络设备通过设备接口、物理线路互连
    - 设备接口可以是 POS(Packet Over SONET/SDH)光接口、以太光接口、以太电接口等
    - 接口间互连线路可以是光纤，也可以是网线电缆
    - 用什么接口、什么线路互连，取决于传输距离、所需带宽等
  - 用户终端接入到网络
    - 有线：服务器、计算机等用户终端通过其网卡、网线接入到网络中
    - 无线：手机、笔记本等用户终端通过无线 Wi-Fi 接入到网络中
  - 接口互连的信道方式
    - 可以以点到点信道方式互连
    - 也可以通过集线器以广播信道方式互连

对互连的网络思考一个问题



- 网络中任意两个结点，不管是否直连，从网络层层面看，都能够相互通信：
  - 不直连的两个结点，通过中间结点转发
  - 直连的两个结点，直接发送到对方
- 不管结点间是直接发送，还是通过中间结点转发，最终都要通过两个直接相连的结点发送数据、接收数据
- 从物理层看，在相连接口和线路上，是在以比特流方式传输数据

➢ 问题：在物理层提供比特流传输服务基础上，相连结点如何实现正确的数据发送与接收？

深入思考几个问题

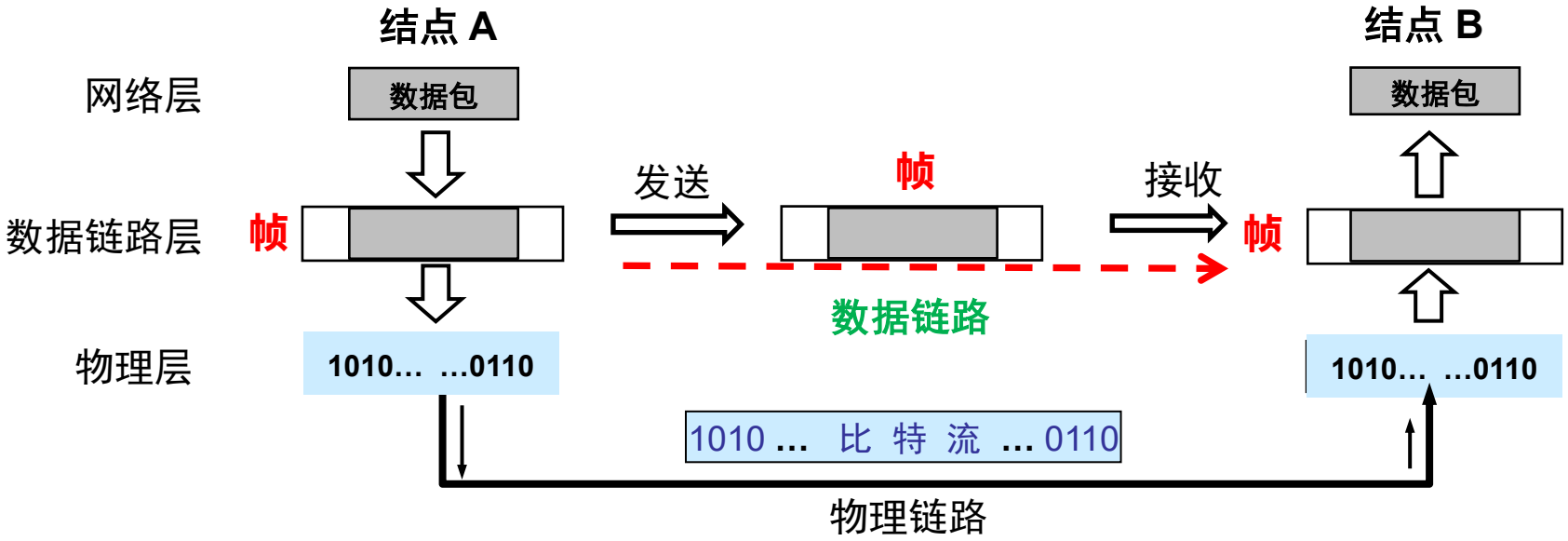
- 相邻结点间，实现正确的数据传输要解决哪些问题？
  - 问题1：分帧与组帧
    - 接收方接收到一串0/1比特流后，如何识别边界、将比特流分割成帧？
  - 问题2：差错控制
    - 数据在线路上传输，可能受到噪声干扰并发生错误，接收方如何检测到错误并纠正错误？
  - 问题3：流量控制
    - 发送方应该如何控制发送速度，保证接收方来得及接收？
  - 问题4：传输协议
    - 收发双方如何协同，需要遵守什么协议？

？

数据链路层的任务和功能

数据链路层的任务

- 提供相邻结点之间的可靠通信，具体体现为：
  - 结点：结点可以是路由器等网络设备，也可以是用户主机或其它业务终端
  - 场合：处理相邻节点间的数据传输
  - 对象：传输的数据单元是帧（Frame）
  - 目的：将不可靠的物理链路变为可靠的逻辑链路

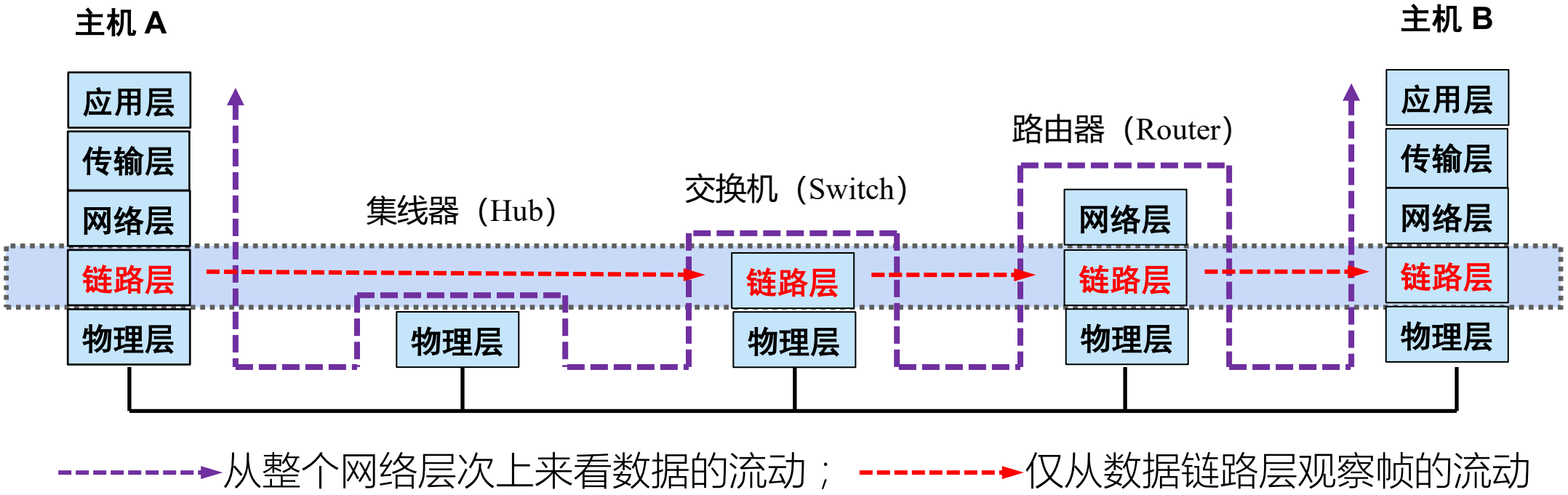


数据链路层的主要功能

- 为网络层提供服务
  - 无确认无连接服务、有确认无连接服务、有确认面向连接服务
- 链路管理
  - 介质访问控制
  - 链路建立、链路维持、链路释放
- 成帧
  - 将数据封装成适合在链路上传送的基本单元
  - 帧格式与传输介质、双方启用的协议有关
- 差错控制
  - 处理数据帧的损坏、丢失、重复和乱序
  - 能检测错误并尽可能纠正错误，为网络层提供可靠的数据传送服务
- 流量控制
  - 发送端控制发送速率
  - 防止因发送端的数据发送速度过快造成信道拥挤和数据丢失

数据链路层在网络中的地位

- 主机A 到主机B所经过的网络链路可以是多种不同类型的链路
- 数据链路层是网络相邻结点间通信非常重要、必须要有的一层
- 网络中的主机、交换机、路由器等都必须实现数据链路层



数据链路层提供的服务

- 相邻结点在链路上进行通信时，需要考虑以下两点：

- 收发数据前，双方是否需要建立连接？
  - 双方协商好参数
  - 发送方做好发送数据的准备，接收方做好接收数据的准备

信道是否要管理？

- 对收发的数据帧，是否需要确认？
  - 接收方收到数据后要给发送方发一个确认消息
  - 发送方通过确认消息确保接收方已经收到数据，否则要重新发送

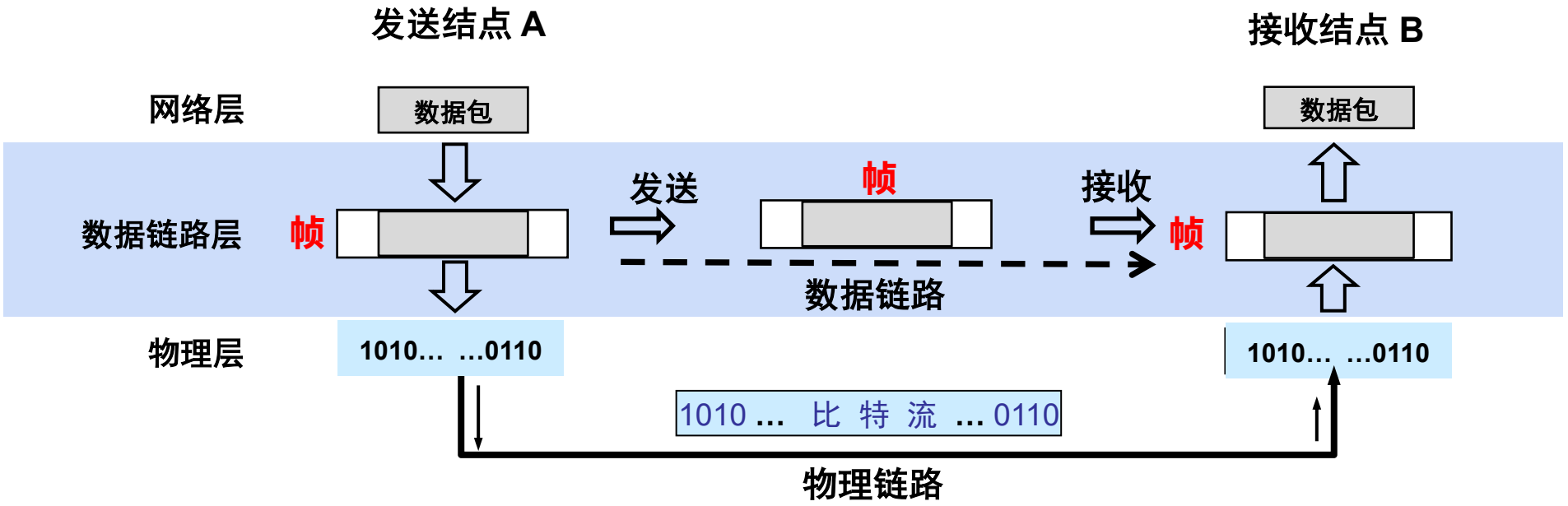
数据收发可靠性是否要保证？

数据链路层提供的服务

- 数据链路层提供的服务可以分为两类：
- 无连接服务(二种)
  - 无确认无连接服务（Unacknowledged connectionless ）
    - 接收方不对收到的帧进行确认
    - 适用场景：误码率低的可靠信道；实时通信
  - 有确认无连接服务（ Acknowledged connectionless ）
    - 每一帧都得到单独的确认
    - 适用场景：不可靠的信道（无线信道）
- 面向连接的服务
  - 适用场景：长延迟的不可靠信道

数据链路层数据单元

- 帧（frame)是数据链路层使用的协议数据单元
- 在发送端“组帧”，在接收端“拆帧”



成帧（framing）

- 成帧：把一串比特流分割并插入标签的技术
- 通过成帧可实现:
  - 分段传输
  - 错误隔离
  - 误差校验
  - 数据恢复
- 不同的数据链路层协议定义不同的帧，如PPP帧、MAC帧等
- 关键问题：如何进行帧定界？
  - 帧定界又称帧同步
  - 指接收方必须能从物理层接收的比特流中区分出一帧的开始和结束
  - 选择何种定界符？
  - 如果传输的数据中含有定界符该如何处理？

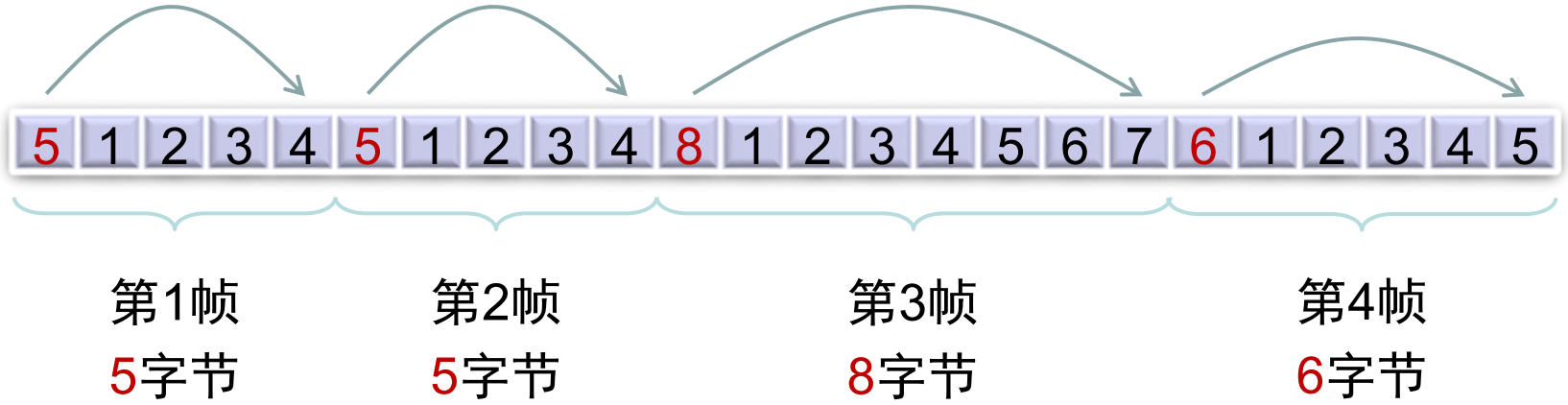
成帧方法

- 字节计数法
  - 早期技术，现在很少使用
- 字节填充的标志字节法
  - 硬件实现比较困难，适合软件实现，如PPP协议
- 比特填充的标志比特法
  - 利于物理层保持同步
  - 硬件实现比较容易，如USB、 HDLC 协议
- 编码违禁法
  - 在物理层实现，如802以太网链路

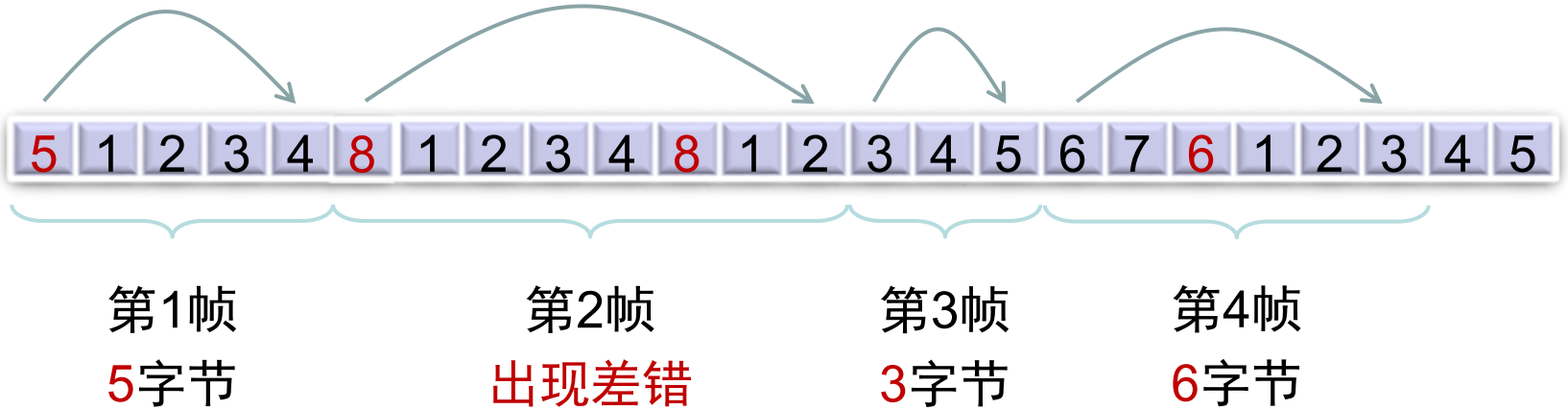
➢ 目前成帧的主要方法：比特填充的标志比特法和编码违禁法

字节计数法

- 无差错时的情形

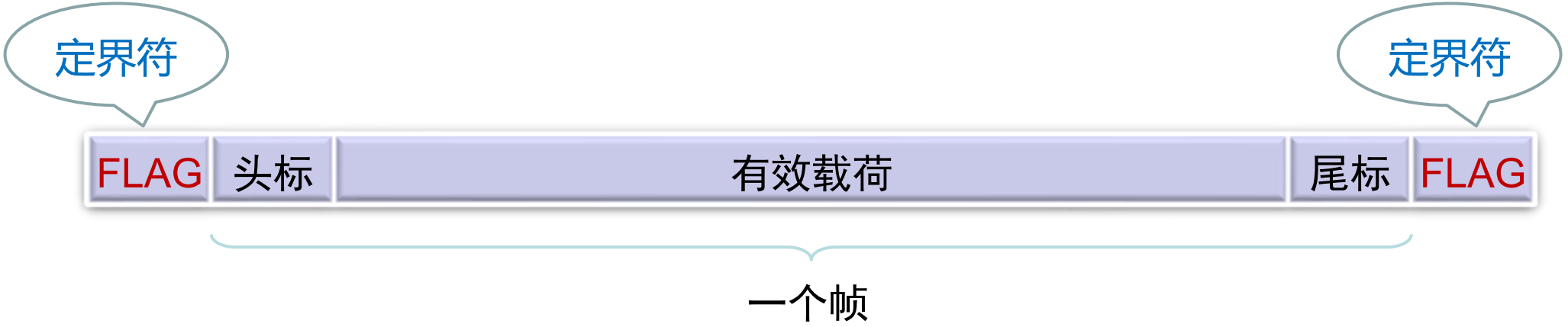


- 发生差错时的情形

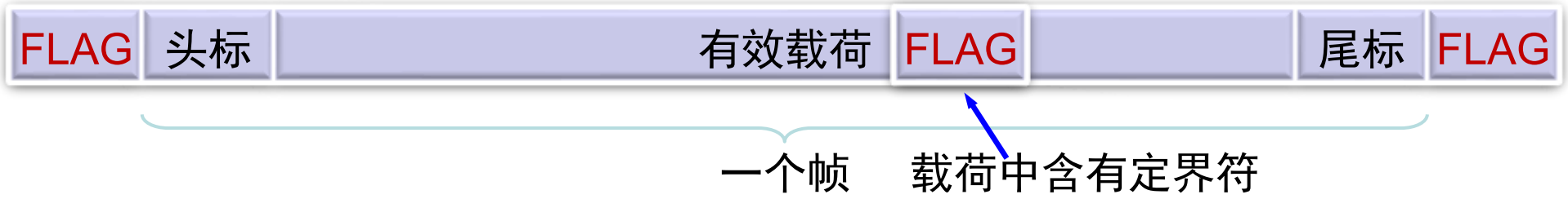


字节填充的标志字节法

- 用特定的字节进行帧定界
  - 开始字节和结束字节可以相同，也可不同
  - 帧和帧之间可以连续发定界符（FLAG），用来作同步填充

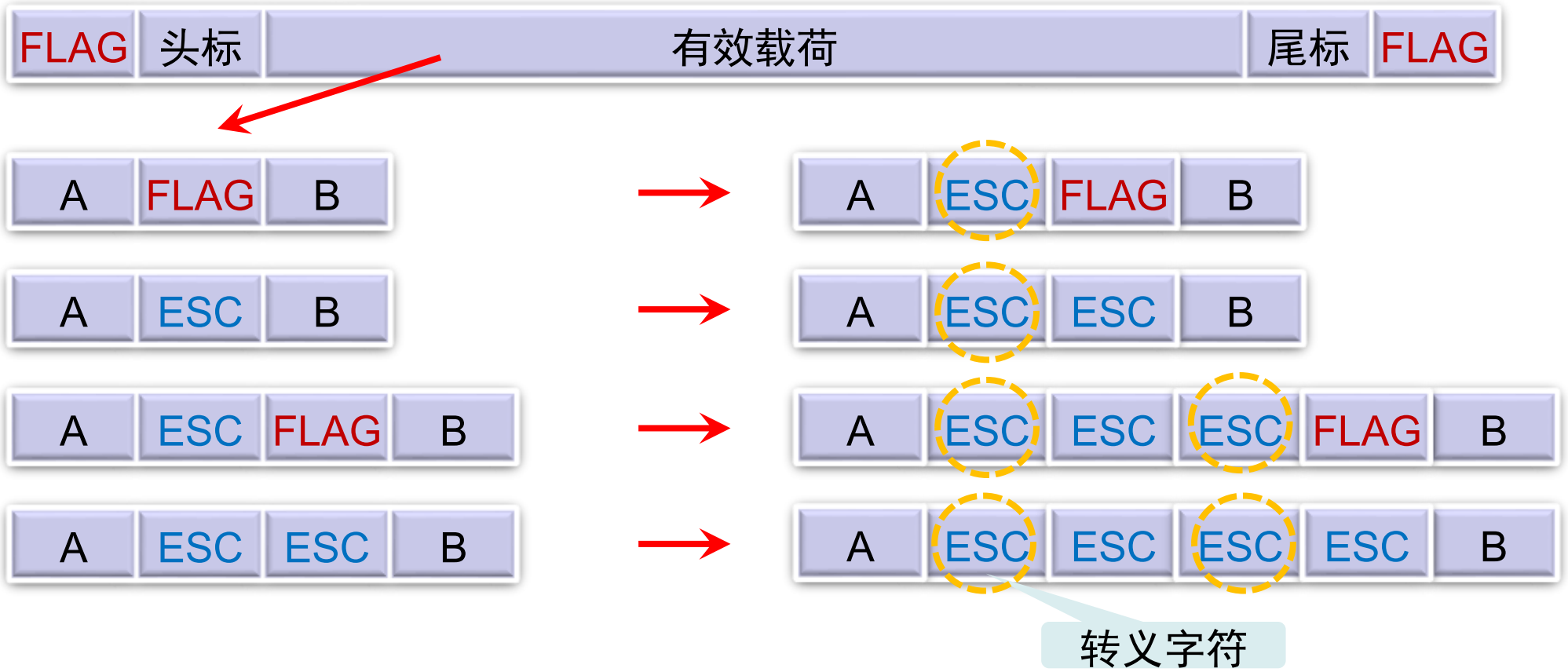


- 关键问题：有效载荷部分包含与“定界符”相同的字节会有什么问题?



字节填充的标志字节法

- 载荷中含有 FLAG定界符的处理方法
  - 前面填充一个特定的字符ESC（ASCII字符1BH，称为转义字符)
  - 载荷中如果出现转义字符ESC，则其前面也要填充转义字符ESC



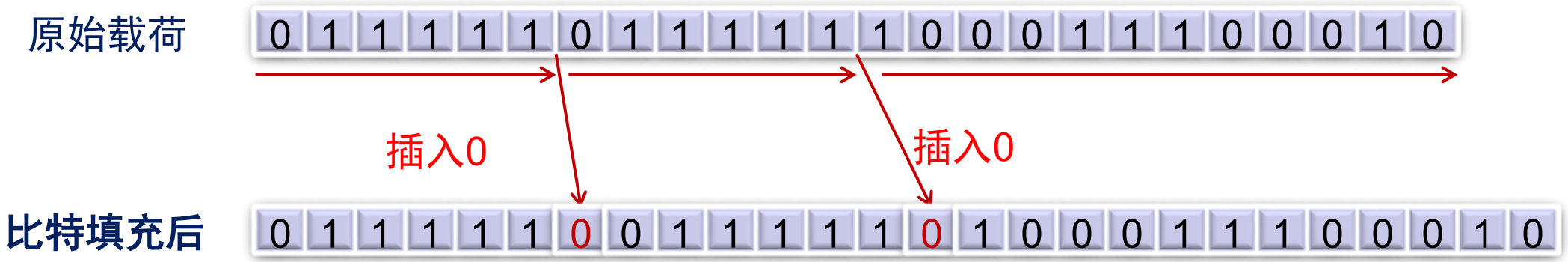
比特填充的标志比特法

- 用一组特定的比特组合作为定界符，来标志一帧的开始和结束
  - 帧的划分在比特级完成
  - 常用于面向二进制位的串行通信中
  - 典型实例：HDLC协议，使用01111110 （0x7E）比特组合进行帧定界

- 问题：有效载荷部分包含与定界符相同的比特组合（01111110），会怎样？

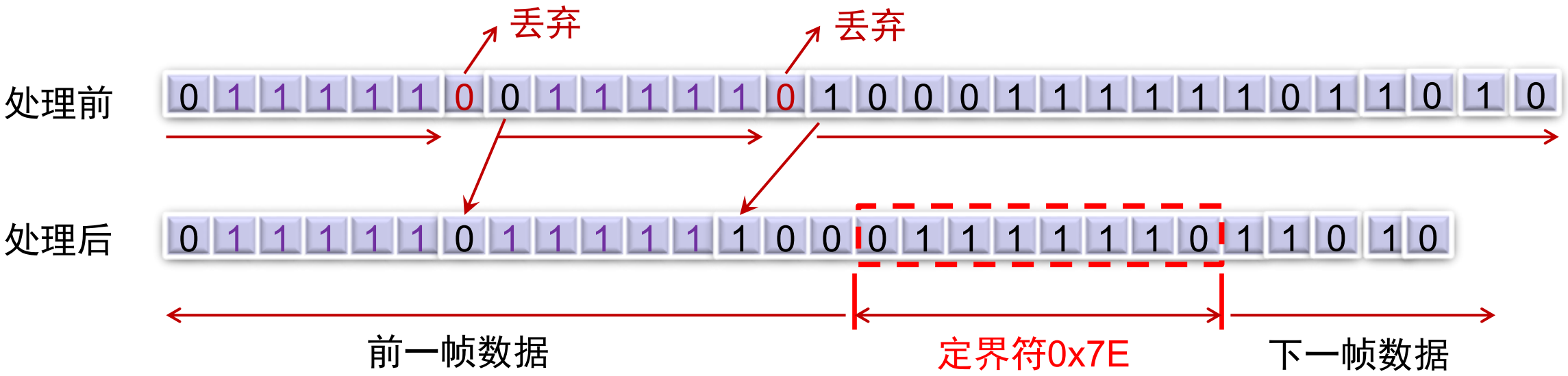
比特填充的标志比特法

- 发送端的处理：比特填充
  - 发送方检查有效载荷，若出现了连续5个1比特后，则直接插入1个0比特，防止和定界符的比特模式 01111110 相同
  - 比特填充法是比特位操作，硬件实现起来比较方便



比特填充的标志比特法

- 接收端的处理
  - 检查接收到的比特流，若出现连续5个1后，则检查下一比特
  - 若为0：则后面为有效载荷，该0比特是发送方填充进去的，直接丢弃
  - 若为1：
    - 如果接下来的一比特是0，则构成定界符，一帧结束
    - 如果接下来一比特是1，则需要作出错处理



编码违禁法

- 利用物理介质编码的“违法标志”来界定帧的开始与结束
  - 仅适用于采用冗余编码的方案中
  - 编码违禁法在物理层实现
- 通常有以下几种编码方法：
  - 曼彻斯特编码 / 差分曼彻斯特编码
    - 高-高或低-低电平对在数据的编码中是违禁、不用的，可以用于帧定界
    - 用于 10Mbps以太网/802.5令牌环网
  - 4B/5B 编码
    - 其中有16种编码未使用，可用作帧定界符
    - 用于100BASE-TX、100BASE-FX快速以太网
  - 8B/10B编码
    - 其中不用的编码用作帧定界符、传输空闲状态等控制码
    - 用于1000BASE-SX、1000BASE-LX千兆以太网802.3z标准中

信道分配与介质访问控制

- MAC子层要解决多点连接环境下如何使用信道的问题
- 静态信道分配:
  - 多路复用技术，将单个信道划分成多个静态子信道
  - 效率低，网络中数据流量具有突发性的特点
- 动态信道分配
  - 5个关键假设：流量独立，单信道，冲突可检测，时间连续或分槽，载波侦听
  - 多路访问协议：如何在一个多点连接网络中获得信道使用权
    - 竞争协议
    - 无冲突协议
    - 有限竞争协议

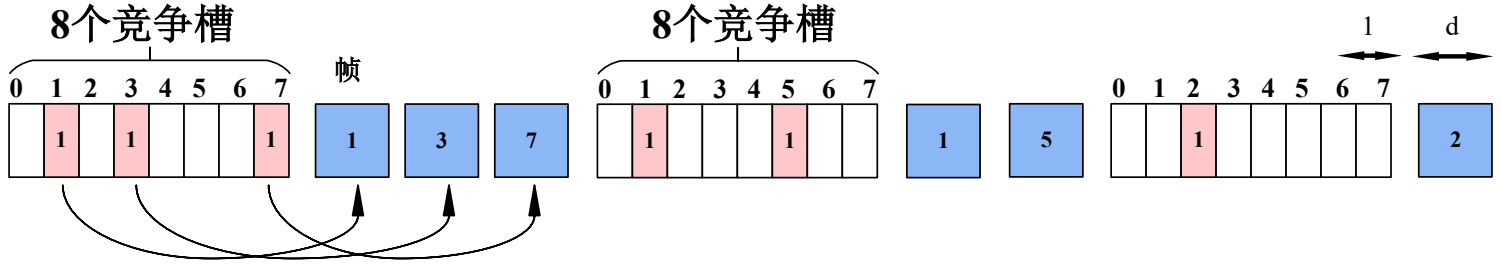
动态分配信道（一）：竞争协议

- Aloha是一种基础的多路访问通信协议
  - 纯Aloha：有数据帧就发送，利用来自接收方的确认来确保帧发送成功，信道利用率低
  - 分槽Aloha：引入了离散槽来减少冲突、提高最大吞吐量
- CSMA/CD：对共享信道的以太网，IEEE 802.3使用载波侦听多路访问/冲突检测
- CSMA/CA：对共享信道的无线局域网，IEEE 802.11使用载波侦听多路访问/冲突避免

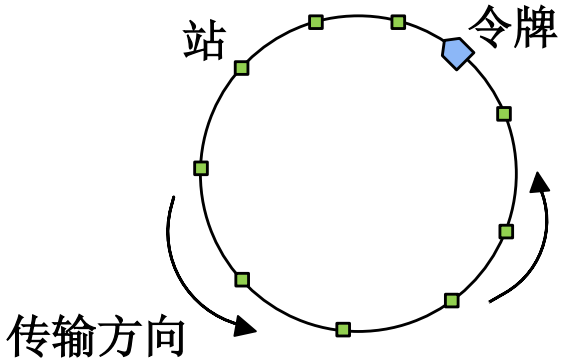
动态分配信道（二）：无冲突竞争协议

- 无冲突竞争协议包括：基本位图协议、令牌传递协议和二进制倒计数

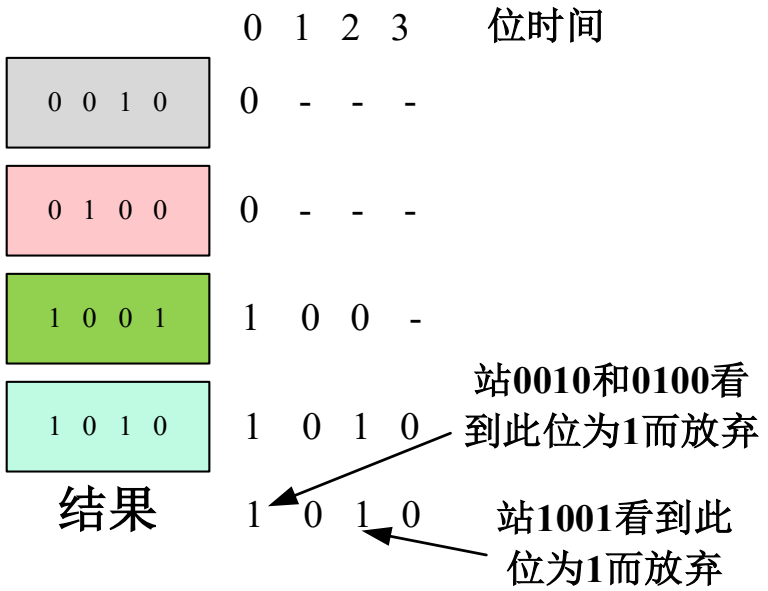
（1）基本位图协议



（2）令牌传递

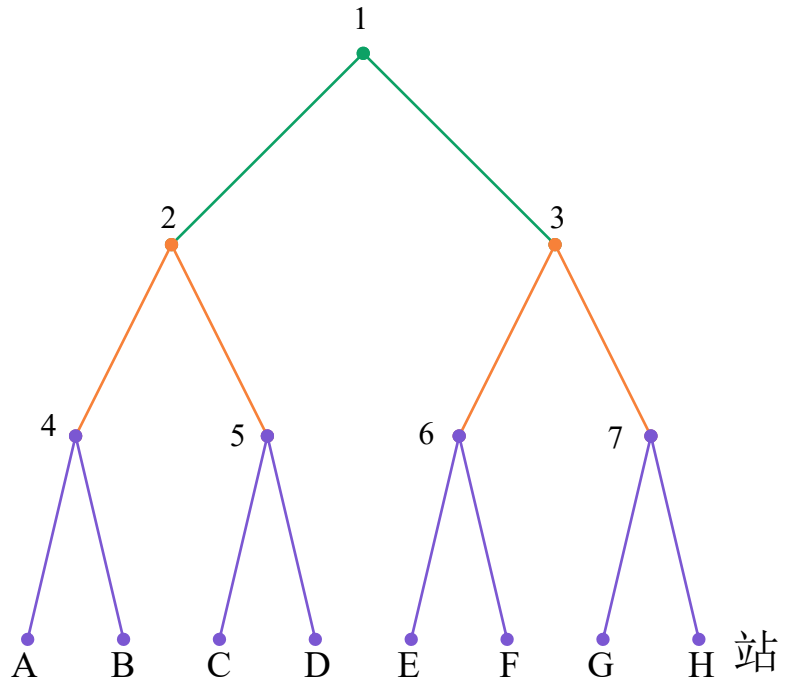


（3）二进制倒计数



动态分配信道（三）：有限竞争协议

- 竞争协议：在负载较轻的情况下更为理想，因为冲突少、延迟短
- 无冲突协议：在低负载情况下延迟相对高，随着负载增加，信道的效率反而提高，因开销固定
- 有限竞争协议
  - 在低负载下用竞争法获得较短的延迟
  - 在高负载下用无冲突技术获得良好的信道效率
  - 例：自适应树遍历协议



数据链路管理

- 线路配置
  - 点到点信道
  - 广播信道
- 传输方式
  - 单工、双工、半双工
  - 并行与串行
  - 同步与异步

线路规程：

通信双方约定好的协议、规则

- 询问/确认方式（ENQ/ACK）
- 轮询/选择方式（Poll/Select）

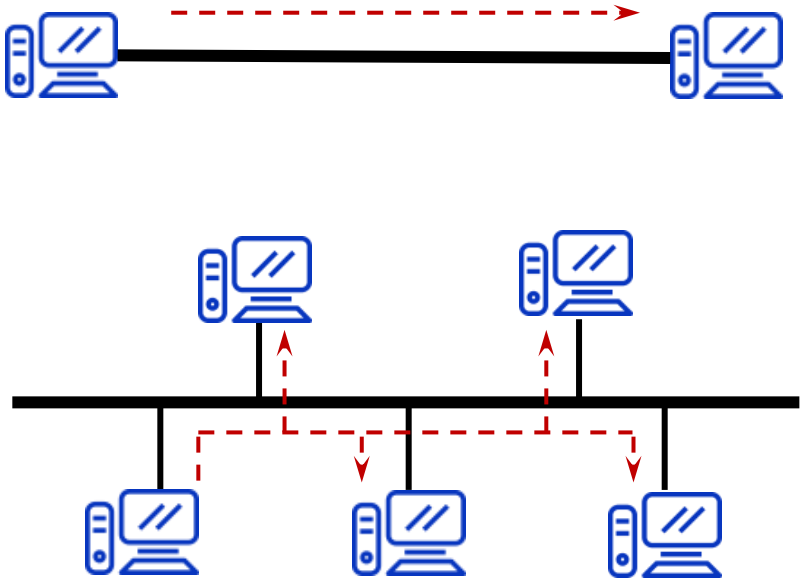
链路状态管理：

发送方确定接收方处于接收状态

- 数据链路建立：收发数据前
- 数据链路维持：收发数据中
- 数据链路释放：数据收发完成后

线路配置

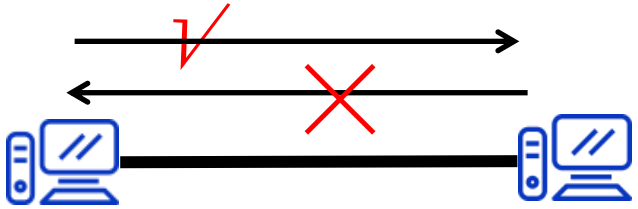
- 两个或两个以上的网络结点用线路连接起来的方式
- 两种线路配置方式：
  - 点到点连接
    - 两个结点间由专用线路连接
    - 信道容量都用于这两个结点间的数据传输
    - 使用一对一的点对点通信方式
  - 多点连接（广播信道）
    - 两个以上的结点共享一条线路
    - 通过复用技术实现信道容量共享，必须使用专用的共享信道协议来协调结点的数据发送
    - 使用一对多的广播通信方式

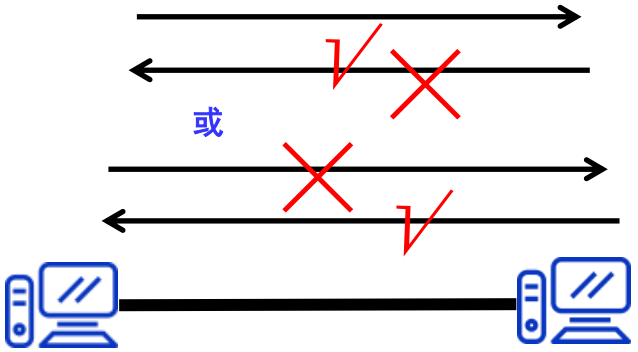


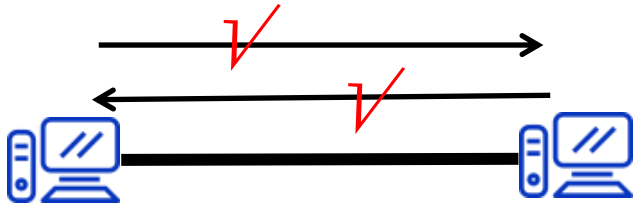
传输方式

- 传输方式定义了比特流从一个设备传到另一个设备的方式，包括：
  - 单工、半双工和全双工通信
  - 并行传输和串行传输
  - 同步传输和异步传输

传输方式：单工、半双工和全双工



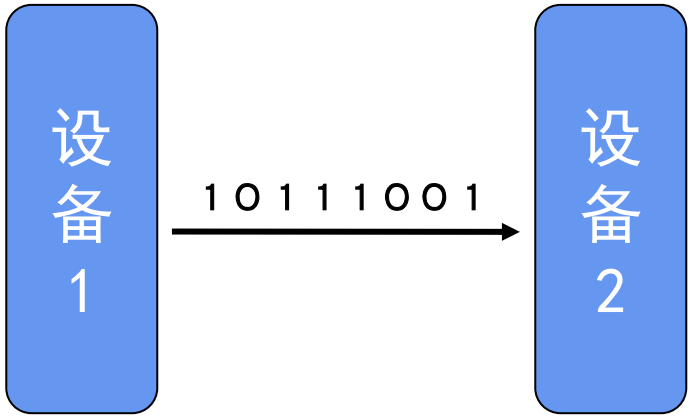




- 单工
  - 数据只能在一个方向上传输
  - 一个结点只能发送，另一个结点只能接收
  - 通信是单向进行的
- 半双工
  - 两个结点都可以发送和接收，但不能同时发送和接收
  - 一个结点在发送时，另一个结点只能接收
- 全双工
  - 两个结点可同时发送和接收
  - 双向信号共享链路带宽
    - 情况1：链路具有两条物理上独立的传输路径，一条发送，一条接收
    - 情况2：将信道带宽一分为二（复用），同时传输两个方向的信号

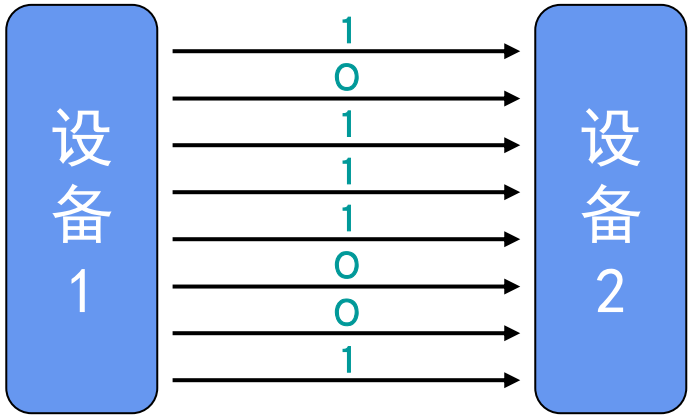
传输方式：串行传输

- 每个时钟脉冲只发送一个比特
- 使用一条线路，逐个比特传送
- 应用场合：
  - 短距离传输
  - 长距离通信



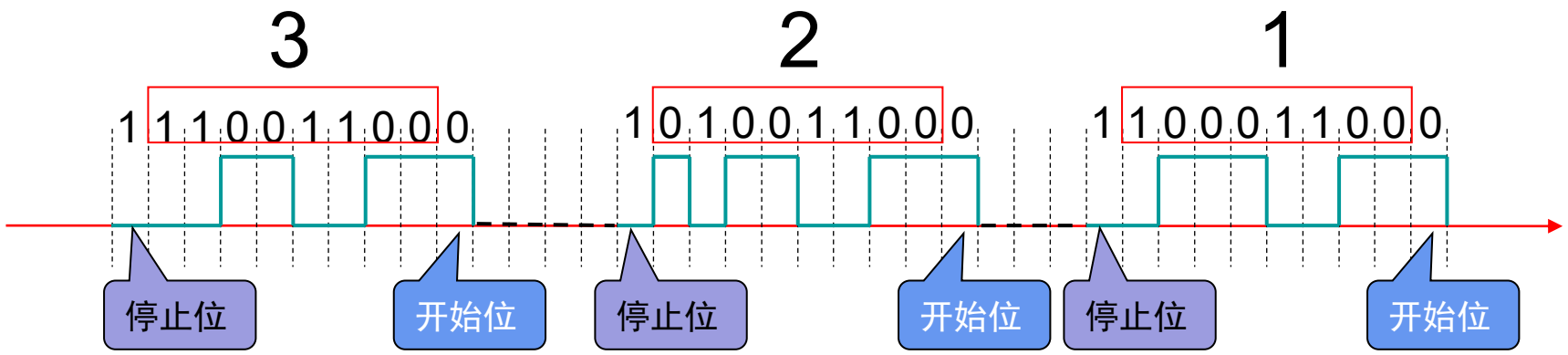
传输方式：并行传输

- 每个时钟脉冲同时发送多个比特
- 每个比特使用单独线路
- 特点：
  - 长距离时，用多条线路传输，成本高
  - 长距离时，需要使用较粗的导线，多条线路捆扎或封装到一条线缆中较为困难
  - 长距离时，接收端同步较困难
- 应用场合：
  - 用于近距离传输，例如计算机和外部设备
  - 不适合于长距离通信



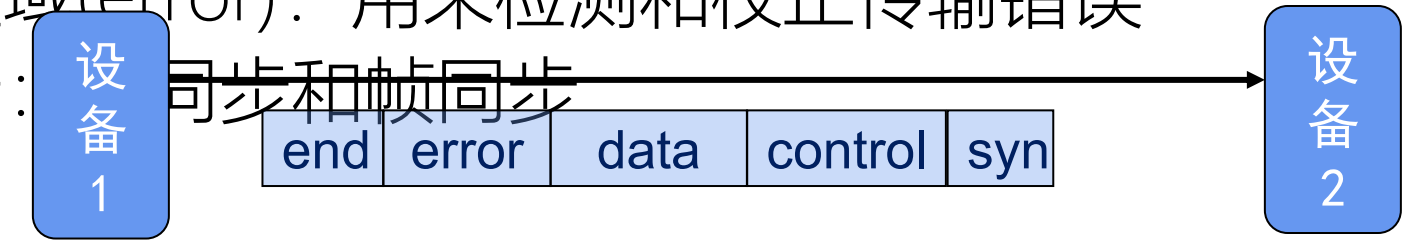
传输方式：异步传输

- 两个设备间以有限几个比特（通常是一个字符）为单位进行的数据传输
- 每个字符的传输时间固定，但两个字符间的时间间隔不固定，也不可预知
- 每几个字符需要一个开始位、停止位，传输开销大、效率低
- 异步传输用于低速设备，比如键盘和某些打印机



传输方式：同步传输

- 将字符或比特流组合成数据帧成块传送，数据帧的格式随协议而定
- 数据帧的一般组织形式：
  - 同步字符(syn)：标识一个数据帧开始
  - 帧结束标记(end)：标识一个数据帧结束
  - 控制域(control)：包含源地址、目的地址、数据长度、序列号、帧类型等
  - 数据域(data)：要发送的信息
  - 错误检查域(error)：用来检测和校正传输错误
  - 两级同步：[illegible]同步和帧同步



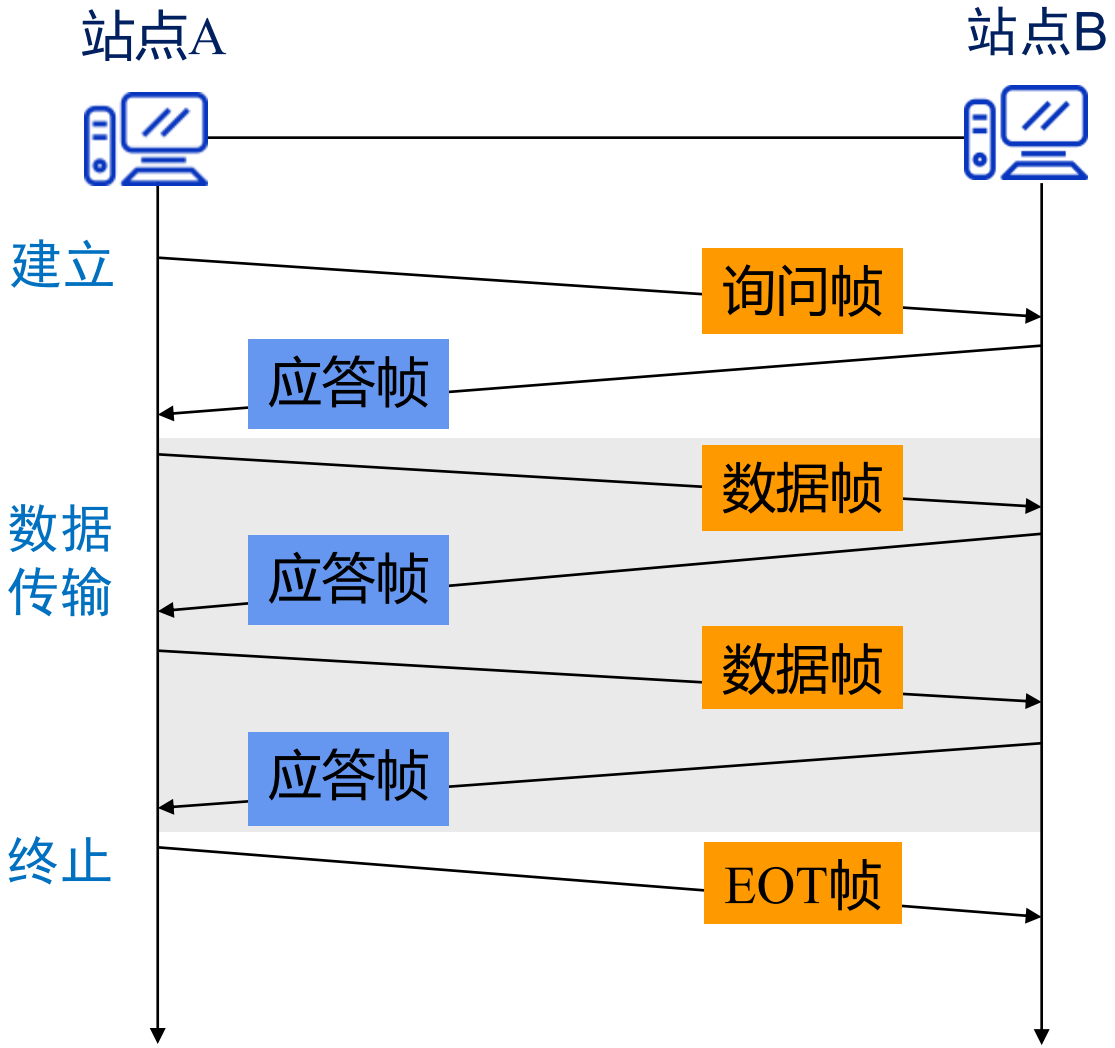
数据帧

线路规程

- 线路规程：监视链路的建立，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/应答方式（ENQ/ACK）
  - 轮询/选择方式（Poll/Select）

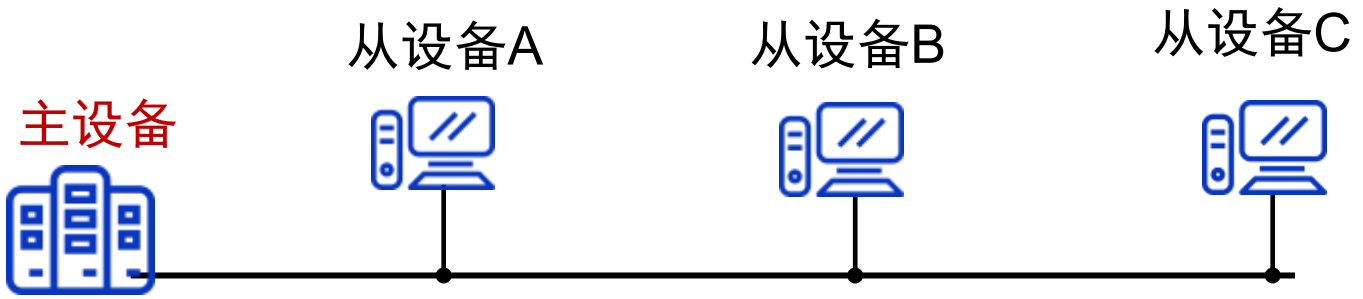
询问/应答模式

- 使用场合
  - 点对点连接系统
  - 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，回答一个应答帧（ACK）
  - 如果没有准备好接收，回答一个否定应答帧（NAK）



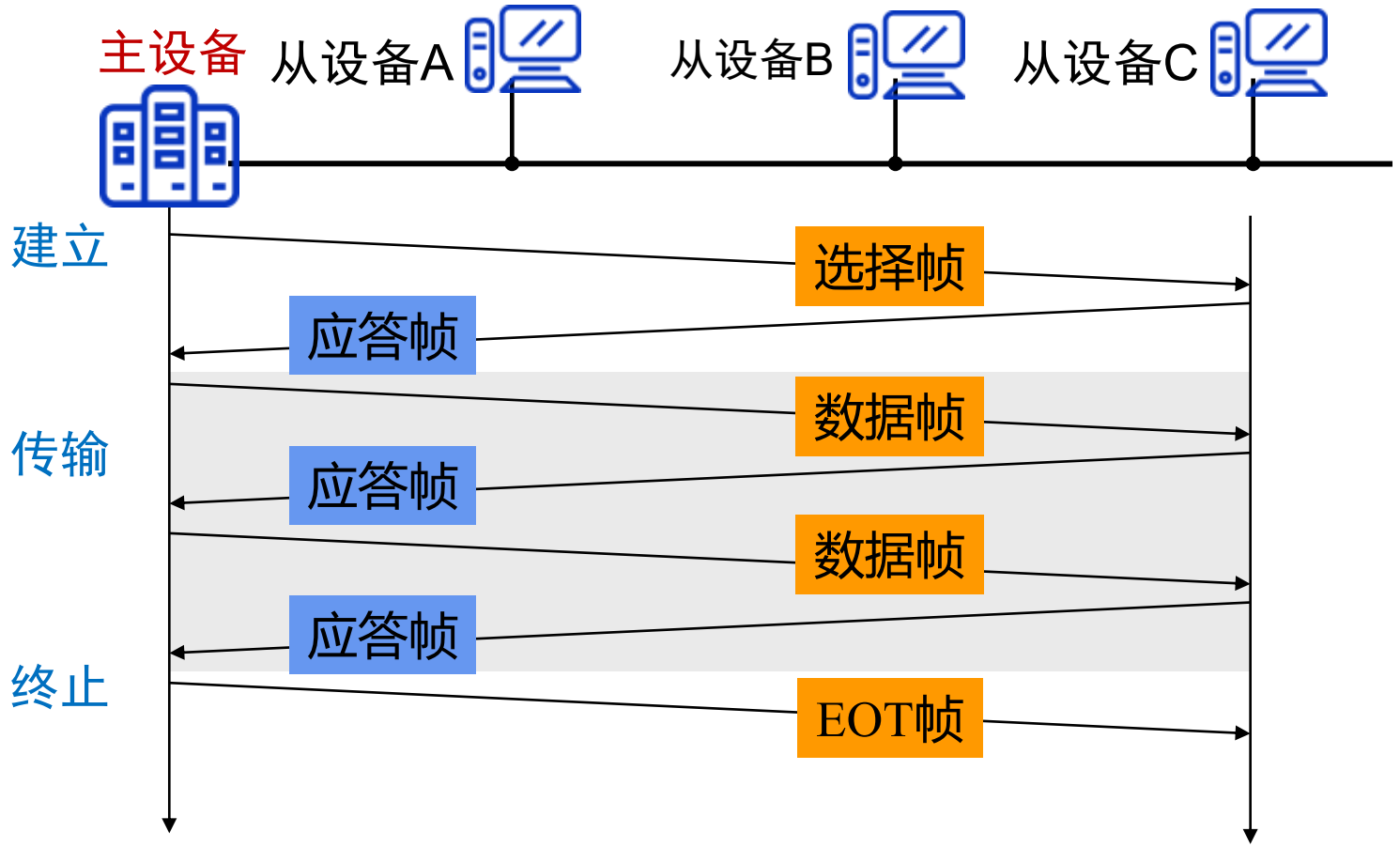
轮询/选择模式

- 使用场合：
  - 应用在多点连接系统中
  - 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
- 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



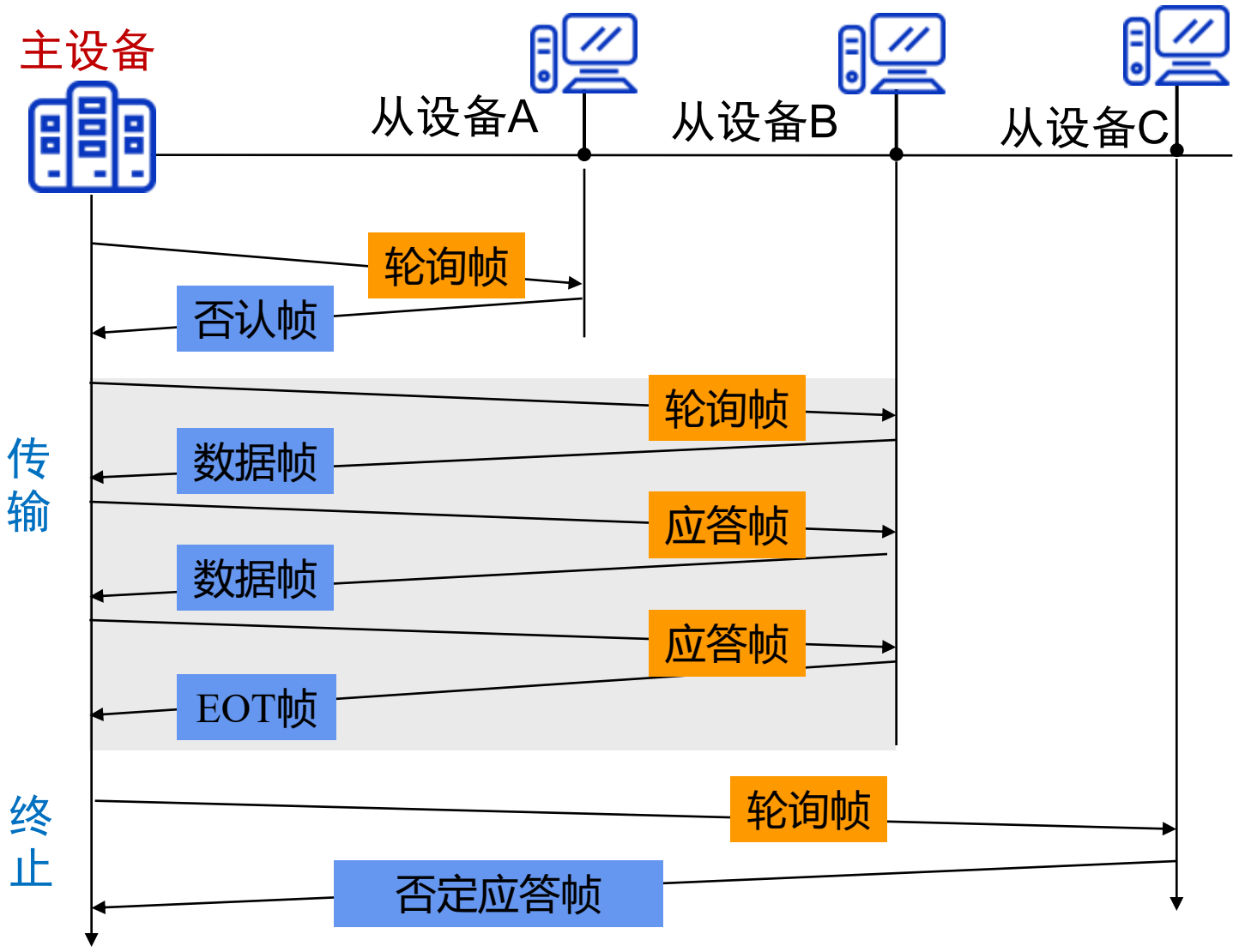
选择模式

- 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



轮询模式

- 主设备希望接收数据，逐个询问从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
- 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



数据链路层不加控制的数据收发

不加任何控制

- 发送端只管发送
- 接收端只管接收
- 不作流量控制
- 不作差错控制

需要基于以下假设条件

- 完美信道：帧不会丢失，也不会受损，信道带宽能力足够
- 始终就绪：接收方始终处于接收就绪状态，随时都能接收
- 能力超强：发送方/接收方能够生成/处理数据的能力无限

存在的问题

- 这样的假设条件在实际传输链路上几乎不存在，技术上也不现实
- 接收方来不及接收以及传输中数据帧发生错误、丢失是难免的
- 数据帧一旦发生错误或丢失，只有依赖高层发现错误、解决错误

数据链路层不加控制将带来两个问题

问题1：接收方来不及处理

- 接收方的处理速率有限，低于发送方的发送速率
- 接收方的接收缓冲区小，缓存不下突发性数据流量

解决方法

流量控制

问题2：信道上发生了差错

- 帧损坏：数据帧的比特发生差错
- 帧丢失：接收方未收到
- 帧乱序：先发后到，后发先到
- 帧重复：相同数据帧，收到多次

解决方法

差错控制

确保数据帧被正确接收

差错控制

- 差错控制主要指错误检测和重传方法
  - 错误检测：接收方能检测出数据帧出错，不将错误的数据交给网络层
  - 确　　认：接收方收到正确的数据帧后给发送方一个肯定应答
  - 自动重复请求(ARQ)：数据帧出现错误，接收方返回一个否定应答帧(NAK)，发送方重传出错的帧，这个过程叫ARQ
  - 超时重传：发送方对发送出去的数据帧启动一个定时器，超时未收到接收方的确认，将数据帧重发，防止丢失

差错控制

- 技术要点
  - 接收方利用数据帧携带的信息能检错与纠错
  - 发送方进行接收确认和超时重传
  - 接收方按序接收，防帧丢失、帧乱序、帧重复
- 重传的三种情况：
  - 帧破坏
  - 帧丢失
  - 应答帧丢失

流量控制

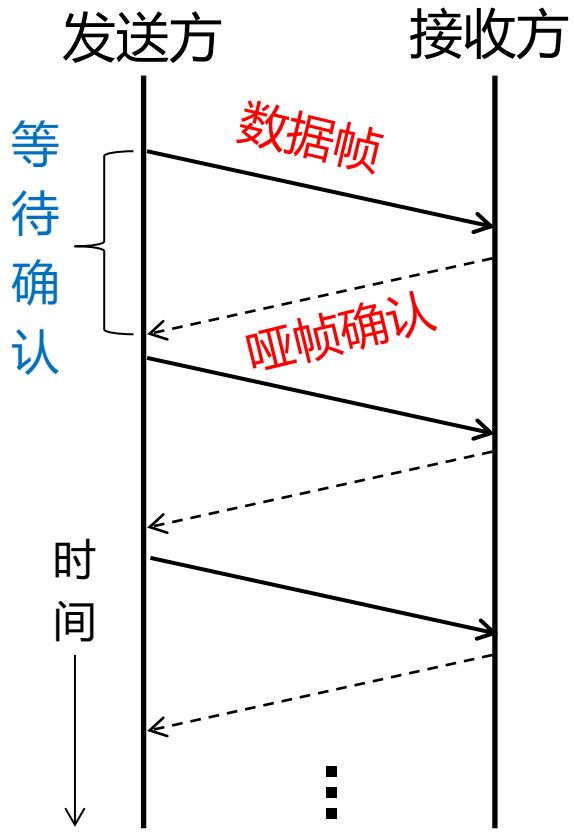
- 流量控制是一组过程，这组过程告诉发送方在收到接收方的应答之前，最多可以传送多少数据
- 流量控制的技术要点
  - 数据流不能使接收方过载
  - 接收方对数据进行确认
- 流量控制的解决方法：
  - 基于反馈 (feedback-based) 的流量控制
  - 接收方反馈，发送方来调整发送速率

流量控制和差错控制的实现技术

- 流量控制和差错控制是结合在一起实现的
- 流量控制和差错控制在技术实现上，既有关联，也有区别：
  - 共同点：
    - 都需要用到“接收确认，超时重传”的控制技术
  - 区别：
    - 差错控制多了一个“差错检测与纠错”环节，对接收到的数据帧先要做差错检测与纠错，是差错控制的先导环节
- 流量控制和差错控制的技术有两种：
  - 停止等待协议
  - 滑动窗口协议

停止等待协议（无错信道）

- 假设
  - 通信信道不会出错：既不出错也不丢失
  - 发送方以高于接收方能处理到达帧的速度发送帧，会导致接收方来不及接收
- 停止等待协议（stop-and-wait）
  - 发送方发送一帧后暂停，等待确认到达后发送下一帧
  - 接收方完成接收后，回复确认
  - 应答帧的内容是不重要的：哑帧

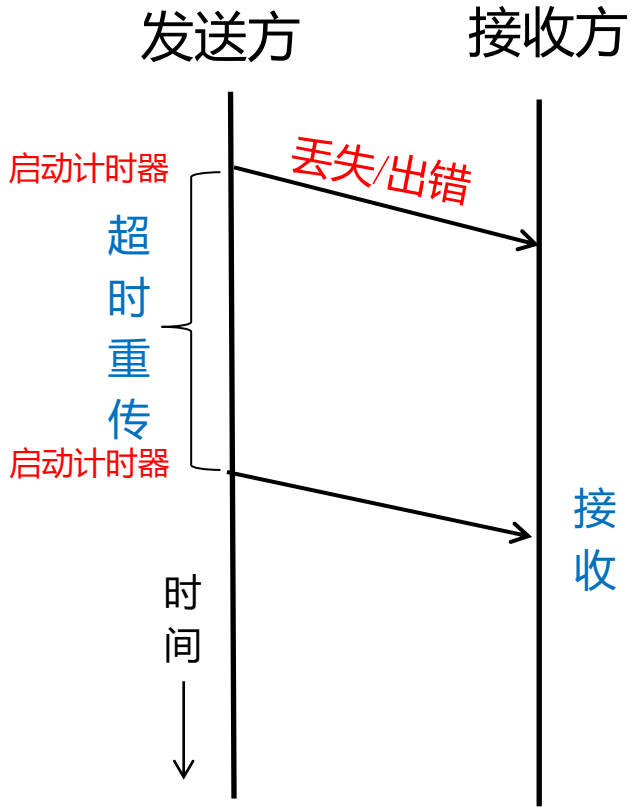


- 完成一帧发送后，等待确认到达
- 确认到达后，发送下一帧

- 完成一帧接收后，交给物理层一个哑帧
- 作为成功接收上一帧的确认

停止等待协议（有错信道）

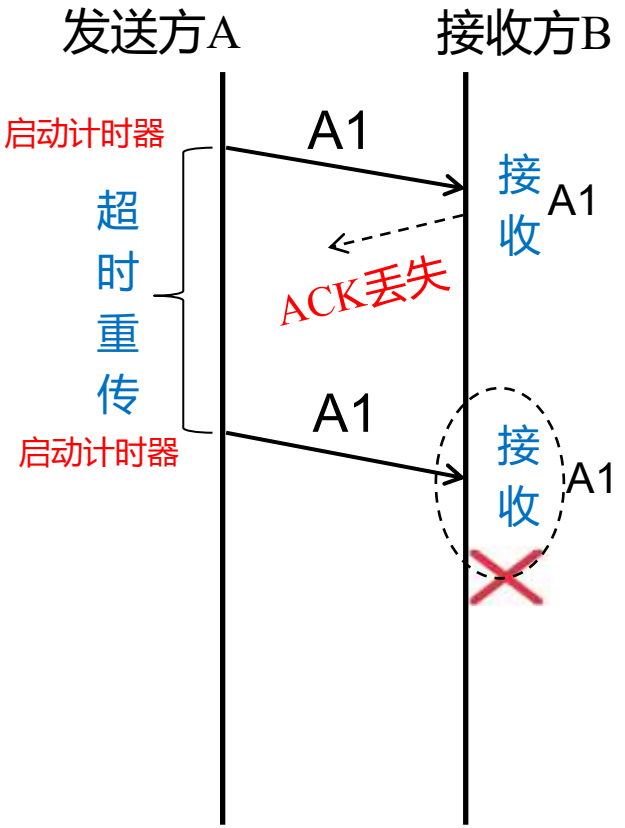
- 假设
  - 接收方接收能力有限
  - 通信信道可能会出错，导致帧损坏或丢失
- 一个简单的解决方案
  - 发送方使用计时器对发出去的数据帧计时
  - 如果超时，认为接收方没有收到或帧出错
  - 发送方超时重传该帧



➢ 问题：会不会接收方已经正确接收，发送方误认为没有收到，超时重传，导致帧数据重复？

停止等待协议（有错信道）

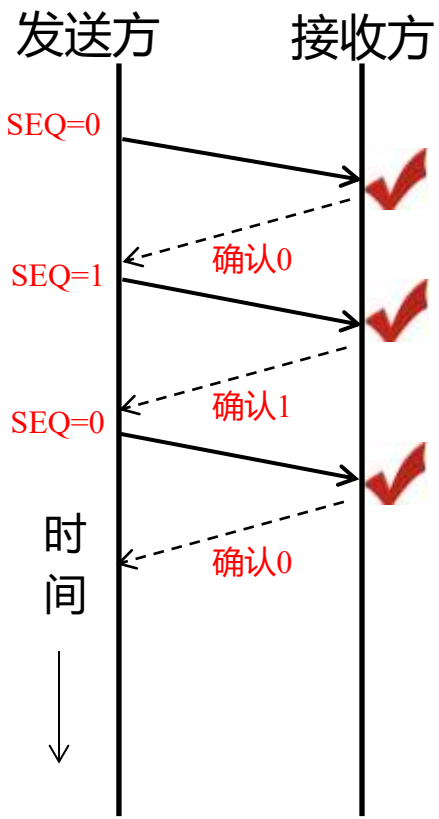
- 接收方重复收到同一数据帧的两种场景：
- 场景一：ACK丢失
  - A发送帧A1
  - B收到了A1
  - B生成确认ACK
  - ACK在传输中丢失
  - A超时，重传A1
  - B收到A1的另一个副本（并把它交给网络层）
- 场景二：超时后，接收方才收到数据帧
  - 超时时间定义过小，或者数据帧传输时间过长
  - 发送方超时重传后，接收方收到了第一次发的数据帧
  - 发送方超时重传，接收方会收到该数据帧的一个副本



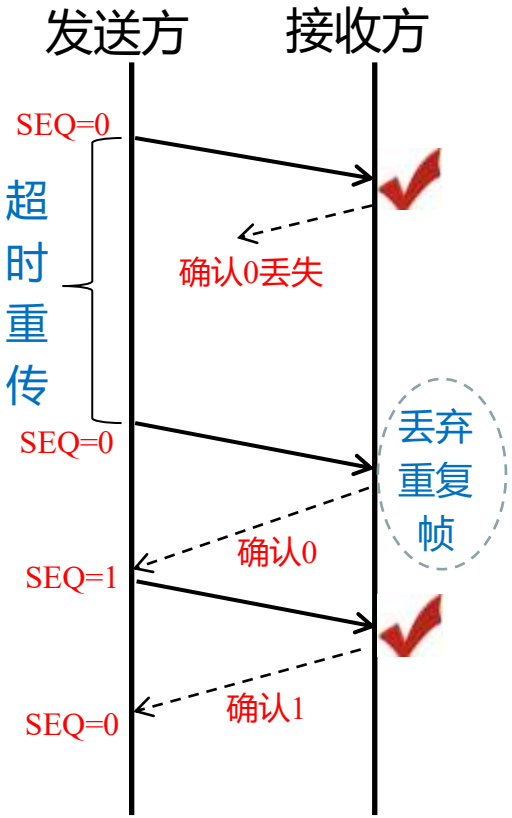
➢ **问题：如何解决接收方重复接收同一帧数据？**

停止等待协议（有错信道）

- 解决数据帧重复接收的方法
  - 序号（SEQ：sequence number）
    - 发送方在帧中携带序号
    - 接收方检查帧序号，确认是否是新帧
    - 序号需要明确的当前帧和它的直接后续帧
    - 1 bit序号(0或1)就足以满足要求
  - 自动重复请求，或带有重传的肯定确认
    - ARQ(Automatic Repeat reQuest)
    - PAR(Positive Acknowledgement with Retransmission)



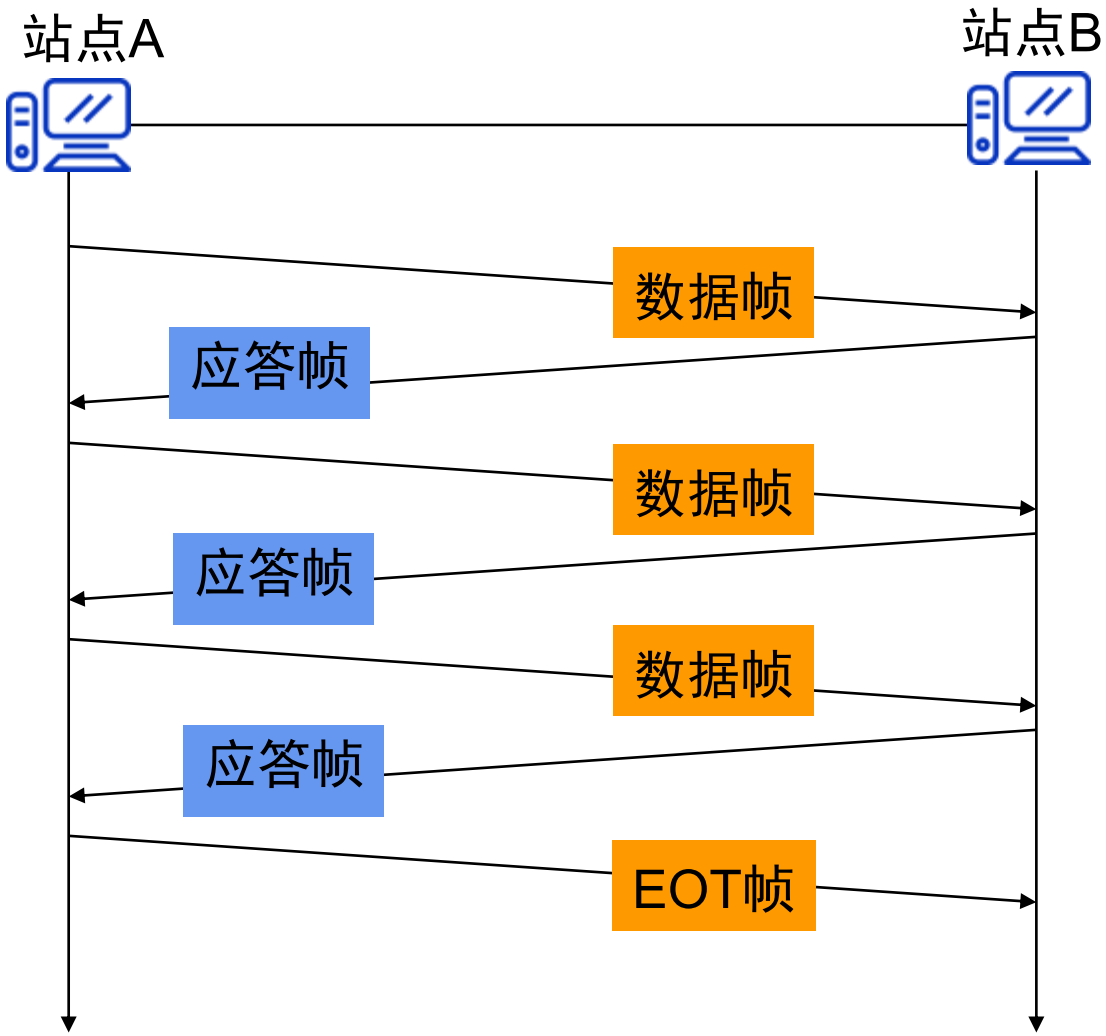
发送方：帧带序号并检查确认序号
- 初始化帧序号0，发送帧
- 等待：肯定确认/否定确认/超时
- 肯定确认：发送下一帧
- 超时/否定确认：重传



接收方：检查帧序号，防止重复帧
- 初始化期待0号帧，等待帧达到
- 正确帧（序号和内容都正确）：交给网络层，并发送该帧确认
- 错误帧：发送上一个成功接收帧的确认

基于停止等待协议实现流量控制

- 停止等待协议
  - 发送方每发送一帧后就等待应答，只有收到一个应答后，才发送下一个帧，直到发送方发送一个传输结束帧
  - 流量控制：只有收到应答后才发送下一帧，发送速率得到控制
  - 协议的优缺点
    - 优点：协议简单
    - 缺点：传输效率低，线路带宽利用率也很低

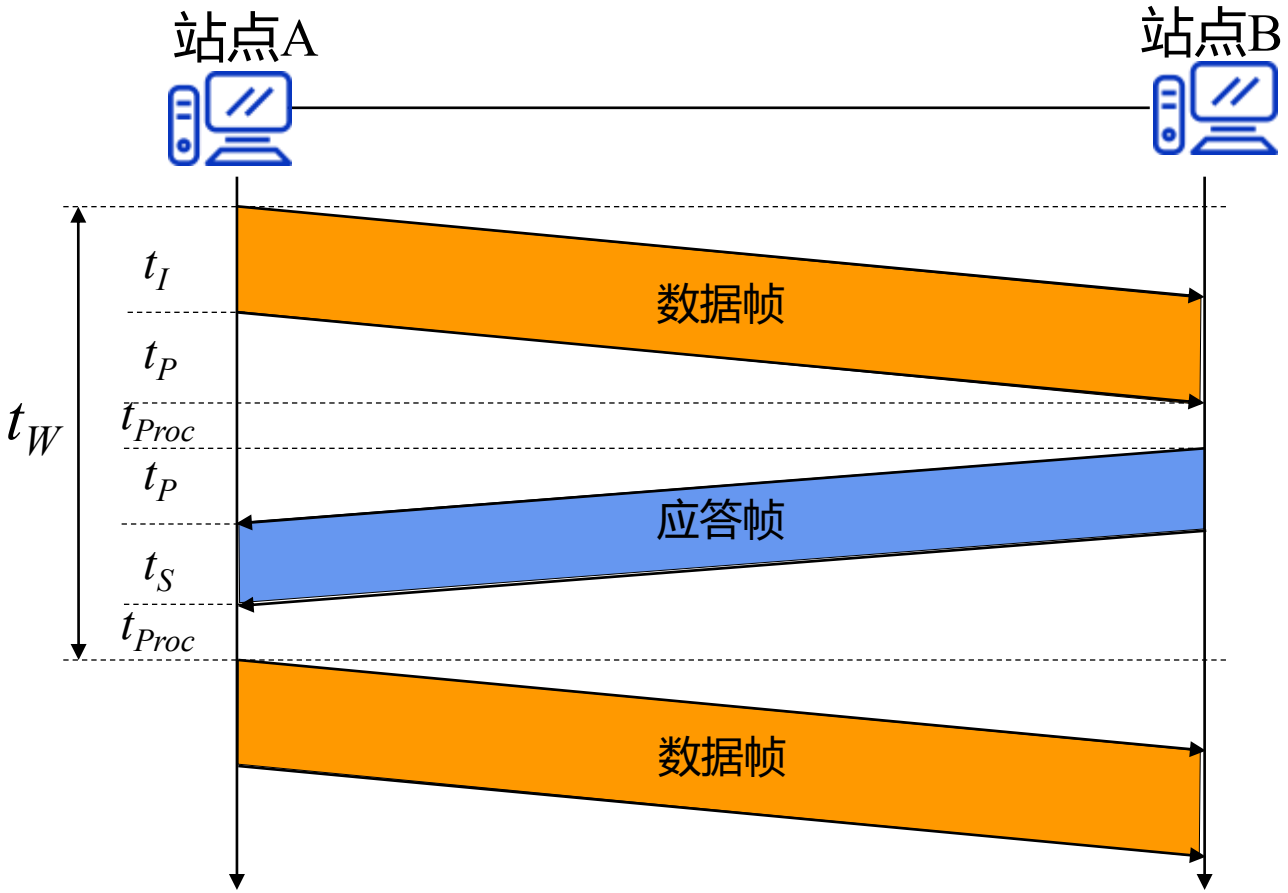


停止等待协议性能评估

- 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：发送端发送数据帧时间
    （帧长/数据传输速率）
  - t_S：接收端发送应答帧时间
    （应答帧长/数据传输速率）
  - t_P：信号传播延时
    （线路距离/信号传播速率）
  - t_{Proc}：节点处理时间



停止等待协议性能评估（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
- 无差错情形时发送信道的利用率P

$$P = t_I / t_W$$

t_I：发送端发送数据帧的时间

t_W：发送一帧的总时间

- 无差错情形时的有效数据传输率S

$$S = N / t_W$$

N：一帧数据帧的有效数据比特数

停止等待协议性能评估（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）

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

- 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

$$M = 1/t_I$$

- 系统传输效率：最大吞吐量/极限吞吐量

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

计算实例

假设条件

C = 数据流发送速率（10Mbps或10bit/ μs ）

S = 数据流在线路上传输速度（200m/ μs ）

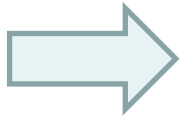
D = 发送方与接收方的距离（200m）

t_{Proc} = 生成一帧的时间（1 μs ）

L_f = 一帧数据的比特数（200bit）

N = 一帧数据的有效数据比特数（160bit）

L_S = 一个应答帧的比特数（40bit）



计算结果

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

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

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

$t_{Proc} = 1(\mu s)$

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

$t_W = 20 + 2\times1 + 2\times1 + 4 = 28(\mu s)$

发送信道利用率：

$$P = t_I/t_W = 20/28 = 71.4\%$$

有效数据传送速率：

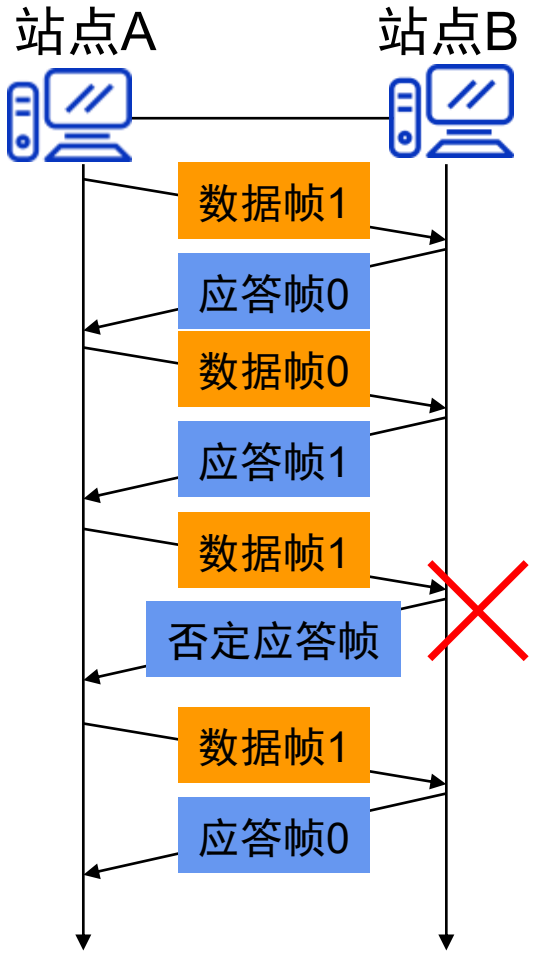
$$S = N/t_W$$

$$= 160/28 = 5.7\text{Mbps}$$

基于停止等待协议实现差错控制

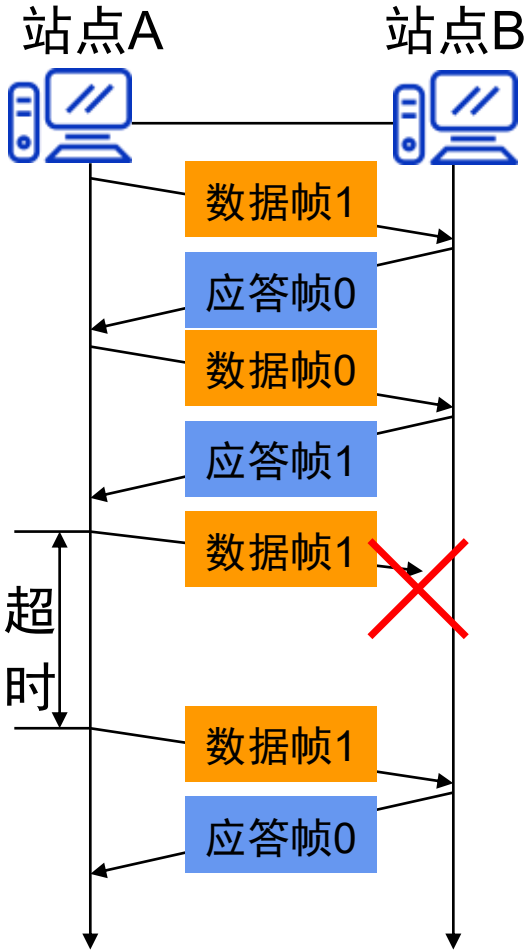
- 为实现差错控制，在停止等待协议中采用自动重复请求（停等ARQ）
- 停等ARQ处理如下三种错误情况：
  - 帧破坏
  - 帧丢失
  - 应答帧丢失

用 ARQ 处理三种差错情形



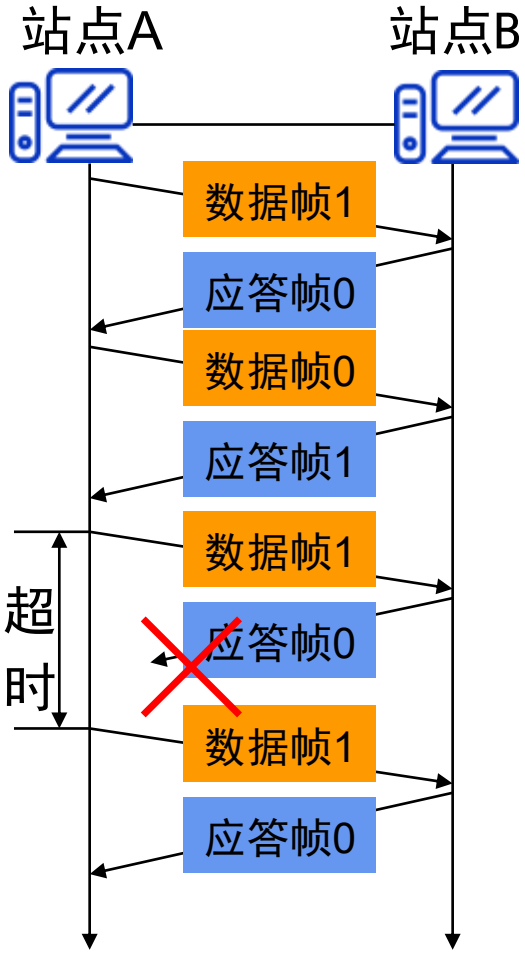
帧破坏

收到否定应答帧后重传



帧丢失

超时重传



应答帧丢失

超时重传

技术要点

- 发送端要保留数据帧的备份
- 发送的数据帧序号必须要0和1交替
  - 如果接收方收到了两个序号相同的相邻数据帧，说明收到了一个重复帧，丢弃
- 应答帧是对接收帧的确认
  - 应答帧序号和接收帧序号相反，表示当前这一帧已收到，期望接收下一帧
- 否定应答帧(NAK)，通知发送方重新发送最近的一帧
- 定时器，判断数据帧在约定时间内没有收到确认，需要重传

停止等待协议的性能问题

- 信道利用率低
  - 假如将链路看成是一根管道，帧是管道中流动的数据，那么在传播延迟较长的信道上，由于：
    - 信道上数据传输速率低
    - 线路长
    - 接收方接收过程时间长，确认回应迟
    - .....
  - 停止等待协议无法使数据充满管道，因而信道利用率很低
- 解决办法
  - 流水线协议或管道协议：允许发送方在没收到确认前连续发送多个帧

滑动窗口协议——基本思想

- 窗口是发送方和接收方存放数据帧的缓冲区
- 发送方在收到应答前可以连续发送若干帧（类似批发）
- 发送方窗口由两部分组成：
  - 一部分是用于存放已经发送但未收到应答的数据帧
  - 另一部分是在收到应答帧之前还可以发送的数据帧，直到窗口满
- 接收方窗口由两部分组成：
  - 一部分用于存放已经接收但未给应答的数据帧
  - 另一部分，在未发送应答帧的情况下可以继续接收数据帧，直到窗口满

滑动窗口协议——基本思想

- 目的
  - 对可以连续发送的最多帧数（已发出但未确认的帧）作限制
- 窗口大小问题
  - 为了应对突发性流量，发送窗口和接收窗口均可以设置大一点
  - 窗口大一点，利于区分帧序号循环产生的新帧和重传的帧，避免重复接收

➢ 问题：发送方一次可以发送多少个帧？帧出错或丢失怎么处理？

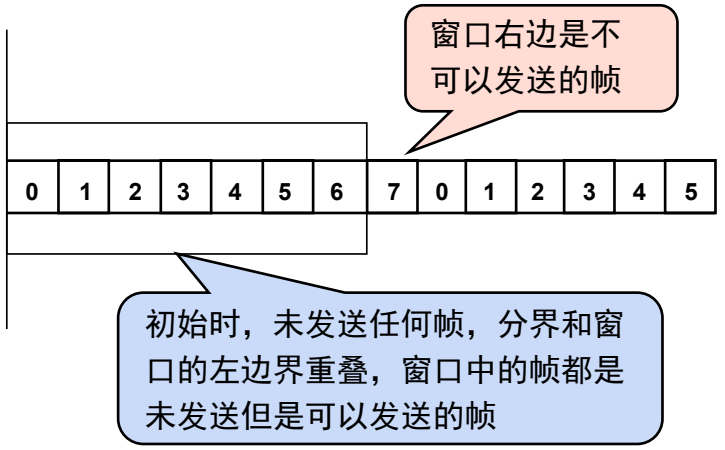
滑动窗口协议——基本思想

- 序号使用
  - 循环重复使用有限的帧序号
- 流量控制：接收窗口驱动发送窗口的转动
  - 发送窗口：在收到确认之前，发送方可以连续发送的数据帧数
  - 接收窗口：接收方可以连续接收的最多数据帧数
- 累积确认：接收方使用一个ACK帧来对多个数据帧的接收进行确认

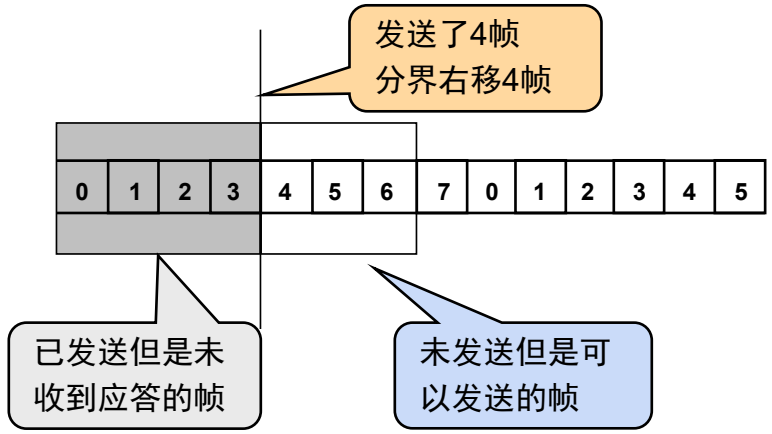
滑动窗口协议——基本思想

- 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x是接收方希望收到的下一帧的编号

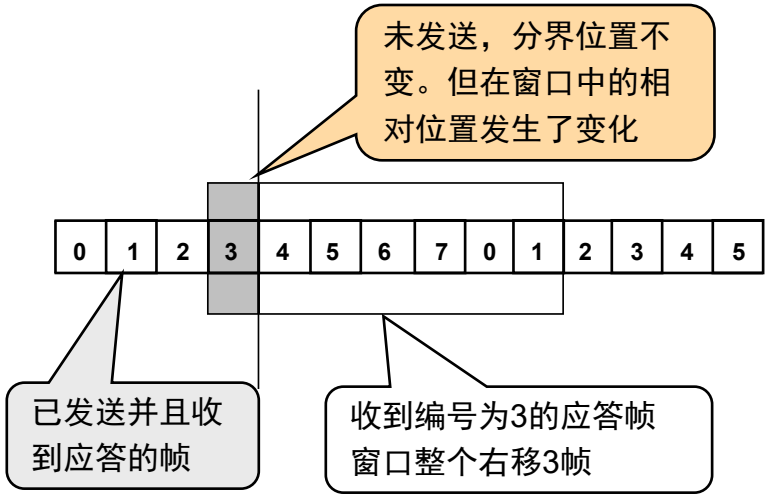
发送方窗口


窗口右边是不
可以发送的帧
0 1 2 3 4 5 6 7 0 1 2 3 4 5
初始时，未发送任何帧，分界和窗口的左边界重叠，窗口中的帧都是未发送但是可以发送的帧

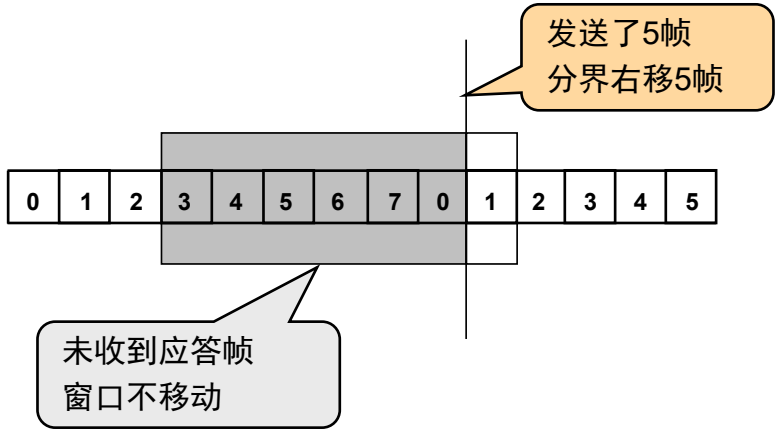
(a)


发送了4帧
分界右移4帧
0 1 2 3 4 5 6 7 0 1 2 3 4 5
已发送但是未
收到应答的帧
未发送但是可
以发送的帧

(b)

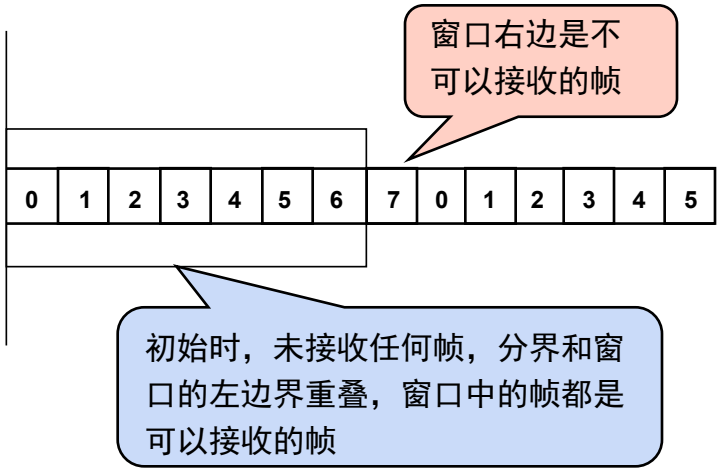

未发送，分界位置不
变。但在窗口中的相
对位置发生了变化
0 1 2 3 4 5 6 7 0 1 2 3 4 5
已发送并且收
到应答的帧
收到编号为3的应答帧
窗口整个右移3帧

(c)

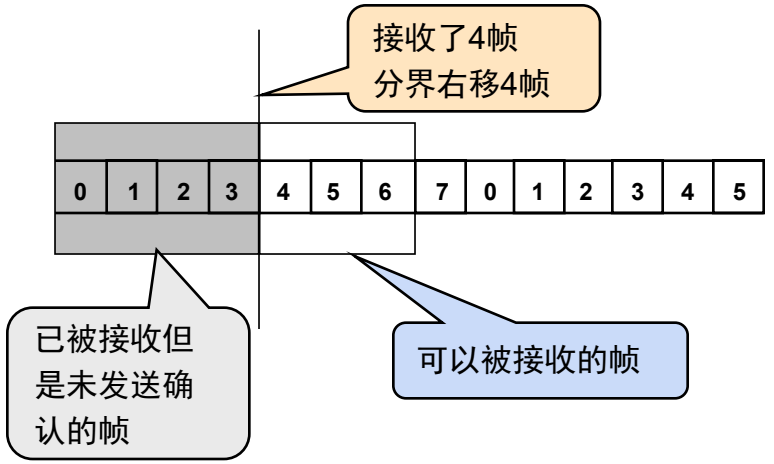

发送了5帧
分界右移5帧
0 1 2 3 4 5 6 7 0 1 2 3 4 5
未收到应答帧
窗口不移动

(d)

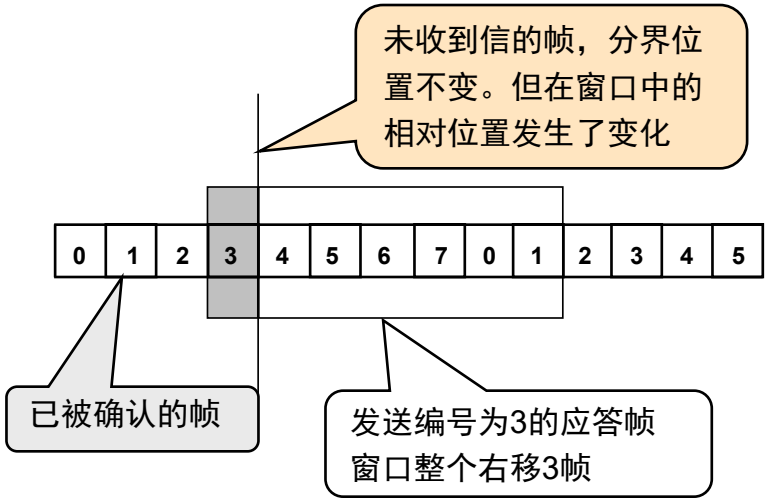
接收方窗口


窗口右边是不
可以接收的帧
0 1 2 3 4 5 6 7 0 1 2 3 4 5
初始时，未接收任何帧，分界和窗口的左边界重叠，窗口中的帧都是可以接收的帧

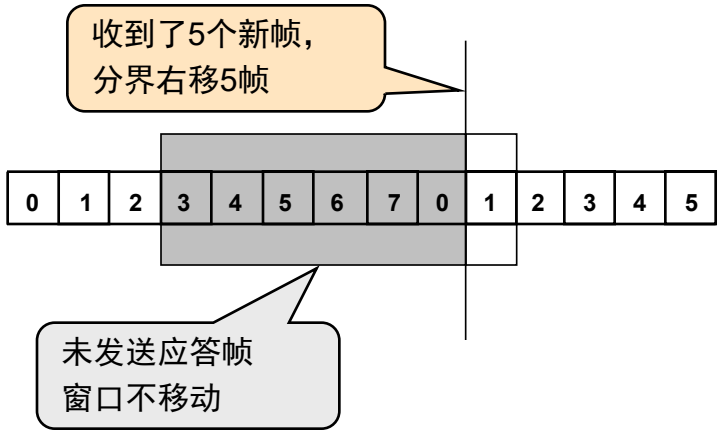
(a)


接收了4帧
分界右移4帧
0 1 2 3 4 5 6 7 0 1 2 3 4 5
已被接收但
是未发送确
认的帧
可以被接收的帧

(b)


未收到信的帧，分界位置不变。但在窗口中的相对位置发生了变化
0 1 2 3 4 5 6 7 0 1 2 3 4 5
已被确认的帧
发送编号为3的应答帧
窗口整个右移3帧

(c)


收到了5个新帧，
分界右移5帧
0 1 2 3 4 5 6 7 0 1 2 3 4 5
未发送应答帧
窗口不移动

(d)

滑动窗口协议中的差错控制

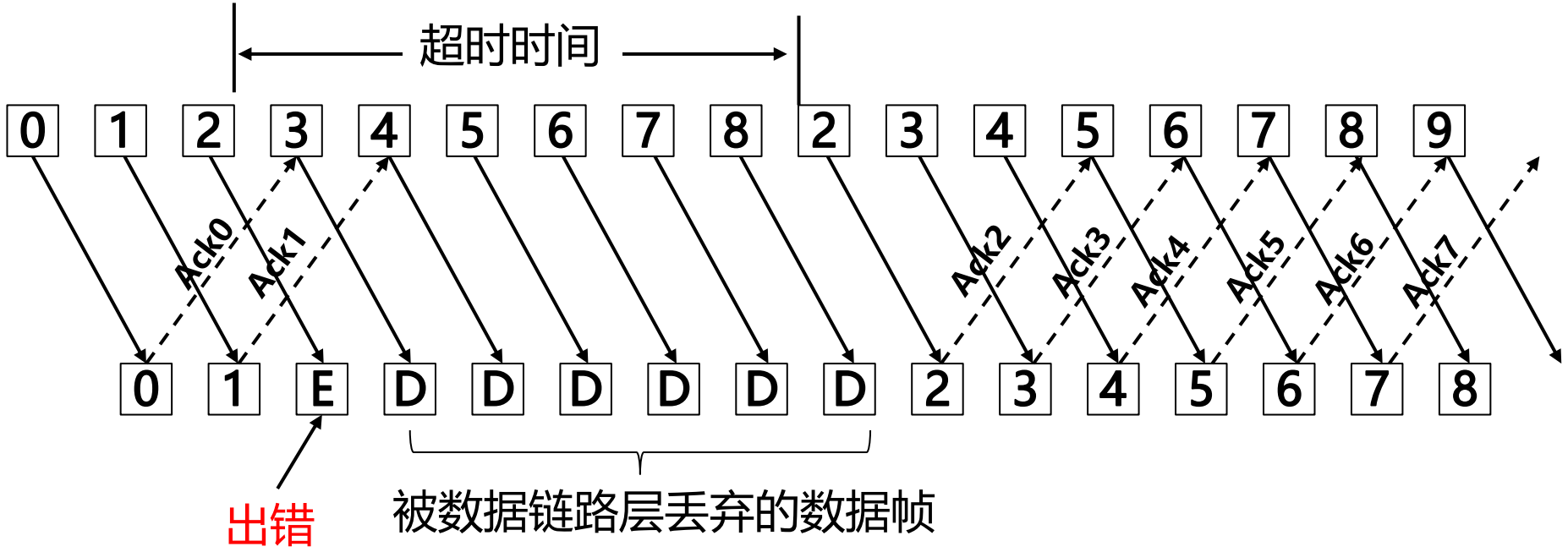
- 有两种实现自动重复请求(ARQ)技术:
  - 回退N自动重复请求(Go-back-N，GBN)
  - 选择拒绝自动重复请求(Select-Rej，SR)
- 要求:
  - 发送站要保留数据帧的备份
  - 除应答帧外，接收方可以发送否定应答帧（NAK)，告诉发送方重新发送损坏帧
  - 定时器，判断数据帧在传输中丢失

回退N协议—设计思想

- 出错全部重传
  - 接收方收到一个出错或乱序帧时，丢弃所有后续帧，并且不为这些帧发送确认
  - 发送方出现超时后，重传所有未被确认的帧
- 适用场景
  - 该策略对应接收窗口为1的情况，即只能按顺序接收帧
- 优缺点
  - 优点：连续发送提高了信道利用率
  - 缺点：按序接收，出错后即便有正确帧到达也丢弃重传

回退N协议—工作原理

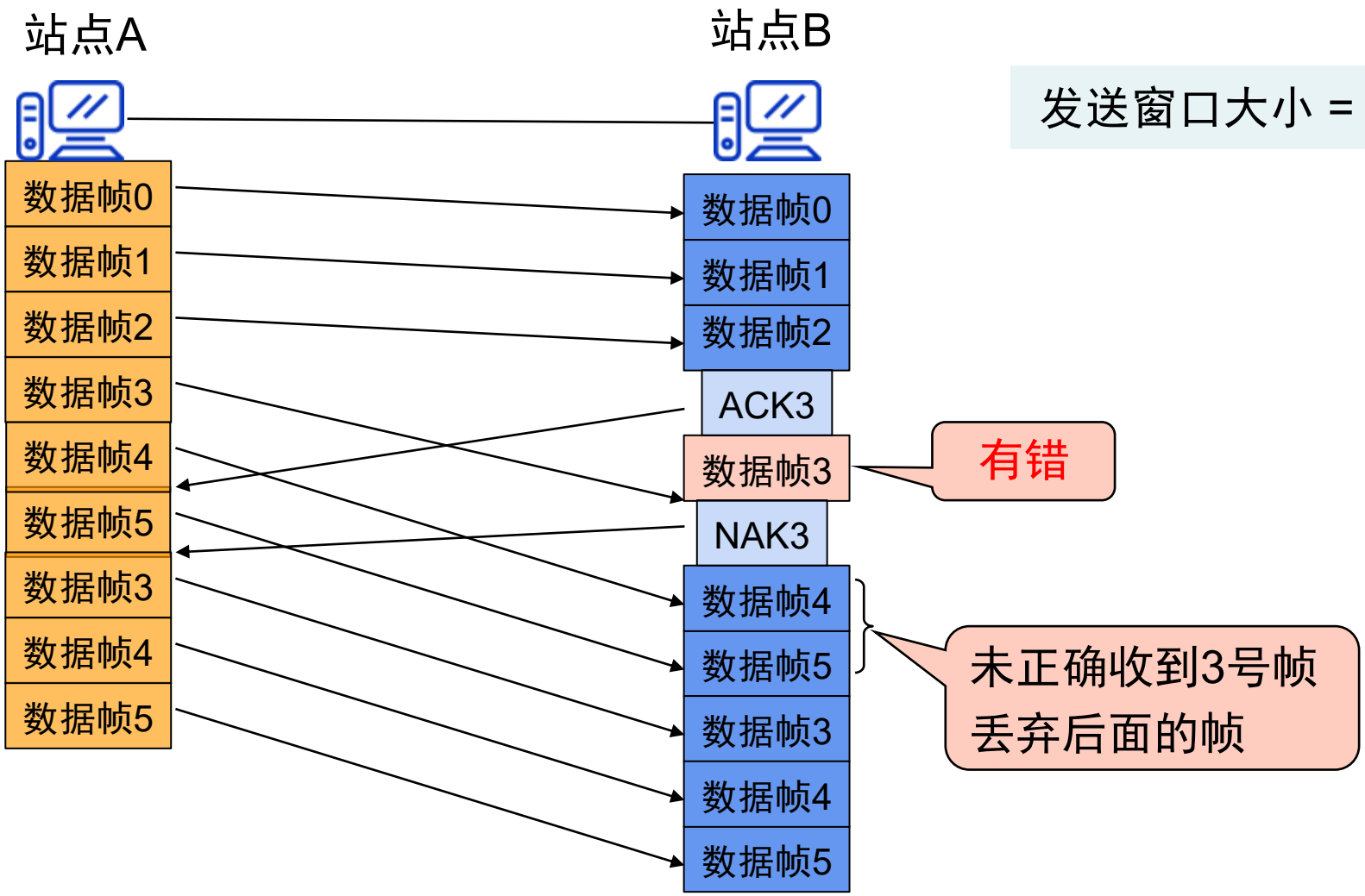
- 基本原理
  - 发送方发送了N个帧后，若发现该N帧的前一帧在超时后仍未收到确认，则判断该帧出错或丢失，发送方重传出错帧及其后的N帧
- 窗口大小
  - 若帧编号为n位，接收窗口为1，发送窗口小于等于 n - 1



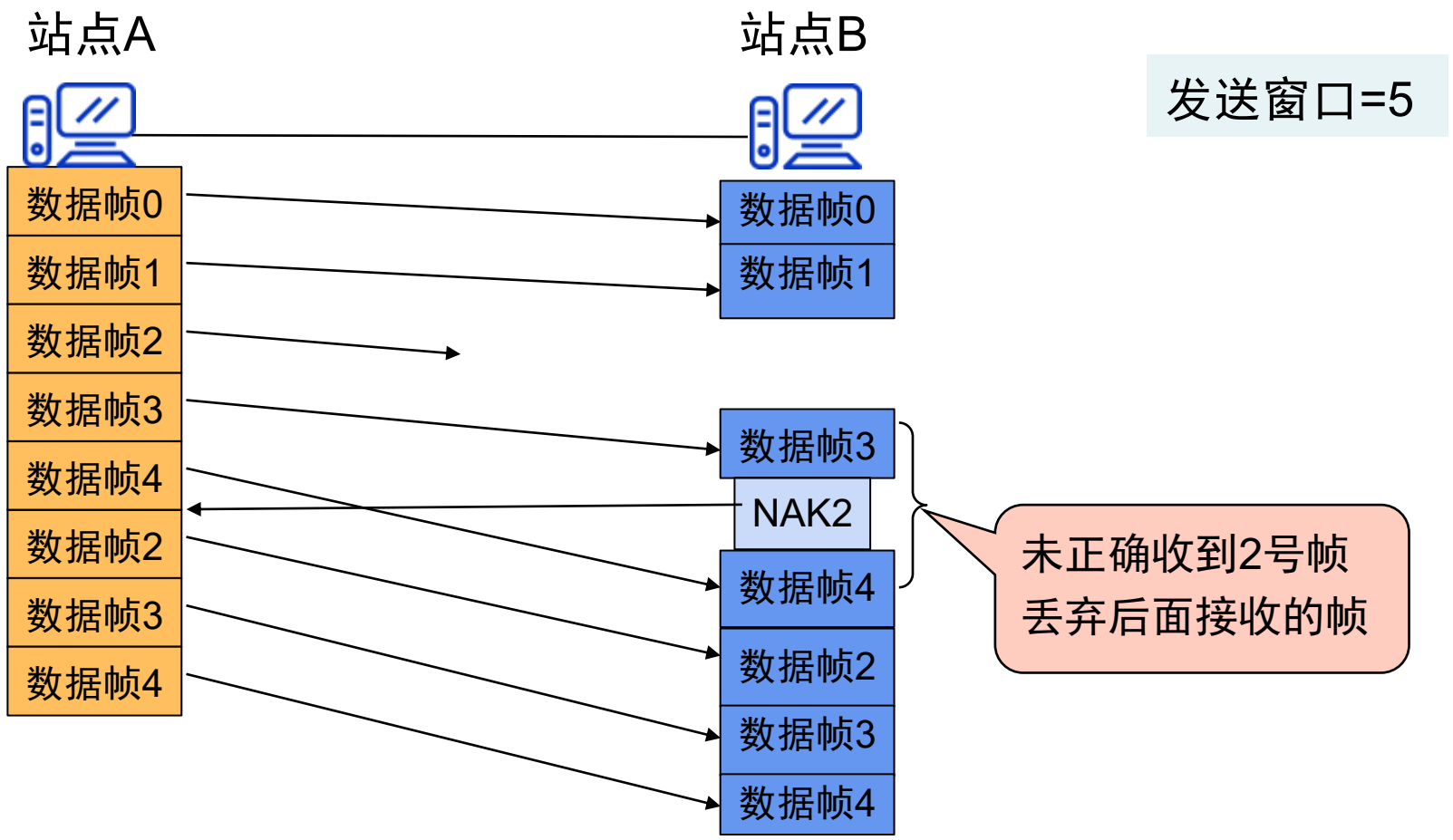
回退N协议— 差错处理

- 差错情况有三种：
  - 帧破坏
  - 数据帧丢失
  - 应答帧丢失

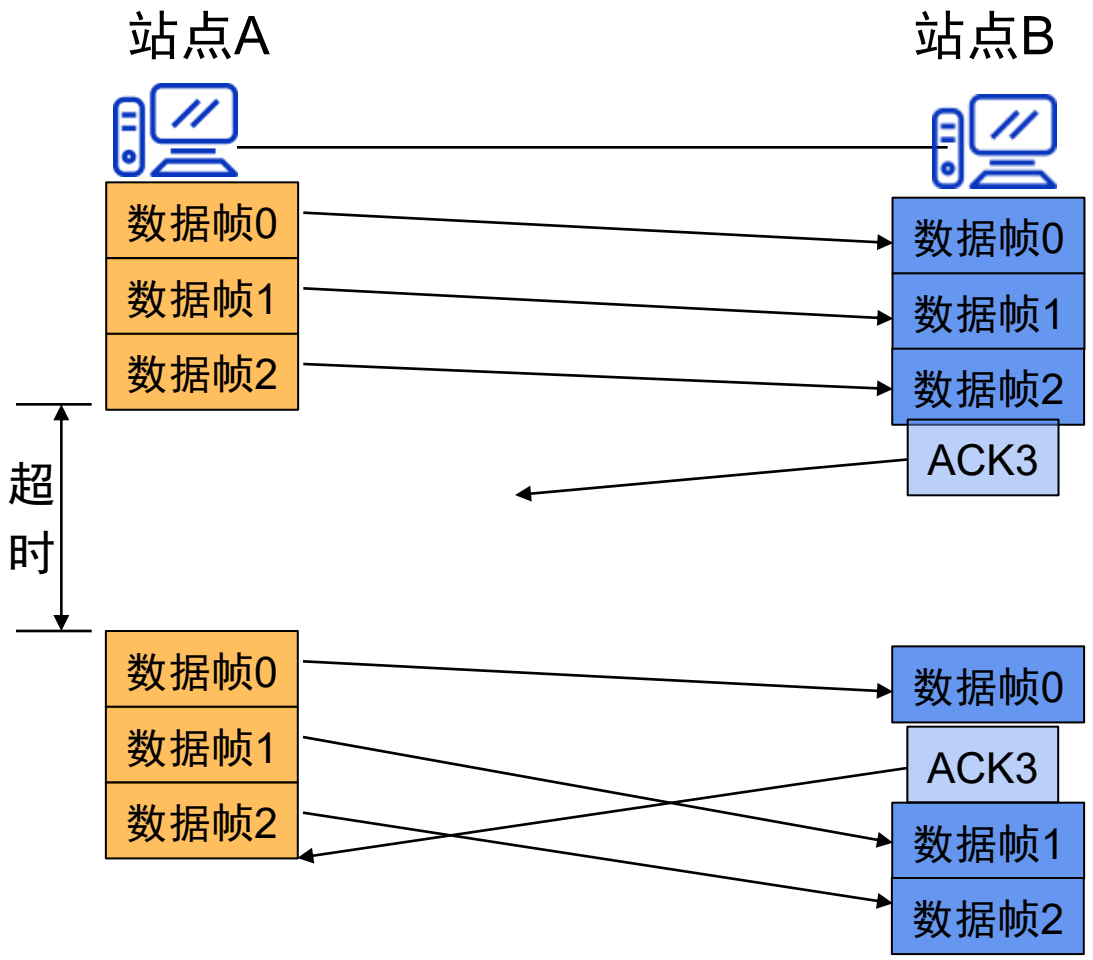
回退N协议— 帧破坏


站点A
站点B
发送窗口大小 = 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
有错
NAK3
数据帧4
数据帧5
数据帧3
数据帧4
数据帧5
未正确收到3号帧
丢弃后面的帧

回退N协议— 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

回退N协议— 应答帧丢失

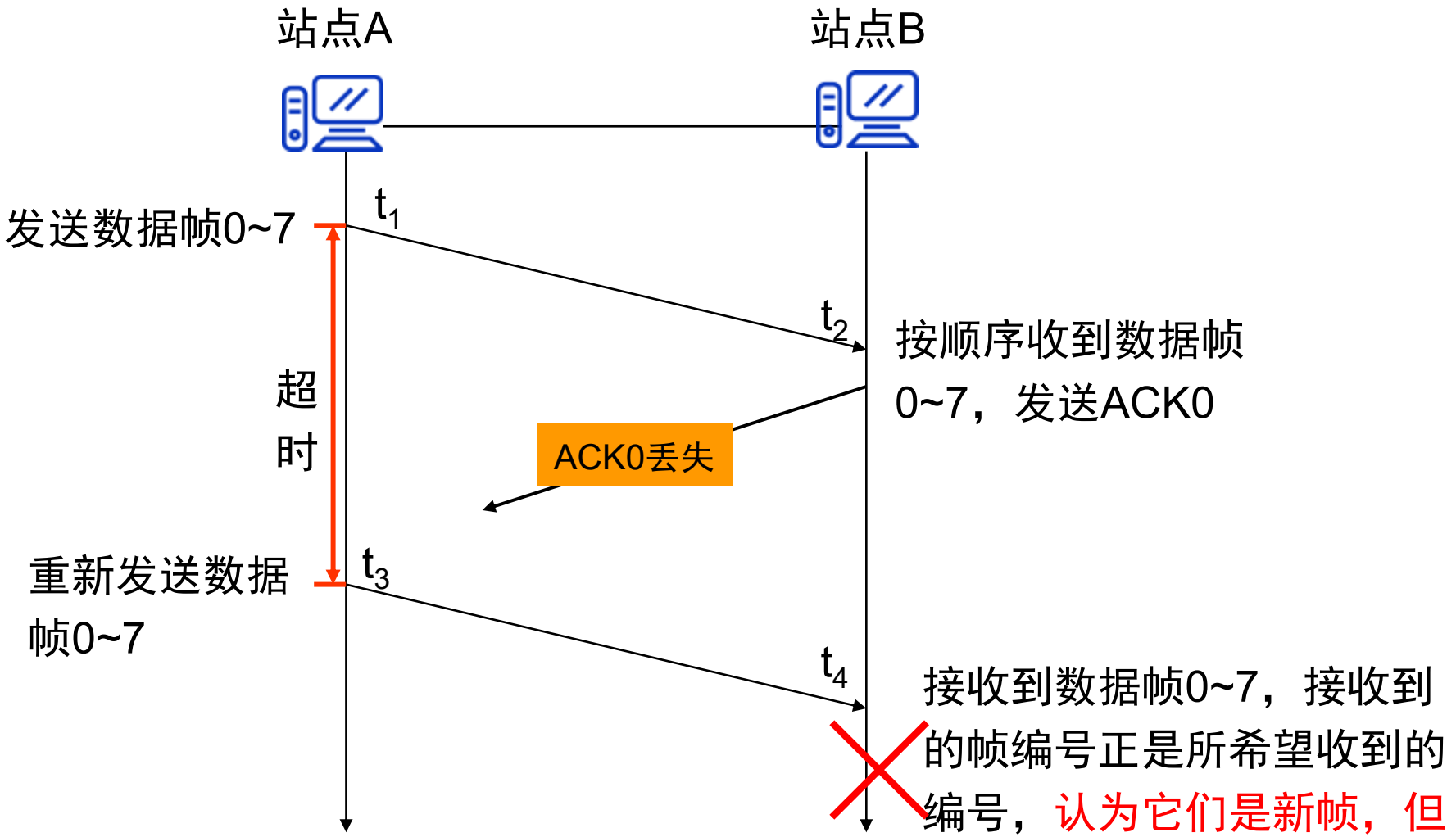


- 定时器启动：当发送窗口满，或者没有数据要发送了

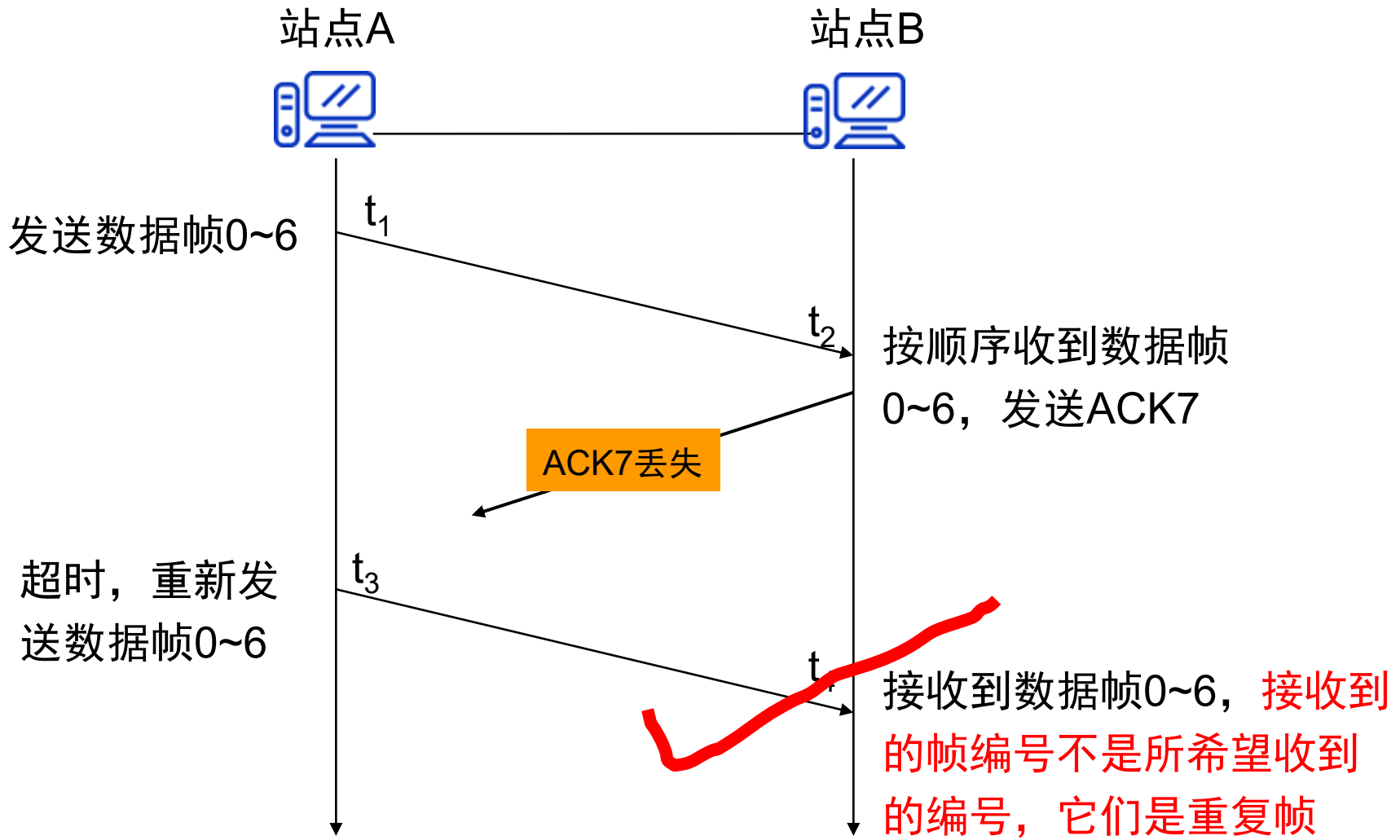
回退N协议—窗口大小与编号范围的关系

- 如果帧的编号范围是0 ~ n-1，则窗口尺寸为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

窗口尺寸等于n时，应答帧丢失时协议失败



窗口尺寸等于 n-1 时，协议成功



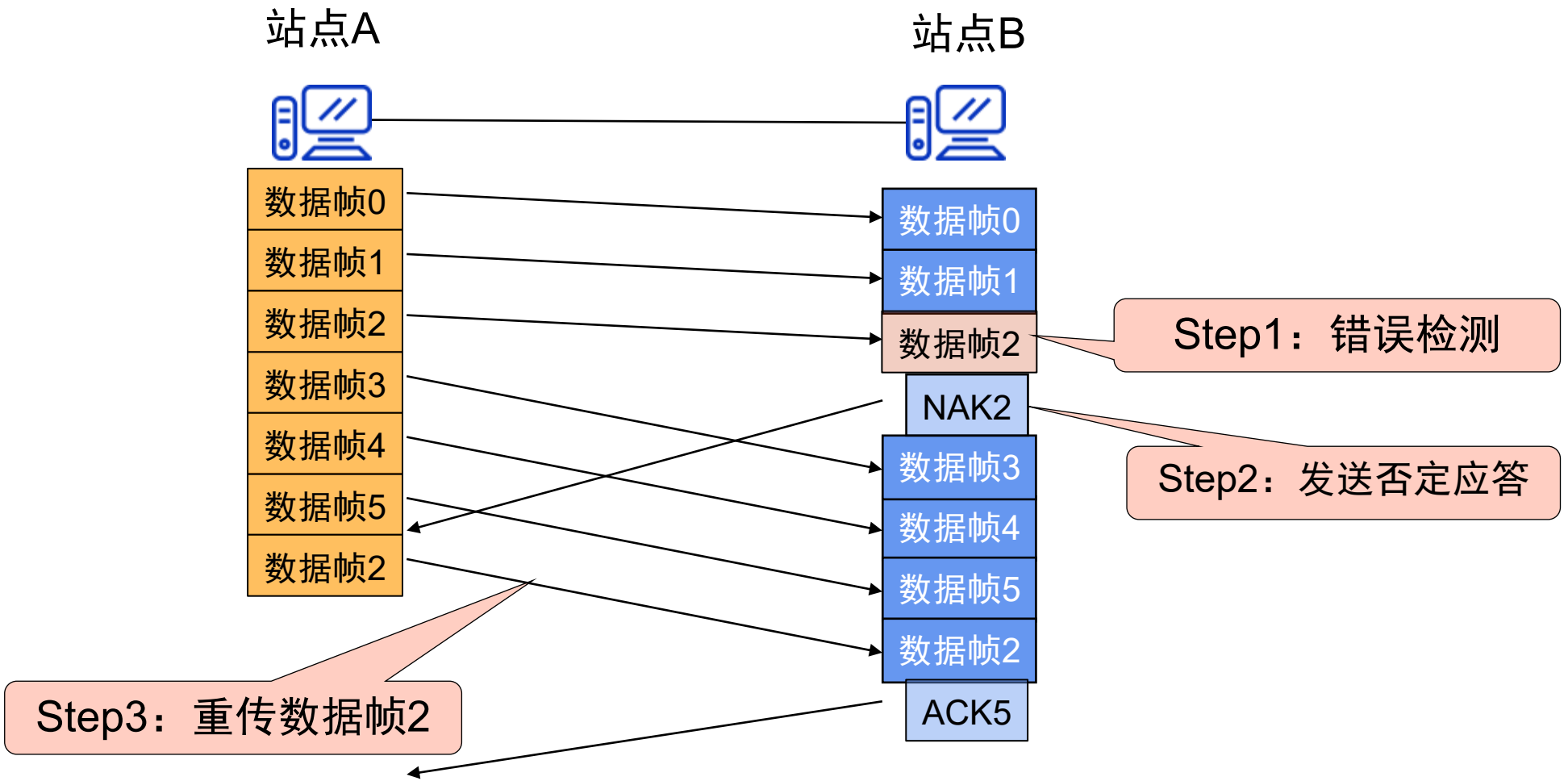
选择拒绝自动重复请求—设计思想

- 若发送方发出连续的若干帧后，收到对其中某一帧的否认帧，或某一帧的定时器超时， 则只重传该出错帧或计时器超时的数据帧
- 适用场景
  - 接收窗口大于1的情况，即暂存接收窗口中序号在出错帧之后的数据帧
  - 接收设备须具有排序功能
  - 发送设备须具有查找机制
- 优缺点
  - 优点：避免重传已正确发送的帧
  - 缺点：在接收端需要占用一定容量的缓存

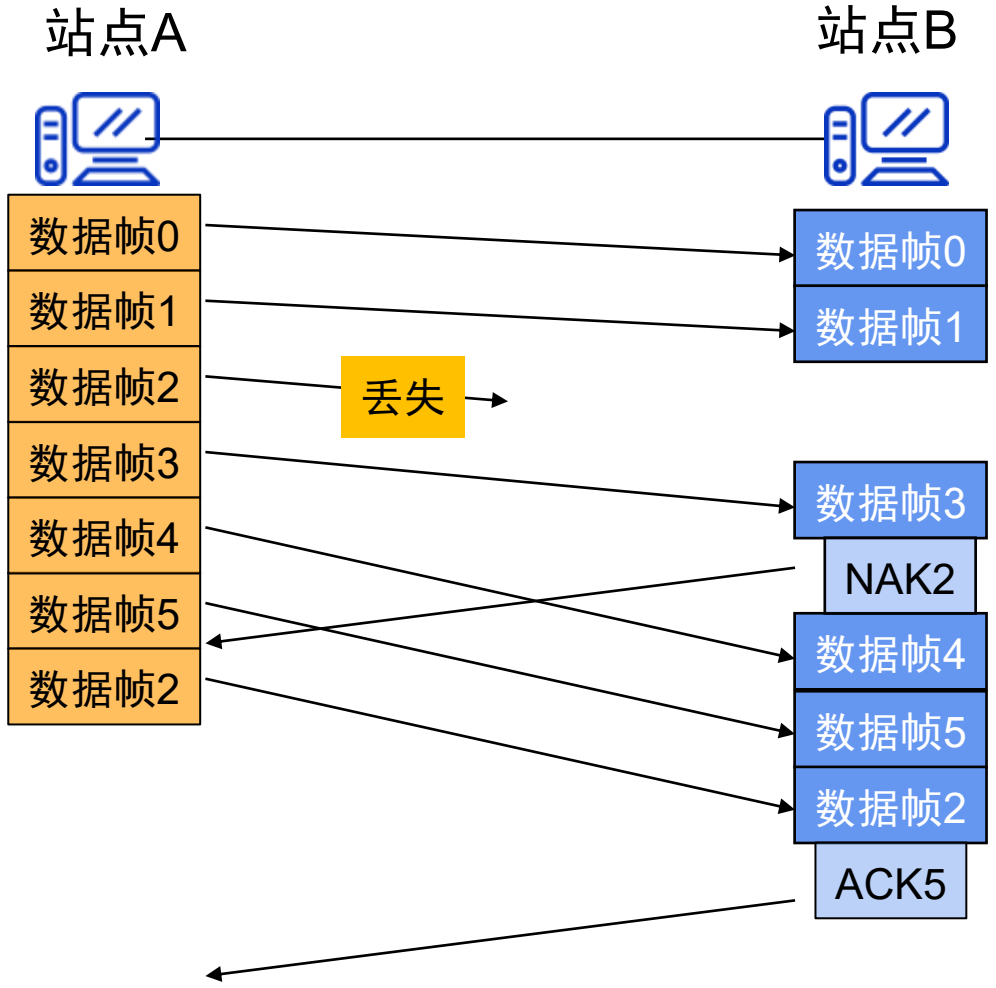
选择拒绝自动重复请求— 差错处理

- 差错情况有三种：
  - 帧破坏
  - 数据帧丢失
  - 应答帧丢失

选择拒绝自动重复请求 — 帧破坏



选择拒绝自动重复请求 — 数据帧丢失



- 问题：如果丢失的是最后一帧？ 接收方不做任何反应，发送方按丢失应答帧进行处理

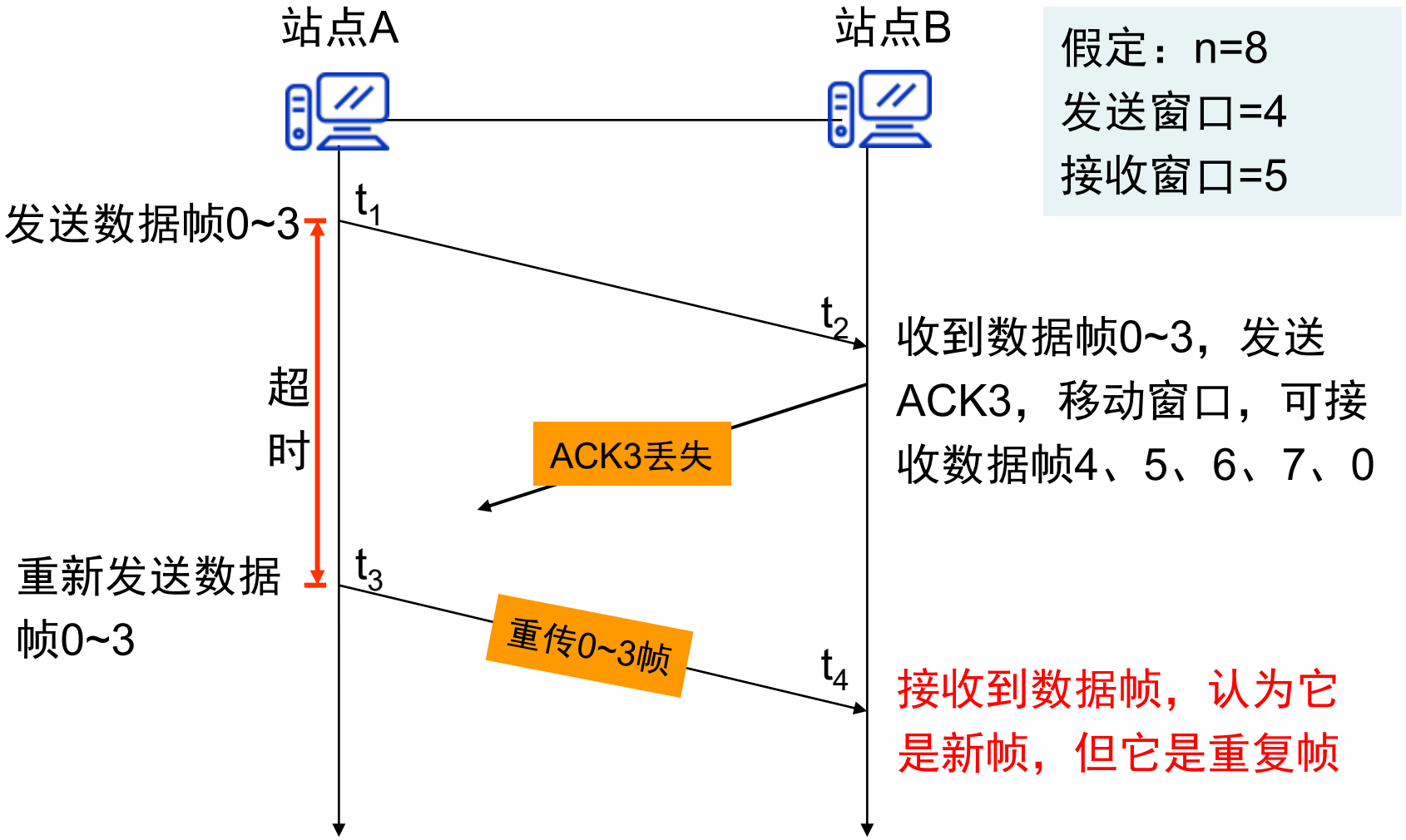
选择拒绝自动重复请求 — 应答帧丢失

- 当发送窗口满时，或传输完毕时，启动超时定时器
- 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

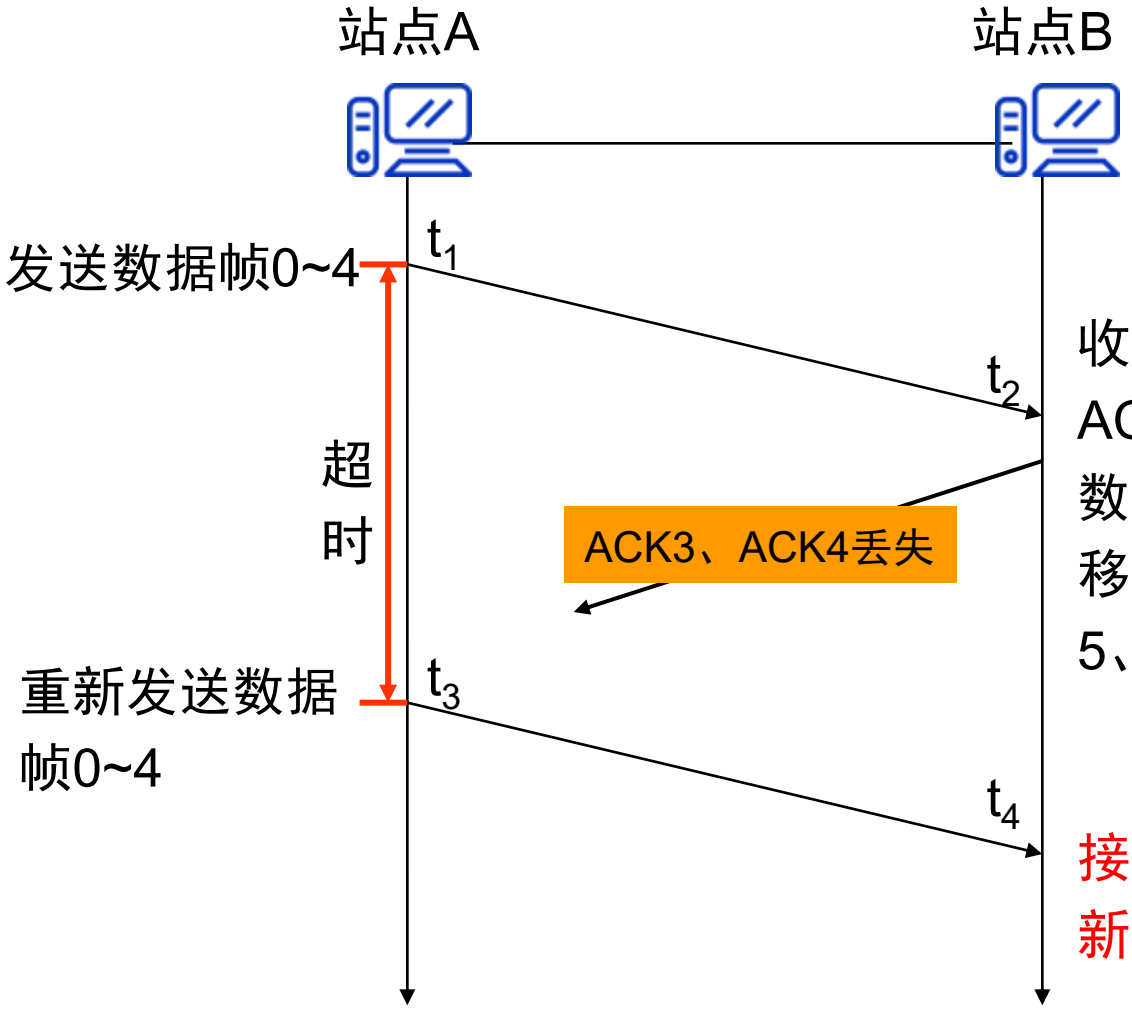
窗口大小与帧编号范围的关系

- 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，协议会失败



接收窗口尺寸太大，协议失败



假定：n=8
发送窗口=5
接收窗口=4

收到数据帧0~3，发送ACK3，移动窗口，接收数据帧4，发送ACK4，再移动窗口，可接收数据帧5、6、7、0

接收到数据帧0，认为是新帧，但它们是重复帧

滑动窗口协议回退N的传输效率

- 不考虑应答帧的丢失，正确传送一帧所需的平均时间为：

$$t_V = t_I + pt_W/(1\text{-}p)$$

- 系统最大吞吐量：

$$\lambda_{max} = 1/t_V$$

- 系统的传输效率：最大吞吐量/极限吞吐量

$$\eta = t_I/t_V$$
$$= (1\text{-}p)/(1+p(t_w/t_i\text{-}1))$$

➢ 结论：

传输效率与差错率有关，无差错时，p=0，滑窗协议的传输效率 η=100%；有差错时，传输效率与传送一帧的总时间和发送端发送一帧的时间之比有关，如果线路传输延时、应答帧时间越长，则传输效率下降越多

计算举例（1）

- 卫星通信中，设p=0.01， 数据帧长度为1200bit，线路速率为4.8kbps，线路长度为160km，传播延迟为250m/μs，应答帧长为120bit，则：

$$t_l = 250\text{ms},\quad t_P = 250\text{ms},\quad t_S = 25\text{ms}$$

  - 对于停止等待协议，传输效率： $\eta = 0.32$
  - 对于滑动窗口协议，传输效率: $\eta = 0.97$

- 结论：传播延迟大，等待应答帧时间很长，即使差错率 p 高达0.01，滑动窗口协议也比停止等待协议好很多

计算举例（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，信号传播速度200m/μs，应答帧长为120bit，则：

$$t_l=125\text{ms},\quad t_P=0.8\text{ms},\quad t_S=12.5\text{ms}$$

所以：

  - 对于停止等待协议，有：η =0.89
  - 对于滑动窗口协议，则有：η=0.989

➢ 结论：当传播延迟相对很小时，二者差别不明显

窗口大小的选择

- 窗口太大，会要求有足够大的缓存空间
- 窗口太小，由于传播和发送延迟，第一个应答帧返回之前，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，应答帧的发送时间是t_S，传播时间为t_P，则窗口的大小n应该满足如下条件：

$$nt_I >= t_I+t_S+2t_P$$

3 种协议方法总结

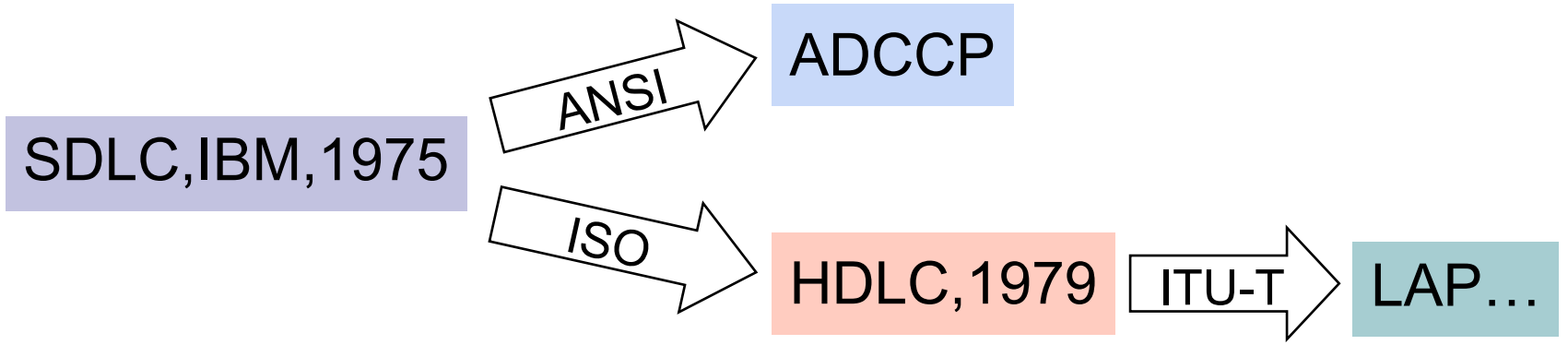
- 停止等待协议：
  - 发送窗口=1，接收窗口=1
  - 传输效率低
  - 协议实现比较简单
- 滑动窗口中的回退N协议：
  - 发送窗口>1，接收窗口=1
  - 传输效率高
  - 协议实现复杂
- 滑动窗口中的选择拒绝协议：
  - 发送窗口>1，接收窗口>1
  - 传输效率高
  - 协议实现更加复杂

4.2 数据链路控制协议

- HDLC通信协议
- PPP协议
- 局域网逻辑链路控制协议LLC

HDLC协议概述

- HDLC(High level Data Link Control)是面向比特的通信协议
- 面向比特的通信协议将帧看作有定界符的连续比特流，通过一些比特位在帧中的位置以及与其它比特位的组合模式来表达含义及协议
  - 1975年，IBM开发了面向比特的协议—同步数据链路控制（SDLC）
  - 1979年，ISO基于SDLC提出了高级数据链路控制（HDLC）协议
  - 1981年开始，ITU-T开发了一系列基于HDLC的链路访问协议（LAPB，LAPD，LAPM等）
  - ITU-T和ANSI研制的PPP协议也是从HDLC协议发展而来的
  - 局域网逻辑链路控制协议LLC思想也源于HDLC协议



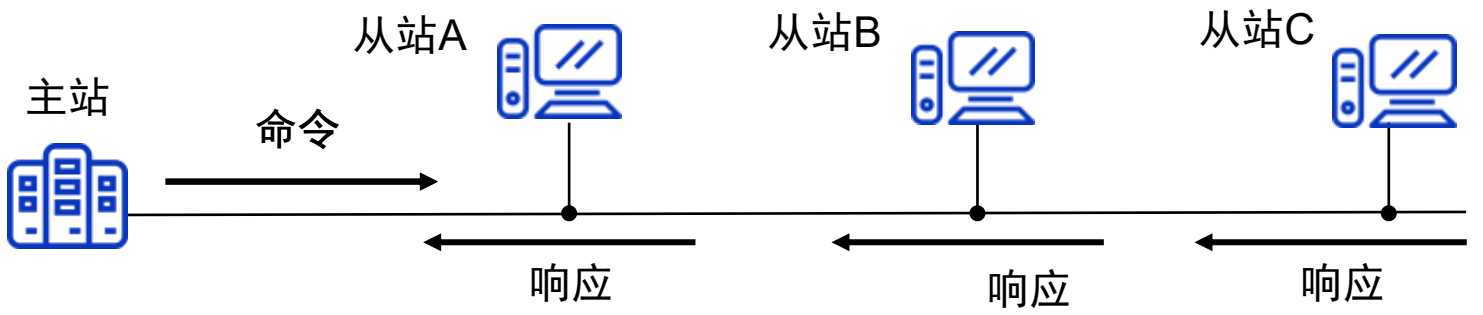
站点类型

- HDLC支持点对点和多点配置下的半双工和全双工模式
- 协议内容主要通过站点类型、链路配置和通信方式及相应的处理来描述
- HDLC协议有三种站点类型：
  - 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

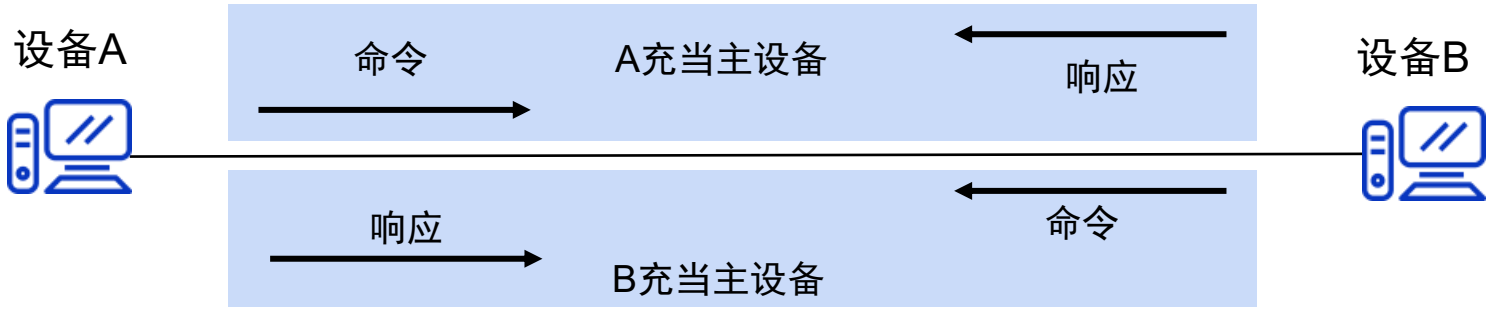
链路配置

- 3种配置方式：非平衡式（多点）、对称式（点到点）、平衡式（点到点）

- 非平衡式



- 对称式



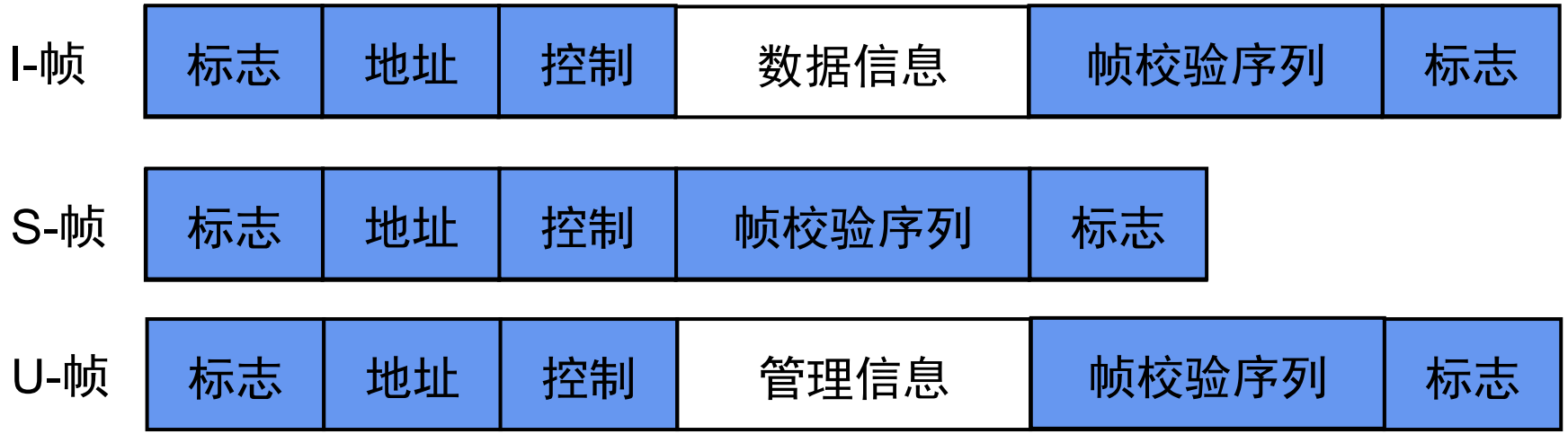
- 平衡式

通信方式

- 通信方式：在一次通信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
- HDLC协议支持 3 种工作方式：
  - 正常应答方式NRM（非平衡式）
    - 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询问并响应后，才可以向主站发送数据帧
  - 异步应答方式ARM（非平衡式）
    - 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
    - 只要信道空闲，从站不需要获得主站许可就可以发送数据
    - 未改变主从关系，即一个从站与另一个从站的数据传输必须经过主站的转发
  - 异步平衡方式ABM
    - 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

HDLC帧格式

- HDLC协议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
  - 监管帧（S-帧）：流量和差错控制信息
  - 无编号帧（U-帧）：链路管理服务



HDLC帧格式：标志字段

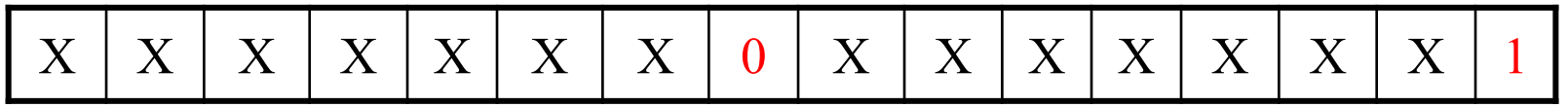
- 一个字节(8位)，比特模式为01111110
  - 表示一个帧的开始和结束
  - 为接收方提供同步手段
- 成帧方法：比特填充法
  - 发送方发送一个含有5个以上连续1的数据时，总是在第五个1后面插入一个冗余的0。不管第六个比特是0还是1
  - 接收方接收时作相反的动作(去掉5个1后面的0)
  - 例如：要发送的序列是01111101111110，发送时变成011111001111101 0

HDLC帧格式：地址字段

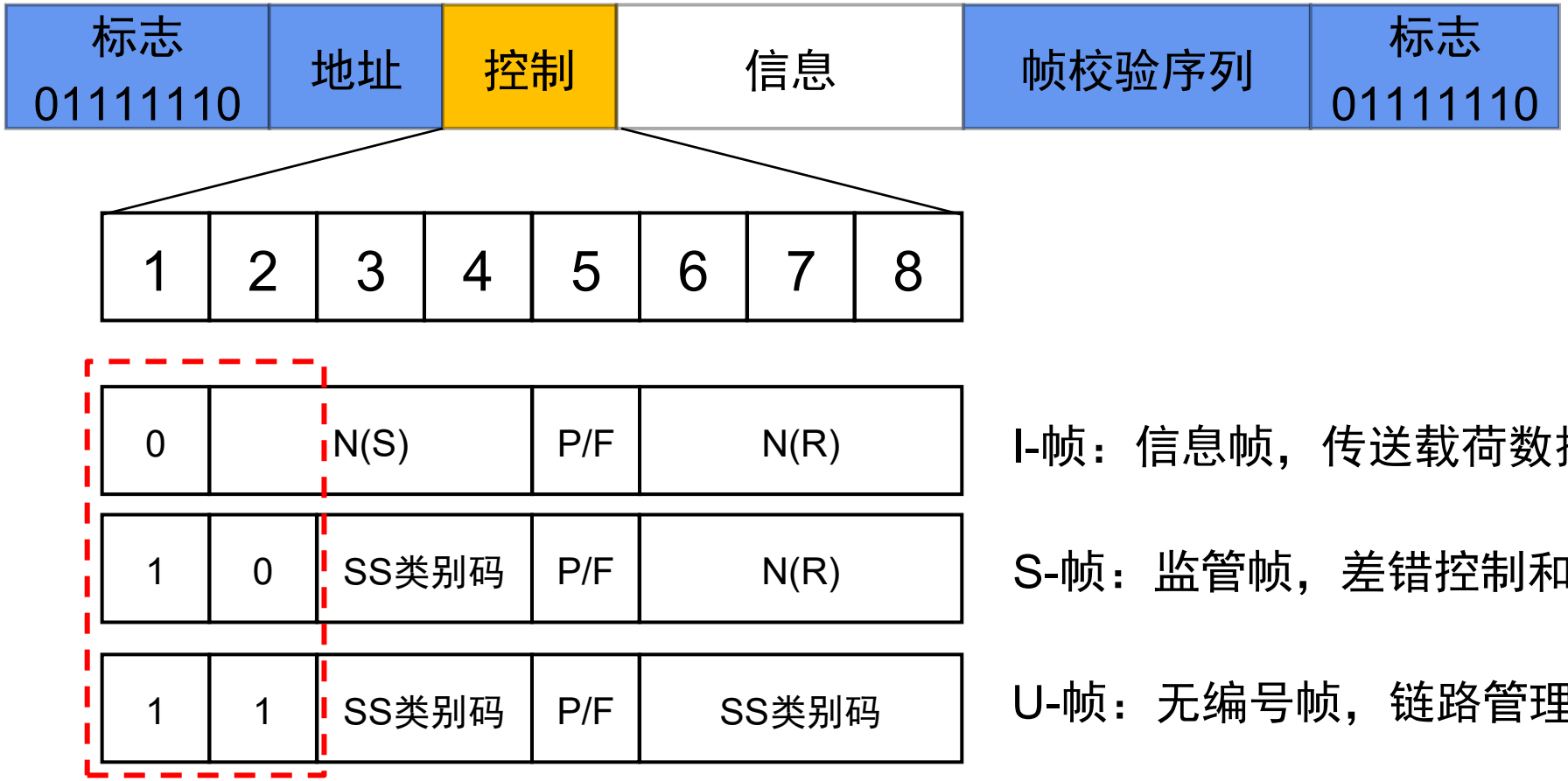
- 该字段是指从站地址，或者是以从站方式运行的复合站地址
  - 如果帧是由主站发送的，则地址表示接收该帧的从站地址
  - 如果帧是由从站发送的，则地址表示发送该帧的从站地址
- 每个从站分配一个唯一的地址
  - 某一地址可分配给多个站点，这种地址称为组地址
  - 全“1”地址来表示包含所有点站的地址，称为广播地址

HDLC帧格式：地址字段

- 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



HDLC帧格式：控制字段



I-帧：信息帧，传送载荷数据

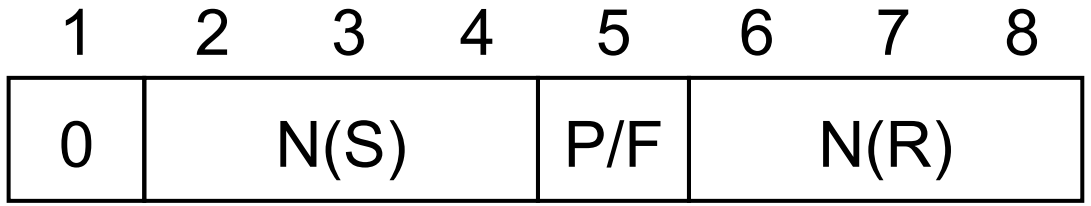
S-帧：监管帧，差错控制和流量控制

U-帧：无编号帧，链路管理

- 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

HDLC帧格式：I-帧的控制字段

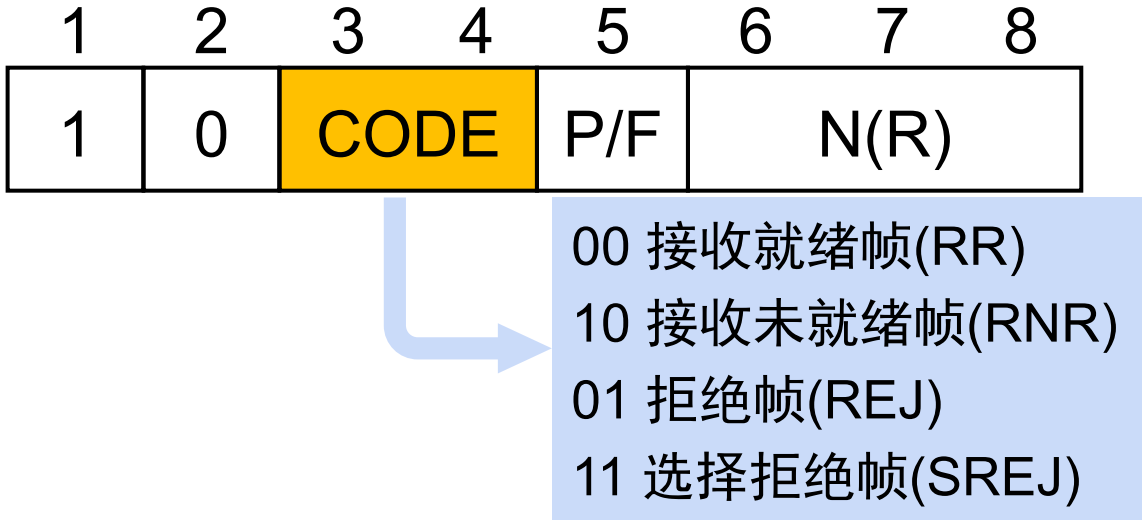
I-帧的控制字段：



- 用途：用于流量管理
- 第1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，取值0~7，可以以滑动窗口协议方式连续发送多帧，多帧一起应答确认
- 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，用于接收方以滑动窗口协议方式接收时，给发送方发送应答消息时期望收到的帧序号，取值0~7

HDLC帧格式：S-帧的控制字段

S-帧的控制字段：



- S-帧没有信息字段，通过控制字段给接收方带去某种信息
- 信息的含义需要通过S-帧的类型和传输上下文来获得，有4种控制信息
- 当不能在一个I-帧上捎带确认信息时，用S-帧来对收到的数据帧进行应答
- 如果最近一帧是正确的，N(R)域是序列中下一帧的编号
- 如果最近一帧是错误的，N(R)域是这个损坏帧的编号

监管帧：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

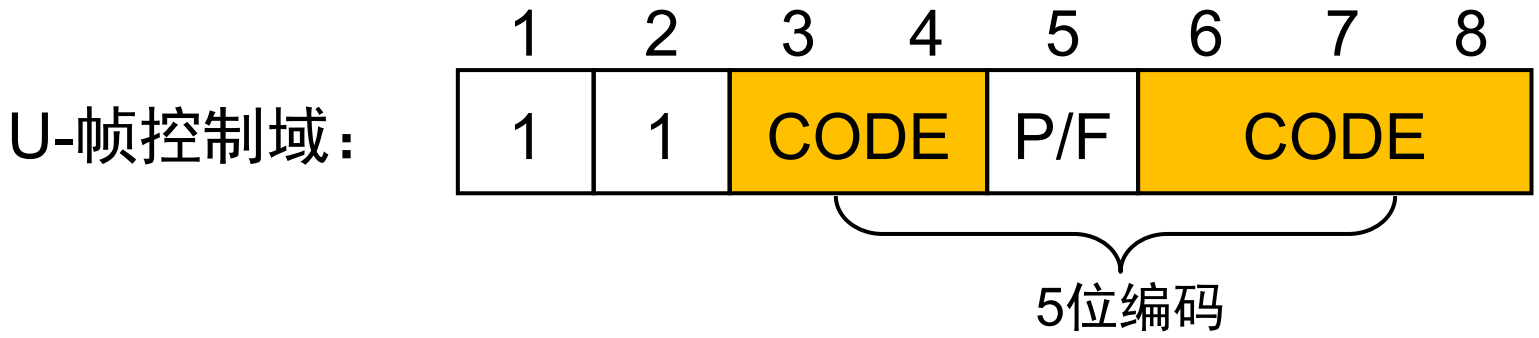
监管帧：接收未就绪(RNR)帧

- 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

监管帧：REJ帧和SREJ帧

- 拒绝 REJ帧
  - 在回退N自动重复请求中，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
- 选择拒绝 SREJ 帧
  - 在选择拒绝自动重复请求中，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

HDLC帧格式：U-帧的控制字段



- 用来在互连设备之间交换会话管理信息和控制信息
- 控制域中有5位编码位，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
- U-帧中的P/F位一般都应置1

各种类型的无编号帧

编码	名称	性质	意义
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。
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。
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。
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。
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。

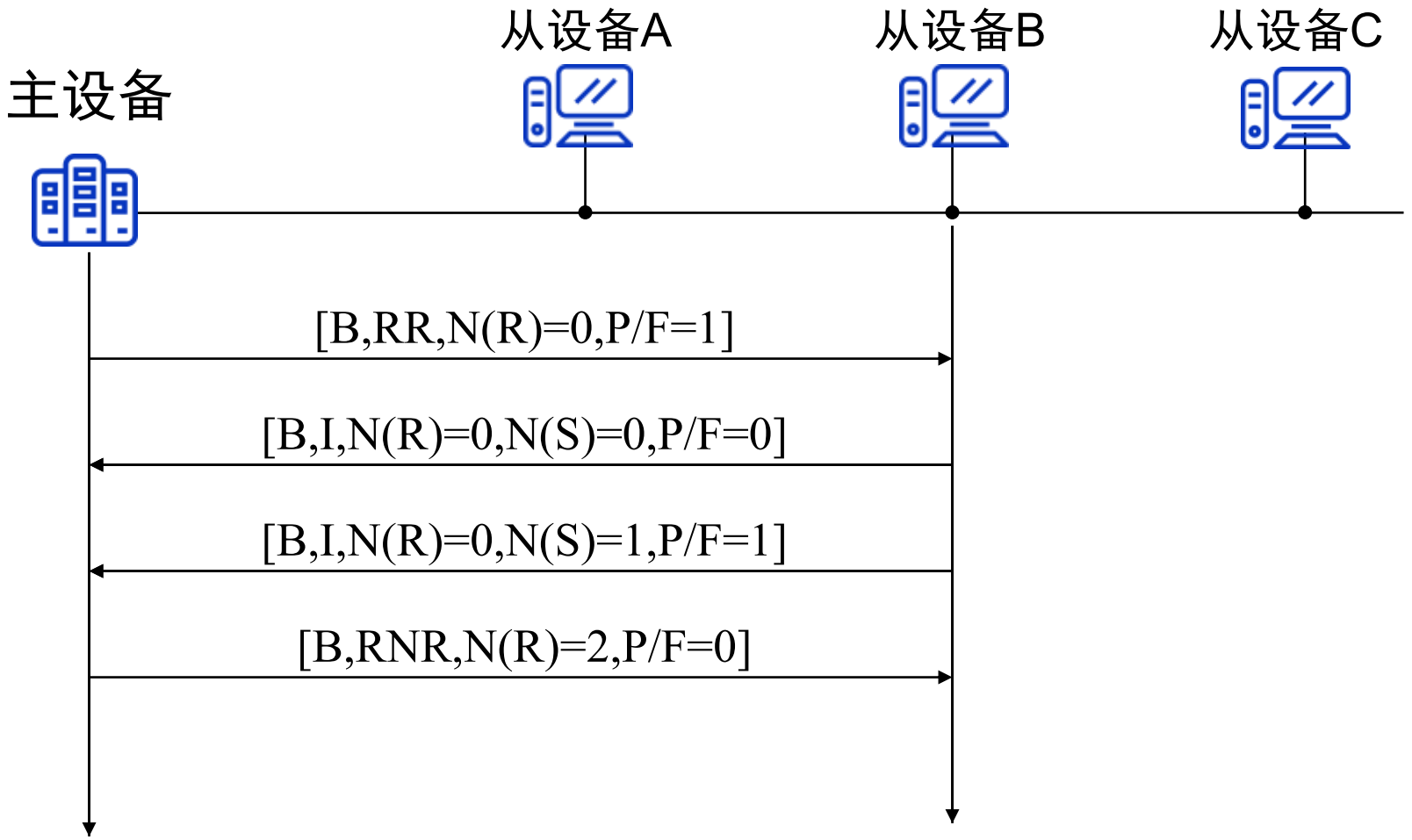
HDLC帧格式：信息字段

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

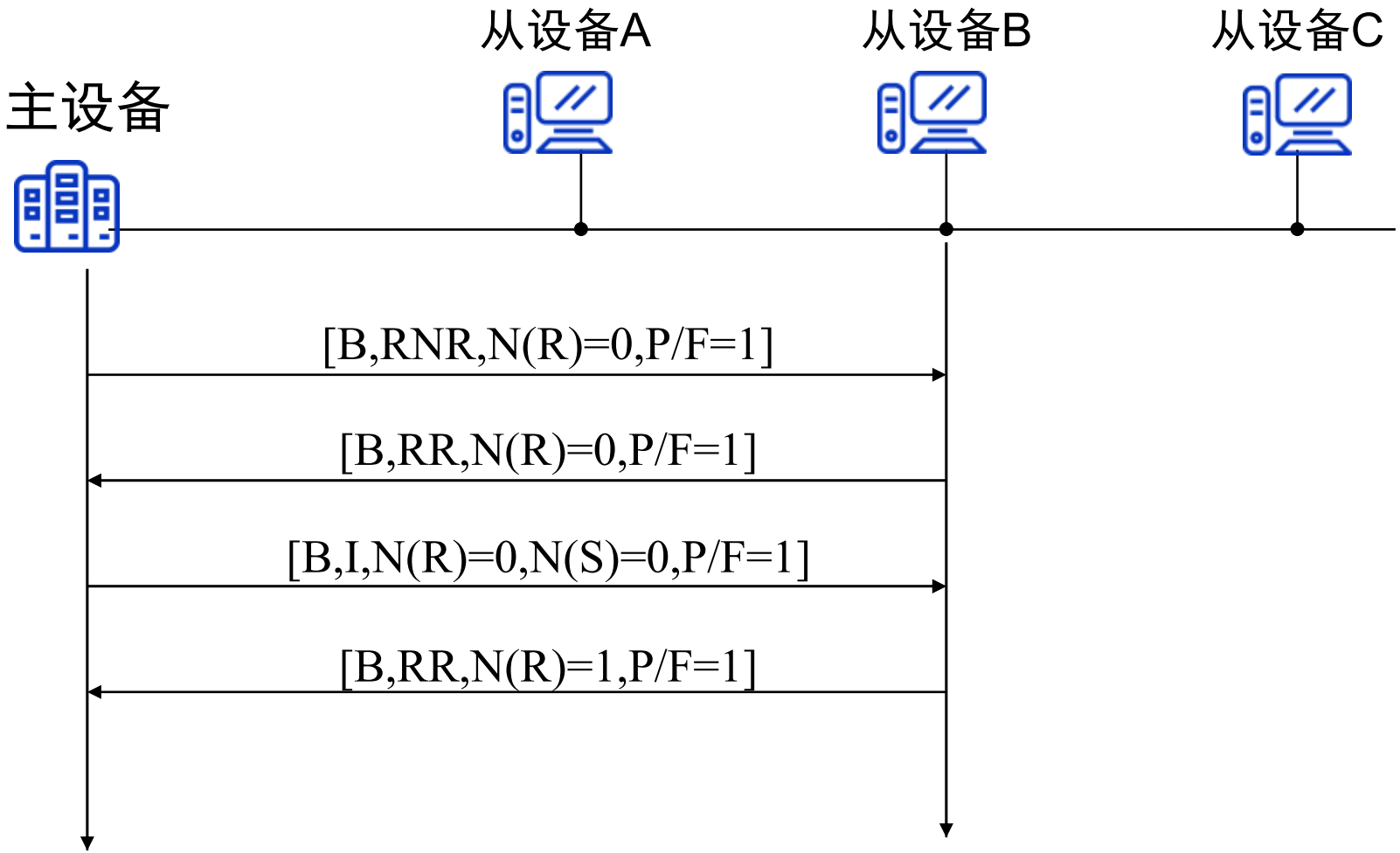
HDLC帧格式：帧校验序列字段

- 帧校验序列是HDLC协议的错误检测域
- 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：$X^{16}+X^{12}+X^{5}+1$

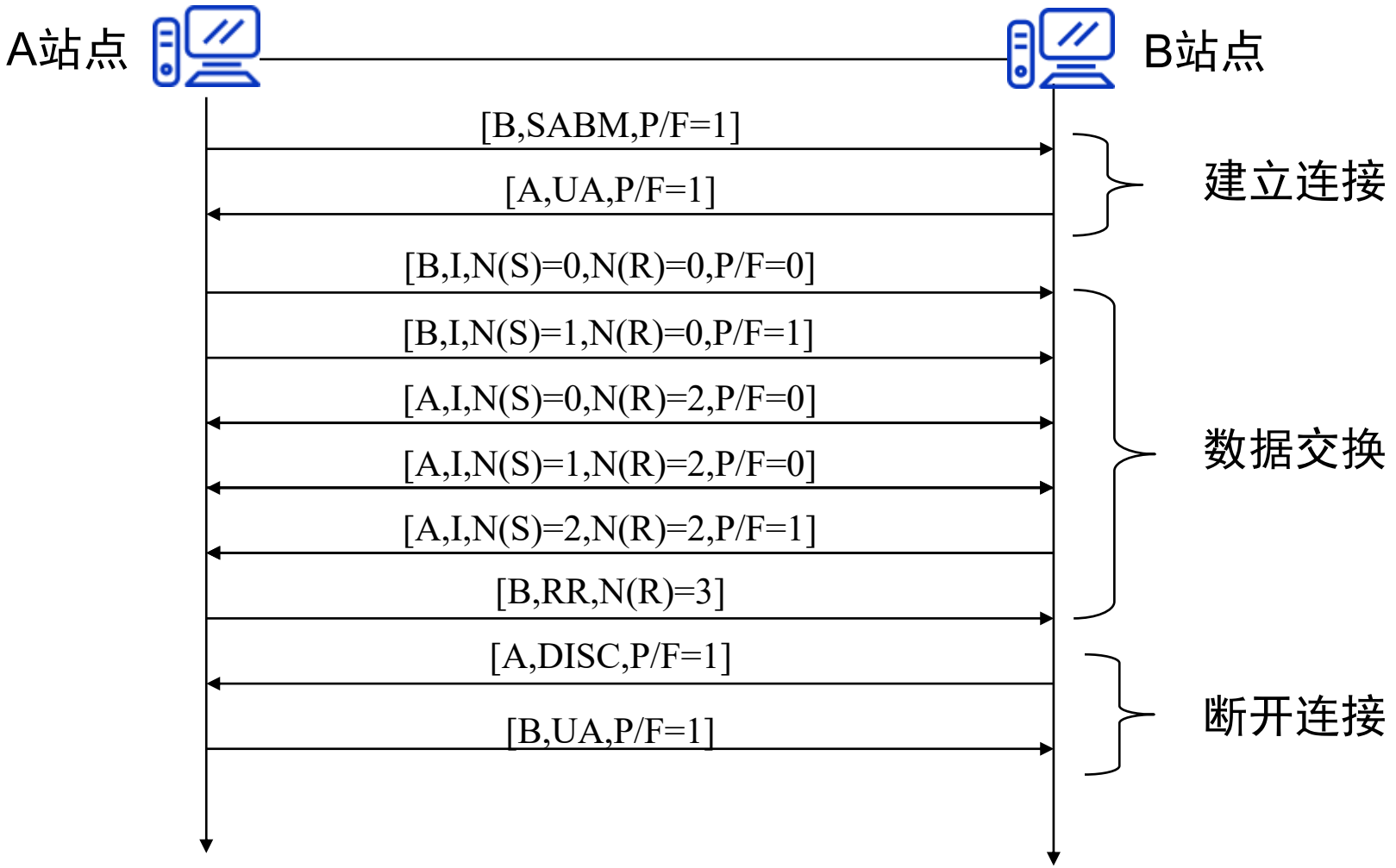
HDLC协议的通信实例：查询/响应



HDLC协议的通信实例：选择/应答



HDLC协议的通信实例：对等通信



PPP协议概述（1）

- PPP(Point-to-Point Protocol)协议是在串行线路互联网协议SLIP基础上发展起来的，由IETF制定，在1994年成为正式标准（RFC1661)
- 能够承载不同的网络层数据包，且能够运行在不同的链路上
- 是目前使用最多的数据链路层协议之一
- 使用点对点链路来传输数据包
- 是面向字节的通信协议，使用字节填充技术，帧长度是字节整数倍

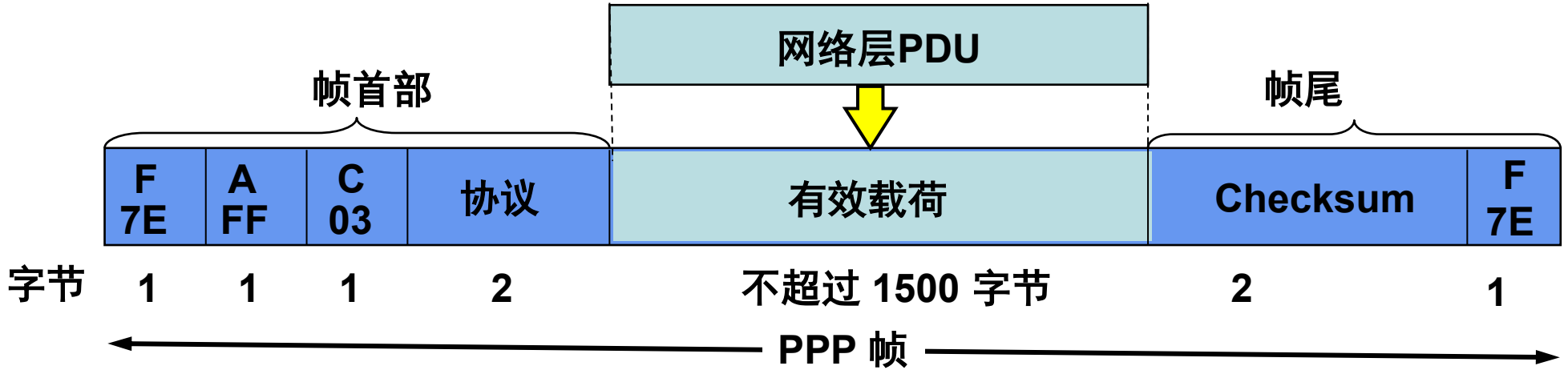
PPP协议概述（2）

- PPP 协议提供 3 种服务：
  - 封装 (Encapsulation)
    - 提供在同一链路上支持不同的网络层协议
    - PPP既支持异步链路（无奇偶检验的8比特数据），也支持面向比特的同步链路
    - IP数据包在PPP帧中是其信息部分，其长度受到MTU的限制
  - 链路控制协议 LCP (Link Control Protocol)
    - 用来建立、配置和测试数据链路，通信双方可协商一些选项
  - 网络控制协议 NCP (Network Control Protocol)
    - 其中每个协议支持一种不同的网络层协议，如IP、OSI的网络层、DECnet、AppleTalk等

PPP协议概述（3）

- PPP协议作为数据链路层协议，支持以下功能：
  - 利用帧定界符封装成帧
  - 填充技术实现透明数据传输
  - 帧的差错检测
  - 链路配置
  - 实时监测链路工作状态
  - 设置链路最大传输单元（MTU）
  - 网络层地址协商机制
  - 数据压缩协商机制
  - 允许身份认证

PPP协议帧格式



- PPP 帧以字节为单位
  - 帧首4 个字段、帧尾2 个字段
  - 标志字段 F： 0x7E （二进制 01111110），作为帧定界符
  - 地址字段 A：置为 0xFF
  - 控制字段 C：通常置为 0x03

- 协议字段：标识有效载荷种类：
  - 值为 0x0021 —— IP 数据报
  - 值为 0x8021 —— 网络控制数据
  - 值为 0xC021 —— 链路控制数据
  - 值为 0xC023 —— 鉴别数据
- 尾部校验和字段：用于差错检测

PPP协议不保证可靠传输

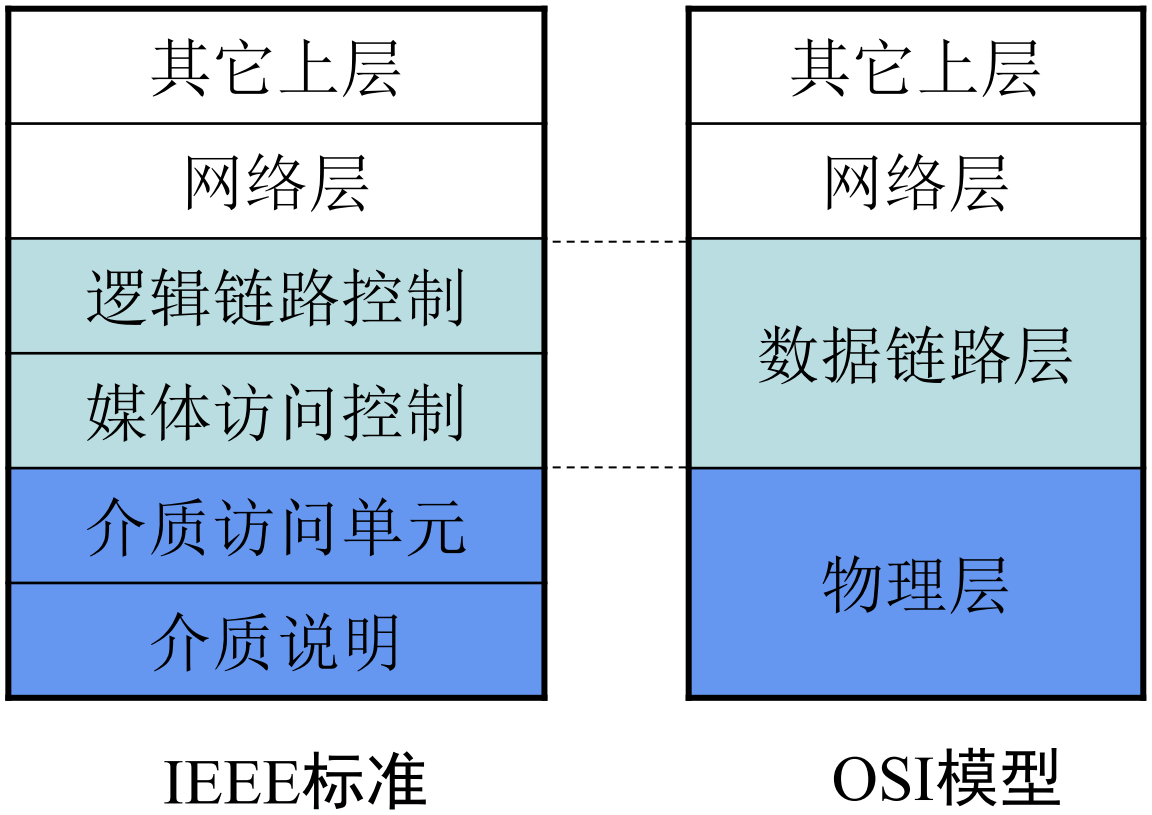
- PPP协议对数据帧进行“差错检测”，错误帧不被接收
  - FCS 字段可保证无差错的数据帧才被交付给网络层
- PPP协议不提供“不丢帧”保证
  - PPP 协议不使用序号和确认机制，不提供“不丢帧”的可靠传输
    - 在数据链路层出现差错的概率不大时，使用比较简单的 PPP 协议较为合理
    - 在互联网环境下，链路层传输的可靠性远比网络层 IP的可靠性高得多

PPP协议和 HDLC 协议的区别

- PPP协议和 HDLC 协议都可用于点对点的数据链路层通信，两者帧格式相似，但它们之间也有不少区别：

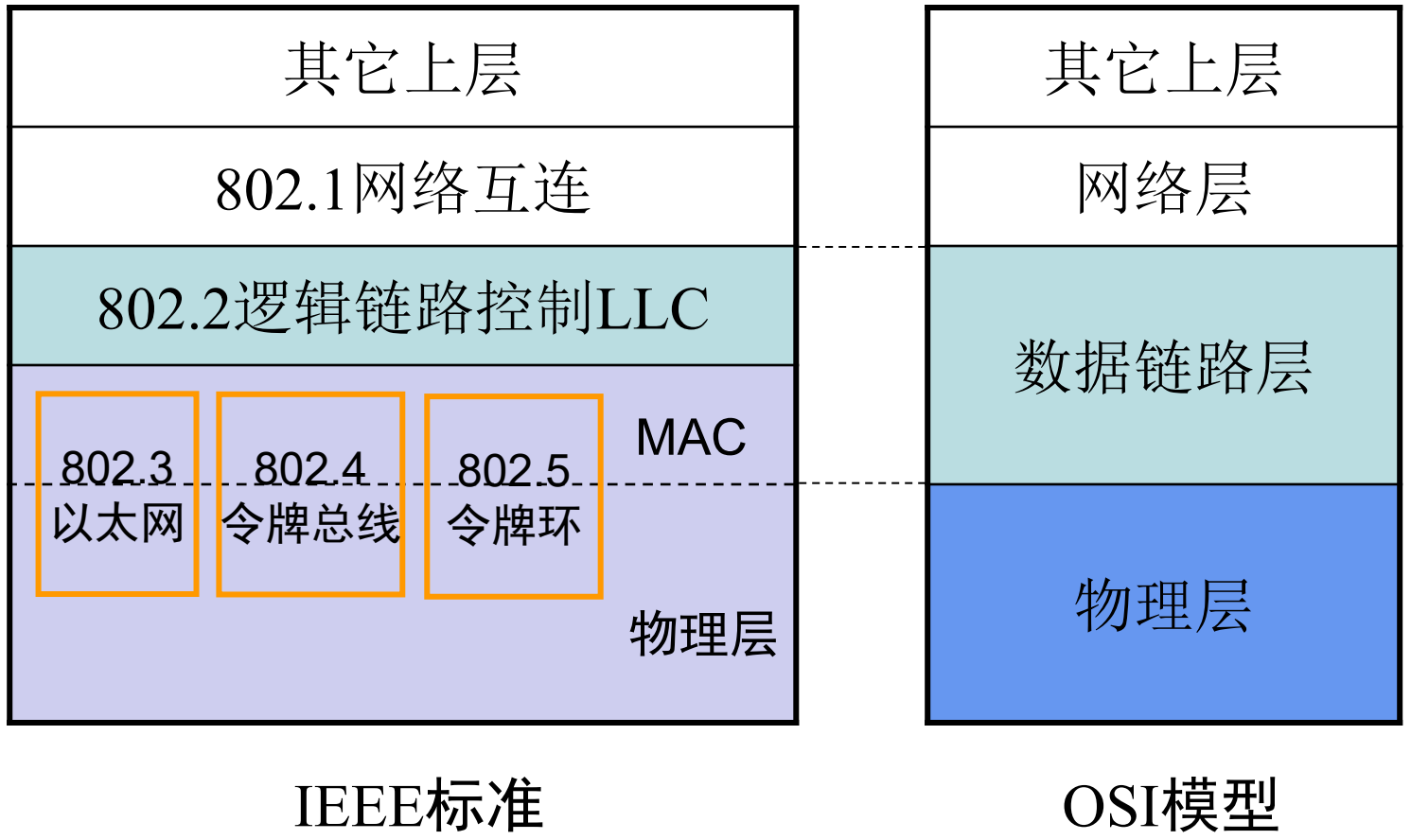
	PPP协议	HDLC协议
标准	IETF	ISO
帧类型	面向字节，帧长度是字节整数倍	面向比特，长度可以不是字节整数倍
成帧方法	字节填充法	比特填充法
差错检测	只检错，但不纠错	检错并纠错
传输可靠性	无差错控制，不保证帧不丢失	采用滑动窗口、确认和超时机制等差错控制技术，保证帧不丢失
适用链路	双工	半双工和双工
通信方数量	仅点对点，两点之间通信	可以点对点，可以多点

IEEE 局域网参考模型



- 与OSI模型对比，IEEE局域网标准的层次划分：
  - 物理层分为两个子层
    - 下子层—介质说明
    - 上子层—介质访问单元(MAU)
  - 数据链路层分为两个子层
    - 下子层—媒体访问控制子层(MAC)
    - 上子层—逻辑链路控制子层(LLC)
  - 增加MAC子层的目的
    - 局域网中各结点是对称的，并支持广播，需要多点平衡配置
    - 逻辑链路控制子层针对不同链路介质可定义统一的协议，屏蔽底层链路差异性

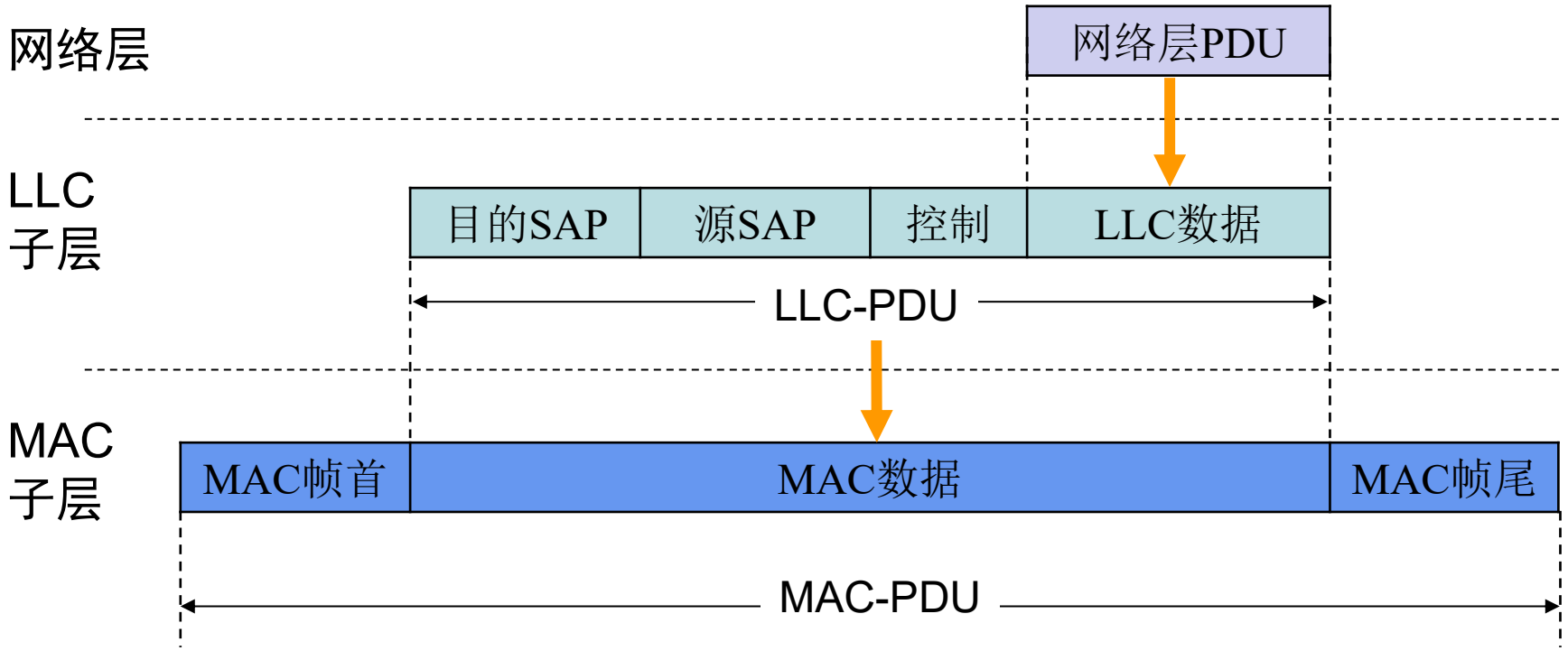
IEEE 局域网模型-802标准



- IEEE 802根据传输介质给出了不同的MAC子层标准，如：
  - 802.3 ： 以太网
  - 802.4 ： 令牌总线
  - 802.5 ： 令牌环
  - 802.11： 无线Wi-Fi
- LLC 子层是透明的，只有下到 MAC子层才知道所连接的是什么样的局域网

逻辑链路控制子层LLC

- LLC子层为网络层提供统一接口
- 通过LLC层实现在不同类型的局域网链路上传送网络层的数据包

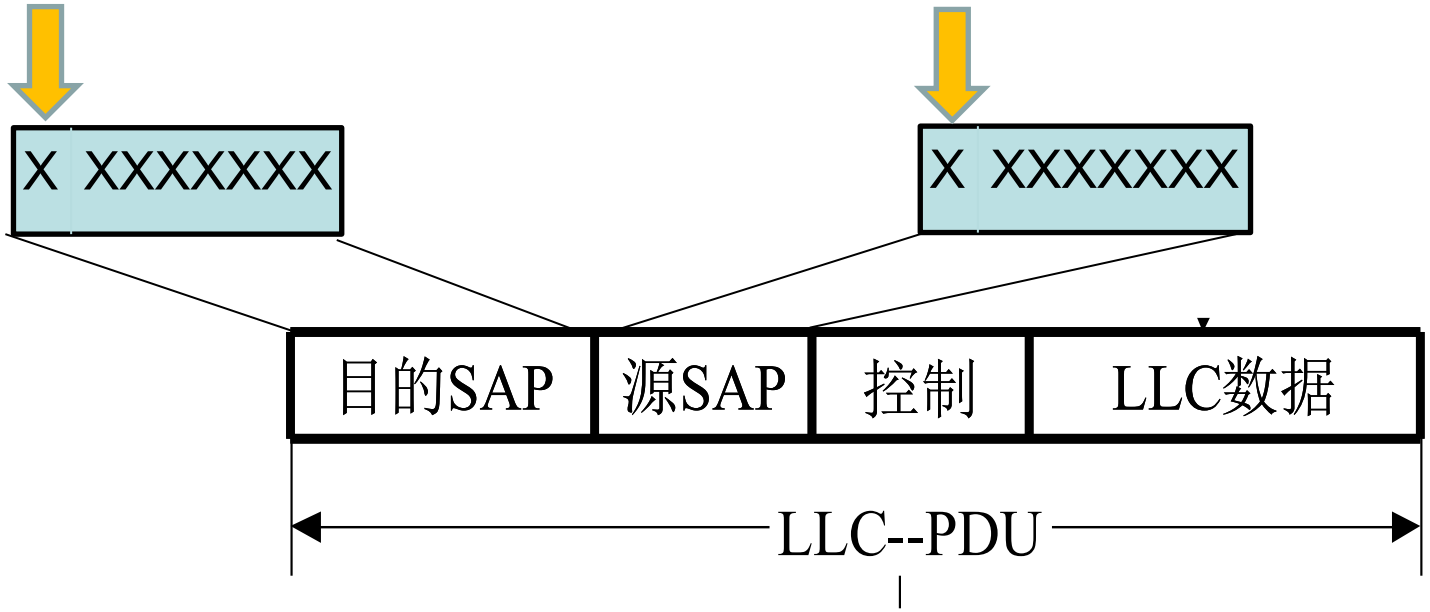


LLC帧格式及和上下层的封装关系

LLC帧中的地址（1）

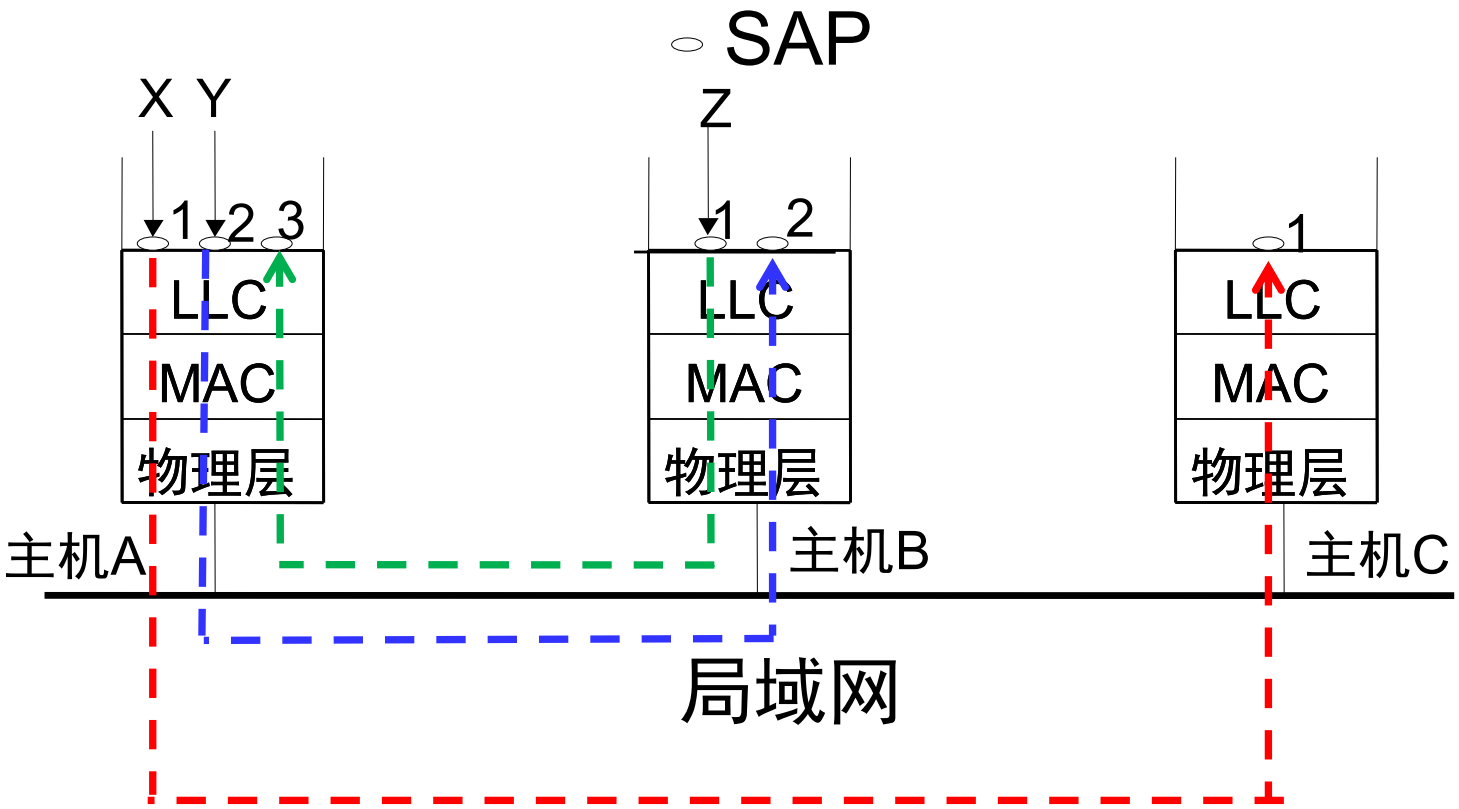
0：单播，向某个体通信
1：组播，向多点通信，multicast
全1：广播通信，broadcast

0：命令帧
1：响应帧



- 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)、控制域和信息域（LLC 数据，来自网络层的 PDU）
- DSAP和SSAP是LLC使用的地址，用来标明接收和发送数据的计算机上的协议栈

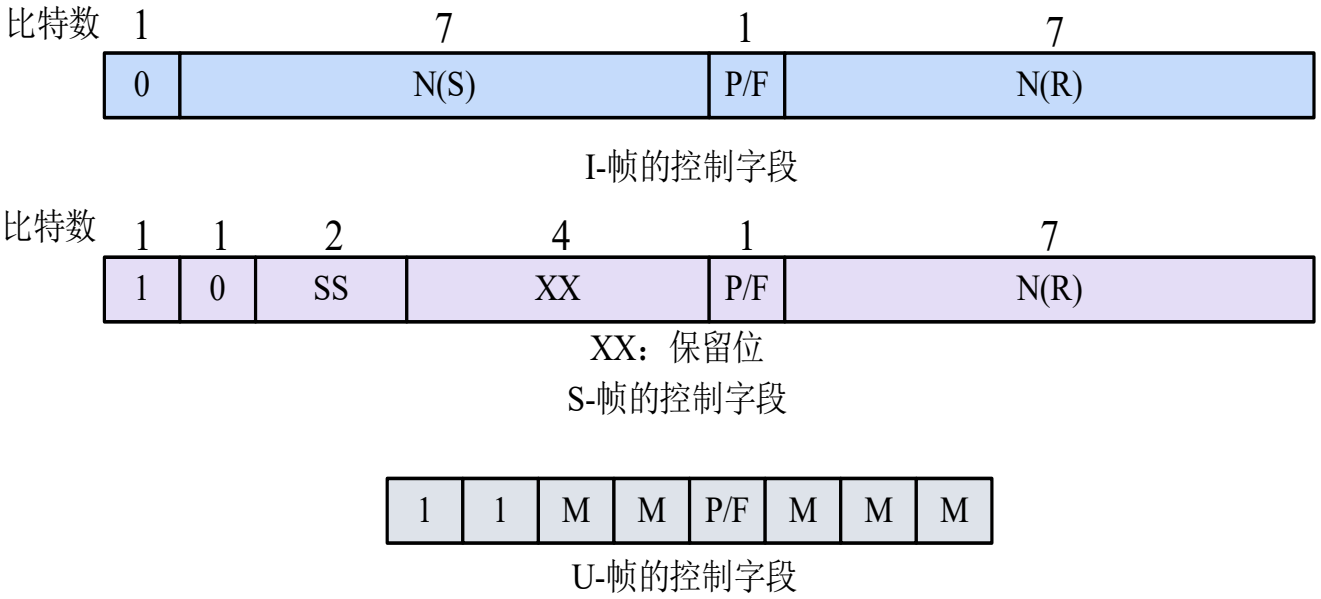
LLC帧中的地址（2）



- 在MAC帧的帧首中，有目的站地址和源站地址，它们是站点的物理地址
- 在LLC帧的帧首中，DSAP和SSAP是逻辑地址，标识的是数据链路层的不同服务访问点
- 在局域网中，一个站点可以有多个SAP，每一个 SAP可以对应一个进程，它们可以复用同个MAC 帧，和另一站的不同 SAP （进程）进行通信

LLC三种帧类型

- LLC帧类型也分为三种：
  - 信息帧、监管帧和无编号帧
  - 通过控制字段来区分
  - 功能和HDLC协议相同
- 与HDLC协议区别：
  - LLC无选择拒绝帧
  - LLC帧无CRC差错校验，CRC校验在MAC子层中进行



LLC 提供的服务

- LLC 层可以提供三类服务
- 无确认无连接服务
  - 发送端将数据帧发送出去后，不需要接收端发送应答帧
  - 端对端的差错控制和流量控制由高层协议（通常是传输层）来实现
  - 应用场景：差错率低的线路，如有线以太网
- 有确认无连接服务
  - 每一帧都得到单独的确认
  - 适用场景：不可靠的信道，如802.11的无线 Wi-Fi
- 有确认有连接服务
  - 通信前建立连接，数据传输时需要接收方确认
  - 适合于无复杂上层协议的简单终端，直接在链路层以上开展网络应用，如蓝牙

4.3 介质访问控制协议

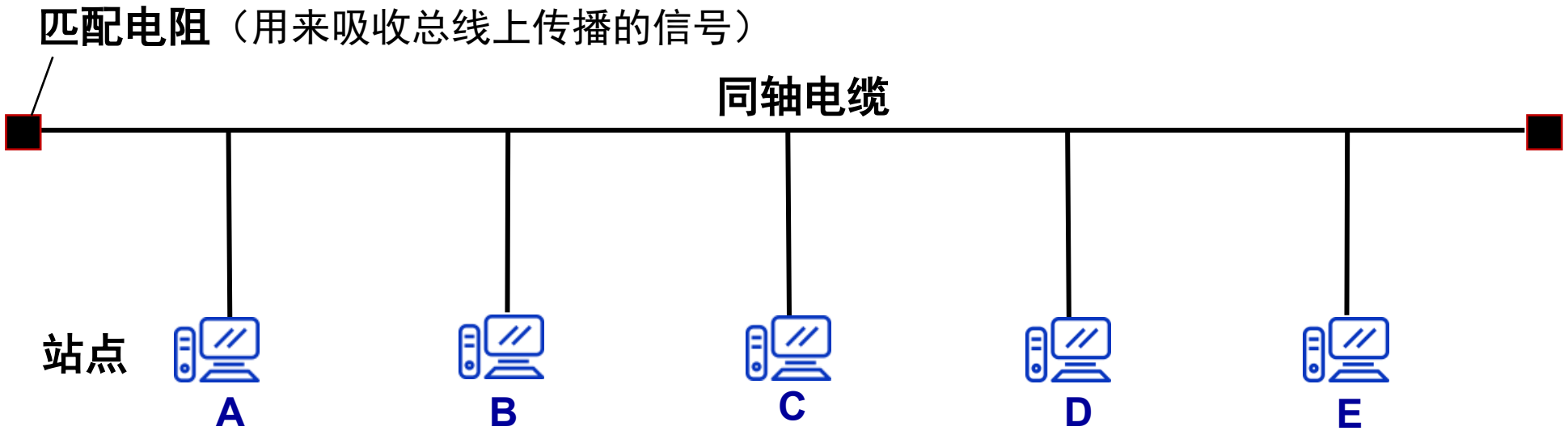
- 以太网
- 无线局域网
- 无线个域网

以太网

- 以太网由 DEC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
  - 基带类划分为5个不同的标准：10Base5、10Base2、10Base-T、1Base5、100base-T
  - 基带使用曼彻斯特编码
- 以太网上的每一个站点都有自己的网络接口卡(NIC)
  - NIC通常安装在站点内部
  - NIC上的物理地址6字节长，任意两个NIC都具有不同的物理地址

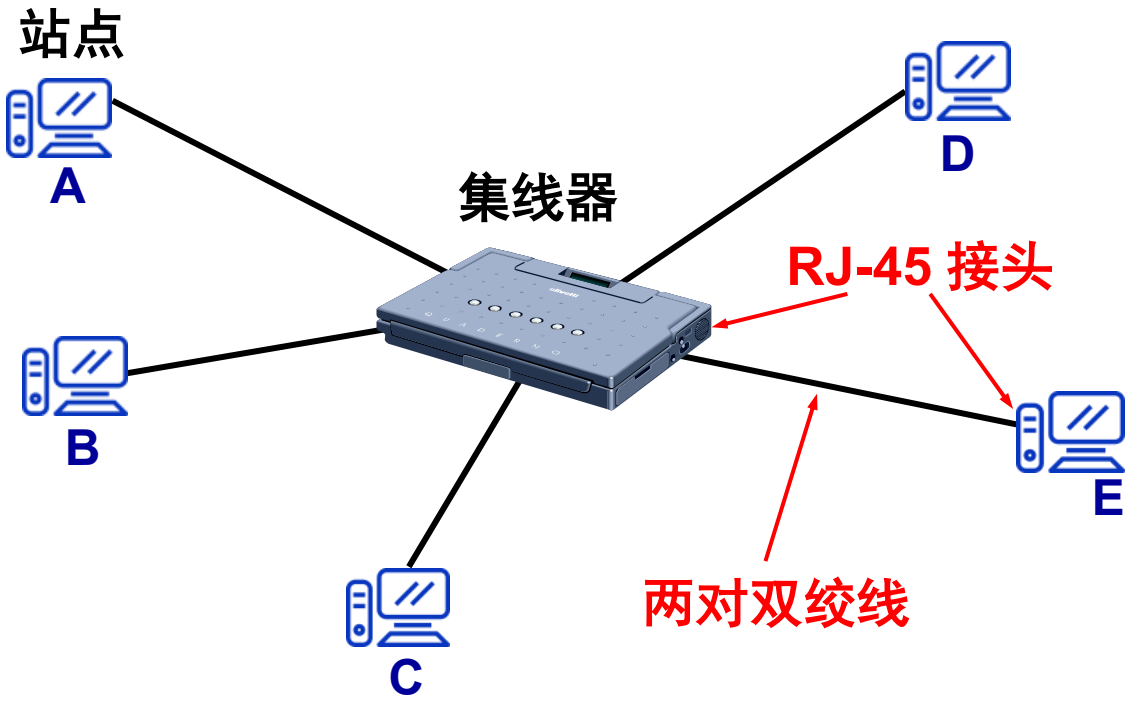
以太网拓扑结构

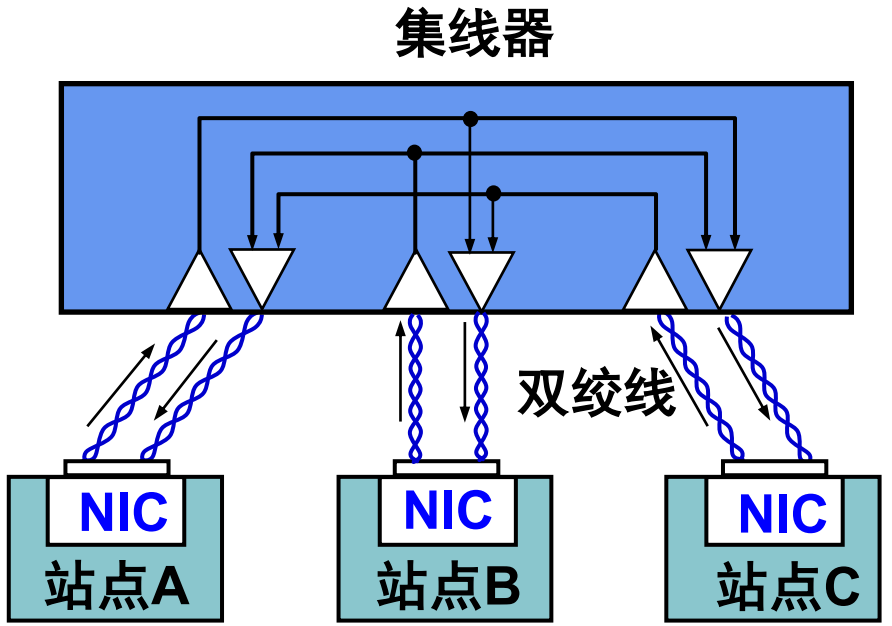
- 1990年前的以太网，使用同轴电缆，采用总线形拓扑结构
  - 10Base5标准,10Base2标准
  - 是共享介质的广播信道



以太网拓扑结构

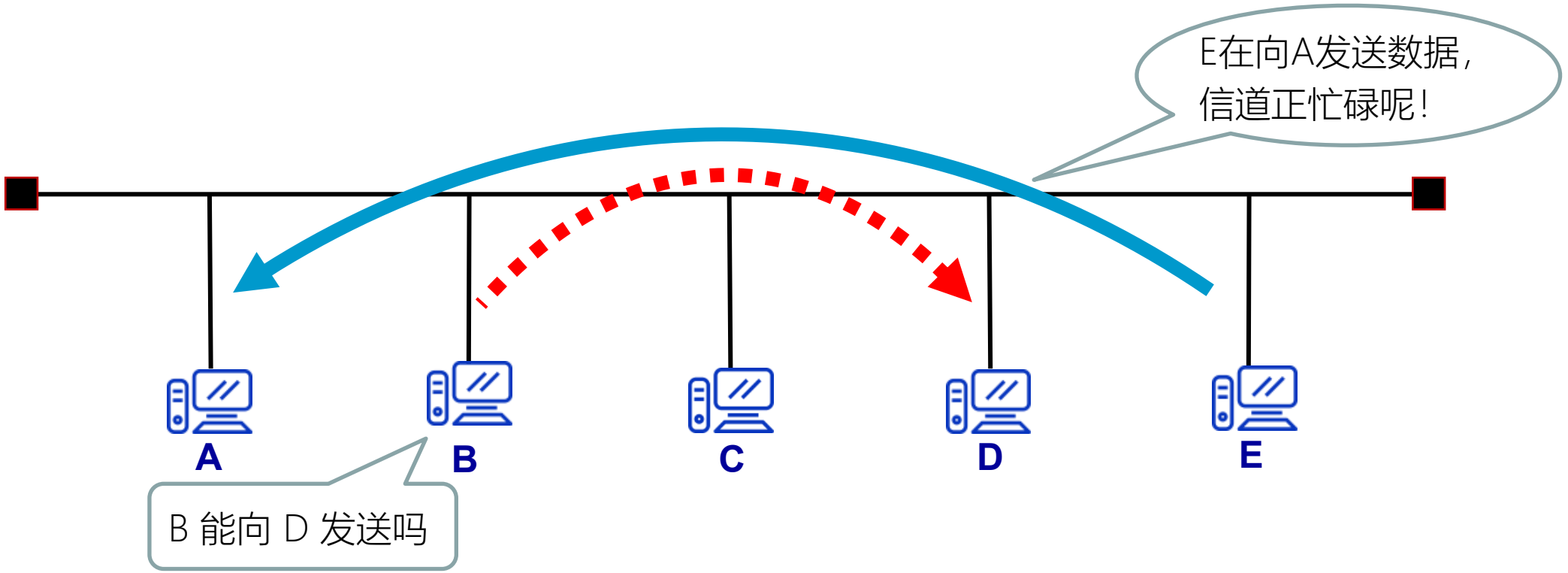
- 1990 年，IEEE 制定出802.3i标准，使用双绞线的星形拓扑
  - 使用集线器和双绞线， 10BASE-T 标准
  - 双绞线便宜，使用方便、灵活
  - 是共享介质的广播信道





共享广播信道以太网的冲突问题

- 冲突：多个用户不加控制同时访问共享信道时，会存在由于不同信号叠加而相互破坏的情况
- 发生冲突时，信道上传输的信号会产生严重的失真，故无法从中恢复出有用的信息来



载波侦听多路访问CSMA

- 多路访问(MA)
  - 多个站点连接在一个共享信道上，一个站点发送的数据，可同时被多个站点接收
  - 无通信管制，依赖于响应，判断数据帧是否被其它通信破坏
- 载波侦听多路访问(CSMA)
  - 每一个站点在发送数据之前检测电路上的电压值，判断信道是否忙碌
  - 如果信道空闲，可以开始传输
  - 如果信道忙，则避让一段时间，然后再试图发送，以减少冲突的机会

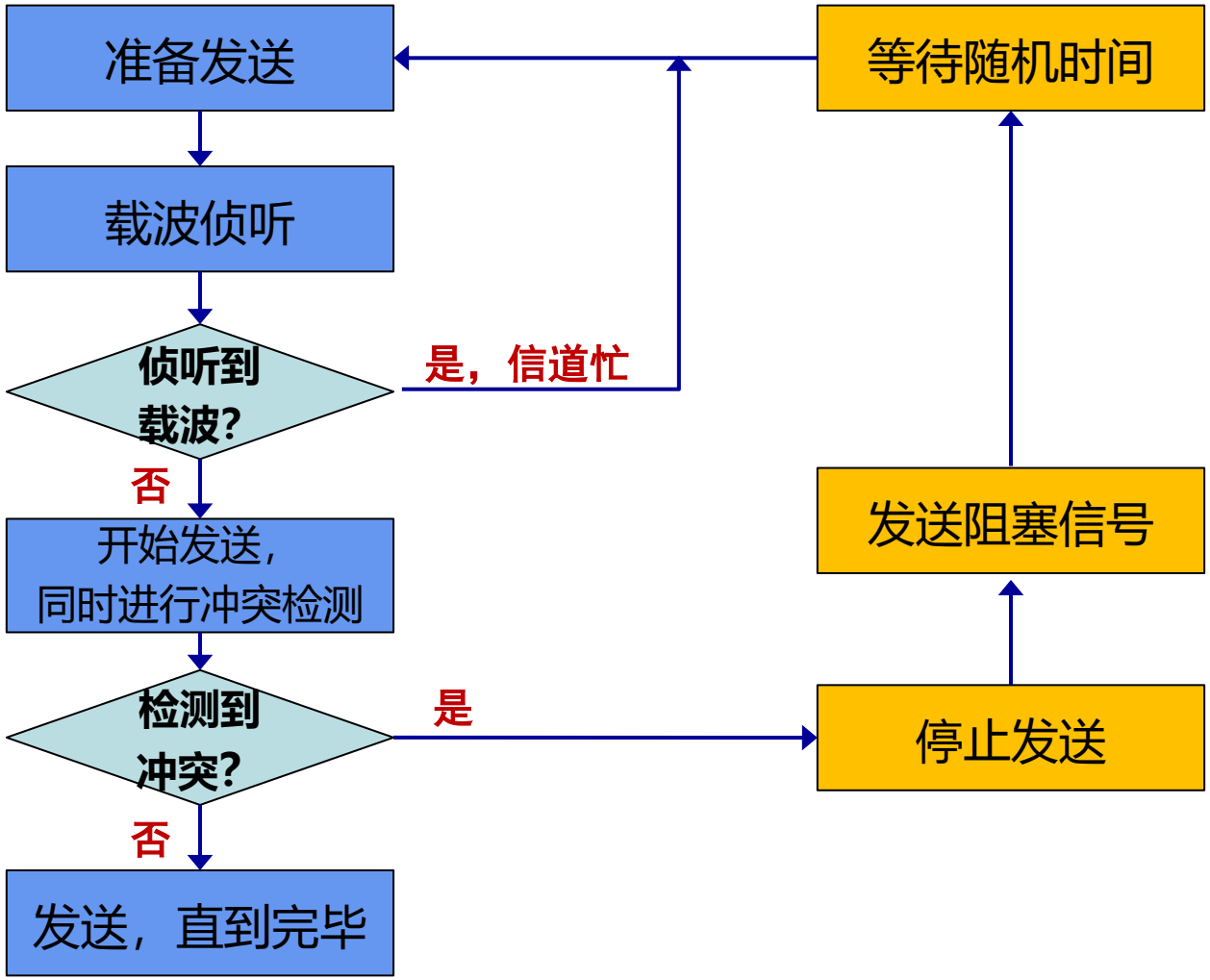
CSMA分类

- 依据避让时间的选择方法，CSMA可分为三种：
  - 非坚持CSMA
  - 坚持CSMA
  - P-坚持CSMA
- 尽管CSMA发送前先侦听链路，由于存在传播延迟，还是会出现冲突

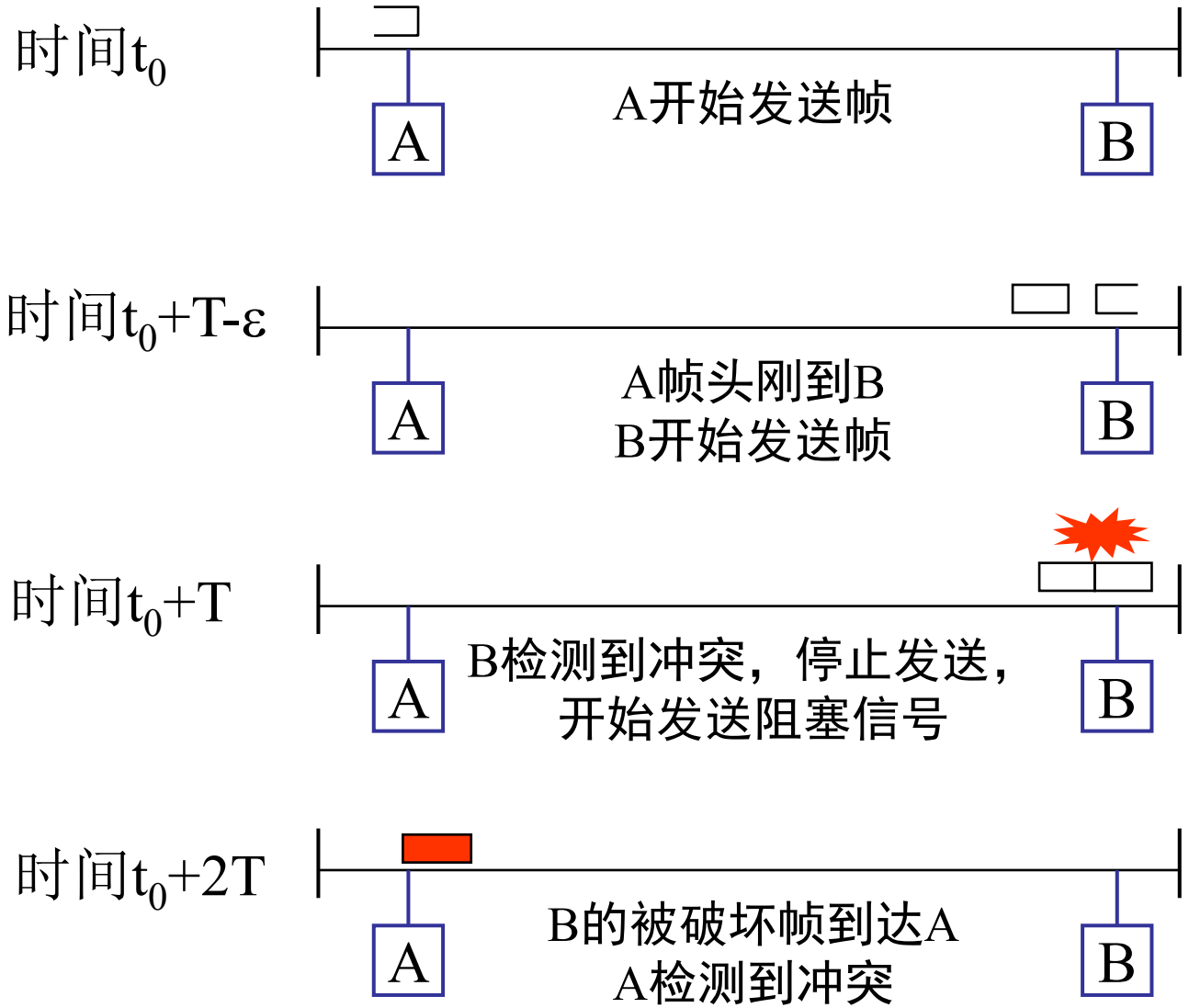
以太网CSMA/CD

- 对共享信道的以太网，IEEE 802.3提出了带有冲突检测的载波侦听多路访问（CSMA/CD）控制方法
- CSMA/CD以争用方式接入到共享信道，是一种动态的随机接入方式
  - 在传输的时候继续侦听链路是否发生冲突
  - 当几个站同时发送数据时，信道上的信号会互相叠加
  - 发现冲突立即停止发送，并发送阻塞信号，延迟一个随机时间之后再试图发送

CSMA/CD 协议工作流程



冲突检测窗口（争用期）



- A 需要单程传播时延的 2 倍时间才能检测到与 B 产生了冲突
- 以太网的端到端往返时延 2T 称为争用期，或冲突检测窗口
- 经过争用期还没有检测到冲突，可确定不会发生冲突

最短帧长

- 最短帧长:争用期内传输的比特数
- 最短帧长的计算公式:

$$L_{min}=2t\times R$$

$$t=D/V$$

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

10 Mbps 以太网帧长度

- 10 Mbps 以太网，在争用期内可发送 512 bit，即 64 字节(最短帧长)
  - 如果发生冲突，一定是在发送的前 64 字节之内
  - 以太网规定最短有效帧长为 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
- 10 Mbps 以太网覆盖范围
  - 以太网上最大的端到端单程时延必须小于争用期的一半（即 25.6 μs），信号在电缆上的传播速度是200m/μs，相当于以太网的最大端到端长度约为 5 km

二进制指数退避算法

- 发生冲突的站点停止发送数据，并发送完阻塞信号
- 为了降低再次冲突的概率，需要等待一个随机的时间，再试图传输
  - 基本退避时间取为争用期 2T
  - 对每一个帧，当第k次发生冲突时，从整数集合 $[1, \ldots, 2^k]$ 中随机地取出一个数，记为 r
    - 该帧重传所需的时延就是r 倍的基本退避时间

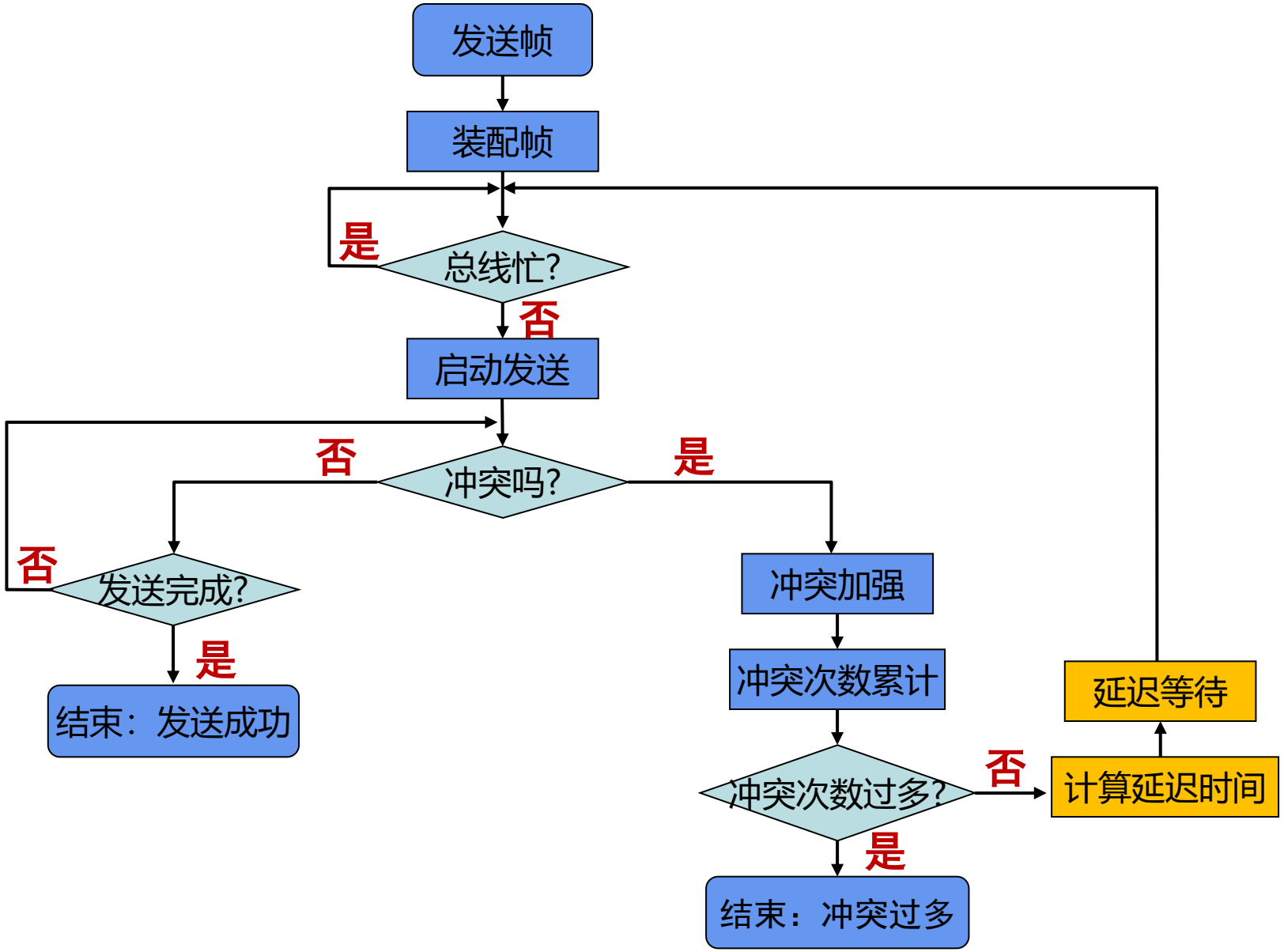
$$t = r \times 2T$$

  - 参数 k 按下面的公式计算：

$$k = \text{Min}(\text{重传次数}, 10)$$

    - 当 $k \leq 10$ 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，并向高层报错

以太网站点发送帧过程

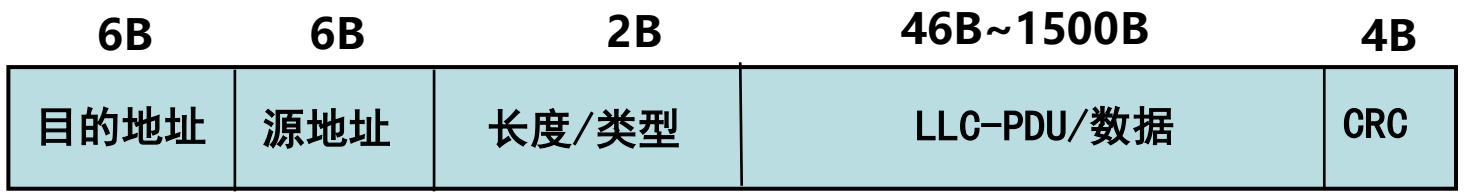

发送帧
装配帧
是
总线忙?
否
启动发送
否
冲突吗?
是
否
发送完成?
是
结束：发送成功
冲突加强
冲突次数累计
冲突次数过多?
否
是
结束：冲突过多
计算延迟时间
延迟等待

CSMA/CD 协议的适用场景

- CSMA/CD 协议适用于：
  - 两个以上站点共享信道、各站点之间以半双工方式进行通信的以太网
  - 如两台以上主机通过集线器互连
- CSMA/CD 协议不适用于：
  - 两点之间点对点、不存在冲突的全双工通信以太网
  - 如主机间通过以太交换机互连
- 无共享信道情况的以太网不需要CSMA/CD

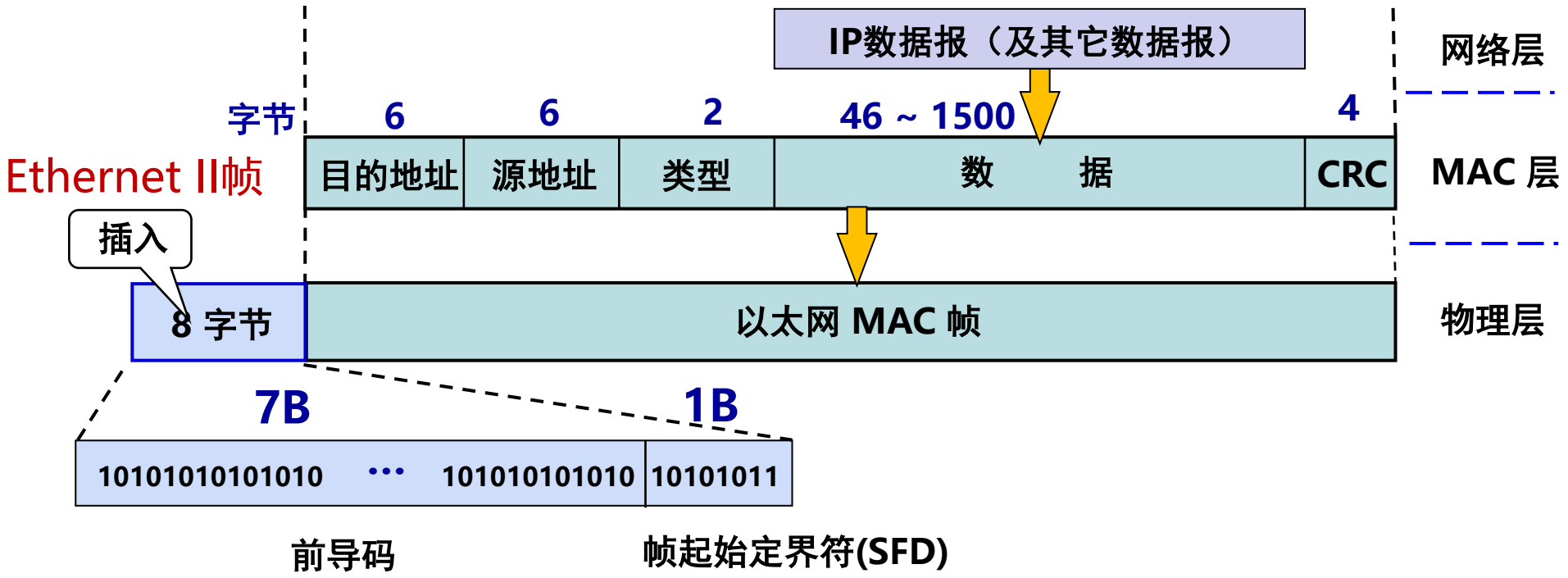
以太网帧结构

- 以太网 MAC 帧格式有两种标准，通过长度/类型字段值来区分
- 最常用的 MAC 帧是 Ethernet II 的格式



  - 长度/类型>0X0600，Ethernet II 标准 (也称 DIX V2 标准)
  - 长度/类型≤ 0X0600，IEEE 的 802.3 标准

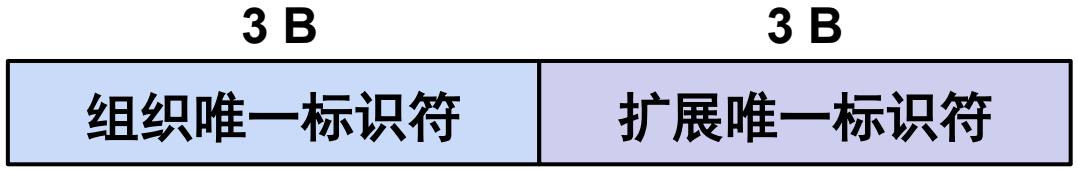
Ethernet II 帧



- 在帧的前面插入（硬件生成）的 8 字节，包含两个字段
  - 第一个字段是 7 个字节的前导码，用来迅速实现 MAC 帧的比特同步
  - 第二个字段 是1 个字节的帧起始定界符，表示后面的信息就是 MAC 帧

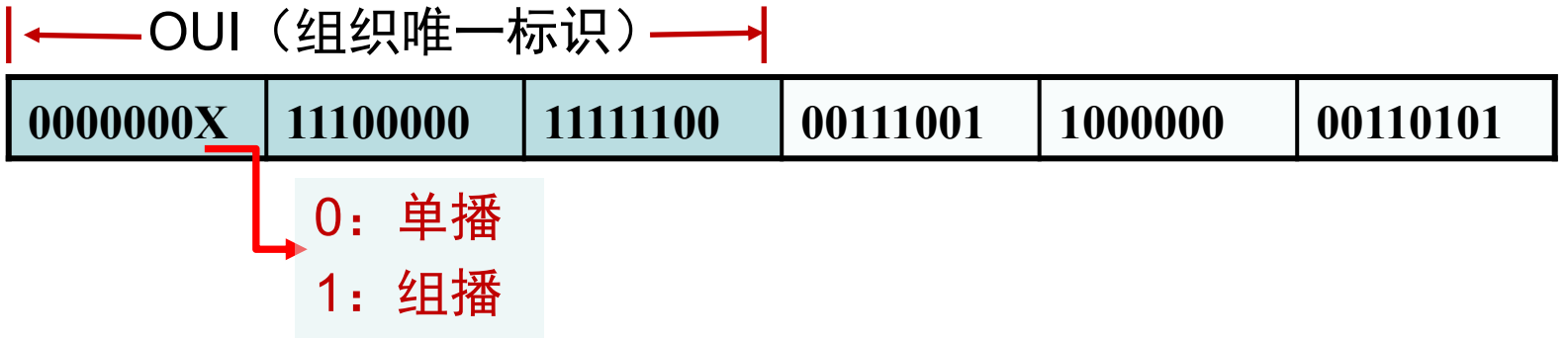
Ethernet II帧 - MAC地址字段（1）

- 目的地址和源地址字段用于标识发送方地址和接收方地址
  - 这个地址是硬件地址，又称为物理地址，或 MAC 地址
  - IEEE 802 标准规定的 MAC 地址长48 比特（6字节）
    - 组织唯一标识符：前三个字节 (高位 24 位)，由IEEE 的注册管理机构 RA 负责向厂家分配
    - 扩展唯一标识符：后三个字节 (低位 24 位)，由厂家自行指派，必须保证无重复地址



Ethernet II帧 - MAC地址字段（2）

- MAC地址分三种：单播地址、组播地址和广播地址
  - IEEE 规定地址字段的第一字节的最低位用于标识单播还是组播地址



    - 最高字节的最低位 = 0 时，是单播地址，只有一个站点拥有该地址
    - 最高字节的最低位 = 1 时，是组播地址，有一组站点拥有该地址
  - 48 位都为 1 时，为广播地址
  - 组播地址和广播地址只能用作目的地址

Ethernet II帧 - 类型字段

- 类型字段指明上层协议，以便接收方把帧里封装的数据交给正确的上层协议
  - 类型字段值 0x0800，表示这个帧内封装的是 IPv4 数据报
  - 类型字段值 0x0806，表示这个帧内封装的是ARP报文
  - 类型字段值 0x8100，表示这个帧内封装的是IEEE 802.1Q帧
  - 类型字段值 0x86DD，表示这个帧内封装的是IPv6数据报

Ethernet II帧 - 数据字段、帧校验字段

- 数据字段里封装的是上层协议的PDU
  - 数据字段的最小长度为46字节(最小帧长64B)
  - 当数据字段的长度小于 46 字节时要进行填充，以满足最小帧长要求
- 帧校验字段包含了差错检测信息，在这种情况下是CRC-32

IEEE 802.3帧格式

- IEEE 802.3标准MAC 帧规定：
  - 第三个字段是“LLC-PDU长度 ”，表示数据字段的长度
  - 数据字段必须装入逻辑链路控制子层的 LLC 帧，且长度≤ 1500字节
- IEEE 802.3标准的以太网在产业中已经很少使用了

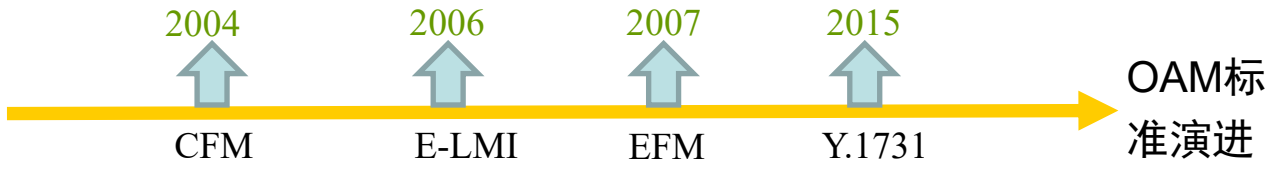
无效的 MAC 帧

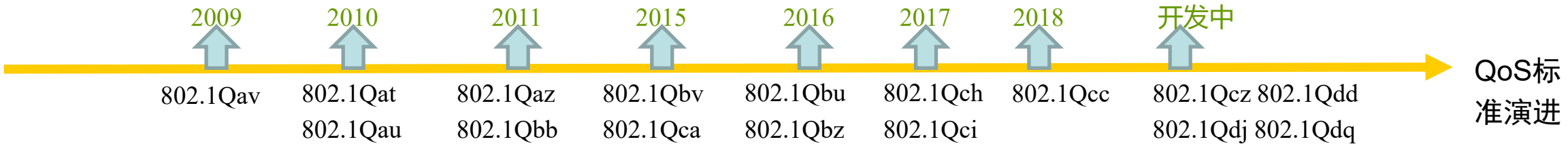
- 有效的 MAC 帧长度为 64 ~ 1518 字节之间
- 如下几种情形属于无效的 MAC 帧：
  - 数据字段的长度与长度字段的值不一致
  - 帧的长度不是整数个字节
  - 用收到的帧校验字段值校验有差错
  - 数据字段的长度不在 46 ~ 1500 字节之间
- 以太网对于检查出的无效 MAC 帧简单丢弃，无需重传

以太网络标准演进

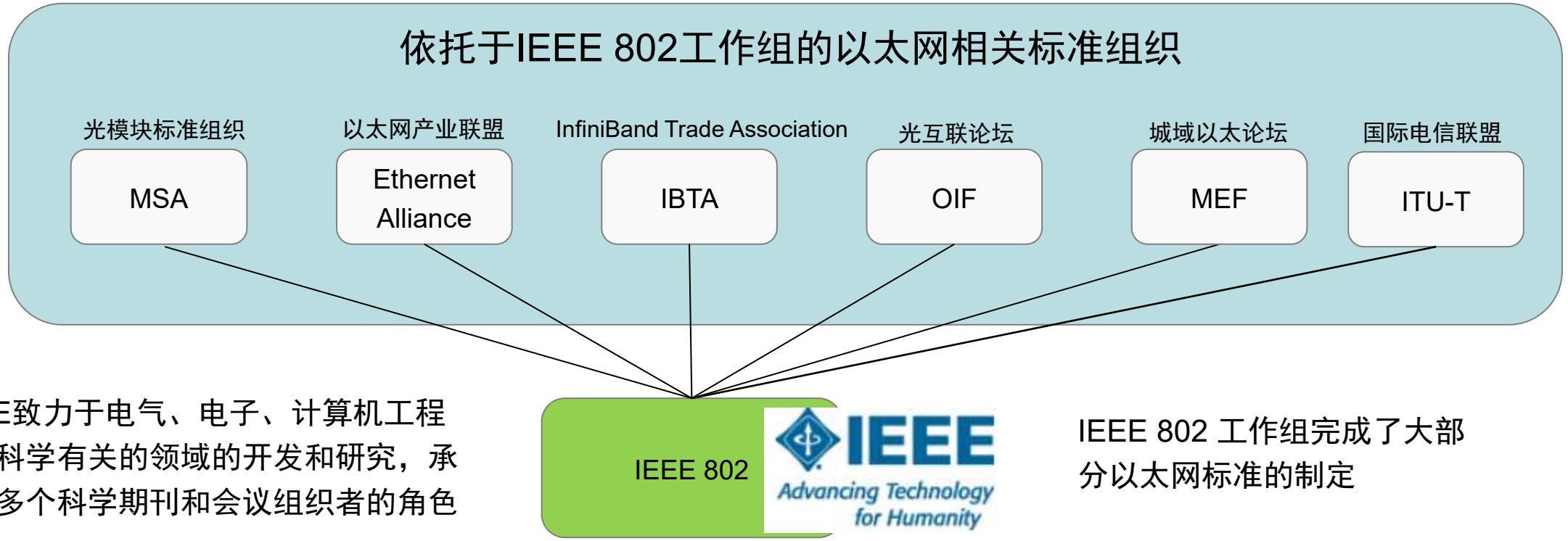
- 超宽、QoS保证、可管理维护三方并进

GE千兆到桌面，10GE端口广泛应用
100GE/200GE/400GE 端口也成熟使用2.5G/5G/25G/50G 分支助力新一代高性能服务器接入





以太网各标准实体：开放、竞争与合作



IEEE致力于电气、电子、计算机工程和与科学有关的领域的开发和研究，承担着多个科学期刊和会议组织者的角色

IEEE 802 工作组完成了大部分以太网标准的制定

IEEE 802 标准衍生的以太网产业系统已经成为互联网以及电信网络最大的生态系统，推动产业界技术的不断创新，满足新的应用场景，技术不断成熟，成本持续降低

OIF：定义光电互联互通标准

- OIF 成立于1998年，通过定义公共协议和组件技术促进光网络产品的互联互通
- 同时， OIF定义测试标准和互操作测试集促进不同厂家产品的互联互通
- OIF与别的标准组织有广泛的合作关系，例如ITU, IEEE802.3, ONF, IBT, TIA和IETF等

ARCH & Signaling WG	Carrier WG	Interoperability WG	OAM WG	Physical & Link WG
从运营商的需求开发网络架构和信令	开发光网络产品功能需求和指南	定义测试标准和测试方法，开发概念原型	聚焦于部署问题	开发物理和数据链路层接口协议

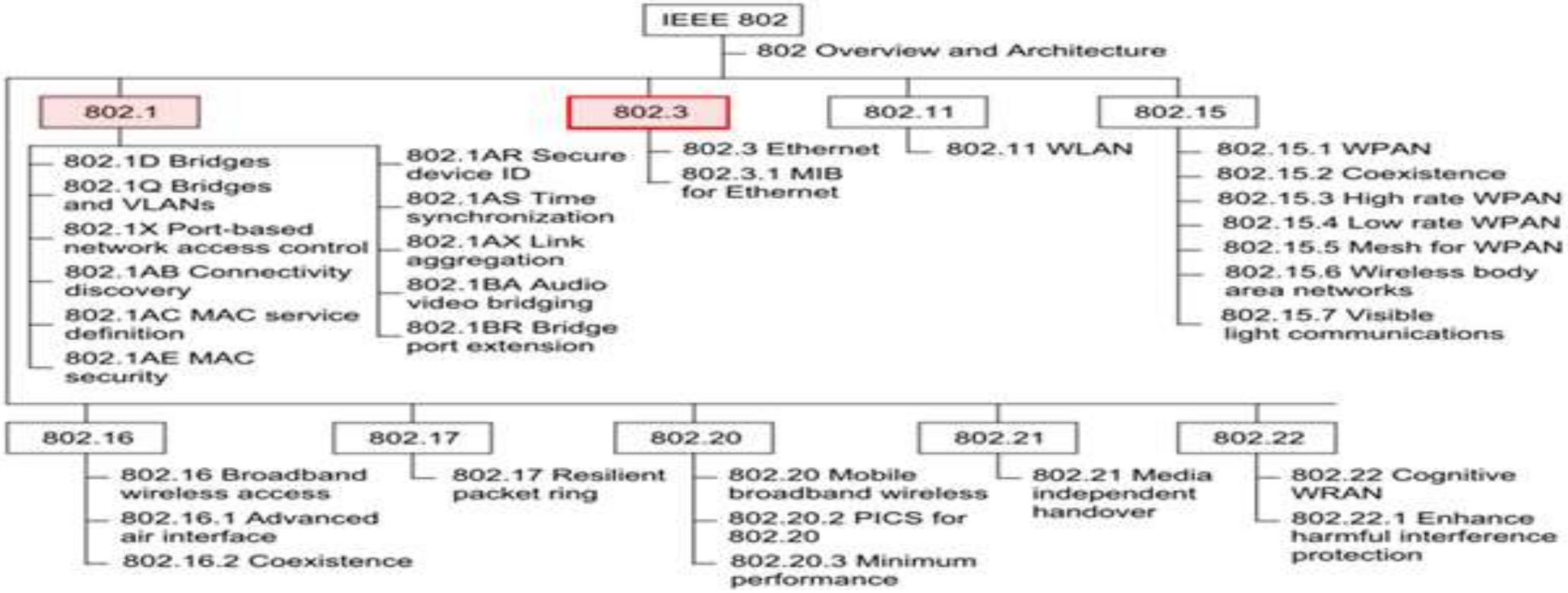
- 主要成果：
  - CEI（common electrical I/O）协议（3.125G，6，11，25-28，56Gbit/s SERDES interface）
  - Flex Ethernet

MEF（Metro Ethernet Forum）：定义城域以太业务及运营框架

- MEF 是一个包括network运营商、cloud运营商和技术提供商的非盈利联盟，聚焦在城域范围内如何开展有 QoS保证的、可运营的、可管理的以太网业务，包括：
  - 城域以太网的架构和城域以太网提供的业务
  - 从用户的角度定义了城域以太网的业务框架，并明确了业务类型（E-Line和E-Lan）
- 城域以太网的保护和QoS：
  - 在保护和QoS方面针对城域以太网提出了保护模式、机制和QoS功能框架，即定义了执行和维护SLA所需的QoS功能和特性
- 城域以太网的管理
  - 在管理方面提出了城域以太网的网络管理接口(EMS-NMS)，并从网络分层、子网划分、子网拓扑、网络连接四个方面对EMS-NMS接口进行规范

IEEE 802：定义LAN架构和不同介质网络技术

- IEEE 802.1：网络架构、应用场景下的相关标准。如：VLAN 802.1Q，生成树 802.1D/W/S，以太安全体系 802.1x/1AE/1AR，TSN 802.1AS等
- IEEE 802.3：ETH网架构，包含物理层和MAC层，如：速率、介质等

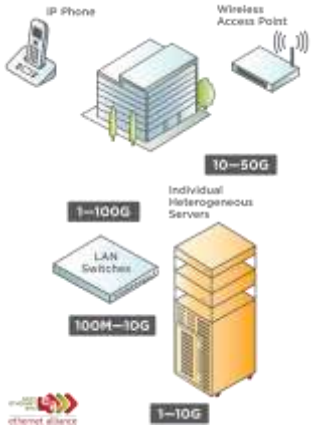


以太网：获得广泛的行业应用

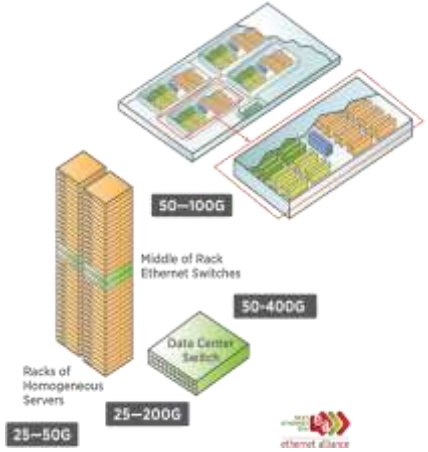
- 以太网速率高、成本低、产业链成熟、标准开放统一，已在家庭网络、园区/办公网络、运营商网络、数据中心、工业物联网等领域获得广泛应用

家庭网络

园区网络

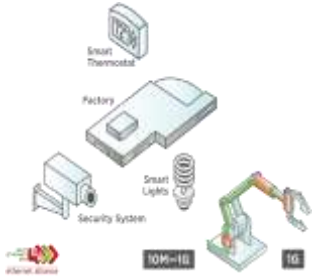


DC网络

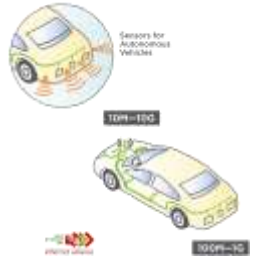


运营商网络

工业物联网



车载以太网



以太网络各速率应用场景

速率	场景说明
100M，GE	一般应用在桌面办公领域
1000BASE-T1	单对双绞线1000M以太网络主要应用在车载领域
2.5GE，5GE，25GE	一般应用在服务器领域，或者高端计算领域
40GE，50GE，100GE	主要有3个主要场景： 1、超高端服务器或者计算领域；2、城域网络或传输；3、数据中心交换机
200GE，400GE	主要有2个场景： 1、城域网络或者传输网络；2、核心交换机
800GE，TE	1、数据中心核心交换机；2、核心网络

无线局域网概述

- 无线局域网 WLAN (Wireless Local Area Network) 是采用无线通信技术覆盖局部地理范围的局域网络
- WLAN 可分为两大类：
  - 有固定基站的 WLAN
  - 无固定基站的 WLAN
- 802.11x 是 IEEE 802 标准组织为 WLAN 制定的媒体访问控制层和物理层协议簇，在有固定基站下使用 900 MHz、2.4G、3.6G、5G、60G 频段的WLAN的系列化标准
- Wi-Fi 联盟将 Wi-Fi 定义为基于802.11协议，可以通过无线接入的产品
- Wi-Fi 和 WLAN 意义是相同的，其名称可以互用

Wi-Fi 技术与标准的发展历程（代际关系）

	802.11n	802.11ac	802.11ad
Throughput	600 Mbps	3.2 Gbps	Up to 7 Gbps
Coverage	Home, 70 m	Home, 30 m	Room, <5m
Freq. Band	2.4/5 GHz	5 GHz	2.4/5/60 GHz
Antennas	4 x 4 MIMO	8 x 8 MIMO	>10 x 10 MIMO
Applications	Data, Video	Video	Uncompressed Video

11n、11ac、11ad三代标准参数对比

802.11
- 2Mbps
- 2.4GHz

DSSS

802.11b
- 11Mbps
- 2.4GHz

OFDM

802.11g-2.4G
802.11a-5GHz
- 54Mbps

802.11n
- 65Mbps~600Mbps
- 2.4G/5GHz

MIMO(4x4 max)
20/40MHz频宽
Block ACK
64QAM

802.11ac
- 433Mbps~3.2Gbps
- 5GHz

MIMO(8x8 max)
20/40/80/160M频宽，MU-MIMO，256QAM

2GHz频宽

802.11ad
- 7Gbps
- 60GHz

OFDMA

802.11ax
- 10Gbps
- 5GHz

1st Generation —2年→ 2nd Generation —3年→ 3rd Generation —5年→ 4th Generation —5年→ 5th Generation —5年→ 6th Generation

1997　1999　2002　2007　2012　2017

- Wi-Fi 技术和产品一直在发展中，从802.11演进到802.11ax（Wi-Fi6）
- 802.11ad 引入60G频段，提升短距带宽；802.11ac 演进到802.11ax，带宽提升到10Gbps

无线局域网标准

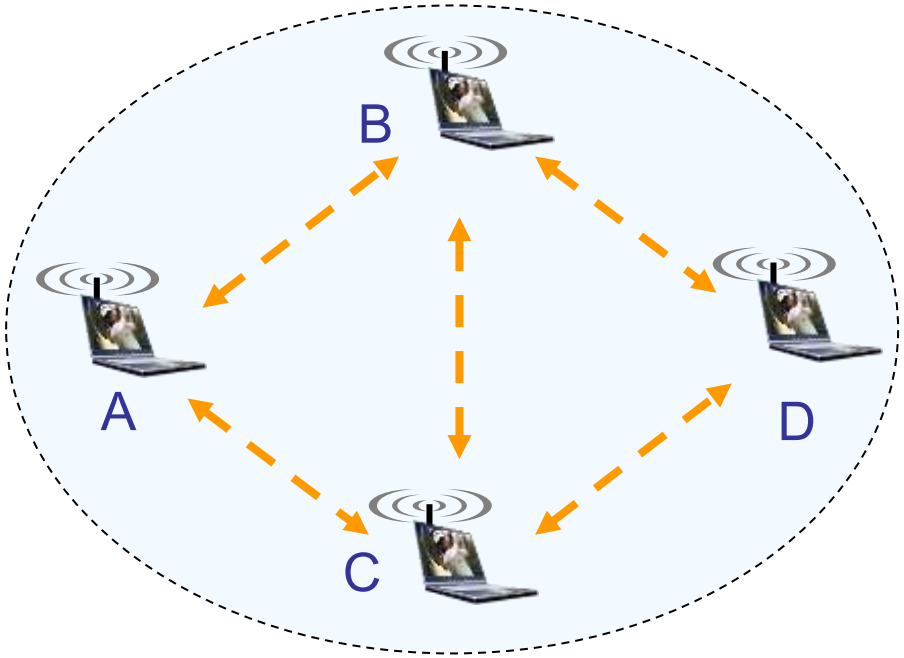
- IEEE 802.11标准定义了3种物理层介质：
  - 跳频扩展频谱FHSS
  - 直接序列扩展频谱DSSS
  - 红外线IR
- 正交频分复用OFDM用于802.11a/g
- 目前，已经标准化的有：
  - 802.11b(1999)
  - 802.11a(1999)
  - 802.11g(2003)
  - 802.11n(2009)
- 很多无线局域网适配器做成双模（802.11a/g)或三模(802.11a/b/g或802.11b/g/n)的

各种标准比较

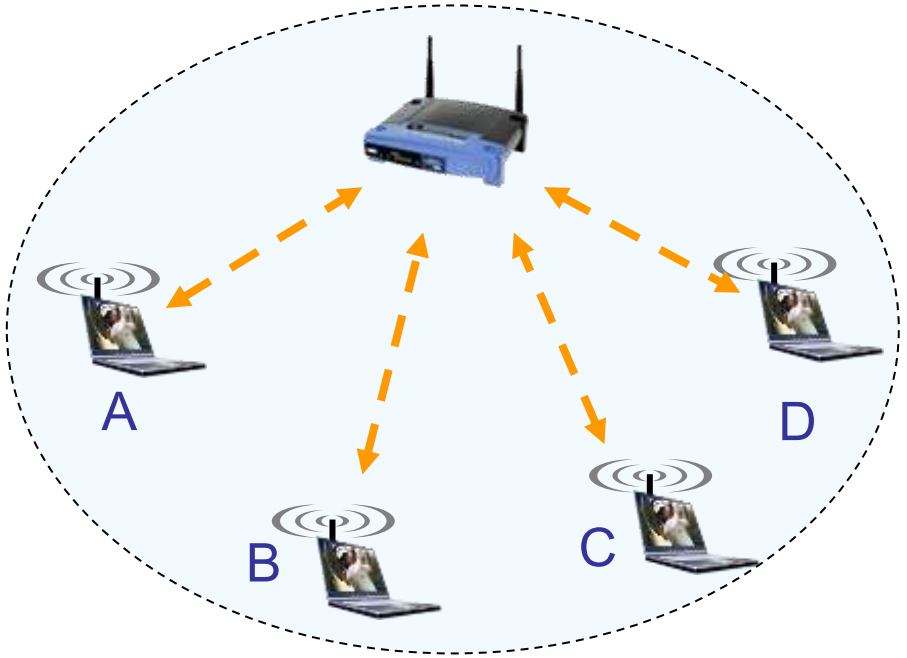
标准	802.11b	802.11g	802.11a
MAC协议	CSMA/CA		
工作频率	2.4GHz		5GHz
物理编码	DSSS	DSSS 或OFDM	OFDM (正交频分复用)
最高速率	11Mbps	54Mbps	54Mbps

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

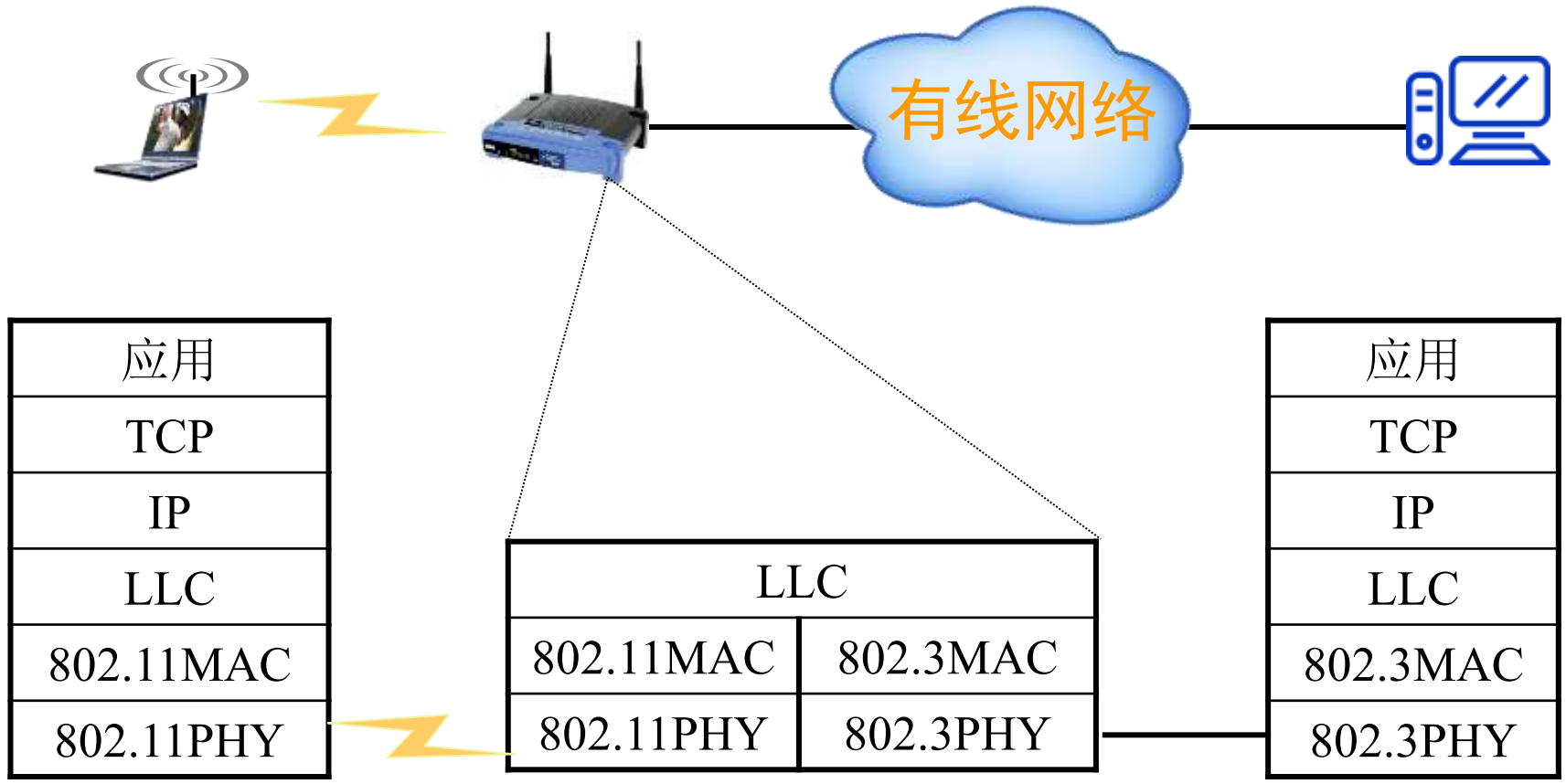


独立BSS

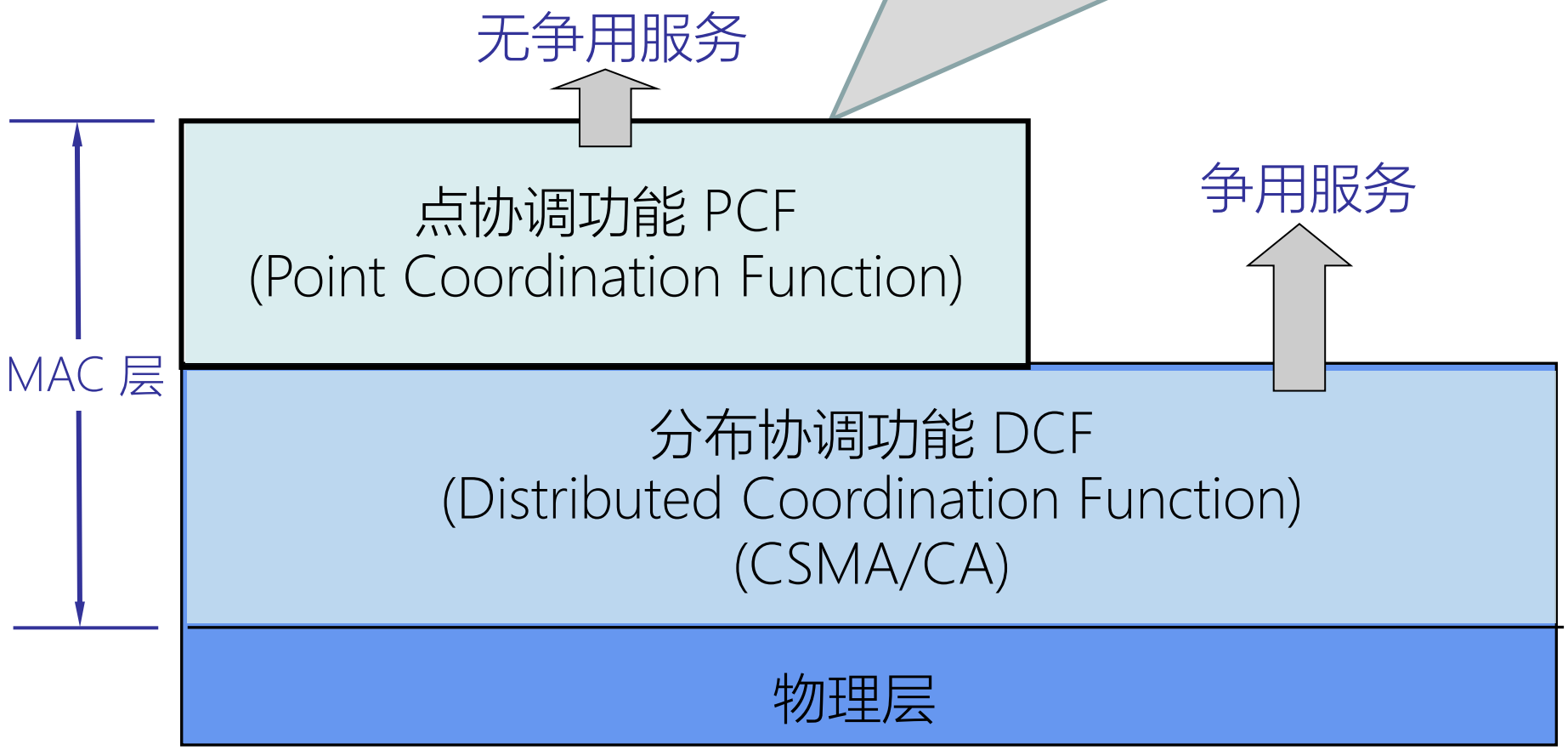


有AP的BSS

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



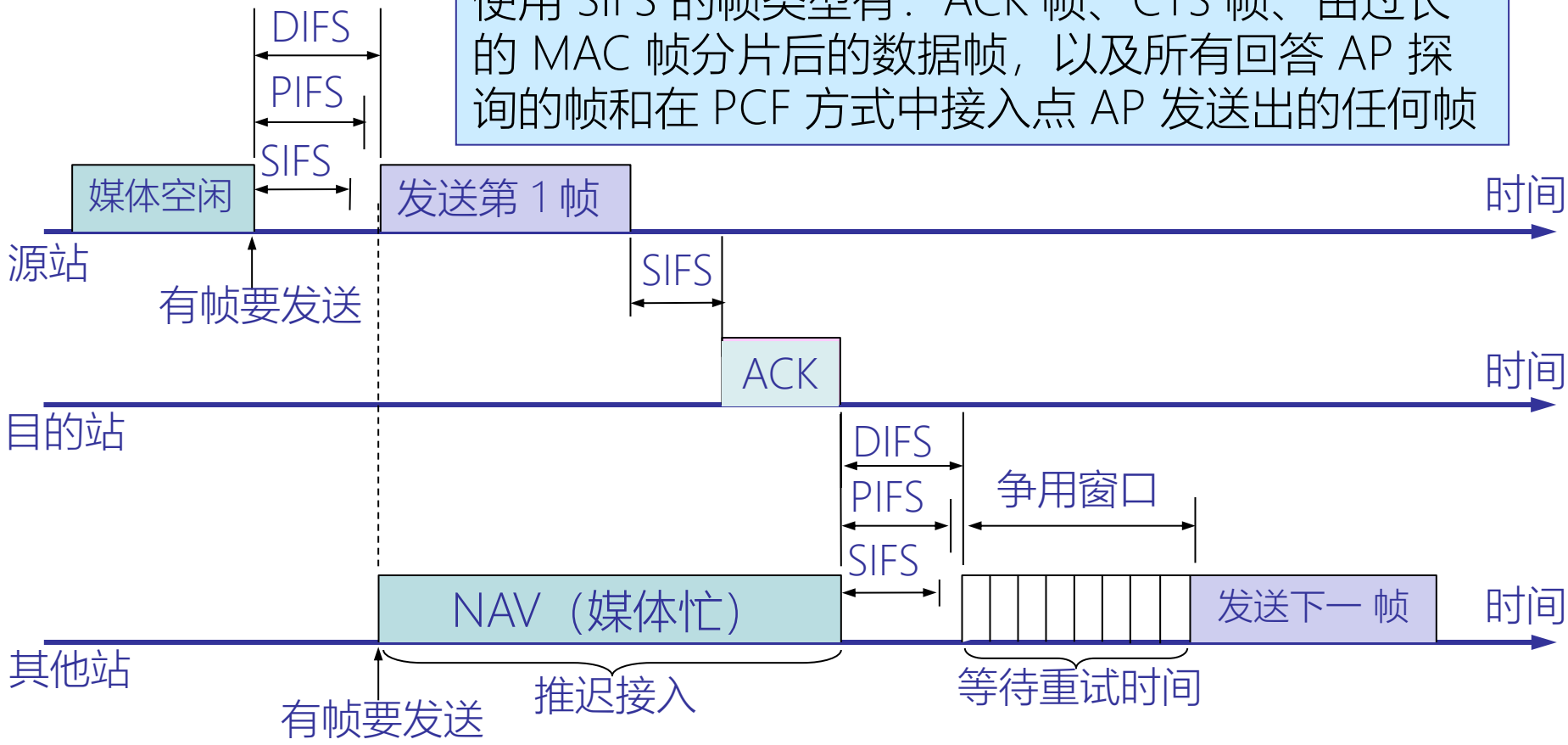
帧间间隔 IFS

- 所有站点在完成发送后，必须再等待一段很短的时间（继续侦听）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。这样就减少了发生冲突的机会

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

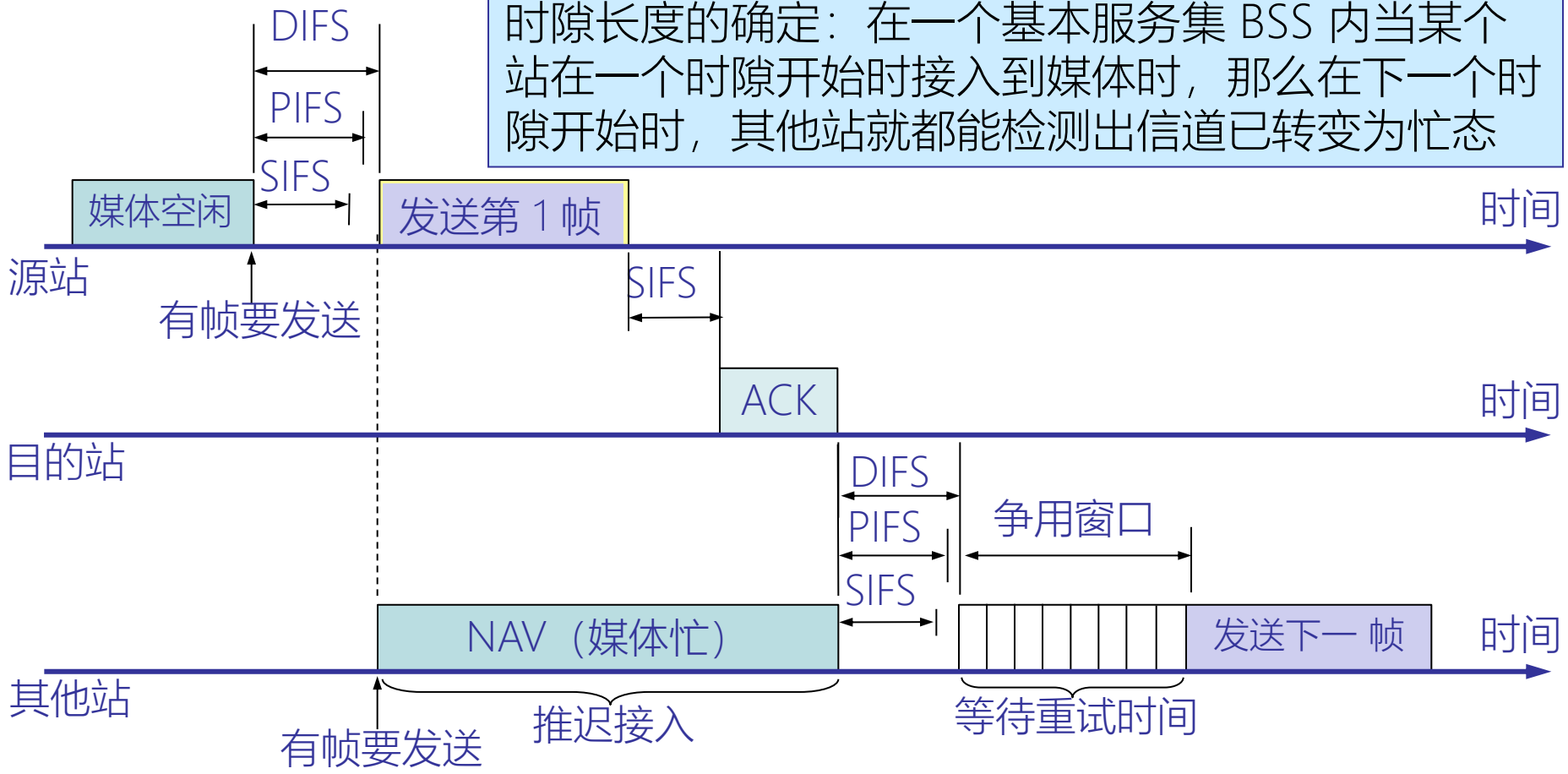
使用 SIFS 的帧类型有：ACK 帧、CTS 帧、由过长的 MAC 帧分片后的数据帧，以及所有回答 AP 探询的帧和在 PCF 方式中接入点 AP 发送出的任何帧



三种帧间间隔

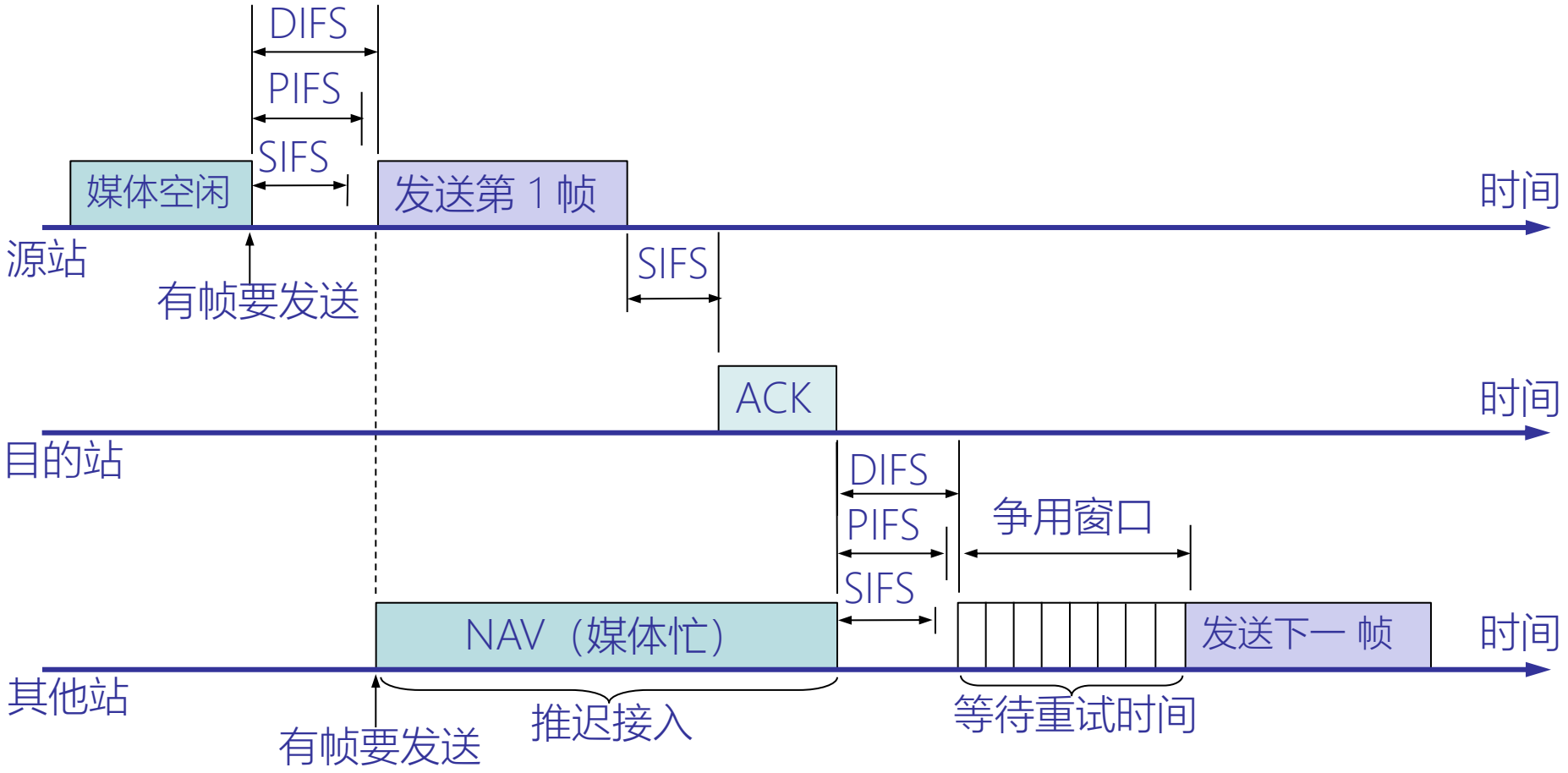
PIFS，即点协调功能帧间间隔，它比 SIFS 长，是为了在开始使用 PCF 方式时（在 PCF 方式下使用，没有争用）优先获得接入到媒体中。PIFS 的长度是 SIFS 加一个时隙(slot)长度

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

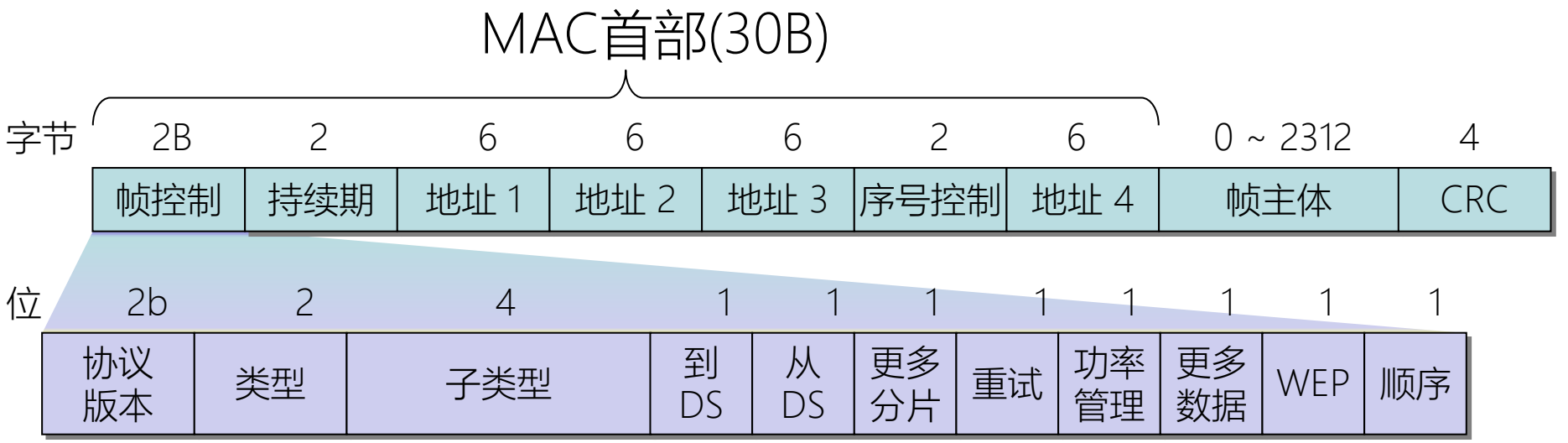


三种帧间间隔

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



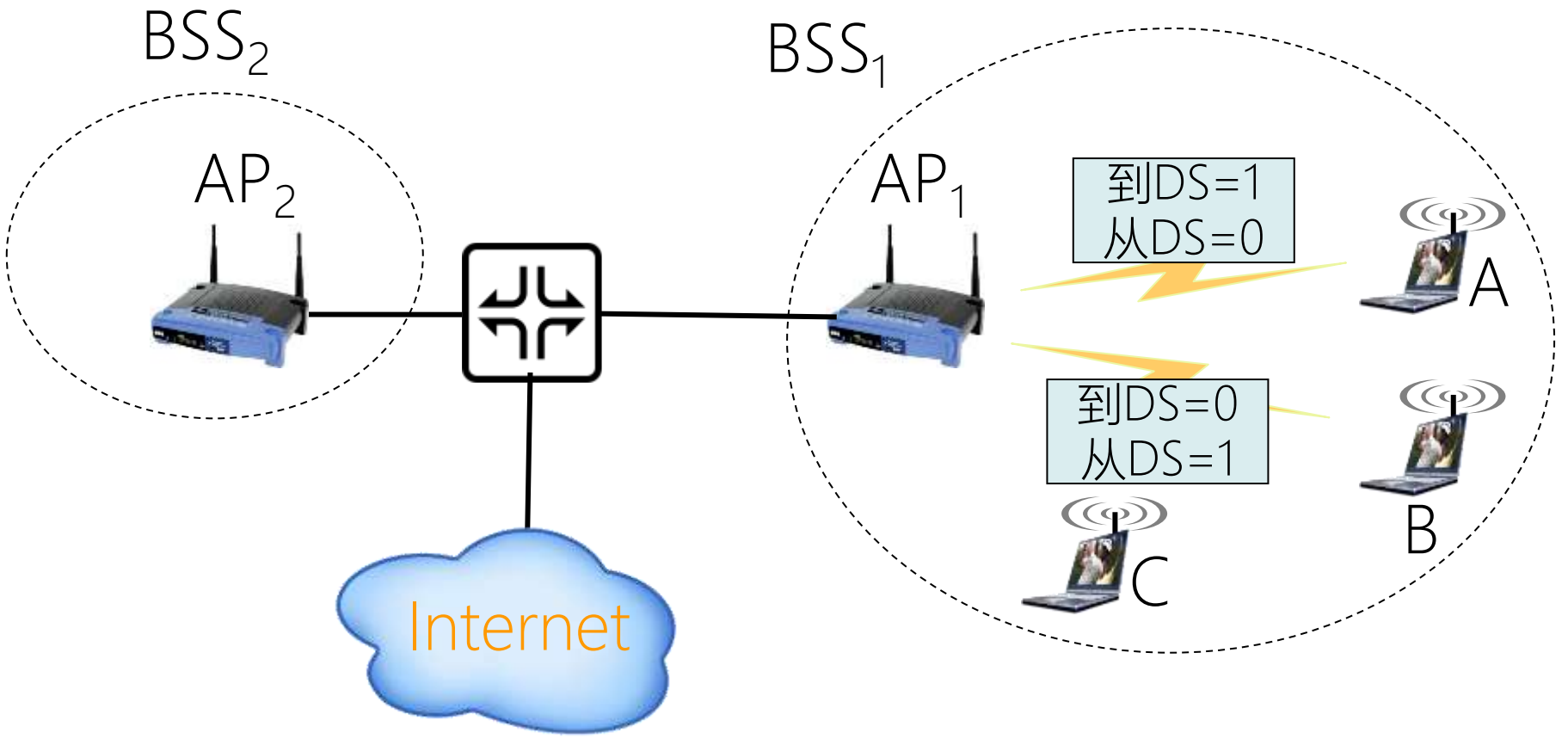
802.11协议族MAC帧结构



- 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

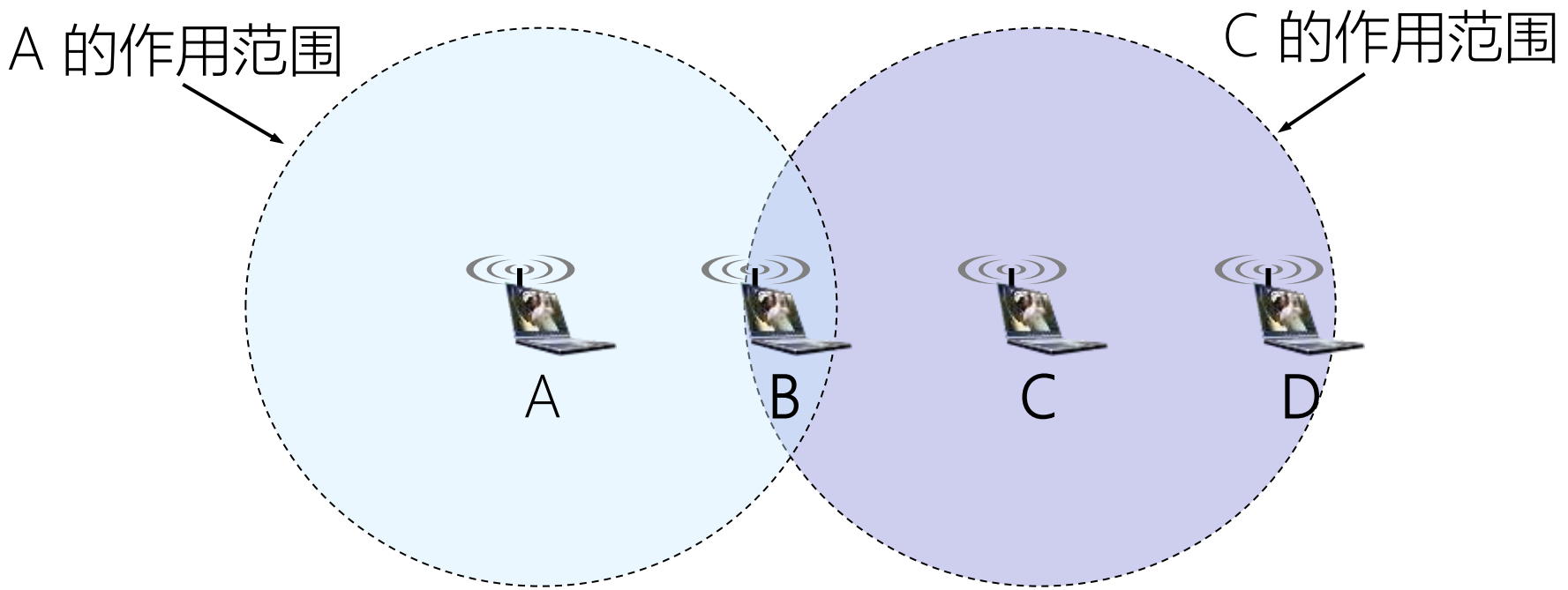


到DS	从DS	地址1	地址2	地址3	地址4
0	1	目的地址	AP地址	源地址	--
1	0	AP地址	源地址	目的地址	--

为何不采用CSMA/CD

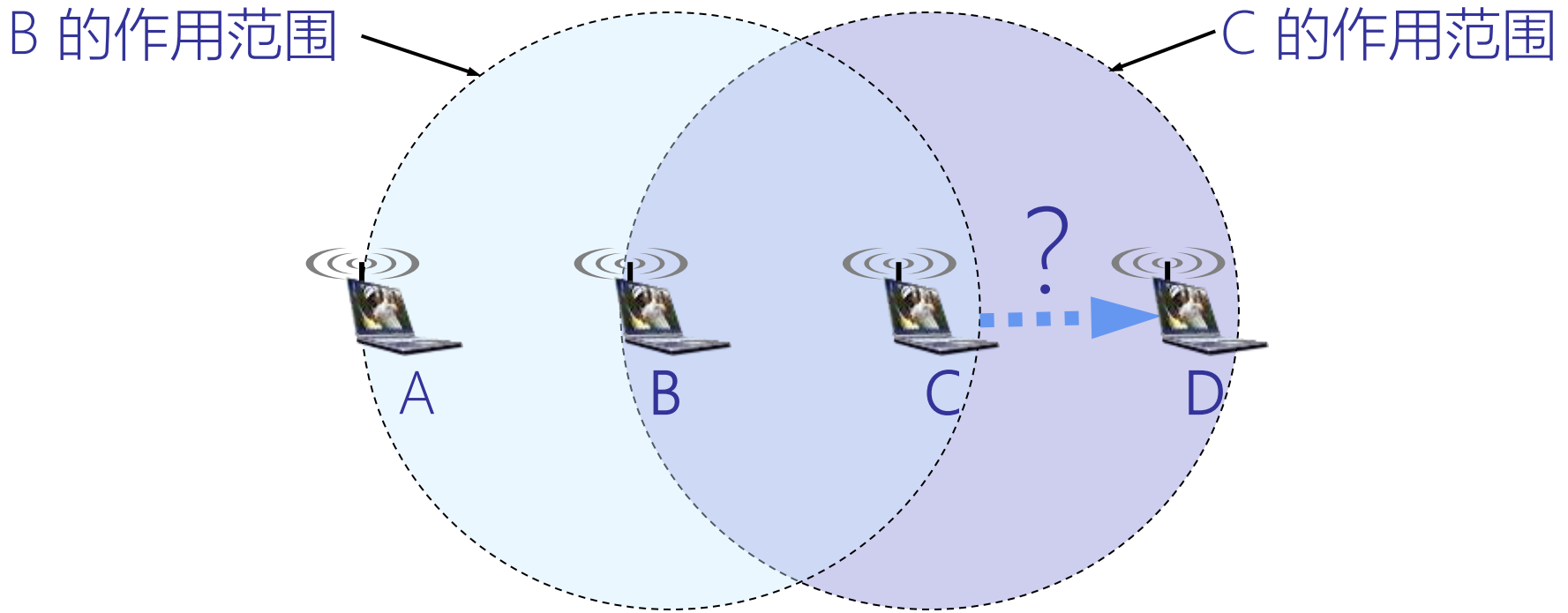
- 要检测冲突，设备必须在发送数据时能够接收数据，对无线设备来说难以实现
- 冲突检测方式不同
  - CSMA/CD：检测电压的变化
  - CSMA/CA：采用能量检测、载波检测和能量载波混合检测
- 即使能够实现冲突检测的功能，并且当发送数据时检测到信道是空闲的，在接收端仍然有可能发生冲突—隐蔽站问题

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，结果发生冲突
- 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

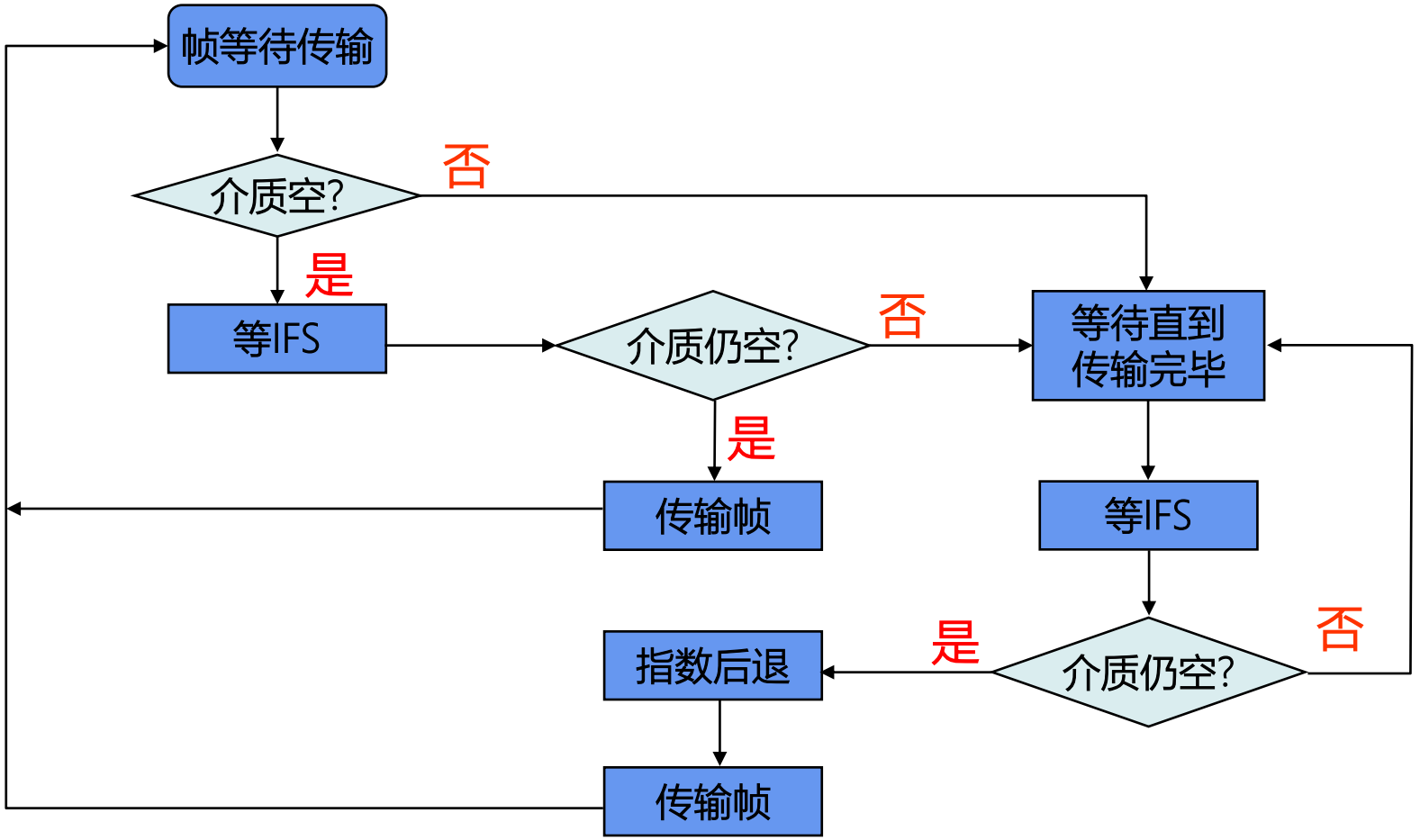


- B 向 A 发送数据，而 C 又想和 D 通信，C 检测到媒体上有信号，于是就不敢向 D 发送数据
- B 向 A 发送数据并不影响 C 向 D 发送数据，这就是暴露站问题(exposed station problem)
- 不影响协议的正常工作，但导致信道利用率下降

CSMA/CA协议

- 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

CSMA/CA工作流程



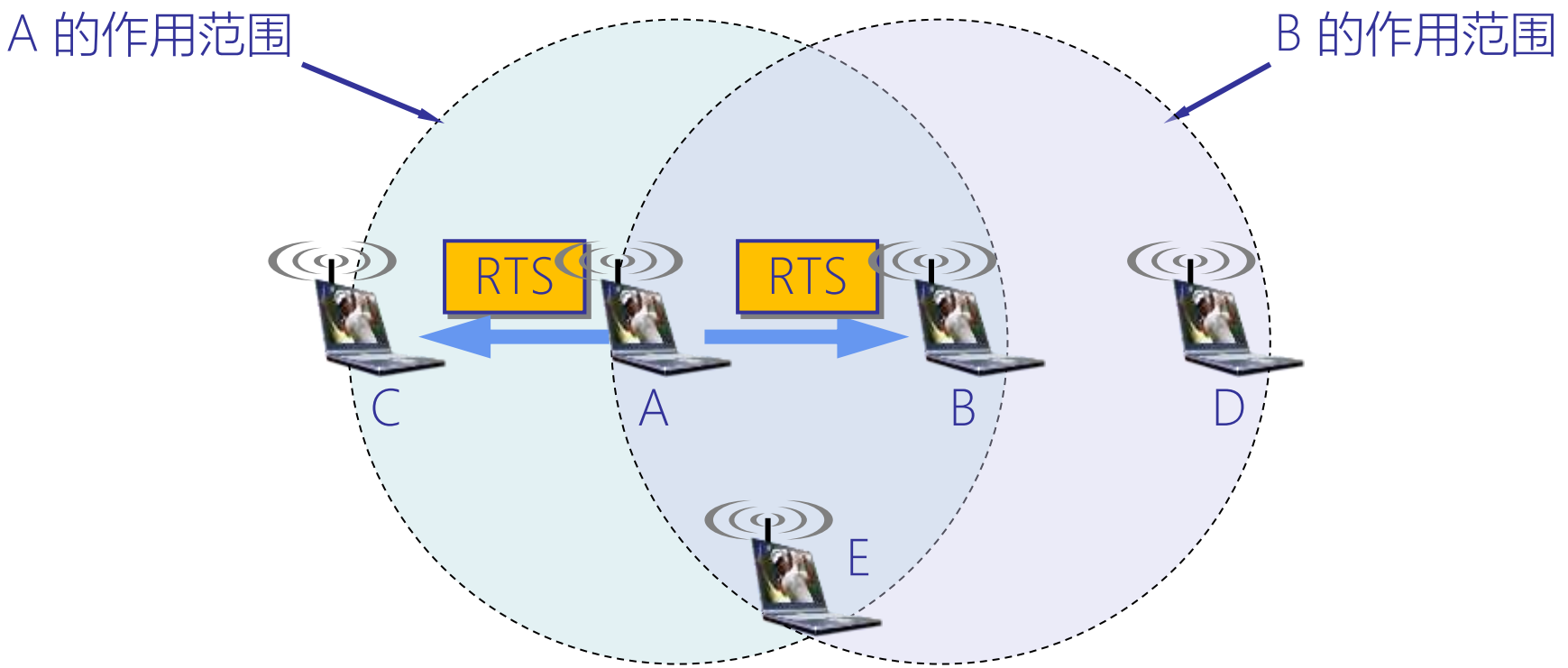
二进制指数退避算法

- 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, …, $2^{2+i}-1$} 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，采用了以下机制来实现：
  - 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

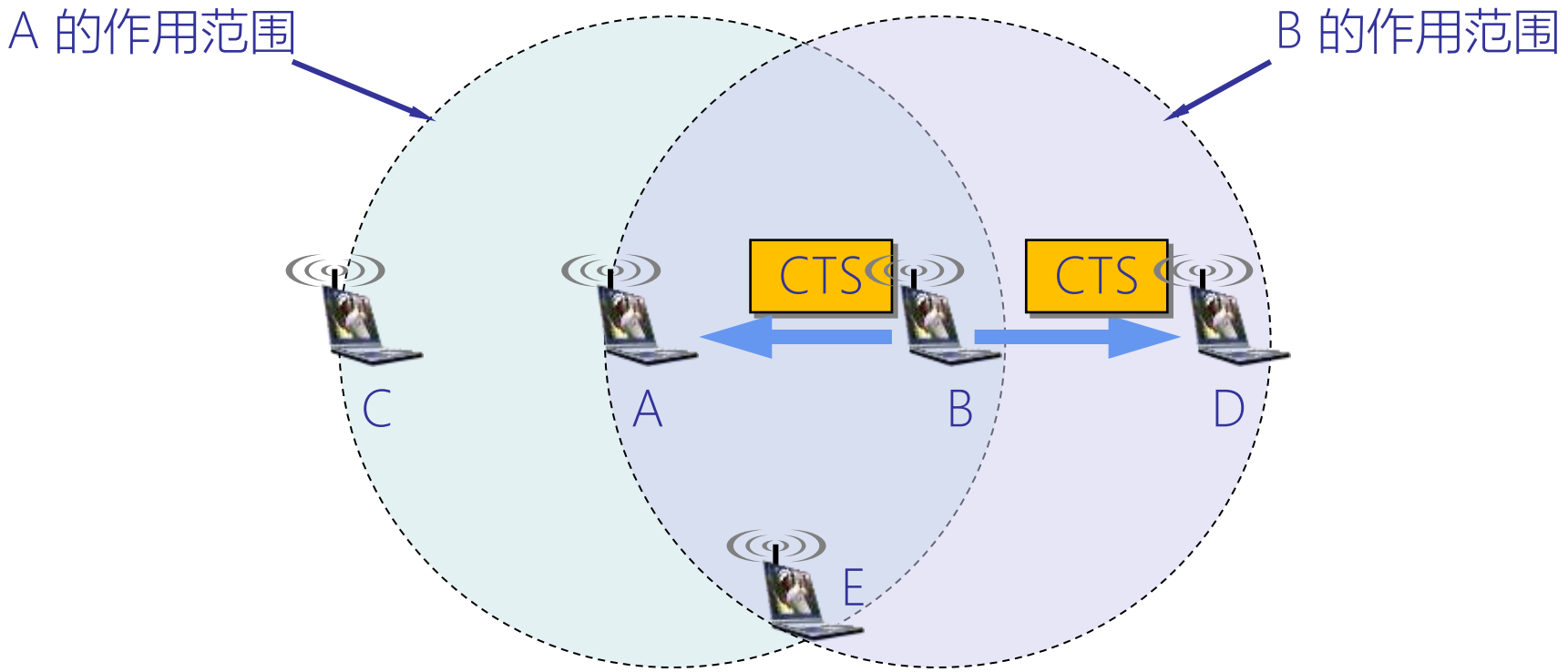
对信道进行预约(RTS)



- 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（包括相应的应答帧）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，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）

Wi-Fi通信的信道特点

- 共享开放的无线介质（频谱资源），没有实体物理层的防护，数据是广播式发送的
- 无线电波传输的不可靠性：反射、穿墙等 —— 因此802.11 MAC层通过ACK确认机制，牺牲部分流量以保证可靠性
- Wi-Fi传输基本原则：在信道范围内，同一时刻只有一台设备可以发信号，其他设备都要等待，只有等到信道空闲时才能发送数据
- 发送者和接收者处于同一频率、不能同时传输 —— 是半双工通信
- 信道空闲的检测方法：CSMA/CA采用能量检测（ED）、载波检测（CS）以及能量载波混合检测等手段

无线个域网WPAN

- 在个人工作区域，把属于个人的电子设备，通过无线技术连接起来组成的网络
  - 网络范围有限，约为10米
  - 使用无线网络连接，但不需要接入点AP
  - WPAN与PAN不同：PAN不一定使用无线连接
  - WPAN与WLAN不同：
    - WPAN以个人为中心，低功率、小范围、低速率、低价格
    - WLAN同时为多用户提供服务，大功率、中等范围、高速率
  - WPAN没有LLC子层，MAC层直接面向高层

蓝牙

- 蓝牙协议是MAC层直接面向应用层的WPAN协议
  - 蓝牙的特点：
    - 同时支持传输语音和数据，抗干扰能力强
    - 低成本，低功耗
    - 蓝牙系统的基本单元是一个微网，包含一个主结点和多个活跃的从节点。多个微网可以通过一个桥结点相连
  - 蓝牙协议体系结构分三层：
    - 物理层
    - 数据链路层
    - 应用层

ZigBee协议

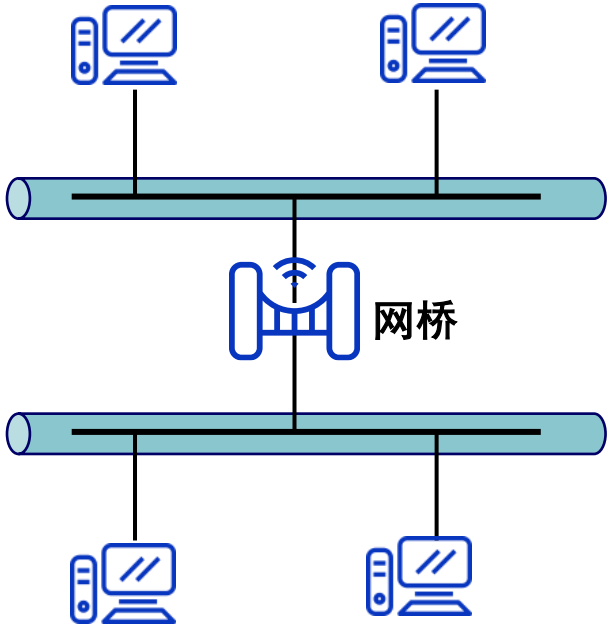
- ZigBee技术主要用于各种电子设备间的无线通信
- ZigBee主要特点：
  - 通信距离短，10-80米
  - 传输数据速率低、成本低
  - 功耗低:工作时，信号的收发时间很短，非工作时，处于休眠状态
  - 网络容量大，一个ZigBee网络最多容纳255个结点，其中一个是主设备，其余是从设备
- ZigBee协议栈分四层：
  - 物理层
  - MAC层：负责无线信道的接入
  - 网络层：负责拓扑结构建立和维护
  - 应用层：由ZigBee设备对象、应用支持子层以及应用框架组成

4.4 数据链路层网络互连

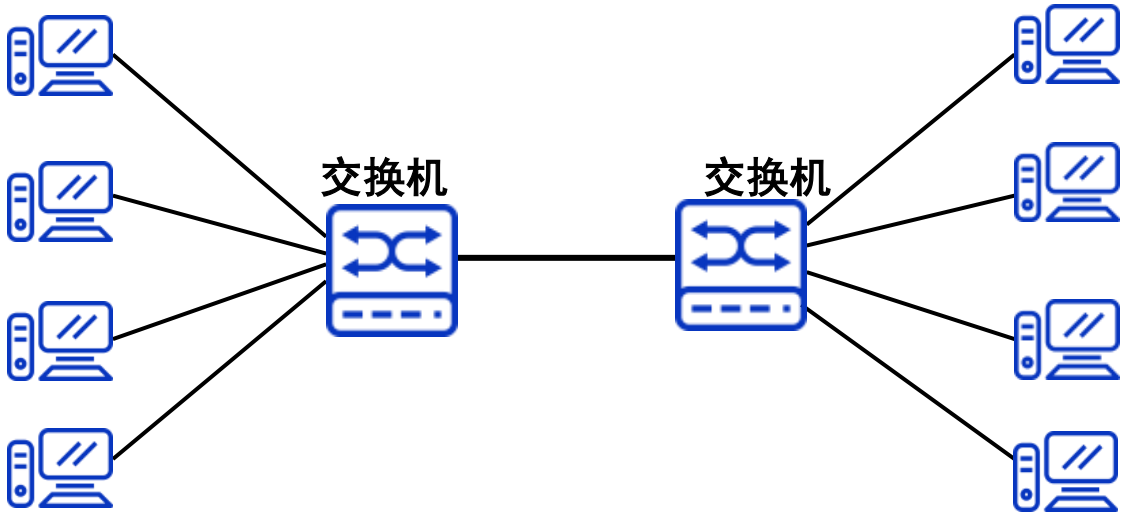
- 网桥
- 网桥路由算法
- 二层交换机
- 虚拟局域网

数据链路层网络互连

- 数据链路层上的互连设备：网桥，交换机



（1）用网桥扩展局域网



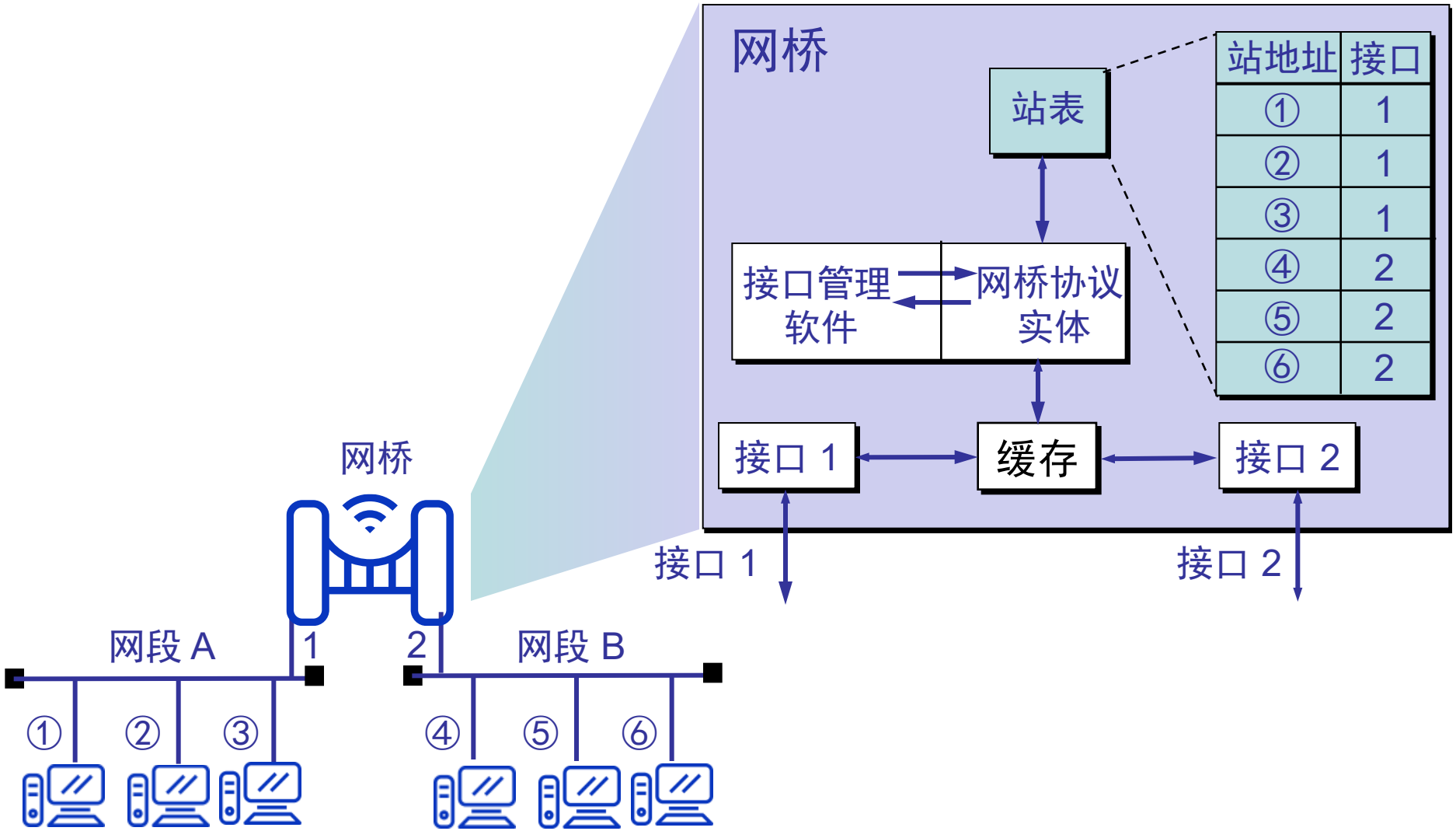
（2）用以太交换机扩展局域网

- 早期使用网桥来扩展以太网，现在已经不用了
- 现在使用以太网交换机来扩展以太网，互连更多的主机

网桥

- 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
- 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

网桥的内部结构

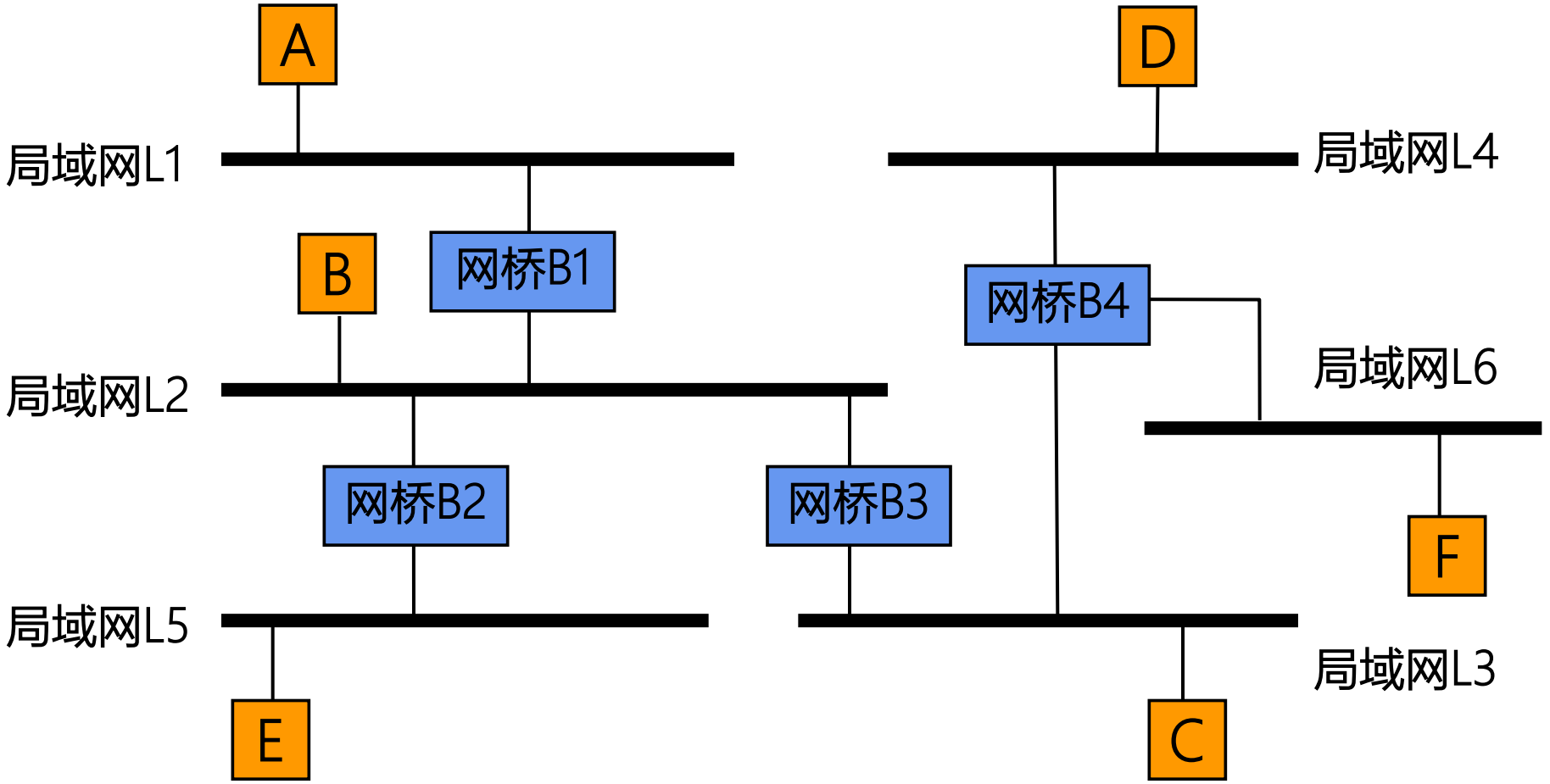

网桥
站表
站地址
接口
①
1
②
1
③
1
④
2
⑤
2
⑥
2
接口管理
软件
网桥协议
实体
接口 1
缓存
接口 2
接口 1
接口 2
网桥
网段 A
1
2
网段 B
①
②
③
④
⑤
⑥

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 子层和不同速率的局域网
- 网桥的功能：
  - 帧路由
  - 错误检测和帧格式转换
  - 隔离通信

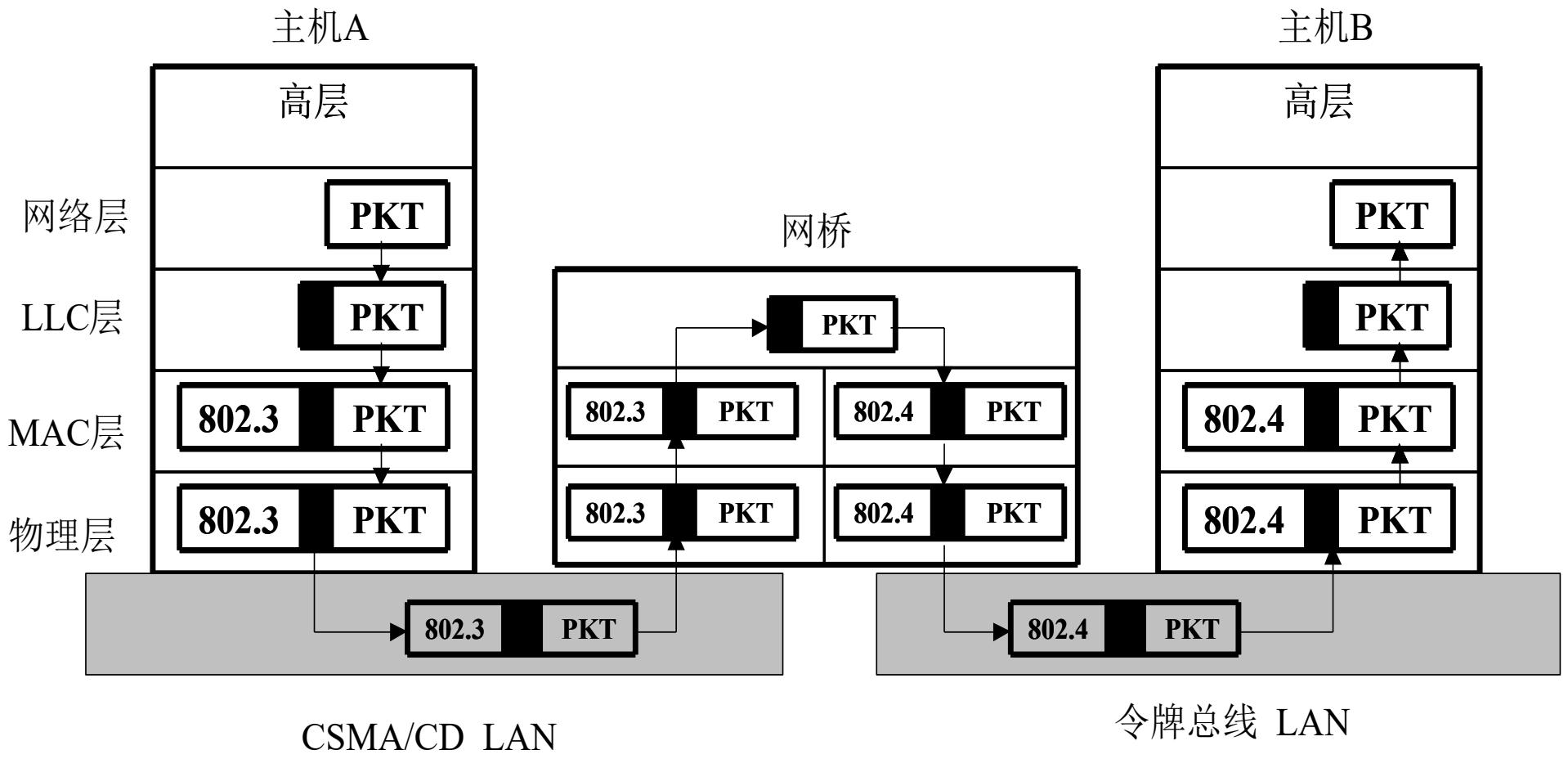
网桥的路由功能

- 网桥具有根据帧的目的地址决定是否接收该帧的功能，也就是具有路由的功能



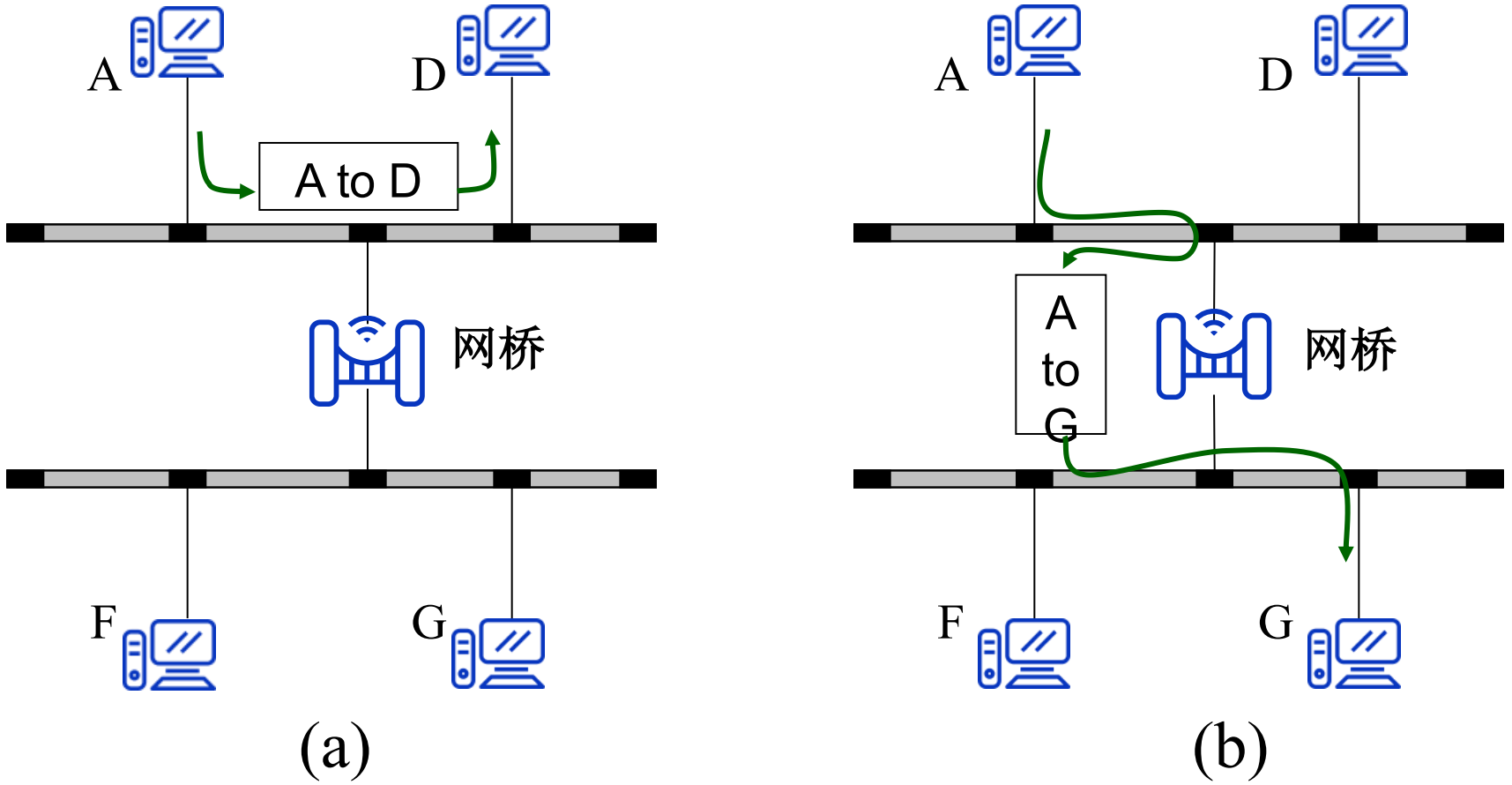
网桥的错误检测和帧格式转换功能

- 网桥工作在数据链路层，可以对数据链路层不一致的帧格式进行转换

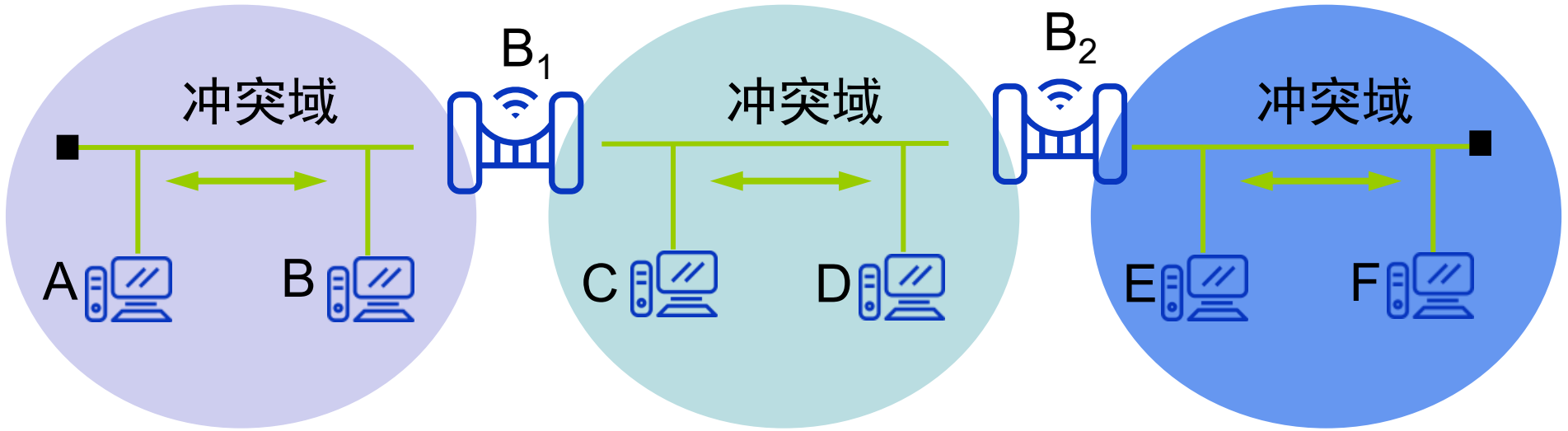


隔离通信功能

- 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冲突域



使用网桥带来的缺点

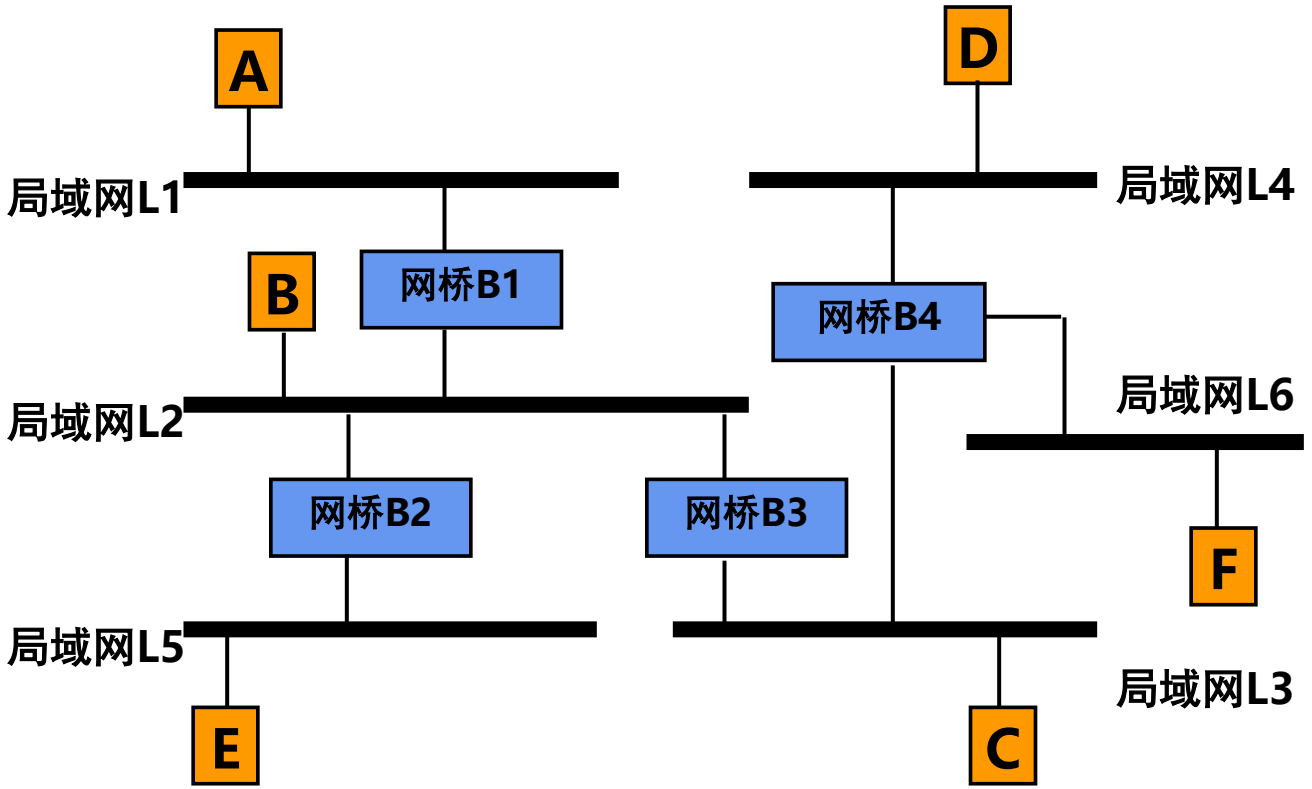
- 存储转发增加了时延
- 在MAC 子层并没有流量控制功能
- 具有不同 MAC 子层的网段桥接在一起时时延更大
- 网桥只适合于用户数不太多(不超过几百个)和通信量不太大的局域网

网桥路由

- 网桥决定转发哪个帧和往哪儿转发的过程被称为路由
- 网桥的三种路由策略：
  - 固定路由策略
  - 路由学习策略
  - 源路由策略
- 三种网桥：
  - 固定路由网桥
  - 透明网桥
  - 源路由网桥

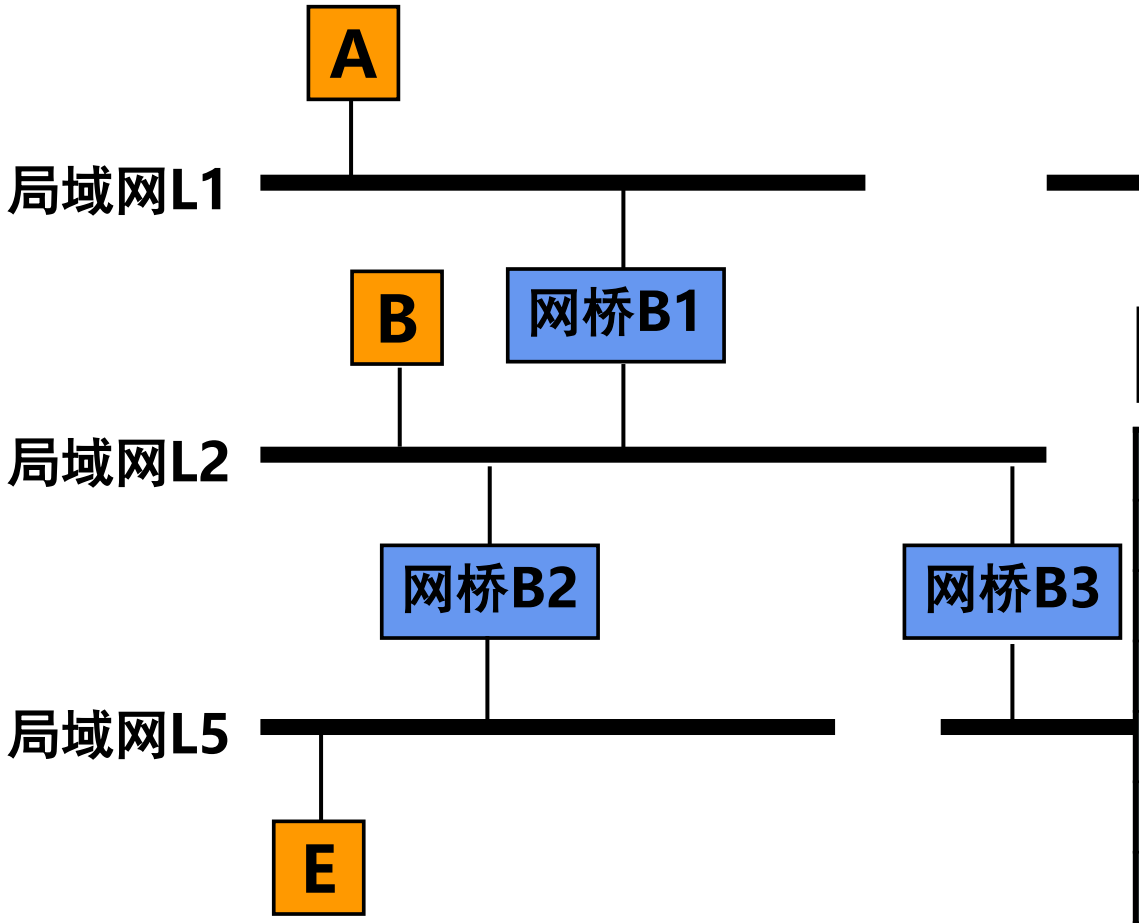
网桥路由算法

- 每个网桥中都有一张路由表
- 路由表中记录了到某个特定站点的帧应转发到哪个局域网中去的信息



固定路由算法

- 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



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	--

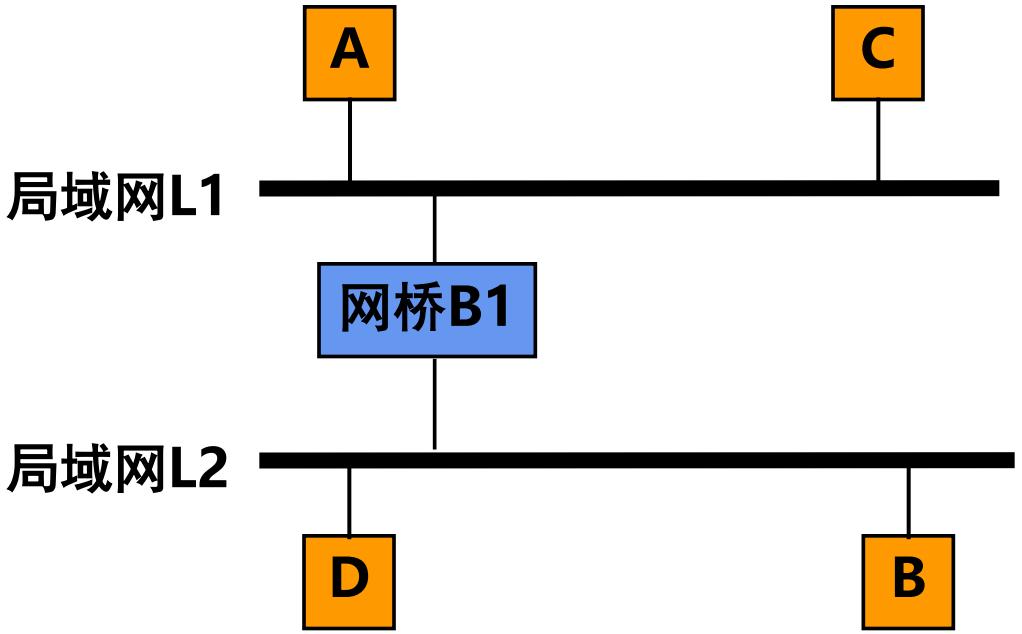
网桥B4

透明网桥(Transparent Bridge)

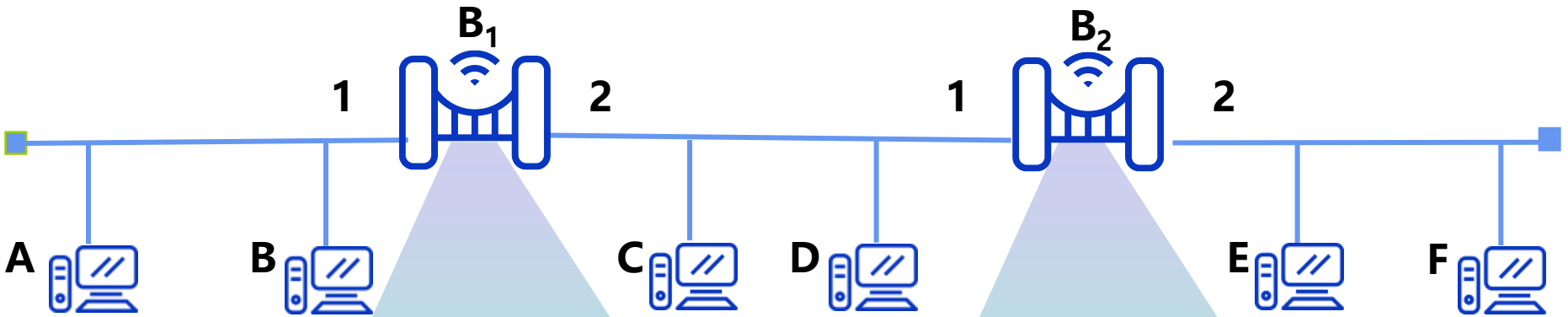
- “透明”是指局域网上的站点并不知道所发送的帧将经过哪几个网桥，因为网桥对各站来说是看不见的
- 透明网桥是一种即插即用设备，其标准是 IEEE 802.1D
- 采用逆向学习算法能够根据网络信息自动生成和修改自己的路由表
- 自动修改和生成路由表的能力称为路由学习或地址学习
- 要解决以下问题：
  - 路由表的初始化
  - 路由表的自动修改
  - 帧循环--生成树算法

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连接的所有局域网发送这个帧



路由表的建立过程举例



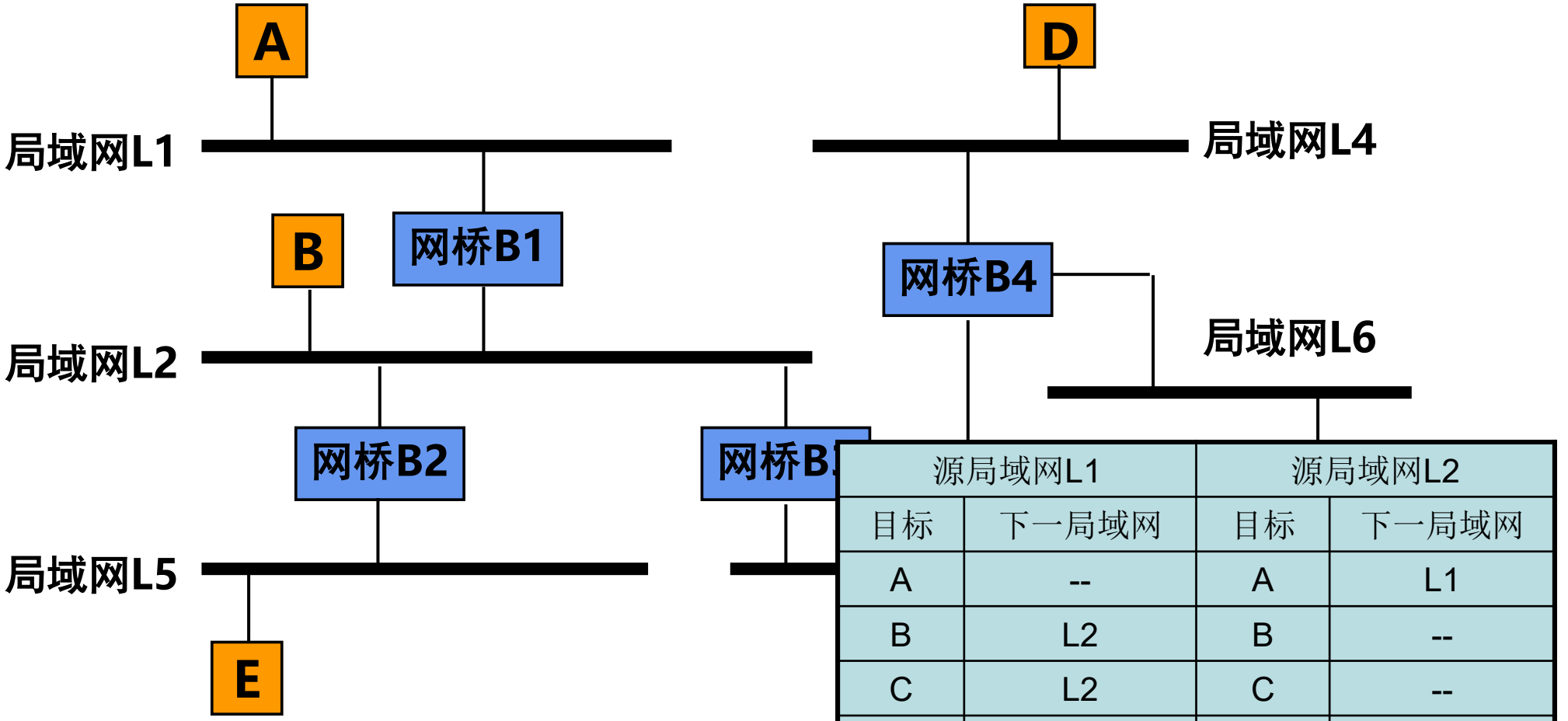
	地址	接口
A → B	A	1
F → C	F	2
B → A	B	1
	…	…

	地址	接口
A → B	A	1
F → C	F	2
	…	…

路由表的自动修改

- 每当网桥接收到一个帧时，就会检查帧的源地址，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地址表表项时，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

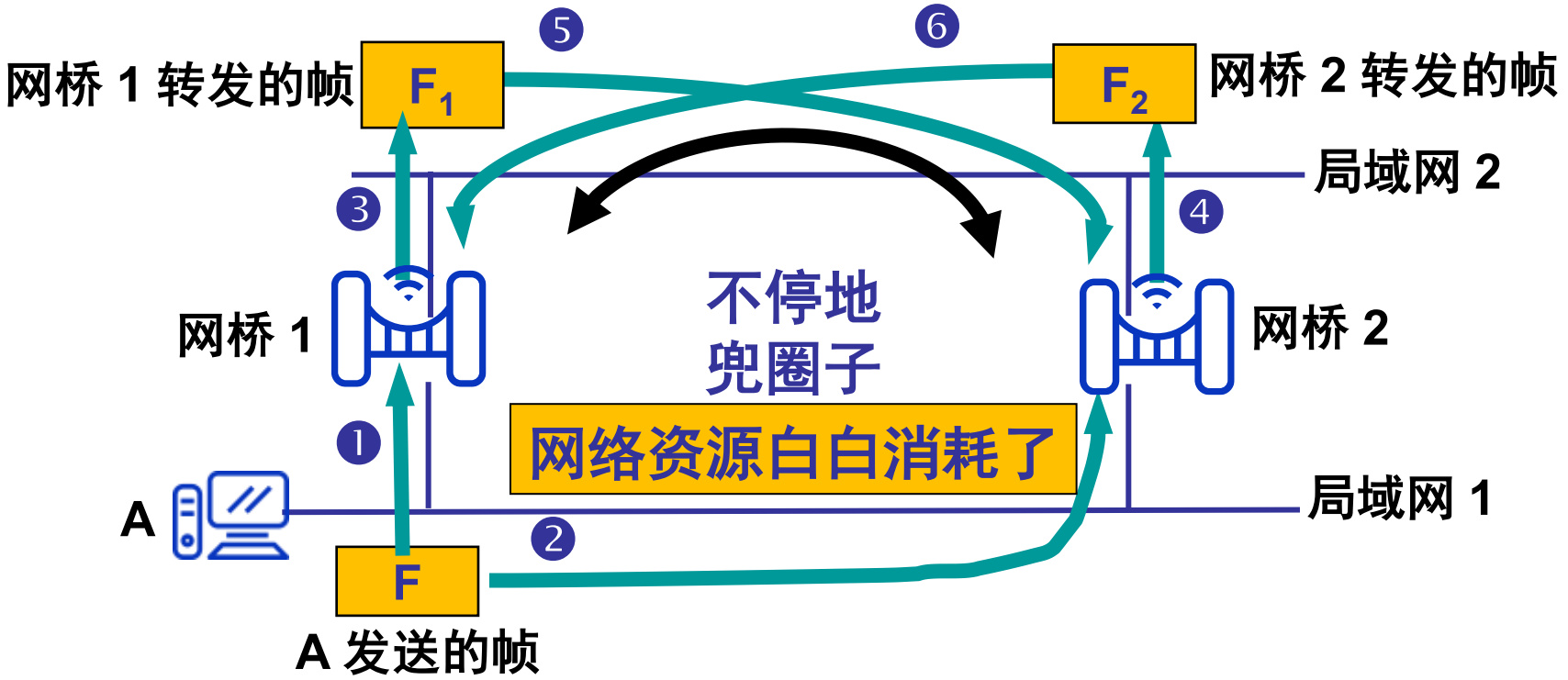
路由表的自动修改示例



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	L2	D	--
E	L2	E	--
F	L2	F	--

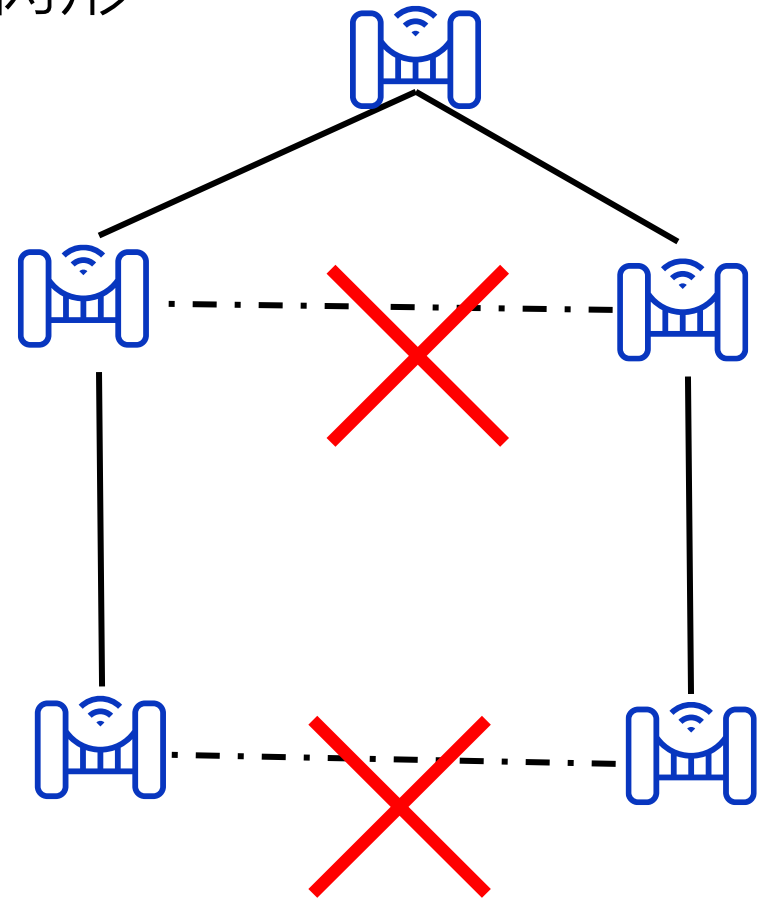
帧循环问题

- 当一个网络有环路时，就可能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



消除环路的思想

- 将存在环路的网络修剪为树形
  - 选择树根
  - 确定最短路径
  - 阻塞冗余链路



生成树算法(Spanning Tree Protocol)

- IEEE 802.1D 标准制定了一个生成树协议 STP，用于消除二层交换网络中的环路拓扑，其要点是：
  - 不改变网络的实际拓扑，但在逻辑上阻断冗余链路，使得从一台主机到所有其他主机的路径是无环路的树状结构，从而消除了兜圈子现象
  - 当前活动路径发生故障时，激活冗余备份链路，恢复网络连通性
  - 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

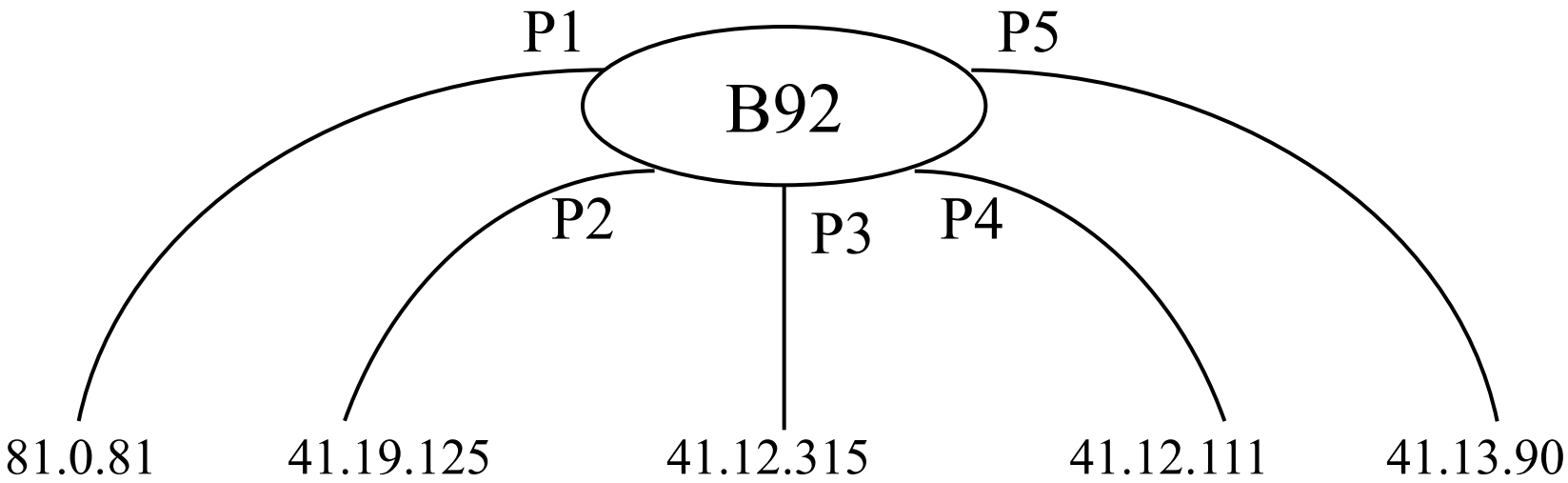
- BPDU：STP的数据单元，在网桥局域网内传递信息
- 根网桥：具有最小网桥ID的网桥被选作根网桥，网桥ID应为唯一的
- 根端口：在指定网桥上面，到根网桥路径费用最小的端口为根端口，如果指定网桥上面有几个端口到根网桥路径费用一样小，那么选择端口id 最小的端口为根端口
- 指定网桥：到根网桥路径费用最少的那个网桥为指定网桥，如果有几个网桥到到根网桥路径花费一样，选择id最小的作为指定网桥
- 指定端口：指定网桥上面和局域网相连的端口叫做指定端口，如果指定网桥上面有几个端口同时和局域网相连，那么选择端口id 最小的端口为所在局域网的指定端口

生成树算法过程

- 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帧来完成的
- 每个网桥确定其根端口，依据是：
  - 到根网桥的费用最低
    - 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：
  - 根路径费用低
  - 所在的网桥ID值最小
  - 端口ID值最小

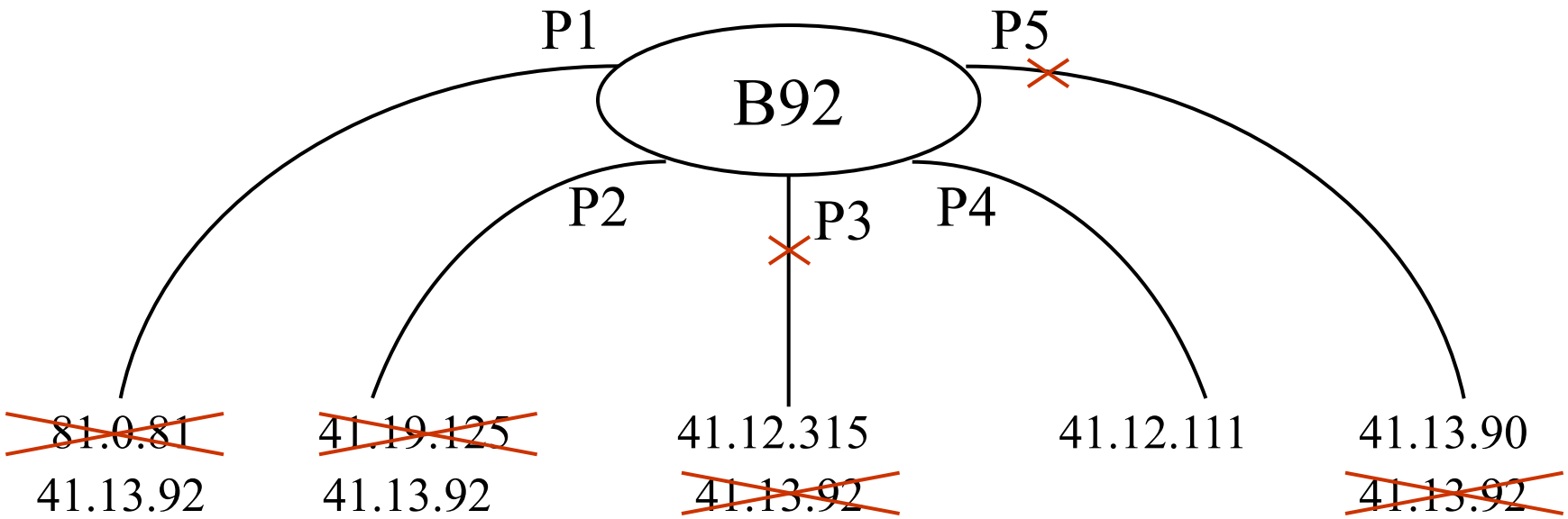
示例

BPDU：根网桥ID.到根费用.发送信息网桥ID



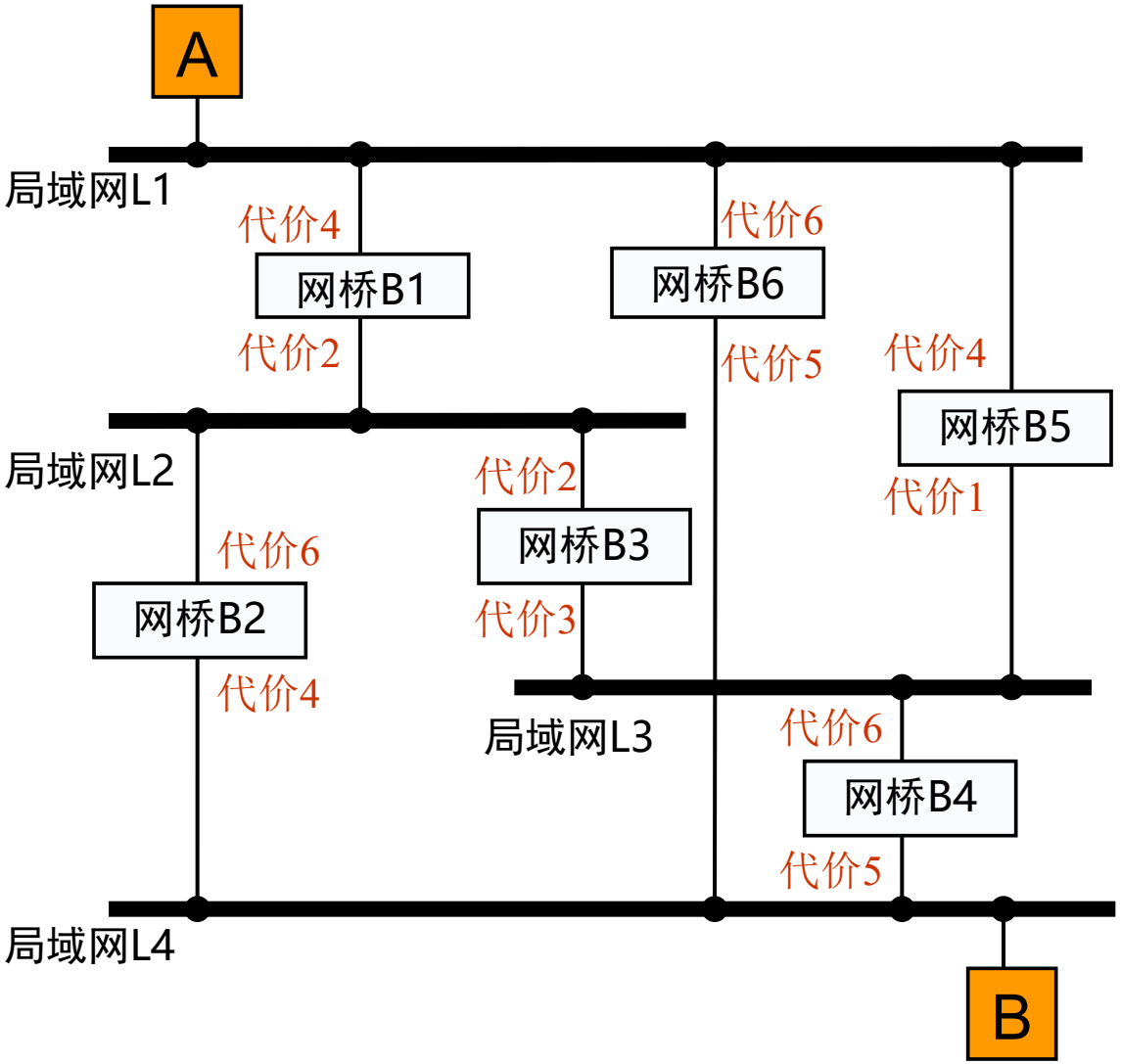
示例

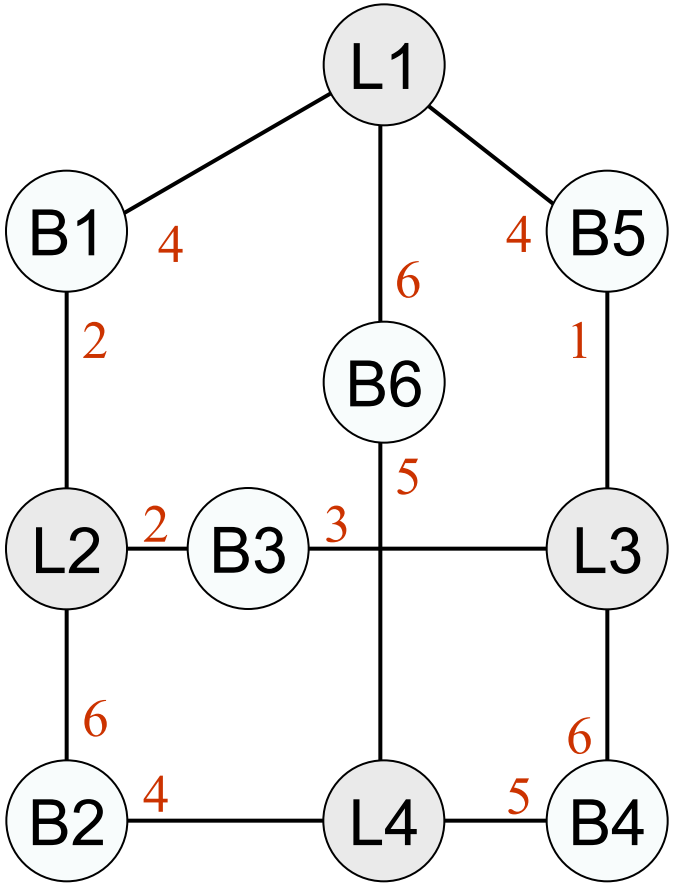
BPDU：根网桥ID.到根费用.发送信息网桥ID



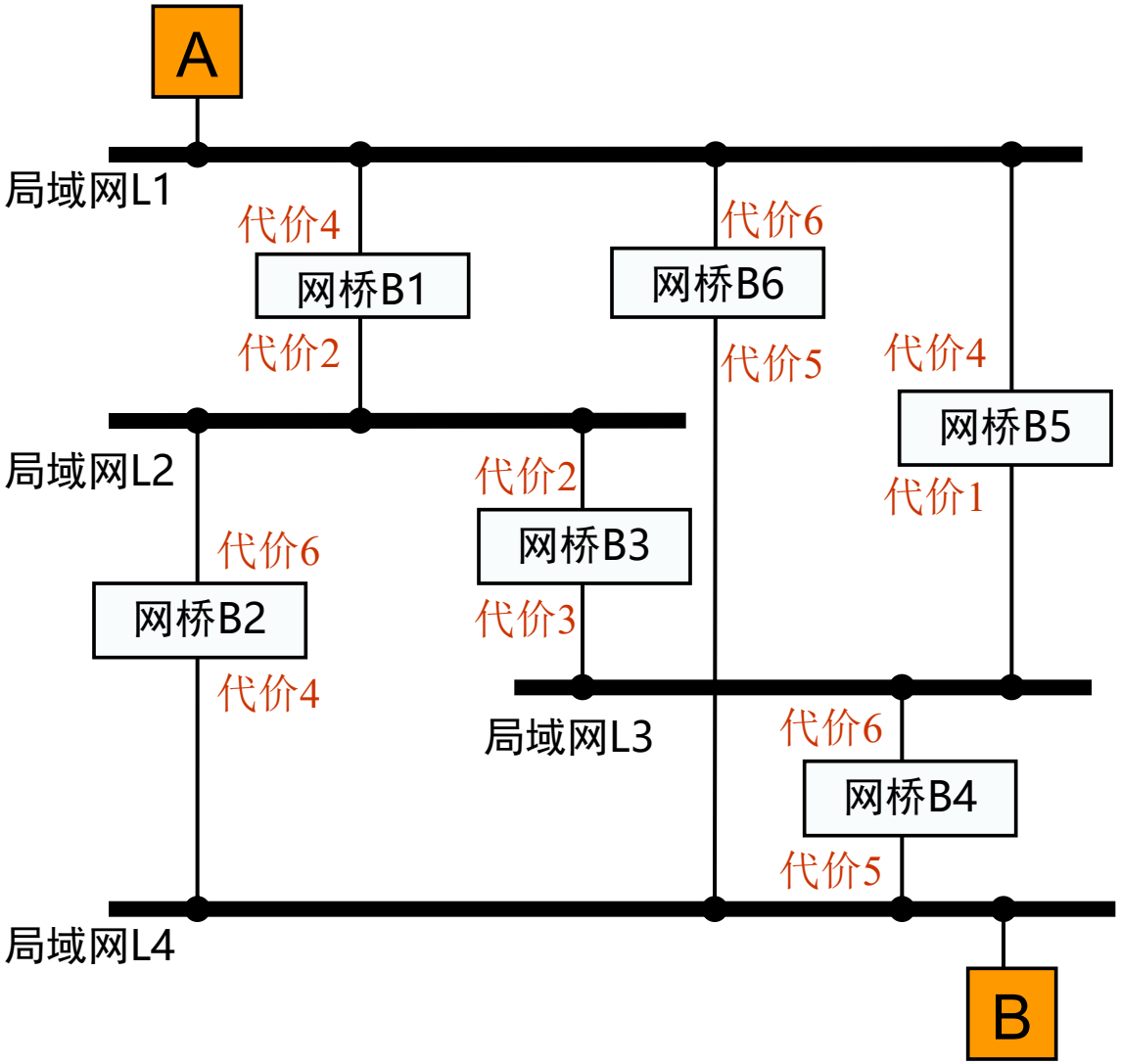
- P4：根端口
- P1、P2：指定端口
- P3、P5：阻塞状态

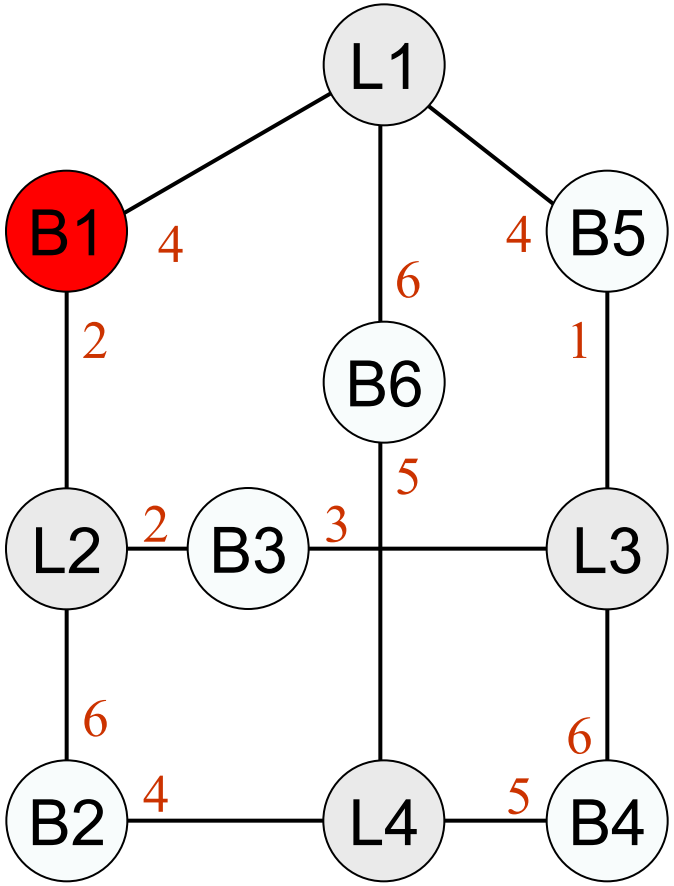
STP实例-互连局域网及其图表示



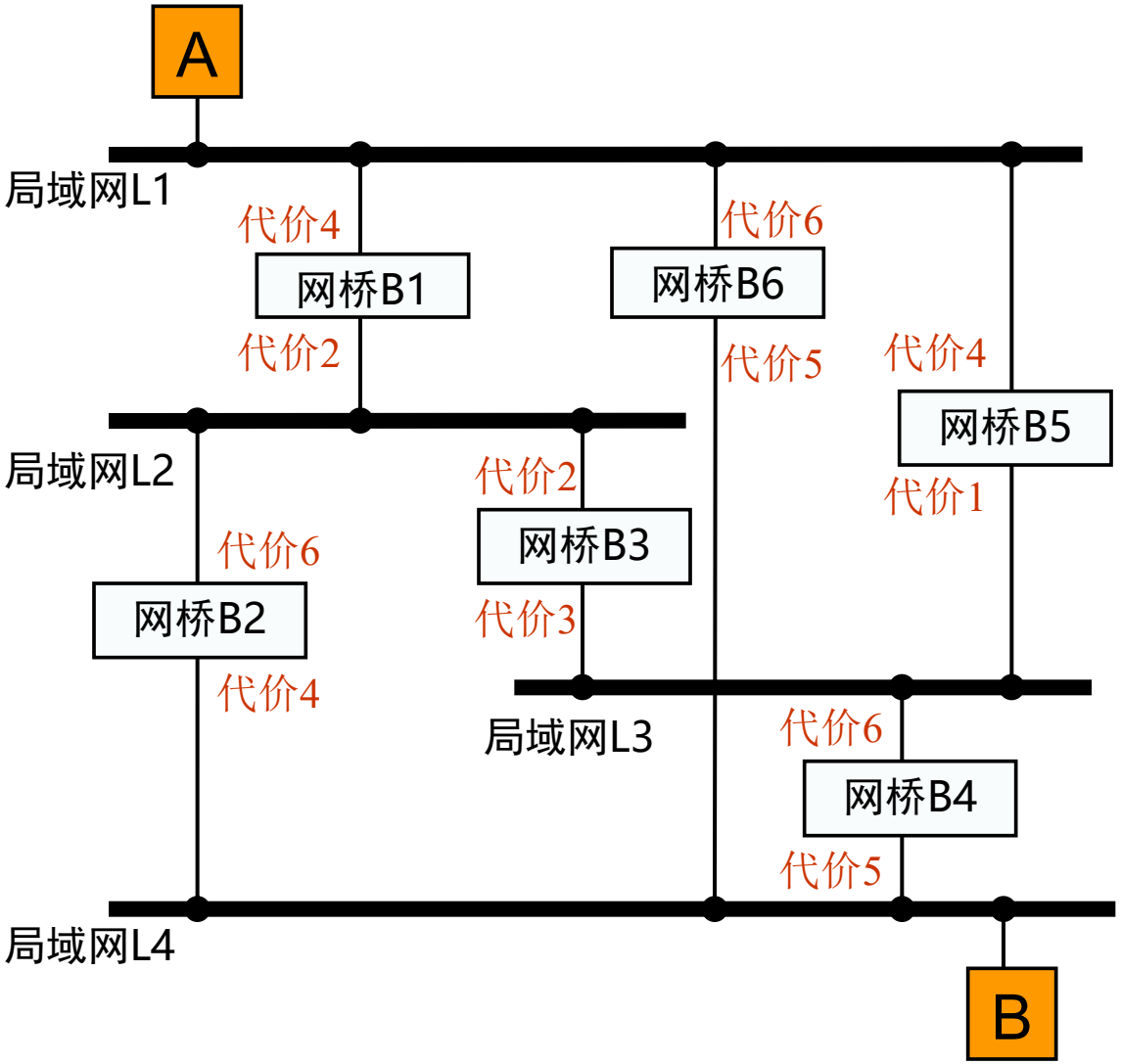


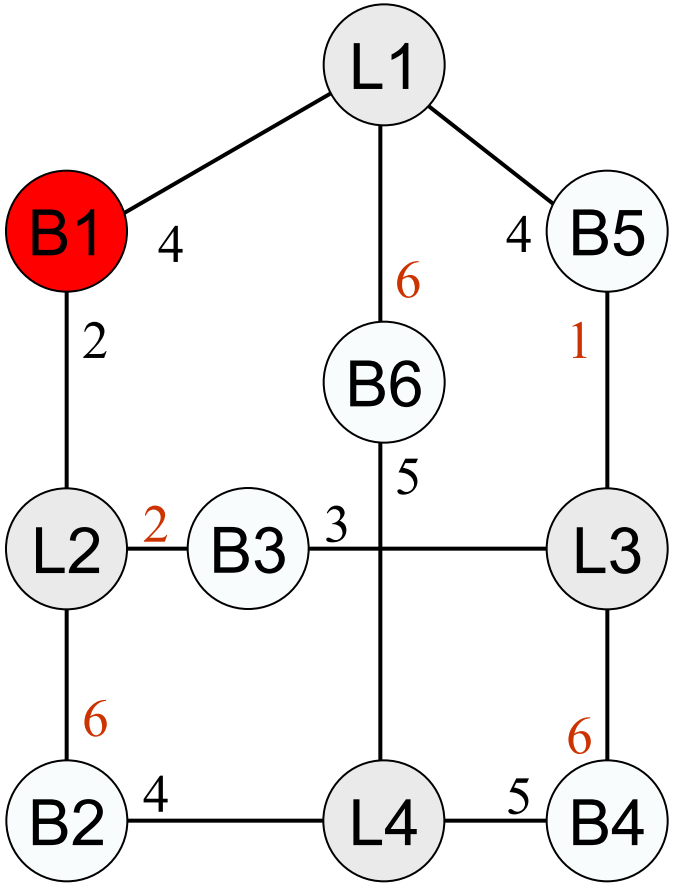
STP实例-确定根网桥


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

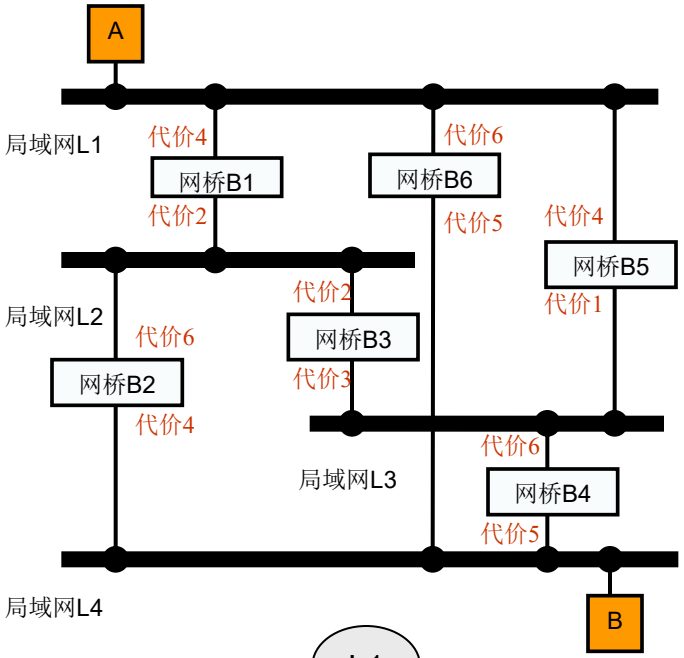

L1
B1
4
4
B5
6
2
1
B6
5
L2
2
B3
3
L3
6
6
B2
4
L4
5
B4

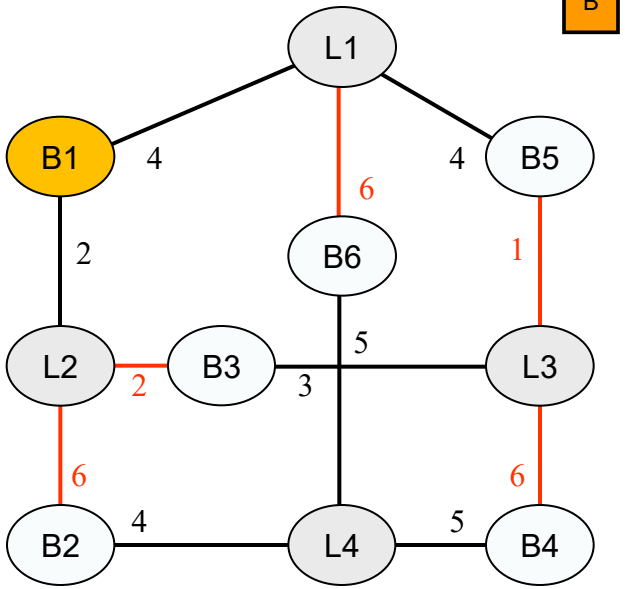
STP实例-确定根端口


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

STP实例-确定根端口

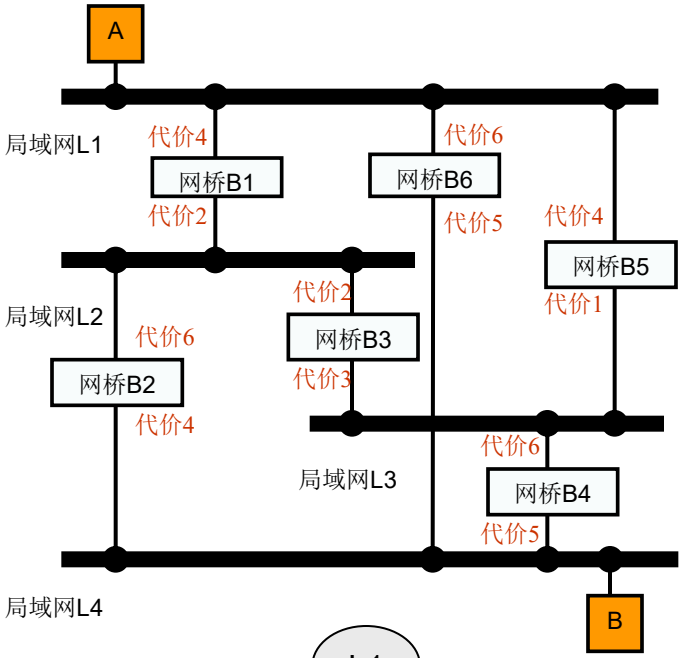


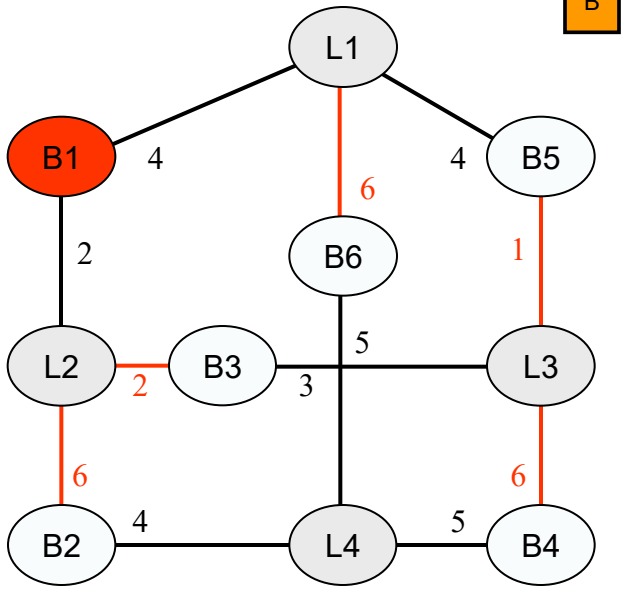


- 到根网桥的费用最低
- 网桥ID最小
- 端口ID最小

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

STP实例-为每个网络指定网桥

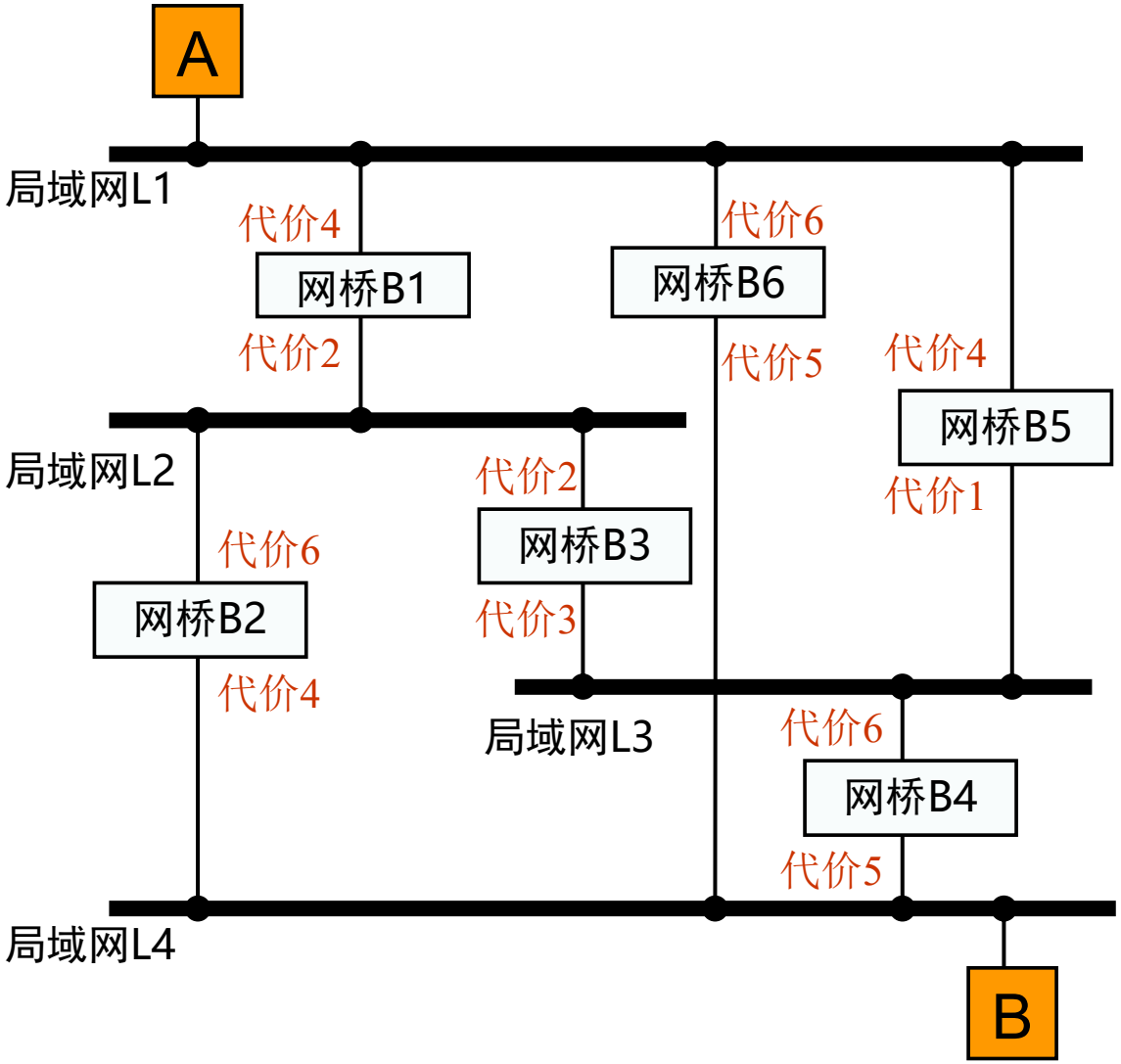


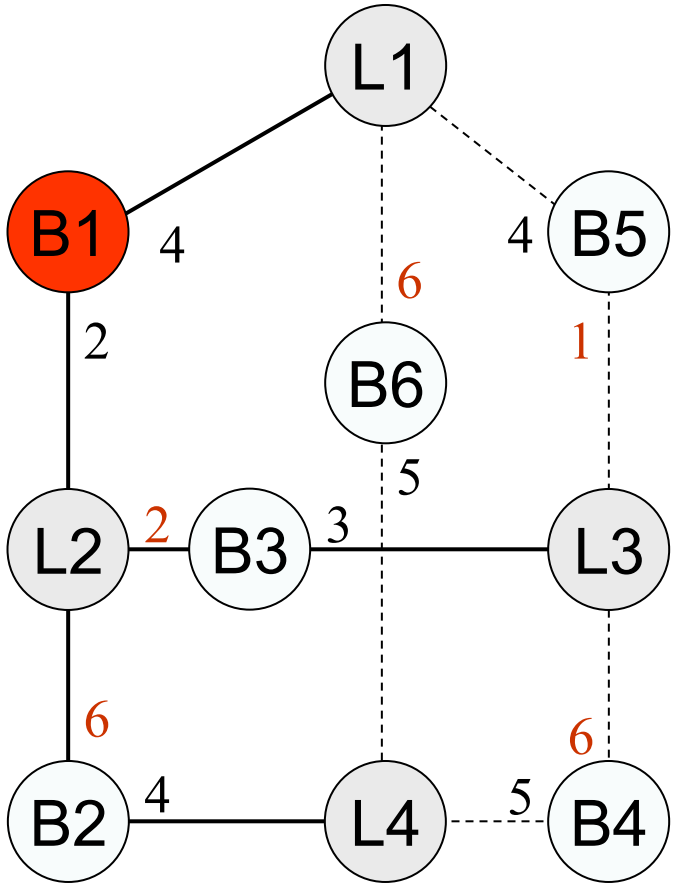


- 通过某网桥至根路径费用最低
- 所在的网桥ID值最小
- 端口ID值最小

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

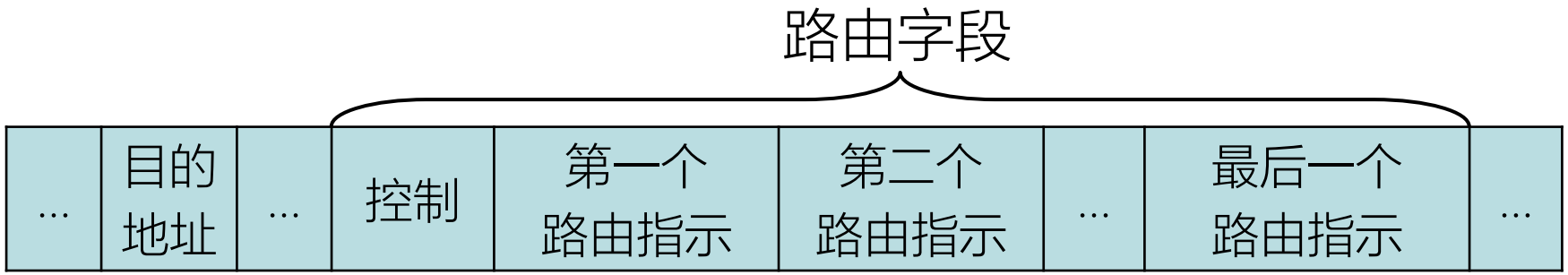
STP实例-生成树的拓扑及图形表示





源路由网桥

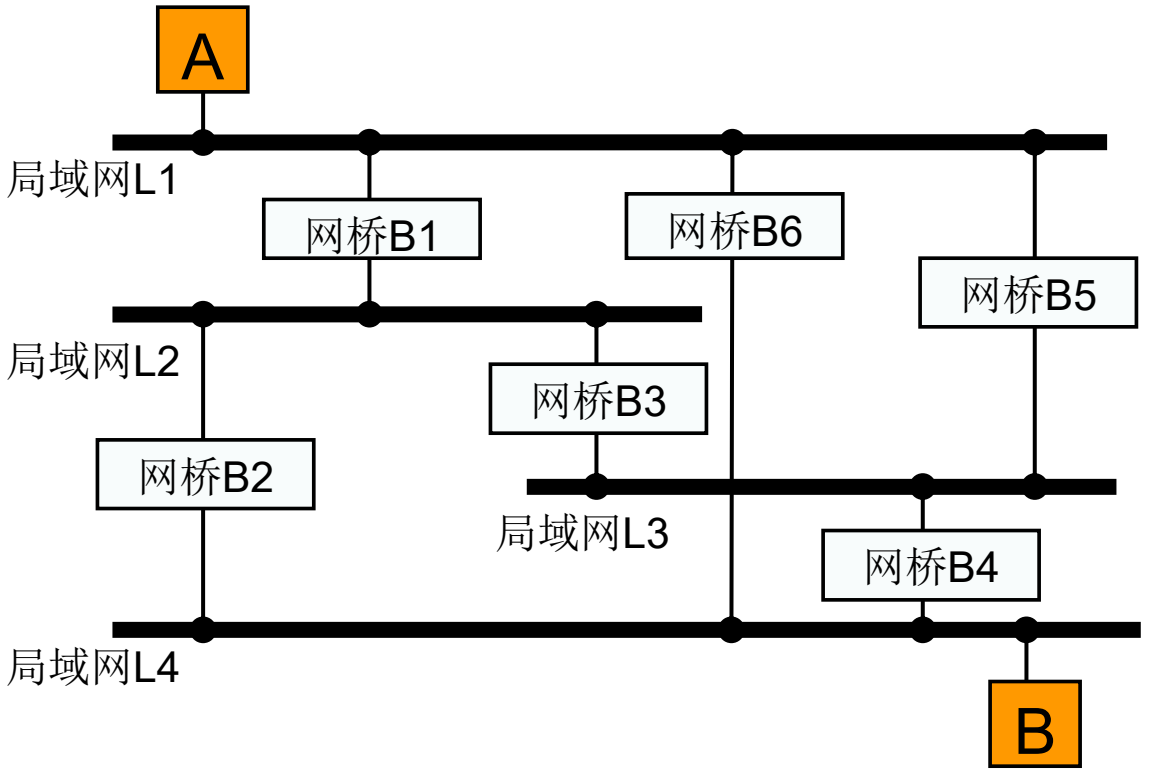
- 源路由网桥：让发送帧的源站点决定转发路由，而不是由网桥来决定
- 发送站点的网络软件确定到目的站点的路由，并将它存储在帧中，这个路由由路由指示（Route Designator）序列组成
- 每个路由指示由一个局域网和一个网桥ID组成



- 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，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

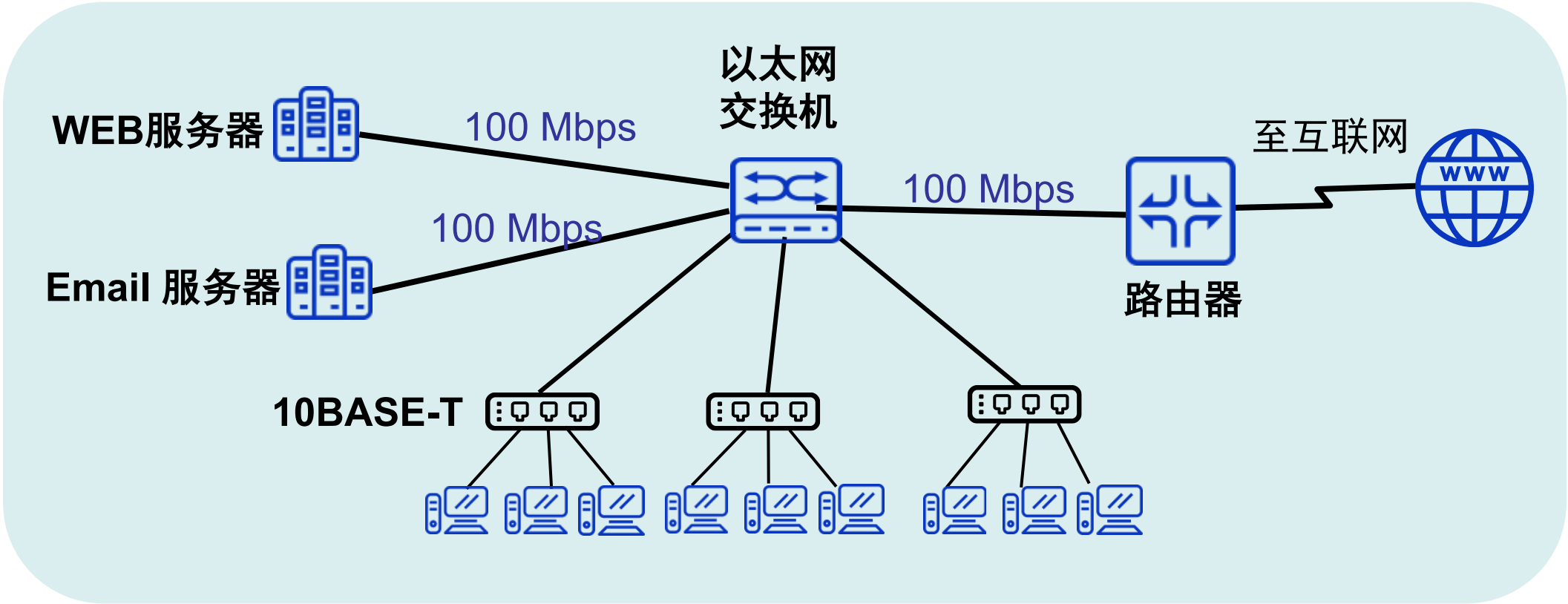
二层交换机

- 二层交换机的定义
  - 第二层交换机是多接口网桥
  - 以太网交换机是针对以太网数据帧中的目的MAC地址进行寻址并转发数据的设备
  - 以太网交换机在2000年前，业界基本上意指为二层以太网交换机。2000年后，对大容量的交换机融进了三层路由功能，出现了三层以太网交换机概念
  - 以太网交换机如果不特指，一般是指二层以太网交换机

二层交换机

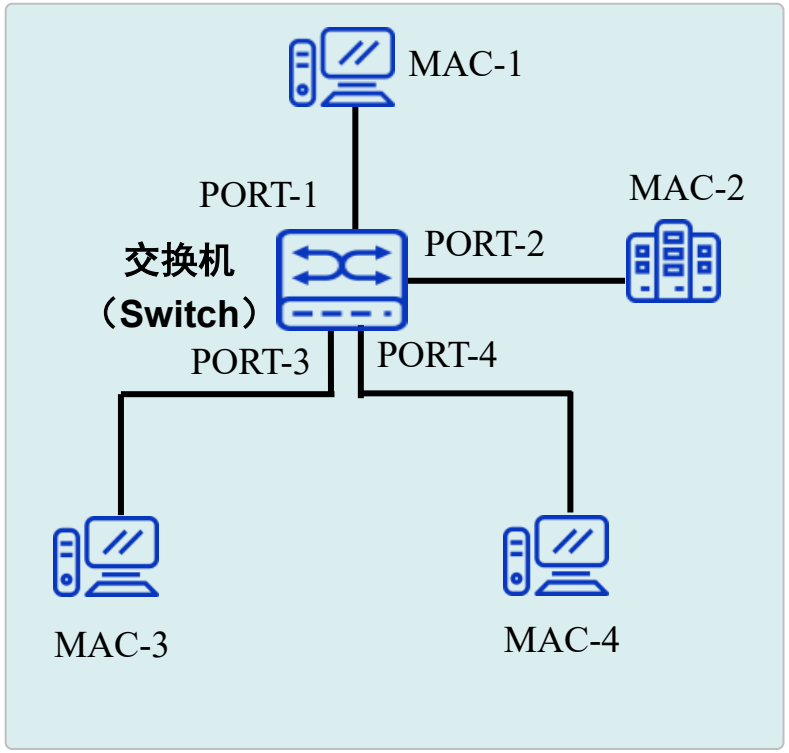
- 以太网交换机的特性
  - 有多个端口，每个端口可接入一台主机或另一个以太网交换机，各端口工作在全双工方式
  - 交换机从某一节点收到一个帧后，立即在其内存中的地址表（端口号 - MAC地址）进行查找，以确认该目的MAC的网卡连接在哪一个端口上
  - 每个端口有各自的带宽，各端口之间并行工作，可以提高网络吞吐量

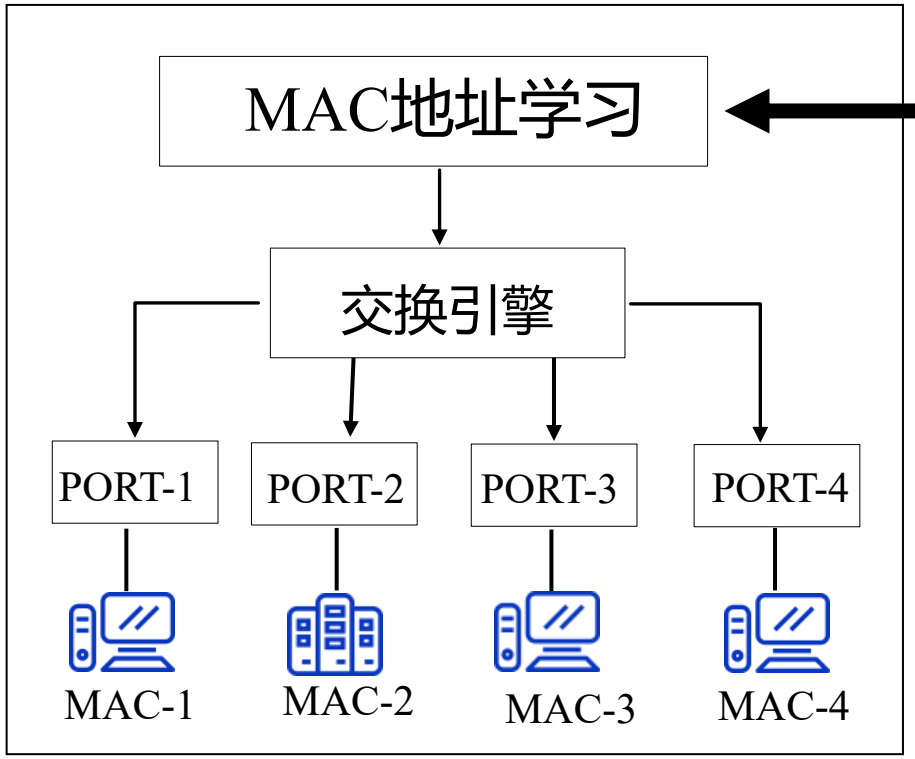
用以太网交换机扩展局域网



以太网交换机的用途

- 主机通过端口接入到交换机上，交换机具有地址学习功能，记忆各主机连接的端口号，通过交换引擎实现不同端口上两台主机之间的数据帧交换





地址	端口
MAC-1	1
MAC-2	2
MAC-3	3
MAC-4	4

地址-端口表

典型的转发模式（1）

- 直通式（Cut Through）
  - 交换机在输入端口检测到一个数据帧时，检查该帧的目的地址，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，把数据帧直通到相应的端口，实现交换功能
  - 优点：不需要存储，延迟小、交换快
  - 缺点：
    - 交换机不保存数据帧内容，不能对数据帧进行循环冗余校验
    - 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

典型的转发模式（2）

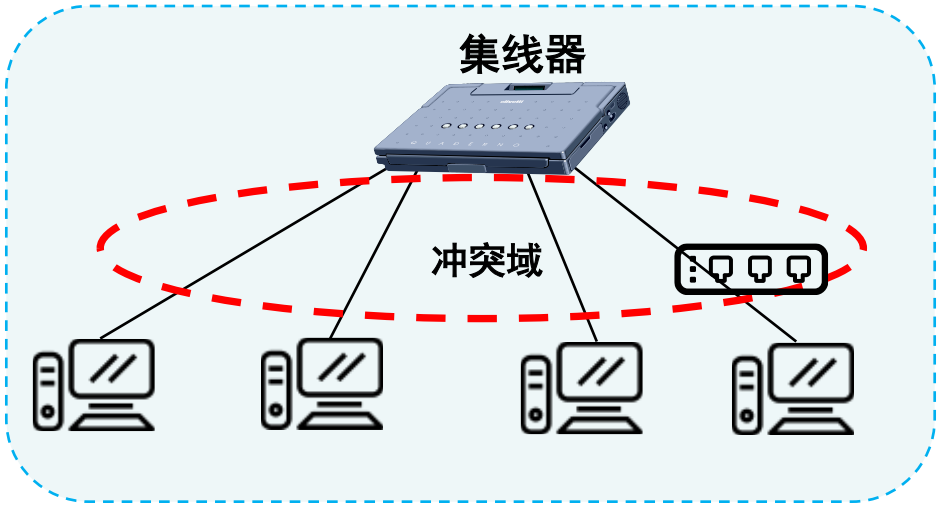
- 存储转发（Store and Forward）
  - 交换机完整地接收整个数据帧，对帧进行差错校验，在对错误帧进行处理后，才通过查找表转换成输出端口转发数据帧
  - 优点：
    - 它可以对进入交换机的数据帧进行错误检测，有效地改善网络性能
    - 可以支持不同速度端口间的转换，保持高速端口与低速端口间的协同工作
  - 缺点：延迟大

典型的转发模式（3）

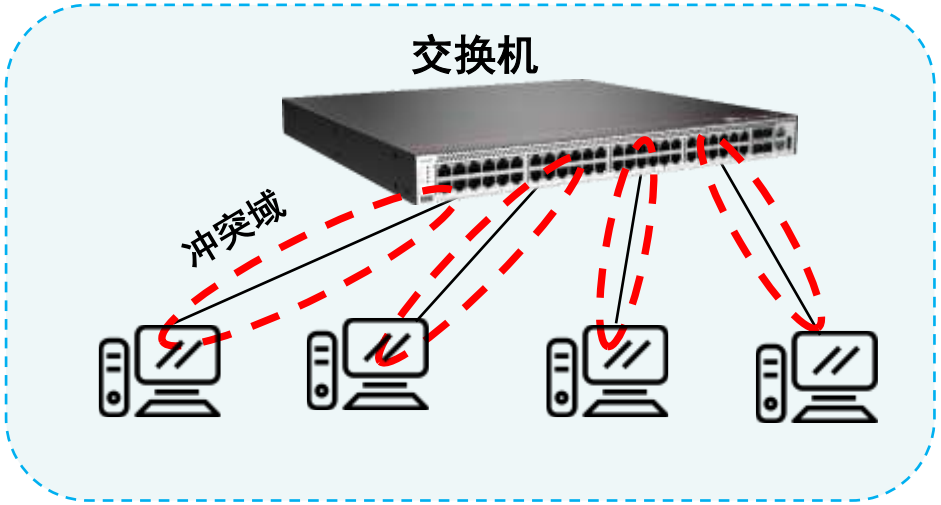
- 无碎片模式(Fragment Free)
  - 交换机等待数据帧进入交换机达到64字节时，读取数据帧首部中的目的MAC地址并转发该数据帧
  - 是存储转发模式与直通模式的折中方案
  - 优点：在转发数据帧之前过滤出冲突碎片，有效避免转发碎片帧
  - 缺点：没有对数据帧进行循环冗余校验,不能完全避免错误帧的转发

用以太网交换机实现链路层互连的优点

- 以太网交换机是一种即插即用设备，使用极其方便、简单
- 相互通信的主机独占传输信道，无冲突地传输数据，独享线路带宽
- 有多种速率的交换机，方便用户根据应用场景需要选择



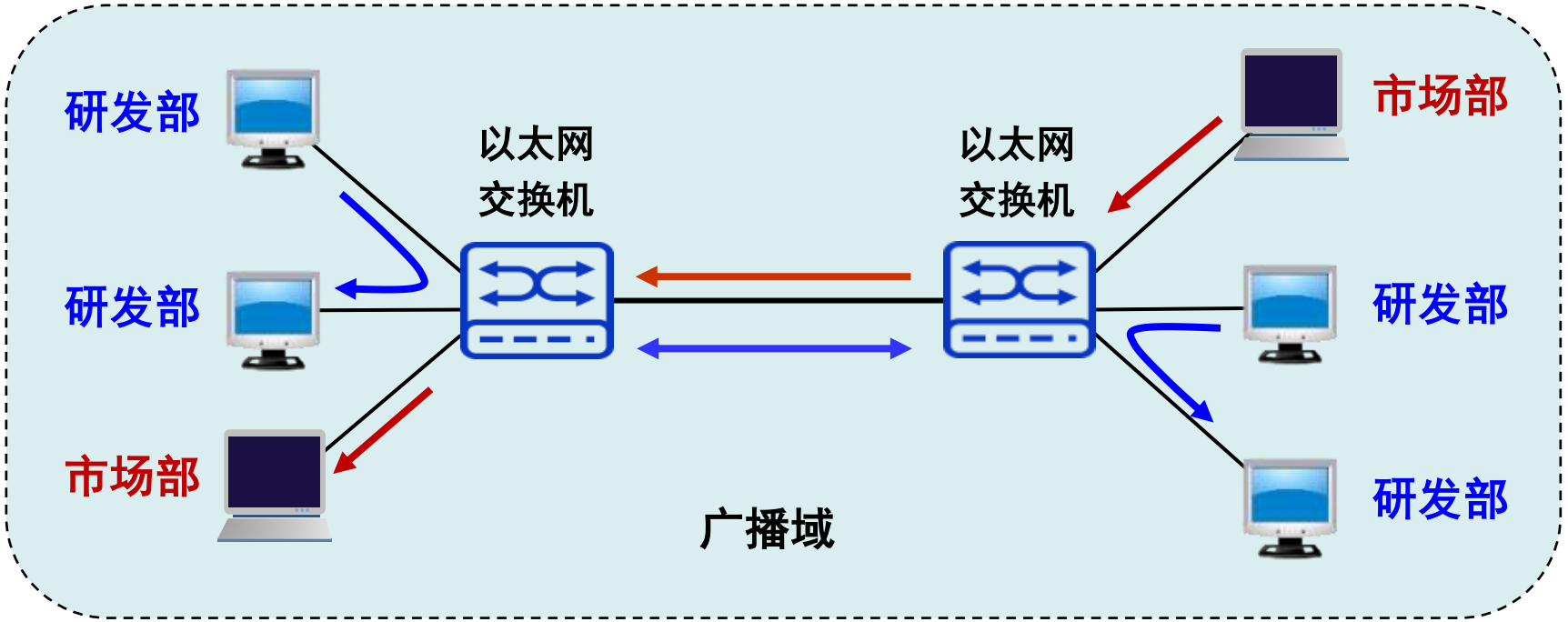
- 集线器的所有接口在同一个冲突域
- N 个用户共享集线器提供的带宽 B
- 平均每个用户仅占有 B/N 的带宽



- 以太网交换机的每个接口是一个冲突域
- 交换机为每个接口提供带宽 B
- N个用户，每个用户独占带宽 B

用以太网交换机互连的网络是一个广播网络

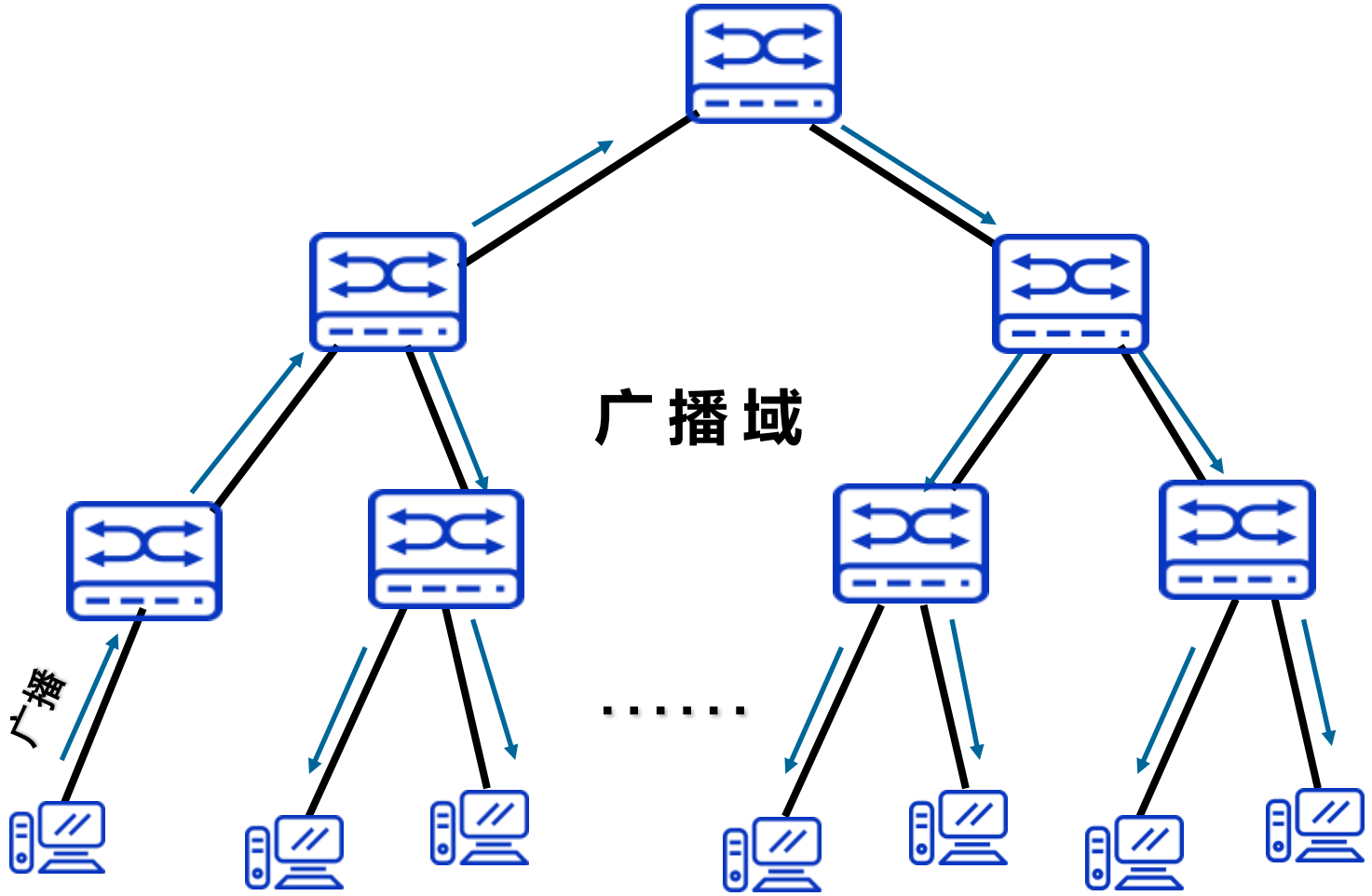
- 每个接口处于一个独立的冲突域，但所有计算机都处于同一广播域
- 广播域：指这样一部分网络，其中任何一台设备发出的数据帧由于交换机不知道目的MAC在哪个路径上而被广播后，网络中所有其他设备都能接收到这一帧



➢ 问题：研发部和市场部的计算机如何隔离？

整个网络在同个广播域带来的问题

- 网络性能受影响
  - 大量的广播报文，无谓消耗线路带宽和设备资源
  - 网络规模越大，问题越严重
- 信息安全问题
  - 用户的数据被广播到大量的、无关的用户主机上，信息被泄密

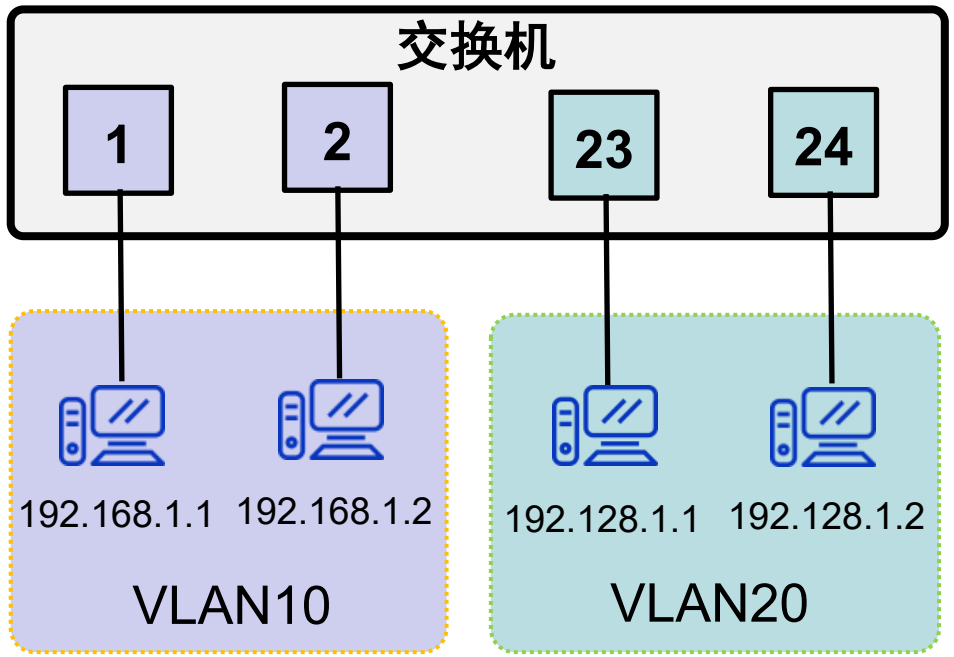


解决问题的方法：虚拟局域网(VLAN)

- IEEE 802.1Q 对虚拟局域网 VLAN 的定义：
  - 虚拟局域网 VLAN （Virtual LAN）是由一些局域网网段构成的与物理位置无关的逻辑组，而这些网段具有某些共同特征与共同需求
- 对虚拟局域网 VLAN 定义的理解：
  - 是按照某种需要将用户主机和网络资源进行一种虚拟的逻辑组合，非物理组合
  - 同个组合里的主机和设备处于同一广播域内，在链路层就能互连互通
  - 在不同组合里的主机和设备在链路层相互隔离，只有在网络层以上才能相互通信
  - 每一个数据帧都有一个明确的标识符，指明发送这个帧的主机属于哪个 VLAN

交换机上如何实现虚拟局域网 VLAN

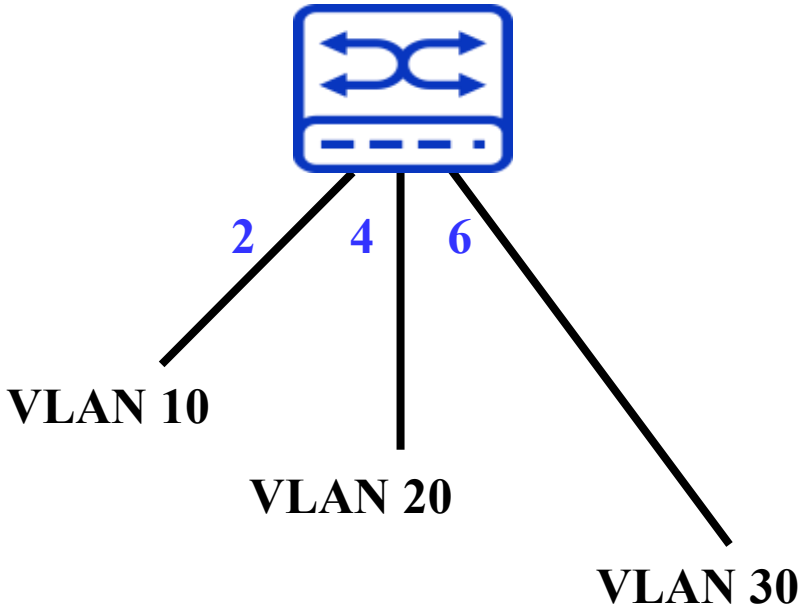
- 划分VLAN 的方法有多种，可以基于：
  - 交换机端口
  - 主机网卡的MAC地址
  - 协议类型
  - IP子网地址
  - 高层应用或服务



➢ 划分VLAN是交换机的一个基本功能

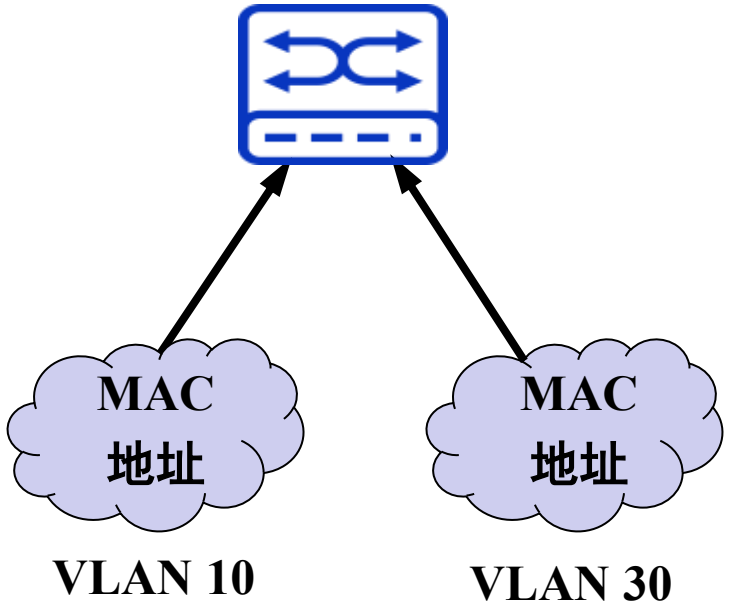
基于端口的静态划分

- 将端口与VLAN绑定
- 特点：
  - 是最简单、最常用的方法
  - 属于在第一层划分VLAN的方法
- 缺点：
  - 不允许用户移动



基于MAC地址的动态划分

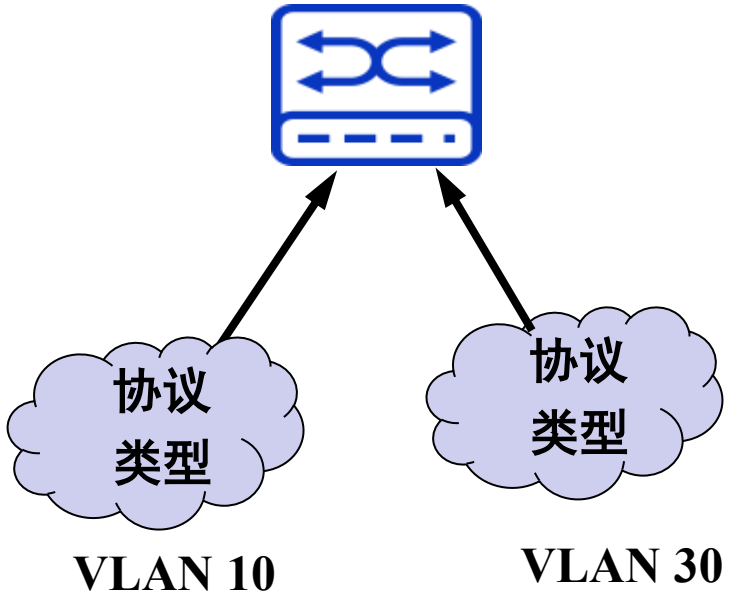
- 根据用户主机的MAC地址划分VLAN
- 属于在第二层划分VLAN的方法
- 允许用户移动
- 缺点：
  - 需要输入和管理大量MAC地址
  - MAC地址变化，需要管理员重新配置VLAN



MAC 地址	VLAN
00-15-F5-CC-C8-14	10
C0-AB-D5-00-18-F4	10
C0-C5-18-DE-BC-E6	30

基于协议的动态划分

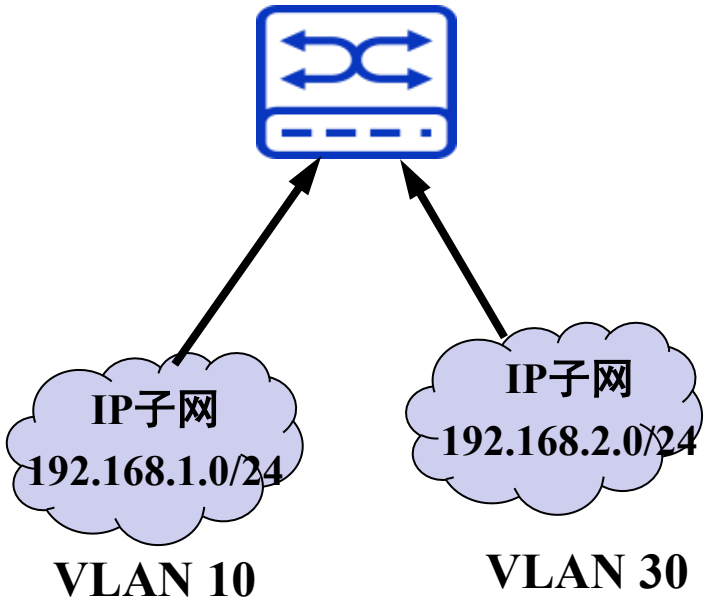
- 根据以太网帧的“类型”字段确定该类型的协议属于哪一个虚拟局域网
- 属于在第二层划分VLAN的方法



“类型”	VLAN
IP	10
IPX	30
……	…

基于IP子网的动态划分

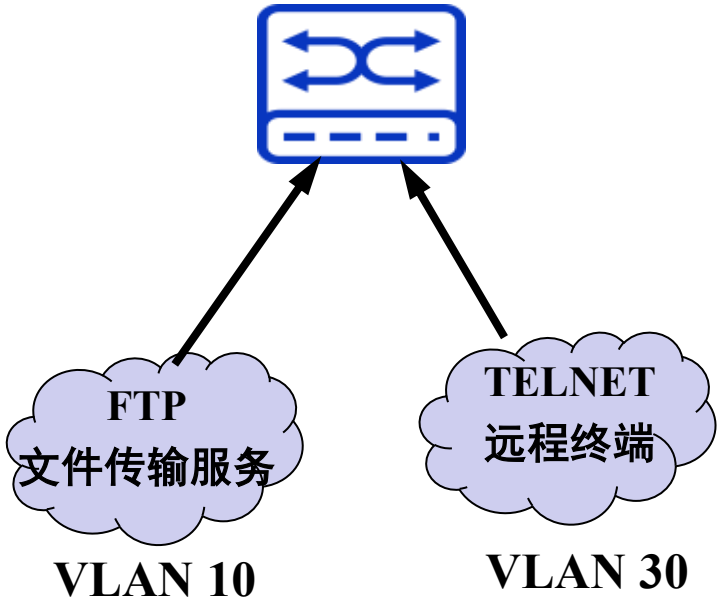
- 根据以太网帧的“类型”字段和IP数据报头中的源 IP 地址字段确定该 IP 数据报属于哪一个虚拟局域网
- 属于在第三层划分虚拟局域网的方法



IP 子网	VLAN
192.168.1.0/24	10
192.168.2.0/24	30
……	…

基于高层应用或服务的动态划分

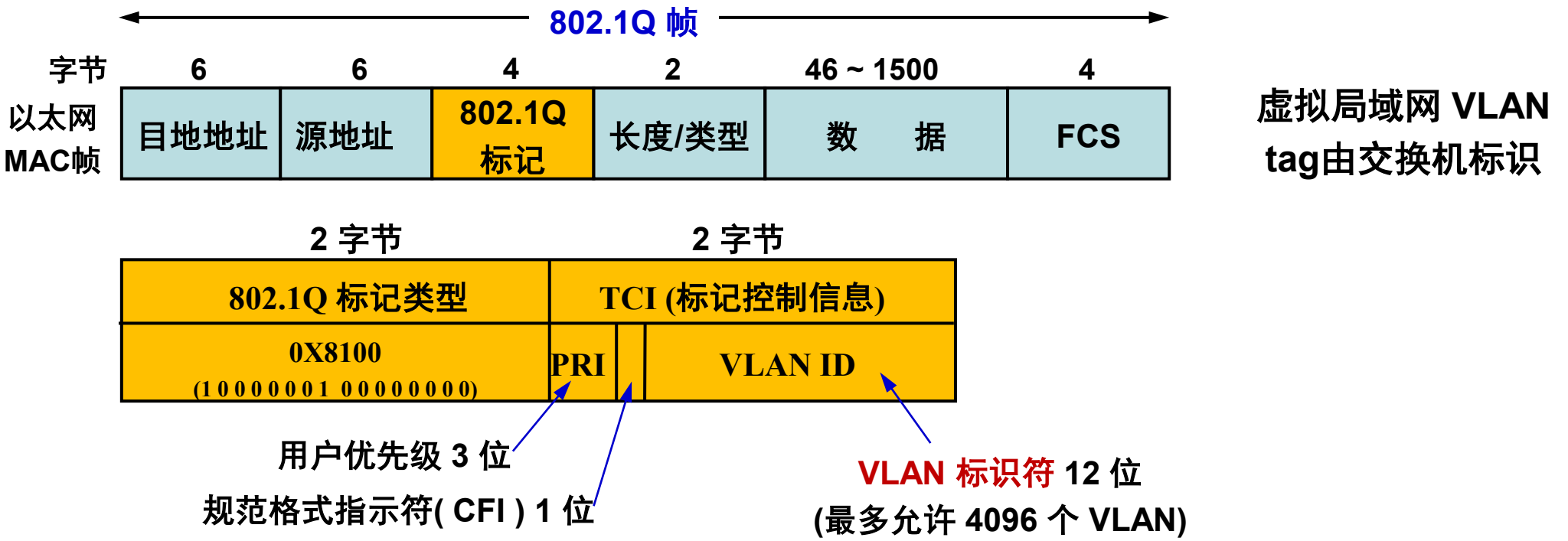
- 根据高层应用或服务、或者它们的组合划分虚拟局域网
- 更加灵活，但更加复杂



应用	VLAN
FTP	10
TELNET	30
……	…

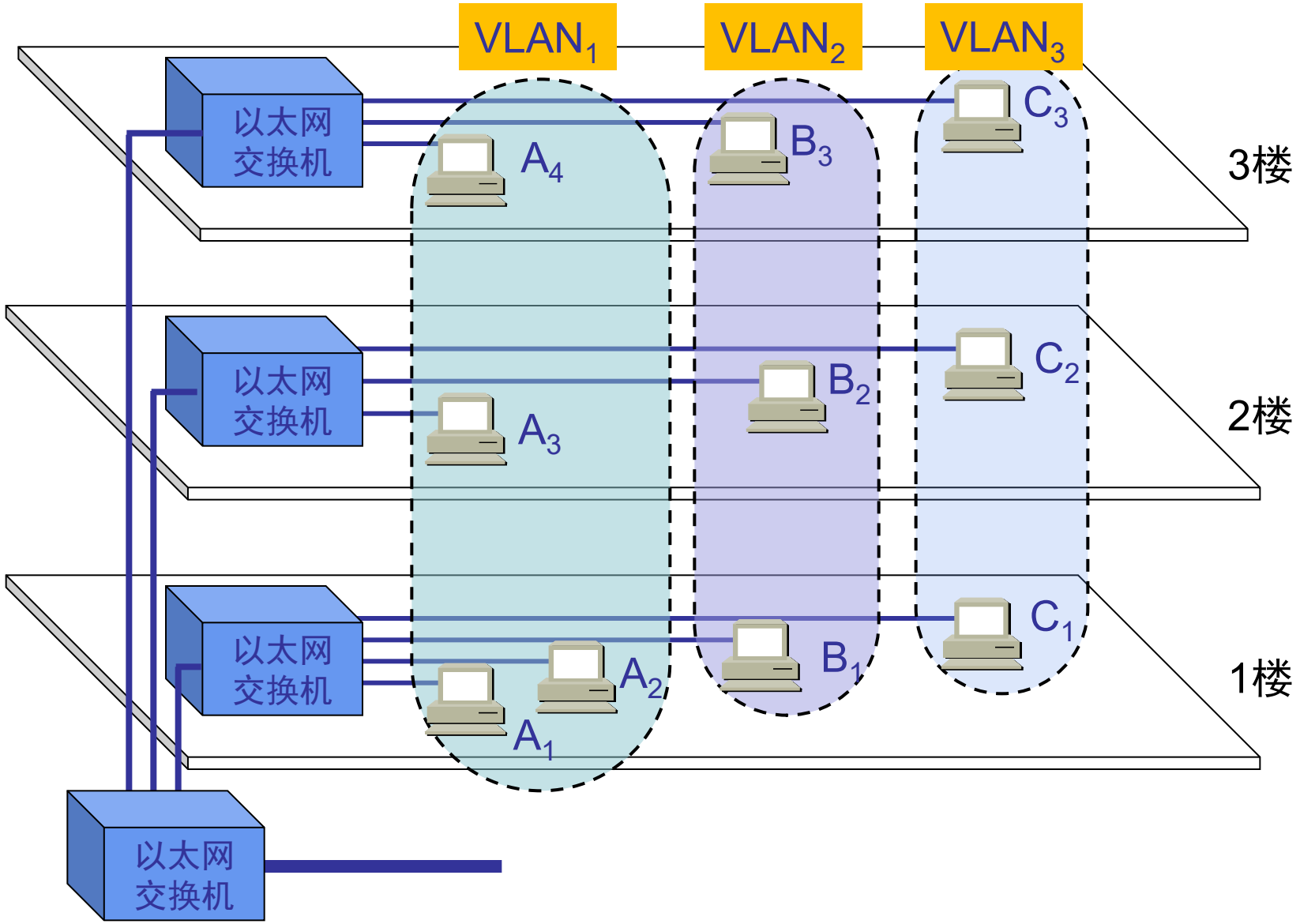
虚拟局域网使用的以太帧格式

- IEEE 的 802.3ac 标准支持虚拟局域网
- 虚拟局域网在以太网的帧格式中插入一个4字节的VLAN 标记 (tag)，用来指明该帧属于哪一个虚拟局域网



- 以太网 MAC 帧的最大帧长从原来的 1518 字节变为 1522 字节

VLAN示例--三个虚拟局域网



虚拟局域网优点

- 虚拟局域网（VLAN）技术具有以下主要优点：
  - 可以缩小网络的广播域范围，减轻“广播风暴”对网络的影响
  - 改善网络性能，避免因广播风暴带来的网络性能下降
  - 可以扩大网络规模，大的广播域被划分变小，网络总规模可扩大
  - 提高网络的安全性，防止数据信息被大范围扩散
  - 降低网络管理、维护、排障的技术难度

4.5 本章总结

- 数据链路层提供相邻结点之间的可靠通信
- 数据链路层的主要工作包括
  - 成帧：将数据封装成适合在本链路上传送的基本单元
  - 数据链路管理：对通信链路进行管理，对介质访问进行有效控制
  - 差错控制：处理数据帧的损坏、丢失、乱序、重复等，为网络层提供可靠的数据传送服务
  - 流量控制：控制发送端发送数据的速率，防止因发送端的数据发送速度过快而造成数据丢失和信道拥挤
- 常用的数据成帧方法有
  - 字节填充的标志字节法：硬件实现比较困难，适合软件实现，如PPP协议
  - 比特填充的标志比特法：硬件实现比较容易，如 HDLC 协议
  - 物理层编码违禁法：物理层实现，如802以太网链路

本章总结

- 数据链路层采用停止等待协议或者滑动窗口协议，通过接收确认、错误重传机制进行差错控制，保证数据帧的可靠传送
  - 停止等待协议：无差错信道的停止等待协议、有差错信道的停止等待协议
  - 滑动窗口协议：回退N自动重复请求、选择拒绝自动重复请求
  - 停止等待协议实现过程简单，但通信效率低
  - 滑动窗口协议实现过程复杂，但通信效率高
- 用停止-等待协议或者滑动窗口协议，在实现差错控制的同时，也控制了发送端发送数据的速度和节奏，确保接收端能够来得及正确接收数据，实现了流量控制

本章总结

- 数据链路层提供三类服务，根据链路特点采用其中的某类服务方式
  - 无确认无连接服务：通信时双方无需建立连接，发送方可直接发送数据，接收方不对收到的帧进行确认，适用于通信实时性要求比较高，信道误码率低的信道，如以太网
  - 有确认无连接服务：接收方对每一帧都需要确认，但不用建立连接，适用于不可靠的信道，如无线Wi-Fi 的802.11协议
  - 有确认有连接服务：通信时双方需要建立连接，在连接好的链路上进行通信，接收方需要对每帧数据进行确认，适用于长延迟的不可靠信道，且上层应用不是基于网络层和传输层的通信服务，而是直接基于链路层进行数据信息交互的场景

本章总结

- HDLC 是面向比特流的高速数据链路控制协议
  - 定义了信息帧（I-帧）、监管帧（S-帧）和无编号（U-帧）三种类型帧
  - 采用0比特填充法进行数据成帧与定界
  - 对数据帧采用CRC校验码进行错误检测
  - 采用滑动窗口、确认和超时机制等差错控制技术，保证帧不丢失
  - 可用于点对点、点对多点的双工或半双工链路的可靠通信

本章总结

- 以太网是目前应用最广的、最重要的局域网技术
  - 以太网信道质量好，在链路层产生数据帧差错的概率很小
  - 为提高数据链路层通信效率，在以太网中可以舍弃 LLC 层，直接由MAC层为网络层提供数据传输服务
  - 在采用多点半双工传输方式的以太网中，站点对传输介质的访问可能会产生冲突，需要通过 CSMA/CD 协议来进行介质访问控制
  - 以太网数据帧的成帧定界采用物理层编码违禁法，具体编码有曼彻斯特、4B/5B、8B/10B等，不同速率的以太网采用不同的编码方式

本章总结

- 以太网 MAC 帧格式有两种标准：
  - IEEE 802.3 帧和 Ethernet_ II 帧，通过帧中的长度/类型值来区分
  - 在TCP/IP中，通常采用Ethernet_ II 帧格式，直接将上层各种数据包封装到 MAC 帧中
- 以太网帧最大长度为1518字节，最小长度为64字节（长度不足64字节时需要进行填充）
- 以太网提供无连接、无确认的链路服务，MAC帧仅有检错功能

本章总结

- 以太网可以在链路层将用户主机或网络设备连接成具有一定规模的网络
- 用网桥和二层交换机进行链路层互连
  - 网桥和二层交换机：
    - 基于 MAC 帧中的源 MAC 地址进行自学习，自动生成 MAC 地址转发表
    - 基于帧中的目的MAC 地址查询 MAC 地址转发表，完成数据帧的转发
    - 如果查询不到 MAC 地址转发项，在所有端口上广播数据帧
  - 网络中存在多个网桥或交换机时，可能会产生环路，由此带来的广播风暴会导致网络瘫痪。需要通过生成树算法，将某些端口停用并作为备份端口，使得从一台主机到所有其他主机的路径是无环路的树状结构，从而消除兜圈子现象
  - 主机通过连接到二层交换机的一个端口上，独享线路带宽，实现点对点全双工信道，无冲突地传输数据，无需运行CSMA/CD 协议
  - 二层交换机典型的转发模式有：直通模式、存储转发式、无碎片模式

本章总结

- 虚拟局域网VLAN
  - 以太交换机互连的主机在同个广播域内，存在广播风暴带来的网络性能下降和用户信息安全问题
  - 虚拟局域网是按照某种需要将用户主机和网络资源进行虚拟的逻辑组合，将具有某种共同需要和共同特征的主机划入一个VLAN组
  - 虚拟局域网的 MAC 帧按照802.1Q 协议来标识 VLAN 标识符
  - 同一个VLAN组在一个广播域内，MAC帧可以在本 VLAN 组内传播
  - VLAN的划分方法有基于端口、MAC地址、协议、IP 地址、高层应用或服务等