



# 物联网通信技术

主讲人：宁磊

Email: [ninglei@sztu.edu.cn](mailto:ninglei@sztu.edu.cn)

# 目录

## CONTENTS

第1章. 物联网通信概述

第2章. 基带传输技术

第3章. 频带传输技术

第4章. 链路传输技术

第5章. 网络传输技术

第6章. 应用传输技术

第7章. 典型物联网通信系统

- 本章主要内容：组帧技术、分段技术、差错检测与控制，静态接入技术与随机接入技术。
- 本章学习目标
  - 了解组帧技术和分段技术的特点；
  - 掌握常见差错检测与控制方法；
  - 熟悉静态接入技术与随机接入技术的典型方法与基本原理。

- 物理传输问题：
  - 无法保证数据的正确性
  - 不可靠，无丢失重发、分拣功能
- 数据链路层包括：
  - 数据链路控制子层：保证“传好”，确保链路上的数据能够正确传输。确定一次传输数据的长度，依据此长度进行分段，定义校验位等
  - 介质访问控制子层：保证“可传”，确保数据有链路可用。定义多个节点如何共享一个介质，产生竞争时如何分配信道的使用权
  - 向下：利用物理层提供的位流服务
  - 向上：向网络层提供明确的服务接口

- 分段

链路层给物理层的单次数据传输的长度有最大值和最小值的限制，设最大值和最小值分别为  $L_{F\max}$  和  $L_{F\min}$  即数据长度应满足：  $L_{F\min} \leq L_F \leq L_{F\max}$ 。而设来自上层的数据分组长度为  $L_P$

□ 当  $L_P > L_{F\max}$  时，按照链路层数据域最大值  $L_{F\max}$  进行分段

$$N_{\text{fragment}} = \left\lceil \frac{L_P}{L_{F\max}} \right\rceil$$

分段的最后一段长度若小于  $L_{F\min}$ ，则需要用一些特殊的字符进行填充

□ 当  $L_{F\min} \leq L_P \leq L_{F\max}$  时，直接传输上层的数据分组

□ 当  $L_P < L_{F\min}$  时，特殊字符进行填充

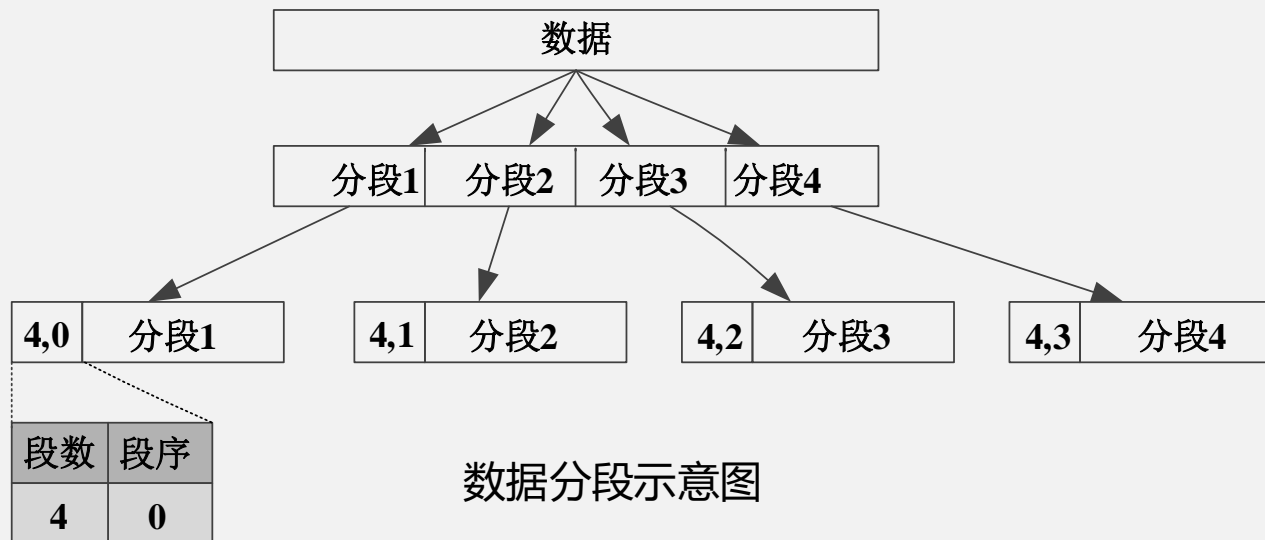
# 数据链路控制

- 分段

因此，若数据分段后的段数为：

$$N_{\text{fragment}} = \begin{cases} \lceil L_P / L_{F\max} \rceil, & L_P > L_{F\max} \\ 1, & L_{F\min} \leq L_P \leq L_{F\max} \\ 1, & L_P < L_{F\min} \end{cases}$$

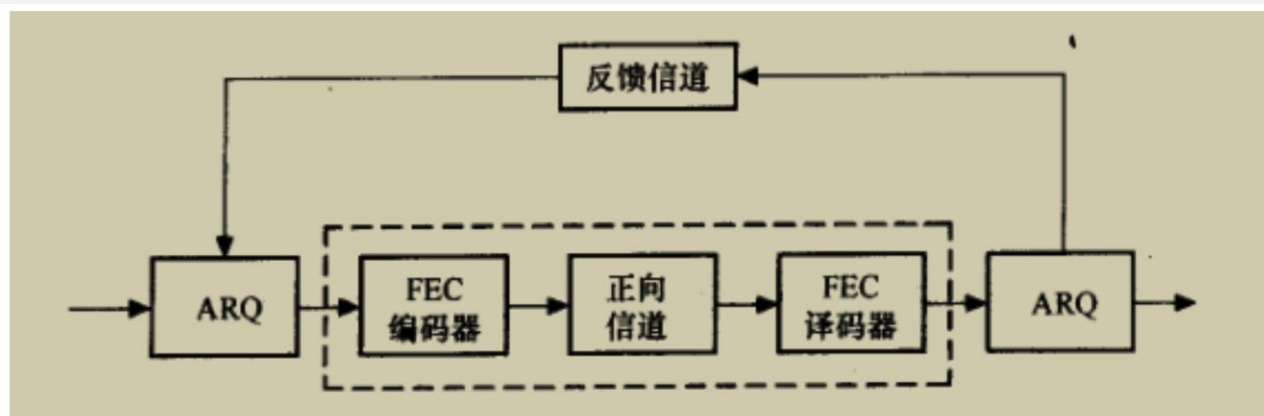
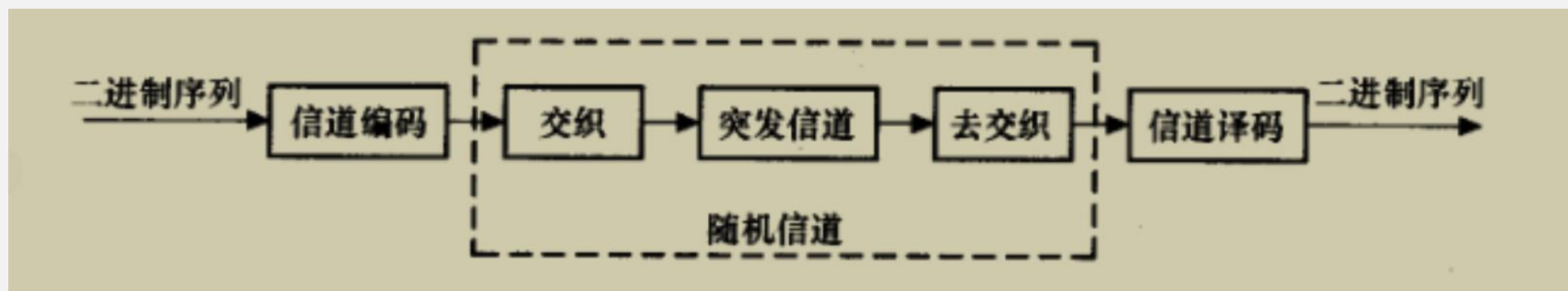
为了保证接收节点知道有多少段及当前段号，要在每个分段前面加上**分段控制字**，如下图所示



# 数据链路控制

- 差错检测

传输错误检测技术的目的是有效发现一帧数据经过物理信道传输后是否正确。常用的检错方法有两类：**奇偶校验**和**循环冗余校验**

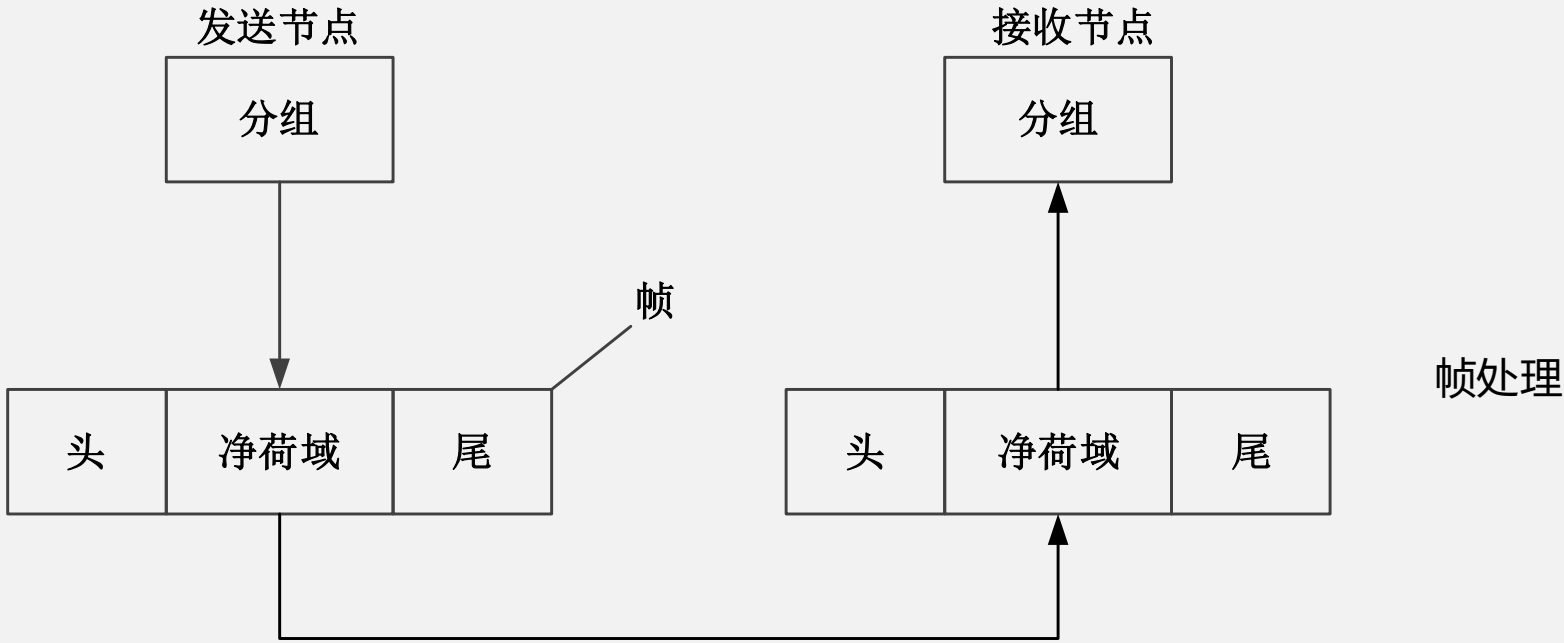


FEC/ARQ 混合方式原理图

# 数据链路控制

- 组帧

为了能使接收节点正确地接收并检查所接收的帧，发送节点必须依据一定的规则把网络层递交的分组封装成帧（即组帧）。如下图所示：



组帧技术主要解决什么时刻是一帧的开始，什么时候是一帧的结束，哪一部分是差错校验比特等问题





- 组帧

三种组帧技术：

- 面向**比特**的组帧技术：用一串特殊的比特来标志帧的起始和结束
- 面向**字符**的组帧技术：用字符填充帧首尾，确定帧的开始和结束
- **长度计数**的组帧技术：用长度标志一帧含有的字符数，从而确定帧的结束

解帧：接收节点收到比特流后，需要按照组帧的规则恢复所传输的数据

# 数据链路控制

- 组帧

- 面向比特的组帧技术

采用一个特殊的比特串 (Flag) , 例如 01111110, 来表示一帧的正常结束和开始

当信息比特流中出现与 Flag 相同的比特串 (如连续出现6 个"1") 时, 那么就会误认为是帧的边界。为了避免这种情形采用的办法是比特插入技术

**零比特填充法:** 高级数据链路控制 (High-Level Data Link Control, HDLC) 协议采用的方法

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

发送端连续发送 5 个 “1” 后, 填入1个 “0”

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

接收端将 5 个连 1 之后的 “0” 删除

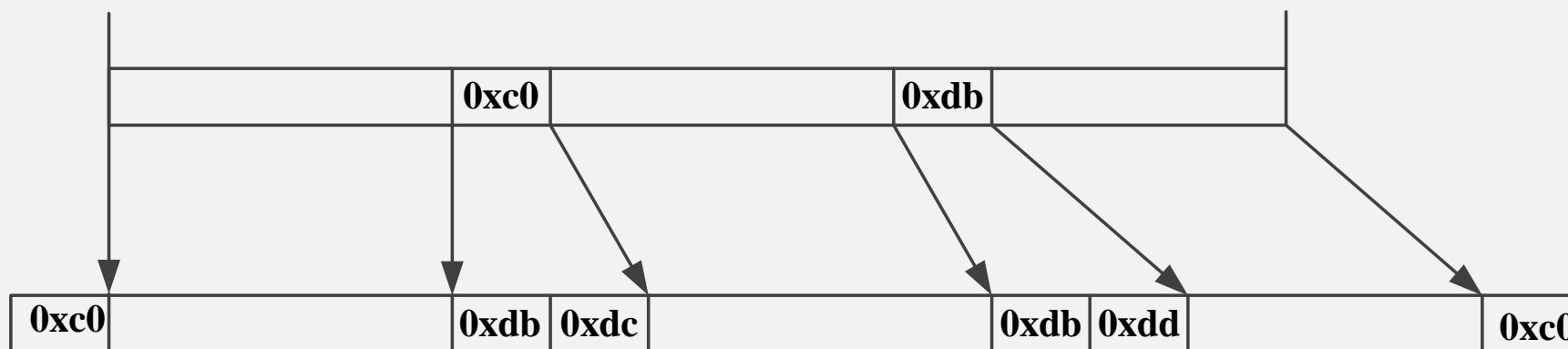
0 1 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 0

面向比特组帧示例

在发送端信息流中, 每出现连续的5个"1"就插入一个"0", 这样被插"0"的信息比特流中就不会有连续的"1"多于5个的比特串。接收端在收到5 个"1"以后, 如果收到的是"0"就将该"0"删去;如果是"1"就表示一帧结束

- 组帧

- 面向**字符**的组帧技术



面向字符的组帧示例

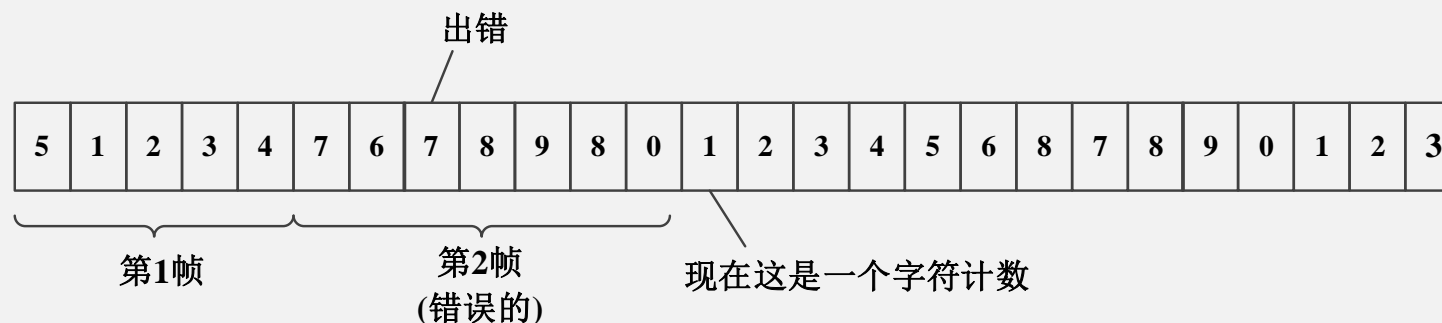
接收端只要收到 0xc0 字符，即表示一帧的开始或结束。每当遇到 0xdb 字符就进行字符转换，恢复数据中原有的 0xc0和0xdb 字符

# 数据链路控制

- 组帧

- 用长度计数的组帧技术

用数据头部的一个区域来指定该帧的字符长度，当接收端接收到这个长度信息后，就知道这一帧在哪里结束

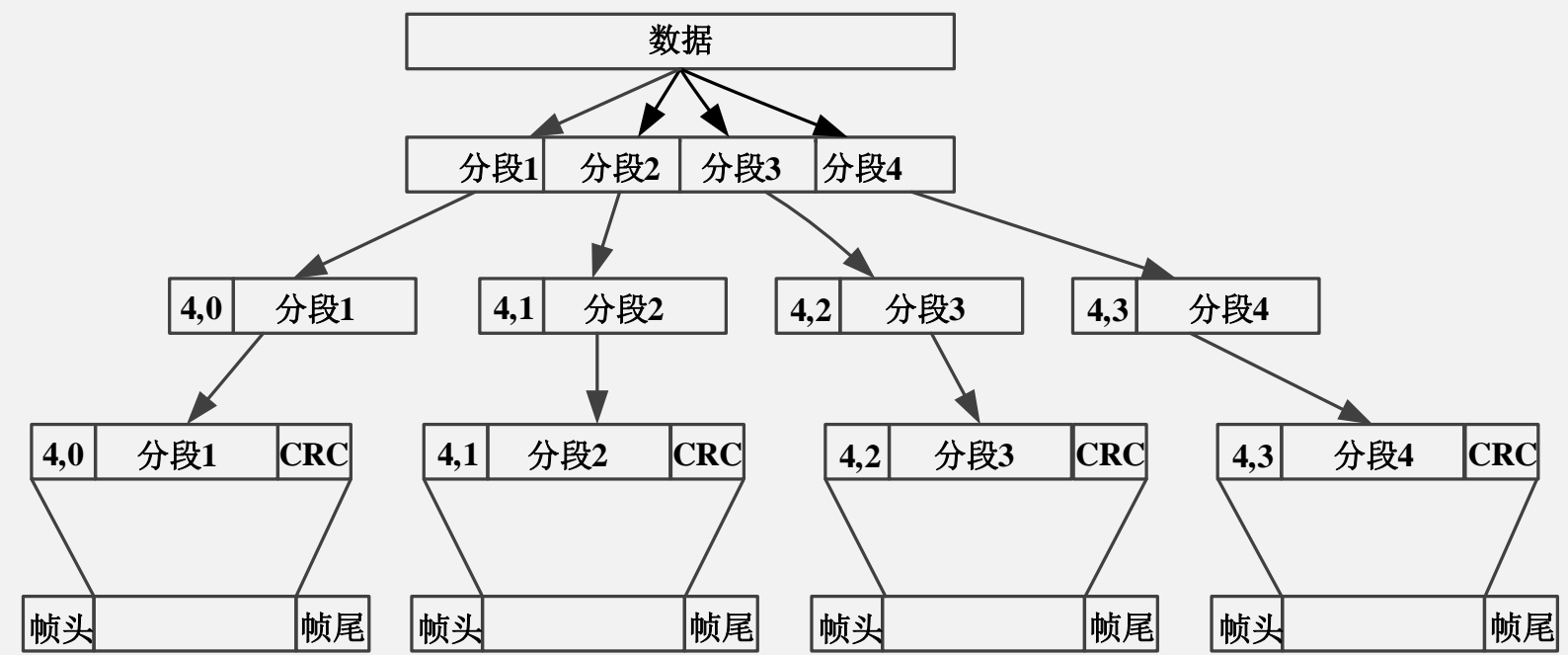


长度计数法示意图

# 数据链路控制

- 组帧

- 总结



数据组帧流程

无论用哪种方式，都会构建出含有帧头的新帧，有的还会含有帧尾

组帧是发送数据的过程中数据进入物理传输之前的一个操作。校验与组帧都是面向信道传输的，不是面向数据的，只要有发送请求，这两个操作就必须进行

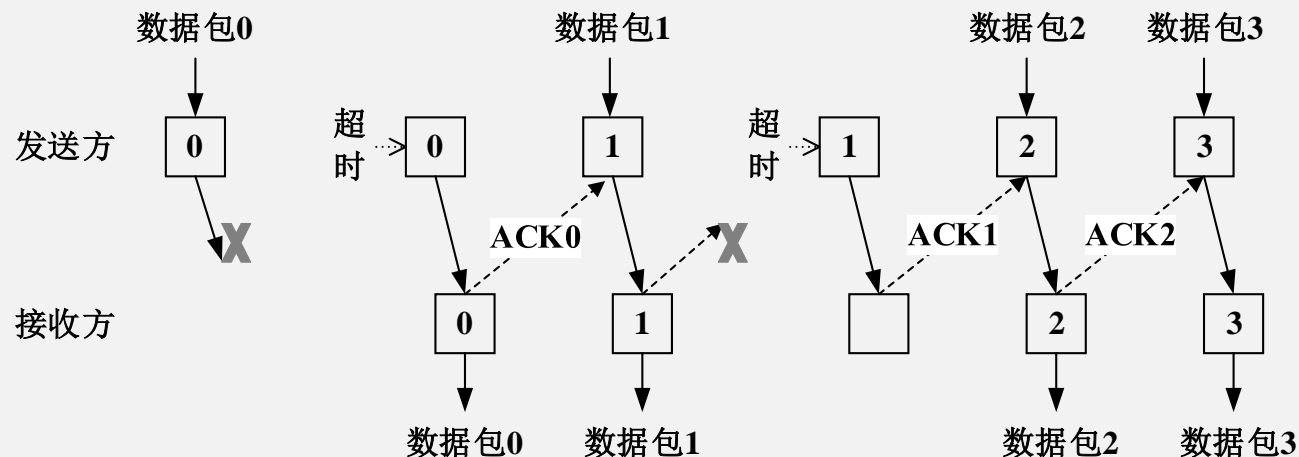
# 数据链路控制

- 差错控制

发送方发出一个帧后，接收方只有在正确接收到该帧后才会返回一个确认帧。如果到达接收方的是一个已经损坏的帧，则它将被丢弃。经过一段时间之后发送方将超时，于是它会再次发送该帧。这个过程将不断重复，直至该帧最终完好地到达接收方

- 停等式 ARQ

- 基本思想： 在开始下一帧传送之前，必须确保当前帧已被正确接收



停等式 ARQ示意图

假定 A发 B收。B 如果接收正确，则返回一个肯定的应答（ACK），否则不做任何应答。A 发送完数据帧后自动启动计时器，当收到接收节点的 ACK 后，停止计时器，进行下一个数据帧的发送。一旦计时器超时，则说明没有收到接收节点的 ACK，表明发送没有成功，发送节点立即重启计时器

- 差错控制

- 停等式 ARQ

- 发送序号和接收序号：为了防止ACK丢失或出错导致发送节点重发一次，接收节点再将该数据送到上层，导致的上层数据混乱
- 引入NAK：在重传机制中，因为 B不正确接收时不做任何表示，所以 A 只能等待，这会导致重传时延较大。如果 B在不正确接收时做出否定的回应（NAK），A 收到 NAK后立即重发，那么重传就不必等待至超时，整个传输时延会减少

需要接收节点能够知道发送节点发送了数据！

- 减少序号开销：可以采用 1bit 位来区分当前帧是否为新帧。当前帧正确接收时记录该位的值，下一次收到数据时，检测所收到的数据中该位的值是否与保存值相同，若相同则丢弃该数据，否则上传数据并更新保存值为当前值

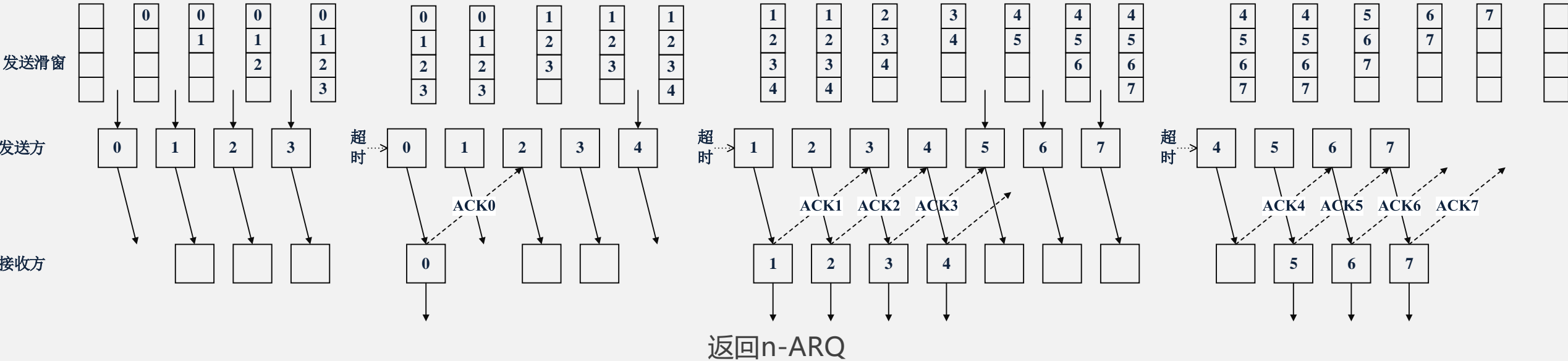
此为无编号 ARQ，目前被用于蓝牙系统中

# 数据链路控制

- 差错控制

- 返回 n-ARQ

一个 ACK 可确认多帧，即一次连续发送多帧，接收节点回应确认来告知发送节点实际的接收情况，可增加一次传输的数据量，从而提高传输速率。返回 n-ARQ 在没有收到接收节点应答的情况下，可以连续发送 n 帧，一旦收到返回的确认，则继续发送后续帧。发送节点保证任何时刻发送出去但未被确认的帧最多为 n 个。n=4 时如下图所示





- 差错控制

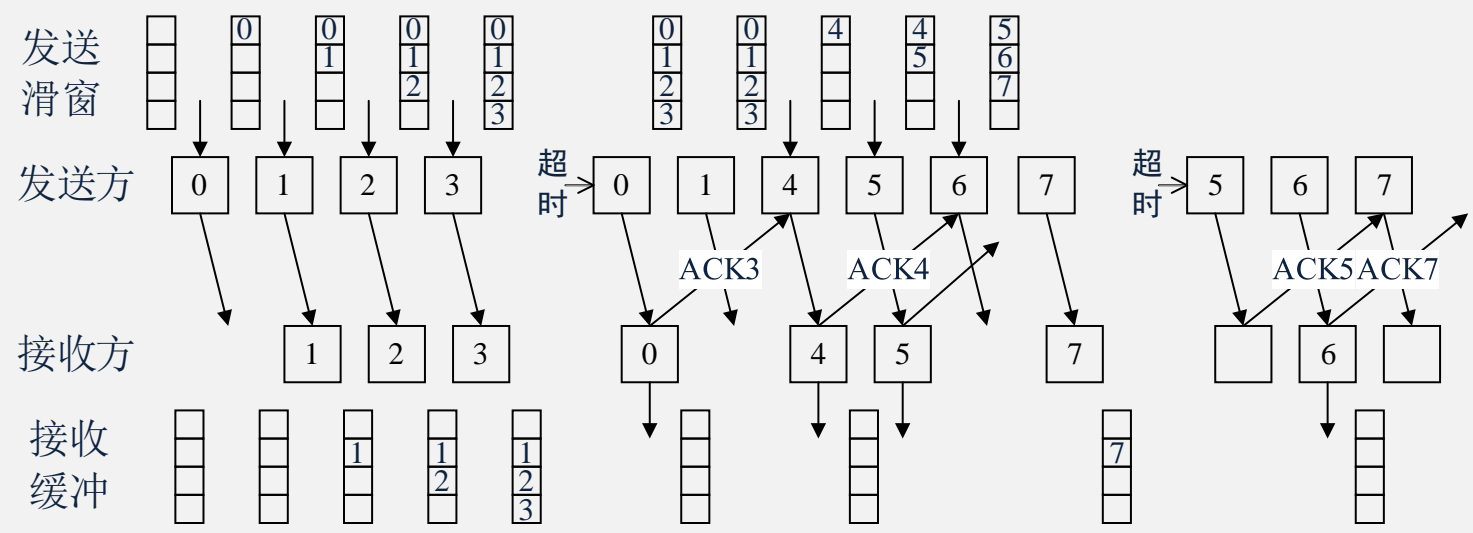
- 返回 n-ARQ

如果某序号帧丢失，那么接收节点不保存后面接收到的数据。这个方式使信道传输浪费太大。在接收节点，若前面序号的帧没有收到，则对接收到的帧按序号进行缓存。一旦收到前面序号的帧，就将所有收到的连续帧上传，收回的 ACK 序号为连续帧中最大的序号，发送节点处理数据的规则是认为 ACK 的序号及之前的帧都已被成功接收，从而可以加快发送节点的发送窗中序号的更新，减少不必要的重传

# 数据链路控制

- 差错控制

- 返回 n-ARQ



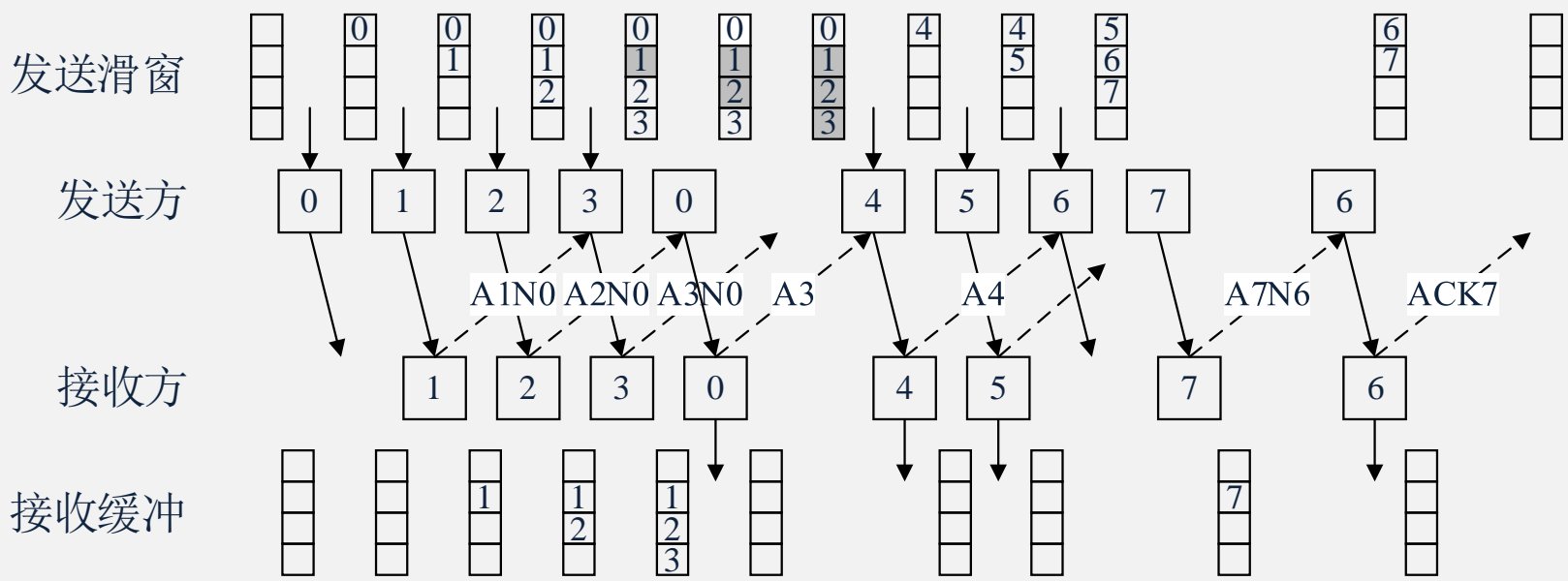
期望接收序号：发送节点收到确认信息后直接从期望接收序号开始设定滑窗

# 数据链路控制

- 差错控制

- 选择重发式 ARQ

与返回n-ARQ不同，接收节点不仅对接收到的帧进行确认，而且也对未正确接收的帧做出否定的回应（**NAK**）

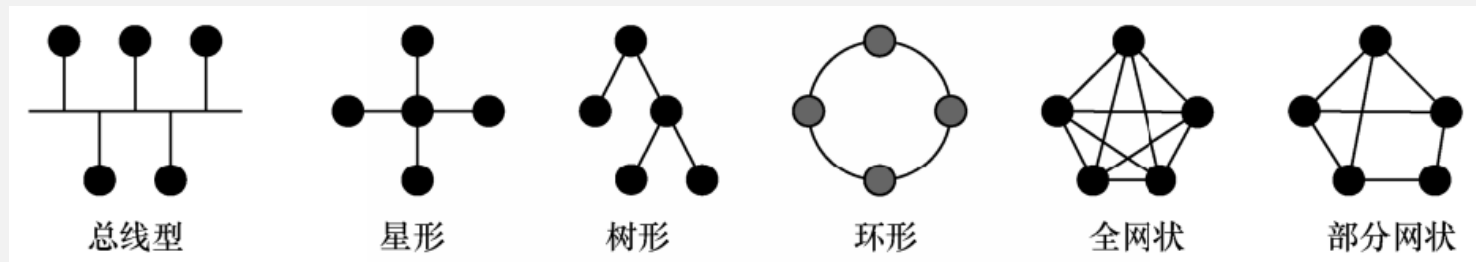


选择重发式ARQ

两个节点之间的通信：

- (1) 单工
- (2) 半双工
- (3) 全双工

实际应用中并不是只有一个节点，可能多个节点都会利用这个信道进行传输。通常是多个节点通过介质互通构成了网络，常见的网络拓扑结构如下



多个节点共享信道时通常采用半双工或全双工方式进行通信。某个节点发送数据时，其他多个节点都可能会接收到，如同广播一样，这种通信方式称为广播方式。由于共享信道本身采用广播方式，多个节点之间进行通信就会出现同时发送导致无法正确接收的情况。

介质访问控制重点：**若干节点如何共享这个介质？**

# 介质访问控制

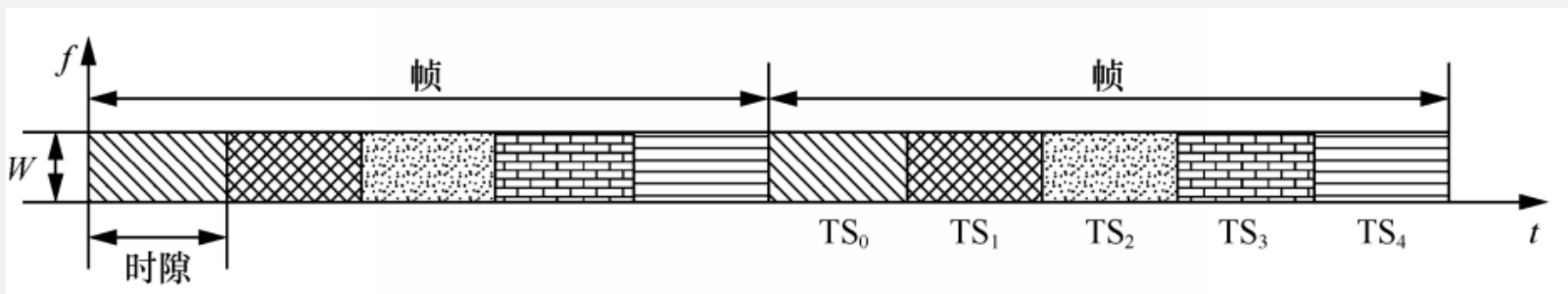
## ● 静态接入技术

指信道的接入方式是统一控制的或预先安排好的，各个节点会按照规定进行信道占用，不会发生两个或更多节点同时发送的情况。各用户可以进行无冲突的数据传输，但当用户没有数据传输时，其所分配的资源也不能分配给其他用户使用。此外，若系统中出现新增用户，也会因为没有分配资源而无法进行数据传输

典型的静态接入技术包括时分多址 (Time Division Multiple Access, TDMA)、频分多址FDMA和码分多址 (Code Division Multiple Access, CDMA )

### □ TDMA

首先将时间分为周期性的帧;每一帧再分为 $n$ 个时隙。这里要求帧和时隙互不重叠。然后按照一定的时隙分配原则，使用户只能在指定的时隙内进行数据传输



$n=5$ 时的TDMA 时帧结构

- 静态接入技术

- TDMA

- 根据一定的时隙分配原则将  $TS_0 \sim TS_4$  分配给 5 个用户使用，且所有用户在同一频带  $W$  内
- 一个用户可以占用一个或多个时隙进行数据传输
- 网络中所有用户的时隙必须对齐

如何做到所有用户的时隙对齐？

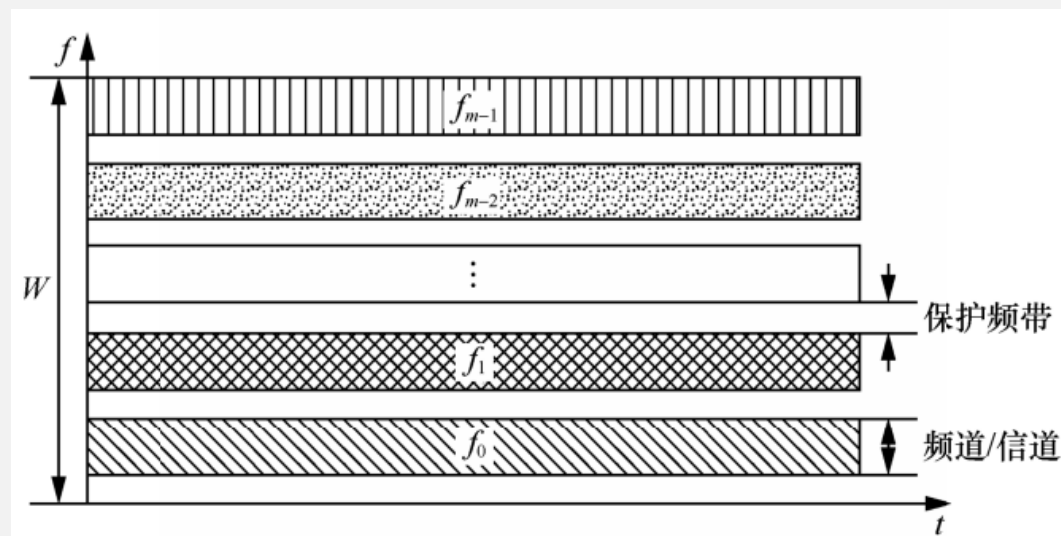
采用**分级同步方式**，即以最先通信的节点为中心节点；由**中心节点周期性地发送网络时戳**，其他节点根据中心节点的网络时戳调整本地时间并转发中心节点的网络时戳，最终达到全网时间同步

# 介质访问控制

- 静态接入技术

- FDMA

通信系统的总频段划分为  $m$  个等间隔的频道或信道，并且这些频道互不重叠；然后按照一定的频道分配原则，将这些频道分配给用户使用。下图为一个FDMA示意图，在系统总的频带  $W$  内划分了  $m$  个频道。为了防止不同频道间的信号干扰，频道间留有保护频带

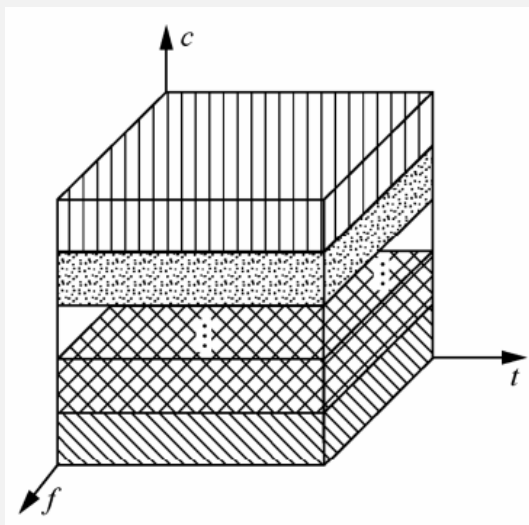


FDMA 频道划分示意图

- 静态接入技术

- CDMA

用不同码型的地址码来划分信道，每个地址码对应一个信道，不同的信道可以在同一个时间段内，同一个频段内通过不同的地址码进行区分。CDMA示意图如下



CDMA 信道划分示意图

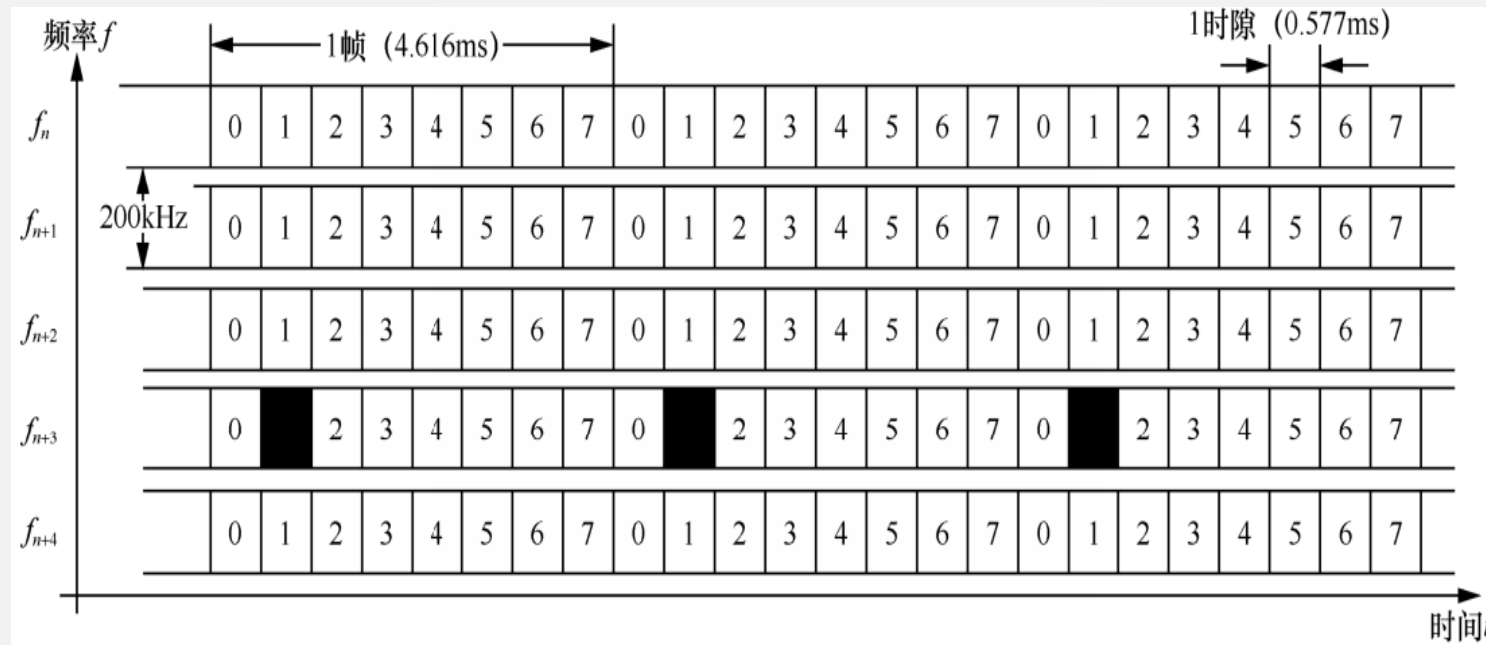


# 介质访问控制

- 静态接入技术

- TDMA/FDMA

结合 **TDMA** 和 **FDMA**，可增加接入的灵活性和效率。全球移动通信系统（Global System for Mobile Communications, GSM）采用FDMA 和TDMA混合的多址接入方式。



共有 $125 \times 8 = 1000$ 个物理信道，根据需要分配给不同的用户使用

GSM 的 TDMA/FDMA 帧结构

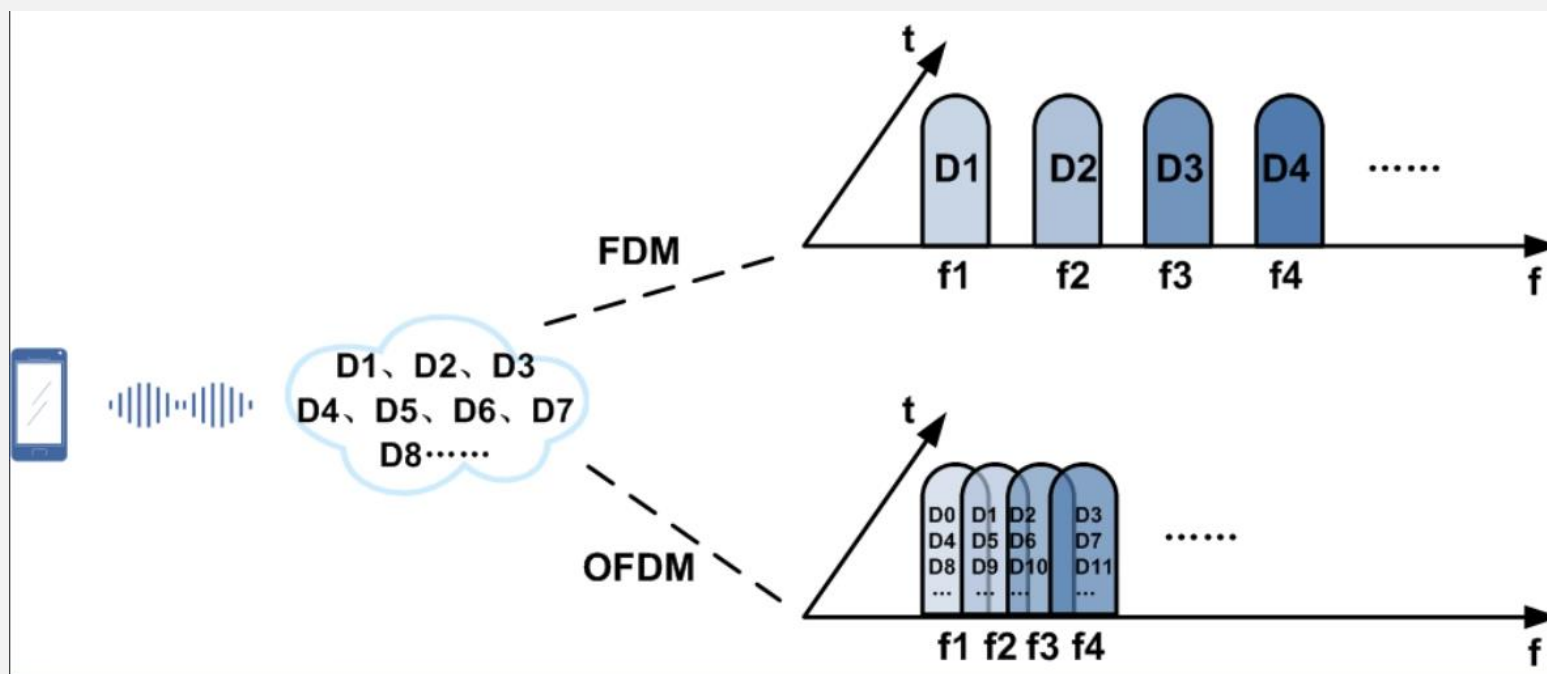
# 介质访问控制

- 静态接入技术

- OFDMA

OFDMA：将 OFDM 和 FDMA 技术结合形成的**正交**频分多址接入技术，是最常见的 OFDM多址技术

传统的FDM 的各个信道之间为了避免相互干扰，需要保留一定的保护带，有一定的频谱损失。而 OFDM 的各个子载波之间没有保护带，而且又是正交的，互相之间没有干扰，所以 OFDM 相比于 FDM有较大的频谱效率优势

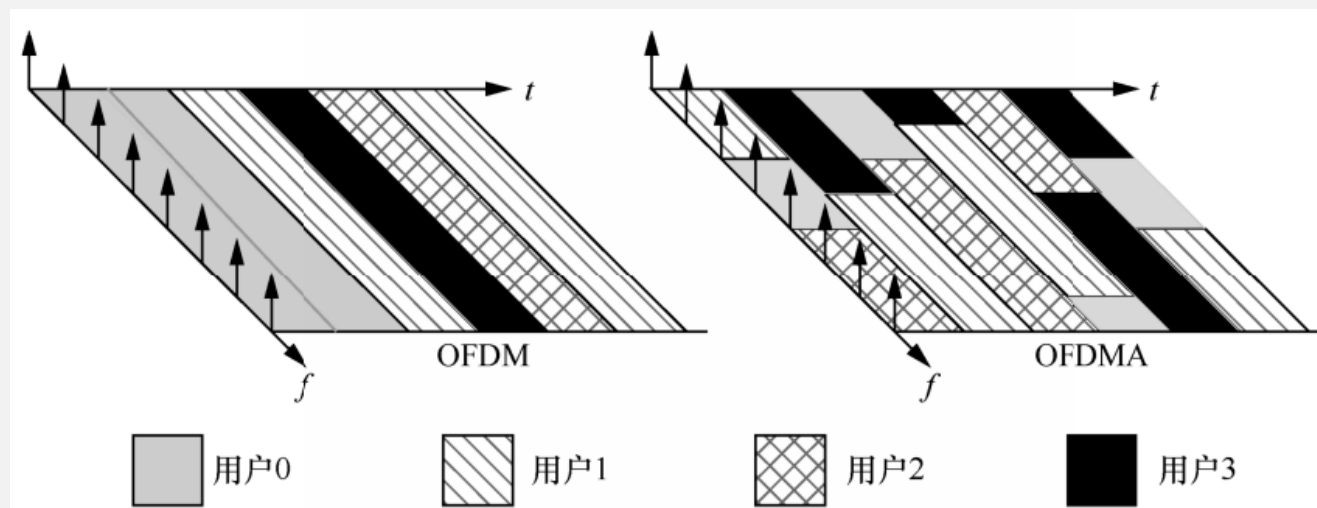


FDM 与 OFDM 频谱

- 静态接入技术

- OFDMA

OFDM 当中的每个子载波扩展后得到的序列叫作一个OFDM符号。在 OFDM 当中，可以把一部分子载波分配给一个用户，把另一部分子载波分配给另外的用户，从而作为多址的手段，称为 OFDMA，如下图所示



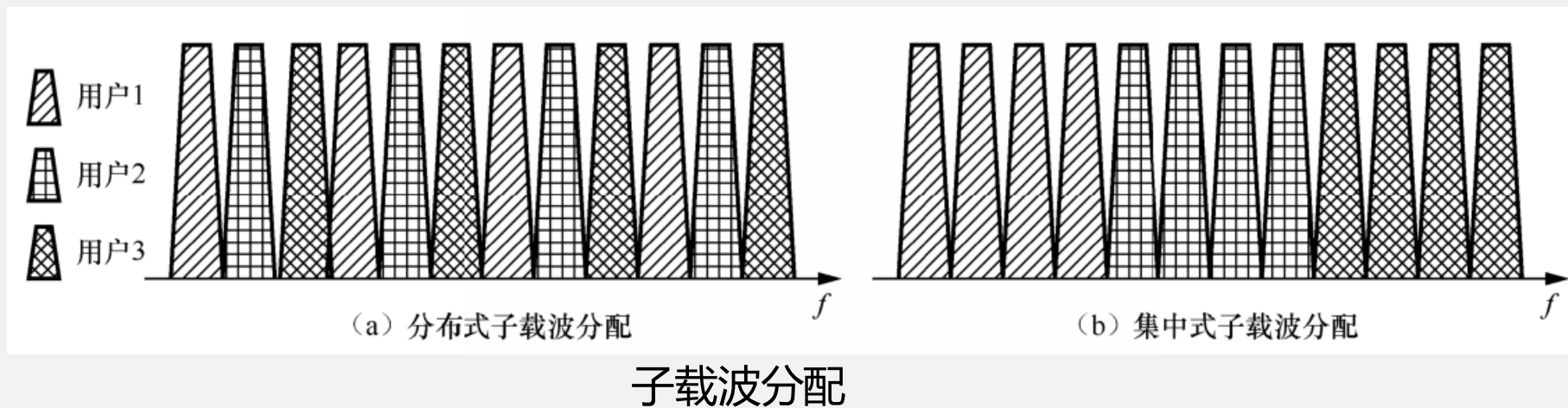
OFDMA 多址接入

- 静态接入技术

- OFDMA

OFDMA 又分为子信道 OFDMA 和跳频 OFDMA。

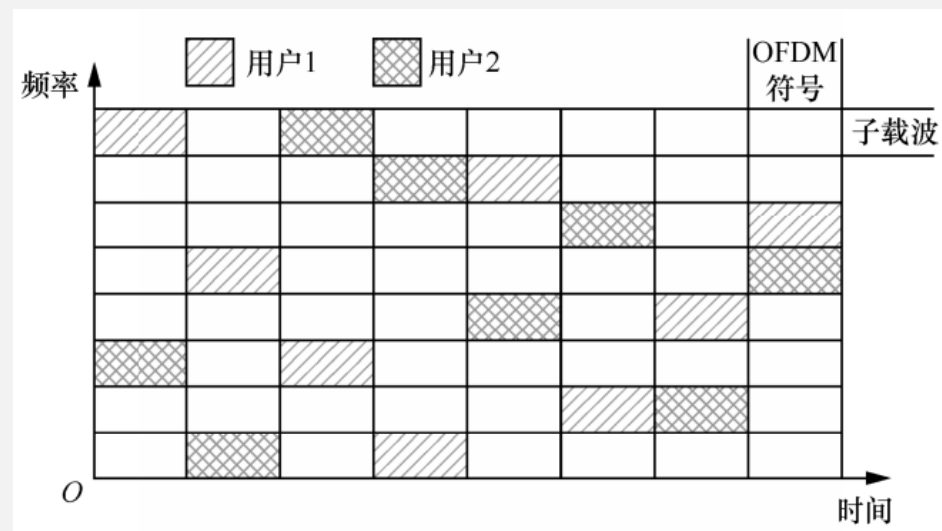
**子信道 OFDMA**：在子信道OFDMA系统中，将整个 OFDM系统的带宽分为若干个子信道，每个子信道包括若干个子载波，将它们分别分配给不同用户。一个用户也可以占用多个子载波。



- 静态接入技术

- OFDMA

**跳频 OFDMA**: 分配给一个用户的子载波资源会快速变化, 每个时隙中此用户在所有子载波中抽取若干子载波加以使用。同一时隙中, 各用户会选用不同的子载波组



跳频 OFDMA

# 介质访问控制

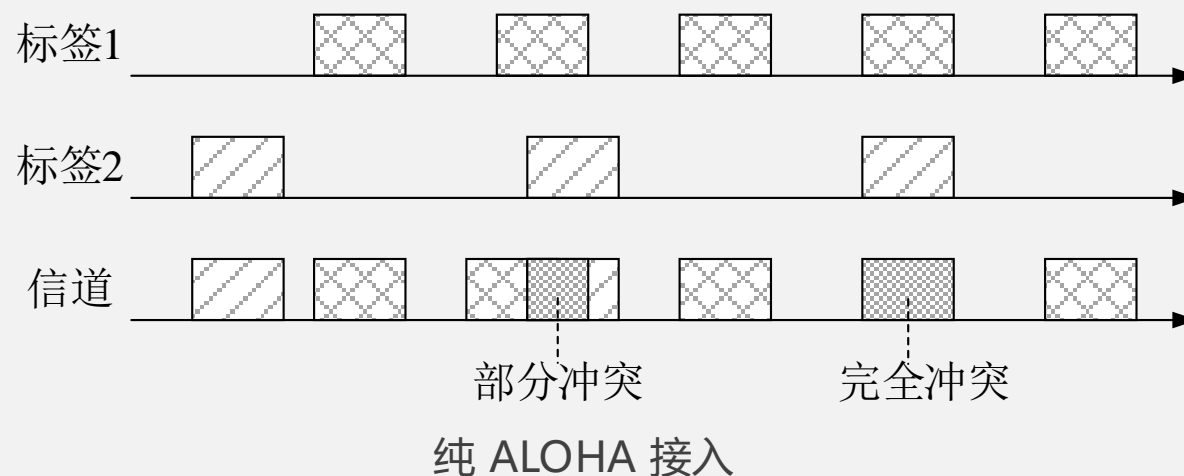
- 随机接入技术

- ALOHA

两个版本：**纯 ALOHA** 和**时隙ALOHA**。它们的区别在于，如果时间是连续的，那么就是纯 ALOHA;如果时间被分成离散时隙，所有帧都必须同步到时隙中，那么就是时隙 ALOHA

(1) 纯ALOHA基本思想

用户有数据需要发送时就传输



**问题：**可能会产生冲突，一旦冲突发生，冲突的帧将会被损坏

# 介质访问控制

- 随机接入技术

- ALOHA

- (2) 时隙 ALOHA

将时间分成离散的间隔，这种时间间隔称为时隙（Slot），所有节点将每个时隙作为"帧时"，这种方法要求用户遵守统一的时隙边界

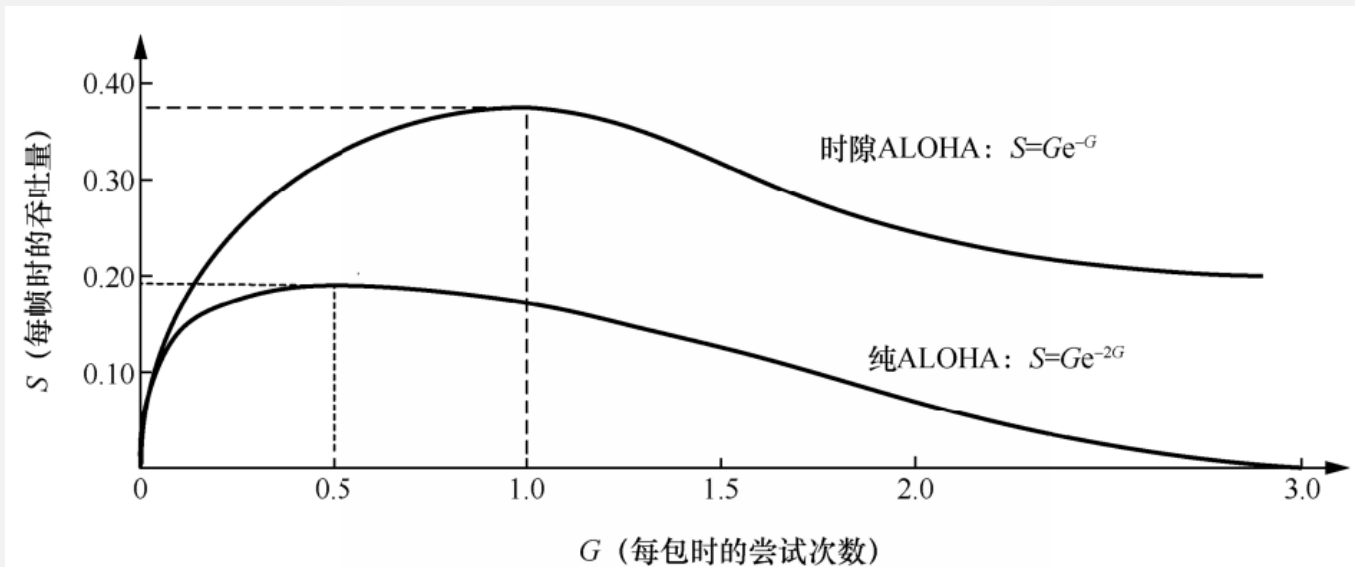
与纯ALOHA不同的是，在时隙ALOHA中，节点不允许用户立即发送帧。必须要等到**下一个时隙的开始时刻**。

# 介质访问控制

- 随机接入技术

- ALOHA

两种ALOHA的吞吐量与负载的关系如图所示



$$S = Ge^{-G}$$

ALOHA 系统的吞吐量与负载的关系

纯ALOHA最大的吞吐量出现在  $G=0.5$ 、 $S=1/2e$  时，大约等于**0.184**；

时隙ALOHA的尖峰出现在  $G=1$  处，此时吞吐量为  $S=1/e$ ，大约等于0.368，是**纯ALOHA的两倍**

<https://www.zhihu.com/question/59641832>



# 介质访问控制

- 随机接入技术

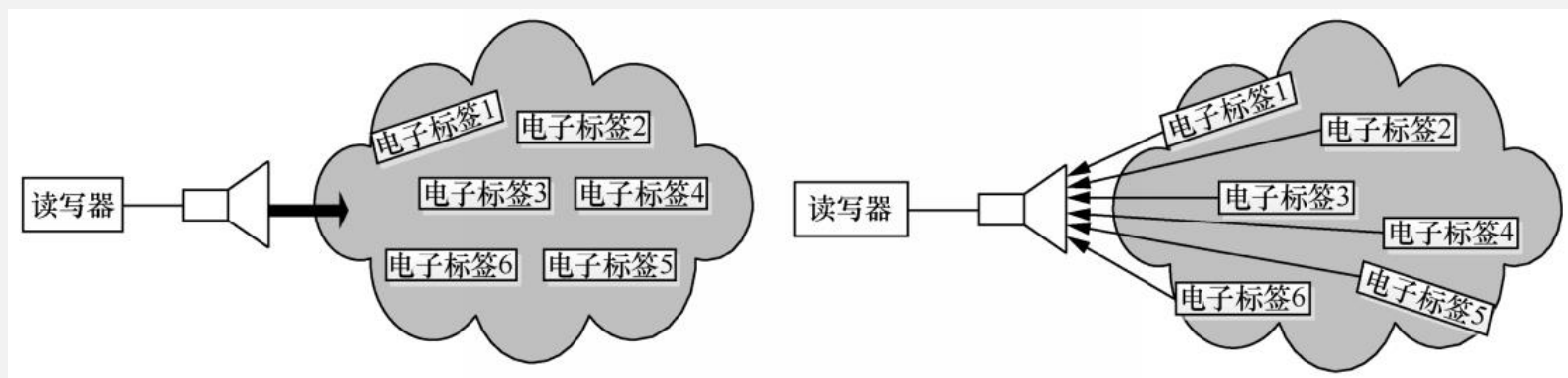
- ALOHA

时隙ALOHA解决了：1) **有线电视电缆访问因特网技术中**

2) **多个 RFID 标签和同一个 RFID 读写器通信时**

在多个竞争用户之间分配一条共享信道的问题

- RFID系统



RFID 网络结构

# 介质访问控制

- 随机接入技术

- 载波侦听多路访问协议

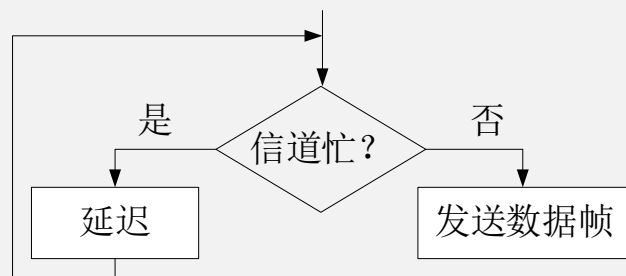
如果在一个协议中，节点监听是否存在载波（即是否有传输），并据此采取相应的动作，则这样的协议称为载波侦听多路访问协议（Carrier Sense Multiple Access, CSMA）

- **1-坚持 CSMA**：首先侦听信道，如果信道忙，它就**等待，直至信道变成空闲**。当节点发现信道空闲时，它传输数据的概率为 1。如果发生冲突，该节点等待一段随机的时间，然后再从头开始上述过程
- **非坚持 CSMA**：节点并不持续对信道进行监听，以便传输结束后立即抓住机会发送数据。相反，它会**等待一段随机时间**，然后重复上述算法
- **p-坚持 CSMA**：适用于分时隙的信道，首先侦听信道，如果信道是空闲的，则它**按照概率p发送数据**，而以概率  $g=1-p$  将此次发送推迟到下一个时间槽

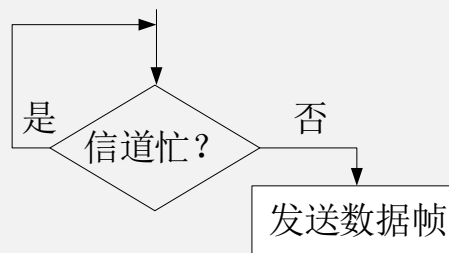
# 介质访问控制

- 随机接入技术

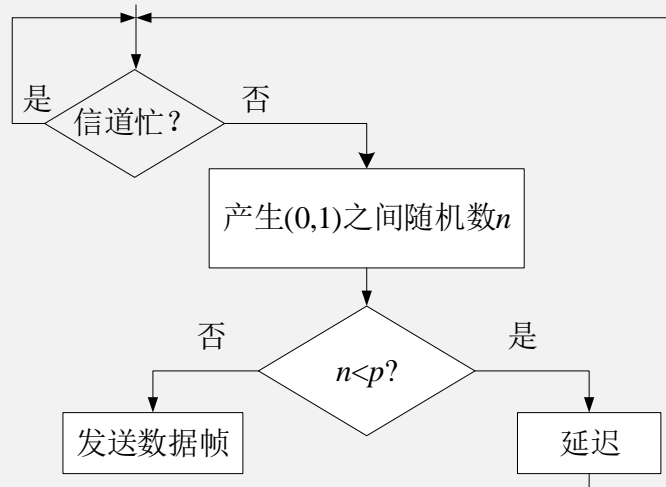
- 载波侦听多路访问协议



(a)



(b)



(c)

(a) 非坚持CSMA

(b) 1坚持CSMA

(c) p坚持CSMA

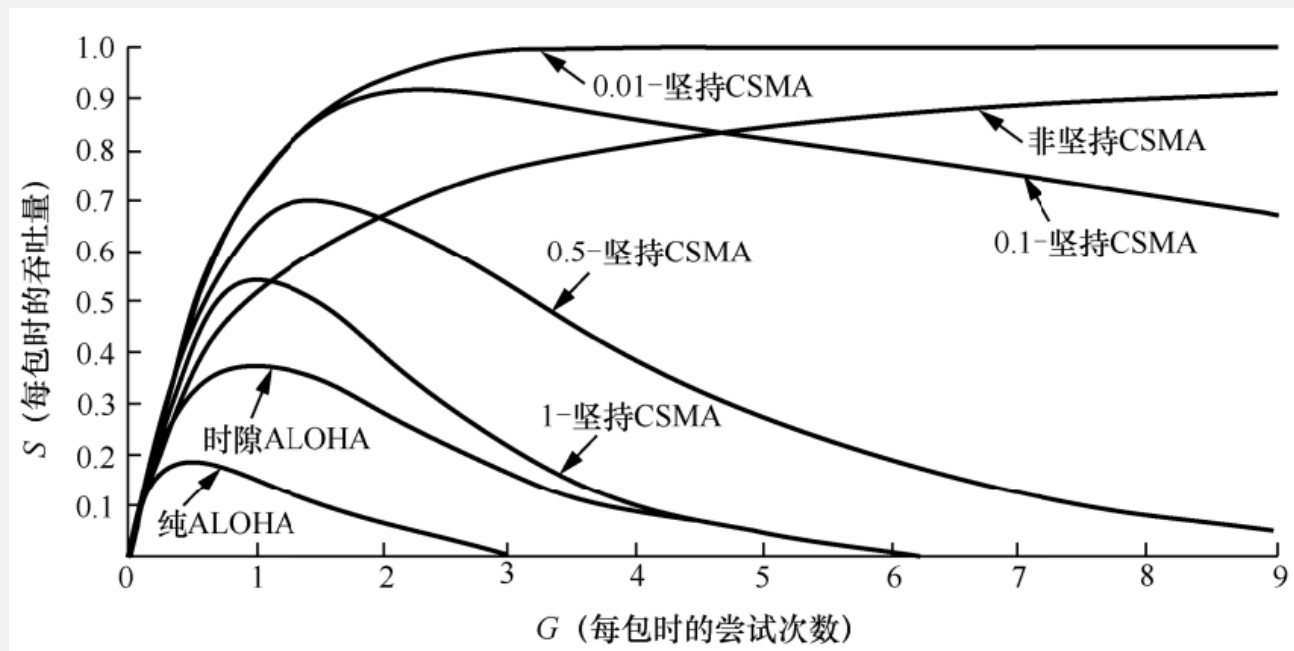
3种 CSMA 数据帧发送流程

# 介质访问控制

- 随机接入技术

- 载波侦听多路访问协议

上述3个协议以及纯 ALOHA 和时隙 ALOHA的可计算吞吐量与负载的关系



不同随机访问协议的信道吞吐量与负载的关系

# 介质访问控制

- 随机接入技术

- CSMA/CD

CSMA/CD (CSMA With Collision Detetion) 是一种带有**冲突检测**的 CSMA 协议。在该协议中，每个节点快速检测到发生冲突后会立即停止传输帧（而不是继续完成传输）

节省时间和带宽！

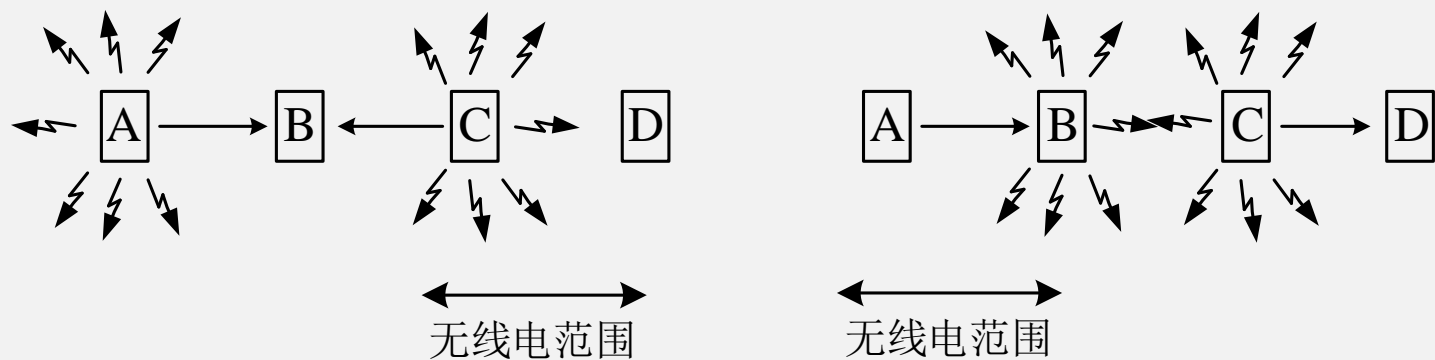
- 节点的硬件**在传输时必须侦听信道**。如果它读回的信号不同于它放到信道上的信号，则它就知道发生了碰撞
- 接收信号相比发射信号不能太微弱
- 必须选择能被检测到冲突的调制解调技术

# 介质访问控制

- 随机接入技术

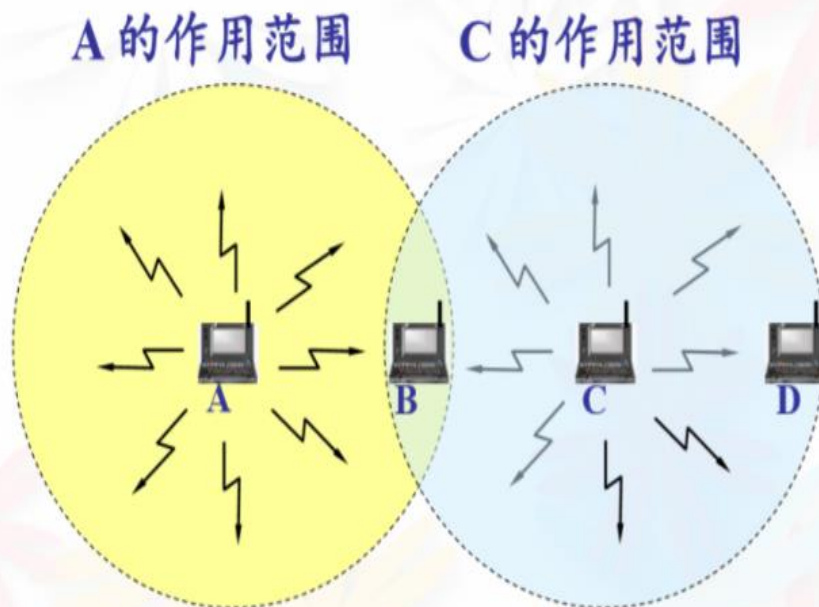
- 避免冲突的多路访问 (Multiple Access with Collision Avoidance, MACA)

在无线网络中，每个节点均会侦听是否有其他节点在传输，并且只有当没有其他节点在传送数据时它才传输。但对于无线信道来说，信号能量会随距离的增大而快速衰减，此时远距离的节点无法正常侦听到数据。这里冲突会发生在接收节点，而不是发送节点。造成了**隐藏终端问题**。此外，载波侦听也会使本可以进行的传输被禁止，造成**暴露终端问题**



无线网络示例

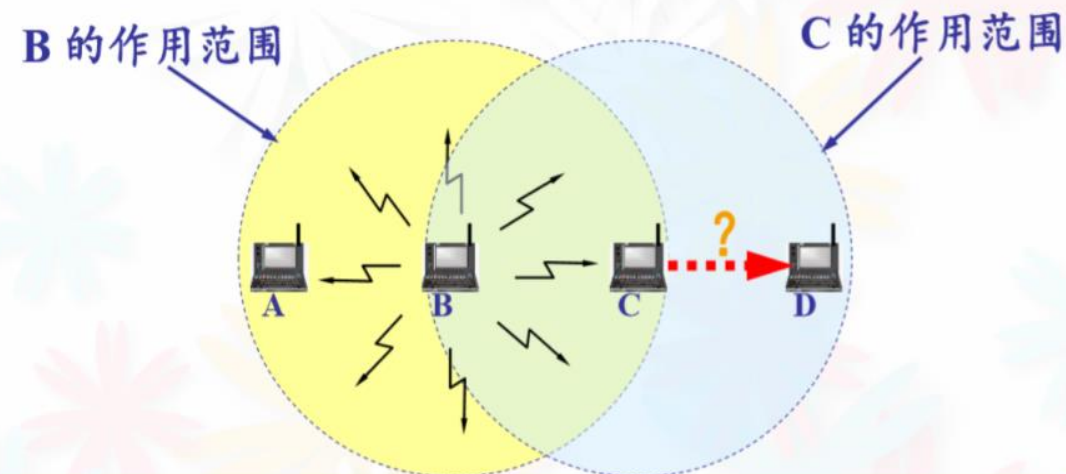
## 隐藏终端问题



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。

## 暴露终端问题

其实 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)



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。

# 介质访问控制

- 随机接入技术
  - 避免冲突的多路访问

MACA的基本思想是发送节点发送一个短帧给接收节点，以便其附近的节点能检测到本次传输，从而避免在后续的数据帧传输中也发送数据。

- RTS (Request To Send) 帧
- CTS (Clear to Send) 帧

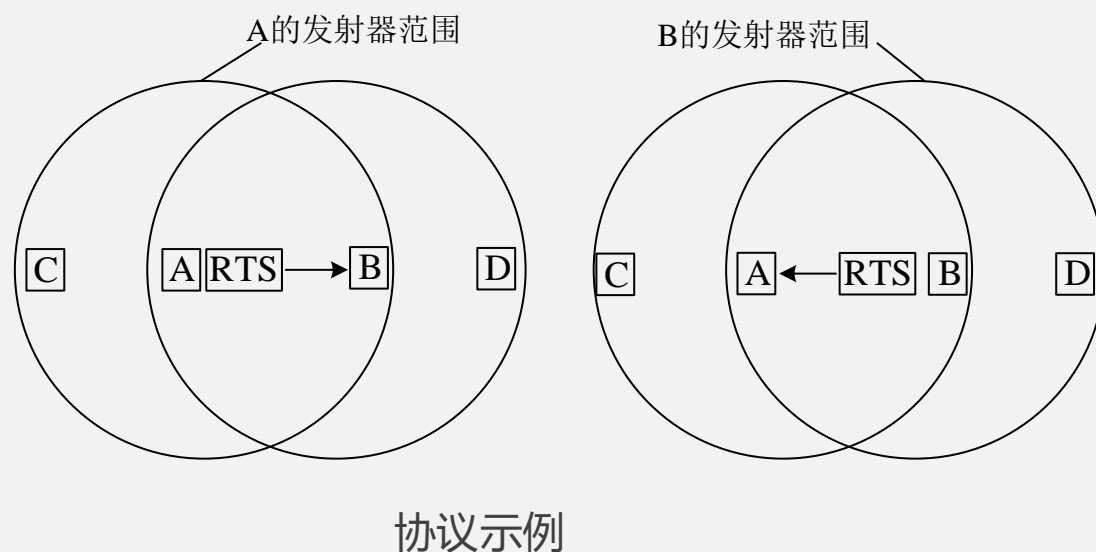
RTS帧包含了随后将要发送的数据帧的长度，CTS帧也包含了数据长度（从 RTS 帧中复制而来）



# 介质访问控制

- 随机接入技术

- MACA避免冲突的多路访问



A首先给 B 发送一个RTS帧，然后，B用一个CTS帧作为应答，A 在收到 CTS 帧之后便开始传输数据帧

如果一个节点收到了RTS 帧，表明它离A很近，此时它必须保持沉默，并等待足够长的时间，以便在无冲突情况下 CTS 帧被返回给A。如果一个节点只听到了CTS 帧，则表明它离B很近，在接下来的数据传送过程中它必须一直保持沉默，只要检查 CTS 帧，该节点就可以知道数据传输要持续多久

冲突仍有可能发生：例如，B和 C可能同时给A发送RTS 帧；无ACK

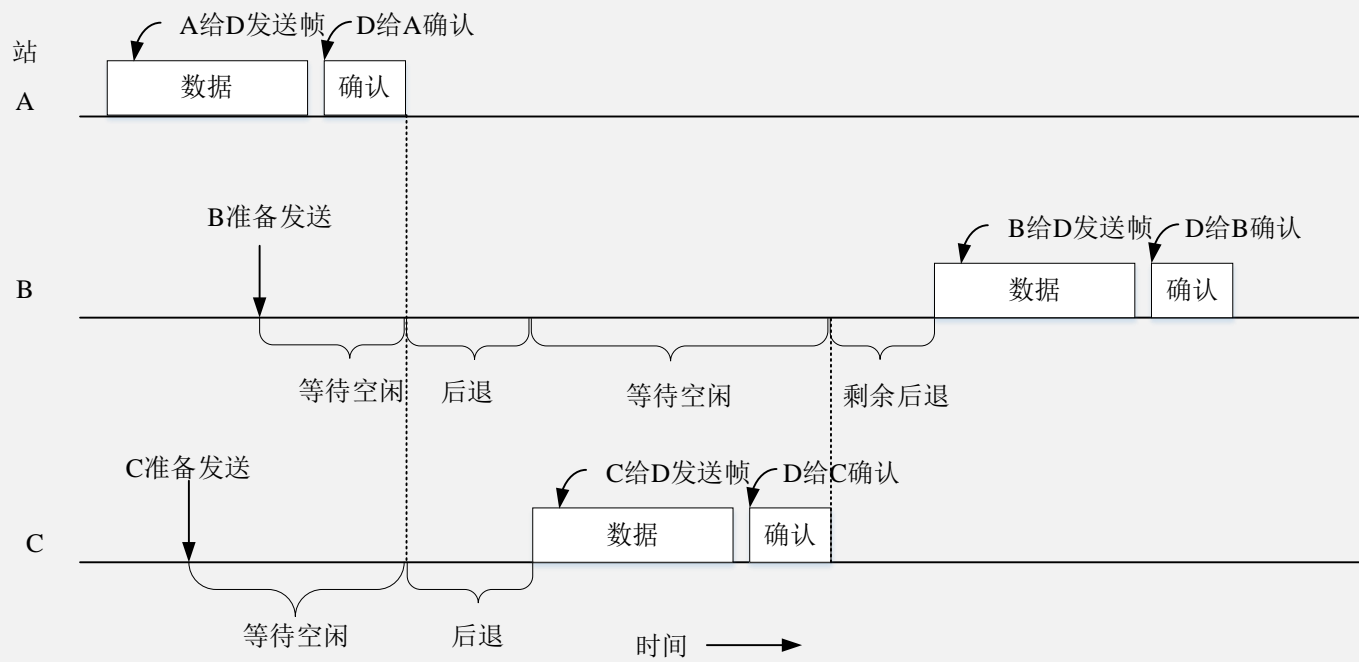
# 介质访问控制

- 随机接入技术

- CSMA/CA

**CSMA/CA** (CSMA with Collision Avoidance) 是一种带有冲突避免的 CSMA。WLAN 和 ZIGBEE系统都采用该协议进行分布式控制

发送节点在发送前侦听信道和在检测到冲突后**指数退避**，且发送节点要以随机退避开始



- 倒计时空闲时隙
- 冲突则加倍退避的时间

在 CSMA/CA机制下发送帧

- 随机接入技术

- CSMA/CA

为了保证 CSMA 正常工作，将载波分为两种：物理载波和虚拟载波

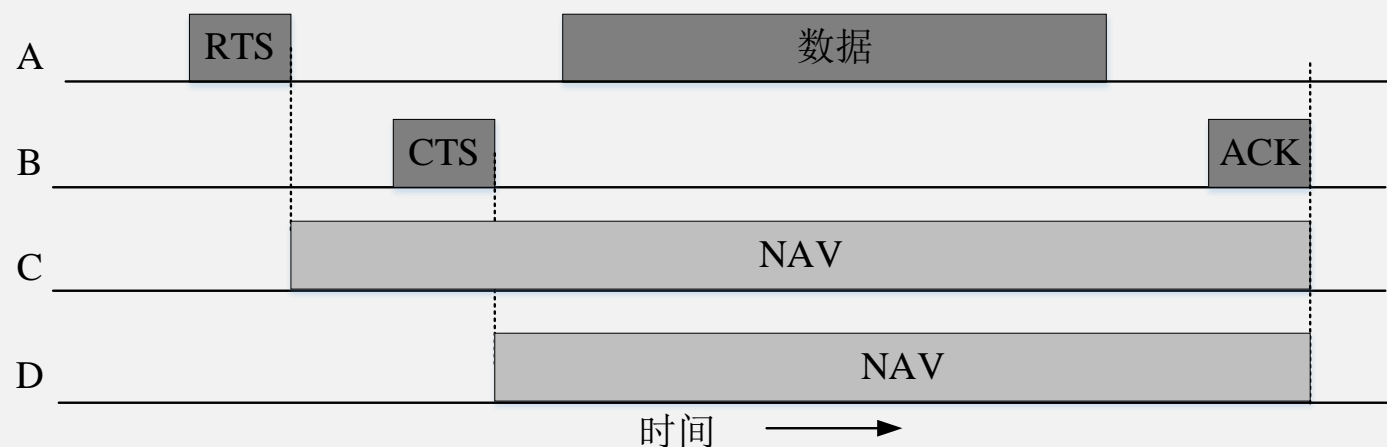
- **物理载波**：信道上的载波，主要根据载波能量来判定信道状态
- **虚拟载波**：表示当前时刻信道是否有节点在发送，监听的实质是每个节点可以保留一个信道何时要用的记录（可通过NAV获得）
- **NAV**：网络分配向量（Network Allocation Vector），一个数据帧的NAV给出了发送一个确认所需要的时间。所有听到该数据帧的节点将在发送确认期间推迟发送，而不管它们是否能听到确认的发送

# 介质访问控制

- 随机接入技术

- CSMA/CA

采用 RTS/CTS 机制使用NAV可以防止隐藏终端在同一时间发送，机制如下图所示



使用 CSMA/CA 的虚拟信道侦听

此例中，A想给B发送，C是A范围内的一个节点（也有可能在B范围内）。D在B范围内，但不在A范围内

# 介质访问控制

- 随机接入技术

- CSMA/CA

退避计数器：协议采用了**二进制指数退避**算法，每次发生冲突时，退避计数器的值加倍；每次交互成功时，退避计数器的值降至最小值

**二进制指数退避算法**：是指节点检测到信道空闲时间大于或等于 DIFS 或认为发生了分组碰撞，就依据均匀分布从  $[CW_{\min}, CW]$  ( $CW$ 为当前的碰撞窗口长度) 区间内随机选择一个数值计算退避时间，即：

$$\text{退避时间} = \text{随机数} \times \sigma$$

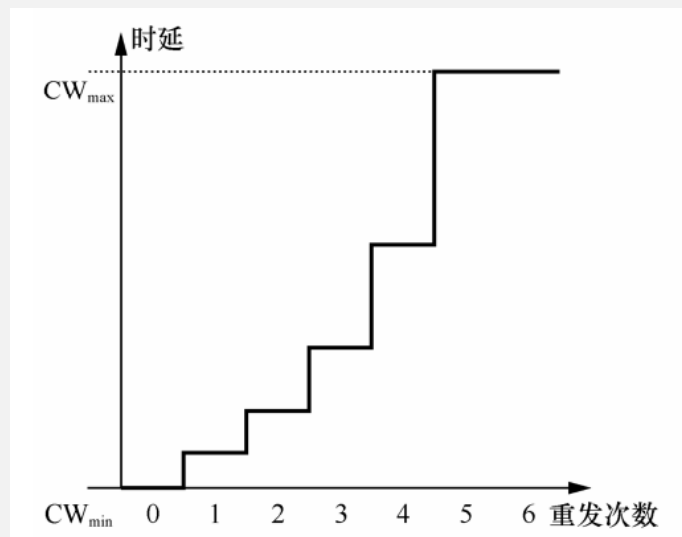
其中， $\sigma$ =传播时延+收发转换时间+载波监测时间。随机数 Random ( ) 为  $[CW_{\min}, CW]$  区间内的随机整数

# 介质访问控制

- 随机接入技术

- CSMA/CA

碰撞窗口长度CW：取决于重传的次数，在某业务分组第一次传输时，CW 等于最小碰撞窗口  $CW_{min}$ ，每次不成功传输都会使CW 增加一倍，直到最大碰撞窗口  $CW_{max} = 2^m \times CW_{min}$

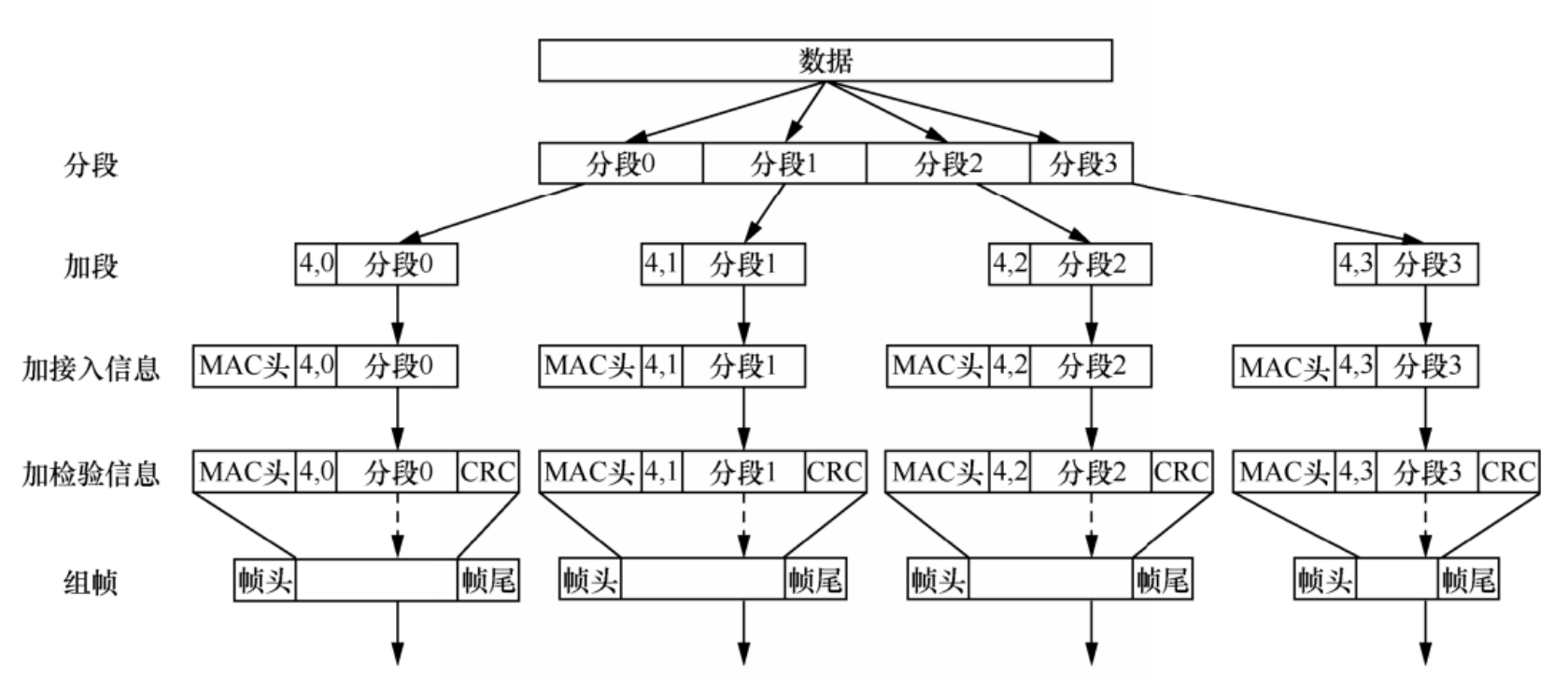


CW时延与重发次数关系示意曲线

# 总体架构

- 数据帧格式的变化流程

链路层协议对数据帧的处理包含分段、加段、加接入信息、加检验信息和组帧 5 部分



链路控制子层数据帧格式

- 本章主要内容：组帧技术、分段技术、差错检测与控制，静态接入技术与随机接入技术。
- 本章学习目标
  - 了解组帧技术和分段技术的特点；
  - 掌握常见**差错检测与控制**方法；
  - 熟悉静态接入技术与**随机接入**技术的典型方法与基本原理。
- **课后题：2,4,5（窗假设为4）**