第二章 2.3 MAC层技术介绍 - 备课笔记（逐字稿版）

课程信息

- **课程名称**：传感器网络技术与应用
- **章节**：第2章 2.3 MAC层技术介绍
- **课时**：6课时（300分钟），每2课时为1节课，共3节课
- **学生背景**：已学习2.2节物理层内容

第一节课（100分钟）

第1课时：2.3.1 数据链路层概述（50分钟）

【开场白】（2分钟）

同学们好！上一节我们完成了物理层的全部内容，大家还记得物理层主要解决什么问题吗？对，物理层解决的是"如何把比特从A传到B"的问题。

但是，光能传输比特还不够。我问大家一个问题：假设教室里有30个同学同时大声说话，你能听清楚每个人在说什么吗？肯定不行。无线传感器网络也一样，如果所有节点同时发送数据，信号就会混在一起，谁也收不到。

那怎么办呢？需要一套"交通规则"来协调大家有序地使用信道。这就是今天开始要学习的**数据链路层**，特别是其中的**MAC协议**。

【课程引入】（3分钟）

我们先回顾一下OSI七层模型。物理层是第一层，在它上面是第二层——数据链路层。

物理层提供的是一条"可能出错的、不可靠的物理连接"。比特在传输过程中可能出错、丢失、乱序。数据链路层的任务就是把这条不可靠的物理连接，改造成一条"逻辑上可靠的数据链路"。

打个比方：物理层是一条坑坑洼洼的泥路，数据链路层就是在这条泥路上铺了柏油，画了车道线，装了红绿灯，变成了一条有秩序的公路。

【第一部分：数据链路层定义与功能】（15分钟）

1.1 什么是数据链路层？

数据链路层位于物理层之上、网络层之下。它以**帧（Frame）**为单位传输数据。

物理层传的是比特流，就是一连串的0和1，没有结构。数据链路层给这些比特加上了"包装"——帧头、帧尾、地址信息、校验码等，形成了有结构的"帧"。

1.2 数据链路层的五大功能

数据链路层有五个核心功能，我们逐一来看。

第一个功能：封装成帧。

就是把网络层交下来的数据包，加上帧头和帧尾，变成一个帧。帧头里包含了源地址和目的地址，帧尾包含校验码。就像寄快递要装进快递盒，贴上寄件人和收件人地址一样。

第二个功能：透明传输。

什么叫透明传输？就是无论传什么内容，都能正确传输。听起来理所当然，但有个问题：如果数据内容里恰好包含了和帧定界符一样的比特序列怎么办？接收端可能会误认为是帧的结束。解决方法是**字节填充**或**比特填充**——在特殊字符前加转义符。

第三个功能：差错控制。

数据在传输过程中可能出错。数据链路层要能检测错误，甚至纠正错误。常用的方法有CRC循环冗余校验、前向纠错（FEC）和自动重传请求（ARQ）。

上节课学的奇偶校验也是一种检错方法，但能力有限。CRC的检错能力就强多了，实际网络中广泛使用。

第四个功能：流量控制。

发送方不能发得太快，不然接收方来不及处理。流量控制就是让发送方和接收方的速度匹配。最简单的方法是停止-等待协议：发一个帧，等确认，再发下一个。

第五个功能：链路管理。

包括链路的建立、维护和释放。就像打电话一样：先拨号建立连接，然后通话，最后挂断释放连接。

大家记住这五个功能：**封装成帧**、**透明传输**、**差错控制**、**流量控制**、**链路管理**。

【第二部分：帧结构与MAC地址】（10分钟）

2.1 帧结构

一个典型的数据帧长什么样呢?

- **帧首部**：包含帧定界符，告诉接收端"一个帧开始了"
- **目的MAC地址**：这个帧要发给谁
- **源MAC地址**：这个帧是谁发的
- **数据**：上层交付的数据，这是帧的"货物"
- **FCS**：帧校验序列（Frame Check Sequence），用于检错

FCS通常使用CRC算法计算。接收端收到帧后，用同样的算法计算一遍，如果结果不一致，就说明传输出错了。

2.2 MAC地址

每一个网络设备都有一个**MAC地址**，它是设备在数据链路层的身份标识。

MAC地址是48位，也就是6个字节。通常写成12位十六进制数，比如：`00:1A:2B:3C:4D:5E`。

MAC地址有两部分：

- 前24位（3字节）：**OUI**，由IEEE分配给厂商。比如苹果、华为各有自己的OUI
- 后24位（3字节）：由厂商自己分配给每个设备

所以，MAC地址在全球是唯一的。每台设备出厂时就固化了，一般不会改变。

大家可以查看自己手机或电脑的MAC地址试试。手机在WiFi设置的高级选项里可以找到。

【第三部分：IEEE 802参考模型】（10分钟）

3.1 为什么要分LLC和MAC?

IEEE 802标准把数据链路层分成了两个子层：

为什么要分？因为不同的网络（以太网、WiFi、蓝牙）使用不同的MAC协议，但上层应用不想关心底层用的是哪种网络。LLC子层提供统一接口，屏蔽了不同MAC的差异。

3.2 MAC子层的核心问题

MAC子层要解决的核心问题是：**多个节点如何共享同一个信道?**

有线网络中，可以用交换机把每个设备隔开，冲突不太严重。但在无线网络中，所有节点共享同一个无线信道，这就像一个大房间里所有人用同一个扩音器——你说话的时候别人就不能说，不然就乱了。

MAC协议就是来制定"谁什么时候可以说话"这个规则的。

【第四部分：差错控制简介】（5分钟）

简单介绍一下常用的差错控制方法：

CRC（循环冗余校验）： 目前最广泛使用的检错方法。发送端对数据进行特定的多项式除法运算，把余数附在数据后面。接收端做同样的运算，如果余数不为零，就说明出错了。CRC能检测所有单比特错误和大部分多比特错误，非常可靠。

FEC（前向纠错）： 不仅能检错，还能纠错。接收端不需要重传就能恢复正确数据。缺点是冗余开销大。

ARQ（自动重传请求）： 检测到错误后，请求发送方重新发送。简单有效，但需要反馈信道。

对于无线传感器网络，ARQ+CRC的组合最常用：用CRC检错，检测到错误后用ARQ重传。

【本课时小结】（5分钟）

好，我们来总结一下这节课学的内容：

1. **数据链路层**位于物理层之上，以帧为单位传输数据
2. **五大功能**：封装成帧、透明传输、差错控制、流量控制、链路管理
3. **MAC地址**：48位，全球唯一，是设备在数据链路层的身份
4. **IEEE 802模型**：将数据链路层分为LLC和MAC两个子层
5. **MAC子层**的核心问题：多个节点如何共享信道

下节课我们学习MAC协议的分类和静态MAC协议。请大家思考：如果让你来设计一个"说话规则"，你会怎么做？

第2课时：2.3.2 MAC协议分类与静态协议（50分钟）

【课前回顾与引入】（5分钟）

同学们好！上节课我们学习了数据链路层的基本功能，最核心的问题是什么？对，**多个节点如何共享信道**。

今天我们来看看前人想出了哪些方法来解决这个问题。先从MAC协议的发展历史说起。

【第一部分：MAC协议发展历史】（10分钟）

1.1 ALOHA协议——一切的起点

1968年，夏威夷大学的Norman Abramson教授面临一个问题：夏威夷群岛上的各个校区需要通过无线网络互联。他设计了世界上第一个无线计算机通信网络——ALOHAnet。

ALOHA协议的规则非常简单粗暴：**想发就发**。

具体来说：

- 节点有数据就直接发送
- 如果两个节点同时发送，就产生碰撞（冲突）
- 碰撞后，各节点随机等待一段时间后重发

这个方法简单吗？太简单了。效率高吗？不高。纯ALOHA的信道利用率只有约**18%**！也就是说，82%的信道容量都浪费在碰撞上了。

后来改进成了**时隙ALOHA**：把时间分成等长的时隙，节点只能在时隙开始时发送。这样信道利用率提高到了**37%**，翻了一倍。

虽然ALOHA效率不高，但它的意义是巨大的——它开创了**随机接入**的思想，是所有现代MAC协议的鼻祖。

1.2 MAC协议发展脉络

从ALOHA出发，MAC协议的发展路线是这样的：

```
ALOHA（1968） — 想发就发
    ↓
CSMA（1970s） — 先听后发
    ↓
CSMA/CD（1980s）— 边听边发，用于以太网
    ↓
CSMA/CA（1990s）— 避免碰撞，用于WiFi
    ↓
S-MAC/T-MAC（2000s）— WSN专用，节能优先
```

可以看到，每一代协议都是在前一代的基础上改进的。这个演进过程，就是我们这三节课要走的路线。

【第二部分：MAC协议分类】（15分钟）

2.1 分类方法一：静态 vs 动态

MAC协议最基本的分类是**静态**和**动态**：

静态MAC协议：预先分配资源，每个节点使用固定的资源，不会冲突。好比我们上课时，每个同学都有固定的座位，不会抢座。

动态MAC协议：按需竞争资源，灵活但可能冲突。好比食堂打饭，谁先到谁先打，但可能排队拥挤。

类型	优点	缺点	代表协议
静态	无冲突、可预测	不灵活、可能浪费	TDMA、FDMA、CDMA
动态	灵活、按需分配	可能冲突	CSMA/CD、CSMA/CA

2.2 分类方法二：基于竞争 vs 基于调度

还有一种分类方法，是根据信道分配方式来分：

基于竞争（Contention-based）：节点竞争信道使用权。谁抢到谁用。如果两个节点同时竞争，可能碰撞。

基于调度（Schedule-based）：由中心节点或分布式算法预先安排好，每个节点在指定的时间或频率上发送。

对于WSN，还有**混合型**：结合两者的优点。

2.3 WSN MAC协议全景图

我们把WSN用到的MAC协议画一张全景图：

```
WSN MAC协议
├── 基于竞争
│   ├── S-MAC（周期性睡眠）
```

```
│    ├── T-MAC（自适应睡眠）
│    └── SIFT（概率发送）
├── 基于调度
│    ├── TDMA（时分多址）
│    ├── TRAMA（流量自适应）
│    └── SMACS/EAR
└── 混合型
     └── Z-MAC（TDMA+CSMA）
```

今天先学习基于调度的静态协议，后面两节课学动态协议和WSN专用协议。

【第三部分：静态MAC协议】（20分钟）

3.1 TDMA——时分多址

TDMA = Time Division Multiple Access，时分多址。

原理很直观：把时间分成若干时隙，每个节点分配一个固定的时隙，只在自己的时隙内发送。

```
时间轴：
┌──A──┬──B──┬──C──┬──A──┬──B──┬──C──┐
│时隙1│时隙2│时隙3│时隙4│时隙5│时隙6│
└─────┴─────┴─────┴─────┴─────┴─────┘
|<──────  一个周期  ──────>|

节点A只在时隙1、4发送
节点B只在时隙2、5发送
节点C只在时隙3、6发送
```

大家想一想，这像什么？像不像轮流发言？老师说"A同学先说，然后B同学，然后C同学"，大家轮着来，不会抢话。

TDMA的优点：

- **无冲突**：每个节点有专属时隙
- **延迟可预测**：知道自己什么时候能发
- **节能**：不是自己的时隙可以休眠

TDMA的缺点：

- **需要同步**：所有节点的时钟要对齐，这在WSN中很难做到
- **不灵活**：如果某个节点没有数据，它的时隙就浪费了
- **需要中心调度**：谁来分配时隙？

3.2 FDMA——频分多址

FDMA = Frequency Division Multiple Access，频分多址。

原理：把可用频带分成若干子频带，每个节点分配一个固定的频率。所有节点可以同时发送，但使用不同频率，互不干扰。

就像收音机的不同电台，每个电台占一个频率，互不干扰。

FDMA的优点：简单、无碰撞、可同时通信。

FDMA的缺点：

- 频谱资源有限，不能支持太多节点
- 需要保护频带，频谱利用率低
- 对WSN来说，需要多频率收发器，增加成本和功耗

所以FDMA在WSN中用得**不多**。

3.3 CDMA——码分多址

CDMA = Code Division Multiple Access，码分多址。

还记得上一章学的扩频通信吗？CDMA就是扩频技术在多址中的应用。

原理：每个节点分配一个唯一的扩频码（称为"码片序列"）。所有节点可以在同一时间、同一频率上发送。接收端用对应的码字去"解码"，就能提取出想要的信号。

这就像在一个嘈杂的国际会议上，每对交流的人使用不同的语言。虽然声音混在一起，但你只听得懂自己的语言，其他语言对你来说就是噪声。

CDMA的优点：频谱效率高、支持节点多、抗干扰能力强。

CDMA的缺点：

- 实现复杂，硬件成本高
- 远近效应：距离近的节点信号强，会"淹没"远处节点的信号
- 对WSN来说，复杂度和功耗偏高

3.4 三种静态协议对比

让我们做个对比：

协议	分配的资源	优点	缺点	WSN中的适用性
TDMA	时间	无冲突、可休眠	需同步、不灵活	高

协议	分配的资源	优点	缺点	WSN中的适用性
FDMA	频率	简单	频谱浪费	低
CDMA	码字	效率高	复杂、功耗高	中

在WSN中，TDMA是最常用的静态MAC协议，因为它允许节点在非时隙期间休眠，这对省电很重要。

【本课时小结】（5分钟）

今天我们学习了MAC协议的发展历史和分类：

1. **ALOHA**（1968年）：开创随机接入，信道利用率低
2. **MAC协议分类**：静态/动态、竞争/调度
3. **TDMA**：时分复用，每个节点固定时隙，无冲突
4. **FDMA**：频分复用，不同频率，但频谱浪费
5. **CDMA**：码分复用，效率高但复杂

下节课我们进入动态MAC协议——首先是CSMA/CD，然后是CSMA/CA。重点来了，请大家预习隐蔽站和暴露站问题。

第二节课（100分钟）

第3课时：2.3.3 CSMA/CD协议（50分钟）

【课前回顾与引入】（5分钟）

同学们好！上节课我们学习了三种静态MAC协议：TDMA、FDMA和CDMA。它们共同的特点是什么？对，**预先分配资源，无碰撞**。

但是，静态MAC有个大问题：**不灵活**。如果某个节点暂时没有数据要发送，它分到的时隙（或频率）就白白浪费了。而另一个节点数据很多，却只能等自己的时隙。

有没有更灵活的方法？有，那就是**按需竞争**——需要发送时再去争取信道。这就是今天要学的CSMA。

从ALOHA的"想发就发"到CSMA的"先听后发"，关键的改进是什么呢？就是加了一个动作：**侦听**。

【第一部分：CSMA基本原理】（10分钟）

1.1 什么是CSMA?

CSMA全称是Carrier Sense Multiple Access，**载波侦听多路访问**。

核心思想只有四个字：**先听后发**。

在发送数据之前，先侦听信道。如果信道空闲，就发送；如果信道忙，就等待。这就像我们在会议上发言一样——先听听有没有人在说，没人说才开口。

1.2 三种侦听策略

信道忙的时候怎么办？有三种不同的策略：

1-坚持CSMA：死等。一直侦听信道，一旦空闲立刻发送。问题是：如果多个节点都在等，空闲后大家同时发送，碰撞概率很高。

非坚持CSMA：随机等。检测到信道忙，就不再侦听了，随机等待一段时间后再试。问题是：信道可能已经空闲了，但节点还在等，信道利用率低。

p-坚持CSMA：折中。信道空闲时，以概率p发送，以概率(1-p)等待一个时隙。这是前两种的折中方案。

1.3 CSMA的改进

CSMA比ALOHA好在哪里？好在"先听"这个动作。ALOHA想发就发，完全不管别人。CSMA至少先看看有没有人在说话。

但CSMA还是有问题：如果发送过程中和别人碰撞了怎么办？纯CSMA只能等发完了才知道碰撞了，浪费了整个发送时间。

能不能在发送过程中就检测到碰撞？这就是CSMA/CD的改进——**边听边发**。

【第二部分：CSMA/CD协议详解】（15分钟）

2.1 CSMA/CD的含义

CSMA/CD = CSMA with **Collision Detection**，带碰撞检测的CSMA。

它是**有线以太网**（IEEE 802.3）的标准MAC协议。家里的网线、公司的局域网，底层用的都是CSMA/CD。

2.2 三大特征

CSMA/CD有三个关键特征，请大家牢记：

第一，先听后发。 发送前先侦听信道是否空闲。这和CSMA一样。

第二，边听边发。 发送数据的同时，继续侦听信道。这是CSMA/CD独有的。如果检测到信号异常（比如电压变化），就知道碰撞了。

第三，碰撞停止。 一旦检测到碰撞，立即停止发送，并发送一个短的Jam信号通知所有节点。然后进入退避等待。

为什么"边听边发"很重要？因为一旦检测到碰撞就停止，可以大大减少信道浪费。如果不检测，整个帧发完才发现碰撞，白白浪费了整个发送时间。

2.3 工作流程

让我把整个流程讲清楚：

第一步，节点准备发送数据。

第二步，侦听信道。如果信道忙，就持续等待（1-坚持策略）。如果信道空闲，进入下一步。

第三步，开始发送帧，同时继续侦听信道。

第四步，检查是否有碰撞。

- 如果没有碰撞：发送完成，成功！
- 如果检测到碰撞：立即停止发送，发出Jam信号。

第五步（碰撞后），执行二进制指数退避算法：

- 计算退避时间
- 等待后回到第二步重试
- 如果重试超过16次，放弃并报告上层

整个过程就是：听→发→听→碰撞？→停→等→重试。

【第三部分：CSMA/CD的致命缺陷——不适合无线网络】（15分钟）★★

现在到了最关键的部分：**为什么CSMA/CD不能用在无线网络中?**

3.1 问题一：碰撞检测困难

在有线网络中，碰撞检测很容易——通过检测线缆上的电压变化就行。但在无线网络中呢?

无线信号有一个特点：**自身发射的信号比接收到的远处信号强得多**。可能强1000倍甚至100万倍！

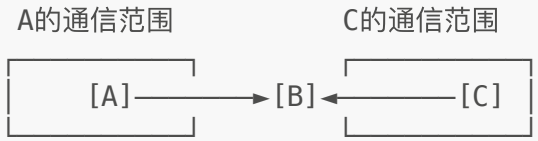
这就好比你大声喊叫的时候，能听清楚远处别人的小声说话吗？听不清！所以，无线设备在发送时，很难检测到其他设备的信号。

另外，无线设备通常是**半双工**的——同一时刻只能发送或接收，不能同时进行。这也让碰撞检测变得不可能。

3.2 问题二：隐蔽站问题（Hidden Terminal）★★

这是一个非常重要的概念，考试经常考。

场景是这样的：有三个节点A、B、C。B在中间，A在左边，C在右边。A和C都在B的通信范围内，但**A和C互相听不到**。



现在，A开始给B发送数据。C不知道A在发送（因为听不到A），C侦听信道，发现是空闲的，于是C也开始给B发送。结果呢？B同时收到A和C的信号，碰撞了！

C对A来说就是一个**隐蔽站**——A看不到C，C也看不到A，但它们都和B通信，导致了冲突。

这就像两个人从走廊两头走来，中间有个拐角，看不到对方，结果撞到一起了。

3.3 问题三：暴露站问题（Exposed Terminal）★★

再看另一个场景：四个节点A、B、C、D排成一排。B正在给A发送数据，C想给D发送数据。

C能听到B的发送，所以C检测到信道"忙"，于是C推迟了发送。

但实际上，C发送给D根本不会影响B发送给A！因为D离B太远，B的信号传不到D那里。C的推迟是**不必要的**。

B对C来说就是一个**暴露站**——C因为听到了B，而不必要地推迟了自己的传输，导致信道利用率降低。

3.4 总结：为什么CSMA/CD不适合无线

三个原因：

问题	说明
碰撞检测难	自身信号太强，掩盖远处信号
隐蔽站	听不到的节点可能正在发送
暴露站	听得到不代表一定会干扰

那怎么办？答案就是下一个课时要学的**CSMA/CA**——不检测碰撞，而是**避免碰撞**。

【本课时小结】（5分钟）

今天我们深入学习了CSMA/CD：

1. **CSMA**：先听后发，三种侦听策略
2. **CSMA/CD三大特征**：先听后发、边听边发、碰撞停止
3. **隐蔽站问题**：A和C互相听不到，同时给B发送导致碰撞 ★
4. **暴露站问题**：C不必要地推迟了发送 ★
5. **结论**：CSMA/CD不适合无线网络

请大家记住隐蔽站和暴露站的场景，下节课我们看看CSMA/CA是如何解决这些问题的。

第4课时：2.3.4 CSMA/CA协议（50分钟）

【回顾与引入】（5分钟）

上节课我们知道了CSMA/CD在无线网络中行不通，主要原因是碰撞检测困难和隐蔽站问题。

既然碰撞"检测不了"，那我们换个思路：**碰撞检测不了，我就提前避免碰撞**。

这就是CSMA/CA的核心思想——Collision **Avoidance**，碰撞**避免**。

CSMA/CA是WiFi（IEEE 802.11）的标准MAC协议。大家每天用WiFi上网，底层就是CSMA/CA在工作。

【第一部分：CSMA/CD vs CSMA/CA】（5分钟）

先做个对比，帮大家理解两者的区别：

特性	CSMA/CD	CSMA/CA
全称	碰撞检测	碰撞避免
策略	碰了再说	提前预防
适用	有线网络	无线网络
信道空闲时	立即发送	随机等待后发送
确认机制	无ACK	有ACK
隐蔽站	无法解决	RTS/CTS解决

可以这样记：CSMA/CD像"先斩后奏"——碰撞了再处理；CSMA/CA像"先礼后兵"——先打招呼再发送。

【第二部分：帧间间隔IFS】（10分钟）

CSMA/CA引入了一个重要概念：**帧间间隔（IFS，Inter-Frame Space）**。

不同类型的帧有不同的优先级，优先级通过等待时间来体现——优先级高的帧等待时间短，优先级低的帧等待时间长。

主要有三种IFS：

SIFS（Short IFS，短帧间间隔）：最短，优先级最高。用于ACK确认帧和CTS帧。因为确认帧很重要，必须优先发送。

DIFS（DCF IFS，分布式协调帧间间隔）：较长。普通数据帧在发送前要等待DIFS时间。

EIFS（Extended IFS，扩展帧间间隔）：最长。用于错误恢复。

时间关系是：**SIFS < DIFS < EIFS**

这意味着什么呢？当信道刚变空闲时：

- ACK/CTS只需等SIFS就可以发送（最快）
- 普通数据帧要等DIFS才能发送（较慢）
- 这样就保证了确认帧优先于数据帧

【第三部分：RTS/CTS握手机制】（15分钟）★★

RTS/CTS是CSMA/CA最精华的部分，也是解决隐蔽站问题的关键。

3.1 RTS和CTS是什么？

- **RTS = Request To Send**，请求发送。发送方在发数据之前，先发一个短小的RTS帧，相当于"举手示意：我要说话了"。
- **CTS = Clear To Send**，清除发送。接收方收到RTS后回复CTS，相当于"请说吧"。

3.2 完整的发送流程

我来一步步讲解：

第一步：发送方A想给接收方B发送数据。A先侦听信道，等待DIFS时间后，发送**RTS帧**。RTS帧很短，包含：谁发的、发给谁、预计传输多长时间。

第二步：B收到RTS后，等待SIFS时间，回复**CTS帧**。CTS帧也很短，同样包含预计传输时长。

第三步：A收到CTS后，等待SIFS时间，开始发送**数据帧**。

第四步：B收到完整的数据帧后，等待SIFS时间，回复**ACK确认帧**。

整个时序是这样的：

```
A（发送方）：——DIFS——|—RTS—|         |————DATA数据帧————|
                              ↘         ↗                    ↘
B（接收方）：     SIFS→   |—CTS—|                        |—ACK—|

时间轴 →→→
```

3.3 如何解决隐蔽站问题？

还记得隐蔽站的场景吗？A和C互相听不到，但都和B通信。

有了RTS/CTS之后：

```
[A]————————►[B]◄————————[C]
    RTS→       ←CTS→C也收到了！
```

第一步，A发送RTS给B。C听不到这个RTS（因为A和C不在范围内）。

第二步，B回复CTS。关键来了——**C能收到B发出的CTS**！因为C在B的通信范围内。

第三步，C从CTS中读到传输时长信息，知道信道即将被占用，于是设置一个计时器（NAV），在此期间**保持静默**，不发送任何数据。

第四步，A安心发送数据给B，不会被C干扰。

隐蔽站问题解决了！关键在于：**虽然C听不到A，但C能听到B的CTS，从而知道要保持安静。**

3.4 NAV机制

刚才提到的NAV = **Network Allocation Vector，网络分配向量**。

NAV是一种**虚拟载波侦听**机制。节点收到RTS或CTS帧后，从中读取"传输时长"信息，设置一个NAV计时器。在NAV倒计时期间，节点认为信道"忙"，不去侦听也不发送。

NAV的好处是什么？**省电**。节点不需要持续侦听信道，只要设了NAV就知道信道什么时候空闲，这段时间可以休息。

```
NAV示意：

信道上：—RTS—CTS————DATA数据————ACK—
其他节点：[————————— NAV计时 —————————]
            不需要侦听，保持静默
```

【第四部分：二进制指数退避算法】（10分钟）

4.1 为什么需要退避?

CSMA/CA规定：即使信道空闲，也不能立即发送，要**随机等待**一段时间。

为什么？因为如果多个节点同时等待信道变空，然后同时发送，还是会碰撞。随机等待可以"错开"各节点的发送时间。

4.2 退避算法详解

退避时间是这样计算的：

第一步，确定竞争窗口CW。初始值是CWmin，比如15。

第二步，从0到CW之间随机选一个数作为退避计数器。比如选了8。

第三步，每过一个时隙，退避计数器减1。减到0就可以发送。

第四步，如果发送成功，CW恢复为CWmin。

第五步，如果碰撞了，CW翻倍：CW = min(2×CW, CWmax)。

举个例子。假设CWmin=15，CWmax=1023：

- 第一次尝试：CW=15，退避时间在0-15之间随机
- 第一次碰撞：CW变成31，退避时间在0-31之间随机
- 第二次碰撞：CW变成63
- 第三次碰撞：CW变成127
- ...以此类推，最大到1023

4.3 为什么指数增长?

碰撞说明什么？说明竞争激烈，有很多节点在争信道。这时候如果退避窗口还很小，大家还是很容易在差不多的时间发送，继续碰撞。

把退避窗口翻倍，就是把发送时间分散到更大的范围里，减少再次碰撞的概率。竞争越激烈，退避窗口越大，这是很合理的。

【本课时小结】（5分钟）

今天我们详细学习了CSMA/CA：

1. **CSMA/CA核心思想**：避免碰撞，而不是检测碰撞
2. **帧间间隔**：SIFS < DIFS，用于优先级控制
3. **RTS/CTS机制**：发送前先"预约"信道 ★
4. **解决隐蔽站**：C收到B的CTS，知道要保持安静 ★
5. **NAV**：虚拟载波侦听，不需要持续监听信道
6. **二进制指数退避**：碰撞后窗口翻倍

CSMA/CA已经很好了，WiFi就用的它。但对WSN来说，还不够省电。为什么？因为节点需要经常侦听信道，即使没有数据要收发，也在浪费能量。

下节课我们学WSN专用的MAC协议，看看如何进一步省电。

第三节课（100分钟）

第5课时：2.3.5 WSN专用MAC协议（50分钟）

【课前回顾与引入】（5分钟）

同学们好！前两节课我们学习了从CSMA/CD到CSMA/CA的演进，解决了无线网络的信道访问问题。

但是我问大家：CSMA/CA足够好了吗？对于WiFi来说够了，但对于WSN呢？

WSN的节点靠电池供电，可能要工作几年不换电池。对它来说，**能量就是生命**。而CSMA/CA有一个很大的问题——**空闲侦听**。

什么是空闲侦听？就是信道上没有数据的时候，节点也要持续侦听信道，看看有没有人发数据给自己。这就像你一直竖着耳朵听电话铃响——即使没人打电话，你也在消耗精力。

研究表明，**空闲侦听消耗的能量几乎和发送数据一样多**！这对WSN来说是不可接受的。

那怎么办？核心思想其实很简单：**不需要的时候就睡觉。**

【第一部分：WSN MAC面临的能量挑战】（5分钟）

传统MAC协议有四大能量浪费来源：

第一，空闲侦听（Idle Listening）。 没有数据也要持续监听，这是最大的能量浪费。研究表明可占总能耗的50%以上。

第二，碰撞（Collision）。 碰撞后两个帧都白发了，能量白白浪费，还要重传。

第三，串扰（Overhearing）。 收到了不是给自己的帧，但还是接收和处理了，浪费能量。

第四，控制开销（Control Overhead）。 RTS、CTS、ACK等控制帧也消耗能量。

WSN MAC协议的设计目标就是**尽可能减少这四种能量浪费**，尤其是空闲侦听。

【第二部分：S-MAC协议】（15分钟）★★

2.1 设计思想

S-MAC是2002年由Wei Ye等人提出的，专门为传感器网络设计的MAC协议。S代表Sensor。

S-MAC的核心思想非常朴素：**既然空闲侦听浪费能量，那就让节点周期性地睡眠。**

2.2 周期性睡眠机制

S-MAC把时间分成等长的周期，每个周期分为**活跃期**和**睡眠期**。

在活跃期，节点开启收发器，可以收发数据。在睡眠期，节点关闭收发器，进入低功耗休眠状态。

假设占空比是10%：

- 周期是1秒
- 活跃期100毫秒
- 睡眠期900毫秒
- **节省了90%的空闲侦听能量！**

占空比可以在5%到20%之间调整。占空比越低越省电，但延迟越大。

2.3 同步问题

但这里有个关键问题：如果A想发数据给B，A在活跃期，B在睡眠期，怎么办？

所以，**邻居节点必须同步**——大家同时醒来，同时睡觉。

S-MAC用**SYNC帧**来同步：

- 每个周期的活跃期开始时，节点广播SYNC帧
- SYNC帧包含"我什么时候醒来"的信息
- 邻居节点收到后，调整自己的调度，和它一致

这样，邻居节点就有了统一的睡眠-醒来调度。

```
节点A:  |活跃|————睡眠————|活跃|————睡眠————|
节点B:  |活跃|————睡眠————|活跃|————睡眠————|
            ↑ 同步醒来              ↑ 同步醒来
```

2.4 消息传递

如果有一条长消息要发，一个活跃期传不完怎么办？

S-MAC使用**突发传输**模式：将长消息分成多个片段，在一个活跃期内连续发送所有片段。接收节点收到第一个RTS后，保持清醒直到所有片段传完。

2.5 S-MAC的优缺点

优点：

- 大幅减少空闲侦听，能量效率高
- 实现相对简单
- 节点间自组织同步

缺点：

- **延迟增加**：如果数据到达时节点在睡眠，要等到下一个活跃期。多跳传输延迟更大。
- **固定占空比**：不管流量多少，占空比都一样。流量低时不够省电，流量高时活跃期不够。
- **边界节点问题**：处于两个不同调度组边界的节点，需要维护两套调度。

【第三部分：T-MAC协议】（12分钟）★★

3.1 设计动机——S-MAC的问题

S-MAC的固定占空比有什么问题？让我举个例子。

假设占空比是10%，活跃期100ms。如果某个时刻只有1个短帧要发，10ms就发完了，剩下90ms节点还是醒着的——白白侦听了90ms。

反过来，如果突然来了大量数据，100ms活跃期不够用，剩余数据要等到下一个周期才能发，延迟增大。

T-MAC就是为了解决这个问题。T代表Timeout（超时）。

3.2 自适应活跃期

T-MAC的核心改进：**活跃期长度不固定，而是根据通信量自适应调整**。

怎么调整？用超时机制：

```
T-MAC超时机制：

定义超时时间TA

活跃期开始 → 等待TA时间
```

```
    ├── 有事件（收到RTS/CTS/DATA）→ 重置TA计时器，继续活跃
    └── TA时间内无任何事件 → 超时！→ 提前进入睡眠

对比：
S-MAC: |──────── 固定活跃期（100ms）────────|──睡眠──|
       |有用的通信|    空闲浪费              |

T-MAC: |── 有用的通信 ──| TA超时 |──── 更长的睡眠 ────|
                                  ↑
                            提前进入睡眠，更省电！
```

TA要设多大？至少要能完成一次RTS-CTS交换，否则正要开始通信就睡着了。

3.3 T-MAC的早睡问题

T-MAC有一个问题：**早睡问题**。

场景：A给B发数据，B要转发给C。当B在处理A的数据时，C在等B的数据。但C等了TA时间没等到（因为B还在忙），C就超时睡着了！等B准备好转发时，C已经睡了。

解决方法：

- **FRTS（Future RTS）**：B在处理A的数据之前，先给C发一个"提前通知"，告诉C"别睡，等一下我有数据给你"。
- **DS位**：在帧中加一个标志位，表示"还有更多数据要发"。

3.4 T-MAC vs S-MAC

特性	S-MAC	T-MAC
活跃期	固定	自适应
省电效果	好	更好（特别是低流量时）
复杂度	较低	较高
早睡问题	无	有，需额外机制
适用场景	流量较稳定	流量变化大

【第四部分：Z-MAC协议】（8分钟）

4.1 设计思想

前面我们学了基于竞争的S-MAC、T-MAC，也学了基于调度的TDMA。各有优缺点：

- CSMA类（竞争型）：低负载时效率高，高负载时碰撞多
- TDMA（调度型）：高负载时无碰撞，低负载时时隙浪费

有没有两全其美的办法？Z-MAC说：**两个都要！**

Z-MAC = Zebra MAC，是TDMA和CSMA的混合协议。根据网络负载自动切换模式。

4.2 双模式工作机制

Z-MAC定义了两个竞争级别：

低竞争级别（LCL）——网络负载低：

- 像CSMA一样工作
- 任何节点可以在任何时隙尝试发送
- 但时隙"拥有者"有优先权

高竞争级别（HCL）——网络负载高：

- 像TDMA一样工作
- 只有时隙拥有者才能在该时隙发送
- 大大减少碰撞

```
Z-MAC自适应切换：

低负载 → LCL模式（像CSMA）→ 灵活
      ↕ 自动切换
高负载 → HCL模式（像TDMA）→ 无冲突
```

4.3 Z-MAC特点

优点：自适应能力强，兼具CSMA的灵活性和TDMA的无碰撞。

缺点：实现复杂，需要时隙分配算法，模式切换开销较大。初始化阶段需要较长时间来建立时隙分配。

【第五部分：其他WSN MAC协议简介】（3分钟）

简单提两个：

SIFT协议：专门为事件驱动型WSN设计。多个节点同时检测到一个事件时，只需要一个节点汇报。SIFT用概率机制控制：竞争窗口固定，发送概率根据竞争节点数调整，让其中一个节点迅速抢到信道。

TRAMA协议：流量自适应的TDMA。根据实际流量动态分配时隙，空闲时隙可以让给有数据的节点。比纯TDMA更灵活。

【本课时小结】（2分钟）

1. **WSN MAC四大能量浪费**：空闲侦听、碰撞、串扰、控制开销
2. **S-MAC**：周期性睡眠，占空比5%-20%，通过SYNC同步 ★
3. **T-MAC**：自适应活跃期，超时提前睡眠，更省电 ★
4. **Z-MAC**：TDMA+CSMA混合，根据负载自动切换

第6课时：2.3.6 总结与协议对比（50分钟）

【第一部分：MAC协议演进回顾】（10分钟）

同学们，这三节课我们走过了MAC协议从诞生到WSN专用的整个发展历程。让我们做一个完整的回顾。

1.1 从ALOHA到Z-MAC

```
ALOHA（1968）
  │ 想发就发，利用率只有18%
  ↓
CSMA（1970s）
  │ 先听后发，但不能检测碰撞
  ↓
CSMA/CD（1980s）
  │ 边听边发，用于以太网
  │ 问题：不适合无线（隐蔽站、碰撞检测难）
  ↓
CSMA/CA（1990s）
  │ 避免碰撞，RTS/CTS，用于WiFi
  │ 问题：空闲侦听浪费能量
  ↓
S-MAC（2002）
  │ 周期性睡眠，减少空闲侦听
  │ 问题：固定占空比不灵活
  ↓
T-MAC（2003）
  │ 自适应活跃期，超时睡眠
  ↓
Z-MAC（2005）
  │ TDMA+CSMA混合，自适应切换
```

每一步改进都是为了解决前一代的问题。这就是技术发展的逻辑。

1.2 分类总结

```
MAC协议全景图：

├── 传统MAC
│    ├── ALOHA – 最简单，效率最低
│    ├── CSMA/CD – 有线网络标准
│    └── CSMA/CA – 无线网络标准
│
├── 静态MAC
│    ├── TDMA – 时分，WSN常用
│    ├── FDMA – 频分，不常用
│    └── CDMA – 码分，复杂度高
│
└── WSN专用MAC
     ├── 基于竞争：S-MAC、T-MAC、SIFT
     ├── 基于调度：TRAMA、SMACS/EAR
     └── 混合型：Z-MAC
```

【第二部分：协议性能详细对比】（10分钟）

2.1 综合对比表

协议	类型	能效	延迟	吞吐量	复杂度	最适合场景
CSMA/CA	竞争	低	低	高	中	WiFi
TDMA	调度	高	固定	固定	低	规则采集
S-MAC	竞争+睡眠	高	高	低	低	低流量监测
T-MAC	竞争+自适应	高	中	中	中	变化流量
Z-MAC	混合	中高	低	高	高	高流量

2.2 MAC协议选择指南

场景一：环境监测（每分钟采集一次温湿度）

- 流量低且稳定 → **S-MAC**
- 原因：简单、省电、低流量延迟可接受

场景二：工厂设备监控（流量时高时低）

- 流量变化大 → **T-MAC**
- 原因：自适应活跃期，闲时省电，忙时延长

场景三：地震监测（平时无数据，事件突发大量数据）

- 突发高流量 → **Z-MAC** 或 **SIFT**
- 原因：Z-MAC高负载时切换TDMA模式；SIFT快速竞争

场景四：精准农业（定时采集土壤数据）

- 规则、可预测的流量 → **TDMA**
- 原因：无碰撞、延迟可控、可休眠

【第三部分：核心知识回顾】（10分钟）

让我们用问答方式回顾最核心的知识点：

问题1：数据链路层的核心功能是什么？ 答：封装成帧、透明传输、差错控制、流量控制、链路管理。

问题2：为什么CSMA/CD不适合无线网络？ 答：三个原因——碰撞检测困难（自身信号太强）、隐蔽站问题、暴露站问题。

问题3：CSMA/CA如何解决隐蔽站？ 答：通过RTS/CTS机制。隐蔽站虽然收不到RTS，但能收到接收方的CTS，从而知道信道被占用。

问题4：S-MAC如何省电？ 答：周期性睡眠。将时间分为活跃期和睡眠期，睡眠期关闭收发器。通过SYNC帧实现邻居同步。

问题5：T-MAC比S-MAC好在哪里？ 答：自适应活跃期。通过超时机制，没有通信活动时提前进入睡眠，避免S-MAC固定活跃期的浪费。

问题6：Z-MAC的设计思想？ 答：结合TDMA和CSMA。低负载时像CSMA灵活使用信道，高负载时像TDMA按时隙发送避免碰撞。

【第四部分：知识框架总结】（5分钟）

```
2.3 MAC层技术介绍
│
├── 2.3.1 数据链路层概述
│   ├── 五大功能
│   ├── 帧结构与MAC地址
│   └── IEEE 802参考模型（LLC + MAC）
│
├── 2.3.2 MAC协议分类与静态协议
│   ├── ALOHA — 开创随机接入
│   ├── 分类：静态/动态、竞争/调度
│   └── TDMA / FDMA / CDMA
│
├── 2.3.3 CSMA/CD
│   ├── 三大特征：先听后发、边听边发、碰撞停止
│   ├── 隐蔽站问题 ★
│   └── 暴露站问题 ★
│
├── 2.3.4 CSMA/CA
│   ├── 帧间间隔（SIFS/DIFS）
│   ├── RTS/CTS机制 ★
│   ├── NAV虚拟载波侦听
│   └── 二进制指数退避算法
│
├── 2.3.5 WSN专用MAC
│   ├── 四大能量浪费
│   ├── S-MAC：周期性睡眠 ★
│   ├── T-MAC：自适应活跃期 ★
│   └── Z-MAC：TDMA+CSMA混合
│
└── 2.3.6 总结与对比
    └── 协议选择指南
```

【第五部分：随堂练习】（10分钟）

（见单独的练习文档）

【衔接下一章】（5分钟）

同学们，至此我们学完了数据链路层。

让我们回顾一下协议栈的学习进度：

- 2.2节物理层：解决"如何传输比特"
- 2.3节数据链路层：解决"如何共享信道、可靠传输帧"
- 下一节网络层：解决"如何将数据从源节点路由到目的节点"

物理层是道路，数据链路层是交通规则，网络层就是导航系统——告诉数据该走哪条路。

下一章我们将学习WSN的路由协议，这也是WSN的核心技术之一。请大家预习最短路径算法和基本图论知识。

谢谢大家！有问题课后来找我。