



# 中华人民共和国国家标准

GB/T 30269.302—2015

## 信息技术 传感器网络 第 302 部分：通信与信息交换： 高可靠性无线传感器网络 媒体访问控制和物理层规范

Information technology—Sensor network—  
Part 302: Communication and information exchange:  
Medium access control(MAC) and physical  
layer(PHY) specification for reliable wireless sensor networks

2015-12-31 发布

2017-01-01 实施



中华人民共和国国家质量监督检验检疫总局  
中国国家标准化管理委员会

发布

## 目 次

前言 .....	I
1 范围 .....	1
2 规范性引用文件 .....	1
3 术语和定义 .....	1
4 缩略语 .....	4
5 总体描述 .....	6
5.1 网络拓扑 .....	6
5.2 协议栈 .....	7
5.3 功能综述 .....	7
5.4 原语的概念 .....	13
6 物理层规范 .....	14
6.1 一般要求 .....	14
6.2 操作频率范围 .....	14
6.3 PHY 服务规范 .....	16
6.4 PPDU 格式 .....	25
6.5 PHY 常数和 RIB 属性 .....	26
6.6 技术要求 .....	28
7 MAC 层规范 .....	33
7.1 MAC 层服务原语 .....	33
7.2 MAC 帧格式 .....	78
7.3 MAC 命令帧 .....	88
7.4 MAC 常量和 RIB 属性 .....	96
7.5 MAC 功能概述 .....	103
7.6 MAC 层和 PHY 交互的消息序列图 .....	138
参考文献 .....	147

## 前 言

GB/T 30269《信息技术 传感器网络》分为以下几部分：

- 第 1 部分：参考体系结构和通用技术要求；
- 第 2 部分：术语；
- 第 301 部分：通信与信息交换：低速无线传感器网络网络层和应用支持子层规范；
- 第 302 部分：通信与信息交换：高可靠性无线传感器网络媒体访问控制和物理层规范；
- 第 303 部分：通信与信息交换：基于 IP 的网络层规范；
- 第 304 部分：通信与信息交换：面向视频的媒体访问控制和物理层规范；
- 第 401 部分：协同信息处理：支撑服务及接口；
- 第 501 部分：标识：传感结点标识符编制规则；
- 第 502 部分：标识：传感结点标识符解析和管理规范；
- 第 503 部分：标识：传感结点标识符注册规程；
- 第 601 部分：信息安全：通用技术规范；
- 第 602 部分：信息安全：网络传输安全技术规范；
- 第 701 部分：传感器接口：信号接口；
- 第 702 部分：传感器接口：数据接口；
- 第 801 部分：测试：通用要求；
- 第 802 部分：测试：低速无线传感器网络媒体访问控制和物理层；
- 第 803 部分：测试：低速无线传感器网络网络层和应用支持子层；
- 第 901 部分：网关：通用技术要求；
- 第 1001 部分：中间件：传感器网络结点数据交换规范。

本部分是 GB/T 30269 的第 302 部分。

本部分按照 GB/T 1.1—2009 给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本部分由全国信息技术标准化技术委员会(SAC/TC 28)提出并归口。

本部分起草单位：深圳市海思半导体有限公司、中国电子技术标准化研究院、重庆邮电大学、北京邮电大学、中国电力科学研究院。

本部分主要起草人：刘培、卓兰、王浩、王平、邹卫霞、全杰。

广东省网络空间安全协会受控资料

# 信息技术 传感器网络

## 第 302 部分:通信与信息交换:

### 高可靠性无线传感器网络

### 媒体访问控制和物理层规范

#### 1 范围

GB/T 30269 的本部分规定了 470 MHz~510 MHz 频段高可靠性无线传感器网络的物理层和媒体访问控制层技术规范。

本部分适用于短距离智能无线抄表、智能家居控制等应用。

#### 2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件,仅注日期的版本适用于本文件。凡是不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

GB/T 15629.2 信息技术 系统间远程通信和信息交换 局域网和城域网 特定要求 第 2 部分:逻辑链路控制

GB/T 30269.2 信息技术 传感器网络 第 2 部分:术语  
微功率(短距离)无线电设备的技术要求(信部无[2005]423 号文件)

#### 3 术语和定义

GB/T 30269.2 中界定的以及下列术语和定义适用于本文件。

##### 3.1

**关联 association**

在 RWSN 中建立成员之间关系的一种服务。

##### 3.2

**自动唤醒 automatically wake-up**

MAC 层提供的一种服务,协调器发送唤醒帧序列,终端设备周期侦听信道时若接收到唤醒帧,即结束休眠状态。

##### 3.3

**信标使能的 RWSN beacon-enabled RWSN**

在 RWSN 内所有协调器都可以发送规则信标,即信标指数小于 0X0F。

##### 3.4

**竞争访问周期 contention access period; CAP**

直接跟在信标帧之后的时间周期,在该时间周期期望发送的设备利用 CSMA-CA 机制竞争信道接入。

[GB/T 15629.15—2010,定义 3.7]

3.5

**CAP 符号 CAP symbol**

在 CAP 期间出现的周期符号。

[GB/T 15629.15—2010, 定义 3.8]

3.6

**候选信道 candidate channel**

非活跃期所扫描的两个待扫描信道中的一个。

3.7

**信道队列 channel queue**

协调器指定的多个待扫描信道集合, 包含多个信道。

3.8

**信道维护周期 channel maintenance period**

进行两次连续的信道维护过程的时间间隔。

3.9

**信道维护 channel maintenance**

约定信道和候选信道经 ED 检测后, 协调器比较约定信道和候选信道的能量等级或平均能量等级大小, 并以比较结果为依据更新约定信道和候选信道的过程。

3.10

**设备 device**

可以执行媒体访问控制与无线媒体的物理接口的任何实体。它可以是一个简化功能设备 (RFD) 或全功能设备 (FFD)。

3.11

**帧 frame**

按时一起发送的所聚集的来自媒体访问控制 (MAC) 子层的若干比特的格式。

[GB/T 15629.15—2010, 定义 3.14]

3.12

**全功能设备 full-function device**

能作为协调器进行操作的设备。

[GB/T 15629.15—2010, 定义 3.15]

3.13

**本地时钟 local clock**

设备内部的符号时钟。

[GB/T 15629.15—2010, 定义 3.23]

3.14

**监测数据 monitoring data**

一种业务数据, 需要网络提供高可靠性传输的支持。

3.15

**非监测数据 non-monitoring data**

一种非业务数据, 不需要网络提供高可靠性传输的支持。

3.16

**包 packet**

通过物理媒体按时一起发送的格式化聚集的若干比特。

[GB/T 15629.15—2010, 定义 3.31]

## 3.17

**净荷数据 payload data**

正在发送的数据消息的内容。

## 3.18

**RWSN 协调器 RWSN coordinator**

能转发消息的全功能设备。RWSN 中起主要控制作用的协调器,一个 RWSN 中有一个确定的 RWSN 协调器。

## 3.19

**约定信道 prescribed channel**

用于广播信标帧和重新传输节点在工作信道上发送失败的数据的信道。

## 3.20

**射频唤醒 radio frequency wake-up**

PHY 和 MAC 层共同提供的一种服务,协调器先后发送射频脉冲序列和唤醒帧,终端设备感应射频脉冲而被唤醒,并接收和解析唤醒帧,进而结束休眠状态。

## 3.21

**简化功能设备 reduced-function device**

不能作为协调器使用的设备。

## 3.22

**子非竞争访问周期 son contention free period;SCFP**

把非竞争访问期 CFP 划分成三段,每一段为一个 SCFP。

## 3.23

**备用信道 spare channel**

用于重新传输节点在约定信道上发送失败的数据的信道。

## 3.24

**事务 transaction**

为了成功地传输 MAC 命令或数据帧而要求的在两个对等的媒体访问控制(MAC)实体之间连续进行的帧交换。

## 3.25

**事务队列 transaction queue**

由给定协调器的 MAC 层所启动的待处理事务的列表,该列表使用迂回的传输来发送,当事务在进行中时,该事务队列通过改协调器来维护,并且其长度与现实相关,但应至少为 1。

[GB/T 15629.15—2010,定义 3.44]

## 3.26

**中间退避期 middle backoff period**

定位中间退避边界的时间窗。

## 3.27

**信道使用列表 using channel list**

记录所述超帧内各信道上最近一次成功发送数据的节点信息。

## 3.28

**工作信道 working channel**

在每轮工作周期中节点之间点到点传输数据的信道。

3.29

**工作信标 working beacon**

设备在每轮工作周期开始时打开接收机所接收到的第一个信标,在本轮工作周期内,设备只接收并跟踪这一个信标。

3.30

**工作周期 working period**

在信标使能的 RWSN 中,设备相邻两次以侦听信标帧为目的而打开接收机的间隔时间,在一个工作周期内,设备仅侦听工作周期开始时的第一个信标帧,其余时间不侦听信标帧;设备默认的工作周期时长为单个超帧的时长。

3.31

**工作超帧 working superframe**

设备在每轮工作周期中接收到的工作信标所对应的超帧,在本轮工作周期内,设备只在该超帧内安排自己的任务。

3.32

**工作设备 working device**

以特定工作超帧作为自己工作超帧的设备。

4 缩略语

下列缩略语适用于本文件。

AW	自动唤醒 (automatic wake-up)
BE	退避指数 (backoff exponent)
BI	信标间隔 (beacon interval)
BLE	电池寿命扩充 (battery life extension)
BO	信标指数 (beacon order)
BSN	信标序列号 (beacon sequence number)
CAP	竞争访问周期 (contention access period)
CCA	空闲信道评估 (clear channel assessment)
CMA	信道调制评估 (channel modulation assessment)
CFP	非竞争周期 (contention-free period)
CRC	循环冗余校验 (cyclic redundancy check)
CSMA/CA	碰撞避免的载波侦听多址访问 (carrier sense multiple access with collision avoidance)
CTR	计数器模式 (counter mode)
CW	竞争窗口 (长度) [contention window (length)]
DSN	数据序列号 (data sequence number)
ED	能量检测 (energy detection)
EIRP	有效全向辐射功率 (effective isotropic radiated power)
ERP	有效辐射功率 (effective radiated power)
EVM	差错矢量幅度 (error-vector magnitude)
FCS	帧检验序列 (frame check sequence)
FFD	全功能设备 (full-function device)
GTS	保证的时隙 (guaranteed time slot)



HWSL	混合唤醒采样侦听(hybrid wake-up sampling listen)
IFS	帧间间隔(inter frame space or spacing)
LE	低消耗(low expend)
LIFS	长帧间间隔(long inter-frame spacing)
LLC	逻辑链路控制(logical link control)
LQI	链路质量指示(link quality indication)
LPDU	LCC 协议数据单元(LLC protocol data unit)
LSB	最低有效位(least significant bit)
MAC	媒体访问控制(media access control)
MB	中间退避(middle backoff)
MCPS	MAC 公共部分子层(MAC common part sublayer)
MCPS-SAP	MAC 公共部分子层-服务访问点(MAC common part sublayer-service access point)
MFR	MAC 帧尾(MAC footer)
MHR	MAC 帧头(MAC header)
MLME	MAC 层管理实体(MAC sublayer management entity)
MLME-SAP	MAC 层管理实体-服务访问点(MAC sublayer management entity-service access point)
MP	单元中间期(unit middle period)
MSB	最高有效位(most significant bit)
MSL	超帧时长的倍数(multiples of superframe length)
MPDU	MAC 协议数据单元(MAC protocol data unit)
MSDU	MAC 服务数据单元(MAC service data unit)
NB	退避(周期)数(number of backoff (periods))
NWBSN	下一个工作信标帧的序列号(next working BSN)
O-QPSK	偏置正交相移键控(offset quadrature phase-shift keying)
OSI	开放系统互连(open systems interconnection)
PD	PHY 数据(PHY data)
PD-SAP	PHY 数据服务访问点(PHY data service access point)
PDU	协议数据单元(protocol data unit)
PER	误包率(packet error ratio)
PHR	PHY 头(PHY header)
PHY	物理层(physical layer)
PLME	物理层管理实体(physical layer management entity)
PLME-SAP	物理层管理实体-服务访问点(physical layer management entity-service access point)
POS	个人工作空间(personal operating space)
PPDU	物理层协议数据单元(PHY protocol data unit)
PSD	功率谱密度(power spectral density)
PSDU	PHY 服务数据单元(PHY service data unit)
ppm	每百万分之一(parts per million)
RF	射频(radio frequency)
RFD	简化功能设备(reduced-function device)

RFW	射频唤醒机制(Radio frequency wake-up)
RIB	RWSN 信息库(RWSN information base)
RWSN	高可靠性无线传感器网络(reliable wireless sensor network)
RX	接收或接收器(receive or receiver)
SAP	服务访问点(service access point)
SCFP	子非竞争访问周期(son contention free period)
SD	超帧持续期(superframe duration)
SDU	服务数据单元(service data unit)
SFD	帧起始定界符(start-of-frame delimiter)
SHR	同步头(synchronization header)
SIFS	短帧间间隔(short inter-frame spacing)
SLW	同步侦听唤醒(synchronization listen wake-up)
SO	超帧指数(superframe order)
SPDU	SSCS 协议数据单元(SSCS protocol data units SCS)
SSCS	服务特定汇聚子层(service specific convergence sublayer)
TRX	收发器(transceiver)
TX	发送或发送器(transmit or transmitter)
WPL	工作周期时长(working period length)
$X_B$	X 个退避周期(X backoff period)

5 总体描述

5.1 网络拓扑

RWSN 是一个低速率、低功耗、易安装、数据传输可靠的短距离无线传感器网络。RWSN 拓扑结构为星型网络,如图 1 所示。在星型网络中,所有终端设备都需要与网络协调器通信。本部分中网络协调器一般使用持续电力系统供电。

星型网络以网络协调器为中心,所有设备只能与网络协调器进行通信。在星型网络形成过程中,一旦网络中有一个 FFD 被启动后,便由它来建立网络并成为网络协调器。选择一个标识符后,网络协调器就允许其他设备加入自己的网络,并为这些设备转发数据帧。

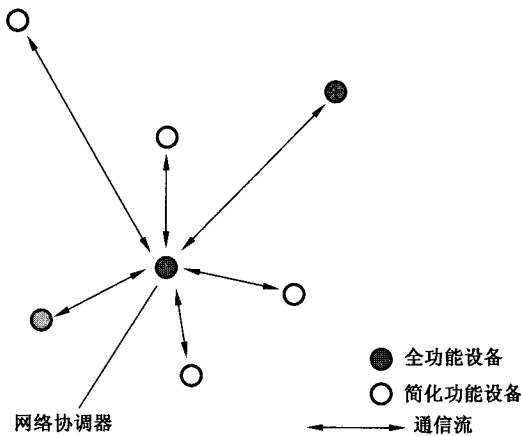


图 1 星型网络结构

## 5.2 协议栈

### 5.2.1 概述

RWSN 协议栈是基于 OSI 模型,如图 2 所示,每一层都实现一部分通信功能,并向高层提供服务。本部分只规定 MAC 和 PHY 的内容。

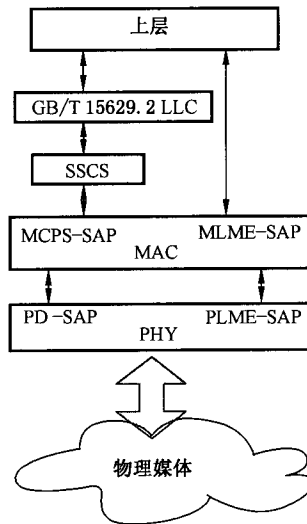


图 2 RWSN 协议栈结构

### 5.2.2 物理层

物理层定义了两种服务:连接到 PLME-SAP 的 PHY 数据服务和 PHY 管理服务。PHY 数据服务在物理发射信道上收发 PPDU<sub>s</sub>。PHY 管理服务维护一个由物理相关数据组成的数据库。

PHY 数据服务包括以下功能:

- a) 启动和休眠射频收发机;
- b) 信道能量检测;
- c) 检测接收数据帧的链路质量并指示;
- d) 空闲信道评估;
- e) 收发数据。

### 5.2.3 MAC 层

MAC 层提供两种服务:连接到 MLME-SAP 的 MAC 层数据服务和 MAC 层管理服务,前者保证 MAC 协议数据单元在 PHY 数据服务中的正确收发,后者维护一个存储 MAC 层协议状态相关信息的数据库。

MAC 层主要功能包括:信标管理、信道接入、SCFP 管理、设备工作周期管理、帧确认、发送确认帧、关联和解关联。

## 5.3 功能综述

### 5.3.1 超帧结构

超帧的活跃期划分为三个阶段:信标帧发送期、CAP 和 CFP。超帧的活跃期划分为若干个等长的时隙,每个时隙的长度、竞争访问周期包含的时隙数等参数,均由协调器设定,并通过超帧开始时发出的

信标帧广播到整个网络。

在 RWSN 中,每个超帧都以网络协调器发出信标帧为开始(见图 3),并被平分为若干个时隙。根据需要,也可将超帧分为活跃期和非活跃期两个部分(见图 3)。在非活跃期,RWSN 中的设备,如果是能量受限的将不进行互相通信,从而可以进入休眠状态以节省能量,如果是能量非受限的,除了已经成功申请在非活跃期进行通信的设备,也将进入休眠状态。在每个超帧的第一个时隙发送信标帧。如果不需要使用超帧结构,协调器将终止信标的传输。信标帧通常用来与已连接设备之间保持同步、标识 RWSN,以及描述超帧结构。如果设备在两个信标之间的 CAP 内进行通信,那么需要采用时隙的 CSMA-CA 算法与其他设备竞争信道。所有通信事务将在下一个网络信标到达前结束。这个信标帧包含了超帧将持续的时间以及对这段时间的分配等信息。除协调器外,网络中的各个设备在其每轮工作周期开始时接收一个信标作为其工作信标,并在工作信标所对应的工作超帧里,根据工作信标帧中的信息安排自己的任务,且设备只在它的工作超帧中才安排自己的任务,如访问信道。在下一个工作周期开始之前,设备不再接收后续信标帧,也不在后续的超帧中安排自己的任务。

图 3 所示为超帧结构示意图。

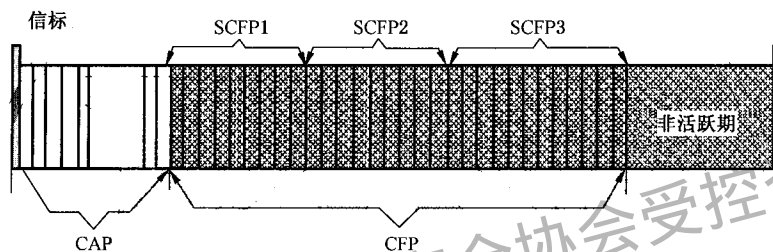


图 3 超帧结构示意图

在超帧的 CAP, RWSN 设备使用带时隙的 CSMA-CA 访问机制,并且任何通信都应在竞争访问周期结束前完成。在非竞争期,协调器根据本超帧的工作设备在其各自的上一个工作超帧中申请数据发送的情况,将非竞争期划分成 3 个 SCFP,即 SCFP1、SCFP2 和 SCFP3。其中 SCFP1 由若干个时隙组成,时隙数目等于申请发送数据的设备数目,SCFP1 用于节点在初始分配的时隙内发送数据,SCFP2 用于节点重新发送在 SCFP1 发送失败的数据,SCFP3 用于继续传输节点在该第一 SCFP 或该第二 SCFP 未完全发送的数据,包括重新发送在 SCFP2 发送失败的数据。其中 SCFP1 和 SCFP2 的时隙个数相同。在非竞争期,每个时隙都指定分配给了在本超帧申请了时隙的工作设备,因此不需要竞争信道。RWSN 要求工作设备应在自己分配的时隙内完成通信。竞争访问周期的功能包括本超帧的各个工作设备可以自由收发数据帧,向协调器申请各自的下一个工作超帧内的 SCFP 时隙,新设备加入当前的 RWSN。

### 5.3.2 数据传输模型

RWSN 中存在着两种数据传输事务:设备发送数据给协调器和协调器发送数据给设备。其中,设备发送给协调器的数据分为非监测数据和监测数据两种。在 RWSN 中,通信方式采用信标能力通信。

在信标使能的网络中,网络协调器定时广播信标帧。设备之间通信使用基于时隙的 CSMA-CA 信道访问机制,RWSN 中设备都通过协调器发送的信标帧进行同步。在时隙 CSMA-CA 机制下,每当设备需要发送数据帧或命令帧时,它首先定位下一个时隙的边界,设备就在下一个可用时隙边界开始发送数据;如果信道忙,设备需要重新等待随机个数的时隙,再检查信道状态,重复这个过程直到有空闲信道出现。在这种机制下,确认帧的发送不需要使用 CSMA-CA 机制,而是紧跟接收帧发送回源设备。

图 4 中是一个有信标能力网络中某一设备传送非监测型数据给协调器的例子。该设备首先侦听网络中的信标帧,如果接收到信标帧,它就同步到以此信标帧开始的超帧上,然后应用时隙 CSMA-CA 机

制,选择一个合适的时机,把数据帧发送给协调器。协调器成功接收到数据以后,回发一个确认帧表示成功收到该数据帧。

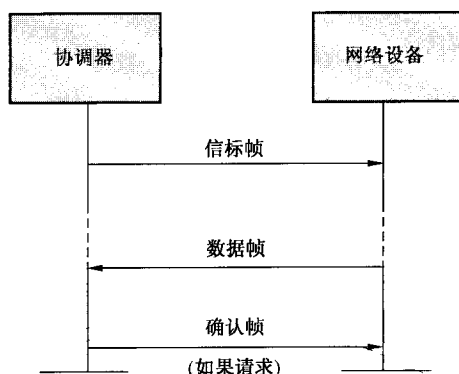


图 4 在有信标能力网络中网络设备发送非监测型数据给协调器

图 5 为一个有信标能力网络中某一设备传送监测型数据给协调器的例子。该设备首先侦听网络中的信标帧,如果接收到信标帧,它就同步到以此信标帧开始的超帧上,然后应用时隙 CSMA-CA 机制,选择一个合适的时机,把数据帧发送给协调器。协调器成功接收到数据以后,回发一个普通确认帧表示成功收到该数据帧。之后,协调器启动数据预测机制,根据相应的预测算法得到预测的数据置信区间。若接收到的监测型数据位于置信区间之内,则协调器确认接受该数据并向设备回发数据接受确认帧;若接收到的监测型数据位于置信区间之外,则协调器向设备发送包含原监测数据的质疑数据帧。

收到质疑数据帧的设备需再次核实原监测数据的正确性:若原数据正确,则设备向协调器发送数据质疑无效确认帧,之后协调器确认接受该数据,并向设备发送数据接受确认帧;若原数据有误,则设备向协调器发送数据质疑有效确认帧,并发送新的监测型数据。重复上述过程,直到监测型数据被协调器接受为止。

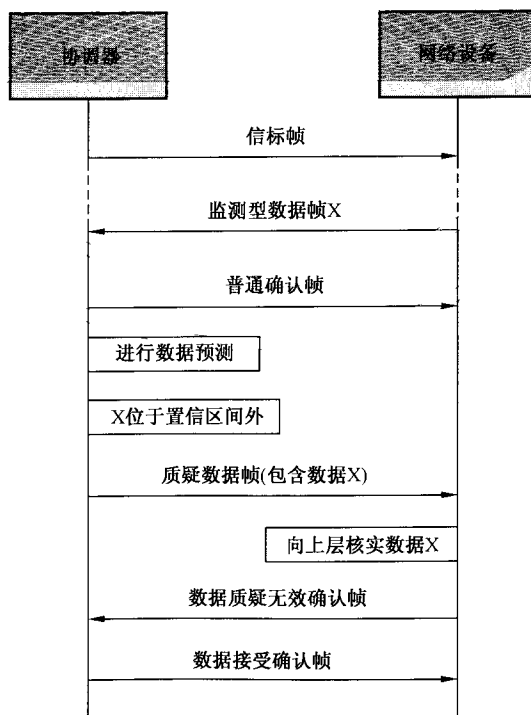


图 5 在有信标能力网络中网络设备发送监测型数据给协调器

图 6 为在有信标能力网络中协调器发送数据帧给网络中某个设备的例子。当协调器需要向某个设备发送数据时,就在该设备的下一个工作信标帧中说明协调器拥有属于该设备的数据正在等待发送。目标设备在周期性的侦听过程中会接收到这个信标帧,从而得知属于自己的数据保存在协调器,这时就会向协调器发送请求传送数据的 MAC 命令。何时发送该命令帧由时隙 CSMA-CA 机制来确定。协调器收到请求帧后,先响应一个确认帧表明收到请求命令,然后开始传送数据。设备成功接收到数据帧后再回发一个数据确认帧,协调器接收到这个确认帧后,才将消息从自己的消息队列中移走。

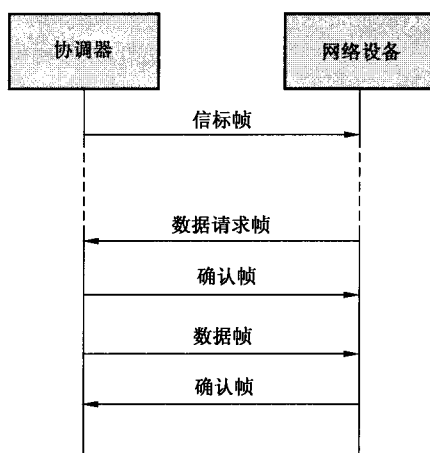


图 6 在有信标能力网络中协调器传送数据给网络设备

### 5.3.3 MAC 层帧类型

#### 5.3.3.1 信标帧

信标帧在 MAC 层产生,协调器可在有信标能力网络中发送网络信标帧。信标帧的净荷数据单元由四部分组成:超帧描述字段、SCFP 字段、周期分配字段、待处理数据目的地址字段和信标帧净荷数据,如图 7 所示。

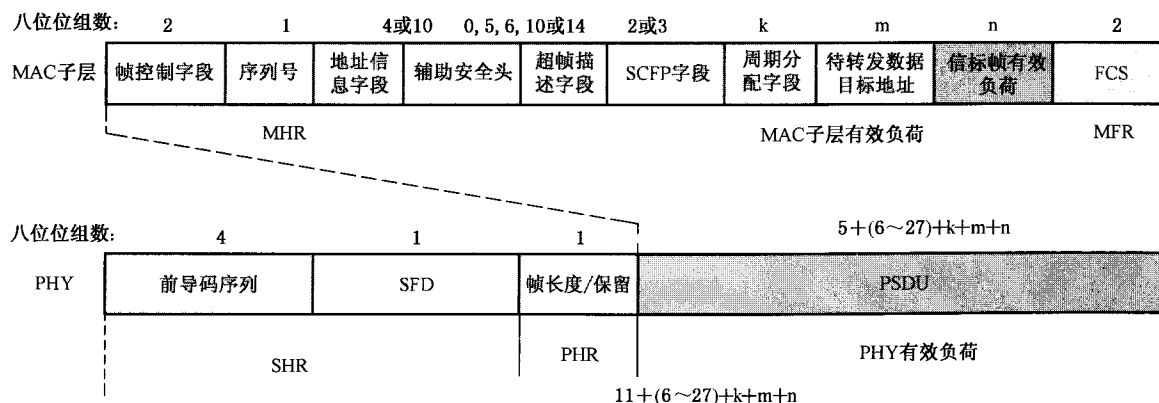


图 7 信标帧的格式

信标帧中超帧规范字段规定了该超帧的持续时间、活跃期持续时间以及竞争访问周期持续时间等信息。

SCFP 字段将非竞争期划分为三个 SCFP, 即: SCFP1、SCFP2 和 SCFP3, 其中 SCFP1 用于节点在协调器初始分配的时隙发送数据; SCFP2 用于节点重新发送在 SCFP1 发送失败的数据; SCFP3 用于继续传输节点在该第一 SCFP 或该第二 SCFP 未完全发送的数据, 包括重新发送在 SCFP2 发送失败的数据。

周期分配字段内列出协调器为网内设备分配的各自的工作周期时长。一个设备如果发现自己的短地址出现在周期分配字段的周期分配列表字段内, 则根据字段内的其他数据更新自己的工作周期。

转发数据目的地址列出与协调器保存的数据相对应的设备地址。一个设备如果发现自己的地址出现在待处理数据目的地址字段里, 则意味着协调器存有属于它的数据, 所以它就会向协调器发出请求传送数据的 MAC 命令帧。

信标帧净荷数据为相邻层协议提供数据传输接口, 信标帧有效负载数据包括协调器规定的约定信道和备用信道信息。

### 5.3.3.2 数据帧

数据帧用来传输上层发到 MAC 层的数据, 按照功能不同分为 4 种, 分别为非监测型数据帧、监测型数据帧、质疑数据帧和更新数据帧。数据帧的净荷字段包含了上层所需要发送的数据。数据净荷发送至 MAC 层时, 被称为 MAC 服务数据单元(MSDU)。它的首尾被分别附加 MAC 帧头和 MAC 帧尾信息后, 就构成了 MAC 帧, 如图 8 所示。

MAC 帧传送至物理层后, 头部增加同步信息 SHR 和帧长度信息 PHR, 就成为物理帧的净荷 PPDU。其中同步信息 SHR 包括用于同步的前导码和 SFD 字段, 均为固定值。帧长度字段 PHR 表示 MAC 帧的长度, 为一个八位位组长且其中低 7 位为有效位, 所以 MAC 帧的长度不超过 127 个八位位组。

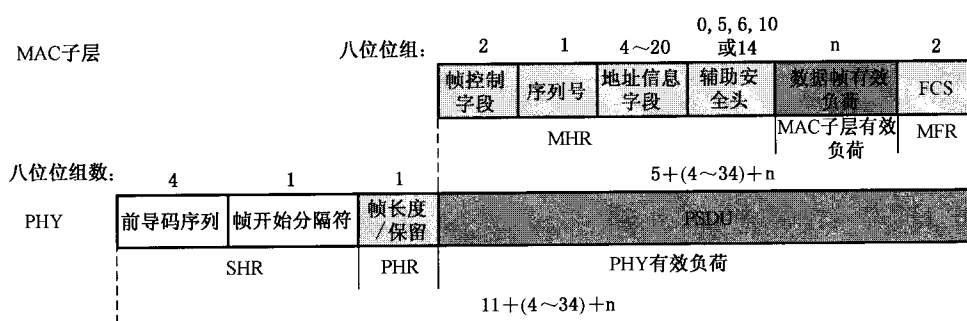


图 8 数据帧的格式

### 5.3.3.3 确认帧

如果设备收到目的地址为其自身的数据帧或 MAC 命令帧, 并且帧控制信息字段的确认请求位置为 1, 设备需要响应一个确认帧, 则收到的数据帧或 MAC 命令帧被称为被确认帧。确认帧的序列号应该与被确认帧的序列号相同, 且净荷长度应该为零。确认帧紧接着被确认帧发送, 不需要使用 CSMA-CA 机制竞争信道。确认帧按照功能不同分为 4 种, 分别为确认帧(普通)、数据接受确认帧、数据质疑无效确认帧和数据质疑有效确认帧。确认帧帧格式如图 9 所示。

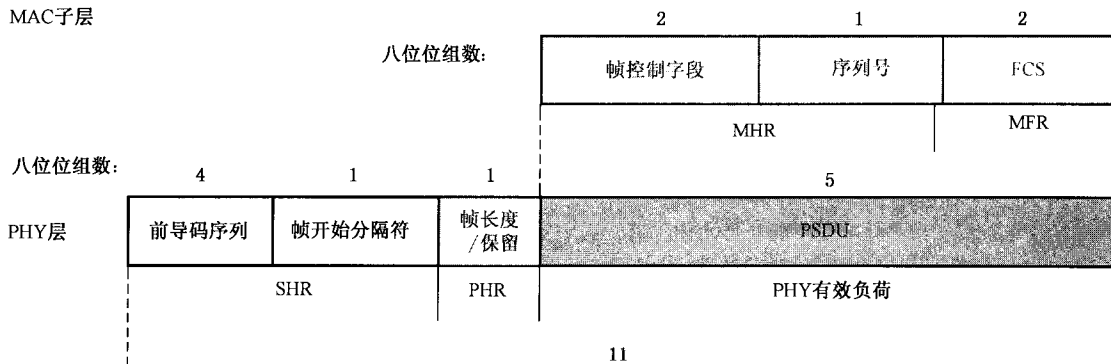


图9 确认帧的格式

### 5.3.3.4 命令帧

MAC 命令帧用于组建 RWSN, 传输同步数据等。命令帧在格式上和其他类型的帧没有太多区别, 只是帧控制字段的帧类型位有所不同, 命令帧定义如图 10 所示。帧头的帧控制字段的帧类型为 011b, 表示这是命令帧, 命令帧的具体功能由帧的净荷数据表示。净荷数据是一个变长结构, 所有命令帧净荷的第一个八位位组是命令类型八位位组, 后面的数据针对不同的命令类型有不同的含义。

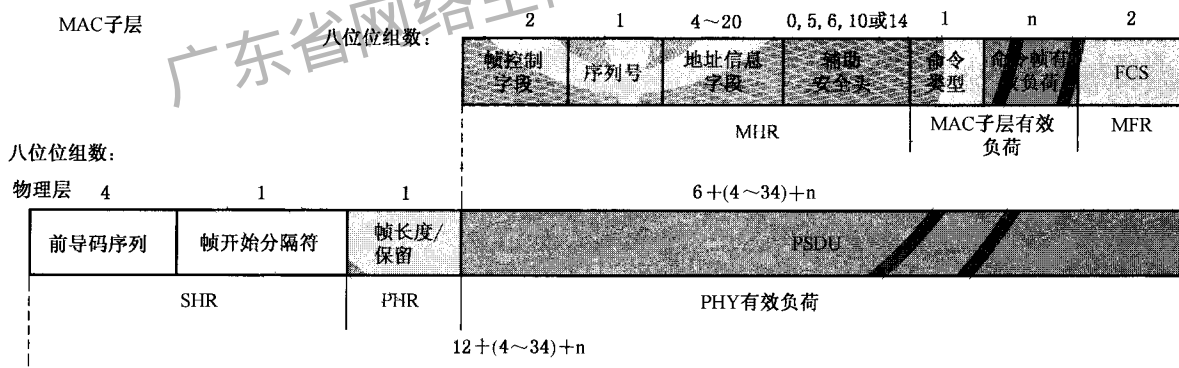


图10 命令帧的格式

### 5.3.4 鲁棒性

#### 5.3.4.1 概述

RWSN 采用 CSMA-CA 机制、帧确认机制和数据验证机制来保证数据传送的鲁棒性。

#### 5.3.4.2 CSMA-CA 机制

设备采用时隙的 CSMA-CA 机制, 退避时隙和信标帧的发送是同时开始。此外, 在同一个网络中, 所有设备的退避时隙由协调器排列。当一个设备要在 CAP 期间传输数据时, 应定位下一个退避时隙的起始边界, 并等待一个随机退避时间。如果感知信道忙, 在随机退避时间结束后, 设备应等待另一段随



机时间,然后再感知信道;如果发现信道空闲,设备则在下一个可用时隙传输数据。确认帧和信标帧的发送并不采取这一机制。

### 5.3.4.3 帧确认机制

帧确认机制是一种可选机制,发送帧的设备可要求目的设备在成功接收数据后发送确认,也可以不要求发送。设备只对数据帧和命令帧使用帧确认机制,在任何情况下都不会为信标帧或确认帧响应确认。设备发送一帧后,如果在一定的时间内没有收到确认帧,就认为传输失败,需要重新选择时机发送该帧。对于不要求确认的数据帧,发送以后就认为发送成功,并从本地缓冲队列中删除该数据帧。

### 5.3.4.4 数据验证机制

在 RWSN 中有两种机制解决传输误码问题。一种机制是使用短帧格式(小于 128 个八位位组)以减少单个帧出错的概率;另一种机制是利用 MAC 帧中的验证机制验证收到的数据是否出错。MAC 帧的校验码长 16 比特,使用 ITU 标准的 16 比特 CRC 校验生成算法产生。

## 5.4 原语的概念

本条将给出服务原语概念的一个简单描述。更详细的规定见 GB/T 15629.2。

某层提供的服务是指该层在下层提供的服务的基础供应上层或者子层的能力。在图 11 给出了通信双方和它们相关的对等协议实体之间的关系和服务层次。

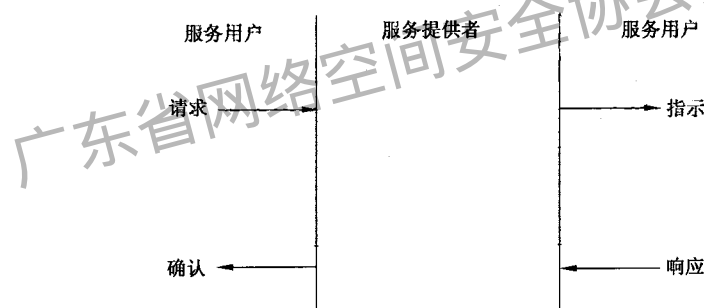


图 11 服务原语

服务通过和服务用户和服务提供者之间传递的信息流来详细说明。这种服务提供信息流采用离散、瞬时事件来模拟。每个事件都是通过层与服务用户相关的 SAP 从一个层向另一个层发送服务原语。服务原语通过提供特定的服务来传达要求的信息。服务原语都是抽象的,因为它们只指定提供的服务,而不指定提供服务的方法。这个定义和其他任何的接口执行相互独立。

服务通过服务原语和该原语相关的参数来指定。一个服务可能有一个或者多个相关的原语,由这些原语构造特定的服务。每一个服务原语都有零个或者多个用来传递提供服务需要信息的参数。

一个原语可能是以下的四个基本原语之一:

- 请求:请求原语从服务用户发送给服务提供者,请求初始化一个服务;
- 指示:指示原语由服务提供者发送给服务用户,通知一个和服务用户相关的内部服务提供者事件。该事件可能和一个远程服务请求逻辑上相关,也可能是由一个服务提供者内部事件引起的;
- 响应:响应原语由服务用户发送给服务提供者,来完成前面由通知原语调用的过程;
- 确认:确认原语由服务提供者发送给服务用户,来传达和先前服务原语相关的结果。

## 6 物理层规范

### 6.1 一般要求

物理层的一般要求如下：

- a) 使用频率:470 MHz~510 MHz;
- b) 发射功率限值:50 mW(ERP);
- c) 占用带宽:不大于 200 kHz;
- d) 频率容限: $100 \times 10^{-6}$ 。

物理层主要作用包括：

- a) 打开和关闭收发器;
- b) 对当前工作信道进行能量检测;
- c) 对接收到的数据包进行链路质量检测并指示;
- d) 为 CSMA/CA 做信道空闲评估;
- e) 信道频率选择;
- f) 数据发送和接收。

在本章中,物理层规定的常数和维度的属性用斜体书写。所有的常数列于表 20 中,其前缀为“a”,如, *aMaxPHYPacketSize*。所有的属性列于表 21 中,其前缀为“phy”,如 *phyCurrentChannel*。

### 6.2 操作频率范围

#### 6.2.1 概述

本部分中使用的频段以及相应的调制和扩频方式见表 1。

表 1 使用频段和数据速率

频带 MHz	扩频参数		数据参数	
	码片速率 Mchip/s	调制方法	比特速率 kbit/s	符号速率 kbit/s
470~510	0.1	O-QPSK	25	6.25

#### 6.2.2 信道划分

物理层总共有 16 个可用信道页。物理层 RWSN 信息库中的属性 *phyPagesSupported* 表示当前 PHY 支持哪些信道页,属性 *phyCurrentPage* 表示当前 PHY 所使用的信道页。PHY RIB 的属性表示方式见表 2。

在设置 *phyCurrentPage* 时,如果新的信道页不同于当前的信道页,且当前的工作信道不在新的信道页中,则 PHY 应当把当前工作信道设置为新信道页的第一个信道,即编号最小的那个信道。

示例：

信道页 1 的编号为 13 的信道可以映射为比特：

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

信道页 0~11,一共有 192 个可用信道,编号分别为 0~191,这些信道包含 470 MHz~510 MHz 频段。这些信道的中心频率为：

$$F_c = 470.1 + 0.2 \times k \text{ (MHz)}; k = 0, 1, \dots, 191$$

信道页 12, 有 8 个可用信道, 编号为 192~199。这些信道包含 470 MHz~510 MHz 频段。这些信道的中心频率为:

$$F_c = 508.3 + 0.2 \times k \text{ (MHz)}; k = 0, \dots, 7$$

信道页 13~15 保留, 用于将来的频段。

对于每一种可支持的物理层方案, 应支持所有该设备工作区字段规范内授权的信道。

表 2 信道页和信道编号

信道页 (十进制)	信道页(二进制) (b19, b18, b17, b16)	信道编号(十进制)					
		(b15, b14, b13, ..., b2, b1, b0)					
0	0000	180	...	24	12	0	0~199 号信道均工作在 470 MHz~510 MHz 频段, 调制方式均为 OQPSK, 且 按频率由低至高依次编号, 0~191 号信道按信道号均 匀分布在信道页 0~11 中, 且每个信道页中相邻信道 信道号相差 12。192~199 号信道连续分布在信道页 12 中
1	0001	181	...	25	13	1	
2	0010	182	...	26	14	2	
3	0011	183	...	27	15	3	
4	0100	184	...	28	16	4	
5	0101	185	...	29	17	5	
...	...	...	...	...	...	...	
11	1011	191	...	35	23	11	
12	1100	199 198 197 196 195 194 193 192					
13~15	1101~1111	保留					

### 6.2.3 最小的长帧间间隔(LIFS)和短帧间间隔(SIFS)周期

每种 PHY 对应的最小长帧间间隔和短帧间间隔周期见表 3, 长帧间间隔和短帧间间隔的描述、使用及图例在 7.5.2.3 的图 60 中示出。

表 3 最小 LIFS 和 SIFS 周期

PHY	最小长帧间间隔周期 macMinLIFSPeriod	最小短帧间间隔周期 macMinSIFSPeriod	单元
470 MHz 频段 O-QPSK	4	2	符号

### 6.2.4 射频功率测量

除非另外声明, 所有的射频功率测量, 不管是发射还是接收, 都应在收发器到天线的接口上进行。测量时应使用与天线接口阻抗相匹配的装置或补偿可能产生的失配。对没有天线接口的设备来讲, 测量应被解释为 EIRP(也就说, 0 dBi 增益的天线), 在所有辐射的测量中, 应补偿天线增益的影响。

### 6.2.5 发射功率

发射功率限值为 50 mW(ERP), 且对符合本部分的设备其标称发射功率应由其 PHY 参数 *phyTransmitPower*(见 6.5)说明。

### 6.2.6 带外杂散辐射

带外杂散辐射具体要求《微功率(短距离)无线电设备的技术要求》相关规定。

6.2.7 接收机灵敏度

表 4 给出了接收机灵敏度的定义,在本部分的其他章条中,凡涉及接收机灵敏度的地方,都应参考此定义。

表 4 接收机灵敏度定义

术语	定义	条件
接收机灵敏度	产生规定 PER 时,输入信号的功率门限	-PSDU 长度=20 八位位组 -PER<1% -在天线终端上测量功率 -不存在干扰
误包率(PER)	发射数据包被错误检测的平均比率	-PSDU 里为随机数据,取测量平均值

6.3 PHY 服务规范

6.3.1 协议参考模型

PHY 包含一个管理实体,称为 PLME。这个实体提供一个 PHY 管理的服务接口,通过它可以调用 PHY 管理功能。PLME 也负责维护一个 PHY 可管理对象的数据库。这个数据库被称做 PHY RIB。

图 12 描述了 PHY 的部件和接口。

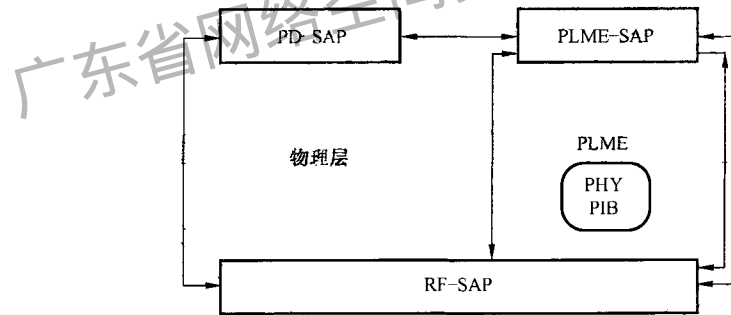


图 12 PHY 参考模型

通过访问 PHY 数据服务接入点(PD-SAP)和 PHY 管理服务接入点(PLME-SAP)两个服务接入点,PHY 可以提供两种服务:PHY 数据服务和 PHY 管理服务。

6.3.2 PHY 数据服务

6.3.2.1 概述

PD-SAP 支持 MPDU 在 MAC 层对等实体间的传输。表 5 列出了 PD-SAP 支持的原语,这些原语会在表 5 中所列的章条里详细描述。

表 5 PD-SAP 的原语

PD-SAP 原语	请求	确认	指示
数据原语 PD-DATA	0	6.3.2.3	6.3.2.4

### 6.3.2.2 PD-DATA.request

#### 6.3.2.2.1 服务原语的语义

PD-DATA.request 原语请求一个从 MAC 层到本地 PHY 实体的 MPDU(也即是 PSDU)传输。

PD-DATA.request 原语的语义如下：

```
PD-DATA.request (
    psduLength,
    psdu
)
```

表 6 规定了 PD-DATA.request 原语的参数。

表 6 PD-DATA.request 的参数

名称	类型	有效范围	描述
psduLength	无符号整型	$\leq aMaxPHYPacketSize$	PHY 实体将要传输的 PSDU 中包含的八位位组数
psdu	八位位组集	--	组成 PHY 实体将要发送的 PSDU 的八位位组集

#### 6.3.2.2.2 产生条件

当 MAC 层实体需要传输一个 MPDU 时,产生 PD-DATA.request 原语,并发布给其 PHY 实体。

#### 6.3.2.2.3 收后效果

收到 PD-DATA.request 原语后,PHY 实体会发送此原语所提供的 PSDU。如果发射机是打开状态(TX\_ON),PHY 实体会构建一个包含所提供 PSDU 的 PPDU,然后发送这个 PPDU。发送完成后,PHY 实体会发布 PD-DATA.confirm 原语,状态参数为 SUCCESS。

如果收到 PD-DATA.request 原语时接收机是打开状态(RX\_ON)或收发器是关闭状态(TRX\_OFF),PHY 实体也会发布 PD-DATA.confirm 原语,状态参数分别为 RX\_ON 或 TRX\_OFF。

### 6.3.2.3 PD-DATA.confirm

#### 6.3.2.3.1 服务原语的语义

PD-DATA.confirm 原语确认从本地 MAC 层实体到另一对等 MAC 层实体 MPDU(即 PSDU)发送的结束。

PD-DATA.confirm 原语的语义如下：

```
PD-DATA.confirm (
    status
)
```

表 7 规定了 PD-DATA.confirm 原语的参数。

表 7 PD-DATA.confirm 的参数

名称	类型	有效范围	描述
status	枚举	SUCCESS, RX_ON 或 TRX_OFF	发送包请求的结果

## 6.3.2.3.2 产生条件

当 PHY 实体响应 PD-DATA.request 原语时,产生 PD-DATA.confirm 原语,并发布给其 MAC 层实体。PD-DATA.confirm 原语返回的状态参数可以是 SUCCESS,表示数据传输请求得到了成功执行,也可以是 RX\_ON 或 TRX\_OFF,表示发生了错误。产生这些状态值的原因见 6.3.2.2.3。

## 6.3.2.3.3 收后效果

收到 PD-DATA.confirm 原语后,MAC 层实体获知了数据发送请求的结果。如果发送成功,状态参数为 SUCCESS,否则,状态参数会指示所发生的错误。

## 6.3.2.4 PD-DATA.indication

## 6.3.2.4.1 服务原语的语义

PD-DATA.indication 原语指示从 PHY 到本地 MAC 层实体的一个 MPDU(也就是 PSDU)传输。

PD-DATA.indication 原语的语义如下:

PD-DATA.indication (

psduLength,

psdu

ppduLinkQuality

)

表 8 规定了 PD-DATA.indication 原语的参数。

表 8 PD-DATA.indication 的参数

名称	类型	有效范围	描述
psduLength	无符号整型	$\leq aMaxPHYPacketSize$	PHY 实体收到的 PSDU 中所包含的八位位组数
psdu	八位位组集	—	组成 PHY 实体收到的 PSDU 的八位位组集
ppduLinkQuality	整型	0x00~0xff	在接收 PPDU 过程中测到的链路质量(LQI)值(见 6.6.3.6.8)

## 6.3.2.4.2 产生条件

当 PHY 实体需要传送一个接收到的 PSDU 时,产生 PD-DATA.indication 原语,并发布给它的 MAC 层实体。当接收到的 psduLength 的值为零或大于  $aMaxPHYPacketSize$  时,不产生此原语。

## 6.3.2.4.3 收后效果

收到 PD-DATA.indication 原语后,MAC 层实体通过 PHY 数据服务获知了一个 MPDU 的到来。

## 6.3.3 PHY 管理服务

## 6.3.3.1 概述

PLME-SAP 允许管理指令在 MLME 和 PLME 之间传输。表 9 列出了 PLME-SAP 支持的原语。这些原语将在表 9 中列出的条中详细讨论。

表 9 PLME-SAP 的原语

名 称	Request	Confirm
空闲信道评估原语(PLME-CCA)	0	6.3.3.3
能量检测原语(PLME-ED)	6.3.3.4	6.3.3.5
读取 PHY RIB 属性原语 PLME-GET)	6.3.3.6	6.3.3.7
设置 TRX 状态原语(PLME-SET-TRX-STATE)	6.3.3.8	6.3.3.9
写入 PHY RIB 属性原语(PLME-SET)	6.3.3.10	6.3.3.11

### 6.3.3.2 PLME-CCA.request

#### 6.3.3.2.1 服务原语的语义

PLME-CCA.request 原语请求 PLME 执行一次 CCA, CCA 的实现见 6.6.3.6.9。

PLME-CCA.request 原语的语义如下:

PLME-CCA.request ( )

PLME-CCA.request 原语没有参数。

#### 6.3.3.2.2 产生条件

当 CSMA-CA 算法需要对信道进行评估时, MLME 产生 PLME-CCA.request 原语, 并发布给它的 PLME。

#### 6.3.3.2.3 收后效果

如果收到 PLME-CCA.request 原语时接收机是打开状态, PLME 会使 PHY 执行一次 CCA。PHY 完成 CCA 后, PLME 会发布一个 PLME-CCA.confirm 原语, 根据 CCA 的结果, 其状态参数为 BUSY 或 IDLE。

如果收到 PLME-CCA.request 原语时收发器是关闭状态 (TRX\_OFF) 或发射机是打开状态 (TX\_ON), PLME 也会发布 PLME-CCA.confirm 原语, 状态参数为 TRX\_OFF 或 TX\_ON。

### 6.3.3.3 PLME-CCA.confirm

#### 6.3.3.3.1 服务原语的语义

PLME-CCA.confirm 原语报告 CCA 的结果。

PLME-CCA.confirm 原语的语义如下:

PLME-CCA.confirm (   
 status   
 )

#### 6.3.3.3.2 PLME-CCA.confirm 的参数

表 10 规定了 PLME-CCA.confirm 原语的参数。

表 10 PLME-CCA.confirm 的参数

名称	类型	有效范围	描述
status	枚举	TRX_OFF, TX_ON, BUSY 或 IDLE	请求执行 CCA 的结果

6.3.3.3.3 产生条件

当 PLME 响应 PLME-CCA.request 原语时,产生 PLME-CCA.confirm 原语,并发布给它的 MLME。PLME-CCA.confirm 会返回状态参数 BUSY 或 IDLE,表示成功执行了 CCA;或者返回状态参数 TRX\_OFF 或 TX\_ON,表示发生了错误。产生这些状态值的原因见 6.3.3.2.3。

6.3.3.3.4 收后效果

收到 PLME-CCA.confirm 原语后,MLME 获知了 CCA 的结果。如果成功执行了 CCA,状态参数为 BUSY 或 IDLE。否则,状态参数会指示错误。

6.3.3.4 PLME-ED.request

6.3.3.4.1 服务原语的语义

PLME-ED.request 原语请求 PLME 执行一次 ED 测量(见 6.6.3.6)。

PLME-ED.request 原语的语义如下:

PLME-ED.request ( )

PLME-ED.request 原语没有参数。

6.3.3.4.2 产生条件

当 MLME 需要一个 ED 测量值时,产生 PLME-ED.request 原语,并发布给它的 PLME。

6.3.3.4.3 收后效果

如果收到 PLME-ED.request 原语时接收机是打开状态,PLME 会使 PHY 执行一次 ED 测量。PHY 完成 ED 测量后,PLME 会发布一个 PLME-ED.confirm 原语,状态参数为 SUCCESS。

如果收到 PLME-ED.request 原语时收发器是关闭状态(TRX\_OFF)或发射机是打开状态(TX\_ON),PLME 也会发布 PLME-CCA.confirm 原语,状态参数为 TRX\_OFF 或 TX\_ON。

6.3.3.5 PLME-ED.confirm

6.3.3.5.1 服务原语的语义

PLME-ED.confirm 原语报告 ED 测量的结果。

PLME-ED.confirm 原语的语义如下:

PLME-ED.confirm (

status,

EnergyLevel

)

表 11 规定了 PLME-ED.confirm 原语的参数。

表 11 PLME- ED.confirm 的参数

名称	类型	有效范围	描述
status	枚举	SUCCESS, TRX_OFF 或 TX_ON	请求执行 ED 测量的结果
EnergyLevel	整型	0x00~0xff	当前信道的 ED 层级



### 6.3.3.5.2 产生条件

当 PLME 响应 PLME-ED.request 原语时,产生 PLME-ED.confirm 原语,并发布给它的 MLME。PLME-ED.confirm 会返回状态参数 SUCCESS,表示成功执行了 ED 测量;或者返回状态参数 TRX\_OFF 或 TX\_ON,表示发生了错误。产生这些状态值的原因见 6.3.3.4.3。

### 6.3.3.5.3 收后效果

收到 PLME-ED.confirm 原语后,MLME 获知了 ED 测量的结果。如果 ED 测量得到了成功执行,状态参数为 SUCCESS。否则,状态参数会指示相应的错误。

### 6.3.3.6 PLME-GET.request

#### 6.3.3.6.1 服务原语的语义

PLME-GET.request 原语请求获知某一 PHYRIB 属性的信息。

PLME-GET.request 原语的语义如下:

```
PLME-GET.request (
    RIBAttribute
)
```

表 12 规定了 PLME-GET.request 原语的参数。

表 12 PLME-GET.request 的参数

名称	类型	有效范围	描述
RIBAttribute	枚举	见表 21	要获取的 PHY RIB 属性标识符

#### 6.3.3.6.2 产生条件

当 MLME 需要从 PHYRIB 中获得信息时,产生 PLME-GET.request 原语,并发布给它的 PLME。

#### 6.3.3.6.3 收后效果

收到 PLME-GET.request 原语后,PLME 会在它的数据库中搜寻这个被请求的 PHYRIB 属性。如果在数据库中找到此属性的标识,PLME 会发布一个 PLME-GET.confirm 原语,状态参数为 UNSUPPORTED\_ATTRIBUTE。

如果成功获取了被请求的 PHYRIB 属性,PLME 也会发布 PLME-GET.confirm 原语,不过状态参数为 SUCCESS。

### 6.3.3.7 PLME-GET.confirm

#### 6.3.3.7.1 服务原语的语义

PLME-GET.confirm 原语报告 PHYRIB 获取信息的结果。

PLME-GET.confirm 原语的语义如下:

```
PLME-GET.confirm (
    status,
    RIBAttribute,
    RIBAttributeValue
)
```

表 13 规定了 PLME-GET.confirm 原语的参数。

表 13 PLME-GET.confirm 的参数

名称	类型	有效范围	描述
status	枚举	SUCCESS 或 UNSUPPORTED_ATTRIBUTE	请求获取 PHY RIB 属性信息的结果
RIBAttribute	枚举	见表 21	请求的 PHY RIB 属性的标识符
RIBAttributeValue	多样	跟属性有关	所请求的 PHY RIB 属性的取值。 当 status 参数为 UNSUPPORTED_ATTRIBUTE 时,本参数为 0

6.3.3.7.2 产生条件

当 PLME 响应 PLME-GET.request 原语时,产生 PLME-GET.confirm 原语,并发布给它的 MLME。PLME-GET.confirm 会返回状态参数 SUCCESS,表示成功读取了 PHYRIB 属性信息,或者返回错误代码 UNSUPPORTED\_ATTRIBUTE。产生这些状态值的原因见 6.3.3.6.3。

6.3.3.7.3 收后效果

收到 PLME-GET.confirm 原语后,MLME 获知了它请求读取 PHYRIB 属性信息的结果。如果读取 PHYRIB 属性的请求被成功执行,状态参数为 SUCCESS。否则,状态参数会指示相应的错误。

6.3.3.8 PLME-SET-TRX-STATE.request

6.3.3.8.1 服务原语的语义

PLME-SET-TRX-STATE.request 原语请求 PHY 实体改变收发器的内部工作状态。

收发器有三种主要状态:

- a) 收发器关闭(TRX\_OFF);
- b) 发射机打开(TX\_ON);
- c) 接收机打开(RX\_ON)。

PLME-SET-TRX-STATE.request 原语的语义如下:

```
PLME-SET-TRX-STATE.request (
    status
)
```

表 14 规定了 PLME-SET-TRX-STATE.request 原语的参数。

表 14 PLME-SET-TRX-STATE.request 的参数

名称	类型	有效范围	描述
status	枚举	RX_ON, TRX_OFF, FORCE_TRX_OFF, 或 TX_ON	收发器配置的新状态。

6.3.3.8.2 产生条件

当收发器当前的工作状态需要被改变时,MLME 生成 PLME-SET-TRX-STATE.request 原语并

发布给它的 PLME。

### 6.3.3.8.3 收后效果

收到 PLME-SET-TRX-STATE.request 原语后,PLME 会使 PHY 转换到所请求的状态。如果状态转换被接受,PHY 会发布 PLME-SET-TRX-STATE.confirm 原语,状态参数为 SUCCESS。如果此原语所要求的状态就是收发器当前的配置状态,PHY 也会发布 PLME-SET-TRX-STATE.confirm 原语,指示当前状态的参数,如 RX\_ON, TRX\_OFF 或 TX\_ON。如果此原语发布时包含 RX\_ON 或 TRX\_OFF 的变量,而 PHY 此时正在发送一个 PPDU,那么 PHY 会发布状态为 BUSY\_TX 的 PLME-SET-TRX-STATE.confirm 原语并推迟状态转换直到发送结束。如果此原语发布时包含 TX\_ON 或 TRX\_OFF 的变量,而 PHY 此时处于 RX\_ON 状态并且已经接收到一个有效的 SFD,那么物理会发布状态为 BUSY\_RX 的 PLME-SET-TRX-STATE.confirm 原语并推迟状态转换直到完成 PPDU 的接收。如果此原语发布时包含的变量为 FORCE\_TRX\_OFF,那么 PHY 会无条件地转换到 TRX\_OFF 状态,不管当前处在什么状态。

### 6.3.3.9 PLME-SET-TRX-STATE.confirm

#### 6.3.3.9.1 服务原语的语义

PLME-SET-TRX-STATE.confirm 原语报告请求转换收发器内部工作状态的结果。

PLME-SET-TRX-STATE.confirm 原语的语义如下:

```
PLME-SET-TRX-STATE.confirm (
    status
)
```

表 15 规定了 PLME-SET-TRX-STATE.confirm 原语的参数。

表 15 PLME-SET-TRX-STATE.confirm 的参数

名称	类型	有效范围	描述
status	枚举	SUCCESS, RX_ON, TRX_OFF, TX_ON, BUSY_RX 或 BUSY_TX	请求转换收发器状态的结果

#### 6.3.3.9.2 产生条件

当 PLME 试着改变收发器的内部工作状态后,生成 PLME-SET-TRX-STATE.confirm 原语并发布给它的 MLME。

#### 6.3.3.9.3 收后效果

收到 PLME-SET-TRX-STATE.confirm 原语后,MLME 获知了它请求转换收发器内部工作状态的结果。状态值 SUCCESS 表示转换收发器内部工作状态的请求已被接受。状态值 RX\_ON, TRX\_OFF 或 TX\_ON 表示收发器已经工作在它所请求转换到的状态。状态值 BUSY\_TX 表示 PHY 正在发送数据却被要求转换到 RX\_ON 或 TRX\_OFF 状态。状态值 BUSY\_RX 表示 PHY 正工作在 RX\_ON 状态、并且已经收到一个有效的 SFD、却被要求转换到 TX\_ON 或 TRX\_OFF 状态。

### 6.3.3.10 PLME-SET.request

#### 6.3.3.10.1 服务原语的语义

PLME-SET.request 原语请求设置某一 PHYRIB 属性为给定的值。

PLME-SET.request 原语的语义如下：

```
PLME-SET.request (
    RIBAttribute,
    RIBAttributeValue
)
```

表 16 规定了 PLME-SET.request 原语的参数。

表 16 PLME-SET.request 的参数

名称	类型	有效范围	描述
RIBAttribute	枚举	见表 21	需要设置的 RIB 属性的标识符
RIBAttributeValue	多样	跟属性有关	需要设置的指定 RIB 的属性值

### 6.3.3.10.2 产生条件

当 MLME 需要改写某一 PHY RIB 属性的取值时，产生 PLME-SET.request 原语，并发布给它的 PLME。

### 6.3.3.10.3 收后效果

收到 PLME-SET.request 原语后，PLME 会试着把给定的值写入它数据库中某一指定的 PHYRIB 属性。如果在数据库中找到 RIBAttribute 参数所指定的 PHYRIB 属性，PLME 会发布状态参数 UNSUPPORTED\_ATTRIBUTE 的 PLME-SET.confirm 原语。如果 RIBAttributeValue 参数所指定的属性的值不在有效有效范围内，PLME 会发布状态参数为 INVALID\_PARAMETER 的 PLME-SET.confirm 原语。

如果所请求的 PHYRIB 属性被成功写入，PLME 会发布状态参数为 SUCCESS 的 PLME-SET.confirm 原语。

### 6.3.3.11 PLME-SET.confirm

#### 6.3.3.11.1 服务原语的语义

PLME-SET.confirm 原语报告设置 RIB 属性的结果。

PLME-SET.confirm 原语的语义如下：

```
PLME-SET.confirm (
    status,
    RIBAttribute
)
```

表 17 规定了 PLME-SET.confirm 原语的参数。

表 17 PLME-SET.confirm 的参数

名称	类型	有效范围	描述
status	枚举	SUCCESS, UNSUPPORTED_ATTRIBUTE 或 INVALID_PARAMETER	尝试设置请求的 RIB 属性的结果
RIBAttribute	枚举	见表 21	被确认的 RIB 属性的标识符

### 6.3.3.11.2 产生条件

当 PLME 响应 PLME-SET.request 原语时,产生 PLME-SET.confirm 原语,并发布给它的 MLME。如果所要求的值被写入指定的 PHYRIB 属性,PLME-SET.confirm 会返回状态参数 SUCCESS,否则,它会返回错误代码 UNSUPPORTED\_ATTRIBUTE 或 INVALID\_PARAMETER。产生这些状态值的原因见 6.3.3.10.3。

### 6.3.3.11.3 收后效果

收到 PLME-SET.confirm 原语后,MLME 获知了它请求设置 PHYRIB 属性的结果。如果所要求的值被写入指定的 PHYRIB 属性,状态参数为 SUCCESS。否则,状态参数会指示相应的错误。

### 6.3.3.12 PHY 枚举变量描述

表 18 给出了物理层规范中定义的 PHY 枚举变量的描述。

表 18 PHY 枚举变量描述

枚举	取值	描述
BUSY	0x00	CCA 检测到信道忙
BUSY_RX	0x01	收发器正在接收数据时被要求转换状态
BUSY_TX	0x02	收发器正在发送数据时被要求转换状态
FORCE_TRX_OFF	0x03	收发器将要被关闭
IDLE	0x04	CCA 检测到信道空闲
INVALID_PARAMETER	0x05	SET/GET 请求被发布时,其原语中的参数不在有效范围之内
RX_ON	0x06	收发器已经在或将要被配置成接收机打开的状态
SUCCESS	0x07	成功进行了 SET/GET、ED 操作或收发器状态转换
TRX_OFF	0x08	收发器已经在或将要被配置成发射机关闭的状态
TX_ON	0x09	收发器已经在或将要被配置成发射机打开的状态
UNSUPPORTED_ATTRIBUTE	0x0a	SET/GET 请求被发布时,其中的属性标识是不被支持的
R-O	0x0c	CMA 检测到信道调制方式是 OQPSK

## 6.4 PPDU 格式

### 6.4.1 概述

本条规定 PPDU 包的格式。

在本标准中,位于 PPDU 数据结构最左边的字段应最先被发射或接收。对所有具有多个八位位组的字段,应最先发射或接收最低位的八位位组;对每个八位位组,应最先发射或接收最低位的比特。当 PHY 和 MAC 层之间传输数据字段时,也应遵循这样的顺序。

每个 PPDU 包由下述基本单元所组成:

- SHR 单元,它使接收设备能够同步和锁定比特流;
- PHR 单元,它包含帧长信息;

c) 可变长度净荷单元,它携带 MAC 层的帧。

### 6.4.2 一般数据包的格式

#### 6.4.2.1 PPDU 的格式

PPDU 的结构应如图 13 所示。

		八位位组数		
		1		可变
Preamble	SFD	帧长 (7 个比特)	保留位 (1 个比特)	PSDU
SHR		PHR		PHY 净荷

图 13 PPDU 的格式

#### 6.4.2.2 Preamble 字段

Preamble 字段被收发器用来对接收信号进行码片和符号同步。Preamble 字段应由 32 个二进制零组成。

#### 6.4.2.3 SFD 字段

SFD 字段用来指示同步(Preamble)字段的结束和数据传输的开始。SFD 应由 8 个二进制零组成。

#### 6.4.2.4 帧长字段

帧长字段由 7 个比特组成,表示 PSDU(也就是 PHY 净荷)中包含的八位位组数。这个数值位于 0 到  $aMaxPHYPacketSize$ (见 6.5)之间。表 19 描述了净荷类型和帧长之间的关系。

表 19 帧的长度

帧的长度	净荷类型
0~4	保留
5	MPDU(Acknowledgment)
6~8	保留
9 到 $aMaxPHYPacketSize$	MPDU

#### 6.4.2.5 PSDU 字段

PSDU 字段具有可变的长度,负责传输 PHY 数据包里的数据。对所有帧长等于五个八位位组或大于七个八位位组的数据包类型,PSDU 包含的是 MAC 层的数据帧(也就是 MPDU)。

### 6.5 PHY 常数和 RIB 属性

#### 6.5.1 PHY 常数

表 20 列出了定义 PHY 特性的一些常数。这些常数是依赖于硬件的,不能在操作中被改变。

表 20 PHY 常数

常数	描述	取值
<i>aMaxPHYPacketSize</i>	PHY 应能接收的最大 PSDU 大小(八位位组)	127
<i>aTurnaroundTime</i>	RX-to-TX 或 TX-to-RX 的最大周转时间(见 6.6.3.6.1 和 6.6.3.6.2)	12

## 6.5.2 PHY RIB 属性

PHYRIB 包含了管理一个设备的 PHY 所需要的属性。每个这样的属性都可以使用 PLME-GET.request 和 PLME-SET.request 原语进行读写操作。其中,以(♯)标注的属性为只读参数;以(\*)标注的属性为特定位只读参数。表 21 列出了 PHY RIB 所包含的属性。

表 21 PHY RIB 属性

属性	标识	类型	有效范围	描述
<i>phyCurrentChannel</i>	0x00	整型	0~15	所有下述发送和接收所使用的 RF 信道(见 6.2)。
<i>PhyChannelsSupported</i> ♯	0x01	阵列	见描述	<i>phyChannelsSupported</i> 的 MSB 的前 4 比特 ( $b_{15}, \dots, b_{12}$ ) 应被设为信道页, LSB 的后 16 比特 ( $b_0, b_1, \dots, b_{15}$ ) 应指示信道页中有效信道中每个信道的状态(1=可用,0=不可用)( $b_k$ 应指示信道 k 的状态)
<i>phyTransmitPower</i> *	0x02	位图	0x00~0xbf	MSB 的前 2 比特表示对发射功率的误差容限: 00 = ±1 dB 01 = ±3 dB 10 = ±6 dB LSB 的后 6 位表示一个二进制补码格式的有符号整数,对应于设备的标称发射功率 1 mW 的 dB 值。 <i>phyTransmitPower</i> 的最小值应被解释为小于或等于 -32 dBm
<i>PhyCCAMode</i>	0x03	整型	1~3	CCA 方式(见 6.6.3.6.9)
<i>phyCurrentPage</i>	0x04	整型	0~15	当前使用的 PHY 信道页 与 <i>phyCurrentChannel</i> 一起唯一标识当前使用的信道
<i>phyMaxFrameDuration</i> ♯	0x05	整型	55,212, 266,1064	一帧中最多能包含的符号个数 = $phySHRDuration + ceiling([aMaxPHYPacketSize + 1] \times phySymbolsPerOctet)$
<i>phySHRDuration</i> ♯	0x06	整型	3,7,10,40	当前 PHY 所使用的同步头(SHR)间隔
<i>phySymbolsPerOctet</i> ♯	0x07	浮点型	0.4,1.6,2,8	当前 PHY 每个八位位组所含的符号个数

6.6 技术要求

6.6.1 数据速率

PHY 的数据速率应为 25 kb/s。

6.6.2 调制方式

6.6.2.1 概述

任何符合本部分的收发机设备都应实现 O-QPSK 调制方式。物理层使用 16 元准正交调制技术。在每个数据符号周期内,4 个信息位用来选择待发射的 16 元准正交的扩频码序列中的一个。相邻数据符号选出的扩频码序列连接在一起,序列中的码片经过 O-QPSK 调制到载波上。

6.6.2.2 调制器

图 14 是一个功能模块框图,作为规定 470 MHz PHY 调制和扩频函数的参考。图中每个模块里的数字指出了下面描述这一功能的章节。

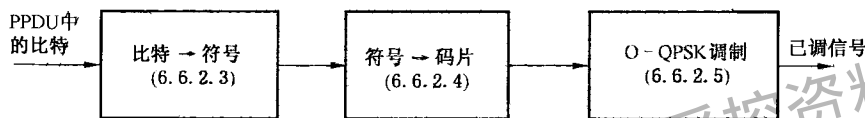


图 14 调制和扩频函数

6.6.2.3 比特到符号的映射

PPDU 中所有的二进制数据都应使用图 14 所示的调制和扩频函数进行处理。本条描述二进制信息是如何被映像成数据符号的。

每个八位位组的低四位( $b_0, b_1, b_2, b_3$ )应映像到一个数据符号,高四位( $b_4, b_5, b_6, b_7$ )应映像到下一个数据符号。PPDU 里的每个八位位组都应按顺序经过调制和扩频函数(如图 14 所示)的处理,从 Preamble 字段开始,到 PSDU 里的最后一个八位位组结束。在每一个八位位组里,先处理低位符号( $b_0, b_1, b_2, b_3$ ),再处理高位符号( $b_4, b_5, b_6, b_7$ )。

6.6.2.4 符号到码片的映射

每个数据符号都应被映像成一个 16 位的相位序列,如表 22 所示。这些相位序列之间是循环移位和/或共轭的关系。

表 22 符号到码片的映射

数据符号 (十进制)	数据符号 (二进制) ( $b_0, b_1, b_2, b_3$ )	扩频序列 ( $c_0, c_1, \dots, c_{14}, c_{15}$ )
0	0 0 0 0	0 0 1 1 1 1 1 1 0 0 0 1 0 0 1 0 1
1	1 0 0 0	0 1 0 0 1 1 1 1 1 1 0 0 0 1 0 0 1
2	0 1 0 0	0 1 0 1 0 0 1 1 1 1 1 1 0 0 0 1 0
3	1 1 0 0	1 0 0 1 0 1 0 0 1 1 1 1 1 1 0 0 0



表 22 (续)

数据符号 (十进制)	数据符号 (二进制) ( $b_0, b_1, b_2, b_3$ )	扩频序列 ( $c_0, c_1, \dots, c_{14}, c_{15}$ )
4	0 0 1 0	0 0 1 0 0 1 0 1 0 0 1 1 1 1 1 0
5	1 0 1 0	1 0 0 0 1 0 0 1 0 1 0 0 1 1 1 1
6	0 1 1 0	1 1 1 0 0 0 1 0 0 1 0 1 0 0 1 1
7	1 1 1 0	1 1 1 1 1 0 0 0 1 0 0 1 0 1 0 0
8	0 0 0 1	0 1 1 0 1 0 1 1 0 1 1 1 0 0 0 0
9	1 0 0 1	0 0 0 1 1 0 1 0 1 1 0 1 1 1 0 0
10	0 1 0 1	0 0 0 0 0 1 1 0 1 0 1 1 0 1 1 1
11	1 1 0 1	1 1 0 0 0 0 0 1 1 0 1 0 1 1 0 1
12	0 0 1 1	0 1 1 1 0 0 0 0 0 1 1 0 1 0 1 1
13	1 0 1 1	1 1 0 1 1 1 0 0 0 0 0 1 1 0 1 0
14	0 1 1 1	1 0 1 1 0 1 1 1 0 0 0 0 0 1 1 0
15	1 1 1 1	1 0 1 0 1 1 0 1 1 1 0 0 0 0 0 1

6.6.2.5 O-QPSK 调制

代表每个数据符号的扩频相位序列经 O-QPSK 和半正弦脉冲成形调制到载波上。偶数位的码片被调制到同相支路(I)载波上,而奇数的码片被调制到正交支路(Q)载波上。因为每个符号是由 16 个码片所表示的,所以码片速率是符号速率的 16 倍。为了形成在同相支路和正交支路码片调制之间的偏移,正交支路的码片比同相支路的码片有  $T_c$  的延迟(见图 15),其中  $T_c$  是符号速率的倒数。

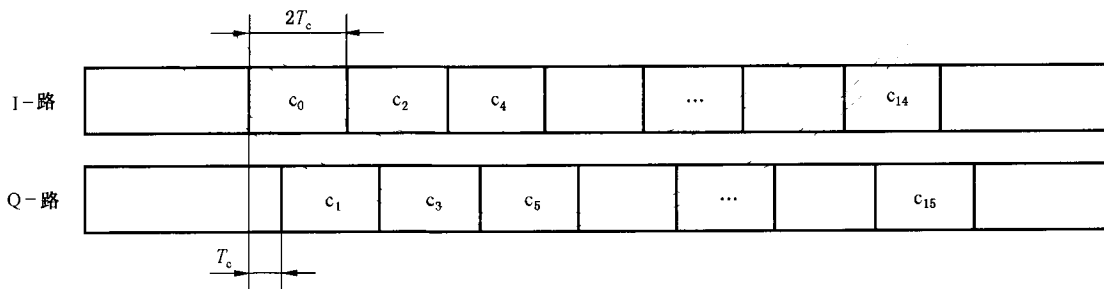


图 15 O-QPSK 调制码片偏移示意图

6.6.2.6 脉冲成形

升余弦滚降脉冲被用来表示每个基带码片,其表达式见式(1):

$$p(t) = \frac{\sin(\pi t/T_c)}{\pi t/T_c} \frac{\cos(r\pi t/T_c)}{1 - 4r^2 t^2/T_c^2} \dots\dots\dots(1)$$

式中:

$r$ ——滚降因子,  $r=0.8$ 。

## 6.6.2.7 码片发射顺序

在每一个符号周期里,最先发射的码片是最低位码片  $c_0$ ,最后发射的码片是最高位码片  $c_{15}$ 。

## 6.6.3 470 MHz~510 MHz 频段射频要求

## 6.6.3.1 概述

除应符合《微功率(短距离)无线电设备的技术要求》的要求以外,在 470 MHz~510 MHz 频段的设备还应满足 6.1、6.2 和 6.6.3.6 的要求。

## 6.6.3.2 发射功率谱密度

发射信号的谱分量应小于表 23 中的限制。不管是相对限制还是绝对限制,在测量平均功率谱时应使用 100 kHz 的分辨率带宽。对相对限制来讲,参考电平应是在载波频率附近±100 kHz 范围内测量得到的最大平均功率谱。

表 23 发射功率谱密度限制

频率	相对限制	绝对限制
$ f-f_c >120\text{ kHz}$	-20 dB	-20 dBm

## 6.6.3.3 符号速率

符号速率应为 6.25 ksymbol/s。

## 6.6.3.4 接收机灵敏度

在 6.2.7 所述条件指定下,接收机灵敏度为-85 dBm。

## 6.6.3.5 接收机干扰抑制

表 24 给出了接收机的最小干扰抑制能力。相邻信道是指期望信道两边与期望信道频率最接近的两个信道中的任一个信道,相隔信道是指相邻信道外边的两个信道。例如,当信道 13 是期望信道时,信道 12 和信道 14 是相邻信道,信道 11 和信道 15 是相隔信道。

表 24 470 MHz PHY 对接收机最小干扰抑制能力的要求

相邻信道抑制	相隔信道抑制
0 dB	30 dB

相邻信道抑制能力应使用下述方法进行测量。期望信号应是符合本部分 470 MHz 频段的 PHY 的伪随机数据信号。当期望信号输入接收机时,它的电平应比 6.6.3.4 给出的最大允许的接收机灵敏度高 3dB。

不管是相邻信道还是相隔信道,一个符合 470 MHz~510 MHz PHY 的信号作为干扰输入接收机时,它的相对电平按 6.6.3.4 中的值进行设定。当进行测试时,每次只能有一个干扰信号。在这样的条件下,接收机应能达到 6.2.7 所规定的误包率。

6.6.3.6 一般要求

6.6.3.6.1 TX-to-RX 的转换时间

TX-to-RX 的转换时间应小于或等于  $aTurnaroundTime$  (见 6.5.1)。

TX-to-RX 的转换时间应在空中接口处测量,从最后一个发射符号的尾沿开始,到接收机准备好接收下一个 PHY 数据包止。

6.6.3.6.2 RX-to-TX 的转换时间

RX-to-TX 的转换时间应小于或等于  $aTurnaroundTime$  (见 6.5.1)。

RX-to-TX 的转换时间应在空中接口处测量,从接收 PPDU 的最后一个码片(最后一个符号中的)的尾沿开始,到下一个传送 PPDU 的第一个码片的前沿(第一个符号)。实际传输开始的时间由 MAC 层加以规定。

6.6.3.6.3 EVM

本部分发射机的调制精度由 EVM 度量来确定。为了计算 EVM 度量,需要接收并存储  $N$  个复数码片值  $(\tilde{I}_j, \tilde{Q}_j)$ 。对每个接收到的复数码片值进行判决以得到相应的发射复数码片值。可以选择的复数码片的理想位置(判决区字段的中心)用矢量表示为  $(I_j, Q_j)$ 。定义误差矢量  $(\delta I_j, \delta Q_j)$  为理想位置 and 实际接收信号点位置之间的距离(见图 16)。

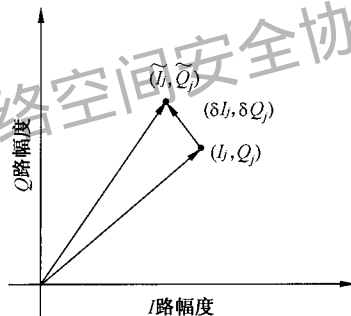


图 16 误差矢量计算

因此,接收矢量是理想矢量和误差矢量之和如式(2)所示。

$$(\tilde{I}_j, \tilde{Q}_j) = (I_j, Q_j) + (\delta I_j, \delta Q_j) \quad \dots\dots\dots (2)$$

EVM 定义为

$$EVM = \sqrt{\frac{1}{N} \sum_{j=1}^N (\delta I_j^2 + \delta Q_j^2)} \times 100\% \quad \dots\dots\dots (3)$$

式中:

$S$  ——到理想星座点矢量的幅值;

$(\delta I_j, \delta Q_j)$  ——误差矢量。

符合本部分的发射机,当使用 1 000 个码片进行测量时应具有小于 35% 的 EVM。误差矢量应使用基带的 I 路和 Q 路码片进行测量,这两路信号由一个参考接收机系统进行恢复。测量进行时,参考接收机应执行载波锁定、符号时间恢复以及幅度调整等功能。

#### 6.6.3.6.4 发射中心频率的误差容限

发射中心频率的误差容限最大为 $\pm 100 \times 10^{-6}$ 。

#### 6.6.3.6.5 发射功率

符合本标准的发射机应至少能发射 $-3$  dBm。在可能的时候,设备应降低发射功率以减少对其他设备和系统的干扰。

#### 6.6.3.6.6 接收机期望信号的最大输入电平

接收机最大输入电平是指当 6.2.7 规定的误包率准则满足时,接收机输入端的期望信号的最大功率电平,用 dBm 来表示。接收机应具有大于或等于 $-20$  dBm 的接收机最大输入电平。

#### 6.6.3.6.7 接收机 ED

接收机 ED 测量是为网络层所使用的,作为信道选择算法的一部分。它是对处于一个短距离无线网络中国标准信道带宽内的接收信号功率的估计,并不对这一信道中的信号进行辨识或解调。ED 的时间应等于 8 个符号周期。

ED 的结果,一个 0x00 到 0xff 之间的 8 比特整数,应使用 PLME-ED.confirm 原语(见 6.3.3.5)报告给 MLME。最小的 ED 值(0)应表示接收功率小于规定的接收机灵敏度(见 6.6.3.4 和 6.6.3.6)以上 10 dB 的那个值,并且 ED 应能表示至少 40dB 的接收功率范围。在这一范围内,从接收功率(以 dB 表示)到 ED 值的映像应为线性并具有 $\pm 6$ dB 以内的精度。

#### 6.6.3.6.8 LQI

LQI 度量是对接收数据包强度或质量的一个表征。这一度量可以由接收机 ED、信噪比估计或这些方法的组合来实现。有关网络层或应用层对于 LQI 结果的使用,本标准不作规定。

每一个接收数据包都应测量 LQI,测量的结果是一个 0x00 到 0xff 之间的整数,应使用 PD-DATA.indication 原语(见 6.3.2.4)报告给 MAC 层。LQI 的最小和最大值(0x00 和 0xff)应对应于接收机可检测的最低和最高质量的信号,之间的 LQI 值应在两个极限间均匀分布。LQI 至少应使用 8 个不同的值。

#### 6.6.3.6.9 CCA 方式

PHY 应至少提供下述三种 CCA 执行方法中的一种:

- a) CCA 方式 1:能量检测。当检测到能量超过 ED 门限时,CCA 应报告信道忙。
- b) CCA 方式 2:载波侦听。只有当检测到具有本标准调制和扩频特性的信号时,CCA 才报告信道忙。这个信号可以高于或低于 ED 门限。
- c) CCA 方式 3:载波侦听联合能量检测。只有检测到具有中国标准调制和扩频特性的信号并且其能量超过 ED 门限时,CCA 才报告信道忙。

不管是哪种 CCA 方式,如果 PHY 收到 PLME-CCA.request 原语(见 6.3.3.2)时正在接收一个 PPDU,CCA 应报告信道忙。检测到 SFD 后,PPDU 接收就被认为正在进行中,并且直到 PHR 规定的八位位组数被全部接收才告结束。

信道忙应使用状态为 BUSY 的 PLME-CCA.confirm 原语来指示。

信道空闲应使用状态为 IDLE 的 PLME-CCA.confirm 原语来指示。

PHYRIB 属性 phyCCAMode(见 6.5)应指示相应的工作方式。CCA 参数应遵循下述准则:

- a) ED 门限不能超过规定的接收灵敏度 10 dB;

b) CCA 的检测时间应等于 8 个符号周期。

## 7 MAC 层规范

### 7.1 MAC 层服务原语

#### 7.1.1 概述

本章定义了 RWSN 的 MAC 层规范。MAC 层提供高层访问物理无线信道的服务,并完成以下任务:

- a) 协调器负责产生并发送网络信标帧;
- b) 普通设备通过周期性的跟踪协调器广播的信标帧与协调器同步;
- c) 支持 RWSN 的关联和解关联操作;
- d) 利用 CAMA-CA 机制访问信道;
- e) 支持时隙保障(SCFP)机制;
- f) 支持不同设备的 MAC 层间可靠传输。

MAC 层提供了 SSCS 和 PHY 的接口。MAC 层包括了管理实体 MLME,提供服务接口,并维护和 MAC 层相关的管理物体的数据库,即 MAC 层信息数据库。图 17 是 MAC 层接口和组成框图。

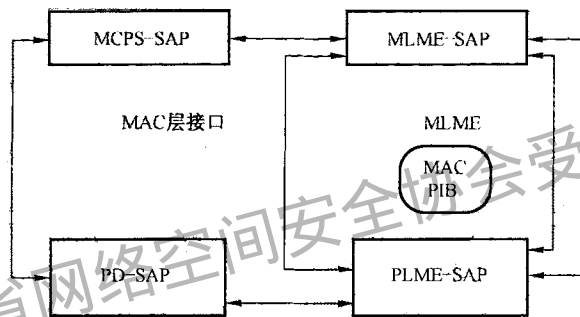


图 17 MAC 层参考模型

MAC 层分别通过两种服务接口(SAP)提供两种不同的服务:

- a) 通过 MCPS-SAP 提供 MAC 层数据服务,该服务保证 MAC 协议数据单元在 PHY 数据服务中的正确收发;
- b) 通过 MLME-SAP 提供 MAC 层管理服务,该服务维护一个存储 MAC 层协议状态相关信息的数据库。

这两种服务分别通过 PD-SAP 和 PLME-SAP 来支持 SSCS 和 PHY 的接口,除了以上的外部接口,MLME 和 MCPS 间还存在一个接口,这个接口使得 MLME 能够获得 MAC 数据服务。

#### 7.1.2 MAC 数据服务

##### 7.1.2.1 概述

MCPS-SAP 支持不同的 SSCS 间传输 SSCS 协议数据单元(SPDU)。表 25 列出了 MCPS-SAP 支持的原语,其中标有◆的可以由简化功能设备任意选择。

表 25 MCPS-SAP 原语

MCPS-SAP 原语	Request	Confirm	Indication
MCPS-DATA	7.1.2.2	7.1.2.3	7.1.2.4
MCPS-PURGE	7.1.2.5	7.1.2.6◆	—

7.1.2.2 MCPS-DATA.request

7.1.2.2.1 服务原语的语义

MCPS-DATA.request 原语请求把数据 SPDU(即 MSDU)从本地 SSCS 实体传送到单个对等 SSCS 实体。

MCPS-DATA.request 原语语义如下：

```
MCPS-DATA.request (
    SrcAddrMode,
    DstAddrMode,
    DstRWSNID,
    DstAddr,
    msduType,
    msduLength,
    msdu,
    msduHandle,
    TxOptions,
)
```

表 26 列出了该原语的参数。

表 26 原语 MCPS-DATA.request 的参数

名称	类型	有效范围	描述
<i>SrcAddrMode</i>	整型	0x00~0x03	源地址的方式,取值如下: 0x00=无地址(忽略地址); 0x01=保留; 0x02=16 比特短地址; 0x03=64 比特扩展地址
<i>DstAddrMode</i>	整型	0x00~0x03	目的地址的方式,取值如下: 0x00=无地址(地址字段忽略,见 7.2.2.2.7); 0x01=保留; 0x02=16 比特短地址; 0x03=64 比特扩展地址
<i>DstRWSNID</i>	整型	0x0000~0xffff	MSDU 接收的 16 比特 RWSN 标识符
<i>DstAddr</i>	设备地址	由参数 <i>DstAddrMode</i> 规定	MSDU 接收的个人设备地址
<i>msduType</i>	整型	0x00~0x03	数据子类型,取值如下: 0x00=非监测数据; 0x01=监测数据; 0x02=质疑数据; 0x03=更新数据
<i>msduLength</i>	整型	$\leq aMaxMACFrameSize$	MAC 层将要传送的 MSDU 中包含的八位位组数
<i>Msdu</i>	八位位组集	—	MAC 层将要传送的 MSDU 的八位位组集

表 26 (续)

名称	类型	有效范围	描述
<i>msduHandle</i>	整型	0x00~0xff	与 MAC 层将要传送的 MSDU 相关的操作
<i>TxOptions</i>	位图	4 比特字段	3 比特( $b_0, b_1, b_2$ )指示了 MSDU 的传输选项 $b_0, 1$ =带确认的传输, 0=无确认传输; $b_1, 1$ =SCFP 传输, 0=CAP 传输(有信标能力 RWSN); $b_2, 1$ =间接传输, 0=直接传输; $b_3, 1$ =带数据接受确认的传输, 0=无数据接受确认的传输

### 7.1.2.2.2 原语的使用

当有数据 SPDU(即 MSDU)传送到对等 SSCS 实体时,本地 SSCS 实体产生 MCPS-DATA.request 原语。

### 7.1.2.2.3 收后效果

当 MAC 层收到这个原语,就开始发送 MSDU。

MAC 层会根据原语提供的参数,构造并发送一个 MPDU。参数 *SrcAddrMode* 和 *DstAddrMode* 的标志位对应于帧控制字段(见 7.2.2.2)中的地址子字段,这些标志位用来组成 MHR 的帧控制和地址字段。如果参数 *SrcAddrMode* 和 *DstAddrMode* 都设为 0x00(即忽略地址字段),MAC 层将发布 MCPS-DATA.confirm 原语,状态为 INVALID\_ADDRESS。

参数 *msduType* 的信息指示了传输数据的子类型,分别对应 7.2.2.2 数据帧格式中的非监测型数据帧、检测行数据帧、质疑数据帧和更新数据帧。

如果参数 *msduLength* 大于 *aMaxMACSafePayloadSize*,MAC 层将帧控制信息的帧版本子字段设为 1。

参数 *TxOptions* 的信息指示了 MAC 层数据服务如何传输已提供的 MSDU。如果参数 *TxOptions* 规定采用带确认的传输,帧控制的确认请求子字段就设为 1(见 7.5.7.4)。

如果参数 *TxOptions* 规定在 SCFP 中传输,MAC 层应确定设备是否分配了一个有效的 SCFP。如果找不到一个有效的 SCFP,MAC 层就发布原语 MCPS-DATA.confirm,状态为 INVALID\_SCFP。如果存在一个有效的 SCFP,MAC 层就等到该 SCFP 开始时传送。如果参数 *TxOptions* 不要求在 SCFP 中传输,对于有信标能力网络 MAC 层将使用时隙 CSMA-CA 算法在 CAP 时段传输 MSDU。如果参数 *TxOptions* 要求在 SCFP 中传输就不考虑间接传输请求。

如果参数 *TxOptions* 要求间接传输,当协调器的 MAC 层接收到此原语,就间接发送数据帧,过程如下:把数据帧添加到协调器的微处理事务列表,在设备根据 7.5.7.3 所描述的算法进行判断后,就提取该数据并发送。当向一个广播地址发送数据时,采用 7.2.2.2.3 的算法。当向单播地址发送时,相关设备根据 7.5.7.3 所描述的算法进行判断后才能提取数据发送。如果没有足够的空间来存储这些数据,MAC 层将丢弃这个 MSDU,并发布原语 MCPS-DATA.confirm,状态 TRANSACTION\_OVERFLOW。而如果有足够的空间,协调器将把相应信息加入列表。如果在 *macTransactionPersistenceTime* 内没有处理完该事务,MAC 层将丢弃事务信息,并发布 MCPS-DATA.confirm 原语,状态 TRANSACTION\_EXPIRED。事务处理过程如 7.5.6 所述。如果参数 *TxOptions* 要求间接传输,当原语接收设备不是协调器且目的地址不存在,或者参数 *TxOptions* 还要求在 SCFP 内传输,就忽略间接传输选项。

如果参数 *TxOptions* 不要求间接传输,在有信标能力 RWSN 中,MAC 层将使用 CSMA-CA 算法在 CAP 时段传输 MSDU。如果参数 *TxOptions* 规定直接传输且 MAC 层在 *macMaxFrameRetries* 的重传时间内没有收到确认,就丢弃 MSDU,并发布 MCPS-DATA.confirm 原语,状态是 NO\_ACK。

如果参数 *msduType* 的信息指示了传输数据的子类型为监测数据,则参数 *TxOptions* 规定采用带数据接受确认的传输,且 MAC 层需收到数据接受确认帧才判定成功发送了 MSDU。如果参数 *msduType* 的信息指示了传输数据的子类型不为监测数据,则参数 *TxOptions* 规定采用无数据接受确认的传输,且 MAC 层只需普通确认帧即判定成功发送了 MSDU。

如果请求的事务太大而不能在 CAP 或 SCFP 内完成,MAC 层会丢弃该帧,并发布原语 MCPS-DATA.confirm,状态为 FRAME\_TOO\_LONG。

如果整个传输使用 CSMA-CA 算法,当算法竞争信道失败,且参数 *TxOptions* 要求直接传输,MAC 层会丢弃该 MSDU,并发布 MCPS-DATA.confirm 原语,状态为 CHANNEL\_ACCESS\_FAILURE。

如果成功发送了 MPDU,且参数 *TxOptions* 规定带确认的传输,MAC 层就发布原语 MCPS-DATA.confirm,状态是 SUCCESS。

如果原语 MCPS-DATA.request 的所有参数设置都超出有效范围,或是不能被支持,MAC 层就发布原语 MCPS-DATA.confirm,状态为 INVALID\_PARAMETER。

### 7.1.2.3 MCPS-DATA.confirm

#### 7.1.2.3.1 服务原语的语义

原语 MCPS-DATA.confirm 报告了本地 SSCS 请求传送 SPDU(MSDU)给另一个 SSCS 的结果。

原语 MCPS-DATA.confirm 语义如下:

```
MCPS-DATA.confirm (
    msduHandle,
    status,
    Timestamp,
)
```

表 27 规定了原语 MCPS-DATA.confirm 的参数。

表 27 MCPS-DATA.confirm 的参数

名称	类型	有效范围	描述
<i>msduHandle</i>	整型	00x00~0xff	待确认 MSDU 的相关操作
<i>status</i>	枚举	SSUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, INVALID_ADDRESS, INVALID_SCFP,NO_ACK, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY, INVALID_PARAMETER	最后一个 MSDU 的传送状态
<i>Timestamp</i>	整型	0x000000~0xfffff	可选项。数据传输的时刻(以符号为单位,见 7.5.5.2)。只有当状态是 SUCCESS 时,该项数值才认为有效;如果状态不是 SUCCESS,该参数将没有任何用处。符号的边缘由 <i>mac-SyncSymbolOffset</i> (见 7.4.1 中表 70)规定。该参数是 24 比特,精确度最少为 20 比特,剩下的 4 位是最低位



### 7.1.2.3.2 产生条件

MAC 层实体响应 MCPS-DATA.request 原语时,会产生 MCPS-DATA.confirm 原语。MCPS-DATA.confirm 原语返回的状态参数可以是 SUCCESS,表示数据发送请求已成功执行,也可以是错误码。状态的值得在 7.1.2.2.3 有详细描述。

### 7.1.2.3.3 原语的使用

收到 MCPS-DATA.confirm 原语后,设备的 SSCS 获知了数据发送请求的结果。如果发送成功,状态参数设为 SUCCESS,否则,状态参数会指示错误。

## 7.1.2.4 MCPS-DATA.indication

### 7.1.2.4.1 服务原语的语义

原语 MCPS-DATA.indication 指示了数据 SPDU(即 MPDU)从 MAC 层到本地 SSCS 实体的发送状况。

原语 MCPS-DATA.indication 语义如下:

```
MCPS-DATA.indication (
    SrcAddrMode,
    SrcRWSNId,
    SrcAddr,
    DstAddrMode,
    DstRWSNID,
    DstAddr,
    msduType,
    msduLength,
    msdu,
    mpduLinkQuality,
    DSN,
    Timestamp,
)
```

表 28 规定了 MCPS-DATA.indication 原语的参数。

### 7.1.2.4.2 产生条件

当本地 MAC 层实体收到数据帧,经过相应的消息过滤过程(见 7.5.7.2),本地 MAC 层实体会向 SSCS 发布原语 MCPS-DATA.indication。

### 7.1.2.4.3 原语的使用

收到原语 MCPS-DATA.indication,SSCS 就获知了数据已传输到目的设备。如果 MAC 层处于混合方式(见 7.5.7.5),收到原语后参数的设置见 7.5.7.4。

表 28 MCPS-DATA.indication 的参数

名称	类型	有效范围	描述
<i>SrcAddrMode</i>	整型	0x00~0x03	接收到的 MPDU 的源地址方式,取值如下: 0x00=无地址(忽略地址字段); 0x01=保留; 0x02=16 比特短地址; 0x03=64 比特扩展地址
<i>SrcRWSNId</i>	整型	0x0000~0xffff	MSDU 发送设备的 16 比特 RWSN 标识符
<i>SrcAddr</i>	设备地址	由参数 <i>SrcAddrMode</i> 规定	MSDU 发送设备的个人设备地址
<i>DstAddrMode</i>	整型	0x00~0x03	MPDU 的目的地址方式,取值如下: 0x00=无地址(忽略地址字段); 0x01=保留; 0x02=16 比特短地址; 0x03=64 比特扩展地址
<i>DstRWSNID</i>	整型	0x0000~0xffff	MSDU 接收设备的 16 比特 RWSN 标识符
<i>DstAddr</i>	设备地址	由参数 <i>DstAddrMode</i> 规定	MSDU 接收设备的个人设备地址
<i>msduType</i>	整型	0x00~0x03	数据类型,取值如下: 0x00=非监测数据; 0x01=监测数据; 0x02=质疑数据; 0x03=更新数据
<i>msduLength</i>	整型	≤ aMaxMACFrameSize	MAC 层收到 MSDU 中包含的八位位组数
<i>MsdU</i>	八位位组集	—	MAC 层收到 MSDU 中的八位位组集
<i>mpduLinkQuality</i>	整型	0x00~0xff	在接收 MPDU 过程中评估 LQI。值越低表示 LQI 越小(见 6.6.3.6.8)
<i>DSN</i>	整型	0x00~0xff	接收数据帧的数据序列号。
<i>Timestamp</i>	整型	0x000000~0xffffff	可选项。数据传输的时刻(以符号为单位,见 7.5.5.2)。符号的边缘由 <i>macSyncSymbolOffset</i> (见表 71) 来描述。 该参数是 24 比特,精确度最少为 20 比特,最低 4 位是最低位

7.1.2.5 MCPS-PURGE.request

7.1.2.5.1 服务原语的语义

原语 MCPS-PURGE.request 允许邻近高层从事务队列里清除一个 MSDU。该原语对 RFD 是可选的。

原语 MCPS-PURGE.request 语义如下：

```
MCPS-PURGE.request (
    msduHandle
)
```

表 29 规定了原语 MCPS-PURGE.request 的参数。

表 29 MCPS-PURGE.request 的参数

名称	类型	有效范围	描述
msduHandle	整型	0x00~0xff	从事务队列里清除 MSDU 的操作

#### 7.1.2.5.2 原语的使用

当要从事务队列清除 MSDU 时,邻近高层就发布 MCPS-PURGE.request 原语。

#### 7.1.2.5.3 收后效果

收到 MCPS-PURGE.request 原语后,MAC 层就在事务队列中查找原语中 msduHandle 标识 MSDU。如果该 MSDU 已离开队列,就不会进行相关操作。如果找到了相匹配的 MSDU,MAC 层将该 MSDU 从事务队列中删除,并发布 MCPSPURGE.confirm 原语,状态为 SUCCESS。如果找不到匹配的 MSDU,MAC 层就发布 MCPSPURGE.confirm 原语,状态为 INVALID\_HANDLE。

#### 7.1.2.6 MCPS-PURGE.confirm

##### 7.1.2.6.1 服务原语的语义

MAC 层发布原语 MCPS-PURGE.confirm 向邻近高层报告请求从事物队列中删除 MSDU 的结果。该原语对于 RFD 是可选的。

原语 MCPS-PURGE.confirm 语义如下：

```
MCPS-PURGE.confirm (
    msduHandle
    status
)
```

表 30 规定了原语 MCPS-PURGE.confirm 的参数。

表 30 MCPS-PURGE.confirm 的参数

名称	类型	有效范围	描述
msduHandle	整型	0x00~0xff	请求从事务队列里清除 MSDU 的操作
status	枚举	SUCCESS 或 INVALID_HANDLE	请求从事务队列清除 MSDU 的状态

##### 7.1.2.6.2 产生条件

为响应原语 MCPS-PURGE.request,MAC 层就会发布原语 MCPS-PURGE.confirm。原语 MCPS-PURGE.confirm 返回的状态可以是 SUCCESS,表示成功执行清除请求,也可以 INVALID\_HANDLE,表示发生了错误。状态的值在 7.1.2.5.3 有详细描述。

7.1.2.6.3 原语的使用

收到原语 MCPS-PURGE.confirm 后,邻近高层就获知了请求从事务队列清除 MSDU 的结果。如果清除请求成功,状态参数就设为 SUCCESS,否则状态的标识有错误。

7.1.2.7 数据服务消息序列图

图 18 是两个设备间数据成功传输的序列示意图。图 88 和图 89(见 7.6)同样是消息序列图,其中包括 PHY 需要进行的操作。

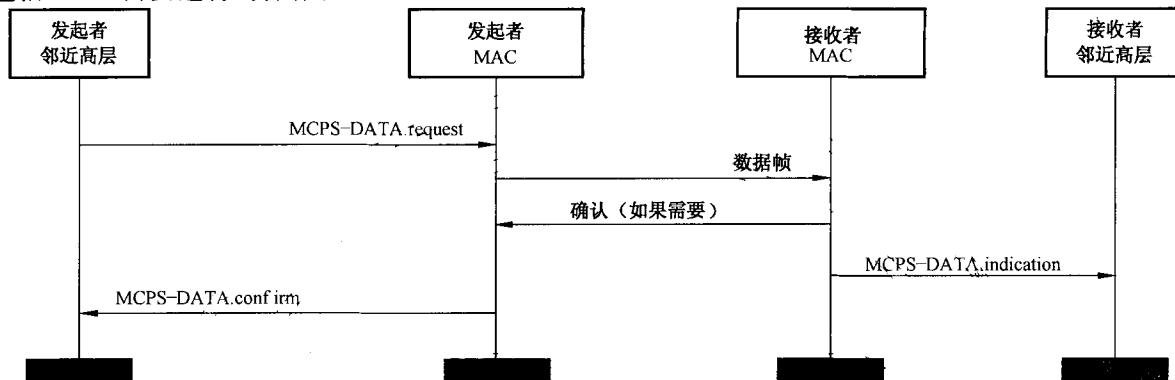


图 18 MAC 数据服务的消息序列图

7.1.3 MAC 管理服务

MLME-SAP 允许在 MLME 和邻近高层之间传输管理命令。表 31 中总结了 MLME 支持的通过 MLME-SAP 接口传输的原语。其中有(♦)标记的原语对 RFD 是可选的,而标有(\*)则是对 FFD 和 RFD 都是可选的。

表 31 通过 MCPS-SAP 接口传输的原语

名称	Request	Indication	Response	Confirm
MLME-ASSOCIATE	7.1.4.2 ♦	7.1.4.3 ♦	7.1.4.4 ♦	7.1.4.5 ♦
MLME-DISASSOCIATE	7.1.5.2 ♦	7.1.5.3		7.1.5.4
MLME-BEACON-NOTIFY		7.1.6.2		
MLME-GET	7.1.7.2			7.1.7.3
MLME-SCFP	7.1.8.2 *	7.1.8.4 *		7.1.8.3 *
MLME-RESET	7.1.9.2			7.1.9.3
MLME-RX-ENABLE	7.1.10.2 *			7.1.10.3 *
MLME-SCAN	7.1.11.2			7.1.11.3
MLME-COMM-STATUS		7.1.12.2		
MLME-SET	7.1.13.2			7.1.13.3
MLME-START	7.1.14.2 ♦			7.1.14.3 ♦
MLME-SYNC	7.1.15.2 *			
MLME-SYNC-LOSS		7.1.15.3		
MLME-POLL	7.1.16.2			7.1.16.3

## 7.1.4 关联原语

### 7.1.4.1 概述

MLME-SAP 关联原语定义设备如何与 RWSN 建立连接。

所有设备为请求和确认关联的原语提供接口。指示和响应关联原语对 RFD 是可选的。

### 7.1.4.2 MLME-ASSOCIATE.request

#### 7.1.4.2.1 服务原语的语义

原语 MLME-ASSOCIATE.request 允许设备请求与协调器关联。

原语 MLME-ASSOCIATE.request 语义如下：

```
MLME-ASSOCIATE.request (
    LogicalChannel,
    ChannelPage,
    CoordAddrMode,
    CoordRWSNId,
    CoordAddress,
    CapabilityInformation,
)
```

表 32 规定了原语 MLME-ASSOCIATE.request 的参数。

表 32 MLME-ASSOCIATE.request 的参数

名称	类型	有效范围	描述
LogicalChannel	整型	选自 PHY 支持的可用逻辑信道(见 6.2.2)	试图在此信道进行关联
ChannelPage	整型	选自 PHY 支持的可用逻辑信道页(见 6.2.2)	试图在此信道页进行关联
CoordAddrMode	整型	0x02~0x03	该原语和后续 MPDU 的协调器寻址方式。取值如下： 0x02 表示 16 比特短地址； 0x03 表示 64 比特扩展地址
CoordRWSNId	整型	0x0000~0xffff	所要关联的 RWSN 的标识符
CoordAddress	设备地址	由参数 CoordAddrMode 规定	所要关联的协调器的地址
CapabilityInformation	位图	见 7.3.2.3	规定了提出关联请求设备的运作能力

#### 7.1.4.2.2 原语的使用

未关联设备的邻近高层向 MLME 发布原语 MLME-ASSOCIATE.request, 请求通过协调器与 RWSN 关联。在有信标能力 RWSN 中, 如果设备希望通过协调器进行关联, MLME 需要在发布该原语之前跟踪协调器的信标, 这是可选的。

## 7.1.4.2.3 收后效果

收到 MLME-ASSOCIATE.request 原语,未关联设备的 MLME 首先更新 PHY 和 MAC RIB 的相关属性,然后产生关联请求命令,具体过程见 7.5.4.1 的关联流程。

如果由于 CSMA-CA 算法竞争信道失败,即信道忙时,关联请求命令无法发送给协调器,MLME 将发布 MLME-ASSOCIATE.confirm 原语,状态为 CHANNEL\_ACCESS\_FAILURE。

如果 MLME 成功发送关联请求命令,MLME 将希望收到对方发返回的确认。如果没有收到确认,MLME 就发布 MLME-ASSOCIATE.confirm 原语,状态为 NO\_ACK。

如果未关联设备的 MLME 成功接收到关联请求命令的确认,MLME 将等待请求的响应(见 7.5.4.1)。如果没有收到相应,MLME 就发布原语 MLME-ASSOCIATE.confirm,状态为 NO\_DATA。

如果设备的 MLME 得到了来自协调器的连接响应命令帧,将发布原语 MLME-ASSOCIATE.confirm,状态和关联相应命令(见 7.3.4)中关联状态字段的设置相同。

当收到关联请求命令,协调器的 MLME 将发布 MLMEASSOCIATE.indication 原语。

如果原语 MLME-ASSOCIATE.request 中任何参数的设置不被支持或者超出有效范围,MLME 将发布 MLME-ASSOCIATE.confirm 原语,状态为 INVALID\_PARAMETER。

## 7.1.4.3 MLME-ASSOCIATE.indication

## 7.1.4.3.1 服务原语的语义

原语 MLME-ASSOCIATE.indication 指示收到关联请求命令。

原语 MLME-ASSOCIATE.indication 语义如下:

MLME-ASSOCIATE.indication (

DeviceAddress,

CapabilityInformation,

)

表 33 规定了原语 MLME-ASSOCIATE.indication 的参数。

表 33 MLME-ASSOCIATE.indication 的参数

名称	类型	有效范围	描述
DeviceAddress	设备地址	扩展的 64 比特地址	关联请求设备的地址
CapabilityInformation	位图	见 7.3.2.3	连接关联设备的运作能力

## 7.1.4.3.2 产生条件

协调器的 MLME 发布原语 MLME-ASSOCIATE.indication,向邻近高层显示已接收到关联请求命令(见 7.3.2)。

## 7.1.4.3.3 原语的使用

当协调器的邻近高层接收到 MLME-ASSOCIATE.indication 原语,协调器决定是否接受这个未关联设备,相关的算法超出了本规范的范围。然后协调器的邻近高层向 MLME 发 MLME-ASSOCIATE.response。

是否关联的决定和响应在时间 macResponseWaitTime 内是有效的,可用的(见 7.5.4.1)。超过这

段时间,提出关联请求的设备就向协调器提取关联响应命令帧,来判断关联是否成功,使用的方法如 7.5.7.3 所描述。

#### 7.1.4.4 MLME-ASSOCIATE.response

##### 7.1.4.4.1 服务原语的语义

原语 MLME-ASSOCIATE.response 用于回应原语 MLME-ASSOCIATE.indication。

原语 MLME-ASSOCIATE.response 语义如下:

```
MLME-ASSOCIATE.response (
    DeviceAddress,
    AssocShortAddress,
    status,
)
```

表 34 描述了原语 MLME-ASSOCIATE.response 的参数。

表 34 MLME-ASSOCIATE.response 的参数

名称	类型	有效范围	描述
DeviceAddress	设备地址	扩展的 64 比特地址	关联请求设备的地址
AssocShortAddress	整型	0x0000~0xffff	成功关联后协调器分配的 16 比特短地址。当关联失败,该参数设为 0xffff
status	枚举	见 7.3.2.3	关联的状态

##### 7.1.4.4.2 原语的使用

协调器的邻近高层向 MLME 发布 MLME-ASSOCIATE.response 原语,以响应原语 MLME-ASSOCIATE.indication。

##### 7.1.4.4.3 收后效果

当收到 MLME-ASSOCIATE.response 原语,协调器的 MLME 产生关联响应命令(见 7.3.3),并发送给关联请求设备,具体过程为:把命令帧添加到协调器的待处理事务列表中,请求设备做出判断后就提取该命令,相关的方法见 7.5.7.3。

收到 MLME-ASSOCIATE.response 原语后,协调器就尝试将该原语中的信息添加到待处理事务列表中。如果没有足够空间存储事务,MAC 层丢弃该帧,并发布原语 MLME-COMM-STATUS.indication,状态为 TRANSACTION\_OVERFLOW。如果有足够空间,协调器就把信息加到列表中。如果在 macTransactionPersistenceTime 内没有处理这个事务,就丢弃该事务信息,MAC 层将发布 MLME-COMM-STATUS.indication 原语,状态为 TRANSACTION\_EXPIRED。事务处理过程如 7.5.6 所描述。

如果帧传输成功,并且在要求的情况下收到确认,MAC 层发布原语 MLME-COMM-STATUS.indication,状态为 SUCCESS。

如果原语 MLME-ASSOCIATE.response 中任何参数的设置不被支持或者超出有效范围,MAC 层将发布 MLME-COMM-STATUS.indication 原语,状态为 INVALID\_PARAMETER。

7.1.4.5 MLME-ASSOCIATE.confirm

7.1.4.5.1 服务原语的语义

原语 MLME-ASSOCIATE.confirm 用于向邻近高层报告关联请求是否成功。

原语 MLME-ASSOCIATE.confirm 语义如下：

```
MLME-ASSOCIATE.confirm (
    AssocShortAddress,
    status,
)
```

表 35 描述了原语 MLME-ASSOCIATE.confirm 的参数。

表 35 MLME-ASSOCIATE.confirm 的参数

名称	类型	有效范围	描述
AssocShortAddress	整型	0x0000~0xffff	关联成功后,协调器分配给设备的短地址。当关联失败时该参数为 0xffff
status	枚举	关联相应命令中 status 字段的值(见 7.3.4), SUCCESS, CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, COUNTER_ERROR, FRAME_TOO_LONG, UNSUPPORTED_LEGACY, INVALID_PARAMETER	关联请求的状态

7.1.4.5.2 产生条件

为相应 MLME-ASSOCIATE.request 原语,MLME 向邻近高层发布原语 MLME-ASSOCIATE.confirm。如果请求成功,关联相应命令的参数 status 指示关联成功。否则,参数 status 指示关联响应命令中的错误码,或者表 35 中的错误码。

7.1.4.5.3 原语的使用

接收到 MLME-ASSOCIATE.confirm 原语之后,发起设备的邻近高层就获知了其于协调器关联请求的结果。如果关联成功,参数 status 将指示关联成功,与关联响应命令的关联状态字段一致,并且发起设备将获得一个 16 比特短地址(见表 70)。如果关联请求失败,地址将为 0xffff,并且参数 status 将指示相应的错误。

7.1.4.6 关联消息序列图

图 19 描述了不跟踪协调器信标的设备用于成功与协调器关联的消息序列。图 84 和图 85 描述了类似的场景。

7.1.5 解关联原语

7.1.5.1 概述

MLME-SAP 解关联原语定义设备的解关联。所有的设备应提供解关联原语的接口。



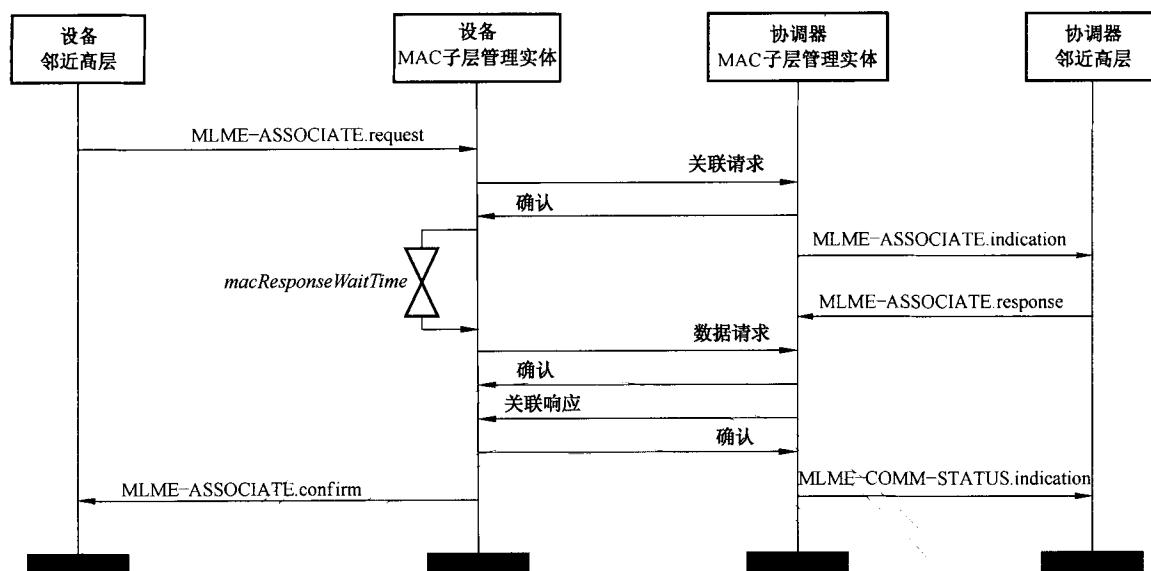


图 19 关联消息序列图

### 7.1.5.2 MLME-DISASSOCIATE.request

#### 7.1.5.2.1 服务原语语义

原语 MLME-DISASSOCIATE.request 用于已关联设备要求离开 RWSN。也可以用于协调器要求关联设备离开 RWSN。

原语 MLME-DISASSOCIATE.request 语义如下：

MLME-DISASSOCIATE.request (

- DeviceAddrMode,
- DeviceRWSNId,
- DeviceAddress,
- DisassociateReason,
- TxIndirect,

)

表 36 描述了原语 MLME-DISASSOCIATE.request 的参数。

表 36 MLME-DISASSOCIATE.request 的参数

名称	类型	有效范围	描述
DeviceAddrMode	整型	0x02~0x03	解关联通知命令接收设备的地址方式
DeviceRWSNId	整型	0x0000~0xffff	解关联通知命令接收设备的 RWSN 标识符
DeviceAddress	设备地址	由参数 Device-AddrMode 规定。	解关联通知命令接收设备的地址
DisassociateReason	整型	0x00~0xff	解关联的原因 (见 7.3.4.3)
TxIndirect	布尔型	TRUE 或 FALSE	如果解关联通知命令采用间接发送方式,该参数设为 TRUE

#### 7.1.5.2.2 原语使用

原语 MLME-DISASSOCIATE.request 由已关联设备的邻近高层产生并发布给其 MLME,要求从

RWSN 中解关联。也可由协调器的邻近高层产生并发布给其 MLME,要求已关联设备离开 RWSN。

### 7.1.5.2.3 收后效果

接收到 MLME-DISASSOCIATE.request 原语后,MLME 对参数 DeviceRWSNId 和 macRWSNId 进行比较。如果两者不等,MLME 发布原语 MLME-DISASSOCIATE.confirm,其状态参数为 INVALID\_PARAMETER。如果相等,MLME 检查原语的地址字段。

如果参数 DeviceAddrMode 等于 0x02,且参数 DeviceAddress 与 macCoordShortAddress 相等,或者参数 DeviceAddrMode 等于 0x03,且 DeviceAddress 与 macCoordExtendedAddress 相等,忽略参数 TxIndirect,且 MLME 在 CAP 时段向其协调器发送解关联通知命令(见 7.1.5.2)。

如果参数 DeviceAddrMode 等于 0x02,且参数 DeviceAddress 与 macCoordShortAddress 不相等,或者 DeviceAddrMode 参数等于 0x03,且参数 DeviceAddress 与 macCoordExtendedAddress 不相等,并且协调器的 MLME 接收到的该原语的参数 TxIndirect 为 TRUE,那么解关联通知命令将采用间接传输的方式发送,即命令帧被添加到协调器的待处理事务列表中,相应的设备根据 7.5.7.3 的方法从协调器中提取命令帧。

如果参数 DeviceAddrMode 等于 0x02,且参数 DeviceAddress 与 macCoordShortAddress 不相等,或者参数 DeviceAddrMode 等于 0x03,且参数 DeviceAddress 与 macCoordExtendedAddress 不相等,并且协调器的 MLME 接收到的该原语的参数 TxIndirect 为 FALSE,那么 MLME 将在 CAP 时段内发送解关联通知命令。

除以上情况,MLME 发布原语 MLME-DISASSOCIATE.confirm,其状态为 INVALID\_PARAMETER,且不产生解关联通知命令。

如果解关联通知命令采用间接传输的方式发送,且没有足够的空间存储该事务,那么 MLME 将丢弃该帧,并发布原语 MLME-DISASSOCIATE.confirm,其状态为 TRANSACTION\_OVERFLOW。而如果有足够空间存储该事务,协调器将把该事务的相关信息添加到列表中。如果该事务在 macTransactionPersistenceTime 时间内没有被处理,该事务相关信息将被丢弃,且 MLME 将发布原语 MLME-DISASSOCIATE.confirm,其状态为 TRANSACTION\_EXPIRED。事务处理过程见 7.5.6。

如果解关联通知命令的发送失败是由于 CSMA-CA 算法失败引起的,且该原语是由协调器的 MLME 接收的(原语参数 TxIndirect 为 FALSE),或是由设备的 MLME 接收的,则 MLME 将发布原语 MLME-DISASSOCIATE.confirm,其状态为 CHANNEL\_ACCESS\_FAILURE。

如果 MLME 成功发送了解关联通知命令,MLME 等待一个回复确认。如果没有收到回复确认,且是协调器的 MLME 接收的该原语(参数 TxIndirect 被设为 FALSE),或者是由设备的 MLME 接收的该原语,MLME 就发布 MLME-DISASSOCIATE.confirm 原语,其状态为 NO\_ACK。

如果 MLME 成功发送了解关联通知命令,并且收到了回复确认,MLME 将发布原语 MLME-DISASSOCIATE.confirm,其状态为 SUCCESS。

接收到解关联通知命令后,接收者的 MLME 将发布 MLME-DISASSOCIATE.indication 原语。

如果原语 MLME-DISASSOCIATE.request 任何参数不被支持或超出有效范围,MLME 就发布 MLME-DISASSOCIATE.Confirm,其状态为 INVALID\_PARAMETER。

### 7.1.5.3 MLME-DISASSOCIATE.indication

#### 7.1.5.3.1 服务原语语义

原语 MLME-DISASSOCIATE.indication 指示接收到一个解关联通知命令。

原语 MLME-DISASSOCIATE.indication 的语义如下:

MLME-DISASSOCIATE.indication (

DeviceAddress,  
DisassociateReason,  
)

表 37 描述了原语 MLME-DISASSOCIATE.indication 的参数。

表 37 MLME-DISASSOCIATE.indication 的参数

名称	类型	有效范围	描述
DeviceAddress	设备地址	扩展的 64 比特 IEEE 地址	请求解关联的设备的地址
DisassociateReason	整型	0x00~0xff	解关联的原因 (见 7.3.4.4)

#### 7.1.5.3.2 产生条件

接收到解关联命令后, MLME 产生并向其邻近高层发布 MLME-DISASSOCIATE.indication 原语。

#### 7.1.5.3.3 原语使用

该原语告知邻近高层解关联的原因。

#### 7.1.5.4 MLME-DISASSOCIATE.confirm

##### 7.1.5.4.1 服务原语语义

原语 MLME-DISASSOCIATE.confirm 报告原语 MLME-DISASSOCIATE.request 的结果。

原语 MLME-DISASSOCIATE.confirm 的语义如下:

MLME-DISASSOCIATE.confirm (

status,

DeviceAddrMode,

DeviceRWSNId,

DeviceAddress

)

表 38 描述了原语 MLME-DISASSOCIATE.confirm 的参数。

表 38 MLME-DISASSOCIATE.confirm 的参数

名称	类型	有效范围	描述
status	枚举	SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, NO_ACK, CHANNEL_ACCESS_FAILURE, COUNTER_ERROR, FRAME_TOO_LONG, INVALID_PARAMETER	解关联的状态

表 38 (续)

名称	类型	有效范围	描述
DeviceAddrMode	整型	0x02~0x03	请求解关联或被协调器通知解关联的设备的地址方式
DeviceRWSNId	整型	0x0000~0xffff	请求解关联或被协调器通知解关联的设备的 RWSN 标识符
DeviceAddress	设备地址	由参数 DeviceAddrMode 规定	请求解关联或被协调器通知解关联的设备的地址

7.1.5.4.2 产生条件

原语 MLME-DISASSOCIATE.confirm 由 MLME 产生并向其邻近高层发布,用于响应原语 MLME-DISASSOCIATE.request。该原语返回状态 SUCCESS,表示解关联请求成功,或者相应的错误码。状态的值完整定义见 7.1.5.2.3。

7.1.5.4.3 原语使用

接收到 MLME-DISASSOCIATE.confirm 原语后,发起设备的邻近高层就获知了解关联请求的结果。如果解关联成功,参数 status 设为 SUCCESS,否则状态参数指示相应错误。

7.1.5.5 解关联消息序列图

解关联请求可以由设备发起,也可以由设备关联的协调器发起。图 20 描述了设备成功从 RWSN 解关联的消息序列。

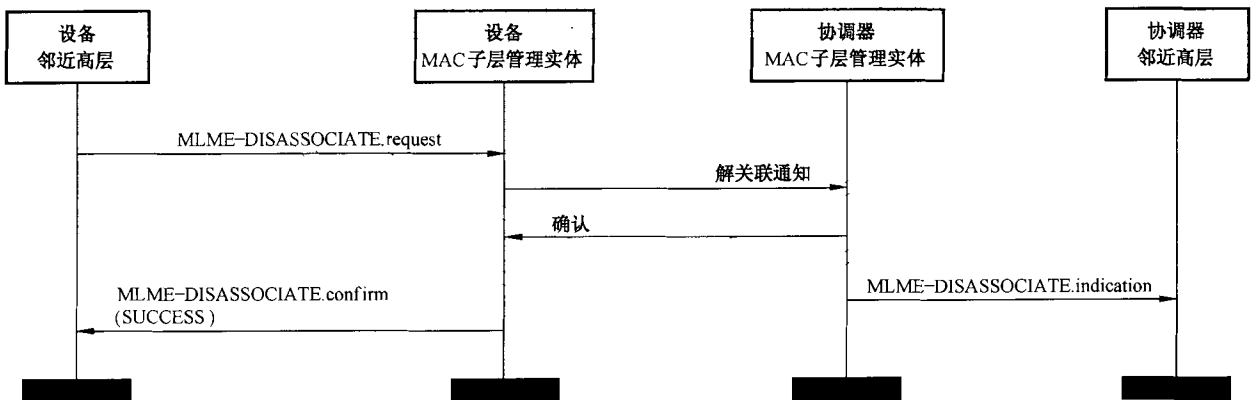


图 20 设备发起的解关联消息序列图

图 21 描述了信标使能的 RWSN 中的协调器利用间接传输方式成功将设备从 RWSN 中解关联的消息序列。

7.1.6 信标通知原语

7.1.6.1 概述

MLME-SAP 信标通知原语,定义在正常工作状态下收到信标帧时如何通知设备。

所有设备都要为信标帧通知原语提供合适的接口。

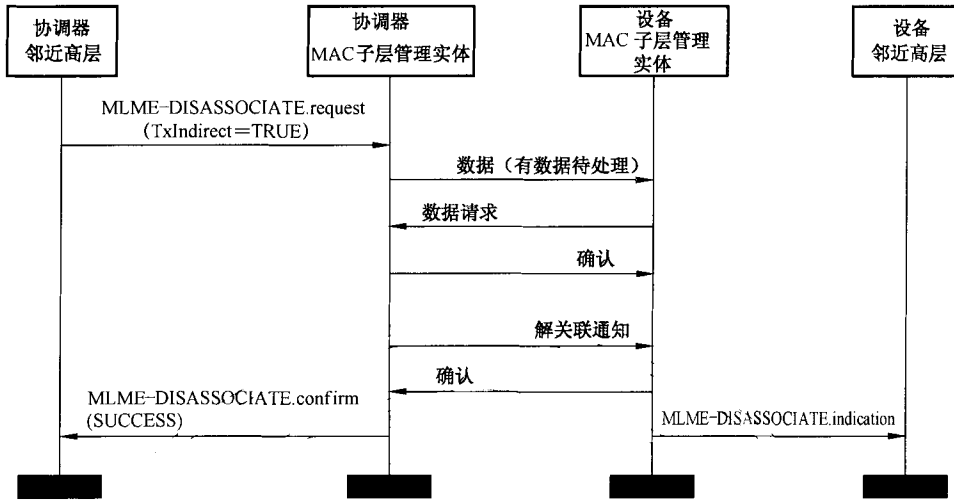


图 21 信标使能的 RWSN 中的协调器发起的利用间接传输方式的解关联的消息序列图

### 7.1.6.2 MLME-BEACON-NOTIFY.indication

#### 7.1.6.2.1 服务原语语义

原语 MLME-BEACON-NOTIFY.indication 将包含在 MAC 层接收的信标帧内的相关参数发送到邻近高层。该原语也用于发送 LQI 的测量值和信标接收时间。

原语 MLME-BEACON-NOTIFY.indication 语义如下：

MLME-BEACON-NOTIFY.indication (

- BSN,
- RWSNDescriptor,
- PendAddrSpec,
- AddrList,
- sduLength,
- sdu
- WorkingPeriodSpec,
- WorkingPeriodList,

)

表 39 描述了原语 MLME-BEACON-NOTIFY.indication 的参数。

表 39 MLME-BEACON-NOTIFY.indication 的参数

名称	类型	有效范围	描述
BSN	整型	0x00~0xff	信标帧序列号
RWSNDescriptor	RWSNDescriptor 值	见表 40	接收到信标的 RWSNDescriptor
PendAddrSpec	位图	见 7.2.3.1.6	信标待处理地址规范
AddrList	设备地址列表	—	协调器有其数据的设备地址列表
sduLength	整型	0-aMaxBeaconPayload- Length	MAC 层接收到的信标帧的净荷中所包含的八位位组数

表 39 (续)

名称	类型	有效范围	描述
sdu	八位位组集	—	MAC 层传送给其邻近高层的包含信标净荷的八位位组集
WorkingPeriodSpec	位图	见 7.2.3.1.9	周期分配规范
WorkingPeriodList	位图	见 7.2.3.1.10	周期分配列表

## 7.1.6.2.2 产生条件

原语 MLME-BEACON-NOTIFY.indication 由 MLME 在接收到信标帧,且 macAutoRequest 属性为 FALSE,或者接收到的信标帧的净荷大于等于一个八位位组时,产生并发布给其邻近高层。

## 7.1.6.2.3 原语使用

接收到 MLME-BEACON-NOTIFY.indication 原语,邻近高层就获知 MAC 层有信标帧到达。表 40 列出了 RWSNDescriptor 的组成元素。

表 40 RWSNDescriptor 的组成元素

名称	类型	有效范围	描述
CoordAddrMode	整型	0x02~0x03	发送信标帧的协调器的地址方式。取如下值: 0x02 表示 16 比特短地址。 0x03 表示 64 比特扩展地址
CoordRWSNId	整型	0x0000~0xffff	接收到的信标帧指示的协调器的 RWSN 标识
CoordAddress	设备地址	由参数 CoordAddr-Mode 规定	接收到的信标帧指示的协调器地址
LogicalChannel	整型	选自 PHY 支持的可用逻辑信道(见 6.2.2)	当前网络占用的逻辑信道
ChannelPage	整型	选自 PHY 支持的可用逻辑信道页(见 6.2.2)	当前网络占用的逻辑信道页
SuperframeSpec	位图	见 7.2.3.1.3	接收到的信标帧指示的超帧规范
SCFPPermit	布尔型	TRUE 或 FALSE	如果信标是由当前可以接受 SCFP 请求的 RWSN 协调器发出的,该参数为 TRUE
LinkQuality	整型	0x00~0xff	网络信标接收的 LQI 值。该值越低表示 LQI 的值越低(见 6.6.3.6.8)
TimeStamp	整型	0x000000~0xfffff	用符号表示的信标接收时间。该值等于信标接收时的时间戳,见 7.5.5.2 该值为 24 比特,其精度最小为 20 比特,且最低 4 位是最低有效位

## 7.1.7 RIB 属性读取原语

### 7.1.7.1 概述

MLME-SAP 的获取原语定义如何读取 RIB 属性。  
所有设备应为 RIB 属性获取原语提供接口。

### 7.1.7.2 MLME-GET.request

#### 7.1.7.2.1 服务原语语义

原语 MLME-GET.request 用于请求指定 RIB 属性的信息。

原语 MLME-GET.request 的语义如下：

```
MLME-GET.request    (
                    RIBAttribute,
                    RIBAttributeIndex
                    )
```

表 41 描述了原语 MLME-GET.request 的参数。

表 41 MLME-GET.request 的参数

名称	类型	有效范围	描述
RIBAttribute	整型	见表 71	特定 RIB 属性的标识
RIBAttributeIndex	整型	属性定义, 见表 71	特定 RIB 属性的表的索引。该参数只对 MAC RIB 属性中的表格属性有效。当访问 PHYRIB 属性时, 忽略该参数

#### 7.1.7.2.2 原语使用

原语 MLME-GET.request 由邻近高层产生并发布给其 MLME, 用以从 RIB 中获取信息。

#### 7.1.7.2.3 收后效果

接收到 MLME-GET.request 原语后, MLME 检查并判断要读取的 RIB 属性是 MAC RIB 属性还是 PHY RIB 属性。如果所请求的属性是 MAC 属性, MLME 将从其数据库中取得该 MAC RIB 属性值。如果在数据库中找到该 RIB 属性的标识, MLME 将发布 MLME-GET.confirm 原语, 其状态为 UNSUPPORTED\_ATTRIBUTE。如果参数 RIBAttributeIndex 指示的表格索引超出有效取值范围, MLME 将发布 MLME-GET.confirm 原语, 其状态为 INVALID\_INDEX。如果成功获取请求的 MAC RIB 属性, MLME 将发布 MLME-GET.confirm 原语, 其状态为 SUCCESS。

如果所请求的属性是 PHY RIB 属性, 该请求将通过发布 PLME-GET.request 原语传递给 PHY。MLME 接收到 PLME-GET.confirm 原语后, 由于 PHY 使用的状态值与 MLME 使用的不同(比如, 在 MAC 和 PHY 的枚举表格中, 状态 SUCCESS 的值分别是 0x00 和 0x07), MLME 将对接收的状态值进行转换, 然后向邻近高层发布 MLME-GET.confirm 原语, 且包含从 PLME 原语中返回的参数 RIBAttribute 和 RIBAttributeValue 对应的值。

7.1.7.3 MLME-GET.confirm

7.1.7.3.1 服务原语语义

原语 MLME-GET.confirm 用于报告 RIB 信息获取请求的结果。

原语 MLME-GET.confirm 的语义如下：

```
MLME-GET.confirm (
    status,
    RIBAttribute,
    RIBAttributeIndex,
    RIBAttributeValue
)
```

表 42 描述了原语 MLME-GET.confirm 的参数。

表 42 MLME-GET.confirm 的参数

名称	类型	有效范围	描述
status	枚举	SUCCESS, UNSUPPORTED_ATTRIBUTE INVALID_INDEX	RIB 属性信息获取请求的结果
RIBAttribute	整型	见表 71	特定 RIB 属性的标识
RIBAttributeIndex	整型	属性索引,定义见表 71	特定 RIB 属性的表的索引。该参数只对 MAC RIB 属性中的表格或阵列有效。当访问 PHY RIB 属性时,忽略该参数
RIBAttributeValue	可变	属性定义,见表 71	特定 RIB 属性的值。 当参数 status 设为 UNSUPPORTED_ATTRIBUTE 时,该参数长度为 0

7.1.7.3.2 产生条件

原语 MLME-GET.confirm 由 MLME 产生并发布给其邻近高层,用于响应原语 MLME-GET.request。该原语返回状态 SUCCESS,指示读取指定 RIB 属性的请求成功,或者返回一个错误码 UNSUPPORTED\_ATTRIBUTE。当返回 UNSUPPORTED\_ATTRIBUTE 时,参数 RIBAttribute 的值设为长度零。status 值的完整定义见 7.1.7.2.3。

7.1.7.3.3 原语使用

接收到 MLME-GET.confirm 原语后,邻近高层就获知了其 RIB 属性读取请求的结果。如果请求成功,参数 status 为 SUCCESS,否则参数 status 将指示发生的错误。

7.1.8 SCFP 管理原语

7.1.8.1 概述

MLME-SAP 的 SCFP 管理原语定义如何请求和维护 SCFP。要使用这些原语和 SCFP 的设备将



跟踪它的 RWSN 协调器的信标帧。

SCFP 管理原语是可选的。

### 7.1.8.2 MLME-SCFP.request

#### 7.1.8.2.1 服务原语语义

原语 MLME-SCFP.request 允许设备向 RWSN 协调器发送请求,申请分配一个新的 SCFP 或者取消一个已有 SCFP。RWSN 协调器也可利用该原语发起取消一个已分配的 SCFP。

原语 MLME-SCFP.request 的语义如下:

```
MLME-SCFP.request      (
                          SCFPCharacteristics,
                          )
```

表 43 描述了原语 MLME-SCFP.request 的参数。

表 43 MLME-SCFP.request 的参数

名称	类型	有效范围	描述
SCFPCharacteristics	SCFP 特性	见 7.3.9.3	SCFP 请求的特性,包括该请求是申请一个新的 SCFP 还是取消一个已有的 SCFP

#### 7.1.8.2.2 原语使用

原语 MLME-SCFP.request 由设备的邻近高层产生并发布给其 MLME,请求分配一个新 SCFP 或请求取消一个已有的 SCFP。也可由 RWSN 协调器的邻近高层产生并发布给其 MLME,请求取消一个已有的 SCFP。

#### 7.1.8.2.3 收后效果

设备接收到 MLME-SCFP.request 后,其 MLME 尝试产生一个 SCFP 请求命令(见 7.3.9),并且请求命令中包含该原语携带的信息,如果产生成功,就发给 RWSN 协调器。

如果 macShortAddress 的值是 0xffff 或 0xffff,则不允许该设备申请 SCFP。这时,MLME 将发布原语 MLME-SCFP.confirm,其状态为 NO\_SHORT\_ADDRESS。

如果因为 CSMA-CA 算法失败而无法发送 SCFP 请求命令,MLME 将丢弃该帧,并发布原语 MLME-SCFP.confirm,其状态为 CHANNEL\_ACCESS\_FAILURE。

如果 MLME 成功发送 SCFP 请求命令,就等待收到回复确认。如果没有收到回复确认,MLME 将发布 MLME-SCFP.confirm 原语,其状态为 NO\_ACK。

如果 SCFP 正在被分配,且已经收到了请求的确认,设备将等待通过 RWSN 协调器的信标帧指定的 SCFP 描述符指示的确认。如果 RWSN 协调器 MLME 能够分配所请求的 SCFP,它将发布原语 MLME-SCFP.indication,携带所分配的 SCFP 的特征,并生成一个 SCFP 描述符,该描述符包含分配的 SCFP 特征和请求设备的 16 比特短地址。如果 RWSN 协调器的 MLME 不能分配所请求的 SCFP,就生成一个 SCFP 描述符,该描述符起始时隙为零,包含请求设备的短地址。

如果设备接收到来自 RWSN 协调器的信标帧,且信标帧中的 SCFP 描述符包含的 16 比特短地址与其 macShortAddress 匹配,设备将处理该描述符。如果没有收到该设备的描述符,MLME 将发布原语 MLME-SCFP.confirm,其状态为 NO\_DATA。

如果收到一个描述符,且其包含的特征与请求的特征相符(RWSN 协调器同意了 SCFP 的分配请

求),设备的 MLME 将发布 MLME-SCFP.confirm 原语,其状态为 SUCCESS,且包含 SCFPCharacteristics 参数,参数的特征类型等于 1,表明分配了一个 SCFP。

如果收到一个描述符,且其起始时隙为 0(RWSN 协调器拒绝了 SCFP 的分配请求),请求 SCFP 的设备将发布 MLME-SCFP.confirm 原语,其状态为 DENIED,指示将忽略参数 SCFPCharacteristics。

如果在设备的请求下一个 SCFP 正在被取消,且 RWSN 协调器已经对请求的进行了确认,设备将发布原语 MLME-SCFP.confirm,其状态为 SUCCESS,以及参数 SCFPCharacteristics,且参数中的特征类型为 0,指示取消了 SCFP。当收到请求取消一个 SCFP 的 SCFP 请求命令,RWSN 协调器对该帧进行确认,并取消该 SCFP。RWSN 协调器的 MLME 将发布 MLME-SCFP.indication 原语,包含相应的 SCFP 特征。如果 RWSN 协调器没有收到取消 SCFP 的请求,协调器将采用相应的对策以保证 SCFP 的连续性(见 7.3.9.3)。

如果 RWSN 协调器的 MLME 收到 MLME-SCFP.request 原语,请求取消已分配的 SCFP,协调器将取消相应的 SCFP,并发布 MLME-SCFP.confirm 原语,其状态为 SUCCESS,其包含参数 SCFPCharacteristics,参数中的特征类型等于 0。

如果设备收到 RWSN 协调器的信标帧,帧中 SCFP 描述符包含的短地址与 macShortAddress 匹配,且起始时隙为 0,设备会立即停止使用该 SCFP。设备的 MLME 发布原语 MLME-SCFP.indication 通知其邻近高层 SCFP 的取消,原语的 SCFPCharacteristics 参数包含所取消 SCFP 的特征。

如果 MLME-SCFP.request 原语中任何参数的取值不被支持或超出范围,MLME 将发布原语 MLME-SCFP.confirm,其状态为 INVALID\_PARAMETER。

### 7.1.8.3 MLME-SCFP.confirm

#### 7.1.8.3.1 服务原语语义

原语 MLME-SCFP.confirm 用于报告分配一个新 SCFP 或者取消一个已存在 SCFP 的请求的结果。

原语 MLME-SCFP.confirm 的语义如下:

```
MLME-SCFP.confirm    (
                        SCFPCharacteristics,
                        status
                        )
```

表 44 描述了 MLME-SCFP.confirm 的参数。

表 44 MLME-SCFP.confirm 的参数

名称	类型	有效范围	描述
SCFPCharacteristics	SCFP 特征	见 7.3.9.3	SCFP 的特征
status	枚举	SUCCESS, DENIED, NO_SHORT_ADDRESS, CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, COUNTER_ERROR, FRAME_TOO_LONG, INVALID_PARAMETER	SCFP 请求的状态

### 7.1.8.3.2 产生条件

原语 MLME-SCFP.confirm 由 MLME 产生并发布给其邻近高层,用以响应先前发布的原语 MLME-SCFP.request。

如果申请或取消 SCFP 的请求成功,该原语将返回状态 SUCCESS,并且 SCFPCharacteristics 参数的特征类型域将分别设为 1 或 0。否则参数 status 将指示错误码。status 取值的原因描述见 7.1.8.2.3。

### 7.1.8.3.3 原语使用

收到 MLME-SCFP.confirm 原语,邻近高层就获知请求分配或取消一个 SCFP 的结果。如果请求成功,参数 status 将指示一个成功的 SCFP 操作。否则参数 status 将指示相应的错误。

## 7.1.8.4 MLME-SCFP.indication

### 7.1.8.4.1 服务原语语义

原语 MLME-SCFP.indication 用于指示一个 SCFP 已经被分配或者取消。

原语 MLME-SCFP.indication 的语义如下:

```
MLME-SCFP.indication    (
                          DeviceAddress,
                          SCFPCharacteristics,
                          )
```

表 45 描述了 MLME-SCFP.indication 的参数。

表 45 MLME-SCFP.indication 的参数

名称	类型	有效范围	描述
DeviceAddress	设备地址	0x0000~0xffffd	已分配或取消一个 SCFP 的设备的 16 比特短地址
SCFPCharacteristics	SCFP 特征	见 7.3.9.3	SCFP 的特性

### 7.1.8.4.2 产生条件

在 MLME 收到 SCFP 请求命令后,无论是分配还是取消了一个 SCFP,RWSN 协调器的 MLME 都将产生并向其邻近高层发布 MLME-SCFP.indication 原语。当 SCFP 取消命令由 RWSN 协调器发起时,RWSN 协调器的 MLME 也会产生该原语。如果分配了一个 SCFP,参数 SCFPCharacteristics 的特征类型字段将为 1,如果取消了一个 SCFP 时为 0。

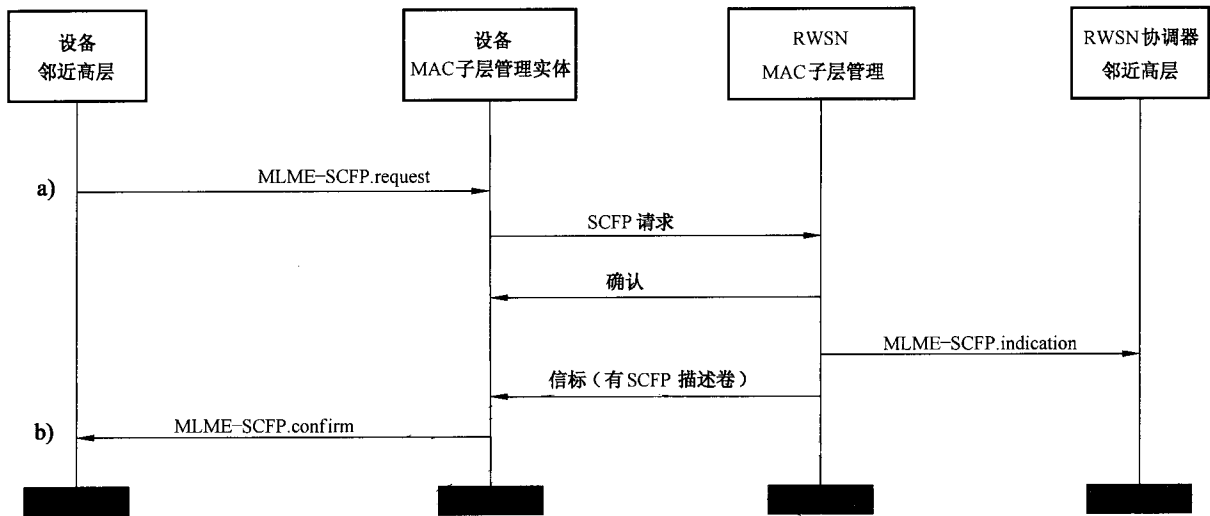
当 RWSN 协调器已取消了一个 SCFP,设备的 MLME 也会产生并向其邻近高层发布该原语。此时,参数 SCFPCharacteristics 的特征类型字段将为 0。

### 7.1.8.4.3 原语使用

收到 MLME-SCFP.indication 原语,邻近高层就获知已分配或取消了一个 SCFP。

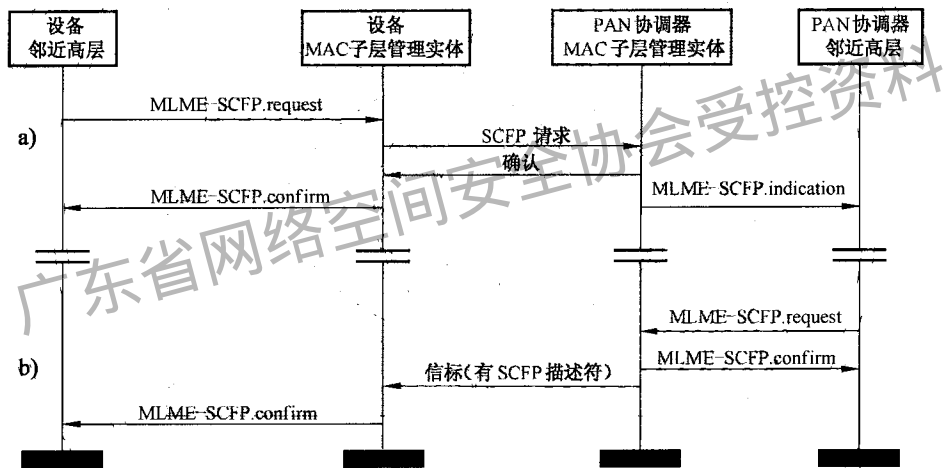
## 7.1.8.5 SCFP 管理消息序列图

图 24 是成功的 SCFP 管理的消息序列图。图 22 是设备发起的 SCFP 分配的消息序列流程,图 23 表示取消 SCFP 的两种情况,由设备发起或由 RWSN 协调器发起。



注：图中 a)部分由设备发起。b)部分 RWSN 协调器发起。

图 22 设备发起的 SCFP 分配消息序列图



注：图中 a)部分由设备发起。b)部分 RWSN 协调器发起。

图 23 SCFP 取消分配消息序列图

### 7.1.9 MAC 层复位原语

#### 7.1.9.1 概述

MLME-SAP 复位原语定义如何将 MAC 层复位,恢复默认值。  
所有设备都要为复位原语提供接口。

#### 7.1.9.2 MLME-RESET.request

##### 7.1.9.2.1 服务原语语义

原语 MLME-RESET.request 允许邻近高层请求 MLME 完成复位操作。  
原语的语义如下：

MLME-RESET.request ( SetDefaultRIB

表 46 描述了 MLME-RESET.request 的参数。

表 46 MLME-RESET.request 的参数

名称	类型	有效范围	描述
SetDefaultRIB	布尔型	TRUE 或 FALSE	如果该参数为 TRUE,则表示 MAC 层复位,所有 MAC RIB 属性设为默认值。如果为 FALSE,则表示 MAC 复位,但所有 MAC RIB 属性保持原语 MLME-RESET.request 产生前的值

#### 7.1.9.2.2 原语使用

邻近高层产生并发布 MLME-RESET.request 原语,请求 MLME 将 MAC 层复位到初始状态。在使用原语 MLME-START.request 或 MLME-ASSOCIATE.request 之前,先发布原语 MLME-RESET.request。

#### 7.1.9.2.3 收后效果

收到 MLME-RESET.request 原语后,MLME 发布 PLME-SET-TRX-STATE.request 原语,其状态为 FORCE\_TRX\_OFF。收到 PLME-SET-TRX-STATE.confirm 原语后,MAC 层将恢复到初始状态,将所有内部参数恢复为默认值。如果参数 SetDefaultRIB 设为 TRUE,MAC RIB 属性设为默认值。完成复位后,发布原语 MLME-RESET.confirm,其状态为 SUCCESS。

#### 7.1.9.3 MLME-RESET.confirm

##### 7.1.9.3.1 服务原语语义

原语 MLME-RESET.confirm 报告复位操作的结果。

原语 MLME-RESET.confirm 的语义如下:

```
MLME-RESET.confirm (
    status
)
```

表 47 描述了 MLME-RESET.confirm 的参数。

表 47 MLME-RESET.confirm 的参数

名称	类型	有效范围	描述
status	枚举	SUCCESS	复位操作的结果

#### 7.1.9.3.2 产生条件

收到 PLME-SET-TRXSTATE.confirm 原语后,MLME 产生原语 MLME-RESET.confirm 并发布给其邻近高层,作为 MLME-RESET.request 原语的响应。

#### 7.1.9.3.3 原语使用

邻近高层收到 MLME-RESET.confirm 原语就获知了请求复位 MAC 层的结果。该原语返回状态

SUCCESS,表明 MAC 层复位成功。

### 7.1.10 接收机使能时间指定原语

#### 7.1.10.1 概述

MLME-SAP 接收机状态原语定义了设备如何在指定时间打开或关闭接收机。这些接收机状态原语是可选的。

#### 7.1.10.2 MLME-RX-ENABLE.request

##### 7.1.10.2.1 服务原语语义

原语 MLME-RX-ENABLE.request 允许邻近高层请求接收机在限定的时间内打开或者关闭。原语 MLME-RX-ENABLE.request 的语义如下：

```
MLME-RX-ENABLE.request    (
                            DeferPermit,
                            RxOnTime,
                            RxOnDuration
                            )
```

表 48 描述了 MLME-RX-ENABLE.request 的参数。

表 48 MLME-RX-ENABLE.request 的参数

名称	类型	有效范围	描述
DeferPermit	布尔型	TRUE 或 FALSE	如果规定时间已过时,请求的操作可以推迟直到下一超帧,该参数设为 TRUE。如果请求的操作只能在当前超帧内进行,该参数设为 FALSE。 如果发起设备是 RWSN 协调器,提到的超帧指的是它的超帧,否则指的是设备所关联的协调器的超帧
RxOnTime	整型	0x000000~0xfffff	该参数是指接收机打开或关闭时距离超帧起始处的符号数。该参数是 24 比特,精度最少为 20 比特,最低 4 比特是最低有效位。 如果发起设备是 RWSN 协调器,超帧指的是它自己的超帧,否则是设备所关联的协调器的超帧
RxOnDuration	整型	0x000000~0xfffff	接收机打开的符号数。如果该参数等于 0x000000,那么接收机被关闭

##### 7.1.10.2.2 原语使用

邻近高层产生并向 MLME 发布 MLME-RX-ENABLE.request 原语,打开接收机。在信标使能的 RWSN 中,在当前或下一个工作超帧起始时打开接收机,而在不使用信标能力的 RWSN 中则立即打开接收机。该原语也可以用于取消先前打开接收机的请求。每次原语的请求都只能打开或关闭接收机一次。

##### 7.1.10.2.3 收后效果

MLME 处理打开或关闭接收机请求的优先级低于设备的其他任务(例如:SCFP、协调器信标跟踪和信标传输)。如果发布该原语用以打开接收机,设备将打开接收机,直到设备出现了冲突或者超过 RxOnDuration 定义的时间。如果有与之冲突的任务,设备将暂停接收操作。中断结束后,需要检查 RxOnDuration 以确定时

间是否结束。如果是,结束操作。如果没有,设备将再次打开接收机,直到出现另一个冲突或者超过持续时间。如果发布该原语是关闭接收机,设备将关闭接收机,除非设备出现冲突。

在信标使能的 RWSN 中,MLME 会在打开接收机之前,先判断(RxOnTime + RxOnDuration)是否小于由 macBeaconOrder 定义的信标间隔。如果(RxOnTime + RxOnDuration)不小于信标间隔,MLME 将发布 MLME-RXENABLE.confirm 原语,其状态为 ON\_TIME\_TOO\_LONG。

然后 MLME 将判断能否在当前超帧内打开接收机。发布该原语的 RWSN 协调器根据自己的超帧作出判断。不是 RWSN 协调器的设备则根据其所关联的协调器的超帧作出判断。如果从超帧开始到当前时刻的符号数目小于(RxOnTime-aTurnaroundTime),MLME 将尝试在当前超帧打开接收机。从超帧开始到当前时刻的符号数目大于等于(RxOnTime-aTurnaroundTime),并且 DeferPermit 为 TRUE,MLME 将延迟到下一个工作超帧打开接收机。否则,如果 MLME 既不能在当前超帧内打开接收机,同时又不允许延迟到下一个工作超帧,MLME 将发布 MLME-RX-ENABLE.confirm 原语,其状态为 PAST\_TIME。

如果参数 RxOnDuration 等于 0,MLME 将请求 PHY 关闭接收机。

如果 MLME-RX-ENABLE.request 原语中的参数设置不被支持或超出有效范围,MAC 层将发布 MLME-RXENABLE.confirm 原语,其状态为 INVALID\_PARAMETER。

如果打开或关闭接收机的请求成功,MLME 将发布 MLME-RXENABLE.confirm 原语,其状态为 SUCCESS。

### 7.1.10.3 MLME-RX-ENABLE.confirm

#### 7.1.10.3.1 服务原语语义

原语 MLME-RX-ENABLE.confirm 报告打开或关闭接收机请求的结果。

原语 MLME-RX-ENABLE.confirm 的语义如下:

```
MLME-RX-ENABLE.confirm    (
                               status
                             )
```

表 49 描述了 MLME-RX-ENABLE.confirm 的参数。

表 49 MLME-RX-ENABLE.confirm 的参数

名称	类型	有效范围	描述
status	枚举	SUCCESS, PAST_TIME, ON_TIME_TOO_LONG 或 INVALID_PARAMETER	请求打开或关闭接收机的结果

#### 7.1.10.3.2 产生条件

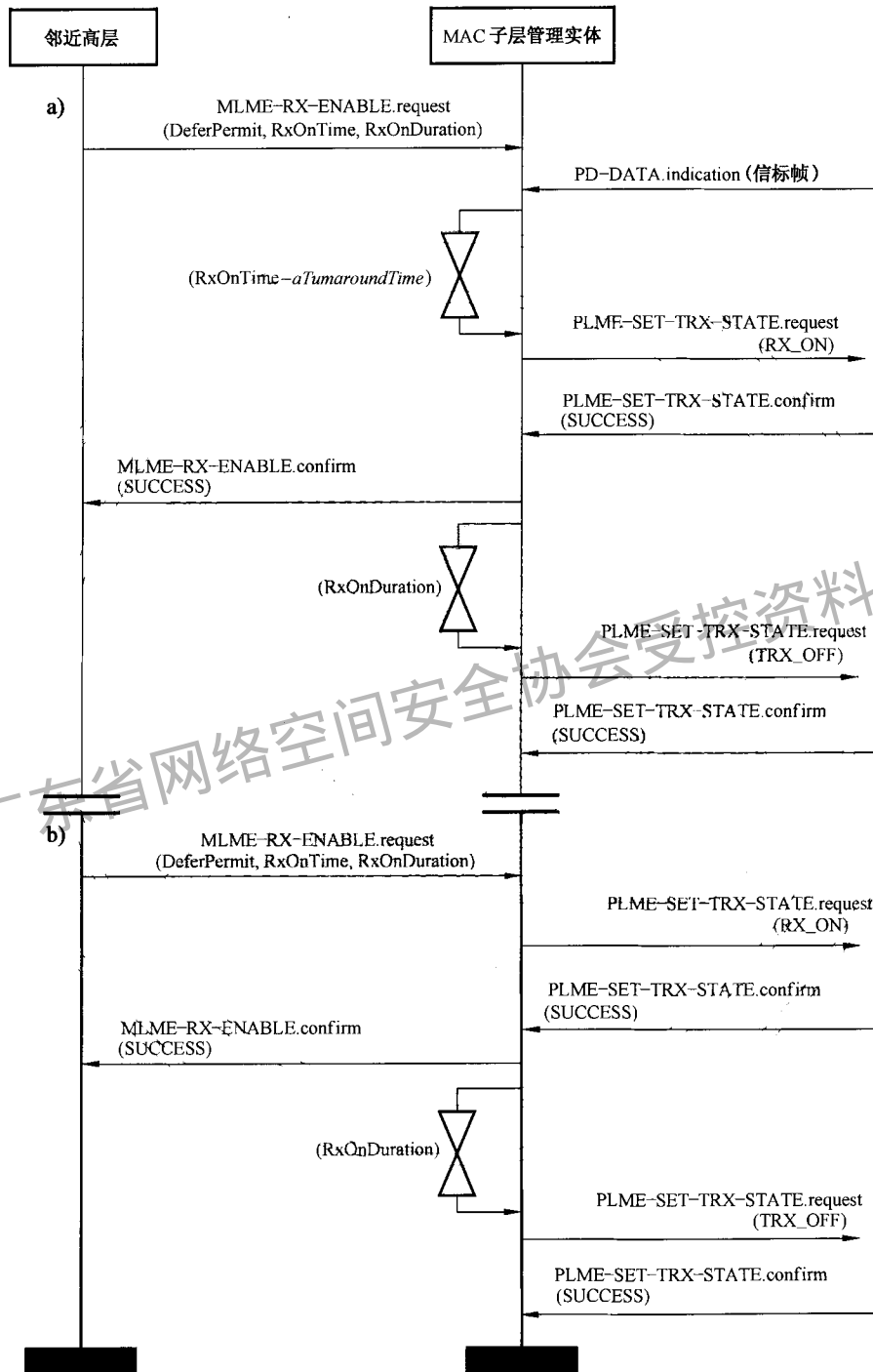
为响应原语 MLME-RX-ENABLE.request,MLME 产生并向其邻近高层发布原语 MLME-RX-ENABLE.confirm。

#### 7.1.10.3.3 原语使用

收到原语 MLME-RX-ENABLE.confirm 后,邻近高层就获知其打开或关闭接收机请求的结果。如果请求成功,该原语返回状态 SUCCESS,否则返回相应的错误码。状态的值完整描述见 7.1.10.2.3。

7.1.10.4 接收机状态转换消息序列图

图 24 描述了在没有中断的情况下,设备打开其接收机并保持一段时间的消息序列图。



注：图中 a) 部分描述了在信标使能的 RWSN 中,MLME 接收到 MLME-RXENABLE.request 原语但在当前超帧中没有足够时间来打开接收机,并且 DeferPermit 参数为 TRUE 的情况下的接收机状态转换的消息序列图。图中 b) 部分描述的是在不使用信标能力的 RWSN 中立即打开接收机的消息序列图。

图 24 接收机状态转换消息序列图



7.1.11 信道扫描原语

7.1.11.1 概述

MLME-SAP 信道扫描原语定义在一个通信信道上设备的能量使用,以及在此信道上是否存在 RWSN。

所有设备都为扫描原语提供接口。

7.1.11.2 MLME-SCAN.request

7.1.11.2.1 服务原语语义

MLME-SCAN.request 用于启动针对给定信道列表的扫描。设备可以通过信道扫描来检测信道上的能量,查找关联的协调器,或者查找扫描设备 POS 内的发送信标的所有协调器。

原语 MLME-SCAN.request 的语义如下:

```
MLME-SCAN.request (
    ScanType,
    ScanChannels,
    ScanDuration,
    ChannelPage,
)
```

表 50 描述了 MLME-SCAN.request 的参数。

表 50 MLME-SCAN.request 的参数

名称	类型	有效范围	描述
ScanType	整型	0x00~0x03	指示扫描的类型: 0x00 表示 ED 扫描(对于 RFD 可选); 0x01 表示主动扫描(对于 RFD 可选); 0x02 表示被动扫描; 0x03
ScanChannels	位图	16 比特字段	该 16 比特( $b_0, b_1, \dots, b_{15}$ )指示参数 ChannelPage 支持的将要扫描的信道(1 表示扫描,0 表示不扫描)
ScanDuration	整型	0~14	该参数用于计算每个信道的 ED、主动或被动扫描所需的时间长度。每个信道的扫描时间为: $[aBaseSuperframeDuration * (2^n + 1)]$ 个符号, 其中 n 表示参数 ScanDuration 的值
ChannelPage	整型	0~15	扫描的信道页(见 6.2.2)

7.1.11.2.2 原语使用

邻近高层产生 MLME-SCAN.request 原语,并发布给 MLME,请求进行信道扫描,搜索设备 POS 范围内的相关活动。该原语可以用于进行 ED 扫描来判断信道使用情况,进行主动扫描或被动扫描来定位包含 RWSN 标识符的信标帧。每种扫描的详细描述见 7.5.3.1。

所有设备都须能执行被动;ED 扫描和主动扫描对于 RFD 是可选的,然而一个 RFD 可以支持主动

扫描来加入到不用信标能力的网络中。

### 7.1.11.2.3 收后效果

如果 MLME 收到 MLME-SCAN.request 原语时,仍在进行上一次扫描操作,将发布原语 MLME-SCAN.confirm,其状态为 SCAN\_IN\_PROGRESS。否则,MLME 将对参数 ScanChannels 指定的所有信道进行扫描。

MLME 反复向 PHY 发布原语 PLME-ED.request 到 PHY 来实现对每个信道的 ED 扫描,持续时间是 $[aBaseSuperframeDuration * (2^n + 1)]$ 个符号,其中  $n$  是参数 ScanDuration 的值。MLME 记录能量的最大值,然后转移到信道列表中的下一个信道。ED 信道扫描过程的细节见 7.5.3.1.2。

对每个信道的主动扫描首先由 MLME 发送信标请求命令(见 7.3.7)开始。然后 MLME 打开接收机,并记录下每个接收到的 RWSN 描述符结构(见表 40)中信标帧携带的信息。当存储的 RWSN 描述符个数等于所指定最大值时,或扫描时间已持续 $[aBaseSuperframeDuration * (2^n + 1)]$ 个符号,这里  $n$  是参数 ScanDuration 的值,将停止在指定信道上的主动扫描。主动信道扫描过程的细节见 7.5.3.1.3。

每个信道的被动扫描是由 MLME 激打开接收机,并记录下每个接收到的 RWSN 描述符结构(见表 40)中信标帧携带的信息来实现的。当存储的 RWSN 描述符个数等于所指定最大值时,或扫描时间已持续 $[aBaseSuperframeDuration * (2^n + 1)]$ 个符号,这里  $n$  是参数 ScanDuration 的值,将停止在指定信道上的被动扫描。被动信道扫描过程的细节见 7.5.3.1.4。

ED 扫描的结果记录在 ED 值列表中,并通过 MLME-SCAN.confirm 原语向邻近高层报告,原语状态为 SUCCESS。

主动扫描和被动扫描的结果通过 MLME-SCAN.confirm 原语报告给邻近高层。如果扫描成功且 macAutoRequest 设为 TRUE,原语的结果会包括 RWSN 描述符的值集。如果扫描成功但 macAutoRequest 设为 FALSE,原语结果包含 RWSN 描述符值的空集,每个 RWSN 描述符的值将通过单独的 MLME-BEACON-NOTIFY 原语(见 7.1.6.2)分别发送给邻近高层。在以上两种情况下,MLME-SCAN.confirm 原语都将包含未扫描信道的列表,和状态 SUCCESS。

如果在主动扫描期间,因为信道接入失败,使得 MLME 无法在参数 ScanChannels 指定的信道上发送信标请求命令,则将该信道记录在原语 MLME-SCAN.confirm 返回的未扫描信道列表中。如果 MLME 能在其中至少的一个信道上发送信标请求命令,但未找到任何信标,那么无论 macAutoRequest 取何值,MLME-SCAN.confirm 原语包含的 RWSN 描述符值的空集和状态 NO\_BEACON。

如果在 ED 扫描、主动扫描或被动扫描期间,已扫描的信道个数达到了指定的最大值就中止扫描,MAC 层发布原语 MLME-SCAN.confirm,其状态为 LIMIT\_REACHED。

如果 MLME-SCAN.request 原语中的参数设置不被支持或超出了有效范围,MAC 层将发布 MLME-SCAN.confirm 原语,其状态为 INVALID\_PARAMETER。

### 7.1.11.3 MLME-SCAN.confirm

#### 7.1.11.3.1 服务原语语义

原语 MLME-SCAN.confirm 报告信道扫描请求的结果。

原语 MLME-SCAN.confirm 的语义如下:

```
MLME-SCAN.confirm      (
                        status,
                        ScanType,
                        ChannelPage,
                        UnscannedChannels,
```

ResultListSize,  
EnergyDetectList,  
RWSNDescriptorList  
)

表 51 描述了 MLME-SCAN.confirm 的参数。

表 51 MLME-SCAN.confirm 的参数

名称	类型	有效范围	描述
status	枚举型	SUCCESS, LIMIT_REACHED, NO_BEACON, SCAN_IN_PROGRESS, COUNTER_ERROR, FRAME_TOO_LONG, UNAVAILABLE_KEY, UNSUPPORTED_SECURITY 或 INVALID_PARAMETER	扫描请求的状态
ScanType	整型	0x00~0x03	扫描的类型： 0x00 表示 ED 扫描(RFD 可选)； 0x01 表示主动扫描(RFD 可选)； 0x02 表示被动扫描； 0x03
ChannelPage	整型	0~15	信道扫描所在的信道页
UnscannedChannels	位图	200 比特字段	请求扫描的信道中没有被扫描的信道 (1=没有扫描,0 表示扫描过或没有要求扫描)。该参数对 ED 扫描无效
ResultListSize	整型	执行细节	返回的结果列表中元素的个数
EnergyDetectList	整数列表	每个整数的范围均为 0x00~0xff	ED 扫描期间每个信道的能量测量的列表。对于主动、被动扫描,该参数为空
RWSNDescriptorList	RWSN 描述符列表	见表 40	RWSN 描述符列表, 如果 macAutoReques 设为 TRUE, 列表中的每个 RWSN 描述符对应于在主动、被动扫描中查找到的信标。主动或被动扫描中 macAutoReques 设为 FALSE 时, 或在 ED 扫描中, 该参数为空

### 7.1.11.3.2 产生条件

当由 MLME-SCAN.request 发起的信道扫描结束后, MLME 产生并向邻近高层发布原语 MLME-SCAN.confirm。如果 MLME-SCAN.request 原语请求执行主动、被动, 参数 EnergyDetectList 将为空。如果 MLME-SCAN.request 原语请求执行 ED 扫描, 参数 RWSNDescriptorList 将为空, 同样, 如果 MLME-SCAN.request 原语请求执行主动或被动扫描且参数 macAutoRequest 设为 FALSE, 参数 RWSNDescriptorList 也将为空。MLME-SCAN.confirm 原语返回状态 SUCCESS, 表明扫描请求成功, 或者相应的错误码。状态的值完整描述见 7.1.11.2.3。

7.1.11.3.3 原语使用

邻近高层收到 MLME-SCAN.confirm 原语后就获知了扫描的结果。如果请求扫描成功,参数 status 将设为 SUCCESS,否则 status 参数指示相应的错误。

7.1.11.4 信道扫描消息序列图

ED 扫描、被动扫描、主动扫描的消息序列图(见图 83、图 86 和图 87)中包括了 PHY 的操作。

7.1.12 通信状态原语

7.1.12.1 概述

MLME-SAP 通信状态原语定义了 MLME 如何向邻近高层报告传输状态(当传输是由响应原语发起时),或者流入帧的安全错误。

所有设备都须为通信状态原语提供接口。

7.1.12.2 MLME-COMM-STATUS.indication

7.1.12.2.1 服务原语语义

原语 MLME-COMM-STATUS.indication 允许 MLME 指示通信状态。

原语 MLME-COMM-STATUS.indication 的语义如下:

MLME-COMM-STATUS.indication (

- RWSNId,
- SrcAddrMode,
- SrcAddr,
- DstAddrMode,
- DstAddr,
- Status,

)

表 52 描述了 MLME-COMM-STATUS.indication 的参数。

表 52 MLME-STATUS.indication 的参数

名称	类型	有效范围	描述
RWSNId	整型	0x0000~0xffff	帧的发送者或者接收者的 RWSN 标识符
SrcAddrMode	整型	0x00~0x03	原语的源地址的方式,取值如下: 0x00 表示无地址(忽略地址域); 0x01 表示保留; 0x02 表示 16 比特短地址; 0x03 表示 64 比特扩展地址
SrcAddr	设备地址	由参数 SrcAddrMode 规定	引起错误的帧的发起者的独立设备地址
DstAddrMode	整型	0x00~0x03	原语的目的地址的方式,取值如下: 0x00 表示无地址(忽略地址域); 0x01 表示保留; 0x02 表示 16 比特短地址; 0x03 表示 64 比特扩展地址

表 52 (续)

名称	类型	有效范围	描述
DstAddr	设备地址	由 DstAddrMode 参数规定	帧接收设备的独立设备地址
Status	枚举	SUCCESS, TRANSACTION_OVERFLOW, TRANSACTION_EXPIRED, CHANNEL_ACCESS_FAILURE, NO_ACK,COUNTER_ERROR, FRAME_TOO_LONG, UNSUPPORTED_LEGACY, 或 INVALID_PARAMETER	通信状态

#### 7.1.12.2.2 产生条件

MLME-COMM-STATUS.indication 原语由 MLME 在响应原语发起的传输之后产生并向 MLME 的邻近高层发布。

在处理完 MLME-ASSOCIATE.response 原语或 MLME ORPHAN.response 原语之后,MAC 层将产生和发布 MLME-COMM-STATUS.indication 原语。该原语返回的状态可以是 SUCCESS,表明传输请求成功,也可以是 TRANSACTION\_OVERFLOW,TRANSACTION\_EXPIRED,CHANNEL\_ACCESS\_FAILURE,NO\_ACK 或 INVALID\_PARAMETER(状态参数见 7.1.4.4.3 及表 71)。

邻近高层收到 MLME-COMM-STATUS.indication 原语就获知了传输过程的通信状态中发生的错误。

#### 7.1.13 RIB 属性设置原语

##### 7.1.13.1 概述

MLME-SAP 设置原语定义了如何写入 RIB 属性。

所有设备都为这些设置原语提供接口。

##### 7.1.13.2 MLME-SET.request

###### 7.1.13.2.1 服务原语语义

原语 MLME-SET.request 用于向指定的 RIB 属性中写入给定的参数值。

原语 MLME-SET.request 的语义如下:

```
MLME-SET.request    (
                    RIBAttribute,
                    RIBAttributeIndex,
                    RIBAttributeValue
                    )
```

表 53 描述了 MLME-SET.request 的参数。

表 53 MLME-SET.request 的参数

名称	类型	有效范围	描述
RIBAttribute	整型	见表 71	写入值的 RIB 属性的标识
RIBAttributeIndex	整型	指定属性见表 71	在写入值的指定 RIB 属性表中索引。该参数只有对是表的 MAC RIB 属性有效;如果要访问 PHY RIB 属性,该参数将被忽略
RIBAttributeValue	可变	指定属性 MAC RIB 是管理设备 MAC 层必备的属性。MAC RIB 中所包含的属性见表 71。标有(†)的属性(即只能由 MAC 层来设置)是只读属性,邻近高层只能通过 MLME-GET.request 原语读取。对于其他属性,邻近高层可以分别通过原语 MLME-GET.request 和 MLME-SET.request 进行读写。标有(◆)的属性表示对 RFD 可选,标有(*)的属性表示对 RFD 和 FFD 均是可选。 只读属性 macAckWaitDuration 取决于常量的组合和 PHY RIB 属性。常量和属性的关联公式见式(5),属性 macMaxTotalFrameTxTime 由邻近高层设置,并取决于 PHY 和 MAC RIB 属性和常量的组合公式见式(6),见表 71	写入 RIB 属性的值

#### 7.1.13.2.2 原语使用

邻近高层产生 MLME-SET.request 原语并发布给 MLME,要求设置指定的 RIB 属性,即写入数据。

#### 7.1.13.2.3 收后效果

收到 MLME-SET.request 原语后,MLME 需要校验指定的 RIB 属性是 MAC RIB 属性还是 PHY RIB 属性。如果是 MAC RIB 属性,MLME 就尝试将给定值写入它的数据库中的指定 MAC RIB 属性。如果参数 RIBAttribute 指定的属性是只读的(见表 71),MLME 就发布原语 MLME-SET.confirm,其状态为 READ\_ONLY。如果参数 RIBAttribute 指定的属性在数据库找不到,MLME 将发布 MLME-SET.confirm 原语,其状态为 UNSUPPORTED\_ATTRIBUTE。如果参数 RIBAttribute 指定的表格索引超出范围,MLME 就发布 MLME-SET.confirm 原语,其状态为 INVALID\_INDEX。如果参数 RIBAttribute 指定的值超出指定属性值的有效范围,MLME 就发布 MLME-SET.confirm,其状态为 INVALID\_PARAMETER。如果成功写入,MLME 将发布 MLME-SET.confirm 原语,其状态为 SUCCESS。

如果参数 RIBAttribute 指示要设置 macBeaconPayloadLength,且信标帧长度超过了 aMaxPHYPacketSize(例如,由于安全处理需要的额外开销),MAC 层就不能更新 macBeaconPayloadLength,并发布 MLME-GET.confirm 原语,其状态为 INVALID\_PARAMETER。

如果请求的属性是 PHY RIB 属性,就发布 PLME-SET.request 原语,把请求传递给 PHY。一旦 MLME 收到 PLME-SET.confirm 原语,就将收到的状态值进行转换,这是因为 PHY 和 MLME 使用的

状态值不相同(例如状态 SUCCESS 的值在 MAC 和 PHY 的枚举表中的值分别是 0x00 和 0x07)。转换后,MLME 就向邻近高层发布原语 MLME-SET.confirm,其状态为是转换后的值,且包含 PLME 原语返回的 RIBAttribute。

### 7.1.13.3 MLME-SET.confirm

#### 7.1.13.3.1 服务原语语义

原语 MLME-SET.confirm 用于报告尝试向 RIB 属性写入值的结果。

原语 MLME-SET.confirm 的语义如下:

```
MLME-SET.confirm      (
                        status,
                        RIBAttribute,
                        RIBAttributeIndex
                        )
```

表 54 描述了 MLME-SET.confirm 的参数。

表 54 MLME-SET.confirm 的参数

名称	类型	有效范围	描述
status	枚举	SUCCESS, READ_ONLY, UNSUPPORTED_ATTRIBUTE, INVALID_INDEX 或 INVALID_PARAMETER	请求向 RIB 属性写入值的结果
RIBAttribute	整型	MAC RIB 是管理设备 MAC 层必备的属性。MAC RIB 中所包含的属性见表 71。标有(†)的属性(即只能由 MAC 层来设置)是只读属性,邻近高层只能通过 MLME-GET.request 原语读取。对于其他属性,邻近高层可以分别通过原语 MLME-GET.request 和 MLME-SET.request 进行读写。标有(◆)的属性表示对 RFD 可选,标有(*)的属性表示对 RFD 和 FFD 均是可选。只读属性 macAckWaitDuration 取决于常量的组合和 PHY RIB 属性。常量和属性的关联公式见式(5),属性 macMaxTotalFrameTxTime 由邻近高层设置,并取决于 PHY 和 MAC RIB 属性和常量的组合公式见式(6),见表 71	指定写入值的 RIB 属性的标识符
RIBAttributeIndex	整型	属性规定;见表 71	在写入值的 RIB 属性表中的索引。该参数只对 MAC RIB 属性中的表格属性有效。当访问 PHY RIB 属性时,忽略该参数

#### 7.1.13.3.2 产生条件

MLME 产生并向邻近高层发布 MLME-SET.confirm 原语,用以响应 MLME-SET.request 原语。MLME-SET.confirm 原语返回状态 SUCCESS,指示请求数值已写入 RIB 属性,或者返回相应的错误

代码。状态值的完整定义见 7.1.13.2.3。

### 7.1.13.3.3 原语使用

邻近高层收到 MLME-SET.confirm 原语后就获知设置 RIB 属性请求的结果。如果请求值被写入指定的 RIB 属性中,参数 status 将被设为 SUCCESS,否则 status 参数指示相应的错误。

### 7.1.14 超帧配置更新原语

#### 7.1.14.1 概述

MLME-SAP 开始原语定义 FFD 如何请求开始使用新的超帧配置,用以建立一个新的 RWSN,或者开始在现有 RWSN 中发送信标,以方便设备发现,或停止信标传输。这些开始原语对于 RFD 是可选的。

#### 7.1.14.2 MLME-START.request

##### 7.1.14.2.1 服务原语语义

原语 MLME-START.request 允许 RWSN 协调器初始化一个新的 RWSN,或者在现有 RWSN 中启用新的超帧配置。也可以被已经关联的设备用于配置一个新的超帧。

原语 MLME-START.request 的语义如下:

```
MLME-START.request (
    RWSNId,
    LogicalChannel,
    ChannelPage,
    StartTime,
    BeaconOrder,
    SuperframeOrder,
    RWSNCoordinator,
    BatteryLifeExtension,
    CoordRealignment,
)
```

表 55 描述了 MLME-START.request 的参数。

##### 7.1.14.2.2 原语使用

邻近高层产生并向其 MLME 发布 MLME-START.request 原语,用于请求设备配置一个新的超帧。

##### 7.1.14.2.3 收后效果

当 macShortAddress 为 0xffff 时,如果收到 MLME-START.request 原语,MLME 将发布原语 MLME-START.confirm,状态为 NO\_SHORT\_ADDRESS。

当参数 CoordRealignment 设为 TRUE,协调器就尝试发送协调器解关联命令帧(见 7.5.3.3.3)。如果由于信道访问失败而无法发送该命令帧,MLME 就不会改变超帧结构(例如不会改变任何一个 RIB 属性),并发布原语 MLME-START.confirm,状态为 CHANNEL\_ACCESS\_FAILURE。如果成功发送命令,MLME 就根据参数 BeaconOrder、SuperframeOrder、RWSNId、ChannelPage 和 LogicalChannel 更新相应的 RIB 属性,同时发布原语 MLME-START.confirm,状态为 SUCCESS。



表 55 MLME-START.request 的参数

名称	类型	有效范围	描述
RWSNId	整型	0x0000~0xffff	设备使用的 RWSN 标识符
LogicalChannel	整型	选自参数 ChannelPage 指定的可用逻辑信道	新配置的超帧的逻辑信道
ChannelPage	整型	选自 PHY 支持的可用信道页(见 6.2.2)	新配置的超帧的信道页
StartTime	整型	0x000000~0xfffffff	信标发射的起始时间。如果该参数为 0x000000, 信标将立即开始发射。否则, 该参数指定的时间为与接收到信标的相对时间, 信标由设备同步的协调器发送。如果 BeaconOrder 等于 15 或 RWSNCoordinator 为 TRUE, 忽略该参数。时间以符号为单位, 并在退避时隙边缘开始。 该参数是 24 比特, 精度最少为 20 比特, 最低 4 比特是最低有效位
BeaconOrder	整型	0~15	该参数指信标发送的频率。值等于 15 表示协调器将不发送信标。 信标顺序和信标间隔之间的关系见 7.5.2.1
SuperframeOrder	整型	0~BO 或 15	包括信标在内的超帧活跃部分的长度。如果参数 BeaconOrder 为 15, 忽略该参数。 超帧顺序和超帧长度间的关系见 7.5.2.1
RWSNCoordinator	布尔型	TRUE 或 FALSE	如果为 TRUE, 设备将成为新 RWSN 的网络协调器。如果为 FALSE, 设备将在已关联的 RWSN 中启动新超帧结构
BatteryLifeExtension	布尔型	TRUE 或 FALSE	如果为 TRUE, 设备接收到信标, 等待一个 IFS 后关闭接收机, 持续时间是 macBattLifeExt Periods 个退避阶段。如果为 FALSE, 信标接收设备的接收机在整个 CAP 内都保持使能状态。 如果参数 BeaconOrder 等于 15, 忽略该参数
CoordRealignment	布尔型	TRUE 或 FALSE	如果在改变超帧的配置之前发送协调器重组命令, 参数就为 TRUE, 否则为 FALSE

当参数 CoordRealignment 设为 FALSE, MLME 就根据参数 BeaconOrder、SuperframeOrder、RWSNId、ChannelPage 和 LogicalChannel 更新相应的 RIB 属性。

协调器在信标帧中使用的地址方式由参数 macShortAddress 的当前值决定, 该参数的值是在原语发布前由上一高层设置的。如果参数 BeaconOrder 小于 15, MLME 就将 macBattLifeExt 设为参数 BatteryLifeExtension 的值。如果 BeaconOrder 等于 15, 忽略参数 BatteryLifeExtension。

如果信标帧的长度超过了 aMaxPHYPacketSize(例如, 安全处理会增加额外的负载), MAC 层就丢弃该信标帧, 同时发布原语 MLME-START.confirm, 状态为 FRAME\_TOO\_LONG。

如果参数 BeaconOrder 等于 15, 显示这是一个非信标使能 RWSN, MLME 将忽略参数 StartTime。如果参数 BeaconOrder 小于 15, MLME 将检查参数 StartTime 以决定传输信标帧的时间; 时间以符号为单位, 并在退避时隙边缘开始。如果参数 RWSNCoordinator 设为 TRUE, MLME 将忽略参数 Start-

Time 并立即传输信标帧。如果 StartTime 设为 0x000000, MLME 也会立即传输信标。当 MLME 收到与之连接的协调器发来的信标时, 如果参数 RWSNCoordinator 设为 FALSE, 且 StartTime 非 0, MLME 把 StartTime 个符号加到本地时钟取得的时间加上, 计算信标传输时间, 其中, 本地时钟是设备接收到所关联的协调器的信标帧时开始计时。如果计算得到的时间造成流出超帧与流入超帧重迭, MLME 就不会发送信标帧。在这种情况下, MLME 发布 MLME-START.confirm 原语, 状态为 SUPERFRAME\_OVERLAP。否则, 当本地时钟的当前时间和计算得到的时间相同时, MLME 将开始传输信标。

如果参数 StartTime 非 0, 且此时 MLME 未跟踪与之关联协调器的信标, MLME 将发布原语 MLME-START.confirm, 状态为 TRACKING\_OFF。

结束该过程后, MLME 发布 MLME-START.confirm 原语作为响应。如果成功启用新的超帧结构, 状态设为 SUCCESS。如果原语中任何一个参数的设置不被支持或超过有效范围, 状态将设为 INVALID\_PARAMETER。

### 7.1.14.3 MLME-START.confirm

#### 7.1.14.3.1 服务原语语义

原语 MLME-START.confirm 报告请求启用新超帧结构的结果。

原语 MLME-START.confirm 的语义如下:

```
MLME-START.confirm      (
                           status
                           )
```

表 56 描述了 MLME-START.confirm 原语的参数。

表 56 MLME-START.confirm 的参数

名称	类型	有效范围	描述
status	枚举	SUCCESS, NO_SHORT_ADDRESS, SUPERFRAME_OVERLAP, TRACKING_OFF, INVALID_PARAMETER, COUNTER_ERROR, FRAME_TOO_LONG, CHANNEL_ACCESS_FAILURE	请求启用新超帧结构的结果

#### 7.1.14.3.2 产生条件

MLME 向其邻近高层发布 MLME-START.confirm 原语, 以响应原语 MLME-START.request。MLME-START.confirm 原语返回状态可以是 SUCCESS, 显示 MAC 层已成功启用新的超帧结构, 也可以是错误码。状态的值可参考 7.1.14.2.3。

#### 7.1.14.3.3 原语的使用

邻近高层收到 MLME-START.confirm 原语就获知请求使用新的超帧结构的结果。如果成功, 参数 status 将设为 SUCCESS, 否则参数的其他值表明有错误。

7.1.14.4 更新超帧结构的消息序列图

图 25 是一个 FFD 发起信标传输的消息序列图。图 25 是 RWSN 协调器在一个新 RWSN 中开始发送信标的消息序列图,该图中包括 PHY 的操作。

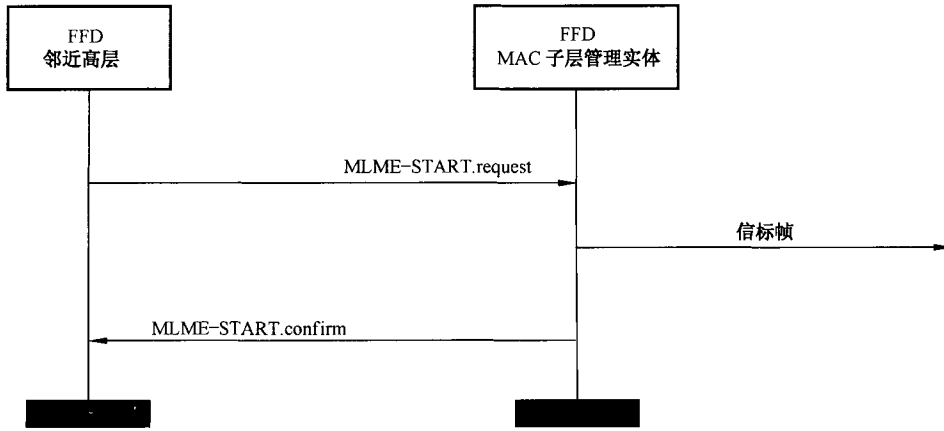


图 25 更新超帧结构的消息序列图

7.1.15 与协调器同步原语

7.1.15.1 概述

MLME-SAP 同步原语定义如何实现与协调器的同步,以及在失去同步时,如何向邻近高层报告。

7.1.15.2 MLME-SYNC.request

7.1.15.2.1 服务原语语义

原语 MLME-SYNC.request 用于请求与协调器同步,通过获得和跟踪(要求时)协调器的信标实现。

原语 MLME-SYNC.request 的语义如下:

```

MLME-SYNC.request (
    LogicalChannel,
    ChannelPage,
    TrackBeacon
)
    
```

表 57 描述了 MLME-SYNC.request 的参数。

表 57 MLME-SYNC.request 的参数

名称	类型	有效范围	描述
<i>LogicalChannel</i>	整型	选自 PHY 支持的可用逻辑信道。	将在其上尝试协调器同步的逻辑信道
<i>ChannelPage</i>	整型	选自 PHY 支持的可用信道页(见 6.2.2)	将在其上尝试协调器同步的逻辑信道页
<i>TrackBeacon</i>	布尔型	TRUE 或 FALSE	如果 MLME 将与下一个信标同步,并尝试跟踪以后的信标,该参数为 TRUE。如果 MLME 只与下一个信标同步,该参数为 FALSE

7.1.15.2.2 原语使用

原语 MLME-SYNC.request 由信标使能的 RWSN 中的设备的邻近高层产生,并发布给其 MLME,用于与协调器同步。

7.1.15.2.3 收后效果

如果 MLME-SYNC.request 原语被信标使能的 RWSN 中的 MLME 接收到,该 MLME 将首先通过发布 PLME-SET.request 原语,把参数 *phyChannelPage* 和 *phyCurrentChannel* 分别更新为原语中参数 *ChannelPage* 和 *LogicalChannel* 的值。如果参数 *TrackBeacon* 为 TRUE,MLME 将跟踪信标,即在每个信标预期时间之前打开其接收机,使每个信标都能被处理。如果 *TrackBeacon* 为 FALSE,MLME 将定位信标,但不会继续跟踪。

如果正在跟踪信标时收到该原语,MLME 将不会丢弃该原语,而是将此原语当作一个新的同步请求。

如果在初始查找阶段或者后续跟踪阶段都无法正确定位信标,MLME 将发布原语 MLME-SYNC-LOSS.indication,其同步失去原因为 BEACON\_LOST。

7.1.15.3 MLME-SYNC-LOSS.indication

7.1.15.3.1 服务原语语义

原语 MLME-SYNC-LOSS.indication 表明与协调器失去同步。

原语 MLME-SYNC-LOSS.indication 的语义如下:

```
MLME-SYNC-LOSS.indication (
    LossReason,
    RWSNId,
    LogicalChannel,
    ChannelPage,
)
```

表 58 规定了 MLME-SYNC-LOSS.indication 原语的参数。

表 58 MLME-SYNC-LOSS.indication 的参数

名称	类型	有效范围	描述
LossReason	枚举	RWSN_ID_CONFLICT, REALIGNMENT 或 BEACON_LOST	失去同步的原因
RWSNId	整型	0x0000~0xffff	设备与其失去同步的协调器或与其重新关联的协调器的 RWSN 的标识
LogicalChannel	整型	选自 PHY 支持的可用逻辑信道(见 6.2.2)	设备在其上失去同步的或者重新关联的逻辑信道
ChannelPage	整型	选自 PHY 支持的可用逻辑信道页(见 6.2.2)	设备在其上失去同步的或者重新关联的逻辑信道页

7.1.15.3.2 产生条件

原语 MLME-SYNC-LOSS.indication 由设备的 MLME 在设备与协调器失去同步时产生,并向其

邻近高层发布。也可以由 RWSN 协调器的 MLME 在检测到 RWSN ID 冲突时产生,并向其邻近高层发布。

如果与 RWSN 协调器关联的设备检测到 RWSN 标识符冲突,并且要通知给 RWSN 协调器,则设备的 MLME 将发布该原语,并设置参数 *LossReason* 为 RWSN\_ID\_CONFLICT。同样,如果 RWSN 协调器接收到 RWSN ID 冲突通知命令(见 7.3.6),其 MLME 将发布该原语,并设置参数 *LossReason* 为 RWSN\_ID\_CONFLICT。

如果设备接收到它与之关联的协调器发送的协调器重新关联命令(见 7.3.8),则设备的 MLME 将发布该原语,设置参数 *LossReason* 为 REALIGNMENT,并根据 7.5.3.3.4 的描述设置参数 *RWSNid*、*LogicalChannel*、*ChannelPage*。

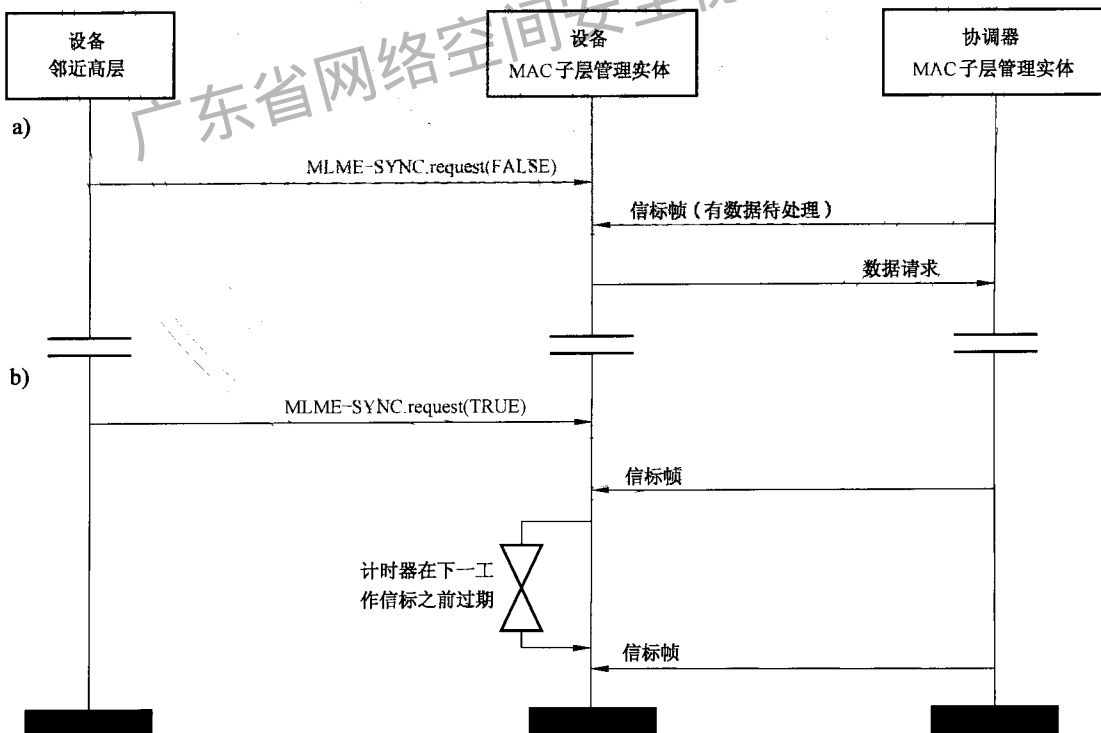
无论是在初始查找信标阶段还是在后续信标跟踪阶段,如果设备在发布 MLME-SYNC.request 原语之后的 *aMaxLostBeacons* 个工作超帧内,设备一直都没有监听到信标或监听到的信标的序列号与设备存储的 NWBSN 值不相符,设备的 MLME 将发布该原语,参数 *LossReason* 设置为 BEACON\_LOST,参数 *RWSNid*、*LogicalChannel* 和 *ChannelPage* 须根据与之失去同步的协调器设置。如果正在跟踪信标,MLME 将不再尝试跟踪信标。

7.1.15.3.3 原语使用

收到 MLME-SYNC-LOSS.indication 原语后,邻近高层就获知了失去同步的信息。

7.1.15.4 与协调器同步的消息序列图

图 26 描述的是设备与协调器同步的消息序列图。



注: 图中 a)部分表示发布单个的同步请求。b)部分表示发布跟踪同步请求。

图 26 在信标使能的 RWSN 中与协调器同步的消息序列图

7.1.16 数据请求原语

7.1.16.1 概述

MLME-SAP 数据请求原语定义设备向协调器请求数据的方式。  
所有设备都为数据请求原语提供接口。

7.1.16.2 MLME-POLL.request

7.1.16.2.1 服务原语语义

原语 MLME-POLL.request 促使设备从协调器请求数据。

原语 MLME-POLL.request 的语义如下：

```
MLME-POLL.request (
    CoordAddrMode,
    CoordRWSNId,
    CoordAddress,
)
```

表 59 描述了 MLME-POLL.request 的参数。

表 59 MLME-POLL.request 的参数

名称	类型	有效范围	描述
CoordAddrMode	整型	0x02~0x03	接收原语的协调器的地址方式。取值如下： 0x02 表示 16 比特短地址； 0x03 表示 64 比特扩展地址
CoordRWSNId	整型	0x0000~0xffff	接收原语的协调器的 RWSN 标识符
CoordAddress	设备地址	由 CoordAddrMode 参数规定。	接收原语的协调器的地址

7.1.16.2.2 原语使用

向协调器请求数据时，邻近高层向其 MLME 发布 MLME-POLL.request 原语。

7.1.16.2.3 收后效果

收到 MLME-POLL.request 原语后，MLME 生成并发送数据请求命令（见 7.3.5）。如果数据请求命令是直接发送给 RWSN 协调器，那么数据请求命令中可以没有任何目的地址信息。否则数据请求命令中总需要有 CoordRWSNId 和 CoordAddress 参数定义的目的地址信息。

如果 CSMA-CA 算法失败而无法发送数据请求命令，MAC 层会丢弃该帧，并发布原语 MLME-POLL.confirm，状态为 CHANNEL\_ACCESS\_FAILURE。

如果 MLME 成功发送数据请求命令，MLME 等待对方确认。如果没有收到确认，MLME 将发布 MLME-POLL.confirm 原语，状态为 NO\_ACK。

如果收到对方确认，并且确认帧的帧待处理子字段为 1，MLME 将请求 PHY 打开接收机。如果确认帧的帧待处理子字段为 0，MLME 将发布 MLME-POLL.confirm 原语，状态为 NO\_DATA。

如果来自协调器的帧净荷为 0，或帧为 MAC 命令帧，MLME 将发布 MLME-POLL.confirm 原语，状态为 NO\_DATA。如果帧净荷非 0，MLME 将发布 MLME-POLL.confirm 原语，状态为 SUCCESS。这时，数据通过 MCPS-DATA.indication（见 7.1.2.4）原语发给邻近高层。

如果在 *macMaxTotalFrameTxTime* 符号之内仍未收到数据帧,即时数据请求命令的确认帧的帧待处理子字段为 1,MLME 仍将发布原语 MLME-POLL.confirm,状态为 NO\_DATA。

如果原语 MLME-POLL.request 中的参数设置不被支持或超出有效范围,MLME 将发布 MLME-POLL.confirm 原语,状态为 INVALID\_PARAMETER。

### 7.1.16.3 MLME-POLL.confirm

#### 7.1.16.3.1 服务原语语义

原语 MLME-POLL.confirm 向协调器报告数据请求的结果。

原语 MLME-POLL.confirm 的语义如下:

```
MLME-POLL.confirm    (
                        status
                      )
```

表 60 描述了 MLME-POLL.confirm 的参数。

表 60 MLME-POLL.confirm 的参数

名称	类型	有效范围	描述
status	枚举	SUCCESS, CHANNEL_ACCESS_FAILURE, NO_ACK, NO_DATA, COUNTER_ERROR, FRAME_TOO_LONG, INVALID_PARAMETER	数据请求的状态

#### 7.1.16.3.2 产生条件

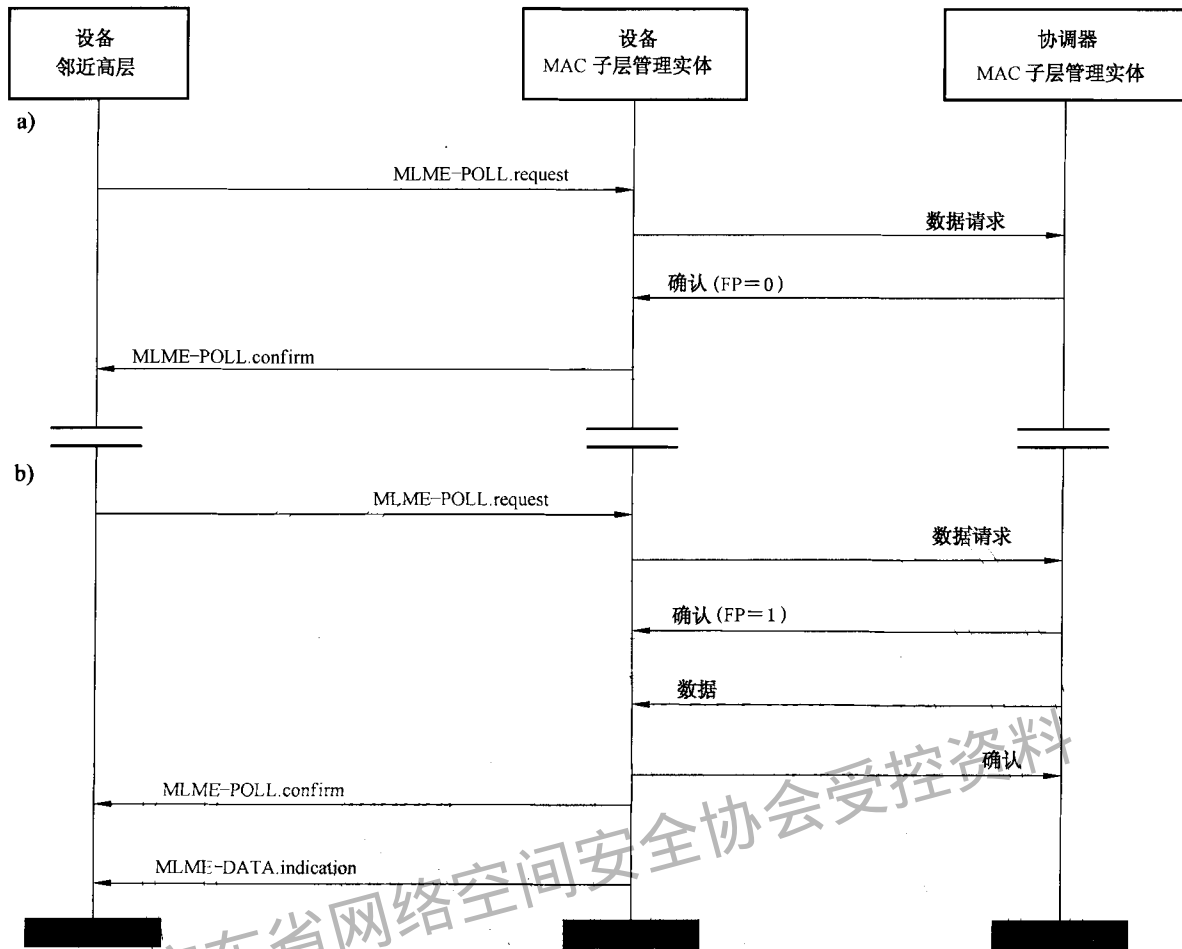
MLME 向邻近高层发布 MLME-POLL.confirm 原语以响应 MLME-POLL.request 原语。如果请求成功,参数 status 将为 SUCCESS,表明请求数据成功。否则参数的其他值表明有错误。参数的值见 7.1.16.2.3。

#### 7.1.16.3.3 原语使用

邻近高层收到 MLME-POLL.confirm 原语后获知向协调器数据请求的状态。

#### 7.1.16.4 从协调器请求数据的消息序列图

图 27 是设备向协调器请求数据的消息序列图,图中包括设备的层操作以及射频接口。在图 27a)和图 27b)中,MLME 收到询问请求后,向协调器发送数据请求命令。



注 1：图中 a) 部分，确认帧的帧待处理字段为 0，同时 MLME 立刻发布数据请求的确认。

注 2：图中 b) 部分，确认帧的帧待处理字段为 1，MLME 打开接收机接收协调器发送的数据。收到数据帧后，MLME 会向邻近高层发布数据请求确认，同时上传收到的数据。

图 27 向协调器请求数据的消息序列图

### 7.1.17 枚举型参数

本条详细描述在 MAC 层定义的原语中使用到的枚举参数。表 61 描述了 MAC 层定义的枚举型参数值的含义。

表 61 MAC 枚举型参数值的含义

名称	取值	描述
SUCCESS	0x00	请求的操作成功完成。对于一个传输请求，该值显示传输成功
--	0x01~0xda, 0xfe~0xff	为 MAC 命令状态和原因代码值保留
--	0x80~0xdf	保留
BEACON_LOSS	0xe0	请求同步后丢失信标
CHANNEL_ACCESS_FAILURE	0xe1	由于信道忙，无法进行传输，例如 CSMA-CA 算法失败



表 61 (续)

名称	取值	描述
COUNTER_ERROR	0xdb	发送帧的设备所使用的帧计数器无效
DENIED	0xe2	RWSN 协调器拒绝 SCFP 请求
DISABLE_TRX_FAILURE	0xe3	关闭收发器的请求失败
FRAME_TOO_LONG	0xe5	处理后帧长度超过 <i>aMaxPHYPacketSize</i> 或者请求的传输事务太大而不适合在 CAP 或 SCFP 中进行
INVALID_ADDRESS	0xf5	发送数据的请求失败,且原因不在于缺少源地址和目的地址
INVALID_SCFP	0xe6	因为指定的 SCFP 没有传输 SCFP 的方向,或未定义 SCFP,致使请求的 SCFP 传输失败
INVALID_HANDLE	0xe7	在事务表中没有找到请求所需要的 MSDU 句柄,就请求从事务队列中清除 MSDU
INVALID_INDEX	0xf9	由于指定的表格索引超出有效范围,使得无法成功将值写入表中的一个 MAC RIB 属性
INVALID_PARAMETER	0xe8	原语中的参数设置不被支持或超出有效范围
LIMIT_REACHED	0xfa	由于存储的 RWSN 描述符个数达到指定的最大值,提前结束扫描操作
NO_ACK	0xe9	在 <i>macMaxFrameRetries</i> 次重发后,仍未收到确认
NO_BEACON	0xea	扫描操作中没有搜索到任何网络信标
NO_DATA	0xeb	在发出请求之后没有收到任何响应
NO_SHORT_ADDRESS	0xec	因为没有分配到 16 比特短地址而无法进行操作
ON_TIME_TOO_LONG	0xf6	因为指定的符号个数大于信标间隔,造成打开接收机的请求失败
OUT_OF_CAP	0xed	由于打开接收机的请求无法在 CAP 内完成,致使请求失败
RWSN_ID_CONFLICT	0xee	检测到 RWSN 标识符冲突,并通知 RWSN 协调器
PAST_TIME	0xf7	因为无法在当前超帧完成,且不允许推迟到下一个工作超帧,使得打开接收机的请求失败
READ_ONLY	0xfb	标识属性设置为只读的 SET/GET(写或读)请求
REALIGNMENT	0xef	收到协调器重组命令
SCAN_IN_PROGRESS	0xfc	由于 MLME 还在执行先前发起的操作,使得扫描请求失败
SUPERFRAME_OVERLAP	0xfd	基于协调器的计时器,通知设备开始发送信标帧,但传输时间和协调器的传输时间重叠
TRACKING_OFF	0xf8	基于协调器的计时器,通知设备开始发送信标帧,但设备当前并没有跟踪协调器的信标
TRANSACTION_EXPIRED	0xf0	事务停止,相应信息被丢弃
TRANSACTION_OVERFLOW	0xf1	没有足够空间存储事务

## 7.2 MAC 帧格式

### 7.2.1 概述

本条详细描述了 MAC 帧(MPDU)的格式。每个 MAC 帧包括以下几个基本组成：

- a) 帧头：包括帧控制信息、帧序列号、地址信息和安全信息；
- b) 净荷：长度可变，具体内容有帧类型决定。确认帧不包含净荷；
- c) 帧尾：含有校验序列 FCS。

MAC 帧是一系列字段的特定排列。所有帧的格式都是按照 PHY 的发送顺序进行描述，从左到右，最左边的首先发送。在每个字段中，比特位从 0(最左边、最低位)到 k-1(最右边、最高位)编号，总长度为 k 比特。对于长度大于 1 个八位位组的字段，发送给 PHY 时，按照从小到大的八位位组编号进行。

对于所有帧中的保留位，发送时设为 0，接收时则忽略。

### 7.2.2 MAC 帧的通用格式

#### 7.2.2.1 通用帧格式描述

每个 MAC 帧格式都由帧头、净荷和帧尾三部分组成。其中帧头具有固定的字段排列顺序，但并不是所有的帧都包含地址字段。MAC 帧的一般格式如图 28 所示。

八位位组:2	1	0/2	0/2/8	0/2	0/2/8	0/5/6/ 10/14	可变	2
帧控制信息	帧序列号	目的设备 RWSN 标识符	目的 地址	源设备 RWSN 标识符	源设备地址	保留	帧净荷	FCS 校验码
地址信息								
帧头							MAC 净荷	帧尾

图 28 MAC 帧的通用格式

#### 7.2.2.2 帧控制字段

##### 7.2.2.2.1 格式描述

帧控制字段长度为 2 个八位位组，包含的信息有帧类型、地址信息和其他控制标记。帧控制字段格式图 29。

位:0~2	3	4	5	6	7	8~9	10~11	12~13	14~15
帧类型	保留	帧待 处理	Ack 请求	RWSN ID 的压缩	保留	帧子 类型	目的地 址方式	保留	源地 址方式

图 29 帧控制字段格式

##### 7.2.2.2.2 帧类型子字段

帧类型子字段长度为 3 比特，设置见表 62。

表 62 帧类型子字段的设置

帧类型值 $b_2 b_1 b_0$	描 述
000	信标
001	数据
010	确认
011	MAC 命令
100~111	保留

#### 7.2.2.2.3 帧待处理子字段

帧待处理子字段长度为 1 比特,若当前帧发送完后设备还有数据发送给接收方,则该子字段设为 1,否则就设为 0(见 7.5.7.3)。

仅当是信标帧,或是在有信标能力 RWSN 中 CAP 时段的发送帧,该子字段才启用。

在其他情况下,发送时将该子字段设为 0,接收时忽略。

#### 7.2.2.2.4 确认请求子字段

确认请求子字段长度为 1 个比特,它定义了当接收设备收到数据帧或 MAC 命令帧时,是否需要回复确认帧。如果设为 1,仅当接收到的帧通过第三层过滤(见 7.5.7.2)时接收设备才发送确认帧;如果设为 0,接收设备将不会发送确认帧。

#### 7.2.2.2.5 RWSN ID 压缩子字段

RWSN ID 压缩子字段长度为 1 比特,它定义了当 MAC 帧中同时含有源地址和目标地址时,却只有一个 RWSN 标识符信息的情况下,是否发送 MAC 帧。如果该子字段为 1,且目的地址和源地址都存在,则帧中将只包含目的 RWSN 标识符信息,且认为源 RWSN 标识符和目的 RWSN 标识符相同。如果为 0,且目的地址和源地址都存在,帧将同时包含目的和源 RWSN 标识符信息。如果仅有一个地址存在,将设为 0,帧将含有和地址相应的 RWSN 标识符。如果帧中不含有地址信息,该子字段设为 0,帧就不含有 RWSN 标识符。

#### 7.2.2.2.6 帧子类型子字段

帧子类型子字段长度为 2 比特,设置见表 63。除数据帧和确认帧之外,其他类型的帧的子类型都为默认子类型,即帧子类型子字段取值为 00,接收时忽略。

表 63 帧子类型子字段的设置

帧子类型值 $b_1 b_0$	描 述
00	默认子类型
01	非默认子类型 1
10	非默认子类型 2
11	非默认子类型 3

## 7.2.2.2.7 目的地址方式子字段

目的地址方式子字段长度为 2 比特,设置(非保留值)见表 64。

如果该子字段为 0,且帧类型子字段也没定义帧是确认帧还是信标帧,源地址方式子字段将为非 0,意味着帧是直接发给 RWSN 协调器的,同时携带有 RWSN 标识符(由源 RWSN 标识符字段定义)。

表 64 目的地址和源地址方式的可能取值

地址方式值 $b_1 b_0$	描 述
00	RWSN 标识和地址字段都不存在
01	保留
10	地址字段包含 16 比特短地址
11	地址字段包含 64 比特扩展地址

## 7.2.2.2.8 源地址方式子字段

源地址方式子字段长度为 2 比特,设置见表 64。

如果该子字段为 0,且帧类型子字段没有定义该帧是确认帧,目的地址方式子字段将为非 0,意味着该帧是 RWSN 协调器发送的,同时包含 RWSN 标识(由目的 RWSN 标识符字段定义)。

## 7.2.2.3 序列号字段

序列号字段长度为 1 个八位位组,它定义了帧的序列标识。

对于信标帧,帧序列号字段定义的是信标帧序列号(BSN)。对于数据帧、确认帧或者 MAC 命令帧,序列号字段将定义的是数据序列号(DSN),用于判断确认帧对应于数据帧或 MAC 命令帧。

## 7.2.2.4 目的 RWSN 标识符字段

如果帧中有目的 RWSN 标识符字段,它有 2 个八位位组,定义了预期接收帧设备的唯一的 RWSN 标识符。如果设为 0xffff,则表示广播 RWSN 标识,所有正在侦听信道的设备将认为这是一个有效的 RWSN 标识符。

仅当帧控制字段的地址方式非 0 时,MAC 帧才包含该字段。

## 7.2.2.5 目的地址字段

如果存在,目的地址字段的长度为 2 或 8 个八位位组,具体长度由帧控制字段(见 7.2.2.2.7)的目的地址方式子字段的值决定,定义了帧接收设备的地址。如果设为 16 比特的 0xffff,表示是广播短地址,所有正在侦听信道的设备认为这是一个有效的 16 比特短地址。

仅当帧控制字段的地址方式子字段非 0 时,MAC 帧才包含该字段。

## 7.2.2.6 源 RWSN 标识符字段

如果存在,源 RWSN 标识字段长度为 2 个八位位组,定义了帧发送端的唯一的 RWSN 标识符。仅当帧控制字段的地址方式子字段非零,且 RWSN ID 压缩子字段为 0 时,MAC 帧才包含该字段。

在关联一个 RWSN 时设备的 RWSN 标识符是确定的,但可以在 RWSN 标识冲突解决过程中发生改变(见 7.5.3.2)。

7.2.2.7 源地址字段

如果存在,源地址字段长度为 2 或 8 个八位位组,具体长度由帧控制字段(见 7.2.2.2.8)的目的地址方式决定,定义了发送端设备的地址。

仅当帧控制字段的源地址方式子字段非 0 时,MAC 帧才包含该字段。

7.2.2.8 帧净荷字段

帧净荷字段长度可变,包括了独立帧类型的相关信息。如果安全使能子字段为 1 时,帧净荷被保护。

7.2.2.9 FCS 字段

FCS 字段长度为 2 个八位位组,包含 16 比特 ITU-T CRC,是通过计算帧头和 MAC 净荷得到的。FCS 是通过 16 位标准生成多项式计算而得:

$$G_{16}(x) = x^{16} + x^{12} + x^5 + 1 \quad \dots\dots\dots(4)$$

传输前需要计算 FCS,算法如下:

- 用多项式  $M(x) = b_0x^{k-1} + b_1x^{k-2} + \dots + b_{l-2}x + b_{l-1}$  表示需要进行运算的比特序列。
- 将  $M(x)$  式乘以  $x^{16}$ ,得到多项式  $x^{16} \times M(x)$ 。
- $x^{16} \times M(x)$  被生成多项式  $G_{16}(x)$  模 2 除,得到余式。
- $R(x) = r_0x^{15} + r_1x^{14} + \dots + r_{14}x + r_{15}$ 。
- FCS 字段取余式  $R(x)$  的各项系数。

其中,二进制多项式就代表字符串,多项式的最高位是字符串的第 1 位。

以一个确认帧为例,其中含有 3 个八位位组的帧头,不含净荷:

0100 0000 0000 0000 0101 0110 [最左边比特位( $b_0$ )首先被传输]

$b_0 \dots\dots\dots b_{23}$

这时,FCS 将按如下进行:

0010 0111 1001 1110

$r_0 \dots\dots\dots r_{15}$  [最左边比特位( $r_0$ )首先被传输]

典型计算过程如图 30 所示。

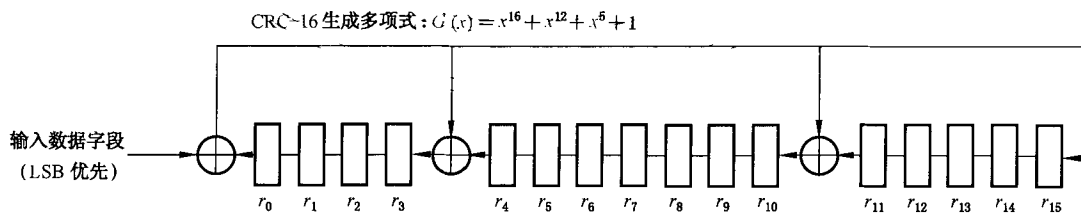


图 30 FCS 的典型计算过程

- a) 将余式寄存器(从  $r_0$  到  $r_{15}$ )初始化为 0;
- b) 按照传输的顺序(LSB 先发送)将帧头和净荷移位;
- c) 当数据字段的最后一位移到除法器,余式寄存器中数值即为 FCS;
- d) FCS 加到数据字段中,故首先发送  $r_0$ 。

7.2.3 4 种类型的帧格式

7.2.3.1 信标帧

7.2.3.1.1 帧格式描述

信标帧格式见图 31 所示。

八位位组:2	1	4/10	0/5/6/ 10/14	2/3	变量	变量	变量	变量	2
帧控制	帧序列号	地址信息	辅助安全头	超帧	SCFP 字段	周期分 配字段	待处理 地址字段	信标负载	FCS
MHR				MAC 净荷					MFR

图 31 信标帧格式

SCFP 字段的格式见图 32,周期分配字段见图 33,待处理地址字段的格式见图 34。

八位位组:1	可变
SCFP 规范	SCFP 列表

图 32 SCFP 字段的格式

八位位组:2	可变
周期分配规范字段	周期分配列表字段

图 33 周期分配字段的格式

八位位组:1	可变
待处理地址定义	地址列表

图 34 待处理地址字段的格式

信标帧各个子字段的顺序和 MAC 帧的一般格式相同。

7.2.3.1.2 信标帧的 MHR 字段

信标帧的 MHR 包括帧控制、帧序列号、源 RWSN 标识符和源地址。

在帧控制字段中,帧类型子字段的设置指示为信标帧(见表 62),源地址方式子字段设为发送信标帧的协调器的地址。如果协调器中有待处理的广播数据帧或命令帧,帧待处理子字段将设为 1。其他子字段将设为 0,接收时忽略。

序列号字段包含了 *macBSN* 的当前值。

地址字段仅包含源地址。源 RWSN 标识符和源地址字段将分别包含信标发送设备的 RWSN 标识符和地址。

### 7.2.3.1.3 超帧规范字段

超帧规范字段长度为 16 比特,格式见图 35。

位:0~2	3~5	6~12	13	14	15
信标顺序	超帧顺序	CAP 的最后 一个时隙	周期分 配指示	RWSN 协调器	关联许可

图 35 超帧规范字段的格式

信标顺序子字段长度为 3 比特,定义信标的传输间隔。在 7.5.2.1 中阐述了信标顺序和信标间隔的关系。

超帧顺序子字段长度为 3 比特,定义超帧处于活跃(例如,接收机处于使能状态)状态的时间,其中包括信标帧发送时间。超帧指数和超帧长度的关系见 7.5.2.1。

CAP 的最后一个时隙子字段长度为 7 比特,定义 CAP 最后一个可用时隙,同时表明,CAP 的长度应大于或等于  $aMinCAPLength$  规定的值。例外情况是,为了维护 SCFP 需要暂时增加信标帧的长度,这会导致 CAP 长度缩短。

周期分配指示子字段长度为 1 比特。如果该子字段设为 1,则在 MAC 负荷中包含周期分配字段。否则设为 0。

RWSN 协调器子字段长度为 1 比特,如果 RWSN 协调器正在发送信标帧,该子字段设为 1,否则设为 0。

关联许可子字段长度为 1 比特,如果参数  $macAssociationPermit$  设为 TRUE(例如,协调器正在接收与 RWSN 关联的请求),该子字段设为 1。如果协调器当前关联请求,就设为 0。

### 7.2.3.1.4 SCFP 字段

SCFP 字段长度为 8 比特,其格式见图 36。

位:0~1	2	3	4	5~7
SCFP 个数	鉴别需求	未经鉴别许可	SCFP 许可	保留

图 36 SCFP 规格字段的格式

SCFP 字段包括:SCFP 个数、鉴别需求、未经鉴别许可、SCFP 许可以及保留字段。其中 SCFP 个数字段由 2 位描述,00、01、10、11 分别代表超帧中有 0、1、2、3 个 SCFP 区域,其他子字段与现有技术类似,不再赘述。

### 7.2.3.1.5 SCFP 列表字段

SCFP 列表字段包括:设备短地址字段、SCFP 标识符字段、起始时隙字段、持续时隙数字段、跳频参数数字段以及数据分片字段,其格式见图 37。

位: 0~15	16~20	21~23	24~25	26~33	34~39	40~55	56~71(选)
设备短地址	工作信道参数 r	保留	SCFP1 标识符	起始时隙	持续时隙数	SCFP2 标识符...	SCFP3 标识符...

图 37 SCFP 描述符的格式

具体描述如下:

- a) 设备短地址字段表明是设备类别;
- b) 工作信道参数,也就是信道选择信息,用于调控节点在第一 SCFP 发送数据帧的信道,该信息由信道选择方案中的跳频算法  $(S+A) \bmod N$  确定;
- c) SCFP 标识符字段表示节点所处的 SCFP,01、10、11 分别代表第 1、2、3 个 SCFP;
- d) 起始时隙字段的描述字节位包含八位二进制数,可描述到第 255 个时隙;
- e) 持续时隙数字段的描述字节包含六位二进制数,是考虑到在第三个 SCFP 中为其他节点分配资源存在着不确定性,增加位数可描述足够多的时隙。

7.2.3.1.6 待处理地址规范字段

待处理地址规范字段的格式见图 38。

位:0~2	3	4~6	7
待处理短地址个数	保留	待处理扩展地址个数	保留

图 38 待处理地址规范字段的格式

待处理短地址的个数有 3 比特,指示了信标帧地址列表中的 16 比特短地址的个数。

待处理扩展地址个数有 3 比特,指示了信标帧地址列表中的 64 比特扩展地址的个数。

7.2.3.1.7 地址列表字段

地址列表字段的大小由信标帧待处理地址规范字段中的数值决定,包含了在协调器中有待处理数据的所有设备的地址列表。地址列表不包括广播短地址 0xffff。

待处理地址最多 7 个,也可能同时包括设备的短地址或扩展地址。列表中扩展地址紧跟在短地址后面。如果协调器能保存 7 个以上的事务,按照先到先服务的原则进行处理,以确保信标中最多包含 7 个地址。

7.2.3.1.8 信标净荷字段

信标净荷字段可选,长度可变,最多可达 *macBeaconPayloadLength* 个八位位组。具体内容由相邻高层定义。*macBeaconPayload* 中包含的八位位组集将复制到该字段中。协调器规定约定信道和备用信道,该信息放在信标帧负载里面,其中类型标识 01 代表约定信道,10 代表备用信道,见图 39。

位:0~4	5~9	10~11	12~15
信道页	信道编号	类型标识符	保留

图 39 信标负载中的约定信道、备用信道信息



## 7.2.3.1.9 周期分配规范字段

周期分配字段长度为 16 比特,其格式见图 40。

位:0~7	8~10	11~15
更新设备数目	信标指数(BO)	保留

图 40 周期分配规范字段

周期分配字段包括更新设备数目、信标指数和周期分配列表。

其中,更新设备数目字段长度为 8 比特,指示本信标帧预计更新其工作周期的设备个数。信标指数字段长度为 3 比特,指示信标指数的大小,其值为 macBeaconOrder 定义的信标间隔 BO,  $0 \leq BO \leq 7$ 。

## 7.2.3.1.10 周期分配列表字段

周期分配列表字段的大小由信标帧中周期分配规范字段的数值决定。周期分配列表包含的是周期分配描述符,每一个描述符代表更新一个设备的工作周期。每个周期分配描述符的有 24 比特,其格式见图 41。

位:0~15	16~23
设备短地址	超帧时长倍数(MSL)

图 41 周期分配描述符

设备短地址子字段长度为 16 比特,是分配到周期分配描述符的设备的短地址。

超帧时长倍数子字段长度为 8 比特,指分配给该设备短地址所对应的设备的工作周期时长是超帧时长  $aBaseSuperframeDuration * 2^{MSL}$  的 MSL 倍,其中  $1 \leq MSL \leq MaxMSL$ 。

## 7.2.3.2 数据帧

## 7.2.3.2.1 帧格式描述

数据帧格式见图 42。

八位位组: 2	1	(见 7.2.3.2.2)	0/5/6/10/14	可变	2
帧控制	序列号	地址字段	保留	数据净荷	FCS
MHR				MAC 净荷	MFR

图 42 数据帧格式

数据帧各字段的顺序和一般的 MAC 帧顺序相同。

## 7.2.3.2.2 数据帧 MHR 字段

数据帧的帧头包括帧控制信息、序列号、目的 RWSN 标识符/地址信息,和/或源 RWSN 标识符/地址信息。

在帧控制字段中,帧类型字段的数值指示为数据帧(见表 62)。

在帧控制字段中,帧子类型字段的数值取值见表 65。

表 65 数据帧子类型字段的设置

帧子类型值 $b_1, b_0$	描述
00	非监测型(默认)数据帧
01	监测型数据帧
10	质疑数据帧
11	更新数据帧

除以下三种场景外,数据帧子类型均设为默认的非监测型数据帧,且需要协调器或设备回发普通确认帧:

- a) 当设备向协调器发送监测型数据时,该数据帧子类型设为监测型数据帧,且需要协调器回发普通确认帧和数据接受确认帧;
- b) 当收到设备发来监测型数据帧的协调器判定接收到的数据不在置信区间内时,协调器向设备发送包含原数据的数据帧,该数据帧子类型设为质疑数据帧,且需要设备回发数据质疑有效确认帧或数据质疑无效确认帧;
- c) 当设备向协调器发送质疑有效确认帧之后,设备需再向协调器发送更新后的数据,该数据帧子类型设为更新数据帧,且需要协调器回发普通确认帧。

序列号的值为 macBSN 当前值。

地址字段包含目的地址字段和/或源地址字段,决定于帧控制字段的设置。

### 7.2.3.2.3 数据净荷字段

数据帧的净荷是邻近高层请求 MAC 层发送的八位位组序列。

### 7.2.3.3 确认帧

#### 7.2.3.3.1 帧格式描述

确认帧格式如图 43 所示。

八位位组:2	1	2
帧控制	序列号	FCS
MHR		MFR

图 43 确认帧格式

确认帧中各字段的顺序和 MAC 帧的一般顺序一致,见图 28。

#### 7.2.3.3.2 确认帧 MHR 字段

确认帧的 MHR 只包括帧控制信息和帧序列号。

在帧控制字段中,帧类型字段的数值指示该帧为确认帧(见表 62)。

在帧控制字段中,帧子类型字段的数值取值见表 66。

表 66 确认帧子类型字段的设置

帧子类型值 $b_1 b_0$	描述
00	普通(默认)确认帧
01	数据接受确认帧
10	数据质疑无效确认帧
11	数据质疑有效确认帧

除以下四种场景之外,确认帧子类型均设为默认的普通确认帧:

- 若收到设备发来监测型数据帧的协调器判定接收到的数据在置信区间内,则协调器向设备发送数据接受确认帧。
- 若协调器收到设备发来的数据质疑无效确认帧,则协调器向设备发送数据接受确认帧。
- 若设备收到协调器发来的质疑数据帧后,与设备上层核实的结果为数据有误,则设备向协调器数据质疑有效确认帧。
- 若设备收到协调器发来的质疑数据帧后,与设备上层核实的结果为数据无误,则设备向协调器数据质疑无效确认帧。

如果是响应已接收的数据请求命令,设备将发送确认帧并判断是否有待处理的数据。如果在发送确认帧之前能做出判断(见 7.5.7.4.3),设备就根据判断来设置帧待处理子字段,否则帧待处理子字段设为 1。如果为了响应数据帧或 MAC 命令帧,设备将帧待处理子字段设为 0。

其他子字段将设为 0,接收时忽略。

如果设备收到需要发送确认帧的帧,确认帧的序列号为接收帧的序列号。

#### 7.2.3.4 MAC 命令帧

##### 7.2.3.4.1 帧格式描述

MAC 命令帧格式参见图 44。

八位位组:2	1	见 7.2.3.4.2	0/5/6/10/14	1	可变	2
帧控制	序列号	地址	保留	命令帧标识	命令净荷	FCS
MHR				MAC 净荷		MFR

图 44 MAC 命令帧格式

MAC 命令帧各个字段的顺序与图 28 所示的 MAC 通用帧格式一致。

##### 7.2.3.4.2 MAC 命令帧 MHR 字段

MAC 命令帧的 MHR 包括帧控制、序列号、目的 RWSN 标识符/地址,和/或源 RWSN 标识符/地址。

在帧控制字段中,帧类型字段的数值指示为 MAC 命令帧(见表 62)。

其他所有子字段的设置则要根据 MAC 命令帧的使用目的。保留字段将设为 0,接收时忽略。

序列号为 macBSN 当前值。

地址字段包含目的地址和/或源地址,这取决于帧控制字段的设置。

7.2.3.4.3 命令帧标识符字段

命令帧标识符字段标识了正在使用的 MAC 命令帧。设置值见表 67 列出的非保留值。

7.2.3.4.4 命令净荷字段

命令净荷字段包含相应的 MAC 命令。每个命令的格式详见 7.3。

7.3 MAC 命令帧

7.3.1 概述

MAC 层定义的命令帧见表 67。RFD 可以传输和接收所有类型的命令帧，例外情况是 SCFP 请求命令。RFD 能传输和接收的命令帧会有一定的限制，见表中有 X 的命令帧。在有信标能力 RWSN 中，只能在 CAP 时段发送 MAC 命令帧，而在不用信标能力 RWSN 中任何时刻都可以。

7.3.1~7.3.7 阐述了 MLME 如何构造各命令帧，MAC 命令帧的接收则要遵循 7.5.7.2 所述规则。

表 67 MAC 命令帧

命令帧标识符	命令名称	RFD		章条号
		Tx	Rx	
0x01	关联请求	X		7.3.2
0x02	关联响应		X	7.3.3
0x03	解关联通知	X	X	7.3.4
0x04	数据请求	X		7.3.5
0x05	RWSN ID 冲突通知	X		7.3.6
0x06	信标请求			7.3.7
0x07	协调器重组		X	7.3.8
0x08	SCFP 请求			7.3.9
0x09	唤醒			7.3.10
0x0a~0xff	保留			—

7.3.2 关联请求命令

7.3.2.1 帧格式描述

关联请求命令允许设备通过 RWSN 协调器或协调器请求与 RWSN 关联。

当未关联设备希望与 RWSN 关联时发送该命令。设备只能通过 RWSN 协调器或协调器与允许关联的 RWSN 关联，并且事先要进行信道扫描。

所有设备都能发送该命令，但是 RFD 不能接收该命令。

管理请求命令的格式见图 45。

八位位组：见 7.2.3.4	1	1
MHR	命令帧标识(见表 67)	性能信息

图 45 关联请求命令格式

### 7.3.2.2 MHR 字段

帧控制字段的源地址方式子字段设为 3(64 比特扩展地址)。目的地址方式子字段与协调器发送的信标帧所指示的方式相同。

帧控制字段的帧待处理子字段设为 0,接收时忽略,确认请求设为 1。

目的 RWSN 标识符字段含有所要关联的 RWSN 标识符。目的地址为发送信标帧的协调器的地址,即关联请求命令的接收设备。源 RWSN 标识符是广播 RWSN 标识符(例如,0xffff)。源地址为 *aExtendedAddress*。

### 7.3.2.3 性能信息字段

性能信息字段格式见图 46。

位:0	1	2	3	4	5	6	7
备用 RWSN 协调器	设备类型	电源	空闲时打开接收机	保留	保留	保留	分配地址

图 46 性能信息字段的格式

备用 RWSN 协调器子字段长度为 1 比特,如果设备能成为 RWSN 协调器,该子字段将设为 1,否则设为 0。

设备类型子字段长度为 1 比特,如果设备是 FFD,该子字段设为 1,否则设为 0,指示是 RFD。

电源子字段长度为 1 比特,如果设备使用交流电,该子字段设为 1,否则设为 0。

空闲时打开接收机子字段长度为 1 比特,如果设备不能在空闲期间关闭接收机以节省能量,该子字段设为 1,否则设为 0。

分配地址子字段长度为 1 比特,如果设备希望协调器在关联后分配一个 16 比特短地址,该子字段设为 1,否则设为 0。

### 7.3.3 关联响应命令

#### 7.3.3.1 帧格式描述

关联响应允许 RWSN 协调器或协调器将关联请求结果发送给请求设备。该命令只能由 RWSN 协调器或协调器发给当前请求关联的设备。

不要求 RFD 能发送该命令,但所有设备都能接收该命令。

关联响应命令的格式见图 47。

八位位组:见 7.2.3.4	1	2	1
MHR	命令帧标识符(见表 67)	短地址	关联状态

图 47 关联响应命令格式

#### 7.3.3.2 MHR 字段

帧控制字段的源地址方式和源地址方式均设为 3(64 比特扩展地址)。

帧控制字段的帧待处理子字段设为 0,接收时忽略,确认请求设为 1。

帧控制字段的 RWSN ID 压缩子字段设为 1。根据 RWSN ID 压缩的值,目的 RWSN 标识包含

*macRWSNId* 的值,忽略源 RWSN 标识。目的地址为请求关联设备的扩展地址。源地址为 *aExtendedAddress*。

### 7.3.3.3 短地址字段

如果协调器无法将请求设备关联到 RWSN,短地址将设为 0xffff,关联状态为失败原因。如果协调器能将该设备关联到 RWSN,该字段为关联后在 RWSN 内通信使用的短地址,直至解关联。

短地址字段为 0xfffe,则表明设备已成功和 RWSN 关联,但还未分配到短地址。这时,设备仍使用 64 比特扩展地址在网内进行通信。

### 7.3.3.4 关联状态字段

关联状态的取值见表 68。

表 68 关联状态的有效值

连接状态	描述
0x00	关联成功
0x01	RWSN 容量饱和
0x02	RWSN 访问被拒绝
0x03~0x7f	保留
0x80~0xff	为 MAC 枚举型参数保留

### 7.3.4 解关联通知命令

#### 7.3.4.1 帧格式描述

RWSN 协调器,协调器或已关联设备都可以发送解关联通知命令。

所有设备都能处理该命令。

解关联通知命令的格式见图 48。

八位位组:见 7.2.3.4	1	1
MHR	命令帧标识(见表 67)	解关联原因

图 48 解关联通知命令格式

#### 7.3.4.2 MHR 字段

根据相应的原语定义的地址方式来设置帧控制字段的地址方式。而源地址方式设为 3(64 比特扩展地址)。

帧控制字段的帧待处理子字段设为 0,接收时忽略,确认请求设为 1。

帧控制字段的 RWSN ID 压缩子字段设为 1。根据该子字段的设置,目的 RWSN 标识符将包含 *macRWSNId* 值,同时忽略源 RWSN 标识。如果协调器希望一个已关联设备离开 RWSN,目的地址为该已关联设备的地址。如果一个已关联设备希望离开 RWSN,目的地址为 *macCoordShortAddress*(目的地址方式为 2)或 *macCoordExtendedAddress*(目的地址方式为 3),源地址字段等于 *aExtendedAddress*。

## 7.3.4.3 解关联原因字段

解关联原因的取值见表 69。

表 69 解关联原因的有效值

解关联原因	描 述
0x00	保留
0x01	协调器希望设备离开 RWSN
0x02	设备希望脱离 RWSN
0x03~0x7f	保留
0x80~0xff	为 MAC 枚举型参数保留

## 7.3.5 数据请求命令

设备发送数据请求命令向 RWSN 协调器或协调器请求数据。

在以下三种情况中会发送该命令。第一种情况是,在有信标能力 RWSN 中,当 *macAutoRequest* 为 TRUE,且协调器的信标帧指使自己有待处理数据,设备将发送该命令。如果设备有待处理数据,协调器就会把设备的地址添加到信标帧的地址列表来指使相应的设备。第二种情况是,收到来自邻近高层的 MLME-POLL.request 原语后,也将发送该命令。第三种情况,收到关联请求命令的确认帧后,设备等待 *macResponseWaitTime* 后,就向协调器发送该命令。

所有设备都能发送该命令。不要求 RFD 能接收该命令。

数据请求命令的格式见图 49。

八位位组:见 7.2.3.4	1
MHR	命令帧标识(见表 67)

图 49 数据请求命令格式

收到的信标帧指示设备有待处理数据时,设备就发送数据请求命令作为响应,如果信标帧的超帧规范字段(见 7.2.3.1.3)显示该信标来自 RWSN 协调器(见 7.2.2.2.7),数据请求命令的目的地址方式设为 0(即表示目的地址信息不存在),否则,就根据目的协调器的特性来设置地址方式。如果数据请求命令包含目的地址信息,则根据 *macCoordShortAddress* 来设置目的地址方式。如果 *macCoordShortAddress* 为 0xfffe,即使用扩展位址,目的地址方式将设为 3,目的地址为 *macCoordExtendedAddress* 的值。否则,将使用短地址,目的地址方式设为 2,目的地址为 *macCoordShortAddress* 的值。

收到的信标帧指示设备有待处理数据时,设备就发送数据请求命令作为响应,其中源地址方式根据待处理地址方式来设置。如果源地址方式为 2,即使用短地址,源地址为 *macShortAddress* 的值。否则,将使用扩展地址,源地址方式设为 3,源地址为 *aExtendedAddress* 的值。

当邻近高层发布 MLME-POLL.request 原语,设备也会发送数据请求命令,这时命令中的目的地址信息由原语规定,并根据 *macShortAddress* 的值来设置源地址方式。如果 *macShortAddress* 小于 0xfffe,就使用短地址,否则使用扩展地址。

如果收到的是关联请求命令的确认帧,设备也会发送数据请求命令,根据接收命令的协调器来设置

目的地址方式。如果 macCoordShortAddress 为 0xfffe,将使用扩展地址,否则将使用短地址。

如果目的地址方式设为 0(即不存在目的地址信息),RWSN ID 压缩子字段将设为 0,源 RWSN 标识设为 macRWSNId 的值。否则 RWSN ID 压缩子字段将设为 1,在这种情况下,目的 RWSN 标识设为 macRWSNId 的值,而源 RWSN 标识将被忽略。

帧控制字段的帧待处理子字段将设为 0,接收时忽略,确认请求设为 1。

### 7.3.6 RWSN ID 冲突通知命令

当检测到 RWSN 标识冲突时,设备会向 RWSN 协调器发送 RWSN ID 冲突通知命令。

所有设备都能发送该命令。不要求 RFD 能接收该命令。

RWSNID 冲突通知命令的格式见图 50。

八位位组:见 7.2.3.4	1
MHR	命令帧标识(见表 67)

图 50 RWSN ID 冲突通知命令格式

帧控制字段的地址方式和源地址方式设为 3(即为 64 比特扩展地址)。

帧控制字段的帧待处理子字段 0,接收时忽略,确认请求设为 1。

帧控制字段的 RWSN ID 压缩子字段将设为 1,相应地,目的 RWSN 标识为 macRWSNId 的值,而源 RWSN 标识将被忽略。目的地址为 macCoordExtendedAddress 的值,源地址为 aExtendedAddress 的值。

### 7.3.7 信标请求命令

#### 7.3.7.1 帧格式描述

在主动扫描期间,设备需要定位 POS 内所有协调器,则应发送信标请求命令。该命令对 RFD 可选。若发送该命令的设备需要协调器重新为其分配工作周期,则该设备需将存储的 MSL 值置 1,即工作周期改为默认时长。

信标请求命令的格式见图 51。

八位位组:7	1	2
MHR	命令帧标识(见表 67)	工作周期特征

图 51 信标请求命令格式

#### 7.3.7.2 MHR 字段

目的地址方式设为 2(即 16 比特短地址),源地址方式子字段设为 2(即 16 比特短地址)。

帧控制字段的帧待处理子字段设为 0,接收时忽略,确认请求子字段和安全使能子字段设为 0。

目的 RWSN 标识为广播 RWSN 标识(即 0xffff)。目的地址为广播短地址(即 0xffff)。

#### 7.3.7.3 工作周期特征字段

工作周期字段格式见图 52。



位:0	1~4	5~7	8~15
重新分配工作周期	保留	信标指数	超帧时长倍数

图 52 工作周期字段的格式

重新分配工作周期字段长度为 1 比特,指示信标请求的发送设备是否需要协调器重新为其分配工作周期。若该字段为 1,则收到信标请求的协调器检查工作周期特征字段的信标指数和超帧时长倍数是否与协调器存储的该设备的相应数值相符;若相符,则协调器在下一个信标帧中按照该数值重新为该设备分配工作周期,并更新该设备的 *NWBSN* 位;若不相符,则协调器通知邻近高层重新计算该设备的 *MSL* 值,并在下一个信标帧中按照新的 *MSL* 值重新为该设备分配工作周期,并更新该设备的 *NWBSN* 位。若该字段为 0,则协调器忽略信标指数字段和超帧时长倍数子字段。

信标指数子字段长度为 3 个比特,指示信标请求的发送设备保存的信标指数的大小,其值为 *macBeaconOrder* 定义的信标间隔 *BO*, $0 \leq BO \leq 7$ 。

超帧时长倍数子字段长度为 8 比特,指该设备目前的工作周期时长是超帧时长的 *MSL* 倍,其中  $1 \leq MSL \leq MaxMSL$ 。超帧时长倍数和信标指数共同定义了设备工作周期的时长,为  $aBaseSuperframeDuration * 2^{BO} * MSL$ 。

### 7.3.8 协调器重组命令

#### 7.3.8.1 帧格式描述

对于 *RWSN* 协调器或协调器,当由于收到原语 *MLME-START.request* 改变了 *RWSN* 配置属性时,就会发送协调器重组命令。

如果 *RWSN* 配置属性(即 *RWSN* 标识、短地址、逻辑信道或信道页)发生改变而发送该命令,则该命令是 *RWSN* 内的广播命令。

所有设备都能接收该命令。并不要求 *RFD* 能发送该命令。

协调器重组命令的格式见图 53。

八位位组:17/18/23/24	1	2	2	1	2	0/1
MHR	命令帧标识 (见表 67)	<i>RWSN</i> 标识符	协调器 短地址	逻辑 信道	短地址	信道页

图 53 协调器重组命令格式

#### 7.3.8.2 MHR 字段

如果是向 *RWSN* 广播,则设为 2(即 16 比特短地址)。源地址方式子字段设为 3(即 64 比特扩展地址)。帧控制字段的帧待处理子字段为 0,接收时忽略。

如果该命令是发给孤立设备,确认请求子字段设为 1,是向 *RWSN* 广播,则设为 0。

如果信道页字段存在,帧版本将设为 0x01,否则将按照表 2 的内容来设置。

目的 *RWSN* 标识符为广播 *RWSN* 标识(例如,0xffff)。如果该命令是发给非孤立设备,目的地址为非孤立设备的广播短地址(例如,0xffff)。源 *RWSN* 标识为 *macRWSNId* 的值,源地址是 *extendedAddress* 的值。

### 7.3.8.3 RWSN 标识符字段

RWSN 标识符是协调器在以后通信中要使用的 RWSN 标识符。

### 7.3.8.4 协调器短地址字段

协调器短地址是 *macShortAddress* 的值。

### 7.3.8.5 逻辑信道字段

该字段的信息是协调器在以后通信中要使用的逻辑信道。

### 7.3.8.6 短地址字段

如果协调器重组命令向 RWSN 广播,短地址将设为 0xffff,接收时忽略。

如果协调器重组命令是发给孤立设备,短地址是设备在 RWSN 内通信所使用的短地址。如果设备没有短地址,使用 64 比特扩展地址,该字段设为 0xfffe。

### 7.3.8.7 信道页字段

如果存在信道页字段,就设为协调器在以后的通信中要使用的信道页。如果新信道页与之前的信道页相同,忽略该字段。

## 7.3.9 SCFP 请求命令

### 7.3.9.1 帧格式描述

当已关联设备要求分配一个新 SCFP 或取消已分配的 SCFP,就发送 SCFP 请求命令,但条件是设备的 16 比特短地址应小于 0xfffe。该命令是可选的。

SCFP 请求命令的格式见图 54。

八位位组:7	1	4
MHR	命令帧标识(见表 67)	SCFP 特征

图 54 SCFP 请求命令格式

### 7.3.9.2 MHR

帧控制域的目的地址模式设为 0(即不存在目的地址信息),源地址模式设为 2(即 16 比特短地址)。帧控制域的帧未处理字段设为 0,接收时忽略,确认请求设为 1。

源 RWSN 标识是 *macRWSNId* 的值,源地址是 *macShortAddress* 的值。

### 7.3.9.3 SCFP 特征

SCFP 特征的格式参见图 55。SCFP 特征域格式见图 56。

位:0~3	4	5	6~9	10~13	14~18	19	20	21~23	24~31
SCFP 长度	SCFP 方向	特征类型	突发长度	数据到达速率	时延要求	分配类型	保留	信标指数	超帧时长倍数

图 55 SCFP 特征格式

位:0~3	4	5	6~9	10~13	14~18	19	20	21~23	24~31
SCFP 长度	SCFP 方向	特征类型	突发长度	数据到 达速率	时延要求	分配类型	保留	信标指数	超帧时 长倍数

图 56 SCFP 特征域格式

SCFP 长度是所请求 SCFP 占据的超帧时隙的个数。

如果请求的 SCFP 只用于接收,SCFP 方向将设为 1。相反,如果只用于发送,将设为 0。设备传输数据帧的方向决定了 SCFP 方向。

如果请求分配 SCFP,特征类型设为 1,而如果是请求取消已分配的 SCFP,则设为 0。

分配类型子字段有 1 个比特,如果 SCFP 请求设备允许与其他设备共享 SCFP,分配类型设为 1。相反,如果不支持共享,则设为 0。如果分配类型设为 0 或 SCFP 方向设为 1,SCFP 请求命令帧不包含突发长度子字段、数据到达速率子字段和时延要求子字段。

突发长度子字段有 4 个比特,是设备在所请求的 SCFP 时隙中将要发送的突发数据的长度(以 bit 为单位)。

数据到达速率子字段有 4 个比特,是设备发送突发数据的速率(以 kbit/s 为单位)。

时延要求子字段有 5 个比特,是设备对在所请求的 SCFP 中传输数据允许的最大时延(以 ms 为单位)。

信标指数子字段长度为 3 个比特,指示信标指数的大小,其值为 macBeaconOrder 定义的信标间隔 BO, $0 \leq BO \leq 7$ 。

超帧时长倍数子字段长度为 8 比特,指该设备的工作周期时长是超帧时长的 MSL 倍,其中  $1 \leq MSL \leq \text{MaxMSL}$ 。超帧时长倍数和信标指数共同定义了设备工作周期的时长,为  $a\text{BaseSuperframeDuration} \times 2^{BO} \times \text{MSL}$ 。

### 7.3.10 唤醒

#### 7.3.10.1 帧格式描述

在 LE 网络中,如 macHWSLSupport 设为 TRUE、macSLWSupport 设为 TRUE 或者 macRFW-Support 设为 TRUE,协调器应能支持发送唤醒帧,同时终端设备应能支持唤醒帧的解析。协调器发送唤醒帧,用于唤醒终端设备进行数据接收。终端设备接收数据的时间可以是立即,或者唤醒帧净荷中指定的时间。

唤醒帧的格式如图 57 所示。

八位位组: 可变	1	6	2
MAC 帧头	命令帧标识(见表 67)	唤醒信息	帧校验

图 57 唤醒帧格式

#### 7.3.10.2 唤醒信息

唤醒信息域携带唤醒帧的信息,格式见图 58。

位: 2	14	16	16
数据帧类型	数据发送剩余时间	协调器地址	唤醒设备地址

图 58 唤醒信息域格式

数据帧类型子字段用于指示发送唤醒帧后,协调器发送的数据帧类型。如发送的帧为单播帧,则此域设为 00,如需要发送的帧为广播帧,则此域设为 01。

数据发送剩余时间子字段用于指示当前时刻到协调器发送数据帧的传输时刻的剩余时间长度,单位为 10 个符号的传输时间长度,此子字段的取值范围为 0x0000~0xffff。由高层设置 AW 唤醒帧中此子字段的数值,如为单播唤醒帧,此域设为 0;广播唤醒序列中的最后一个广播唤醒帧,此域也设为 0。

协调器地址子字段用于指示唤醒帧发送的源地址,非本网络的设备接收到此唤醒帧,查询此域,地址不符,则并不被唤醒。

唤醒设备地址(可以有多个)子字段用于指示协调器需要唤醒的终端设备地址,当且仅当数据帧类型子字段标识此唤醒帧为单播唤醒帧时有效。接收到单播唤醒帧的终端设备,查询此域,若非本设备地址,并不被协调器唤醒。

## 7.4 MAC 常量和 RIB 属性

### 7.4.1 MAC 常量

MAC 层所涉及到的特征常量见表 70。

表 70 MAC 常量

常量	描述	取值
<i>aBaseSlotDuration</i>	当超帧顺序为 0 时,一个超帧时隙包含的符号数(见 7.5.2.1)	60
<i>aBaseSuperframeDuration</i>	当超帧指数等于 0 时,一个超帧包含的符号数	$aBaseSlotDuration * aNumSuperframeSlots$
<i>aExtendedAddress</i>	分配给设备的 64 比特扩展地址	由设备定义
<i>aSCFPDescPersistenceTime</i>	设备侦听其工作信标中是否含有其 SCFP 分配信息的最大次数	4
<i>aMaxBeaconOverhead</i>	MAC 层能在信标帧 MAC 净荷添加的最大八位位组数	75
<i>aMaxBeaconPayloadLength</i>	信标净荷的最大八位位组数	$aMaxPHYPacketSize - aMaxBeaconOverhead$
<i>aMaxLostBeacons</i>	接收设备连续丢失的工作信标个数达到该值,MAC 层就报告和网络失去同步。若收到序列号与设备存储的 NWBSN 值不符,认为丢失该信标	4
<i>aMaxMACSafePayloadSize</i>	未保护的 MAC 帧能容纳的最大 MAC 净荷八位位组数,但要保证 MAC 帧长度不超过 <i>aMaxPHYPacketSize</i>	$aMaxPHYPacketSize - aMaxMPDUUnsecuredOverhead$
<i>aMaxMACPayloadSize</i>	MAC 净荷能容纳的最大八位位组数	$aMaxPHYPacketSize - aMinMPDUOverhead$
<i>aMaxMPDUUnsecuredOverhead</i>	除了安全信息,MAC 层向 PSDU 所能添加的最大八位位组数	25
<i>aMaxSIFSFrameSize</i>	一个 MPDU 的最大八位位组数,在 MPDU 后紧随一个 SIFS	18

表 70 (续)

常量	描述	取值
<i>aMinCAPLength</i>	形成 CAP 的最少符号数。 该参数确保在使用 SCFP 时,MAC 命令能顺利地发送给设备。例外情况是,为了维护 SCFP,需要暂时增加信标帧长度(见 7.2.3.1.4),允许 CAP 长度小于该参数值	440
<i>aMinMPDUOverhead</i>	MAC 能在 PSDU 中添加的最小八位位组数	9
<i>aNumSuperframeSlots</i>	一个超帧的时隙个数	取决于申请发送数据的设备数
<i>aUnitBackoffPeriod</i>	CSMA-CA 算法需要的基本符号数	20
<i>aMaxMSL</i>	信标帧的周期分配字段中,协调器能够分配给设备的超帧时长倍数的最大值	255

7.4.2 MAC RIB 属性

MAC RIB 是管理设备 MAC 层必备的属性。MAC RIB 中所包含的属性见表 71,标有(†)的属性(即只能由 MAC 层来设置)是只读属性,邻近高层只能通过 MLME-GET.request 原语读取。对于其他属性,邻近高层可以分别通过原语 MLME-GET.request 和 MLME-SET.request 进行读写。标有(◆)的属性表示对 RFD 可选,标有(\*)的属性表示对 RFD 和 FED 均是可选。

只读属性 *macAckWaitDuration* 取决于常量的组合和 PHY RIB 属性。常量和属性的关联公式见式(5),属性 *macMaxTotalFrameTxTime* 由邻近高层设置,并取决于 PHY 和 MAC RIB 属性和常量的组合公式见式(6)。

$$macAckWaitDuration = aUnitBackoffPeriod + aTurnaroundTime + phySHRDuration + [6 \times phySymbolsPerOctet] \dots\dots\dots (5)$$

$$macMaxFrameTotalWaitTime = \left[ \left( \sum_{k=0}^{m-1} 2^{macMinBE+k} \right) + (2^{macMaxBE} - 1) \right] \cdot (macMaxCSMABackoffs - m) \times aUnitBackoffPeriod + phyMaxFrameDuration \dots\dots\dots (6)$$

式中:m 是  $\min(macMaxBE - macMinBE, macMaxCSMABackoffs)$ 。

表 71 MAC RIB 属性

属性	标识符	类型	有效范围	描述	默认值
<i>macAckWait-Duration</i> †	0x40	整型	见式(5)	在传输数据帧完成后,确认帧的最大等待时间。该值由所支持的 PHY 决定,即取决于所选的逻辑信道和信道页。该参数值是 ACK 的开始传输时间和 ACK 帧的长度之和,开始传输时间见 7.5.7.4.3	由所支持的 PHY 决定,主要是参数 <i>phy-CurrentPage</i>

表 71 (续)

属性	标识符	类型	有效范围	描述	默认值
<i>macAssociated-RWSN Coord</i>	0x56	布尔型	TRUE 或 FALSE	该参数显示设备是否通过 RWSN 协调器和 RWSN 关联。如果设备已通过 RWSN 协调器和 RWSN 关联,参数为 TRUE;否则为 FALSE	FALSE
<i>macAssociation-Permit</i> ◆	0x41	布尔型	TRUE 或 FALSE	该参数显示协调器当前是否允许关联。如果允许,则参数为 TRUE	FALSE
<i>macAutoRequest</i>	0x42	布尔型	TRUE 或 FALSE	该参数指示了当信标帧中包含设备地址时,设备是否能自动发送数据请求命令。如果可以自动发送数据请求命令,则为 TRUE。	TRUE
<i>macBattLifeExt</i>	0x43	布尔型	TRUE 或 FALSE	该参数指示是否通过减少协调器接收机 CAP 时段的运作时间来延长电源寿命。如果为 TRUE,则使用电池寿命扩展机制。该参数还会影响 CS-MA-CA 算法中的退避指数,见 7.5.2.4	FALSE
<i>macBattLifeExt-Periods</i>	0x44	整型	6~41	在电池寿命扩展方式中,紧随信标的 IFS 后接收机处于使能状态的退避次数,在这段时间,该参数值由所支持的 PHY 决定,是以下三个部分的和: 1:( $2^x-1$ ),式中 x 等于 2,电源寿命扩展方式下 macMinBE 的最大值。即 3 次退避 2:第一个竞争窗口的长度,因此这部分等于 2 次。 3:PHY 前导码和 SFD 的长度加起来,凑成整数个退避时隙	取决于当前选择的 PHY,参数 <i>phyCurrentPage</i> 指示了当前 PHY 的设置
<i>macBeaconPayload</i> ◆	0x45	八位位组集	—	信标净荷的内容	NULL
<i>macBeaconPayloadLength</i> ◆	0x46	整型	0~aMaxBeaconPayloadLength	信标净荷的长度(八位位组数)	0
<i>macBeaconOrder</i> * ◆	0x47	整型	0~7	协调器的信标发送间隔。如果 BO=7,协调器将不发送信标。信标顺序和信标间隔的关系见 7.5.2.1	7

表 71 (续)

属性	标识符	类型	有效范围	描述	默认值
<i>macBeaconTx-Time</i> * ◆	0x48	整型	0x000000 ~0xfffff	设备发送最后一个信标帧的时间,以符号为单位。对于每个信标帧的发送时间,都在帧内同一个符号边缘开始计时,该位置与具体实现有关。 该参数是 24 比特,精度最少为 20 比特,后 4 位是最低位	0x000000
<i>macBSN</i> ◆	0x49	整型	0x00~0xff	加到发送信标帧的帧序列号	在有效范围内的随机数值
<i>macCoordExtendedAddress</i>	0x4a	IEEE 地址	64 比特扩展地址	与设备关联的协调器的 64 比特扩展地址	—
<i>macCoordShort-Address</i>	0x4b	整型	0x0000~0xffff	与设备关联的协调器的 16 比特短地址。0xfffe 表示协调器只使用 64 比特扩展地址。0xffff 表示该值未知	0xffff
<i>macDSN</i>	0x4c	整型	0x00~0xff	发送数据帧或 MAC 命令帧时帧的序列号	在有效范围内的随机数值
<i>macSCFPPermit</i> *	0x4d	布尔型	TRUE 或 FALSE	如果 RWSN 网协调器接受 SCFP 请求,参数为 TRUE,否则为 FALSE。	TRUE
<i>macMaxBE</i>	0x57	整型	3~8	CSMA-CA 算法中退避指数 BE 的最大值。BE 的详细内容见 7.5.2.4	5
<i>macMaxCSMA-Backoffs</i>	0x4e	整型	0~5	CSMA-CA 算法中最大退避次数,如果退避达到该值,则信道竞争失败	4
<i>macMaxFrameTotal-WaitTime</i>	0x58	整型	见式(5)	等待帧的最大时间。帧是数据请求帧的确认帧,或是紧随帧待处理设为 1 的信标后的广播帧。等待时间以符号为单位,在有信标能力 RWSN 中是 CAP 的最大符号数,而在不用信标能力 RWSN 中则是最大的符号数。属性只能由邻近高层来设置,取值由 <i>macMinBE</i> 、 <i>macMaxCSMA-Backoffs</i> 、 <i>macMaxBE</i> 以及每个八位位组的符号数决定	取决于当前选择的 PHY, 参数 <i>phyCurrentPage</i> 指示了当前 PHY 的设置

表 71 (续)

属性	标识符	类型	有效范围	描述	默认值
<i>macMaxFrameRetries</i>	0x59	整型	0~7	传输失败后允许重传的最大次数	3
<i>macMinBE</i>	0x4f	整型	0- <i>macMaxBE</i>	CSMA-CA 算法中 BE 的最小值。详见 7.5.2.4	2
<i>macMinLIFSPeriod</i> *		整型	见第 6 章	构成 LIFS 的最小符号数	取决于当前选择的 PHY, 参数 <i>phyCurrentPage</i> 指示了当前 PHY 的设置
<i>macMinSIFSPeriod</i> *		整型	见第 6 章表 5	构成 SIFS 的最小符号数	取决于当前选择的 PHY, 参数 <i>phyCurrentPage</i> 指示了当前 PHY 的设置
<i>macRWSNId</i>	0x50	整型	0x0000~0xffff	设备所在 RWSN 的 16 比特标识符。如果为 0xffff, 则表示该设备未关联	0xffff
<i>macPromiscuousMode</i> ◆	0x51	布尔型	TRUE 或 FALSE	指示 MAC 层是否处于混合方式。如果为 TRUE, 说明 MAC 层将接收所有 PHY 收到的帧	FALSE
<i>macResponseWaitTime</i>	0x5a	整型	2~64	在发送请求命令帧后, 设备就等待有效的响应命令帧。最大等待时间是 <i>aBaseSuperframeDuration</i> 的倍数	32
<i>macRxOnWhen-Idle</i>	0x52	布尔型	TRUE 或 FALSE	该参数表示在空闲期间 MAC 层是否需要打开接收机。在有信标能力 RWSN, 该参数只应用于流入超帧的 CAP 内	FALSE
<i>macShortAddress</i>	0x53	整型	0x0000~0xffff	设备在 RWSN 内通信用的 16 比特短地址。如果设备是 RWSN 协调器, 该值在建立 RWSN 前就已确定。否则是在关联时由协调器分配。0xfffe 表示设备已关联, 但未分配短地址, 0xffff 表示设备没有短地址	0xffff



表 71 (续)

属性	标识符	类型	有效范围	描述	默认值
<i>macSuperframeOrder</i> * ◆	0x54	整型	0~7	该参数定义了流出超帧活跃部分的长度(包括信标帧)。如果 $SO = 7$ , 超帧中的活跃部分只有信标。超帧指数和超帧长度的关系见 7.5.3.1.2	7
<i>macSyncSymbolOffset</i> *	0x5b	整型	0x000~0x100	MLME 在某符号边界发送或接收帧的时间, 和第一个经过 SFD 的符号时间之间的差值, 以符号为单位	由实际情况决定
<i>macTimestampSupported</i> *	0x5c	布尔型	TRUE 或 FALSE	该参数表示 MAC 层是否支持流入和流出数据帧的可选时间戳特性	由实际情况决定
<i>macTransactionPersistenceTime</i> ◆	0x55	整型	0x0000~0xffff	协调器存储一个事务和并在信标中指示的最大时间, 以单位阶段的 <i>macMaxMSL</i> 估计。这个单位阶段取决于参数 <i>macBeaconOrder</i> , $BO$ 。当 $0 \leq BO \leq 6$ , 单位阶段是 $aBaseSuperframeDuration * 2^{BO}$ 。当 $BO = 7$ , 单位阶段是 $aBaseSuperframeDuration$	0x01f4
<i>macMLS</i>	0x5d	整型	0~255	表示设备目前的工作周期是超帧时长的 <i>macMLS</i> 倍。在设备接收到携带设备的工作周期分配信息的信标帧后, 该值将被更新	
<i>macNWBSN</i>	0x5f	整型	0~255	设备下一个工作信标帧的序列号。在设备接收到该设备的工作信标后, 该值将被更新	
<i>WorkingPeriodSpec</i>	0x60	位图	见 7.2.3.1.9	周期分配规范	
<i>WorkingPeriodList</i>	0x61	位图	见 7.2.3.1.10	周期分配列表	
<i>macAWPeriod</i>	0x60	整型	0x000~0xffff	终端设备周期信道采样侦听的周期, 单位为 10 个符号的传输时间	0
<i>macHWSLSupport</i>	0x61	布尔型	TRUE 或 FALSE	标识 LE 网络中的设备是否支持 HWSL 唤醒机制, TRUE 为支持	TRUE

表 71 (续)

属性	标识符	类型	有效范围	描述	默认值
<i>macHWSLWakeupInterval</i>	0x62	整型	$0 \sim aMaxPHYPacketSize$	HWSL 唤醒序列中, 连续两个唤醒帧之间侦听信道的时间长度, 单位为 1 个字节的传输时间	0
<i>macHWSLMaxInterval</i>	0x63	整型	$macAWPeriod \sim 0xffff$	HWSL 唤醒序列的最长持续时间, 单位为 10 个符号的传输时间	macAWPeriod
<i>macLEenabled</i>	0x64	布尔型	TRUE 或 FALSE	标识网络是否支持低功耗模式, TRUE 为支持	TRUE
<i>macRFWminpulsewidth</i>	0x65	整型	$0 \sim aMaxPHYPacketSize$	RFW 脉冲序列中每一个脉冲的最小宽度, 单位为 1 个字节的传输时间	0
<i>macRFWpulseInterval</i>	0x66	整型	$0 \sim aMaxSIFSFrameSize$	RFW 脉冲序列中, 连续两个脉冲之间时间长度, 单位为 1 个字节的传输时间	0
<i>macRFWwakeupfactor-num</i>	0x67	整型	0x000~0xff	RWSN 中唤醒因子个数, 由中心管理机为每个 RWSN 分配并由协调器下发, 确定预唤醒设备所属的 RWSN 网络, 不同 RWSN 具有不同的预设唤醒因子	0
<i>macRFWSupport</i>	0x68	布尔型	TRUE 或 FALSE	标识 LE 网络中的设备是否支持 RFW 唤醒机制, TRUE 为支持	TRUE
<i>macSLWSupport</i>	0x69	布尔型	TRUE 或 FALSE	标识 LE 网络中的设备是否支持 SLW 唤醒机制, TRUE 为支持	TRUE
<i>macDataAckWaitDuration</i>	0x70	整型	0x0000~0xffff	设备向协调器发送的监测型数据帧完成后, 数据接受确认帧的最大等待时间。该值为 $aBaseSuperframeDuration * 2^{BO}$ , 其中 $0 \leq BO \leq 6$ ; 当 $BO = 7$ , 等同于 $BO = 0$ 时的取值	0x01f4

## 7.5 MAC 功能概述

### 7.5.1 概述

本条详细描述 MAC 的功能。在 7.5.2 中描述两种信道访问机制:基于竞争的访问机制和非竞争的访问机制,前者是设备以完全分布式的形式利用 CSMA-CA 退避算法来访问信道,而后者是由 RWSN 协调器分配时隙来实现。

在 7.5.3 中描述如何生成并维护一个 RWSN 的机制。设备通过扫描信道获得信道当前的状况,并在自身的个人操作空间(POS)内定位所有的信标帧,或定位一个自己失去同步的信标。在生成一个新的 RWSN 前,先利用信道扫描的结果选择一个恰当的逻辑信道,以及未被其他 RWSN 使用的 RWSN ID。由于具有相同 RWSN ID 两个 RWSN 的 POS 会重叠,所以应有相应的程序来解决这一问题。在扫描信道并选择恰当的 RWSN ID 后,由一个全功能设备(FFD)作为 RWSN 协调器。

在 7.5.4 中描述设备和 RWSN 的关联或解关联机制。

在 7.5.5 中描述设备如何和协调器同步。

在 7.5.6 中描述协调器如何控制多个数据收发。

在 7.5.7 中描述发送、接收以及确认帧的机制,其中包括如何间接地发送和重传。

### 7.5.2 信道的访问

#### 7.5.2.1 超帧结构

##### 7.5.2.1.1 概述

RWSN 中的协调器能够利用超帧结构来限制带宽。协调器发送的信标帧规定了超帧的结构,它由活跃期和非活跃期两部分组成。在非活跃期间,除了已经申请了需要在非活跃期进行传输的能源非受限设备,其他设备进入休眠状态以省能量。

MAC 的两个 RIB 属性 *macBeaconOrder* 和 *macSuperframeOrder* 描述了超帧的结构。前者描述了协调器发送信标帧的间隔,后者描述了超帧活跃期的持续时间(包括信标帧的长度)。*macBeaconOrder* 的值 BO 和与信标帧发送间隔 BI 的关系为:当  $0 \leq BO \leq 6$ ,  $BI = aBaseSuperframeDuration * 2^{BO}$  符号。如果  $BO = 7$ ,协调器将不会发送信标帧(除非被要求发送信标帧,如:协调器收到信标帧请求命令),且忽略 *macSuperframeOrder* 的值。

MAC RIB 属性 *macSuperframeOrder* 描述了超帧活跃期的长度(包括信标帧)。*macSuperframeOrder* 的值,即 SO,与超帧持续时间 SD 的关系为:当  $0 \leq SO \leq BO \leq 6$ ,  $SD = aBaseSuperframeDuration * 2^{SO}$  符号。如果  $SO = 7$ ,紧跟在信标帧后的超帧活跃期长度为 0。如果  $BO = 7$ ,就不存在超帧(*macSuperframeOrder* 的值被忽略),*macRxOnWhenIdle* 的值将指示当设备不进行收发时接收机是否处于使能状态。

每个超帧的活跃期有三部分组成:信标帧、竞争时段(CAP)和非竞争时段(CFP),它们被划分成 *aNumSuperframeSlots* 个等长的时隙,每个时隙的持续时间为  $2^{SO} * aBaseSlotDuration$ 。信标帧的传输在时隙 0 的起始处,且不使用 CSMA 算法,CAP 阶段紧跟在信标帧后。时隙 0 的起始定义为信标帧 PPDU 的第一个符号被传输的那一刻。如果超帧中存在 CFP,它紧跟在 CAP 后面,并持续到超帧活跃期结束。CFP 由三个 SCFP 组成,即:SCFP1、SCFP2 和 SCFP3

MAC 层应保证超帧计时的完整性,例如要补偿时钟的偏移。

在使用超帧的 RWSN 中,*macBeaconOrder* 的取值为  $0 \sim 6$ ,*macSuperframeOrder* 的取值为  $0 \sim macBeaconOrder$ 。

7.5.2.1.2 CAP

CAP 紧随信标帧,并在 CFP 开始前的时隙边缘结束。如果没有 CFP,CAP 持续到超帧的活跃期结束。CAP 的最小长度为  $aMinCAPLength$ ,特殊情况是为了维护 SCFP(见 7.2.3.1.4)需要增加信标帧的长度,相应地,应减少 CAP 的长度,而且能根据 CFP 的大小自适应地增加或减少 CAP 的长度。

除了确认帧和紧随数据请求命令的确认帧后发送的数据帧,在 CAP 阶段发送任何帧都应采用时隙的 CSMA-CA 机制,设备应确保在 CAP 结束前的一个 IFS 之前完成事务的处理(包括所有确认帧的接收),当无法实现时,设备就将发送推迟到下一个工作超帧的 CAP 阶段。

MAC 命令帧一般是在 CAP 阶段发送。

7.5.2.1.3 CFP

超帧中规定 CFP 应跟在 CAP 后面,并在超帧结束前的一个 IFS 时段(见 7.5.2.3)前完成收发操作。在 CFP 时段,RWSN 协调器根据 RWSN 网络中设备申请时隙的情况以及网络情况,将 CFP 划分成三个 SCFP,即:SCFP1、SCFP2 和 SCFP3。其中,SCFP1 用于节点在协调器初始分配的时隙发送数据,SCFP2 用于节点重新发送在 SCFP1 发送失败的数据,SCFP3 用于继续传输节点在该第一 SCFP 或该第二 SCFP 未完全发送的数据,包括重新发送在 SCFP2 发送失败的数据。

在 CFP 中收发数据无需使用 CSMA-CA 算法来竞争信道。但设备应确保在自己的 SCFP 结束前的一个 IFS 时段(见 7.5.2.3)前完成收发操作。

7.5.2.2 流入超帧和流出超帧的计时

在信标使能的 RWSN 中,不是 RWSN 协调器的协调器应同时维护它的 RWSN 协调器发送信标的超帧(流入超帧)的计时,以及自己信标的超帧(流出超帧)的计时。MLME-START.request 原语中的参数 StartTime 定义了这两种超帧计时的相关信息。两种帧的联系如图 59 所示。

在 RWSN 中,所有超帧的信标帧顺序和超帧顺序是相等的。所有设备只能在超帧的活跃期和 RWSN 联系。

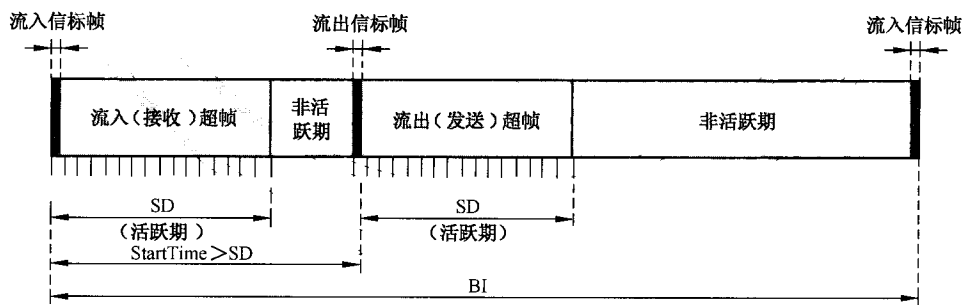
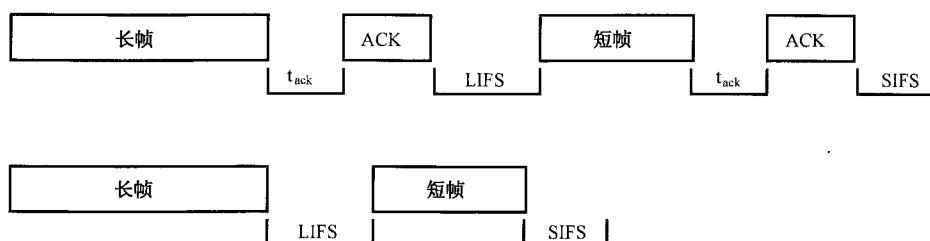


图 59 流入信标帧和流出信标帧的联系图

7.5.2.3 帧间间隔 IFS

MAC 层需要一定的时间来处理 PHY 收到的数据,所以在连续发送两个帧时,至少要隔开一个 IFS (帧间间隔)时段;当邻近帧需要确认时,确认帧和下一要发送的帧也至少要隔开一个 IFS。IFS 的大小取决于发送的邻近帧的长度。当帧(如 MPDU)的长度达到  $aMaxSIFSFrameSize$  个八位位组时,跟随着的帧间间隔为一个 SIFS,且长度至少是  $aMinSIFSPeriod$ ;而帧长度大于  $aMaxSIFSFrameSize$  时,帧间间隔为一个 LIFS,最小长度是  $aMinLIFSPeriod$ 。具体如图 60 所示。



注:  $aTurnaroundTime \leq t_{ack} \leq (aTurnaroundTime + aUnitBackoffPeriod)$

图 60 帧间间隔 IFS

#### 7.5.2.4 CSMA-CA 算法

在 CAP 期间传输数据帧或 MAC 命令帧时,要采用 CSMA-CA 算法来竞争信道,但当帧能够被快速发送并得到及时确认,则可以不采用这一访问机制。在有信标能力 RWSN 中的信标帧、响应帧以及 CFP 期间发送的数据帧均不使用 CSMA-CA 算法,可以直接访问信道。

如果在某个 RWSN 中使用了周期的信标帧,在 CAP 期间进行数据收发时应采用时隙的 CSMA-CA 算法;实现算法时以退避时段为单位,退避时段定义为一个  $aUnitBackoffPeriod$  符号。

在基于时隙 CSMA-CA 的 RWSN 中,每个设备的退避时间的起始应和协调器的超帧一致,即设备开始退避和协调器发送信标帧应同时开始。此外,MAC 层应确保在退避开始时,PHY 也开始传输数据。

在 CSMA-CA 算法中,每个设备需要维护三个重要的参数:NB(退避次数)、CW(竞争窗口,仅用于时隙 CSMA-CA)、BE(退避指数)。NB 是退避次数,在发起新的数据传输时初始化为 0。CW 是竞争窗口长度,仅用于时隙 CSMA-CA,是每次传输开始前感知信道状态的退避阶段,初始化值是 2,每次感知信道忙时,重新设为 2。BE 是退避指数,和设备在访问信道前需等待的退避次数有关。需要注意的是,如果  $macMinBE$  设为 0,在第一次执行算法时是无法避免冲突的。

即使在算法的 CCA 阶段接收端处于使能状态,设备也会丢弃在此期间接收到的任何帧。

图 61 是时隙 CSMA-CA 算法的流程图。

在步骤 1:MAC 层首先初始化 NB、CW、BE,并定位于退避起始边界。

在步骤 2:MAC 层推迟发送数据,并进入退避状态,根据退避指数 BE 随机选择退避周期  $X_B$ ,其中,  $X_B = X \times aUnitBackoffPeriod$ , X 为在  $0 \sim (2^{BE} - 1)$  中随机选择的整数。

在步骤 3:如果  $X \geq 4$ ,计算中间退避期  $M_B = (X_B) \times (MP) / 100$ ,其中 MP 为单元中间期,是一个与 X 取值有关的随机值。当  $4 \leq X \leq 10$  时,MP 的值在 30、40、50 和 60 中随机选取,当  $11 \leq X \leq 31$  时,MP 的值在 10、20、30 或 40 中随机选取;确定中间退避期  $M_B$  并退避完成后在其边界处执行 CCA,这个过程中只执行一次 CCA,若信道是空闲则进行数据传输,即剩余的退避时间不执行;若信道非空闲则继续退避到退避周期  $X_B$  结束时进入步骤 4 执行 CCA,此处 CCA 次数由 CW 值确定。

如果  $X < 4$  时,直接退避,进入步骤 4,在退避边界执行 CCA。

在步骤 5:判断信道是否空闲,若信道空闲,则执行  $CW = CW - 1$  操作,直到 CW 为 0 时接入信道成功,若 CW 不为 0 时,返回到步骤 4 进行 CCA;如果信道非空闲,则判断 CW 是否为 1。如果 CW 为 1,则  $CW = 2$ ,  $NB = NB + 1$ ,  $BE = 1$ ,使其在下次接入信道的时候的退避变小,接入信道概率增大;如果 CW 不为 1,则  $CW = 2$ ,  $NB = NB + 1$ ,  $BE = \min(BE + 1, aMaxBE)$ 。

在步骤 6:对退避次数 NB 进行判定,如果退避次数 NB 大于 MAC RIB 属性  $macMaxCSMABackoffs$  的值,直接退出算法;如果 NB 没有超过  $macMaxCSMABackoffs$  的大小,就返回到步骤 2 继续以上步骤。

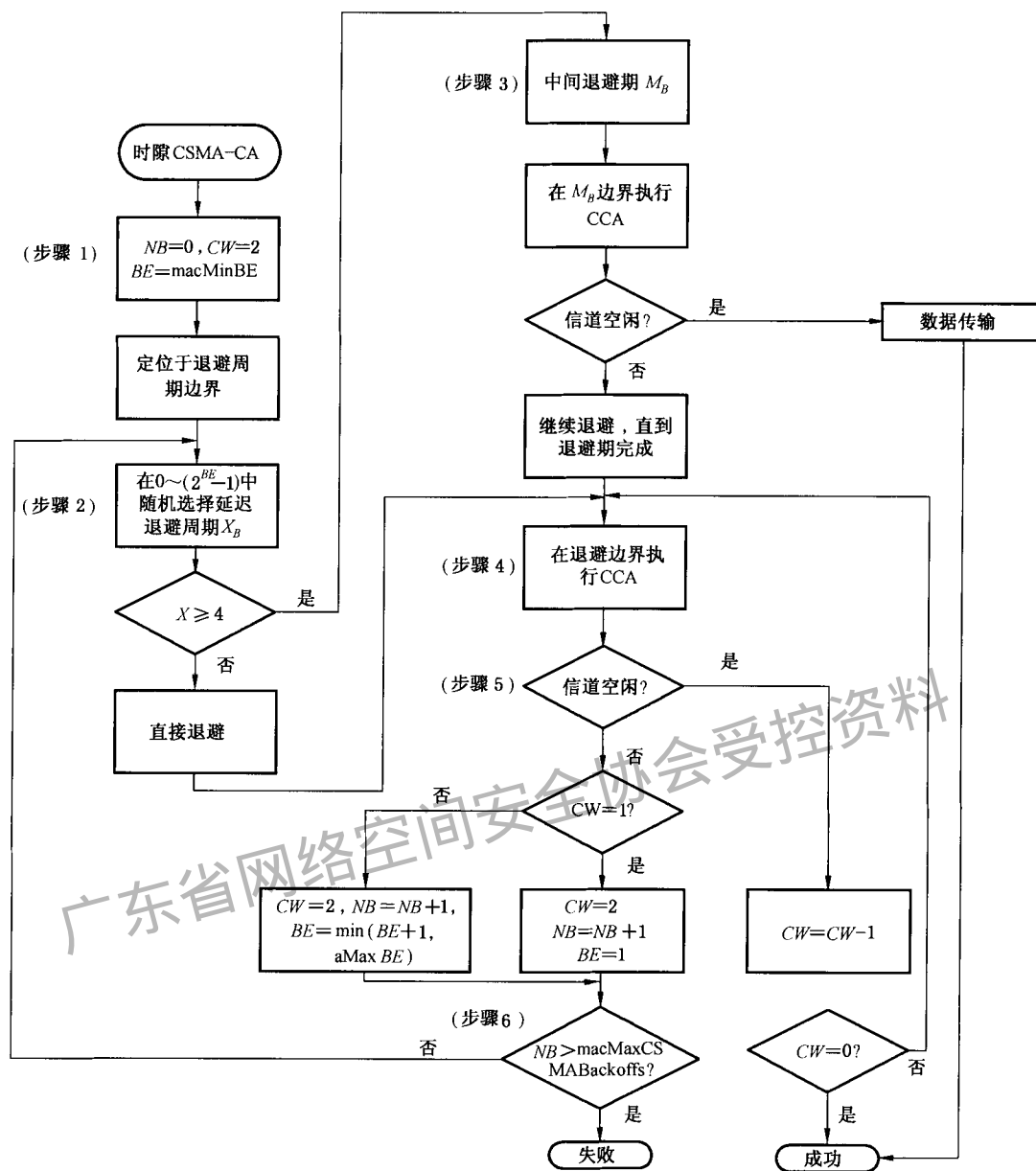


图 61 时隙 CSMA-CA 的流程图

### 7.5.2.5 信道分配算法

信道分配算法流程如图 62 所示。

在信道分配算法中,协调器维护三个重要的参数:当前超帧的时间参数(S)、设备的标识(A)、设备在跳频模式下的工作信道数量(N)。S是当前超帧随机而且唯一的时间参数,对于某一个网络而言,网络协调器初始广播的帧序列号作为时间参数。A是节点标识,对于同一个网络而言,每个设备的地址作为该设备的标识是随机而且唯一的。因为在当前超帧内,协调器只会为当前超帧的工作设备分配 SCFP 时隙,所以协调器只为当前超帧的工作设备分配工作信道,信道分配算法中所说的节点设备均指当前超帧的工作设备。N是节点在跳频模式下的除备用信道和约定信道外的其他工作信道的数量。

协调器根据设备在跳频模式下的工作信道的数量、设备的标识以及对应于当前超帧的时间参数,由

公式 $(S+A)\text{mod } N$  计算设备的工作信道标号,在当前超帧内,设备在对应的工作信道上执行数据传输。

判决条件如下:

- 若协调器判断工作信道标号大于等于约定信道标号或备用信道标号中任一项,则将工作信道标号累加预设数值,直到经过累加的工作信道标号不等于约定信道的标号或备用信道的标号,将经过累加的工作信道标号对应的信道确定为工作信道;
- 若协调器判断工作信道标号大于等于约定信道标号和备用信道标号,则将工作信道标号累加两倍的预设数值,直到经过累加的工作信道标号不等于约定信道的标号或备用信道的标号,将经过两倍累加的工作信道标号对应的信道确定为工作信道;
- 若协调器判断工作信道标号小于约定信道标号和备用信道标号,则将节点的工作信道分配为相应的工作信道标号对应的工作信道。

其中预设数值为网络中相邻的工作信道标号之间的差值。

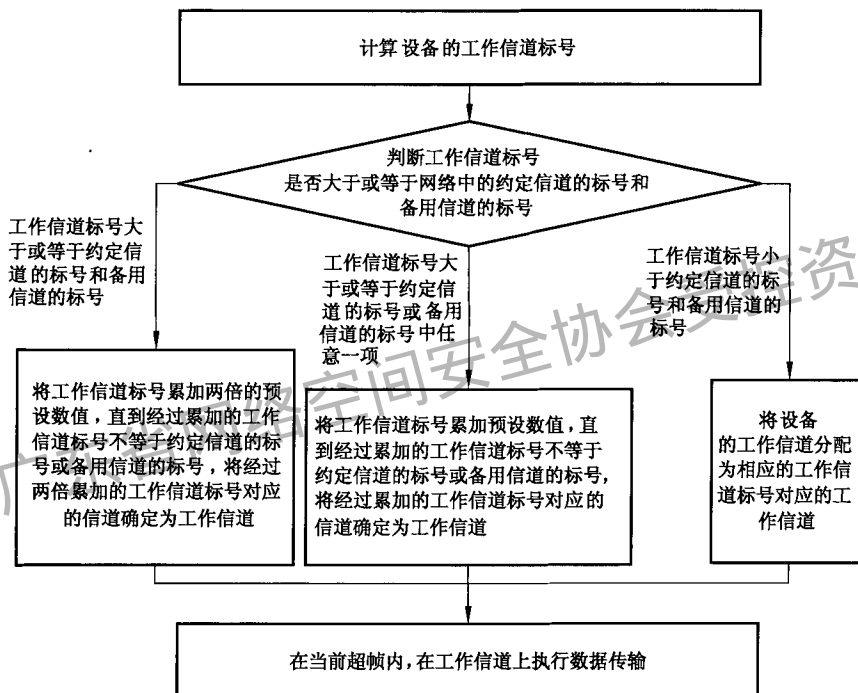


图 62 约定信道和备用信道属于工作信道情况下的信道分配算法

同一超帧的 CFP 被划分为 3 个 SCFP,即 SCFP1、SCFP2 和 SCFP3,SCFP2 用于重新传输节点在 SCFP1 发送失败的数据,SCFP3 用于继续传输节点在该第一 SCFP 或该第二 SCFP 未完全发送的数据,包括重新发送在 SCFP2 发送失败的数据,且每个节点的 SCFP1 中的时隙顺序和 SCFP2 的时隙顺序相同,即相对于一个节点,SCFP1 中的各时隙顺序和 SCFP2 中的各时隙顺序相同,节点在 SCFP1 的某个时隙发送数据失败,则仍在 SCFP2 中的该某个时隙再次发送数据。由于在 SCFP1 和 SCFP2 中使用相同的时隙重发数据,协调器在不同时隙接收到数据时可根据时隙判断该数据的源节点。

若节点在 SCFP1 内发送数据失败,节点并不立即重传,而是根据接收到的协调器广播的信道划分信息,跳至信道质量较好的约定信道,在 SCFP2 对应时隙再次发送该数据。如果仍发送失败,在 SCFP3 再次发送在 SCFP2 中发送失败的数据,节点发送数据失败,有两种情况:一是数据包丢失,二是确认帧丢失,如图 63 非竞争期的重传时隙选择机制所示。

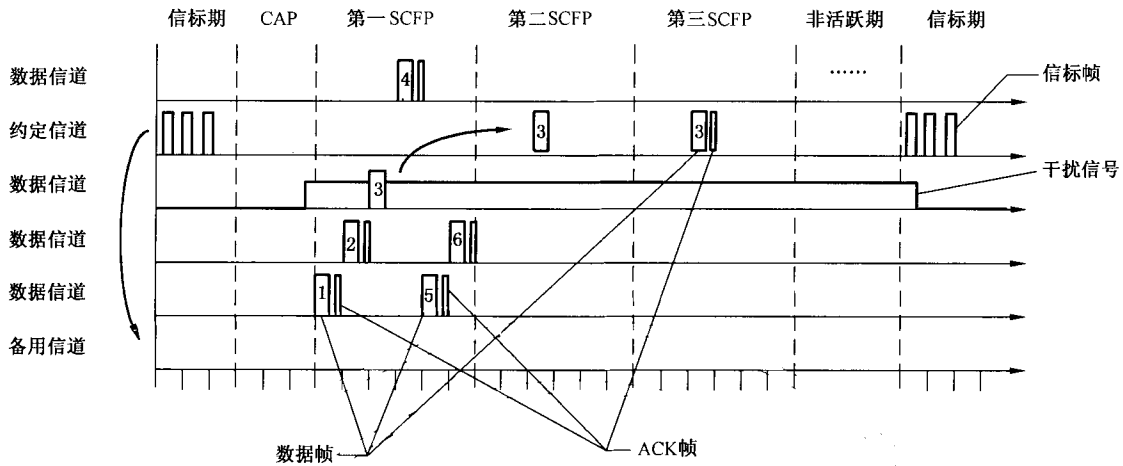


图 63 非竞争期的重传时隙选择机制

### 7.5.3 RWSN 的建立和维护

#### 7.5.3.1 信道的扫描

##### 7.5.3.1.1 概述

所有的设备都能够对指定的信道进行被动扫描和孤立节点扫描,而 FFD 设备还能进行信道能量检测和主动信道扫描。参数 *phyChannelsSupported* 指明了设备能扫描的信道。

原语 MLME-SCAN.request 产生后,设备就按信道的编号从小到大扫描信道。在扫描期间,设备应推迟发送信标帧,并且只能接收通过 PHY 数据服务得到的帧,一般这些帧和正在执行的扫描操作有关。得到扫描结果后,设备就重新发送信标帧,并通过 MLME-SCAN.confirm 原语返回扫描结果。

##### 7.5.3.1.2 信道能量检测(ED)扫描

设备对每个指定的信道进行 ED 扫描,就能获得该信道的峰值能量。在产生一个新的 RWSN 前,可能成为 RWSN 协调器的设备会指定几个信道,所有设备通过能量检测获得这些信道的峰值能量。扫描期间 MAC 层丢弃所有通过 PHY 数据服务接收的帧。

对指定的逻辑信道进行 ED 扫描是由原语 MLME-SCAN.request 发起的,将参数 ScanType 设置为 ED 扫描。MLME 首先通过设置 *phyCurrentChannel* 和 *phyChannelPage* 切换到该逻辑信道上,然后反复进行 ED 度量,持续时间是  $[aBaseSuperframeDuration * (2^n + 1)]$ ,其中  $n$  是原语中参数 ScanDuration 的值。MLME 发布原语 PLME-ED.request 来实现 ED 度量,并返回一个有效值。在进入下一个信道扫描时,应保存扫描过程中获得的最大 ED 度量值,设备的存储个数的有效范围是 1 至指定扫描的信道个数。

当存储的数目等于要求的信道数或者已经完成每个指定逻辑信道的能量度量,停止 ED 扫描。

##### 7.5.3.1.3 主动信道扫描

设备通过主动信道扫描,能够搜索到自己 POS 内可以发送信标帧的所有协调器。

预期的 RWSN 协调器可利用这一结果在建立新的 RWSN 前选择一个 RWSN ID,普通设备则利用扫描结果进行关联。

扫描期间 MAC 层只接收通过 PHY 数据服务接收到的信标帧。当设备收到信标帧,且帧中为处理



地址列表中有自己的地址,设备就不能提取待处理数据。

在主动扫描前,MAC层应存储 *macRWSNID* 的属性值,并在扫描期间设为 0xffff 以便接收机滤波器接收所有的信标帧,而不局限于设备所在 RWSN 的信标帧。扫描结束后,*macRWSNID* 恢复扫描前的设置。

将原语 MLME-SCAN.request 的参数 *ScanType* 设置为主动扫描,就能对指定的逻辑信道进行主动扫描。扫描过程:首先通过设置 *phyCurrentChannel* 和 *phyCurrentPage* 的属性切换到相应的逻辑信道,并发送信标帧请求命令;发送成功后,设备就打开接收机,使能状态的持续时间不能大于规定的最大时间,即 $[aBaseSuperframeDuration * (2^n + 1)]$ ,其中  $n$  是参数 *ScanDuration* 的属性值,即扫描的持续时间。扫描过程中,设备只接收信标帧,并记录 RWSN 描述符结构(见表 39 中信标帧的信息。在参数 *macAutoRequest* 设置为 TRUE 的前提下,当收到信标帧,MAC 层就存储 RWSN 描述符结构列表,直至扫描结束,再利用原语 MLME-SCAN.confirm 参数 *RWSNDescriptor* 把它递交给邻近高层。设备的存储能力的有效范围是 1 至规定的 RWSN 描述符的最大个数。如果接收到的信标帧含有一个 RWSN 标识符和一个源地址,且在当前扫描信道中还没找到该地址,就认为该信标帧是唯一的。相反,当参数 *macAutoRequest* 是 FALSE 时,接收到信标帧后,就通过独立原语 MLME-BEACON-NOTIFY.indication 把每一个保存的 RWSN 描述符递交给邻近高层。当接收的信标帧含有 1 个或多个八位位组的净荷时,设备也会执行如上的操作。

在有信标能力 RWSN 中,协调器收到的信标请求命令后,会解析信标请求命令的内容,但不发送单独的信标帧,而继续发送周期性信标帧。而在不用信标能力 RWSN 中,协调器就会采用非时隙的 CSMA-CA 算法发送一个单独的信标帧。

如果 *macAutoRequest* 是 TRUE,当发现的信标帧个数等于规定的最大值(即设备一次能够进行扫描的信道的最大个数),或是对某信道的扫描时间达到了规定的扫描持续时间,就中止对该信道的主动扫描。而如果 *macAutoRequest* 是 FALSE,只有在达到扫描时间后,才能中止对某一特定信道的主动扫描,相反,就认为该信道还没有进行扫描。

当 *macAutoRequest* 是 TRUE 时,保存的 RWSN 描述符的数目等于规定的最大值(即设备一次能够进行扫描的信道的最大个数),或是指定的可用信道均已扫描完毕,整个主动信道扫描程序才会中止。如果 *macAutoRequest* 是 FALSE,只有在对每个信道进行主动扫描后,才会结束主动扫描程序。

#### 7.5.3.1.4 被动信道扫描

和主动信道扫描一样,设备通过被动信道扫描搜索自己通信范围内可以发送信标帧的所有协调器。不同的是,设备并不发送信标帧请求命令。在关联之前设备可以使用被动信道扫描。

在扫描期间,MAC 层只接收通过 PHY 数据服务得到的信标帧。如果扫描设备发现信标帧的待处理地址列表中含有自己的地址,就不会提取待处理数据。

在被动扫描前,MAC 层应存储 *macRWSNID* 的属性值,并在扫描期间设为 0xffff 以便接收机滤波器接收所有的信标帧,而不局限于设备所在 RWSN 的信标帧(见 7.5.7.2)。扫描结束后,*macRWSNID* 恢复扫描前的设置。

将原语 MLME-SCAN.request 的参数 *ScanType* 设置为被动扫描,设备就能对指定的逻辑信道进行被动扫描。扫描过程:设备首先通过设置 *phyCurrentChannel* 和 *phyCurrentPage* 的属性切换到相应的逻辑信道,然后打开接收机,且使能状态的持续时间不能大于规定的最大时间,即 $[aBaseSuperframeDuration * (2^n + 1)]$ ,其中  $n$  是参数 *ScanDuration* 的属性值,即扫描的持续时间。扫描过程中,设备丢弃收到的非信标帧,并记录 RWSN 描述符结构(见 7.1.13.3.1)中所有特定信标帧的信息。在参数 *macAutoRequest* 设置为 TRUE 的前提下,当收到信标帧,MAC 层就存储 RWSN 描述符结构列表,直至扫描结束后,再利用原语 MLME-SCAN.confirm 中的参数 *RWSNDescriptor* 把它递交给邻近高层。设备的存储能力的有效范围是 1 至规定的 RWSN 描述符的最大个数。如果接收到的信标帧含

有一个 RWSN 标识符和一个源地址,且在当前扫描信道中还没找到该地址,就认为该信标帧是唯一的。相反,当参数 *macAutoRequest* 是 FALSE 时,接收到信标帧后,设备就通过独立原语 MLME-BEACON-NOTIFY.indication 把每一个保存的 RWSN 描述符递交给邻近高层。扫描一结束,就向邻近高层发布原语 MLME-SCAN.confirm,返回一个无效的 *RWSNDescriptorList*。当接收的信标帧含有 1 个或多个八位位组的净荷时,就发布原语 MLME-BEACON-NOTIFY.indication 向邻近高层递交 RWSN 描述符。

如果 *macAutoRequest* 是 TRUE,当发现的信标帧个数等于规定的最大值(即设备一次能够进行扫描的信道的最大个数),或是对某信道的扫描时间达到了规定的扫描持续时间,就中止对该信道的被动扫描。而如果 *macAutoRequest* 是 FALSE,只有在扫描时间结束后,才能中止对某一特定信道的被动扫描,相反,扫描时间没有达到最大值,就认为该信道还没有进行扫描。

当 *macAutoRequest* 是 TRUE 时,保存的 RWSN 描述符的数目等于规定的最大值(即设备一次能够进行扫描的信道的最大个数),或是指定的每个可用信道均已扫描完毕,整个被动信道扫描程序才会中止。如果 *macAutoRequest* 是 FALSE,只有在对每个信道进行被动扫描后,才会结束被动扫描程序。

### 7.5.3.1.5 协调器信道扫描与选择方法

在信道扫描与选择方法中,协调器负责维护三个参数:待扫描信道队列  $Q_i (i=0,1,2,\dots)$ ,信道队列中每个信道经 ED 检测得到的能量等级 E,信道队列中每个信道的能量等级平均值 EAV。

由单个信道扫描时间公式  $Scanduration = aBaseSuperframeDuration * (2^n + 1)$ ,其中  $0 \leq n \leq 14$ ,令  $T_s = aBaseSuperframeDuration$ ,可得到表 72。

表 72 单个信道扫描时间 单位为秒

<i>n</i>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Scanduration	1	3	5	9	17	33	65	129	257	513	1 025	2 049	4 097	8 193	16 387

协调器信道扫描的流程如下:

- a) 网络建立时,协调器根据  $(RWSN\ Id) \bmod 16$  的结果,从信道页 0~15 中对应选择一个信道页,并从该信道页中随机选取  $N (N \leq 16)$  个待扫描信道,从而得到待扫描信道队列  $Q_0$ ;
- b) 协调器以最短的单个信道扫描时间  $Scanduration = t_s$ ,即单个信道扫描时间公式  $[aBaseSuperframeDuration * (2^n + 1)]$  中  $n=0$  时,通过 MLME-SCAN.request 原语对指定某信道页的若干个信道,按信道号由低至高依次进行 ED 扫描,若检测到某个信道能量值小于 ED 门限,则立即停止信道扫描,以此信道作为协调器所在 RWSN 网络初建立时的约定信道。同时通过 MLME-SCAN.confirm 原语报告信道扫描结果,其中,EnergyDetectList 列表包含了已扫描信道的能量等级;
- c) 网络建立后,如图 64 所示,协调器在超帧的非活跃期对约定信道(prescribed channel)和候选信道(Candidate Channel)以相等的单个信道扫描时间进行 ED 检测,候选信道为  $Q_0$  队列中约定信道的后一个信道。此时协调器根据超帧配置,以非活跃期时长 IP(Inactive Period)为依据,结合单个信道扫描时间公式,求得满足不等式  $aBaseSuperframeDuration * (2^n + 1) \leq IP/2$  的  $n$  的最大值(其中  $0 \leq n \leq 14$ ),进而确定非活跃期进行信道扫描的单个信道扫描时间。同时,从表 72 中选择为已确定的单个信道扫描时间的整数倍值,以这些整数倍值个超帧时长作为执行信道维护周期。如选定单个信道扫描时间为 3T,则为 3T 整数倍的有 9T, 33T, 129T, 513T, 2 049T, 8 193T, 那么此后分别以 3 个, 11 个, 43 个, 171 个, 683 个, 2731 个超帧时长为信道维护周期执行信道扫描;

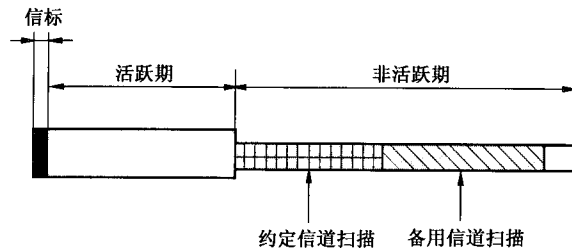


图 64 超帧非活跃期信道扫描结构

- d) 开始第一轮信道扫描过程,在非活跃期通过 MLME-SCAN.request 原语指定约定信道和候选信道两个信道进行扫描,扫描结束后比较约定信道的能量等级  $E_p$  和候选信道的能量等级  $E_c$ ,执行信道维护过程,更新约定信道和候选信道,并将更新后的约定信道和候选信道信息插入信标帧负载中。按照此方法依次扫描,直至扫描至  $Q_0$  队列中的最后一个信道,第一轮信道扫描过程结束,此时按所扫描信道的能量等级  $E_i$  由低至高排列得到信道队列  $Q_1$ ,第一轮信道维护周期为一个超帧时长;
- e) 第二轮至最后一轮,依次对前一轮信道扫描得到的队列进行扫描:第二轮依次扫描信道队列  $Q_1$ ,第三轮依次扫描信道队列  $Q_2 \dots$ 。同理,每个超帧非活跃期内通过 MLME-SCAN.request 原语指定对应待扫描的约定信道和候选信道进行扫描,同时以步骤 2 确定的信道维护周期进行信道维护过程,维护过程中比较信道维护周期内,所扫描约定信道能量等级平均值  $EAV_p$  和候选信道的能量等级平均值  $EAV_c$  ( $EAV_p$  和  $EAV_c$  分别由维护周期内多个超帧 ED 检测得到的  $E_p$  和  $E_c$  取平均值得到),信道维护过程结束后,和第一轮一样,更新约定信道、备用信道和候选信道,并将更新后的约定信道和备用信道信息插入信标帧负载中。每轮信道结束后同样按照每个信道的能量等级平均值  $EAV_i$  ( $i=2,3,\dots$ ) 由低至高排列得到每轮对应的信道队列  $Q_i$  ( $i=2,3,\dots$ )。
- f) 最后一轮信道扫描结束后,回复至步骤 3,重新开始第一轮信道扫描,如此往复。

信道维护过程如下:

- a) 协调器在非活跃期内首先扫描约定信道,随后扫描候选信道,扫描结束后,记录两个信道的能量等级  $E_p$  和  $E_c$ ,同时执行约定信道和候选信道的维护过程,如图 65 所示;

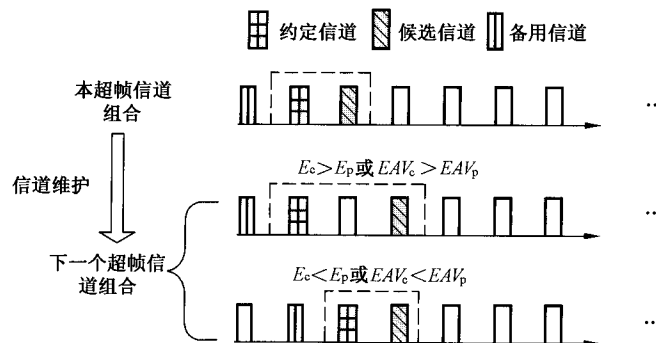


图 65 信道维护过程

- b) 若  $E_c$  大于  $E_p$ , 或  $EAV_c$  大于  $EAV_p$ , 则在下一超帧中仍维持当前约定信道作为下一个超帧的约定信道,当前候选信道作为下一个超帧的候选信道,而下一个超帧扫描的候选信道则为本超帧候选信道所在信道队列  $Q$  中的后一个信道;

- c) 若  $E_c$  小于  $E_p$ , 或  $EAV_c$  小于  $EAV_p$ , 则本超帧的候选信道成为下一个超帧的约定信道, 本超帧的约定信道成为下一个超帧的候选信道, 而下一个超帧所扫描的候选信道则为本超帧中候选信道在信道队列  $Q$  中的后一个信道。

协调器进行信道扫描与选择过程一个范例的流程如图 66 所示。

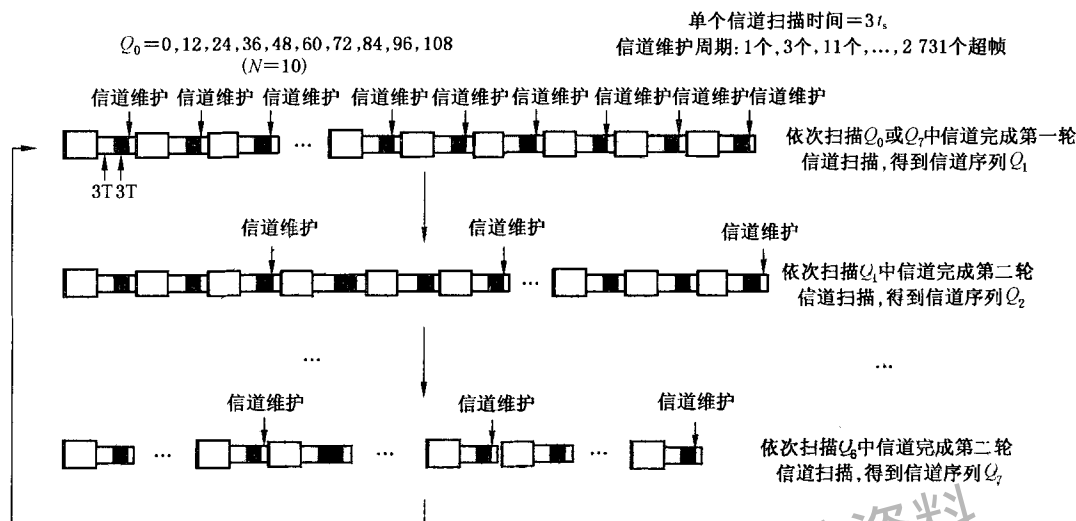


图 66 信道扫描与选择方法流程图

### 7.5.3.2 RWSN ID 的冲突处理

#### 7.5.3.2.1 概述

在相同通信范围内的两个 RWSN 使用同一个 RWSN ID, 就产生了冲突, RWSN 协调器和设备就应执行冲突处理程序, 而这一过程对于 RFD 是任选的。

#### 7.5.3.2.2 冲突的检测

当以下情况发生时, RWSN 协调器就判断 RWSN ID 产生冲突:

- 所收信标帧, 其中 RWSN coordinator 位设为 1 (见 7.2.3.1.3), 且 RWSN ID 等于  $mac-RWSNID$ ;
- 收到 RWSN 中关联设备发出的 RWSN ID 冲突通知命令 (见 7.3.6)。

与 RWSN 协调器关联的设备则在满足如下条件时判断为 RWSN ID 冲突:

信标帧的 RWSN coordinator 位设为 1, 且 RWSN ID 等于  $macRWSNID$ , 但地址既不为  $macCoord-ShortAddress$ , 也不为  $macCoordExtendedAddress$ 。

当设备和不是 RWSN 协调器的协调器关联时, 无法检测到 RWSN ID 产生冲突。

#### 7.5.3.2.3 冲突的处理

RWSN 中的设备发现 RWSN ID 产生冲突后, 就向 RWSN 协调器发送 RWSN ID 冲突的通知命令 (见 7.3.6)。由于命令中包含确认请求, 协调器收到命令后立即给设备发送确认帧。设备收到确认帧后, MLME 发布  $MLME-SYNC-LOSS.indication$  原语, 其中参数  $LossReason$  设置为  $RWSN\_ID\_CONFLICT$ 。若设备没有及时收到确认, MLME 不会将 RWSN ID 冲突告知邻近高层。

RWSN 协调器检测到 RWSN ID 产生冲突, MLME 就向邻近高层发布原语 MLME-SYNC-LOSS.indication, 其中参数 *LossReason* 设置为 RWSN\_ID\_CONFLICT。高层就会进行主动信道扫描, 并利用扫描结果选择一个新的 RWSN ID, 然后通过 MLME-START.request 原语, 参数 *CoordRealignment* 设为 TRUE, 以便重组 RWSN(见 7.5.3.3)。

### 7.5.3.3 建立和重组 RWSN

#### 7.5.3.3.1 概述

本条内容规定了 RWSN 协调器如何建立和重组 RWSN, 以及网络内部设备的重要关联。

#### 7.5.3.3.2 建立 RWSN

RWSN 的产生只能由 FFD 完成: FFD 产生 MLME-RESET.request, 原语的参数 *SetDefaultRIB* 设置为 TRUE, 来复位 MAC 层, 并执行主动信道扫描, 选择合适的 RWSN ID。要注意一点, FFD 在设置 *macShortAddress* 时, 属性值小于 0xffff。从利用主动信道扫描返回的 RWSN 描述符列表中选择合适的 RWSN ID 的算法不在标准的范围内。

RWSN 建立后, FFD 就通过 MLME-START.request 原语(*RWSNCoordinator* 设为 TRUE, *CoordRealignment* 设为 FALSE)控制该 RWSN。MAC 层收到 MLME-START.request 原语后, 就更新超帧的结构和信道参数(见 7.5.3.4), 然后向 FFD 发布 MLME-START.confirm 原语, 且返回状态是 SUCCESS, 该 FFD 就作为 RWSN 的协调器。

#### 7.5.3.3.3 重组 RWSN

当协调器收到原语 MLME-START.request, 参数 *CoordRealignment* 设为 TRUE, 就发送重组命令, 命令中携带 RWSN ID, 逻辑信道和信道记录(如果存在)的新信息。

如果协调器已经发送了信标帧, 而且参数 *CoordRealignment* 设为 TRUE, 下一个信标帧就使用当前的超帧结构, 且帧控制字段的帧待处理子字段设置为 1。信标发送完毕后就在当前信道采用 CSMA-CA 机制发送重组命令。

如果协调器没有发送信标帧, 且参数 *CoordRealignment* 设为 TRUE, 就应立即采用 CSMA-CA 机制在当前信道发送重组命令。

如果信道竞争失败而无法成功发送重组命令, MLME 就向邻近高层发布原语 MLME-START.confirm, 状态为 CHANNEL\_ACCESS\_FAILURE。邻近高层收到后可能会重新发布 MLME-START.request 原语。

如果重组命令发送成功, 就会在下一信标中采用新的超帧结构和新的信道参数或是立即采用新参数(前提是协调器还没有发送信标帧), 相应地, MAC 层发布原语 MLME-START.confirm, 状态是 SUCCESS。

#### 7.5.3.3.4 RWSN 内部结构的重组

设备收到来自和其关联协调器重组命令, MLME 就会发布原语 MLME-SYNC-LOSS.indication, 参数 *LossReason* 设置为 REALIGNMENT, *RWSNId*、*LogicalChannel*、*ChannelPage* 以及重组命令帧中对应参数的设置相同, 同时将自身 MSL 置为默认值 1。协调器的邻近高层会发布 MLME-SYNC-LOSS.indication 原语, 且参数 *CoordRealignment* 设置为 TRUE; 相应地, 不是协调器的设备高层会根据 MLME-SET.request 原语改变超帧结构和信道参数。

#### 7.5.3.4 更新超帧结构和信道 RIB 属性

为了能及时更新超帧结构和信道属性, MLME 将根据原语 MLME-START.request 的参数来设置

相应的 RIB 属性,例如:MLME 根据参数 *BeaconOrder* 的属性来设置 *macBeaconOrder*,如果 *macBeaconOrder* 为 15,*macSuperframeOrder* 也设为 15。在这种情况下,原语构造一个不用信标能力 RWSN。如果 *macBeaconOrder* 的值小于 15,MAC 层就会根据参数 *SuperframeOrder* 的属性来设置 *macSuperframeOrder*。此外,MAC 层也要分别根据参数 *RWSNId*、*ChannelPage* 和 *LogicalChannel* 的属性及时更新 *macRWSNID*、*phyCurrentPage* 和 *phyCurrentChannel*,这一过程要连续两次发布原语 PLME-SET.request 来实现。

### 7.5.3.5 信标帧的生成

只要设备的 *macShortAddress* 不是 0xffff,就能够发送信标帧。

只有当 *BeaconOrder* 的值小于 15,FFD 通过原语 MLME-START.request 开始发送信标帧。

FFD 既可以作为新建 RWSN 的 RWSN 协调器发起信标帧,也可以作为已建 RWSN 的普通设备发起信标帧,而这取决于参数 *RWSNCoordinator* 的值(见 7.1.14.2)。FFD 只有成功关联到一个已建 RWSN 后,才能在 RWSN 中传输信标帧。

若 FFD 是 RWSN 协调器(*RWSNCoordinator* 设为 TRUE),MAC 层就忽略参数 *StartTime* 的值,并立即发送信标帧。而 *StartTime* 为 0 时,也会促使 MAC 层执行相同的操作。如果 FFD 不是 RWSN 协调器,且 *StartTime* 不为 0,就需要采用以下方法计算信标帧的发送时间:首先将参数 *StartTime* 和退避时隙边缘一致,当 MAC 层收到关联的协调器发送的信标帧,就把接收时刻加到参数 *StartTime* 上。当本地时钟和所计算的时间相同,MAC 层就发送信标帧。为了计算信标帧的发送时间,MAC 层首先要跟踪关联协调器的信标帧。如果发布了原语 MLME-START.request,参数 *StartTime* 不为 0,MAC 层此时也并未跟踪协调器的信标帧,MLME 就不会发送信标帧,而是发布原语 MLME-START.confirm,状态设置为 TRACKING\_OFF。

如果设备连续错过了 1 至 (*aMacLostBeacons* - 1) 个来自其协调器的工作信标帧,设备将根据 *macBeaconOrder*(见 7.5.3.4)和本地时钟继续发送自己的信标帧。如果设备发送自己的信标帧后收到协调器的信标帧,说明设备没有失去同步,就根据参数 *StartTime* 和流入信标重新发送自己的信标帧。如果设备与协调器失去了同步,设备的 MLME 将向邻近高层发布原语 MLME-SYNC-LOSS.indication,并立即停止发送自己的信标帧。不论何时收到原语 MLME-SYNC-LOSS.indication 之后,邻近高层可能发布新的 MLME-START.request 原语来重新传输信标帧。

收到 MLME-START.request 原语后,MAC 层将 RWSN 标识符保存到 *macRWSNId*,且在信标帧的源 RWSN 标识符字段中使用该属性值。如果 *macShortAddress* 等于 0xffff,信标帧源地址字段的地址将为 *aExtendedAddress*,否则为 *macShortAddress*。

在周期性发送信标帧的过程中,协调器需要对每个与之关联的设备维护一个参数:该设备的下一个工作信标帧的序列号 NWBSN。若某个信标帧的序列号与某个已关联设备的 NWBSN 值相等,那么在广播该信标帧的同时,协调器按以下公式更新上述设备的 NWBSN 值:更新后的 NWBSN = 更新前的 NWBSN + 该设备的 MSL 值。协调器发送信标帧之前,先检查其所维护的每个已关联设备的 NWBSN 值与否与当前要发送的信标帧的序列号相等,与之相等的设备即为当前要发送的信标帧所对应的超帧的工作设备。在当前要发送的信标帧的 SCFP 字段、周期分配字段、待转发数据目标地址字段和净荷字段内,只安排其对应超帧的工作设备的相关事务。

信标帧的最新发送时间保存在 *macBeaconTxTime* 中,为了确保在每个信标帧中的保存位置都是相同的符号边缘,需要计算信标帧的发送时间。属性 *macSyncSymbolOffset* 详细规定了符号边缘,它和流入信标帧的时间戳有相同的用法(见 7.5.5.2)。

每个信标帧都在超帧的开始处发送,且发送间隔是  $aBaseSuperframeDuration * 2^n$  个符号,其中  $n$  是 *macBeaconOrder* 的值(信标帧的结构见 7.2.3.1)。

信标帧的发送优先级高于其他收发操作。

### 7.5.3.6 设备的发现

所有协调器通过发送信标帧向其他设备表明自己的存在,这也使其他设备能执行设备发现操作。

当协调器不是 RWSN 协调器时,只有在和 RWSN 成功关联后,才能发送信标帧,其他设备则通过原语 MLME-START.request(*RWSNCoordinator* 设为 FALSE)开始发送信标帧,过程如下:MLME 收到 MLME-START.request,就根据参数 *StartTime* 的值发送一个信标帧,在该过程中要用到以下参数:关联 RWSN 的 *macRWSNId* 以及设备的地址信息。如果 *macShortAddress* 为 0xffff,就采用扩展地址 *aExtendedAddress*,否则,就采用短地址 *macShortAddress*。每隔  $aBaseSuperframeDuration * 2^n$  个符号就发送一个信标帧,其中  $n$  是 *macBeaconOrder* 的值。

## 7.5.4 关联和解关联

### 7.5.4.1 关联

设备通过发布原语 MLME-RESET.request(*SetDefaultRIB* 为 TRUE)对 MAC 层复位,并进行主动或被动信道扫描后,才能和 RWSN 关联,其中信道扫描的结果用于选择 RWSN,但如何选择 RWSN 不在标准讨论范围之内。

选择要关联的 RWSN 后,邻近高层就通过原语 MLME-ASSOCIAT.request 请求 MLME 设置 PHY 和 MAC 层的 RIB 属性以便关联:

- phyCurrentChannel* 和原语 MLME-ASSOCIAT.request 的参数 *LogicalChannel* 相同;
- phyCurrentPage* 和原语 MLME-ASSOCIAT.request 的参数 *ChannelPage* 相同;
- macRWSNId* 和原语 MLME-ASSOCIAT.request 的参数 *CoordPABNId* 相同;
- macCoordExtendedAddress* 或 *macCoordShortAddress* 取决于欲关联的协调器发送的信标帧,它和 MLME-ASSOCIATE.request 的参数 *CoordAddress* 相同。

只有协调器的 *macAssociationPermit* 属性为 TRUE 时,它才允许关联。同样,设备只能通过允许关联的协调器和一个 RWSN 建立关联。*macAssociationPermit* 为 FALSE 的协调器,会忽略收到的关联请求命令。在信道扫描结果中将会指示协调器是否允许关联。

为了优化有信标能力网络的关联过程,设备可以跟踪欲关联的协调器发送的信标帧。这一过程是通过邻近高层发布原语 MLME-SYNC.request(*TrackBeacon* 为 TRUE)实现的。

原语 MLME-ASSOCIATE.request 指示设备和已存在的 RWSN 关联,而不是产生自己的 RWSN。

未关联设备的 MAC 层向已生成的 RWSN 的网络协调器发送关联请求来开始关联操作;如果由于信道接入失败而无法成功发送关联请求,MAC 层就通知邻近高层。协调器应对收到的关联请求发送确认帧。

对关联请求的确认并不意味着允许关联,因为协调器的邻近高层需要时间来判断目前 RWSN 的资源是否足够允许设备关联,且应在 *macResponseWaitTime* 个符号内作出决定。如果协调器的邻近高层发现设备已经和 RWSN 关联,就会删除得到的信息。如果有足够可利用的资源,协调器的邻近高层就会给设备分配一个 16 比特的短地址,而且 MAC 层产生关联响应命令,关联响应命令中包含这个新地址并指示关联成功;相反,如果没有足够可利用的资源,就不能进行关联,协调器的邻近高层将通知 MAC 层,相应地,产生关联响应命令指示关联失败。关联响应命令将间接地发送给请求关联的设备。

若关联请求帧里地址位为 1,协调器将根据支持的地址方式,在地址范围内选择一个 16 比特短地址,否则地址为 0xffff,在这种情况下,设备虽然已经成功关联但并没有获得一个 16 比特短地址,但在 RWSN 中只能使用 64 比特的扩展地址来标识自己。

设备在收到关联请求的确认帧之后,等待 *macResponseWaitTime* 个信号时间以便协调器做关联请求判断。RIB 属性 *macResponseWaitTime* 是一个与网络拓扑相关的参数,其值符合设备所要关联的网

络的特定要求。如果设备正在跟踪信标帧,当收到的信标帧中指示有它的数据时,设备就尝试从协调器提取关联响应帧;如果设备没有跟踪信标帧,设备在 *macResponseWaitTime* 信号时间之后才尝试从协调器提取关联响应帧。如果设备在 *macResponseWaitTime* 信号时间内没有从协调器提取到关联响应帧,MLME 就要向邻近高层发送原语 MLME-ASSOCIATION.confirm,状态为 NO-DATA,关联请求失败。这时,设备的邻近高层通过发送一个参数 *TrackBeacon* 为 FALSE 的原语 MLME-SYNC.request 来终止对信标帧的跟踪。

由于关联响应命令中包含确认请求,请求关联的设备应回复一个确认帧。如果命令帧的关联状态字段的值指示关联成功,设备将命令帧中的 16 比特短地址字段内的地址信息保存到 *macShortAddress*;设备利用短地址在 RWSN 中的通信方式取决于参数 *macShortAddress* 的有效范围。如果扫描后用于关联的信标帧携带了协调器的短地址,关联响应命令帧的 MHR 字段的协调器扩展地址将保存到 *macCoordExtendedAddress*。关联成功后,协调器将在下一个信标帧中为设备分配工作周期。

如果关联相应命令的关联状态字段显示关联失败,设备将 *macRWSNId* 设为默认值 0xffff。16 比特短地址的利用见表 73。

表 73 16 比特短地址的利用

<i>macShortAddress</i> 的值	描述
0x0000~0xffff	如果含有源地址,对于信标帧和数据帧,采用短地址,而对于 MAC 命令帧采用其他合适的寻址方式(见 7.3)
0xfffe	如果含有源地址,对于信标帧和数据帧,采用扩展地址,而对于 MAC 命令帧采用其他合适的寻址方式(见 7.3)
0xffff	设备不能关联,所以进行任何的数据收发。而对于 MAC 命令帧采用合适的源地址寻址方式(见 7.3)

#### 7.5.4.2 解关联

当邻近高层向 MLME 发布 MLME-DISASSOCIATE.request 原语,MAC 就执行解关联程序。

如果协调器希望某个已关联的设备离开 RWSN,协调器的 MLME 子层将发送解关联通告命令,发送该命令的方式取决于邻近高层已发送的 MLME-DISASSOCIATE.request 原语中的参数 *TxIndirect*。如果 *TxIndirect* 为 TRUE,协调器将使用非直接传输方式发送解关联通告命令给设备,这意味着解关联通告命令帧将添加到协调器的待处理事务列表,接收设备将采用 7.5.7.3 所描述的方法来提取解关联通告命令,即使接收设备无法成功提取解关联通告命令,协调器也认为设备已经解关联。如果 *TxIndirect* 为 FALSE,协调器将解关联通告命令直接发送给设备,在这种情况下,若信道竞争失败而无法成功发送该命令,MAC 层就会通知邻近高层。

解关联通告命令中包含确认请求(见 7.3.4),接收设备应回复一个确认帧。然而,即使未收到确认帧,协调器也认为解关联成功。

如果已关联设备希望脱离 RWSN,设备的 MLME 将向协调器发送解关联通告命令。若信道竞争失败而无法成功发送命令,MAC 层就会通知邻近高层。由于解关联通告命令中包含确认请求,协调器回复发送确认帧。然而,即使未收到确认帧,设备也认为解关联成功,自己已经脱离该 RWSN。

如果解关联通告命令中的源地址是 *macCoordExtendedAddress*,设备认为自己已经和 RWSN 解关联。如果协调器收到解关联命令,且源地址不等于 *macCoordExtendedAddress*,协调器将验证源地址对应的设备是否已关联设备;如果是,协调器将认为该设备已解关联。若以上条件均不满足,协调器就丢弃该命令帧。



已关联设备通过删除和 RWSN 相关的所有相关信息来解关联; MLME 将参数 *macRWSNId*、*macShortAddress*、*macAssociatedRWSNCoord*、*macCoordShortAddress* 和 *macCoordExtendedAddress* 设置为默认值。协调器通过删除设备的所有相关信息来和设备解关联。

请求解关联设备的邻近高层将通过 MLME-DISASSOCIATE.confirm 原语获得解关联的结果。

## 7.5.5 同步

### 7.5.5.1 概述

本条规定了协调器如何生成信标帧以及设备节点如何与协调器进行同步。对于不支持信标帧的 RWSN,设备通过向协调器请求数据来实现同步对于支持信标帧的 RWSN,设备通过接受信标帧并解码来实现同步。在支持信标帧的 RWSN 中,不同的设备将按照不同的工作周期跟踪信标帧。

### 7.5.5.2 信标同步

在信标使能的 RWSN(即 *macBeaconOrder* < 15)中,为了检测所有待处理数据或跟踪信标帧,所有设备将执行信标同步。只有当信标帧的 RWSN 标识符和参数 *macRWSNId* 规定的相同时,设备才能与该信标帧进行同步。如果 *macRWSNId* 设置为广播 RWSN 标识符(0xffff),设备不执行信标同步。

原语 MLME-SYNC.request 指示设备跟踪信标帧。如果原语定义了跟踪,设备将获取信标帧,并按照设备的工作周期,周期性的打开接收机以跟踪信标帧保持同步。如果未定义跟踪,设备会在下一信标帧到来时中止同步。

在周期性接收信标帧的过程中,设备需维护自身的一个参数:该设备的下一个工作信标帧的序列号 NWBSN。设备在接收到新的工作信标之后,会按以下公式更新自身存储的 NWBSN 值:更新后的 NWBSN = 更新前的 NWBSN + 该设备的 MSL 值。

为了与信标同步,设备打开接收机搜索信标帧,时间不能超过规定的最长时间:

$[aBaseSuperframeDuration * (2^n + 1)]$ , 式中  $n$  是 *macBeaconOrder* 的值,若设备在该时间内未能收到携带设备所在 RWSN 的 RWSNId 的信标帧或收到的信标帧的序列号与设备存储 NWBSN 值不相等,则判定未收到有效信标,MLME 就在下一个工作周期开始时再次打开接收机重复上述搜索过程。一旦未收到有效信标的连续搜索次数达到 *aMaxLostBeacons*, MLME 将向邻近高层发布 MLME-SYNC-LOSS.indication 原语,状态参数为 BEACON\_LOSS。

MLME 在收到的信标帧的相同码元边缘处计时,位置的选择 *macSyncSymbolOffset* 的属性决定,同时该位置也与流出信标帧计时的码元边缘位置一样,后者取决于 *maxBeaconTxTime* 的属性。计时的数值等于当设备处于码元边缘位置的时间值,所以它是相对时间,不过也可以根据需要变换为绝对时间。

收到信标后,MLME 将提取源地址以及 MAC 帧头中的源 RWSNId 字段的信息,并分别与协调器的源地址(源地址是 *macShortAddress* 还是 *macCoordShortAddress* 取决于寻址方式)和 RWSNId (*macRWSNId*)进行对比,若不匹配,就丢弃该信标帧。

若收到有效信标帧,且 *macAutoRequest* 为 FALSE,MLME 就通过原语 MLME-BEACON-NOTIFY.indication 向上层传送信标帧参数。

当设备收到有周期分配字段的信标帧,且周期分配列表字段包含有自己的设备段地址,则 MLME 就提取周期分配字段的剩余数据,更新自己的工作周期。之后,设备将与协调器广播的下一个信标同步,同时开始进入新的工作周期循环。

如果网络处于信标跟踪状态,则 MLME 会在其下一个工作周期开始之前打开接收机。如果连续不能收到 *aMaxLostBeacons* 个有效信标帧,MLME 将向邻近高层发布 MLME-SYNC-LOSS.indication 原语,状态参数为 BEACON\_LOST。

## 7.5.6 事务处理

本条主要针对低成本设备,即设备一般使用电池供电,因此事务的处理请求来自设备而不是协调器,且数据传输一般采用间接方式,即协调器应在信标帧中指示设备有待处理的数据,或由设备询问协调器是否有其待处理的事务。在信标使能的网络里,协调器只会在信标帧中安排该信标帧所对应的超帧的工作设备安排相关事务。

当协调器的邻近高层发布原语 MCPS-DATA.request,或是 MLME 请求发送 MAC 命令(一般 MAC 命令由邻近高层发布的原语产生,如 MLME-ASSOCIATE.request,见 7.1.4.2),协调器开始处理事务。事务处理完毕,MAC 层就给邻近高层返回状态值。如果请求原语发起间接传输,就要使用相应的确认原语返回状态值。相反,如果响应原语发起间接传输,就要使用原语 MLME-COMM-STATUS.indication 来返回状态值。要注意的是,应检查目标地址的信息,使原语 MLME-COMM-STATUS.indication 和其对应的相应原语关联。

间接数据传输请求里包含的信息构成了一个事务,协调器至少要能储存一个事务。当协调器收到请求而没有足够的存储空间,MAC 层就向邻近高层发布相应的原语,返回状态 TRANSACTION-OVERFLOW。在收到间接数据传输请求后进行事务处理。

若协调器有足够的存储空间,则确保同一个设备的事务按照它们到达 MAC 层的顺序进行传输。如果所传输的事务不是最后一个,MAC 层就把帧中的帧待处理子字段设为 1,指示还有需要处理的事务。

每个事务在协调器中的停留最长时间为 *macTransactionPersistenceTime*,若超过,MAC 层将丢弃该事务的信息,且向邻近高层发布一个相应的原语,返回 TRANSACTION\_EXPIRED 状态。为确保每个事务都能成功提取,设备应对接收的事务回复确认帧。

如果成功处理某个事务,MAC 层就删除该事务的所有信息,并通过相应的原语向邻近高层返回状态 SUCCESS。

协调器在发送信标帧时,则应在地址列表中列出事务相关设备的地址,且在待处理地址字段中说明地址的数目。若协调器能储存 7 个以上的事务,则按照先到先服的原则在信标帧中指明,确保帧中包含的地址不超过 7 个。对用 SCFP 进行传输的事务,信标帧不包含对应设备的地址,协调器只是在分配给设备的 SCFP 中进行传输。在信标使能的网络里,协调器只会在信标帧中安排该信标帧所对应的超帧的工作设备安排相关事务。

在有信标能力网络里,如果有发往广播地址的事务,帧中控制字段中的帧待处理子字段设置为 1,且紧跟在信标传输结束后采用 CSMA-CA 机制发送待处理事务,如果还有类似的事务,应推迟到以后的超帧中发送,即每个超帧只能间接传输一个事务。

在有信标能力 RWSN 中,设备收到包含自身地址的信标帧就将向协调器提取数据,而在非信标使能 RWSN 里,设备在收到 MLME-POLL.request 原语后向协调器提取数据。设备提取待处理数据的过程见 7.5.7.3。如果设备收到的信标帧中待处理子字段的值为 1,设备的接收机会保持打开状态来接收来自协调器的广播数据,持续时间是 *macMaxFrameTotalWaitTime* 个符号。

## 7.5.7 发送、接收和确认

### 7.5.7.1 发送

每个设备都将其当前的 DSN 存储在 MACRIB 属性 *macDSN* 中,并将其初始化为一个随机数,选择随机数的算法不在规范的制订范围。每当生成一个新的数据帧或 MAC 命令帧,MAC 层都将 *macDSN* 的值复制到流出帧的 MHR 的对应字段中,再将 *macDSN* 加 1。不论设备要和哪几个设备通信,都只会产生一个 DSN。又因为 *macDSN* 长度有限,达到最大值后,要重新复位,因此允许 *macDSN*

循环使用。

而对于协调器,则要把自己当前的信标序列号(BSN)存储在 MACRIB 属性 *macDSN* 中,并将其初始化为一个随机数;随机数的选择也不在规范的制订范围。每生成一个信标帧,MAC 层将 *macBSN* 的值复制到流出帧的 MHR 的对应字段中,并将 *macBSN* 加 1。同样也允许 *macBSN* 循环使用。

要注意的是,DSN 和 BSN 都只有 8 比特,对于邻近高层而言,其利用范围是极其有限的,如 DSN,一般用于检测重传的帧。

若帧中有源地址字段,字段值应携带发送设备的地址。当一个设备已经关联,并分配有一个 16 比特的短地址(*macShorAddress* 不是 0xffff 或 0xffff),短地址的使用优先级高于 64 比特的扩展地址(即 *aExtendedAddress*)。而如果设备还没有和 RWSN 关联或 *macShorAddress* 是 0xffff,设备就要在所有要求源地址的通信中使用 64 比特的扩展地址。如果帧中不存在源地址字段,帧来自 RWSN 协调器,目的地址字段中包含接收设备的地址。

如果帧中存在目的地址字段,应携带接收设备的地址,该地址可以是 16 比特短地址,也可以是 64 比特扩展地址。如果目的地址字段不存在,可以认为接收帧的设备是 RWSN 协调器,源地址字段中包含发送设备的地址。

如果目的地址信息和源地址信息均存在,MAC 层就将目的 RWSN 标识符和源 RWSN 标识符进行对比:如果相同,帧控制字段的 RWSN ID 子字段设为 1,并忽略发送帧中的源 RWSN 标识符;如果不相同,帧控制字段的 RWSN ID 子字段设为 0,发送帧包含两者的信息。如果只存在一个地址信息,帧控制字段的 RWSN ID 子字段设为 0,帧中包含地址所在网络的 RWSN 标识符。

在有信标能力 RWSN 中传输帧时,设备应在帧发送前找到信标帧。如果未跟踪到信标帧(见 7.5.5.2)而无法预知信标帧将何时出现,设备将打开接收机来搜索信标帧,搜索时间为  $[aBaseSuperframeDuration * (2^n + 1)]$  个符号,这里  $n$  是 *macBeaconOrder* 的值。如果在这段时间内找不到信标帧,设备采用非时隙的 CSMA-CA 算法竞争信道,并在信道竞争成功后发送帧。如果已经搜索到或正在跟踪信标帧,设备就在其工作周期中侦听到的第一个信标帧所对应的超帧内发送帧。若设备处在跟踪信标帧的状态,则该设备不能在其工作周期中除第一个信标帧以外的其他信标帧所对应的超帧内发送帧。想在 CAP 阶段传输,则应采用时隙 CSMA-CA 算法成功竞争信道(见 7.5.5.2),如果是在 SCFP 中传输则不需使用 CSMA-CA 算法。

如果是在不用信标能力 RWSN 中传输帧,设备应使用非时隙的 CSMA-CA 算法(见 7.5.2.4)成功竞争信道后发送帧。

无论是有信标能力 RWSN 还是不用信标能力 RWSN,若采用直接传输方式,当使用 CSMA-CA 算法没有成功竞争到信道时,应通知邻近高层。而如果采用了间接传输方式,信道竞争失败时,帧会停留在事务队列中,直至要求重新发送或是队列已满时,才会发送出去。

为了发送帧,MAC 层首先要打开发射机,方法是:向 PHY 发布原语 PLME-SET-TRX-STATE.request,状态为 TX\_ON。接收到 PLME-SET-TRX-STATE.confirm 原语后,不管状态为 SUCCESS 还是 TX\_ON,生成的帧将通过 PD-DATA.request 原语发送出去。当收到 PD-DATA.confirm 原语后,MAC 层将关闭发射机,方法是:向 PHY 发布原语 PLME-SET-TRX-STATE.request,状态为 RX\_ON 或 TRX\_OFF,这取决于传输之后接收机是否处于使能状态。如果帧控制字段中确认请求子字段设置为 1,在成功发送帧后,MAC 层向 PHY 发布 PLME-SET-TRX-STATE.request 原语(状态为 RX\_ON)来打开接收机。

### 7.5.7.2 接收和拒收

在空闲阶段,每个设备的 MAC 层都要选择是否让接收机保持使能状态,因为 MAC 层仍要处理来自邻近高层的收发请求。收发任务可以是带有确认(如果要求)的发送请求,也可以是接收请求。

每完成一个收发任务,MAC层将根据 *macBeaconOrder* 和 *macRxOnWhenIdle* 值来请求 PHY 打开或关闭接收机。如果 *macBeaconOrder* 小于 15,仅在下一超帧 CAP 的空闲期间才考虑 *macRxOnWhenIdle* 的值。如果 *macBeaconOrder* 等于 15,总是需要考虑 *macRxOnWhenIdle* 的值。

由于无线通信的特性,接收机处于使能状态的设备不仅能接收来自同一信道和 POS 内的数据并对其译码,也能处理其他设备的干扰,其中数据都是遵从本标准的设备发送的。因此,MAC 层应能够过滤流入的数据帧,并提交发往邻近高层的数据帧。

经过第一层过滤,MAC 层将根据 7.2.2.9 描述的算法,丢弃所有 MFR 中 FCS 字段值不正确的帧。接收端根据接收帧的 MAC 帧头和帧净荷计算帧校验值,并和帧中携带的 FCS 值进行对比。若相同,就接收,否则丢弃。

第二层过滤则取决于 MAC 层是否工作在混合方式。若是,MAC 层把所有通过第一层过滤的帧直接传送到邻近高层,不需要再进行任何过滤或处理。当 *macPromiscuousMode* 设为 TRUE,MAC 层的工作方式为混合方式。

如果 MAC 层不处于混合方式(即 *macPromiscuousMode* 设为 FALSE),它将只接收通过第三层过滤的帧:

- a) 帧类型子字段不含保留帧类型;
- b) 帧版本子字段不含保留值;
- c) 如果帧中包含目的 RWSN 标识符,它应和 *macRWSNId* 相匹配,或是广播 RWSN 标识符(0xffff);
- d) 如果帧中含有目的设备的短地址,它应和 *macShorAddress* 匹配,或是广播地址(0xffff)。相反,如果包含的是扩展地址,应和 *aExtendedAddress* 匹配;
- e) 如果接收帧是信标帧,源 RWSN 标识符应和 *macRWSNId* 匹配,而当 *macRWSNId* 是 0xffff 时,忽略源 RWSN 标识符并接收该信标帧。

如果数据帧或 MAC 命令帧只包含源地址字段,仅当设备是 RWSN 协调器,并且源 RWSN 标识符与 *macRWSNId* 匹配时,才接收帧。

如果不满足第三层的过滤条件,MAC 层就丢弃接收的帧。通过第三层过滤的帧被认为是有效帧,才能进一步处理。对于非广播的有效帧,如果是数据帧或 MAC 命令帧,且帧控制字段的确认请求字段置为 1,MAC 层需要回复确认帧,且确认帧的序列值和所收数据帧或 MAC 命令帧的序列值相同,这样做的目的在于使发送数据帧或 MAC 命令帧的设备收到确认帧后,可以判定接收的确认帧是发给自己的。

如果帧中包含目的地址和源地址的信息,且帧控制字段的 RWSN ID 压缩子字段设为 1,MAC 层就认为被忽略的源 RWSN 标识符和目的 RWSN 标识符相同。

如果该有效帧是数据帧,MAC 层将发布 MCPS-DATA.indication 原语,把帧的相关信息传递给邻近高层。

如果该有效帧是 MAC 命令帧或信标帧,MAC 层就作进一步处理,并向邻近高层发送相应的确认或指示原语。

### 7.5.7.3 从协调器提取待处理数据

有信标能力 RWSN 中的设备能够通过检查收到信标帧的内容来判断是否有待处理的帧,见 7.5.5.2。如果信标帧的地址列表中包含设备的地址,且 *macAutoRequest* 是 TRUE,设备的 MLME 将在超帧的 CAP 阶段向协调器发送数据请求命令(见 7.3.3.2),并将帧控制字段的确认字段设为 1。唯一例外的情况是,当设备进行主动或被动信道扫描,接收到信标帧后并不进行以上所述的操作(见 7.5.3.1)。在这两种情况下,设备的 MLME 向协调器发送数据请求:第一种情况是,MLME 收到邻近高层的 MLME-POLL.request 原语;第二种情况是,MLME 收到数据请求命令的确认帧后并等待规定的最大

时间,即  $macResponseWaitTime$  个符号,例如在关联过程中。如果数据请求是发往 RWSN 协调器,忽略目的地址信息。

成功收到数据请求命令后,协调器将回复确认帧给发送设备以确认收到该命令。如果在发送确认帧(见 7.5.7.4.3)之前,协调器有足够时间判断设备是否有待处理的帧,它将根据判断结果设置确认帧的帧控制字段中的帧待处理子字段。如果没有足够时间,协调器将确认帧的帧控制字段中的帧待处理子字段设为 1。

如果确认帧的帧待处理子字段为 0,设备就判断协调器中没有需要处理的数据。

如果确认帧的帧待处理子字段为 1,设备将打开接收机,并持续规定的最大时间,即  $macMaxFrameTotalWaitTime$  个 CAP 符号,以接收来自协调器的数据帧。如果协调器中的确有请求设备的待处理数据,协调器将使用本条描述的方法向设备发送数据帧。如果没有,协调器将向设备发送无需确认的数据帧,且帧净荷为空,指示目前没有要处理的数据,其中传输方式也采用本条描述的方法。

对数据请求命令进行应答后,将选择如下方法之一发送数据帧:

- a) 不使用 CSMA-CA 算法,前提是 MAC 层能在退避时隙边缘处  $aTurnaroundTime$  和  $(aTurnaroundTime + aUnitBackoffPeriod)$  符号之间传输数据帧,以及在 CAP 中传输消息、一定的 IFS 和确认帧(见 6.5.1 关于  $aTurnaroundTime$  的阐述)。如果在数据帧发送完毕后没有收到确认帧,就按照收到一个新数据请求命令的流程重新开始;
- b) 使用 CSMA-CA 算法。

如果请求设备在  $macMaxFrameTotalWaitTime$  个码元后没有收到来自协调器的数据帧,或者收到的数据帧的净荷为零,设备判定协调器中没有自己待处理的数据。收到来自协调器的数据帧,请求设备将根据协调器的要求发送确认帧以确认收到数据。

如果数据帧的帧控制字段中的帧待处理子字段设为 1,说明协调器中还有待处理数据,设备将重新向协调器发送数据请求命令来提取数据。

#### 7.5.7.4 确认和重传的应用

##### 7.5.7.4.1 概述

在发送数据帧或 MAC 命令帧时,将根据需要恰当地设置帧控制字段的确认请求子字段,而对于信标帧或确认帧,确认请求子字段设为 0,同样的,所有广播帧的确认请求子字段设为 0。特别的,如果是设备向协调器发送监测数据,则帧控制字段的确认请求子字段设为 1,而且设备需收到协调器发来的普通确认帧和数据接受确认帧两个确认帧之后才判定通信成功。

##### 7.5.7.4.2 无确认

如果发送帧的确认请求子字段设为 0,接收方将不回复确认帧,且发送设备也会认为帧已发送成功。

图 67 是无确认时的数据成功传输流程图。在这种情况下,发送的数据帧的帧控制字段的确认请求(AR)子字段为 0。

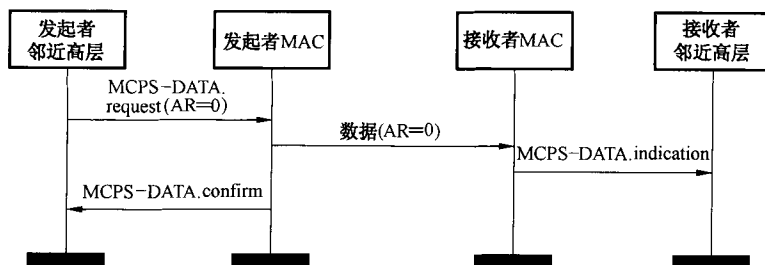


图 67 无确认的数据成功传输流程

### 7.5.7.4.3 确认

如果帧控制字段的确认请求子字段是 1,接收端就应回复一个确认帧。正确接收帧后,接收设备将生成并发送确认帧,且确认帧中包含的 DSN 与接收的数据帧或 MAC 命令帧的 DSN 相同。

在信标非使能 RWSN 中或在 CFP 阶段,完成数据帧或 MAC 命令帧的接收后经过  $aTurnaroundTime$  个符号,设备开始发送确认帧。在 CAP 阶段,要等待  $aTurnaroundTime$  个符号,或是在退避时隙边缘开始发送确认帧。如果是后一种情况,确认帧要在  $aTurnaroundTime$  个符号与  $(aTurnaroundTime + aUnitBackoffPeriod)$  个符号之间的某时刻发送。常量  $aTurnaroundTime$  的定义见表 20(见 6.5.1)。

图 68 是带确认帧的数据成功传输流程图,在这种情况下,发送设备发送数据帧时,将帧控制字段的确认请求子字段设为 1 来要求接收设备回复确认帧。

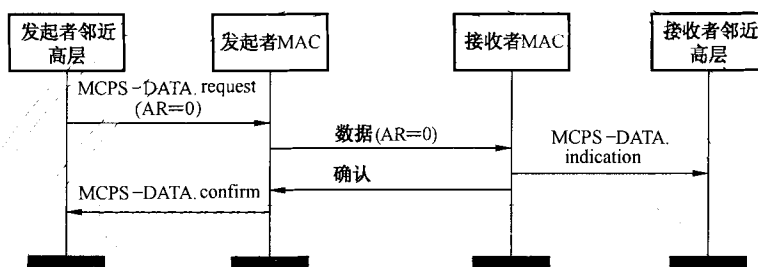


图 68 带确认帧的数据成功传输流程

特别的,如果是设备向协调器发送监测数据,即 MCPS-DATA.request 中的参数  $msduType$  设为 0x01,则帧控制字段的确认请求子字段设为 1,且参数  $TxOptions$  规定采用带数据接受确认的传输,即设备需收到协调器发来的普通确认帧和数据接受确认帧两个确认帧之后才判定通信成功,如图 69 所示。

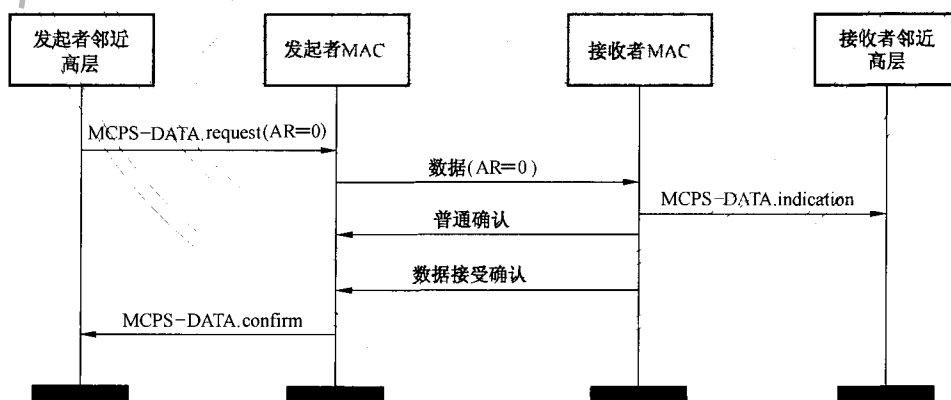


图 69 带数据接收确认帧的数据成功传输流程

### 7.5.7.4.4 重传

当设备发送的帧不要求确认(帧控制字段的确认请求子字段为 0),它总是认为接收端能够成功接收,所以不会进行重传。

当设备发送的是数据帧或 MAC 命令帧,且要求确认(帧控制字段的确认请求子字段为 1),就要等待相应的确认帧。如果在规定的最大时间内,即  $macAckWaitDuration$  个符号,收到确认帧,且 DSN 与所发送帧的 DSN 相同,发送设备判定发送成功;如果规定时间内没有接收到确认帧,或是确认帧的

DSN 和发送帧的 DSN 不相同,设备判定此次发送失败。

当协调器发送数据帧或 MAC 命令帧时,若采用间接传输方式,且没有成功接收,不会进行重传。相反,若采用直接传输方式,该帧将保留在协调器的事务队列中,并且只在收到新的数据请求命令后,才提取该帧进行重传,要注意的是,重传时使用的 DSN 与第一次发送时使用的 DSN 相同。

当设备发送数据帧或 MAC 命令帧时,若采用直接传输方式,且没有发送成功,设备就会重传该帧,并等待相应的确认帧,直到达到最大重复次数 *macMaxFrameRetries*。其中,重传的帧和第一次发送的帧具有相同的 DSN,而且,重传只能在超帧中的同一阶段进行,即第一次传输若是在 CAP 期间,重传也只能在该阶段进行。而如果此时没有足够的时间,重传推迟到下一超帧。当重传次数达到 *macMaxFrameRetries* 后仍未收到确认帧,MAC 层就判定此次发送失败,并将这一信息报告给邻近高层。

特别的,当设备向协调器发送监测型数据帧时,采用直接传输方式,且要求确认(帧控制字段的确认请求子字段为 1),就要等待普通确认帧和数据接受确认帧。若设备在 *macAckWaitDuration* 个符号内未收到普通确认帧,或是确认帧的 DSN 和发送帧的 DSN 不相同,则判定没有发送成功,设备就会重传该帧,并再次等待普通确认帧。若设备在 *macAckWaitDuration* 个符号内收到了普通确认帧,但在 *macDataAckWaitDuration* 时长内未收到数据接受确认帧,或是确认帧的 DSN 和发送帧的 DSN 不相同,则判定没有发送成功,设备就会重传该帧,并再次等待普通确认帧和数据接受确认帧。如果设备重传的次数累计达到最大重复次数 *macMaxFrameRetries*,MAC 层就判定此次发送失败,并将这一信息报告给邻近高层。

#### 7.5.7.5 混合方式

设备通过设置 *macPromiscuousMode* 来启动混合方式。当 MLME 收到请求将 *macPromiscuousMode* 设置为 TRUE 后,就向 PHY 发布 PLME-SET-TRX-STATE.request 原语,其中状态为 RX\_ON,即请求 PHY 打开接收机。

设备处于混合方式时,MAC 层根据 7.5.7.2 内容对接收的帧进行处理,并利用原语 MCPS-DATA.indication 向邻近高层递交所有接收到的有效帧。发送端和接收方的寻址方式参数设置为 0x00,MSDU 参数要包含 MAC 帧头 MHR,且参数 *msduLength* 包含 MHR 和 MAC 净荷的所有八位位组数。*mpduLinkQuality* 设为有效。

如果 *macPromiscuousMode* 为 FALSE,MLME 就向 PHY 发布原语

PLME-SET-TRX-STATE.request,并相应地改变状态,以要求 PHY 根据 *macRxOnWhenIdle* 来改变接收机的状态。

#### 7.5.7.6 传输场景

由于无线信道的固有缺陷,传输的帧不一定总能到达目的设备。图 70 展示了三种数据传输场景:

- a) 数据传输成功:发送设备的 MAC 层通过 PHY 数据服务向接收设备发送数据帧。发送结束后,发送设备 MAC 层等待确认并开始计时,计时器达到 *macAckWaitDuration* 个符号后就停止。接收设备的 MAC 层收到数据帧后,向发送设备回复确认帧,并将数据帧传递给邻近高层。如果发送设备 MAC 层在计时器停止前收到接收设备的确认,将停止计时,并复位计时器。特别的,若发送的数据帧为监测型数据帧,则发送设备 MAC 层在发送结束后还需等待接收设备回复的数据接受确认帧并开始计时,计时器达到 *macDataAckWaitDuration* 时长后就停止,接收设备在与邻近高层确认接受该数据后向发送设备回复数据接受确认帧。如果发送设备 MAC 层在计时器停止前收到接收设备的数据接受确认帧,将停止计时,并复位计时器。这时数据传输完成,发送设备的 MAC 层向邻近高层发布数据发送成功的确认原语;
- b) 数据帧丢失:发送设备的 MAC 层通过 PHY 数据服务向接收设备发送数据帧。发送结束后,发送设备 MAC 层等待确认并开始计时,计时器达到 *macAckWaitDuration* 个符号后就停止。

接收设备的 MAC 层没有收到数据帧,就不会回复确认帧。发送设备在计时器停止前没有收到确认帧,就判定数据传输失败。特别的,如果发送的数据帧为监测型数据帧,且发送设备在 *macDataAckWaitDuration* 时长内未收到数据接受确认帧,也判定数据传输失败。如果发送数据时采用直接传输,发送设备就重传数据,而整个流程中数据重传的累计次数不能超过规定的最大重传次数 *macMaxFrameRetries*;如果第  $(1 + macMaxFrameRetries)$  次数据传输仍然失败,发送设备的 MAC 层向邻近高层发布一个传输失败的确认原语。如果是间接传输,数据帧将保留在事务队列中,直至接收到新的数据请求和相应的确认帧,或是保留时间达到 *macTransactionPersistenceTime*。一旦保留时间达 *macTransactionPersistenceTime*,就丢弃传输信息,MAC 层向邻近高层发布传输失败的确认原语;

- c) 确认帧丢失:发送设备的 MAC 层通过 PHY 数据服务向接收设备发送数据帧。发送结束后,发送设备 MAC 层等待确认并开始计时,计时器达到 *macAckWaitDuration* 个符号后就停止。接收设备的 MAC 层收到数据帧后,向发送设备回复确认帧,并将数据帧传递给邻近高层。发送设备在计时器停止时没有收到确认帧,就判定数据传输失败。特别的,如果发送的数据帧为监测型数据帧,且发送设备在 *macDataAckWaitDuration* 时长内未收到数据接受确认帧,也判定数据传输失败。如果发送数据时采用直接传输,发送设备就重传数据,而整个流程数据重传的累计次数不能超过规定的最大重传次数 *macMaxFrameRetries*;如果第  $(1 + macMaxFrameRetries)$  次数据传输仍然失败,发送设备的 MAC 层向邻近高层发布一个传输失败的确认原语。如果是间接传输,数据帧将保留在事务队列中,直至接收到新的数据请求和相应的确认帧,或是保留时间达到 *macTransactionPersistenceTime*。一旦保留时间达到 *macTransactionPersistenceTime*,就丢弃传输信息,MAC 层向邻近高层发布传输失败的确认原语。

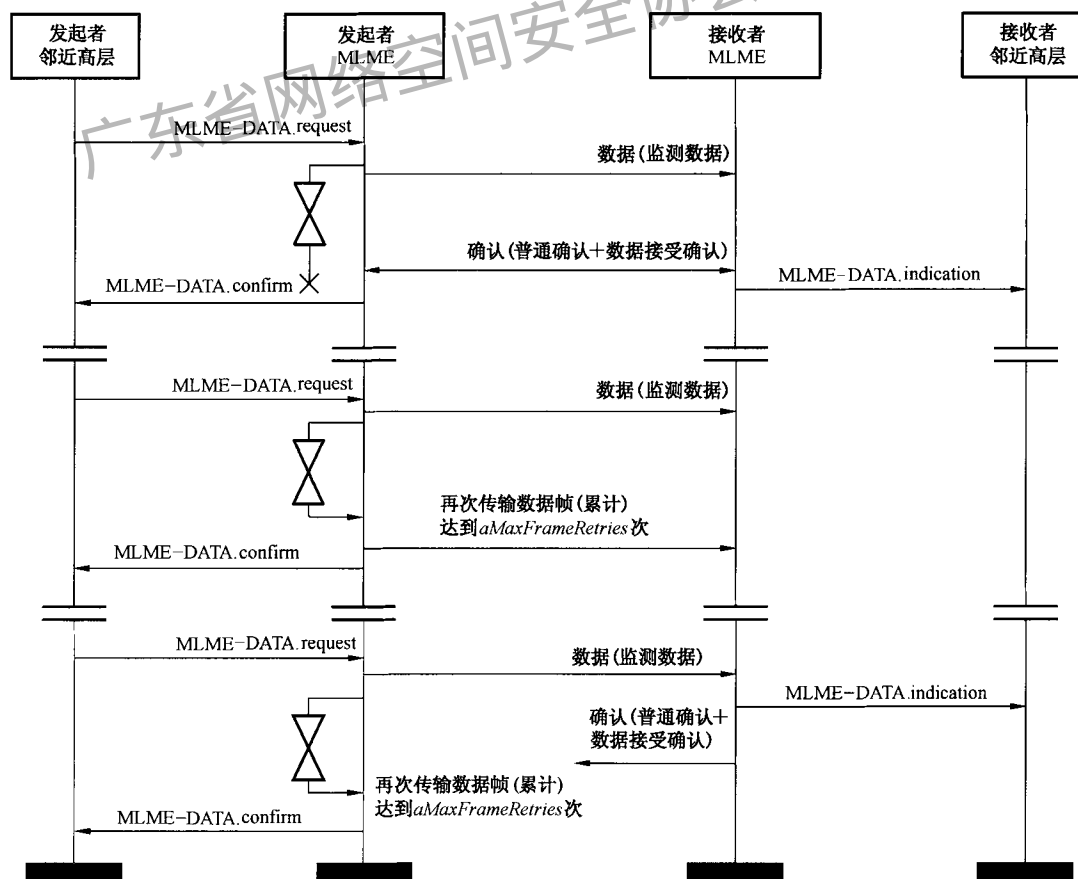


图 70 采用直接传输方式的传输流程



### 7.5.7.7 数据预测与重传机制

#### 7.5.7.7.1 概述

当设备向协调器发送监测型数据时,协调器启动数据预测与重传机制,分为数据预测机制和数据重传机制两部分。若发送监测型数据帧的过程位于超帧的 CAP 阶段,则准备发送相应质疑数据帧、更新数据帧和数据接受确认帧的设备或协调器需在 CAP 阶段采用 CSMA-CA 竞争方式进行发送,但发送时机不限于当前超帧的 CAP 阶段;若发送监测型数据帧的过程位于超帧 CFP 阶段的 SCFP 时限内,则准备发送相应质疑数据帧、更新数据帧和数据接受确认帧的设备或协调器需在相应的 SCFP 时限内进行发送,但发送时机不限于当前超帧的 CFP 阶段。

#### 7.5.7.7.2 数据预测机制

数据预测机制采用下述的历史业务数据的平均值和标准差得到预测值  $Y$  和相应的置信区间。设设备的监测周期为  $T_{\text{mon}}$ ,在一个监测周期中,设备向协调器汇报  $m$  次数据,每个数据汇报时刻的时间间隔为  $T_{\text{mon}}/m$ ,汇报时刻的时间间隔为  $T_{\text{mon}}$  的两个数据间有一定的相关性,协调器在第  $n$  个监测周期内可记录  $m$  个数据,分别为  $X_{n,1}, X_{n,2}, \dots, X_{n,m-1}, X_{n,m}$ 。假设协调器在第  $n'$  个监测周期的第  $m'$  个汇报时刻接收到一个新的汇报数据  $X_{n',m'}$ ,并启动数据预测机制。

此时,协调器取出在之前  $N$  个监测周期中同时刻接收到的  $N$  个数据(每次接收新数据后更新存储的  $N$  个数据),分别为  $X_{n'-N,m'}, X_{n'-N+1,m'}, \dots, X_{n'-2,m'}, X_{n'-1,m'}$ ,并对这  $N$  个数据求平均值,即数据  $X_{n',m'}$  的均值为

$$X_{n',m' \text{ 均}} = \left( \sum_{i=1}^N X_{n'-i,m'} \right) / N$$

再对  $N$  个数据求标准差  $\sigma$ ,数据  $X_{n',m'}$  的标准差为

$$\sigma_{n',m' \text{ 标}} = \sqrt{\left[ \sum_{i=1}^N (X_{n'-i,m'} - X_{n',m' \text{ 均}})^2 \right] / N}$$

则由预测算法得:数据  $X_{n',m'}$  的预测值  $Y_{n',m'}$  为  $X_{n',m' \text{ 均}}$ ,其对应的置信区间为

$$[X_{n',m' \text{ 均}} - \delta_{\text{mon}} \cdot \sigma_{n',m' \text{ 标}}, X_{n',m' \text{ 均}} + \delta_{\text{mon}} \cdot \sigma_{n',m' \text{ 标}}]$$

其中,容忍度系数  $\delta_{\text{mon}}$  取正整数,取值越大,汇报数据落入置信区间的概率越大。

#### 7.5.7.7.3 数据重传机制

设备向协调器发送监测型数据帧,帧子类型子字段为 01。其中,包含监测型数据  $X$ 。协调器在接收到该数据值  $X$  之后,首先回发帧子类型子字段为 00 的普通确认帧。然后,协调器将采用 7.5.7.7.2 中的预测算法,计算出与当前发送时刻对应的该设备监测数据的预测值  $Y$  以及相应的置信区间。下面分为两种情况讨论,如图 71 所示:

- a) 如果设备发送的数据帧中的业务数据值  $X$  落在预测值  $Y$  所对应的置信区间之内,那么协调器向设备回发帧子类型子字段为 01 的数据接受确认帧,表示协调器接受该数据帧;
- b) 如果设备发送的数据帧中的业务数据值  $X$  落在预测值  $Y$  所对应的置信区间之外,那么协调器记录该数据值  $X$ ,然后重新接入信道向设备发送帧子类型子字段为 10 的质疑数据帧,该数据帧中包含数据值  $X$ 。收到质疑数据帧的设备,需要向上层重新核实数据值  $X$  的正确性。此处需再分为三种情况进行讨论:
  - 1) 若核实结果为数据无误,则设备向协调器发送帧子类型子字段为 10 的数据质疑无效确认

- 帧。收到该数据质疑无效确认帧的协调器，确认接受并记录数据值  $X$ ，同时回发帧子类型子字段为 01 的数据接受确认帧；
- 2) 若核实结果为数据有误，则设备先发送帧子类型子字段为 11 的质疑有效确认帧，以示质疑有效。然后，设备重新接入信道向协调器发送帧子类型子字段为 11 的更新数据帧，其中包含核实后的新数据值  $X'$ 。协调器收到后需对新数据值  $X'$  再进行一次是否落在置信区间内的判断，并重复步骤 a)，直到数据值确认被接受为止；
  - 3) 特别的，设备收到质疑数据帧后，若发现质疑数据帧内所含的数据与原上报数据不符，则直接向协调器重发原数据帧，而不需要向上层重新核实数据的正确性。

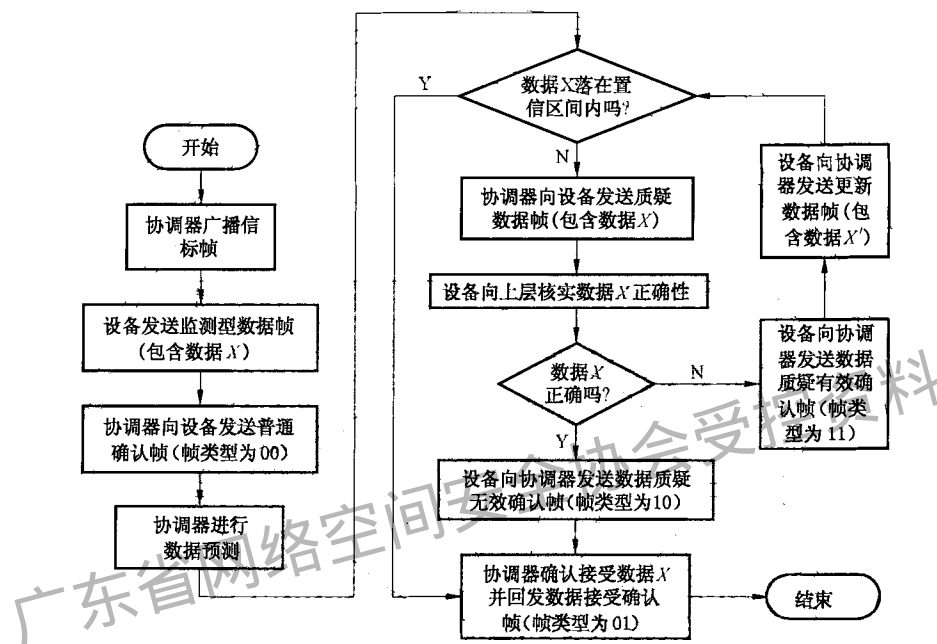


图 71 数据重传机制流程

### 7.5.7.8 非竞争期的重传时隙选择机制

同一超帧的 CFP 被划分为 3 个 SCFP，即 SCFP1、SCFP2 和 SCFP3，SCFP2 用于重新传输节点在 SCFP1 发送失败的数据，SCFP3 用于重新传输节点在 SCFP2 发送失败的数据，且每个节点的 SCFP1 中的时隙顺序和 SCFP2 的时隙顺序相同，即相对于一个节点，SCFP1 中的各时隙顺序和 SCFP2 中的各时隙顺序相同，节点在 SCFP1 的某个时隙发送数据失败，则仍在 SCFP2 中的该某个时隙再次发送数据。由于在 SCFP1 和 SCFP2 种使用相同的时隙重发数据，这样才使得协调器在不同时隙接收到数据时可根据时隙判断该数据是由哪个节点发送的。

节点发送数据失败，有两种情况：一是数据包丢失，二是确认帧丢失。若节点在 SCFP1 内发送数据失败，节点并不立即重传，而是根据接收到的协调器广播的信道划分信息，跳至信道质量较好的约定信道，在 SCFP2 对应时隙再次发送该数据。如果仍发送失败，在 SCFP3 再次发送在 SCFP2 中发送失败的数据。

信道使用列表用于记录所述超帧内各信道上最近一次成功发送数据的节点信息，而在前两个 SCFP 的每个时隙对不同信道质量进行评估，生成最新信道使用列表，根据该最新信道使用列表，节点和协调器间在第三 SCFP 中自适应地切换信道和备用信道，且实现切换的无缝衔接。如图 72 所示。

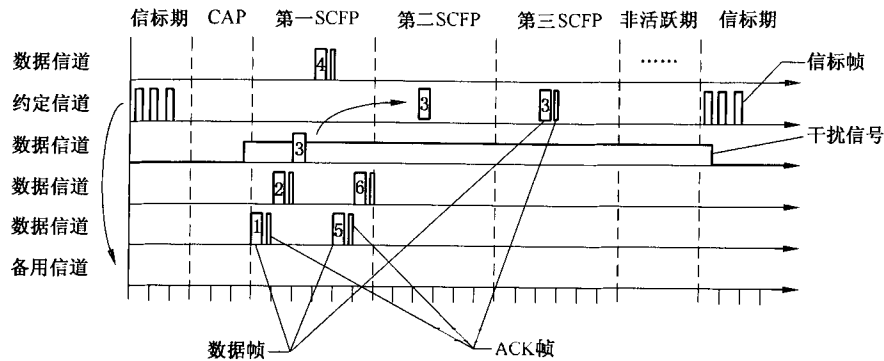


图 72 非竞争期的重传时隙选择机制

## 7.5.8 SCFP 的分配和管理

### 7.5.8.1 概述

一个 SCFP 可使某个设备或若干个设备共享在超帧中的特定时间收发数据。一个 SCFP 由若干个时隙组成。每个时隙长度是根据特定的应用场景来设计的。SCFP 只能由协调器来分配,并且仅用于 RWSN 协调器和设备节点之间的通信,应通过协调器和 RWSN 关联的设备。为保证超帧有足够容量及比较高的利用率,RWSN 协调器利用信标动态调整 SCFP 中的时隙数。

RWSN 协调器根据本超帧的各工作设备在其各自的上一个工作超帧中对本超帧的 SCFP 时隙的申请情况和本超帧当前的容量来分配 SCFP。SCFP 的分配遵循先到先服务的原则和共享的原则,前者用于超帧容量足够时,而后者用于超帧容量不足时。所有 SCFP 将连续地排列在超帧中 CAP 的后面,设备间共享的 SCFP 将由这些设备在连续的超帧中同一 SCFP 期间轮流使用时隙。同时,可以根据 RWSN 协调器或申请 SCFP 时隙的设备的要求,随时调整每个 SCFP 的时隙数。已分配了 SCFP 的设备仍可在 CAP 期间进行收发操作。

在 SCFP 内传输数据帧只能使用短地址。

RWSN 协调器对 SCFP 进行管理。为方便管理,协调器会保存已分配的 SCFP 的必要信息。对于每个 SCFP,协调器都会保存它的起始时隙、长度、方向和关联设备的地址。

SCFP 中时隙的方向,即为发送或接收,与拥有 SCFP 设备的数据流有关。因此,设备的地址和方向就唯一定义了 SCFP 中的一个时隙。

每个设备可能请求一个发送 SCFP 时隙和/或一个接收 SCFP 时隙。对分配到的每个 SCFP 时隙,设备都会保存其起始时隙、长度和方向。如果已分配了一个接收 SCFP 时隙,设备将保持接收机的使能状态,并持续整个时隙。同样地,如果设备已分配了一个发送 SCFP 时隙,RWSN 协调器将在这个 SCFP 时隙打开自己的接收机。如果在接收 SCFP 时隙内收到一个数据帧,并需要确认,设备就发送确认帧。同样地,设备也能在 SCFP 时隙内发送接收确认帧。

只有当设备在跟踪信标时,才会尝试分配并使用一个 SCFP。如果 MLME 收到邻近高层发布的 MLME-SYNC.request 原语,且 TrackBeacon 参数设为 TRUE,就开始跟踪信标帧。如果设备和 RWSN 协调器失去同步,将丢失分配给自己的所有 SCFP 时隙。

SCFP 的使用是可选的。

### 7.5.8.2 CAP 的维护

RWSN 协调器保存参数  $aMinCAPLength$  的值,即最小 CAP 长度,且当无法满足 CAP 的最小长

度时,协调器会采取预防性的措施。然而,也有例外的情况,就是为了保证 SCFP 需要增加信标帧的长度,应缩短 CAP 的时间长度。如果 CAP 的最小长度不能满足,可以采取如下措施:

- a) 限制信标中待处理地址的个数。
- b) 信标帧不包含净荷字段。
- c) 适当调整 SCFP 时隙长度。

### 7.5.8.3 SCFP 时隙的分配

通过原语 MLME-SCFP.request,设备可以在其工作超帧内向协调器申请在该设备的下一个工作周期为其分配一个新 SCFP 时隙,并根据 SCFP 申请原语的要求设置 SCFP 相应的特性。

为了请求分配一个新 SCFP 时隙,SCFP 时隙的请求设备的 MLME 应向 RWSN 协调器发送 SCFP 时隙请求命令(见 7.3.9)。请求命令中 SCFP 特征字段的特征类型子字段设为 1(表示分配 SCFP 时隙),长度和方向子字段根据要求设置,如果请求设备允许与其他设备共享 SCFP,SCFP 特征字段的分配类型子字段设为 1,并需相应地设置突发长度子字段、数据到达速率子字段和时延要求子字段;如果设备不支持共享,分配类型子字段设为 0。SCFP 特征字段还包含信标指数和设备的 MSL,以指示该设备的下一个工作超帧,协调器将保存该申请的内容并在该设备的下一个工作超帧的 SCFP 内为该设备分配时隙。因为 SCFP 请求命令包含确认请求(见 7.3.4.3),RWSN 协调器回复确认帧以确认接收到命令。

收到从某个设备发来的申请 SCFP 时隙的请求命令后,RWSN 协调器首先根据 CAP 剩下的长度和请求 SCFP 时隙的长度,检查该设备的下一个工作超帧内是否有足够的可用容量。如果 SCFP 个数没有达到最大值,且分配请求 SCFP 的长度后不会使 CAP 的长度小于  $aMinCAPLength$ ,超帧中就有足够的可用容量。如果有足够的可用带宽,RWSN 协调器根据先到先服务的原则分配 SCFP:每个申请 SCFP 时隙的设备分别分配到各自独立的 SCFP 时隙。协调器会暂时存储分配信息,并等到该设备的下一个工作超帧开始时,广播含有相应 SCFP 分配信息的信标帧。如果没有足够的可用带宽,RWSN 协调器根据共享的原则分配 SCFP 时隙:首先,RWSN 协调器检查 SCFP 时隙请求命令帧中的分配类型位,若为 0(设备不支持共享),拒绝该 SCFP 申请请求;若为 1(设备支持共享),提取命令帧中的突发长度子字段、数据到达速率子字段和延时要求子字段的信息,然后将这些信息结合当前已分配的 SCFP 信息判断是否满足 SCFP 共享条件,若满足,则该请求设备和其他设备共享 SCFP,最后利用这些信息判断共享设备间的先后使用顺序;若不满足,则拒绝该 SCFP 申请请求。

判断是否满足 SCFP 共享条件的过程是:首先,协调器检测当前已分配 SCFP 的各设备的 SCFP 利用率,然后按照 SCFP 利用率由低到高的顺序选择一个或一个以上的设备作为共享设备;判断 SCFP 利用率最低的设备所分配的 SCFP 中包含的时隙数是否满足 SCFP 请求要求,若满足,则选择该设备为共享设备;否则,判断 SCFP 利用率最低的设备与 SCFP 利用率次低的设备所分配的时隙数之和是否满足 SCFP 请求要求,若满足,则选择上述两个设备为共享设备;若仍不满足,则进一步检测其他设备,直至满足 SCFP 请求的要求。上述判断选择的设备是否满足 SCFP 请求要求的过程是:首先,协调器根据从命令帧中提取的突发长度信息,获取 SCFP 时隙请求设备以及共享设备的服务延迟,并且分别判断发起 SCFP 时隙请求的设备以及共享设备的服务延迟是否小于或等于各自的时延要求;若小于或等于,则进一步判断发起 SCFP 时隙请求的设备以及共享设备的数据到达率之和是否小于共享设备已分配的 SCFP 时隙相对带宽,若小于,则分配发起 SCFP 时隙请求的设备与共享设备共享使用该 SCFP 发送数据。而 RWSN 协调器应在  $aSCFPDescPersistenceTime$  个该设备工作周期的时间内作出决定是否要分配 SCFP。

收到 SCFP 请求命令的确认后,设备将继续跟踪信标,并等待规定的最大时间,即  $aSCFPDescPersistenceTime$  个工作周期。如果这段时间内,信标帧中没有请求设备的 SCFP 描述符,设备的 MLME 就发布原语 MLME-SCFP.confirm,状态为 NO\_DATA(见 7.1.8.3),向邻近高层报告 SCFP 申请失败。

在 RWSN 协调器使用先到先服务原则和共享原则来分配请求的 SCFP 时,会生成 SCFP 描述符,描述符中携带有申请设备的要求信息和 16 比特短地址。某个信标帧所要携带的多个 SCFP 描述符可能分别由协调器在之前多个超帧内接收到的 SCFP 请求生成,上述 SCFP 请求所申请的 SCFP 时隙均位于该信标帧所对应的超帧中,协调器接收到上述 SCFP 请求后会暂时将 SCFP 时隙的分配信息存储下来,并在广播该信标帧之前统一生成各自的 SCFP 描述符并写入该信标帧。如果 SCFP 分配成功,RWSN 协调器将根据超帧中的时隙,在 SCFP 描述符中设置起始时隙和 SCFP 的长度。另外,MLME 发布 MLME-SCFP.indication 原语,向邻近高层报告分配了一个新的 SCFP(见 7.1.8.4),其中原语中携带已分配 SCFP 的特征。如果没有足够容量来分配 SCFP,而经过共享 SCFP 条件判断后 SCFP 分配成功,RWSN 协调器将根据超帧中的共享的时隙,在 SCFP 描述符中设置起始时隙和 SCFP 的长度,此时,共享该 SCFP 时隙的设备轮流占用 SCFP,其先后顺序由 SCFP 描述符中的 16 比特短地址表征。如果 SCFP 分配失败,RWSN 协调器将起始时隙设为 0,长度设置为当前支持的最大 SCFP 时隙长度。RWSN 协调器发送的信标帧将包含这个 SCFP 描述符,并相应地更新信标帧中的 SCFP 规范字段。最后,RWSN 协调器更新信标帧中超帧规范字段的 CAP 最终时隙子字段,指示 CAP 可用的时隙个数。对应某设备的 SCFP 描述符将由该设备的工作信标帧携带,其他信标帧不要求携带此 SCFP 描述符,经过  $aSCFPDescPersistenceTime$  个该设备的工作周期后,此 SCFP 描述符将被自动删除。由于信标帧包含了 SCFP 描述符,导致长度暂时增加,允许 CAP 长度小于  $aMinCAPLength$ 。

当设备收到含有 SCFP 描述符的信标帧,且对应自己的  $macShortAddress$ ,设备就处理该描述符。不论是否成功申请到 SCFP,MLME 都会发布 MLME-SCFP.confirm 原语来向邻近高层报告,状态参数为 SUCCESS(如果 SCFP 描述符的起始时隙大于 0)或 DENIED(如果起始时隙等于 0 或长度与请求的长度不相同)。

如果申请 SCFP 时隙的设备为能源非受限设备,且协调器可以分配给该设备 SCFP,则尽量将分配给该设备的 SCFP 时隙设置在与非活跃时段相邻的位置。相应地,由于新的 SCFP 时隙的分配和位置设置,使得原来分配的 SCFP 时隙的位置都相应发生了变化,所以,协调器应在信标中将相应信息通知相应设备。如果超帧中与非活跃期相邻的 SCFP 时隙所对应的设备已经为能源非受限设备,则只要该 SCFP 时隙没有终止或取消,其他能源非受限设备申请和分配的 SCFP 不能设置在与非活跃期相邻的位置,换句话说,对于能源非受限设备而言,获得与非活跃期相邻的 SCFP 的位置,也遵从先到先得的原则。

#### 7.5.8.4 SCFP 的使用

如果设备不是 RWSN 协调器,当 MAC 层收到 MCPS-DATA.request 原语,参数  $TxOptions$  显示为传输 SCFP,设备就判断自己是否有有效的发送 SCFP 时隙。如果找到一个有效的 SCFP 时隙,则在该 SCFP 时隙内传输数据。这时,为保证数据的传输在 SCFP 时隙结束前完成,MAC 层立即传输 MPDU,且不使用 CSMA-CA 算法。如果不能在 SCFP 时隙结束前完成,MAC 层将推迟到该设备的下一个工作超帧中的 SCFP 规定的时隙内才进行。如果设备为能源非受限设备,且选择发送数据的 SCFP 允许扩展到非活跃期,同样,为保证数据的传输在 SCFP 规定时间结束前完成,MAC 层立即传输 MPDU,且不使用 CSMA-CA 算法,如果不能在 SCFP 规定时间结束前完成,MAC 层将推迟到该设备的下一个工作超帧中的 SCFP 中规定的时隙内才进行。要注意一点,MAC 层在做出推迟发送的决定时,需要考虑 PHY 净荷过重。

在设备发送数据给协调器的情况中,如果参数  $TxOptions$  显示为带数据接受确认的传输,那么,设备在有效的 SCFP 时隙内收到普通确认帧后,只认为在 SCFP 时隙结束前完成了数据的传输,但只有接收到协调器发来的数据接受确认帧之后,才判定成功发送了 MPDU。

如果设备有接收 SCFP,MAC 层应在 SCFP 时隙开始前打开接收机,并持续到接收 SCFP 时隙结束。

如果设备需要接收时隙,MAC 层应在时隙开始前打开接收机,并持续到接收时隙结束。

当 RWSN 协调器的 MAC 层收到 MCPS-DATA.request 原语,参数 *TxOptions* 显示为传输时隙,协调器要判断是否有对应于请求设备的有效接收 SCFP。如果有这样的 SCFP,RWSN 协调器推迟传输到接收 SCFP 开始。这时,申请 SCFP 时隙的设备的地址不会添加到信标帧的待处理地址列表(见 7.5.6)。在接收 SCFP 时隙开始时,为保证能在 SCFP 时隙结束之前完成数据发送,MAC 层将不使用 CSMA-CA 算法。如果不能在 SCFP 时隙结束之前完成,MAC 层将传输推迟到该设备的下一个工作超帧中的 SCFP 规定的时隙内才进行。

对于所有已分配的发送 SCFP(与设备相关),RWSN 协调器的 MAC 层应在 SCFP 开始前打开接收机,并在整个 SCFP 时隙期间保持使能状态。对于已分配的发送 SCFP(与设备相关),RWSN 协调器的 MAC 层应在与 SCFP 时隙对应的时隙开始前打开发射机,并在整个 SCFP 时隙期间保持使能状态。

对于要在 SCFP 时隙中收发所有设备,设备基于所述信道选择信息及所述时隙划分信息发送的数据,且应确保能够在 SCFP 时隙结束前完成数据传输,接收到相应的确认帧(如果要求)以及与数据帧大小匹配的 IFS。对于要在 SCFP 中收发所有设备,应在相应的 SCFP 时隙开始前确保能够在相应的 SCFP 时隙结束前完成数据传输,接收到相应的确认帧(如果要求)以及与数据帧大小匹配的 IFS。

如果设备错过了信标帧,就不能使用 SCFP 时隙,直至再次正确收到信标。如果丢失信标帧而导致失去同步,设备就认为分配给自己的 SCFP 时隙将被全部取消。

### 7.5.8.5 SCFP 时隙的取消

#### 7.5.8.5.1 概述

设备通过 MLME-SCFP.request 原语(见 7.1.8.2)来取消已分配的 SCFP,原语携带其取消的 SCFP 的特征。一旦原语产生,设备就不能再使用该 SCFP 时隙,其保存的特征也将被复位。

为了请求取消已分配的 SCFP,MLME 将向 RWSN 协调器发送 SCFP 时隙请求命令(见 7.3.9)。请求命令中 SCFP 特征字段的特征类型子字段设为 0(表示取消 SCFP),长度和方向子字段将根据要取消的 SCFP 进行设置。SCFP 请求命令包含确认请求(见 7.3.4.3),RWSN 协调器向请求设备回复确认帧。收到 SCFP 请求命令的确认帧后,MLME 就通知邻近高层:发布原语 MLME-SCFP.confirm(见 7.1.8.3),状态参数为 SUCCESS,SCFP 特征参数的特征类型子字段设为 0。如果 RWSN 协调器没有正确收到 SCFP 请求命令,协调器将判断设备已经采用 7.5.8.5.3 的机制中止使用 SCFP。

收到 SCFP 请求命令,其中 SCFP 特征字段的特征类型子字段设为 0(表示取消 SCFP),RWSN 协调器将试图取消已分配的 SCFP。如果请求命令中的 SCFP 特征与已知 SCFP 的特征不匹配,协调器将忽略该请求。若匹配,协调器的 MLME 将取消已分配的 SCFP 时隙,并通知邻近高层,即发布 MLME-SCFP.indication 原语(见 7.1.8.4),其中 SCFP 特征参数包含所取消 SCFP 的特征,且特征类型子字段为 0。此外,协调器还要更新信标帧中超帧规格字段的 CAP 最终时隙子字段,且不必在信标帧中添加描述符来表示 SCFP 的取消。

RWSN 协调器也可以发起 SCFP 的取消,一般的情况是协调器的邻近高层产生取消 SCFP 的请求,或是 SCFP 到期(见 7.5.8.5.3),也可以是为了保证 CAP 最小长度  $aMinCAPLength$ (见 7.5.8.2)。

若是协调器的邻近高层发起 SCFP 的取消,MLME 会收到 MLME-SCFP.request 原语,请求原语的 SCFP 特征字段设为 0(即取消 SCFP),长度和方向子字段将根据要取消的 SCFP 来设置。

当 SCFP 到期或为了维护 CAP,RWSN 协调器要求取消 SCFP,且 MLME 向邻近高层报告:发布 MLME-SCFP.indication 原语,SCFP 特征参数包含要取消的 SCFP 的特征,并且特征类型子字段设为 0。

当由 RWSN 协调器发起取消 SCFP,协调器取消 SCFP 后,还会在信标帧中添加 SCFP 描述符,但起始时隙设置为 0。每个 SCFP 的描述符都由该描述符所对应的设备的工作信标帧携带,携带次数是

$aSCFPDescPersistenceTime$  次,达到这个次数后,协调器会移除该 SCFP 描述符。由于添加了 SCFP 描述符导致信标帧长度暂时增加,所以协调器允许 CAP 的长度小于  $aMinCAPLength$ 。

收到的信标帧包含对应  $macShortAddress$  的 SCFP 描述符,且开始间隙为 0,设备立即停止使用 SCFP 时隙。设备的 MLME 会通知邻近高层取消 SCFP,发布 MLME-SCFP.indication 原语,SCFP 特征参数包含要取消的 SCFP 的特征,且特征类型子字段设为 0。

### 7.5.8.5.2 SCFP 的重新分配

SCFP 的取消可能会导致超帧变成几个碎片。图 73 是含有 SCFP 的超帧结构变化图。在阶段 1 中,超帧中分配了 3 个 SCFP,起始时隙分别是 30、41 和 52。在阶段 2 中,如果取消 SCFP3,超帧中将出现一个无用的缺口。为解决这类问题,应移动 SCFP1 和 SCFP2 来填补这个缺口,并增加了 CAP 的大小(阶段 3)。

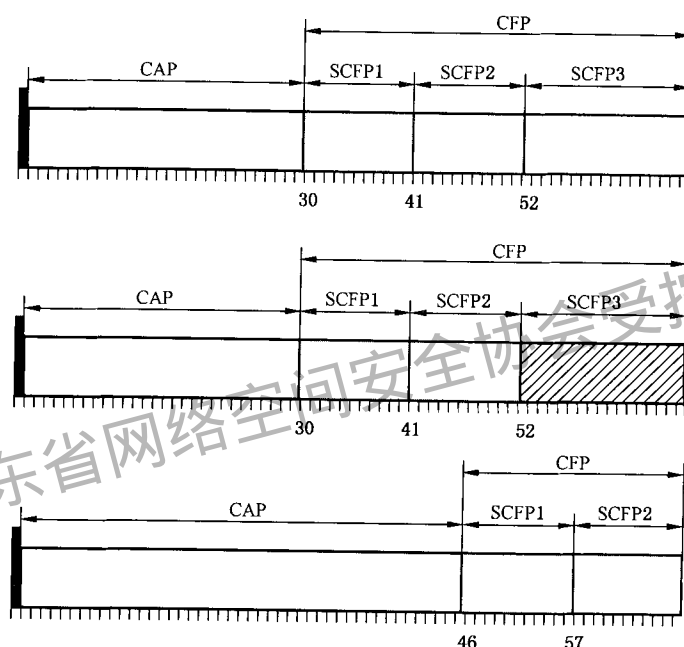


图 73 取消 SCFP 后的 CFP 碎片整理

由于取消 SCFP3 导致 CFP 中出现缺口,所以 RWSN 协调器应移除缺口来增加 CAP 的长度。

当 RWSN 协调器发起 SCFP 的取消,协调器将在信标帧中添加一个 SCFP 描述符,表明该 SCFP 已被取消。如果由设备发起 SCFP 的取消分配,RWSN 协调器就不会在信标帧中添加 SCFP 描述符。对于每个分配有 SCFP 的设备,如果其 SCFP 起始时隙小于要被取消 SCFP 的起始时隙,RWSN 协调器更新这些 SCFP,设置新的起始时隙,并针对所作的调整在信标帧中添加一个 SCFP 描述符。需要精确计算新的起始时隙,以保证 CFP 的末端没有缺口。

当同时有多个 SCFP 的重新分配,RWSN 协调器会按步骤来完成。每个 SCFP 的描述符都由该描述符所对应的设备的工作信标帧携带,携带次数是  $aSCFPDescPersistenceTime$  次,达到这个次数后,协调器会移除该 SCFP 描述符。

设备收到信标帧后,若发现包含的 SCFP 描述符对应自己的  $macShortAddress$ ,并且方向和长度与自己拥有的某个 SCFP 相匹配,设备就根据 SCFP 描述符的信息调整该 SCFP 的起始时隙,并立即使用。

当在信标帧中添加一个 SCFP 描述符时,允许 RWSN 协调器减少 CAP 长度并小于  $aMinCAPLength$ ,来适应信标帧长度的暂时性增加。

### 7.5.8.5.3 SCFP 的终止

RWSN 协调器的 MLME 能够检测到这一情况,即设备采用以下规则停止了一个 SCFP 的使用:

- a) 对于发送 SCFP,如果在 $(2 * n * \text{对应设备的 MSL 值})$ 个超帧内没有收到设备发送的数据帧, RWSN 协调器的 MLME 就判定设备已经停止使用该 SCFP;
- b) 对于接收 SCFP,如果在 $(2 * n * \text{对应设备的 MSL 值})$ 个超帧内没有收到设备发送的确认帧, RWSN 协调器的 MLME 就判定设备已经停止使用该 SCFP。如果在 SCFP 中发送的数据帧不需要确认, RWSN 协调器的 MLME 就无法检测到设备是否在使用接收 SCFP。然而, RWSN 协调器可以随时取消已分配的 SCFP。

上述变量  $n$  的定义是:

$$n = 2^{(8 - \text{macBeaconOrder})} \quad 0 \leq \text{macBeaconOrder} \leq 8$$

$$n = 1 \quad 9 \leq \text{macBeaconOrder} \leq 14$$

## 7.5.9 低功耗模式

### 7.5.9.1 概述

若网络中的数据传输主要为从终端设备到协调器,网络可以设置为 LE 模式。

当 MIB 属性 *macLEenabled* 设为 TRUE 时,LE 模式被激活。LE 模式的基本交互如图 74 所示。

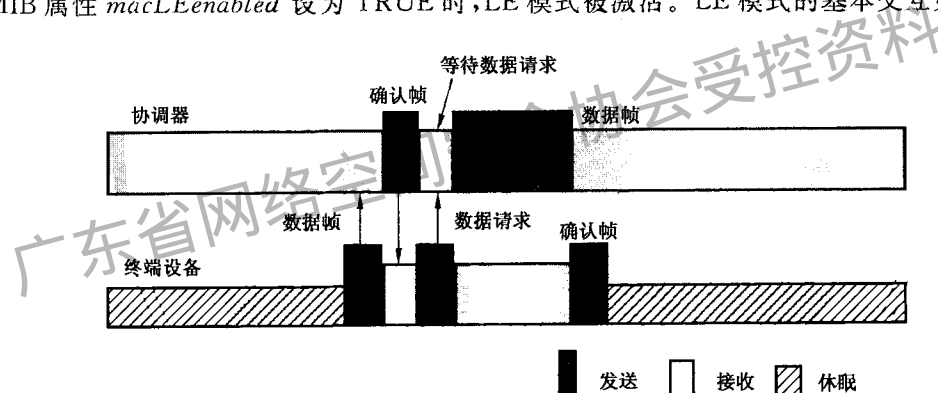


图 74 低功耗模式示意图

当网络设置为 LE 模式时,协调器保持信道侦听,终端设备若无数据需要发送,则保持休眠状态,若需要发送数据,则通过 CSMA 或 Aloha 机制发送数据帧。终端设备发送完毕切换到接收状态,等待协调器的应答帧。

如果终端设备没有接收到协调器的应答帧,且 *macMaxFrameRetries* 不为零,则执行数据重发机制。

如果终端设备接收到协调器的应答帧,并且应答帧对应指示位标志协调器有数据需要发送给本终端设备,则终端设备向协调器发送数据请求帧。否则切换至休眠状态,直到下一次需要发送数据帧时为止。

终端设备发送数据请求帧后,若未接收到协调器的应答帧,且 *macMaxFrameRetries* 不为零,则重发数据请求帧,直到重发 *macMaxFrameRetries* 次为止。若接收到协调器的应答帧,则维持信道侦听,等待协调器发送对应的数据帧。协调器发送的数据帧中,帧控制域的帧持续标志位用于指示接收数据帧的终端设备,在接收完此帧后,是继续维持信道侦听状态,还是在此帧接收完成后切换到休眠状态。



## 7.5.9.2 自动唤醒机制

### 7.5.9.2.1 概述

自动唤醒机制由 MAC 层提供的一种服务,由协调器发送唤醒帧,终端设备周期侦听信道,若接收到唤醒帧,即结束休眠状态的一种机制。可以采用混合唤醒采样侦听机制(HWSL 机制)或同步侦听唤醒机制(SLW 机制)。

### 7.5.9.2.2 HWSL 机制

#### 7.5.9.2.2.1 概述

当协调器发送给终端设备的数据帧较少,或对数据延时的要求不高时,协调器应等到接收到对应终端设备的数据请求帧,才能发送数据帧给该设备。

LE 模式中,网络是否支持 HWSL 唤醒机制通过 MIB 属性 *macHWSLSupport* 进行配置。

HWSL 唤醒机制的基本操作如图 75 所示。

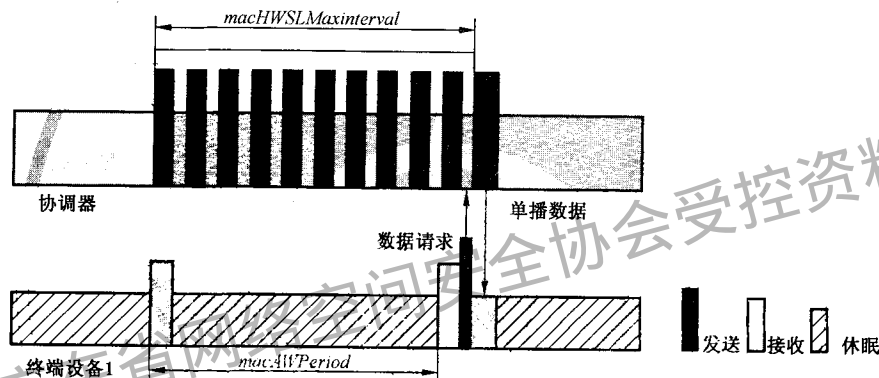


图 75 HWSL 唤醒机制示意图

协调器发送延时要求较高的数据帧时,在发送数据帧之前,首先发送 AW 唤醒序列。HWSL 唤醒序列由连续的 AW 唤醒帧构成,连续两个唤醒帧之间的间隔为  $macHWSLWakeupInterval$ 。协调器发送连续两个唤醒帧之间,切换到接收状态,侦听信道。唤醒帧序列的长度为  $macHWSLMaxInterval$ 。

支持 HWSL 唤醒机制的终端设备,在没有数据帧发送时保持休眠状态,同时维持采样间隔为  $macAWPeriod$  的信道侦听,如果每次侦听信道没有侦听到协调器发送的唤醒帧,终端设备切换至休眠状态,直到下一次采样侦听时刻。

#### 7.5.9.2.2.2 单播数据传输

如果协调器通过 HWSL 机制和终端设备进行数据传输的帧为单播帧,唤醒帧的协调器地址应设为该网络协调器的地址,目的地址应设为对应的终端设备地址。

接收到单播唤醒帧的终端设备,首先判断唤醒帧的协调器地址是否为本网络协调器的地址,如果是,再判断目的地址是否为本设备,如果地址匹配,则通过原语告知上层需要终止周期采样侦听。然后向协调器发送响应消息,若终端设备请求协调器发送数据,则响应消息为数据请求帧;若终端设备从休眠状态进入唤醒状态,将与协调器进行数据传输,包括向协调器发送数据,则响应消息为确认帧,协调器根据响应消息与终端设备进行单播数据传输。

如果终端设备在发送响应消息后,仍然收到协调器发送的单播唤醒帧,且唤醒帧的目的地址为本设备,则重新发送响应消息。

协调器发送完单播唤醒帧后,若接收到对应终端设备的响应消息,则立即与对应终端设备进行单播

数据传输。此单播数据帧应要求接收设备收到后回复应答,接收设备等待的最长应答时间为  $macAckWaitDuration$ ,否则根据  $macMaxFrameRetries$  的设定,判定是否重发数据帧。

如果协调器有多个单播数据帧需要发送给同一个终端设备,协调器在发送给终端设备的所有帧中,除最后一个帧外,MAC 层帧头的帧控制域中的帧持续标志位均置为 1。

### 7.5.9.2.2.3 广播数据传输

如果协调器通过 HWSL 机制与终端设备进行数据传输的帧为广播帧,协调器仍发送间隔为  $macHWSLWakeupInterval$  的唤醒帧序列,唤醒帧的目的地址设为广播地址,并且通过唤醒帧携带广播帧发送的相关时间信息。唤醒机制广播帧发送示意图如图 76 所示。

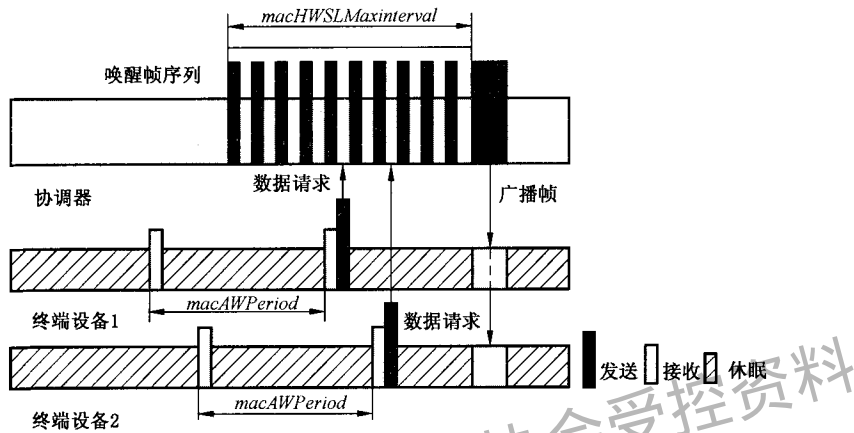


图 76 唤醒机制广播帧发送示意图

终端设备周期信道采样侦听时,若接收到广播唤醒帧,则通过原语请求上层停止周期信道采样侦听,并向协调器发送数据请求帧,然后进入休眠状态,直到广播唤醒帧中携带的信息所指示的广播帧发送时刻到来为止,此时终端设备开启接收机,接收对应的广播帧。

如果协调器发送广播唤醒帧的间隔内接收对应终端设备的数据请求帧,若接收到所有终端设备的数据请求帧,则停止发送唤醒帧,否则唤醒序列的持续时间为  $macHWSLMaxInterval$ 。

协调器在广播唤醒帧中携带的信息所指定的时间发送相应的广播帧。

### 7.5.9.2.3 SLW 机制

#### 7.5.9.2.3.1 概述

网络可以选择自动唤醒机制,LE 模式中,网络是否支持 SLW 机制通过 MIB 属性  $macSLWSupport$  进行配置。SLW 机制的基本操作如图 77 所示。

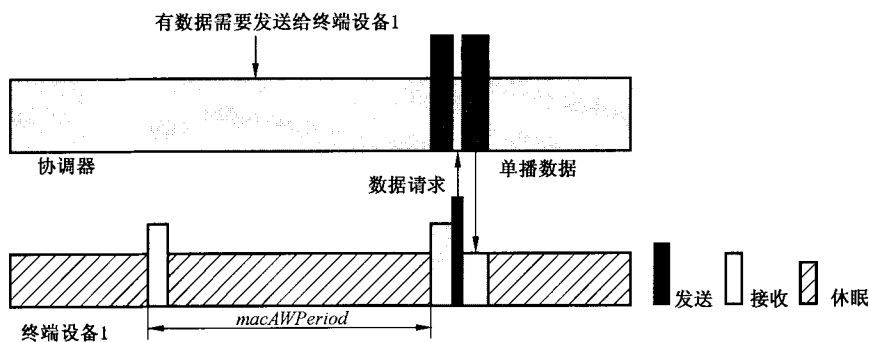


图 77 SLW 唤醒机制示意图

终端设备根据自己的预设侦听时间与协调器实现同步,预设侦听时间可以是终端设备在部署时,在终端设备上设置的侦听唤醒帧的侦听时间,或者,终端设备在部署后,由协调器在超帧的活跃期为终端设备设置的侦听唤醒帧的侦听时间。当协调器需要发送延时要求较高的数据帧时,在该设备最近的预设侦听时间到来时发送唤醒帧。协调器发送唤醒帧之后立即切换到接收状态,侦听信道。

支持 SLW 机制的终端设备,在没有数据帧需要发送时保持休眠状态,同时维持采样间隔为  $macAWPeriod$  的信道侦听,如果每次侦听信道没有侦听到协调器发送的唤醒帧,终端设备切换至休眠状态,直到下一次采样侦听时刻。

### 7.5.9.2.3.2 单播数据传输

如果协调器通过 SLW 机制与终端设备进行数据传输的帧为单播帧,唤醒帧的目的地址应设为对应的终端设备地址。

接收到单播唤醒帧的终端设备,首先判断唤醒帧的协调器地址是否为本网络协调器的地址,如果是,再判断目的地址是否为本设备,如果地址匹配,则通过原语告知上层需要终止周期采样侦听。然后向协调器发送响应消息,若终端设备请求协调器发送数据,则响应消息为数据请求帧;若终端设备从休眠状态进入唤醒状态,将与协调器进行数据传输,包括向协调器发送数据,则响应消息为确认帧,协调器根据响应消息与终端设备进行单播数据传输。如果地址不匹配,设备继续转入休眠状态。

如果终端设备在发送响应消息后,仍然收到协调器发送的单播唤醒帧,且唤醒帧的目的地址为本设备,则重新发送响应消息。

协调器发送完单播唤醒帧后,若接收到对应终端设备的响应消息,则立即与对应终端设备进行单播数据传输。此单播数据帧应要求接收设备收到后回复应答,接收设备等待的最长应答时间为  $macAckWaitDuration$ ,否则根据  $macMaxFrameRetries$  的设置,判定是否重发数据帧。

如果协调器有多个单播数据帧需要发送给同一个终端设备,协调器在发送给终端设备的所有帧中,除最后一个帧外,MAC 层帧头的帧控制域中的帧持续标志位均置为 1。

### 7.5.9.2.3.3 广播数据传输

如果协调器通过 SLW 机制与终端设备进行数据传输的帧为广播帧,协调器仍在终端设备预设侦听时间发送唤醒帧,唤醒帧的目的地址设为广播地址,并且通过唤醒帧携带广播帧发送的相关时间信息。唤醒机制广播帧发送示意图如图 78 所示。

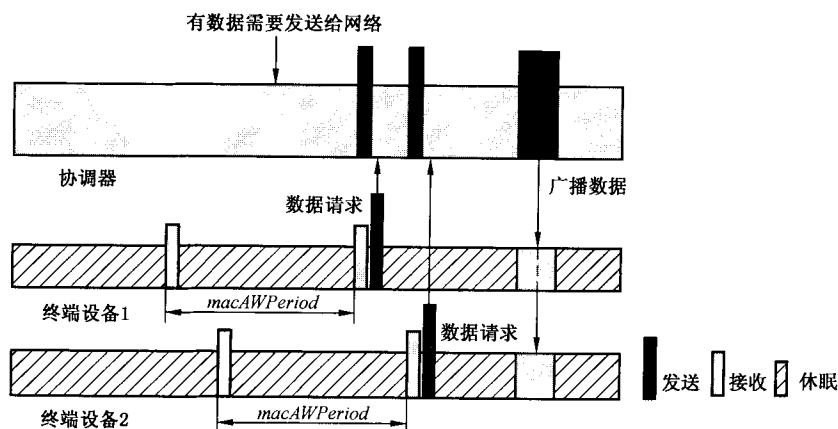


图 78 唤醒机制广播帧发送示意图

终端设备周期信道侦听时,若接收到广播唤醒帧,则通过原语请求上层停止周期信道侦听,并向协调器发送数据请求帧,然后进入休眠状态,直到广播唤醒帧中携带的信息所指示的广播帧发送时刻到来

为止,此时终端设备开启接收机,接收对应的广播帧。

协调器发送广播唤醒帧完成后接收对应终端设备的数据请求帧,若接收到所有终端设备的数据请求帧,则停止发送唤醒帧,否则广播发送时刻到来时则停止发送唤醒帧。

协调器在广播唤醒帧中携带的信息所指定的时间发送相应的广播帧。

### 7.5.9.3 射频唤醒机制

#### 7.5.9.3.1 概述

射频唤醒机制是由 PHY 和 MAC 层共同提供的一种服务,由协调器先后发送射频脉冲序列和唤醒帧,终端设备感应射频脉冲而被唤醒,并接收和解析唤醒帧,进而结束休眠状态。

LE 模式中,网络是否支持 RFW 唤醒机制通过 MIB 属性 *macRFWSupport* 进行配置。

RFW 机制的基本操作如图 79 所示。

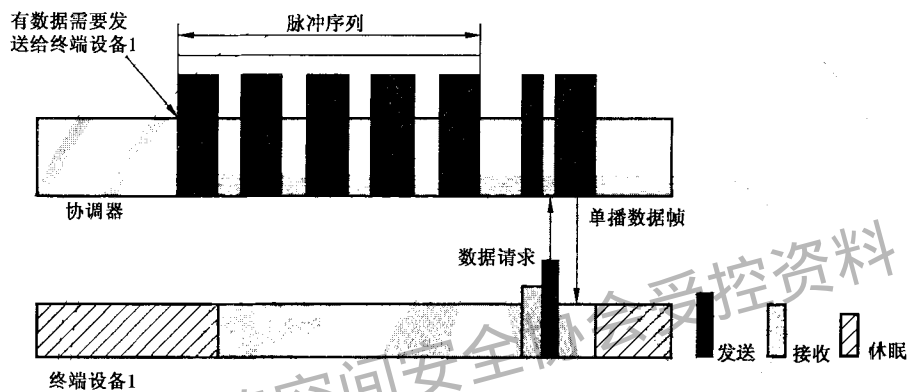


图 79 RFW 唤醒机制示意图

网络可以预先配置唤醒因子,预设唤醒因子用于确定预唤醒设备所属的 RWSN 网络,不同 RWSN 网络具有不同的预设唤醒因子。协调器发送延时要求较高的数据帧时,在发送数据帧之前,协调器发送射频脉冲序列。射频脉冲序列由连续的电磁脉冲信号构成,每个射频脉冲为脉宽大于等于  $macRFWminpulsewidth$  的持续高电平信号,连续两个射频脉冲之间的间隔为  $macRFWpulseInterval$ 。协调器发送射频脉冲序列的个数为该协调器所在 RWSN 网络预设唤醒因子个数  $macRFWWakeupfactornum + 1$ 。射频唤醒序列发送完毕之后,等待大于  $macRFWpulseInterval$  的时间间隔发送唤醒帧。

支持 RFW 唤醒机制的终端设备,在保持休眠状态时,射频天线可以通过继电器模块与唤醒模块相连,随时感应协调器发送的射频唤醒脉冲序列,若感应到射频唤醒脉冲,则产生中断信号使终端设备处理器模块从休眠状态转为唤醒,并对后续感应到的射频唤醒脉冲序列计数,若脉冲间隔大于  $macRFWpulseInterval$ ,则停止计数,并比较计数值与该 RWSN 网络的预设唤醒因子个数  $macRFWWakeupfactornum$ ,若不等,则 RWSN 内终端设备将直接切换至休眠状态;若相等,则终端设备开启无线通信模块,并通过继电器模块使射频天线与无线通信模块相连接接收并解析唤醒帧。

#### 7.5.9.3.2 单播数据传输

如果协调器通过 RFW 机制与终端设备进行数据传输的帧为单播帧,唤醒帧的目的地址应设为对应的终端设备地址。

接收到单播唤醒帧的终端设备,首先判断唤醒帧的目的地址是否为本设备,如果地址匹配。则向协调器发送响应消息,若终端设备请求协调器发送数据,则响应消息为数据请求帧;若终端设备从休眠状态进入唤醒状态,将与协调器进行数据传输,包括向协调器发送数据,则响应消息为确认帧,协调器根据

响应消息与终端设备进行单播数据传输。

如果终端设备在发送响应消息后,仍然收到协调器发送的单播唤醒帧,且唤醒帧的目的地址为本设备,则重新发送响应消息。

协调器发送完单播唤醒帧后,若接收到对应终端设备的响应消息,则立即与对应终端设备进行单播数据传输。此单播数据帧应要求接收设备收到后回复应答,接收设备等待的最长应答时间为 *macAckWaitDuration*,否则根据 *macMaxFrameRetries* 的设定,判定是否重发数据帧。

如果协调器有多个单播数据帧需要发送给同一个终端设备,协调器在发送给终端设备的所有帧中,除最后一个帧外,MAC 层帧头的帧控制域中的帧持续标志位均置为 1。

### 7.5.9.3.3 广播数据传输

如果协调器通过 RFW 机制与终端设备进行数据传输的帧为广播帧,协调器射频唤醒序列发送完成之后发送唤醒帧,唤醒帧的目的地址设为广播地址,并且通过唤醒帧携带广播帧发送的相关时间信息。唤醒机制广播帧发送示意图如图 80 所示。

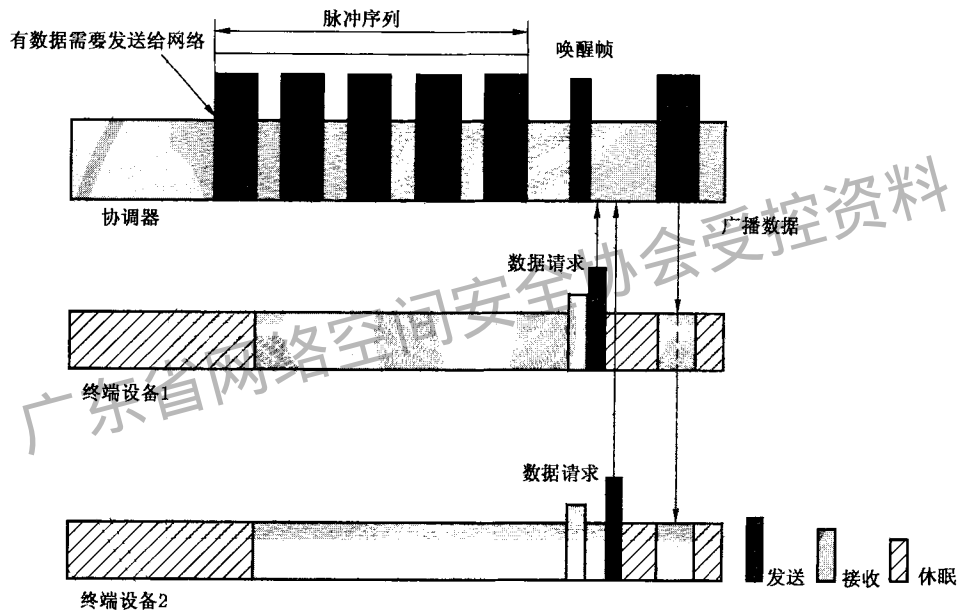


图 80 唤醒机制广播帧发送示意图

终端设备感应并接收到唤醒因子对应数目的射频唤醒脉冲序列后,若接收到广播唤醒帧,则通过 CSMA-CA 机制接入信道,向协调器发送响应消息,若终端设备请求协调器发送数据,则响应消息为数据请求帧;若终端设备从休眠状态进入唤醒状态,将与协调器进行数据传输,包括向协调器发送数据,则响应消息为确认帧,协调器根据响应消息与终端设备进行数据传输。终端设备响应消息发送完毕后进入休眠状态,直到广播唤醒帧中携带的信息所指示的广播帧发送时刻到来为止,此时终端设备开启接收机,与协调器进行数据传输。

协调器发送广播唤醒帧完成后接收对应终端设备的响应消息,若接收到所有终端设备的响应消息,则停止发送唤醒帧,否则广播发送时刻到来时则停止发送唤醒帧。

协调器在广播唤醒帧中携带的信息所指定的时间发送相应的广播帧。

### 7.5.10 设备的工作周期

工作周期是指在信标使能的 RWSN 中,设备相邻两次以侦听信标帧为目的而打开接收机的间隔时间,在一个工作周期内,设备仅侦听工作周期开始时的第一个信标帧,其余时间不侦听信标帧;某设备的

工作周期时长  $WPL = [aBaseSuperframeDuration * 2BO] * \text{该设备的MSL值}$ , MSL 为 RWSN 协调器分配给该设备的超帧时长整数,  $1 \leq MSL \leq aMaxMSL$ 。决定设备工作周期时长的 MSL 值由协调器根据网络和该设备的性能指标通过某种算法计算得到, 其中设备的性能指标可以包括节点自身能量剩余能量、收发数据吞吐量、数据上报频率和报警时延等其中的一项或多项。具体算法不在本协议规定范围内。设备默认的工作周期时长为  $aBaseSuperframeDuration * 2BO$ , 即 MSL 默认时为 1。

若一个设备不是协调器, 则在该设备与其所处的 RWSN 关联成功后, RWSN 协调器将广播含有该设备的工作周期信息的信标帧。设备的工作周期信息位于信标帧 MAC 负荷的周期分配字段, 其中包含信标指数 BO、该设备的短地址和分配给该设备的超帧时长倍数 MSL, 则该设备的工作周期时长  $WPL = [aBaseSuperframeDuration * 2^{BO}] * MSL$ , MSL 的数值由协调器的邻近上层决定。从设备接收到含有其工作周期信息的信标帧的时刻算起, 经过一个超帧时长  $[aBaseSuperframeDuration * 2^{BO}]$  后, 该设备进入工作周期时长为 WPL 的工作周期循环, 同时, 该设备打开接收机以侦听进入新的工作周期循环后的第一个信标帧。进入工作周期时长为 WPL 的工作周期循环的设备, 每隔 WPL 打开一次接收机以跟踪信标帧, 并与 RWSN 协调器同步。该设备侦听到第一个信标帧后, 随即进入本信标帧对应的超帧阶段, 该超帧的 SCFP 阶段结束时, 设备关闭接收机进入休眠状态。之后, 该设备保持休眠状态, 直到下一个工作周期开始再打开接收机以跟踪信标帧, 期间忽略  $(MSL - 1)$  个信标及其对应超帧阶段。

图 81 是设备工作周期的示意图。协调器通过广播信标为设备 A、B、C 和 D 分配工作周期 WPL, 其时长为超帧时长的倍数。协调器广播信标帧之前, 先将已存储的各设备的 NWBSN 值与要发送的信标帧的序列号比对, 数值相符的设备即为下一个超帧的工作设备。协调器仅在要发送的信标帧中安排与上述工作设备相关的事务, 如数据收发。协调器可根据网络性能的需要更新各设备的工作周期, 具体的时长选择算法超出了本规范的范围。

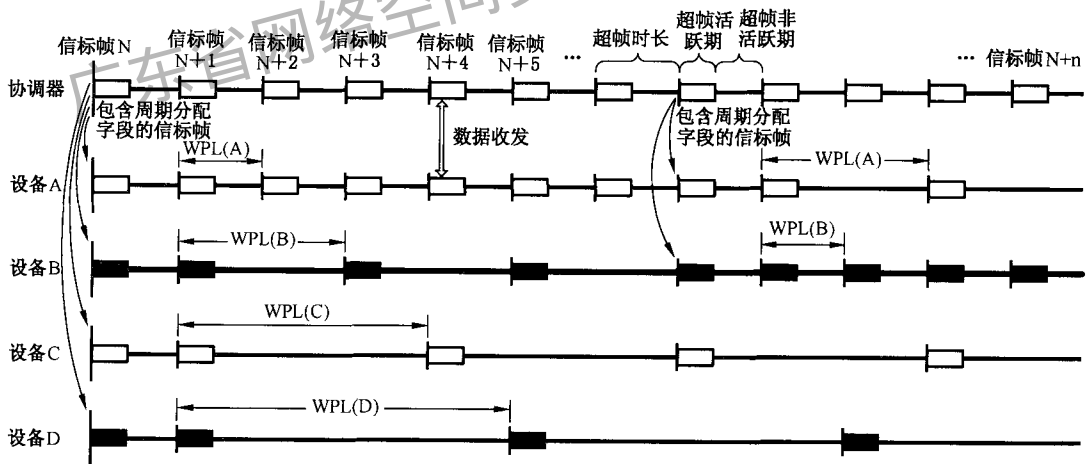


图 81 设备的工作周期示意图

### 7.6 MAC 层和 PHY 交互的消息序列图

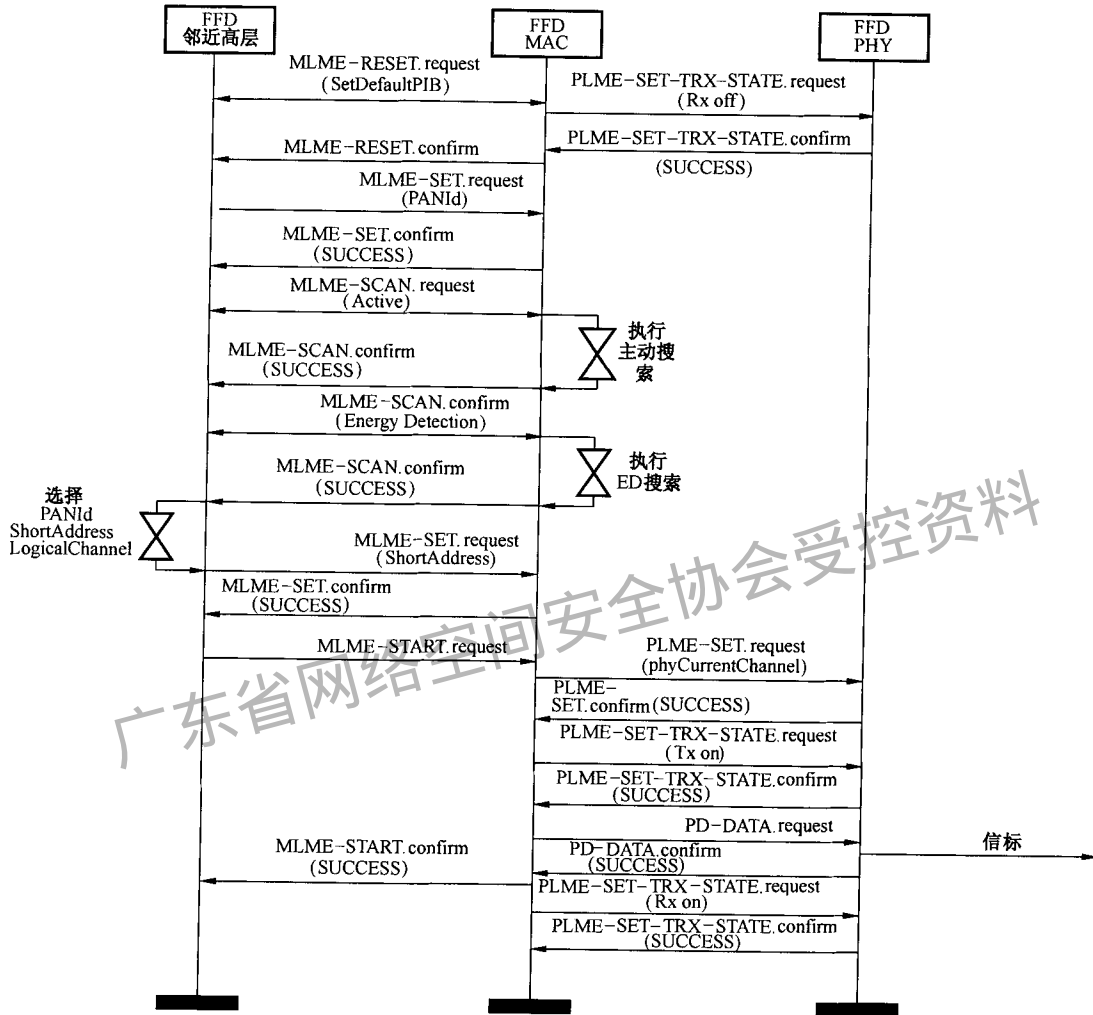
本条举例说明了本规范规定的主要任务。通过消息序列图定义每一个任务, 且图例详细展示了各个步骤。

RWSN 协调器构造一个新的 RWSN 用到的相关消息序列见图 82。在复位 MAC 层后, 邻近高层发起扫描以搜索在区域内的其他 RWSN。当采用主动扫描时, 可选择是否执行 ED 扫描。主动扫描和 ED 扫描的具体步骤如图 87 和图 83 所示。

建立起新的 RWSN 后, RWSN 协调器就开始接收其他设备的加入请求。设备请求关联时的所需

的原语见图 84,而协调器允许关联的步骤见 7.5.4.1。在加入 RWSN 过程中,请求关联的设备执行主动或被动信道扫描,以确定该区域的哪一个 RWSN 允许关联。图 86 和图 87 分别展示出被动扫描和主动扫描所需的原语。

设备传输数据包的步骤如图 88 所示,而接收设备的接收过程如图 89 所示。



注 1: ED 扫描是可选的。

注 2: MLME-START.confirm 和 PLME-SET-TRX-STATE.request 的顺序并不总是如图所示。

图 82 RWSN 起始消息序列图——RWSN 协调器

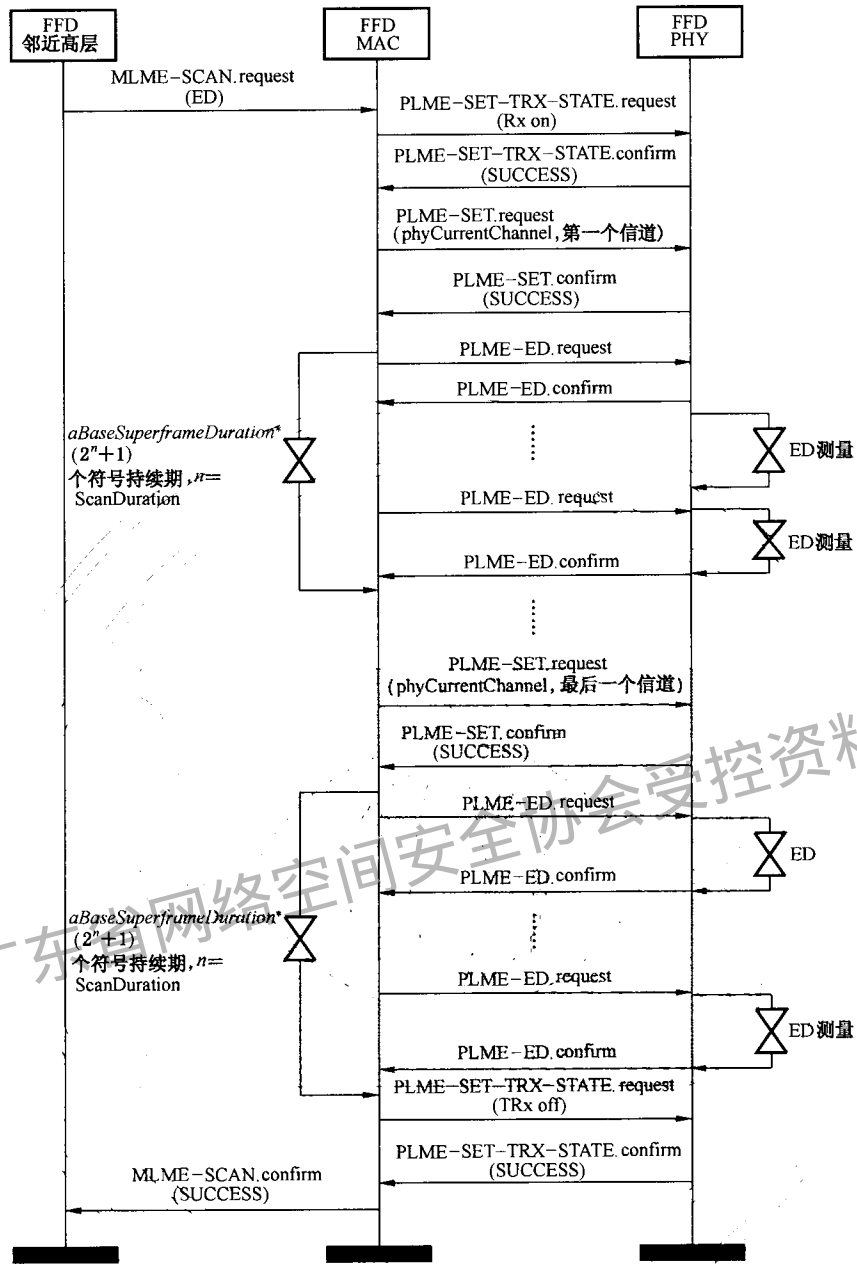


图 83 ED 扫描消息序列图



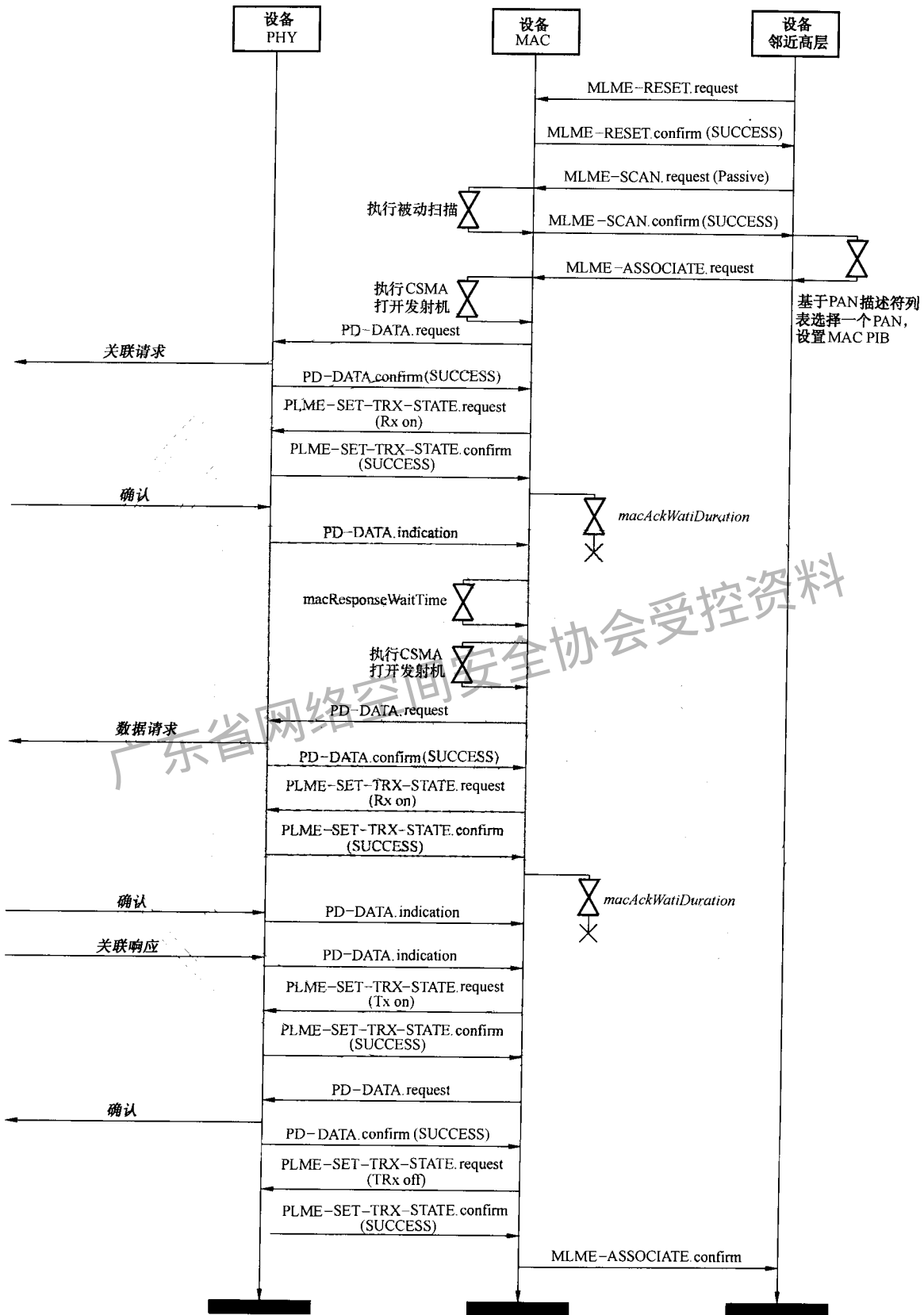


图 84 关联过程的消息序列图——设备

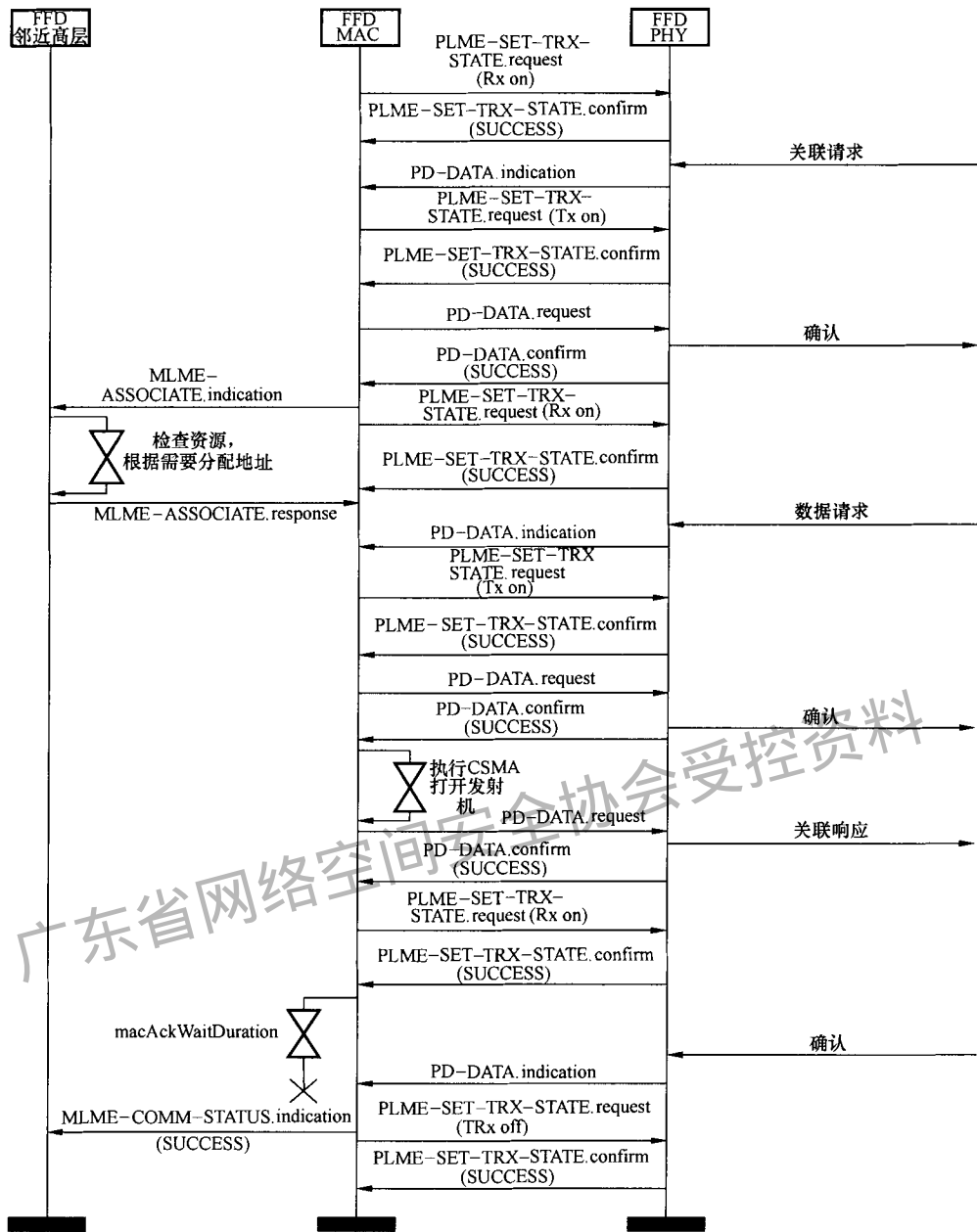


图 85 关联过程消息序列图——协调器

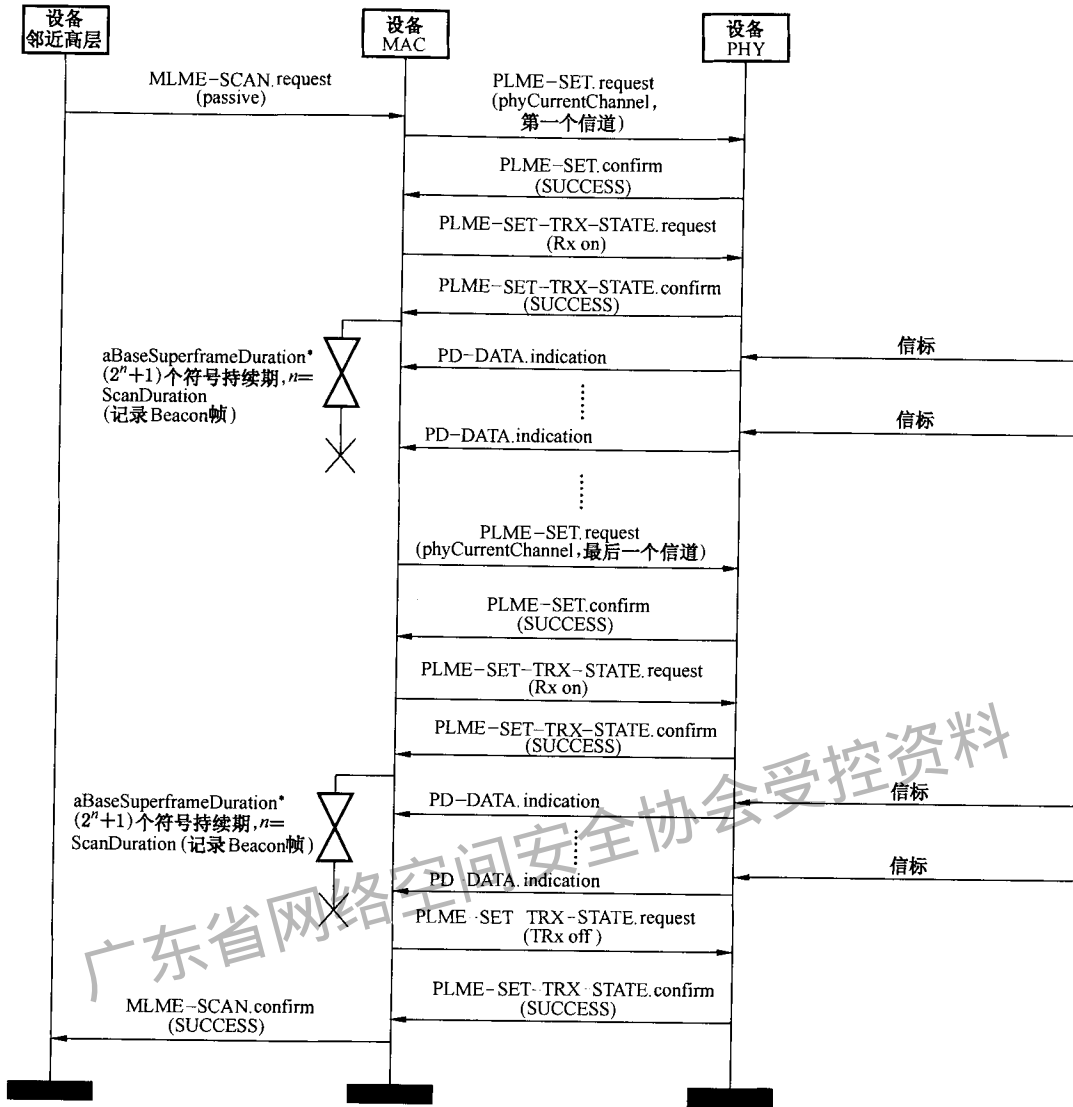


图 86 被动扫描消息序列图

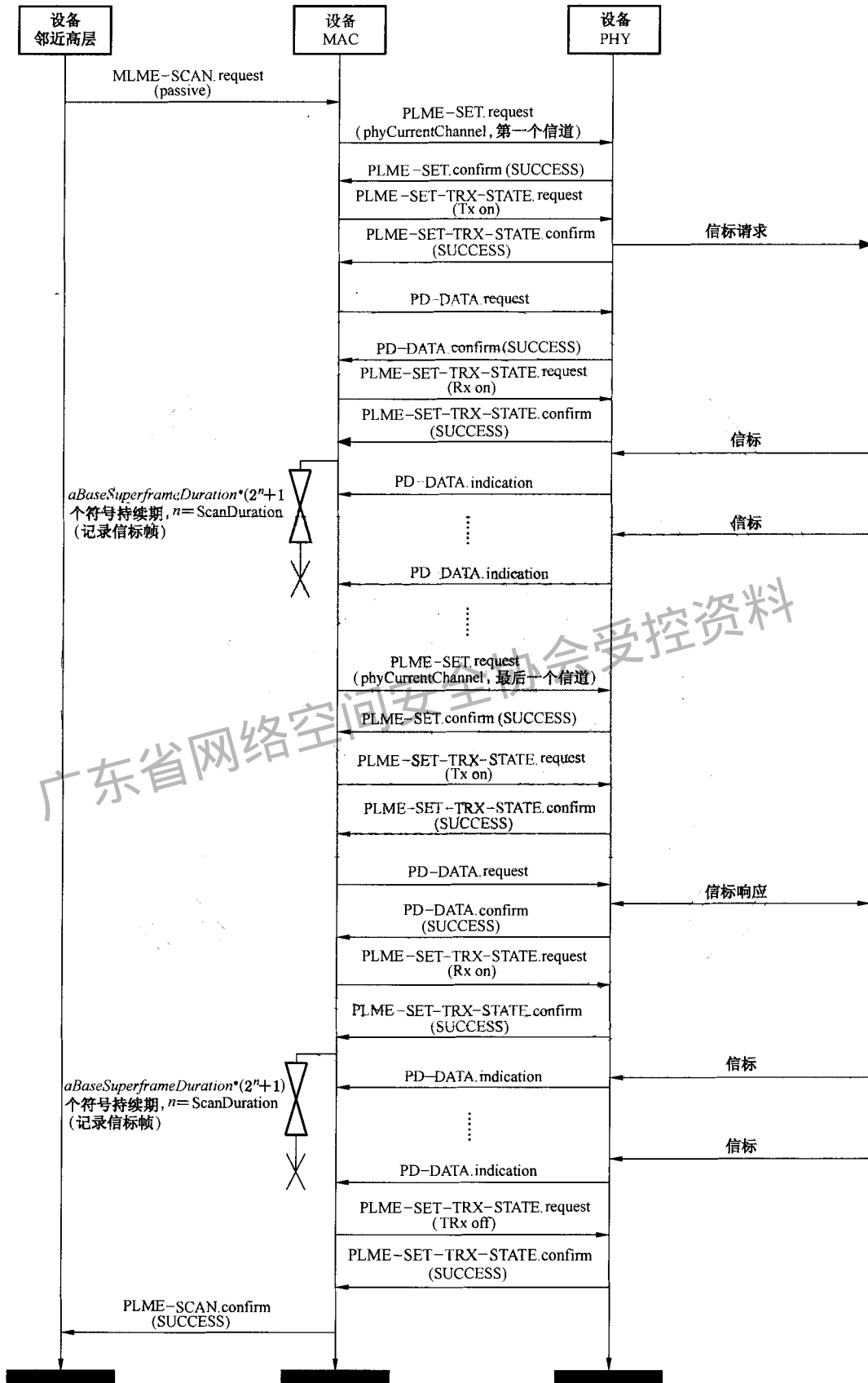


图 87 主动扫描消息序列图

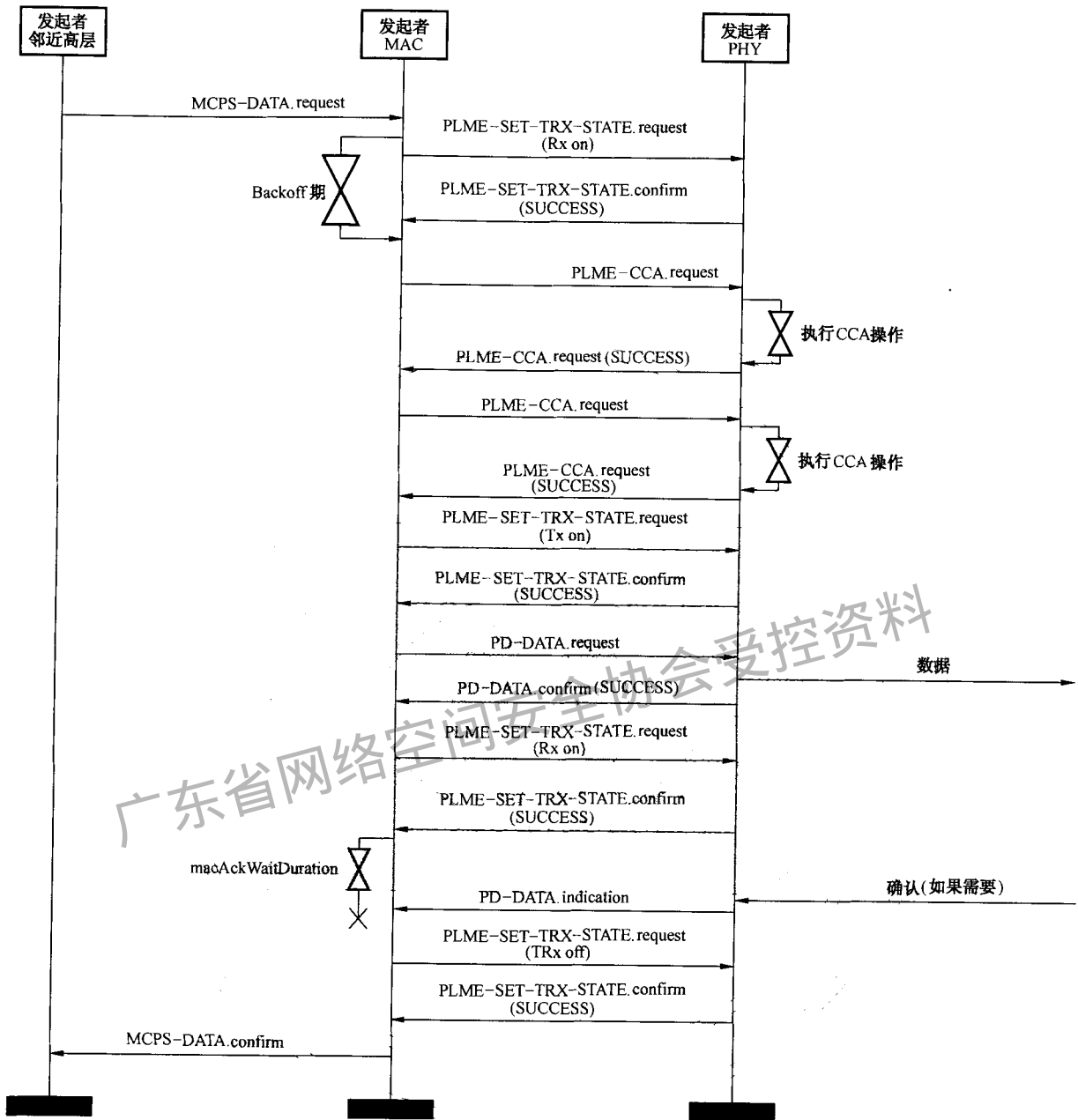


图 88 数据传输消息序列图——发起者

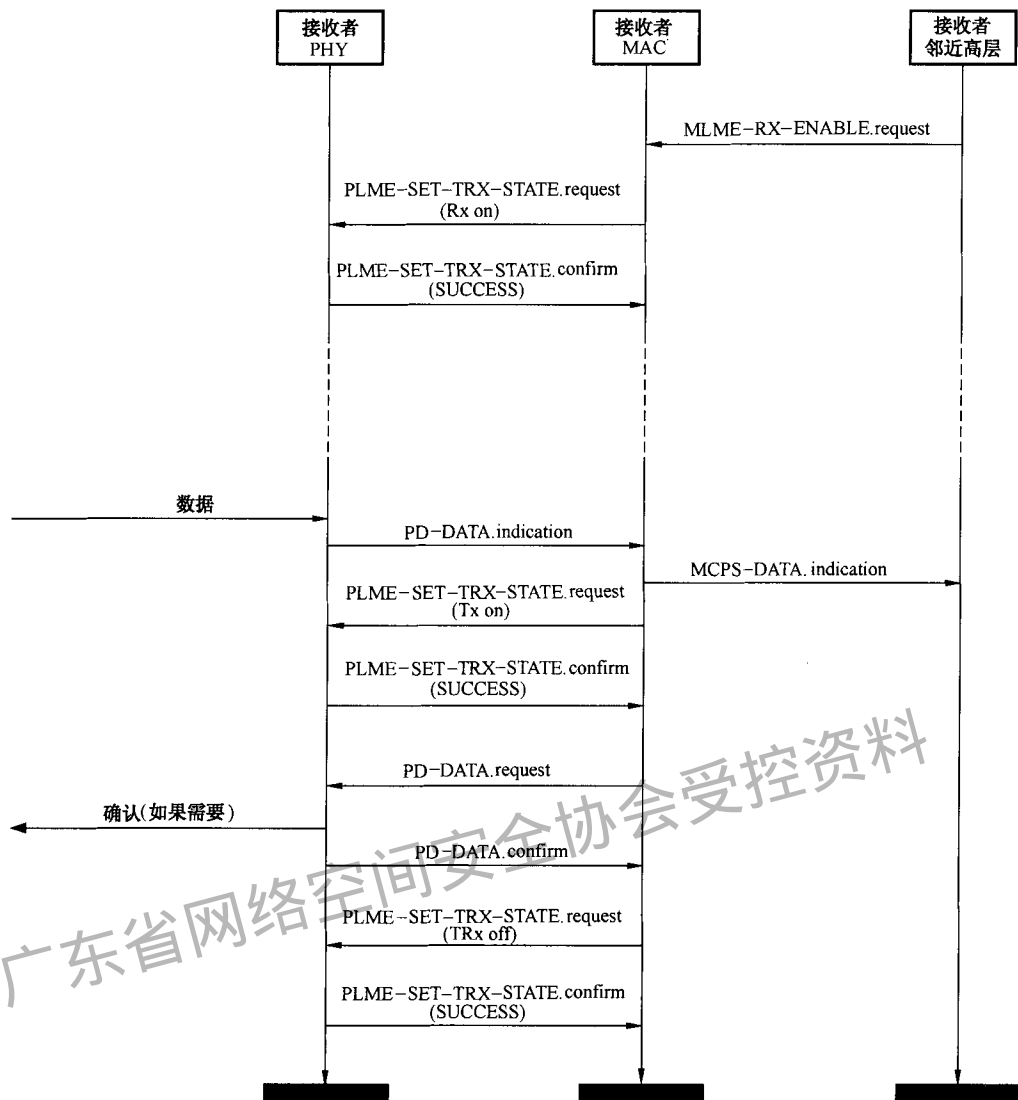


图 89 数据传输消息序列图——接收者

参 考 文 献

- [1] GB/T 15629.15—2010 信息技术 系统间远程通信和信息交换 局域网和城域网 特定要求 第15部分:低速无线个域网(WPAN)媒体访问控制和物理规范
- 

广东省网络空间安全协会受控资料

广东省网络空间安全协会受控资料

中华人民共和国  
国家标准  
信息技术 传感器网络  
第 302 部分:通信与信息交换:  
高可靠性无线传感器网络  
媒体访问控制和物理层规范  
GB/T 30269.302—2015

\*

中国标准出版社出版发行  
北京市朝阳区和平里西街甲 2 号(100029)  
北京市西城区三里河北街 16 号(100045)

网址 [www.spc.net.cn](http://www.spc.net.cn)

总编室:(010)68533533 发行中心:(010)51780238  
读者服务部:(010)68523946

中国标准出版社秦皇岛印刷厂印刷  
各地新华书店经销

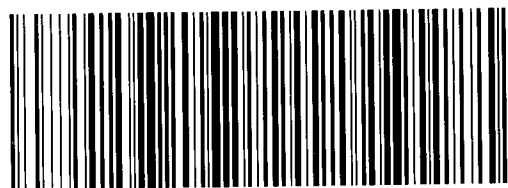
\*

开本 880×1230 1/16 印张 9.5 字数 290 千字  
2016 年 5 月第一版 2016 年 5 月第一次印刷

\*

书号: 155066 · 1-52905 定价 104.00 元

如有印装差错 由本社发行中心调换  
版权专有 侵权必究  
举报电话:(010)68510107



GB/T 30269.302-2015