

# 中华人民共和国国家标准

GB/T 26790.2—2015

## 工业无线网络 WIA 规范 第 2 部分： 用于工厂自动化的 WIA 系统 结构与通信规范

Industrial wireless networks WIA specification—Part 2: WIA system  
architecture and communication specification for factory automation(WIA-FA)

2015-12-10 发布

2016-07-01 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



中华人民共和国  
国家标准  
**工业无线网络 WIA 规范 第 2 部分：**  
**用于工厂自动化的 WIA 系统**

**结构与通信规范**

GB/T 26790.2—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 10 字数 306 千字  
2016 年 5 月第一版 2016 年 5 月第一次印刷

\*

书号: 155066 · 1-53418 定价 106.00 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

## 目 次

前言 .....	III
1 范围 .....	1
2 规范性引用文件 .....	1
3 术语、定义、缩略语和约定 .....	1
3.1 术语和定义 .....	1
3.2 缩略语 .....	4
3.3 约定 .....	5
4 数据编码规则 .....	7
4.1 概述 .....	7
4.2 基本数据类型编码 .....	7
4.3 结构化数据类型编码 .....	11
5 WIA-FA 概述 .....	11
5.1 设备类型 .....	11
5.2 拓扑结构 .....	13
5.3 协议栈结构 .....	13
6 系统管理 .....	14
6.1 概述 .....	14
6.2 设备管理应用进程 .....	15
6.3 设备编址方法和地址分配 .....	27
6.4 通信资源分配 .....	28
6.5 现场设备入网和离网过程 .....	29
6.6 网络性能监视 .....	31
6.7 管理信息库及其服务 .....	32
7 物理层 .....	44
7.1 概述 .....	44
7.2 WIA-FA 物理层一般性要求 .....	45
7.3 WIA-FA 物理层附加要求 .....	46
8 数据链路层 .....	48
8.1 概述 .....	48
8.2 数据链路层数据服务 .....	56
8.3 数据链路层管理服务 .....	59
8.4 数据链路层帧格式 .....	78
8.5 数据链路层状态机 .....	85
9 接入设备与网关设备有线服务 .....	93
9.1 概述 .....	93
9.2 接入设备加入网络 .....	94
9.3 网关设备与接入设备有线连接的帧格式 .....	94

10 应用层 .....	96
10.1 概述 .....	96
10.2 应用层协议栈 .....	96
10.3 应用层功能 .....	97
10.4 应用数据 .....	98
10.5 用户应用进程 .....	99
10.6 应用层服务 .....	107
10.7 应用子层 .....	113
11 安全 .....	135
11.1 概述 .....	135
11.2 安全相关服务 .....	138
11.3 安全加入 .....	144
11.4 密钥管理 .....	145
11.5 数据链路层安全通信 .....	148
11.6 安全告警 .....	149
11.7 安全相关帧格式 .....	149
附录 A (规范性附录) WIA-FA 网络的安全策略 .....	153
A.1 WIA-FA 网络的风险分析 .....	153
A.2 WIA-FA 安全原则 .....	153
A.3 WIA-FA 安全目标 .....	153
A.4 安全系统的分级 .....	153
参考文献 .....	154

## 前　　言

GB/T 26790《工业无线网络 WIA 规范》拟分为以下 8 部分：

- 第 1 部分：用于过程自动化的 WIA 系统结构与通信规范；
- 第 2 部分：用于工厂自动化的 WIA 系统结构与通信规范；
- 第 3 部分：WIA-PA 协议一致性测试规范；
- 第 4 部分：WIA-FA 协议一致性测试规范；
- 第 5 部分：WIA-PA 互操作性测试规范；
- 第 6 部分：WIA-FA 互操作性测试规范；
- 第 7 部分：WIA 产品通用条件；
- 第 8 部分：WIA 行业规范。

本部分为 GB/T 26790 的第 2 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。

本部分由中国机械工业联合会提出。

本部分由全国工业过程测量和控制标准化技术委员会(SAC/TC 124)归口。

本部分主要起草单位：中国科学院沈阳自动化研究所、机械工业仪器仪表综合技术经济研究所、北京科技大学、浙江大学、浙江中控研究院有限公司。

本部分主要起草人：于海斌、梁炜、刘丹、王沁、张晓玲、万亚东、刘帅、杨雨沱、曾鹏、丁露、冯冬芹、齐悦、陈小枫、梅恪、罗新强、刘敏、施一明、陈积明、陈建飞、张思超、王恺、孙亮、程鹏。

# 工业无线网络 WIA 规范 第 2 部分： 用于工厂自动化的 WIA 系统 结构与通信规范

## 1 范围

GB/T 26790 的本部分定义了基于 IEEE STD 802.11-2012 射频(RF)的 WIA-FA(Wireless Networks for Industrial Automation - Factory Automation)系统结构与通信规范。

本部分适用于工厂自动化测量、监视与控制的无线网络系统。

## 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 25931—2010 网络测量和控制系统的精确时钟同步协议(IEC 61588:2009, IDT)

IEEE STD 802.11-2012 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 11 部分：无线局域网媒体访问控制和物理层规范 [Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) specifications]

## 3 术语、定义、缩略语和约定

### 3.1 术语和定义

下列术语和定义适用于本文件。

#### 3.1.1

**绝对时隙号 absolute timeslot number**

从网络形成开始时计数，到当前时间的时隙号。

注：该值以递增 1 的方式计数，为当前时隙的序列号，且最大值为( $2^{48} - 1$ )，达到最大值后，该值重置为 0。

#### 3.1.2

**接入设备 access device**

安装在工业现场，负责将现场设备上的传感器数据、告警及网络管理相关信息转发到网关设备，或将网关设备的控制信号、管理信息和配置信息转发给现场设备。

#### 3.1.3

**聚合 aggregation**

将多个用户应用对象的数据合并成一个包，或将多个现场设备的帧合并成一个帧的过程。

#### 3.1.4

**组态 application configuration**

为现场设备上的用户应用进程配置完成工厂自动化应用中的某一具体任务的过程。

3.1.5

**应用子层 application sub-layer**

提供应用层数据和管理服务的协议子层。

3.1.6

**退避 backoff**

当现场设备发送帧失败时,在规定的重传时隙内重新发送的过程。

3.1.7

**信标 beacon**

在 WIA-FA 网络中由接入设备广播的帧。

注: 新的现场设备在加入 WIA-FA 网络前首先要监听信标。

3.1.8

**信道 channel**

从发送端到接收端传递帧的无线射频介质。

3.1.9

**共存 coexistence**

工厂中的所有无线网络利用共享介质完成各自应用通信需求的状态。

注: 见 IEC 62657-2 3.1.12 条。

3.1.10

**通信资源 communication resource**

用于传输帧的信道和时隙。

3.1.11

**解聚 disaggregation**

将聚合后的包拆分为多个用户应用对象数据,或将聚合后的帧拆分为多个帧的过程。

3.1.12

**现场设备 field device**

安装在工业现场,连接传感器和执行器,负责发送现场数据和接收控制命令的 WIA-FA 设备。

3.1.13

**网关设备 gateway device**

连接 WIA-FA 网络与其他网络的设备。

3.1.14

**手持设备 handheld device**

用于配置网络和固件更新的手持便携设备。

3.1.15

**心跳信号 heartbeat message**

正在工作的网关设备向冗余的网关设备周期性地发送的表示网关设备正常工作的信号。

3.1.16

**主控计算机 host computer**

操作人员、维护人员和管理人员执行组态、网络配置与数据显示等功能的接口,以及执行控制功能。

注: 主控计算机负责网络配置、组态和数据显示功能。

3.1.17

**互操作 interoperability**

两个或两个以上的网络彼此交互信息且利用所交互信息的能力。

注: 见 ISO/IEC TR 10000-1 的 3.2.1。

3.1.18

**加入 joining**

WIA-FA 设备通过认证，并且允许加入 WIA-FA 网络的过程。

3.1.19

**链路 link**

由参数集合构成的两个相邻 WIA-FA 设备之间的通路。

注：参数集合包括链路标识符、链路类型、对端设备的短地址、相对时隙号、当前采用的信道编号以及超帧标识符。

3.1.20

**网络地址 network address**

WIA-FA 网络中，惟一标识设备的 8 位或 16 位无符号整数，也称为短地址。

3.1.21

**网络配置 network configuration**

为网络中的设备分配网络运行和通信相关参数的过程。

3.1.22

**网络管理者 network manager**

负责配置网络、分配通信资源、监视和汇报网络性能的逻辑角色。

注：在 WIA-FA 网络中有且仅有一个网络管理者。

3.1.23

**被动离开 passive leaving**

在 WIA-FA 网络中，一个现场设备被网关设备强制要求离开网络的过程。

3.1.24

**物理地址 physical address**

WIA-FA 网络中，用于标识设备的 64 位扩展惟一标识符，也称为长地址。

注：物理地址由制造商分配。

3.1.25

**主网关设备 primary gateway device**

WIA-FA 网络中正在工作的网关设备。

注：一个 WIA-FA 网络中，有且仅有一个主网关设备。

3.1.26

**预配置 provisioning**

WIA-FA 网络预先配置网络标识符、安全等级、加入密钥和共享密钥等静态信息。

3.1.27

**冗余网关设备 redundant gateway device**

主网关设备的热备份。

3.1.28

**相对时隙号 relative timeslot number**

在当前超帧内计数的时隙号。

3.1.29

**安全管理者 security manager**

在 WIA-FA 网络中，负责提供整个网络的安全策略配置、密钥管理和设备认证的逻辑角色。

3.1.30

**超帧 superframe**

一组周期性重复出现的信道和时隙集合。

注：超帧中时隙的数目决定了超帧循环的频率。

## 3.1.31

**时隙 timeslot**

在 WIA-FA 网络中交换数据所采用的基本时间单位。

注：WIA-FA 网络中的时隙长度是可配置的。

## 3.2 缩略语

下列缩略语适用于本文件。

ACK	Acknowledgement	应答
AI	Analog Input	模拟量输入
AD	Access Device	接入设备
AL	Application Layer	应用层
AMCL	ASLState Machine of Client	应用子层客户机状态机
AMPB	ASLState Machine of Publisher	应用子层发布者状态机
AMRK	ASLState Machine of Report Sink	应用子层报告汇状态机
AMRS	ASLState Machine of Report Source	应用子层报告源状态机
AMSB	ASLState Machine of Subscriber	应用子层预订者状态机
AMSV	ASLState Machine of Server	应用子层服务器状态机
AO	Analog Output	模拟量输出
APDU	Application Protocol Data Unit	应用层协议数据单元
ASDU	Application Service Data Unit	应用层服务数据单元
ASL	Application SubLayer	应用子层
ASLDE	Application SubLayer Data Entity	应用子层数据实体
ASLM	ASLState Mahine	应用子层状态机
ASLME	Application SubLayer Management Entity	应用子层管理实体
ASN	Absolute timeSlot Number	绝对时隙号
C/S	Client/Server	客户机/服务器
CCM *	Extension of counter with cipher block chaining message authentication code	增强的密码段链接消息验证码协议计数器
DGO	DisaGgregation Object	解聚对象
DI	Digital Input	数字量输入
DLDE	Data Link layer Data Entity	数据链路层管理实体
DLL	Data Link Layer	数据链路层
DLME	Data Link layer Management Entity	数据链路层管理实体
DLPDU	Data link Layer Protocol Data Unit	数据链路层协议数据单元
DMAP	Device Management Application Process	设备管理应用进程
DO	Digital Output	数字量输出
DoS	Deny of Service	拒绝服务
DSSS	Direct Sequence Spread Spectrum	直接序列扩频
EIRP	Equivalent Isotropic Radiated Power	等效各向同性辐射功率
ENC	ENCryption	加密
EUI-64	Extended Unique Identifier-64 bits	扩展的 64 位惟一标识符
FCS	Frame Check Sequence	帧校验序列
FD	Field Device	现场设备
FDMA	Frequency Division Multiple Access	频分多路访问
FHSS	Frequency-Hopping Spread Spectrum	跳频扩频

GACK	Group ACK	ACK 组
GW	Gateway Device	网关设备
HC	Host Computer	主控计算机
HD	Handheld Device	手持设备
KED	Data Encryption Key	数据密钥
KEDB	Broadcast Data Encryption Key	广播数据加密密钥
KEDU	Unicast Data Encryption Key	单播数据加密密钥
KEK	Key Encryption Key	密钥加密的密钥
KJ	Join Key	加入密钥
HMAC	Keyed-Hash Message Authentication Code	Hash 加密消息认证码
KS	Shared Key	共享密钥
ID	Identifier	标识
LQI	Link Quality Indication	链路质量指示
LSB	Least Significant Bit	最低有效位
MAC	Medium Access Control	介质访问控制子层
MIB	Mangement Information Base	管理信息库
MIC	Message Integrity Code	消息完整性代码
MSB	Most Significant Bit	最高有效位
NACK	Negative Acknowledgement	否定应答
NM	Network Manager	网络管理者
NONCE	Number used once, a value that has (at most) a negligible chance of repeating	临时随机数
NRT	Non-Real-Time	非实时
OFDM	Orthogonal Frequency Division Multiplexing	正交频分复用
OSI	Open System Interconnection	开放系统互联
PAGO	Packet Aggregation Object	包聚合对象
PDU	Protocol Data Unit	协议数据单元
PHY	PHYSical layer	物理层
PLCP	Physical Layer Convergence Protocol	物理层汇聚协议
P/S	Publisher/Subscriber	发布者/预订者
R/S	Report source/report Sink	报告源点/报告汇点
SAP	Service Access Point	服务访问点
SM	Security Manager	安全管理者
SN	Sign Notation	符号标识
TDMA	Time Division Multiple Access	时分多路访问
UAO	User Application Object	用户应用对象
UAP	User Application Process	用户应用进程
VCR	Virtual Communication Relationship	虚拟通信关系
WIA-FA	Wireless Network for Industrial Automation—Factory Automation	用于工厂自动化的工业无线网络

### 3.3 约定

本文件采用图的形式表示一个状态机。状态机定义如图 1 所示。

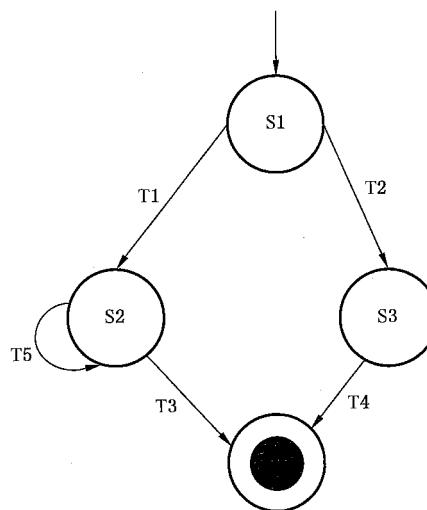


图 1 状态机定义

状态机的定义包括：

- S1、S2 等符号标识的圆圈表示设备当前处于的状态，实心圆圈●表示结束状态，该状态下，设备停止一切动作；
- 有向线表示状态转移，表示从一个状态转移到另一个状态；
- 状态转移用字母“T”表示，包含转移条件以及相应的动作（可能不存在），线上的转移条件和线下的动作通过一个水平线进行分割。

状态转移的定义如表 1 所示。

表 1 状态转移定义

编号	当前状态	事件/条件 =>动作	下一状态
状态转移 迁移编号	状态转移对 应的当前状态	触发状态转移的事件或触发条件 => 事件和触发条件发生/满足后，发生的动作；动作通常在 事件/触发条件下面进行表示	状态转移发生后的下一个状态

状态转移的定义包括：

- := 表示右边条目的值取代左边条目的值；对于右边条目是参数的情况，表示由输入事件生成的参数的值取代左边条目的值；
- == 表示判断左边条目等于右边条目的逻辑条件；
- && 表示逻辑与“AND”；
- || 表示逻辑或“OR”；
- != 表示左边条目不等于右边条目的逻辑条件。

状态转移的定义允许在一次转移过程中循环执行一系列的动作，该循环从 start\_value 开始，执行到 end\_value。

示例：

```

for (Identifier=start_value to end_value)
    动作
end
  
```

状态转移的定义允许在一次转移过程中,根据触发条件选择执行的动作。触发条件可以为某些标识符,或者前一个动作的输出。

示例:

```
If (触发条件)
    动作
else
    动作
endif
```

## 4 数据编码规则

### 4.1 概述

WIA-FA 数据编码规定了由各层服务传递的独立于机器的数据语法。WIA-FA 支持基本数据类型和结构化数据类型的定义和传输。

基本类型是原子(atomic)类型,即不能再被划分为更小的类型。结构化类型由若干基本类型和其他结构化类型组成,其嵌套的复杂度和深度不受本部分约束。

### 4.2 基本数据类型编码

#### 4.2.1 整数(Integer)类型编码

整数类型数据的值是有符号的整数,编码如图 2 所示。表 2 以 Integer16 为例给出了八位位组中每位的编码。数据传输时,首先发送该类型数据最高有效八位位组的 MSB。

符号: Integer8, Integer16, Integer24, Integer32		
数据类型	取值范围	长度
Integer8	$-128 \leq i \leq 127$	1 个八位位组
Integer16	$-32\,768 \leq i \leq 32\,767$	2 个八位位组
Integer24	$-2^{23} \leq i \leq 2^{23} - 1$	3 个八位位组
Integer32	$-2^{31} \leq i \leq 2^{31} - 1$	4 个八位位组

二进制补码表示法  
MSB(最高有效位)是第一个八位位组符号位(SN)之后的比特  
SN=0:正数和 0  
SN=1:负数

图 2 整数类型数据编码

表 2 Integer16 整数类型数据编码

八位位组	位							
	7	6	5	4	3	2	1	0
1	SN	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$
2	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

#### 4.2.2 无符号(Unsigned)类型编码

无符号类型数据的值是无符号整数,编码按照图 3 所示。表 3 以 Unsigned16 为例给出了八位位组

中每位的编码。数据传输时，首先发送该类型数据最高有效八位位组的 MSB。

符号: Unsigned8, Unsigned16, Unsigned24, Unsigned32, Unsigned40, Unsigned48, Unsigned64, Unsigned80		
数据类型	取值范围	长度
Unsigned8	$0 \leq i \leq 255$	1 个八位位组
Unsigned16	$0 \leq i \leq 65535$	2 个八位位组
Unsigned24	$0 \leq i \leq 2^{24}-1$	3 个八位位组
Unsigned32	$0 \leq i \leq 2^{32}-1$	4 个八位位组
Unsigned40	$0 \leq i \leq 2^{40}-1$	5 个八位位组
Unsigned48	$0 \leq i \leq 2^{48}-1$	6 个八位位组
Unsigned64	$0 \leq i \leq 2^{64}-1$	8 个八位位组
Unsigned80	$0 \leq i \leq 2^{80}-1$	10 个八位位组

图 3 无符号类型数据编码

表 3 Unsigned16 无符号类型数据编码

八位位组	位							
	8	7	6	5	4	3	2	1
1	$2^{15}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$	$2^8$
2	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

#### 4.2.3 浮点(Float)类型编码

浮点值按照图 4 和图 5 所示进行编码。首先发送符号和指数的 MSB，再发送指数的剩余部分以及

图 4 单精度浮点类型数据编码

分数的 MSB 到 LSB。如果浮点数据类型的对象值未知，则发送值 0x7F, 0xA0，后续全部为零(0x00)；该值表示“非数(Not-a-number)”。

图 5 双精度浮点类型数据编码

#### 4.2.4 八位字符串(Octetstring)类型编码

表 4 所示为八位字符串(Octetstring)的编码方式。对于 N 个八位位组的字符串,首先发送该类型数据最高有效八位位组的 MSB。

表 4 Octetstring 类型数据编码

八位位组	位							
	7	6	5	4	3	2	1	0
1	$2^{8N-1}$	$2^{8N-2}$	$2^{8N-3}$	$2^{8N-4}$	$2^{8N-5}$	$2^{8N-6}$	$2^{8N-7}$	$2^{8N-8}$
2	$2^{8N-9}$	$2^{8N-10}$	$2^{8N-11}$	$2^{8N-12}$	$2^{8N-13}$	$2^{8N-14}$	$2^{8N-15}$	$2^{8N-16}$
...	...	...	...	...	...	...	...	...
N	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

#### 4.2.5 比特域(Bit Field)类型编码

比特域数据类型用于编码固定长度的单比特数据。BitFieldn 表示 n 个比特的比特域。表 5、表 6 和表 7 给出了比特域数据类型的比特编号方式。比特域数据的长度(以比特为单位)值应为 8 的倍数，

并以一个八位位组序列在网络上传输,首先发送数据的最高有效八位位组的 MSB。

表 5 1 个八位位组的 BitField8 类型数据编码

八位位组	位							
	7	6	5	4	3	2	1	0
1	7	6	5	4	3	2	1	0

表 6 2 个八位位组的 BitField16 类型数据编码

八位位组	位							
	7	6	5	4	3	2	1	0
1	15	14	13	12	11	10	9	8
2	7	6	5	4	3	2	1	0

表 7 3 个八位位组的 BitField24 类型数据编码

八位位组	位							
	7	6	5	4	3	2	1	0
1	23	22	21	20	19	18	17	16
2	15	14	13	12	11	10	9	8
3	7	6	5	4	3	2	1	0

#### 4.2.6 比特串(Bitstring)类型编码

比特串数据类型用来编码可变长度的单比特数据。表 8 给出了比特串数据类型的比特编码方式。该类型的数据被定义为一个比特序列,且应同时规定其比特长度。比特串的数据被打包成若干八位位组并以一个八位位组序列在网络上传输。八位位组的个数等于能包含所有比特值的最小八位位组个数。对于多于 1 个八位位组的数据,首先发送数据的最高有效八位位组的 MSB。

表 8 Bitstring 类型数据编码

八位位组	位							
	7	6	5	4	3	2	1	0
1	X <sup>a</sup>	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X
... <sup>b</sup>	... <sup>c</sup>							

<sup>a</sup> 比特串的 MSB。

<sup>b</sup> 八位位组的个数等于能包含所有比特值的最小八位位组个数。

<sup>c</sup> 最后比特 (LSB) 的位置=8-(比特长度除以 8 的余数);并且,应使用从 MSB 到该位置的比特。

#### 4.2.7 时间(TimeData)类型编码

时间数据类型是 64 比特的无符号整数,表示以  $1 \mu\text{s}$  递增的时间。

#### 4.2.8 密钥(KeyData)类型编码

密钥数据类型是 128 比特的无符号整数。

### 4.3 结构化数据类型编码

#### 4.3.1 结构体(Struct)类型编码

结构体类型是由不同基本数据类型或结构化数据类型构成的一个有序集合。这些基本数据类型或结构化数据类型的数据被称为结构体的成员,由成员标识符(MemberID)标识。结构体类型的数据可被完整地访问,或者通过规定成员标识符 MemberID 来单独访问结构体数据的某个成员。

#### 4.3.2 列表(List)类型编码

列表类型是由相同数据类型构成的一个有序集合。每一条数据被称为列表的一个记录,由存储索引(FirstStoreIndex)标识。列表类型数据可被完整地访问,或者通过规定起始索引和记录个数来访问列表的某个或某些记录。

## 5 WIA-FA 概述

### 5.1 设备类型

本部分定义了以下五类设备:

- 主控计算机(HC, Host Computer);
- 网关设备(GW, Gateway Device);
- 接入设备(AD, Access Device);
- 现场设备(FD, Field Device);
- 手持设备(HD, Handheld Device)。

为了提高网络的可靠性,WIA-FA 网络中允许存在备用的网关设备作为运行网关设备的热备份,允许存在多个接入设备并行工作。

**注 1:** WIA-FA 网络中,冗余网关设备(见 3.1.27)与正在工作的网关设备(主网关设备,见 3.1.25)采用相同的网络地址。二者采用有线连接方式且同步更新。冗余的网关设备周期性地接收正在工作的网关设备以有线连接方式发来的网关心跳信号。冗余的网关设备在“PriGwFailureTime”(见表 15)时间内没有收到网关心跳信号,则接替原有正在工作的网关设备的全部工作。

**注 2:** 冗余网关上的 NM 和 SM 作为主网关上 NM 和 SM 的备份,具体实现由制造商完成。

**注 3:** 接入设备与网关设备以有线方式连接(见第 9 章)。多个接入设备以冗余方式并行工作。

#### 5.1.1 主控计算机

主控计算机是操作人员、维护人员和管理人员执行组态、网络配置与数据显示等功能的接口,以及执行控制功能。主控计算机负责网络配置、组态和数据显示功能。主控计算机的具体实现不在本部分定义范围内。

#### 5.1.2 网关设备

网关设备是连接 WIA-FA 网络与其他网络的设备。网关设备包括以下主要功能:

- 提供 WIA-FA 网络与现场总线等外部网络连接的接口,利用数据映射和协议转换功能实现 WIA-FA 网络与现场总线等外部网络的互连;
- 负责网络管理和安全管理功能;
- 通过接入设备与 WIA-FA 网络中的其他设备进行通信,交换设备间的信息;
- 作为全网惟一的时钟源,实现网络时间同步。

### 5.1.3 接入设备

接入设备安装在工业现场,负责将现场设备上的传感器数据、告警及网络管理相关信息转发到网关设备,或将网关设备的控制信号、管理信息和配置信息转发给现场设备。接入设备包括以下主要功能:

- 接收现场设备采集到的数据并发送给网关设备;
- 将网关设备的控制命令发送给现场设备中的执行器;
- 将网关设备的管理、配置和组态信息发送给现场设备;
- 接收现场设备的告警及网络管理相关信息并发送给网关设备。

注:接入设备与网关设备之间以有线方式连接,两者之间的同步方式不在本部分范围内。

### 5.1.4 现场设备

现场设备是安装在工业现场,连接传感器和执行器,负责发送现场数据和接收控制命令的 WIA-FA 设备。现场设备的供电方式包括线路供电、电池供电及其他,具体供电方式不做定义。

### 5.1.5 手持设备

手持设备是用于配置网络和固件更新的手持便携设备。手持设备与直连的设备通信,不参与 WIA-FA 网络中其他设备的通信。手持设备采用有线方式预配置现场设备、接入设备和网关设备,为现场设备、接入设备和网关设备写入安全等级、加入密钥和共享密钥(当安全等级不为 0 时)、网络标识符(NetworkID)。

注:手持设备采用的有线方式可包括 RS232、RS485、USB 等。

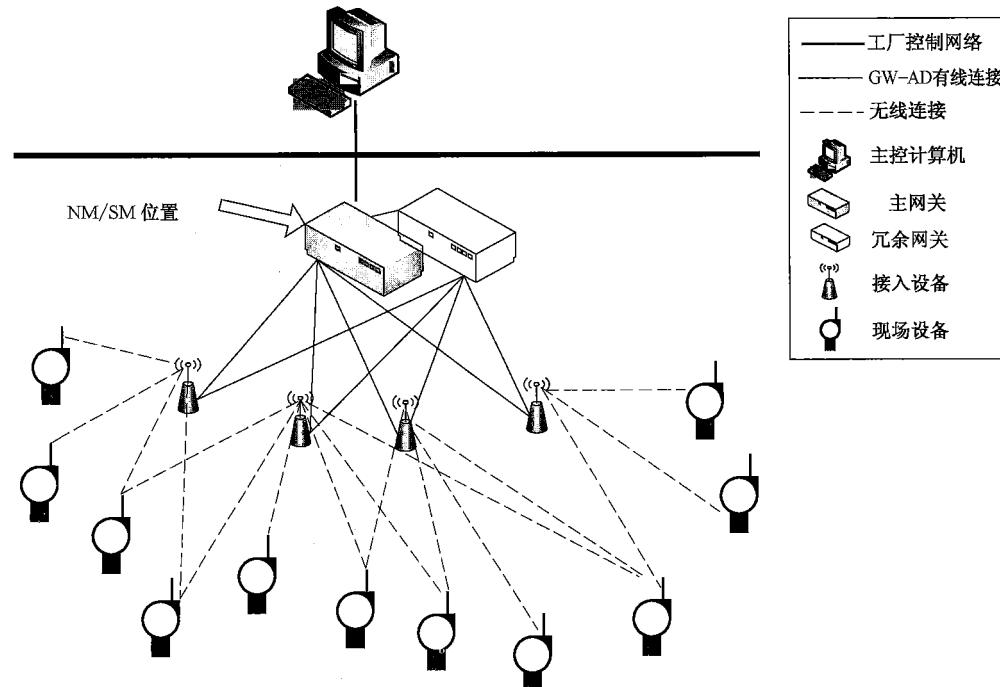


图 6 WIA-FA 网络增强星型拓扑结构

## 5.2 拓扑结构

如图 6 所示,WIA-FA 定义为增强星型拓扑结构(Enhanced star topology),包括一个中心及若干现场设备。中心由一个网关设备(可存在冗余网关设备)及一或多个接入设备组成。

## 5.3 协议栈结构

WIA-FA 网络协议遵循 ISO/IEC 7498 OSI 的基本参考模型,定义了物理层(PHY)、数据链路层(DLL)及应用层(AL)。图 7 所示为 WIA-FA 的协议栈与 OSI 基本参考模型的映射关系。

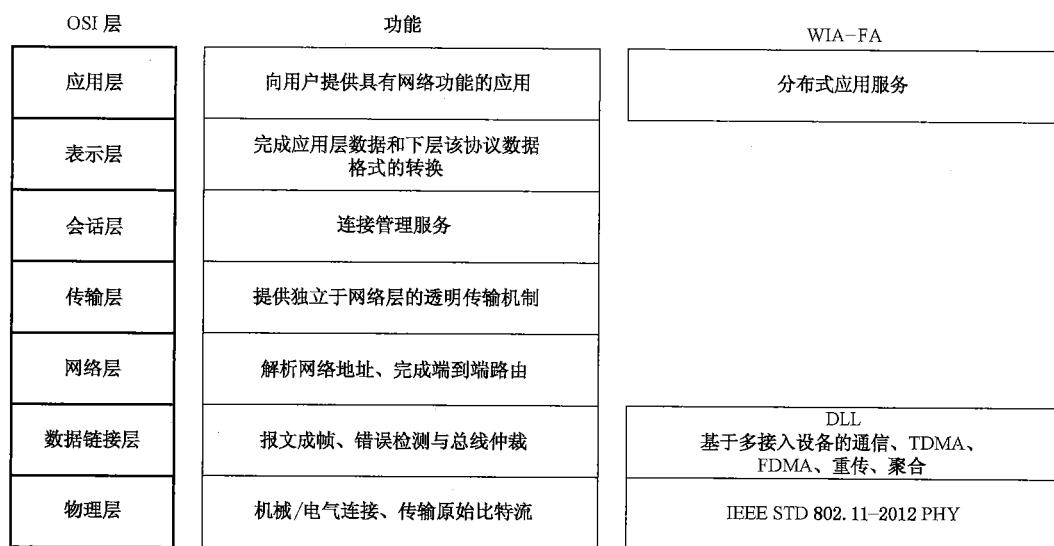


图 7 OSI 基本参考模型与 WIA-FA 网络协议层映射关系

图 8 所示为 WIA-FA 的协议栈结构。WIA-FA 协议栈结构包括:

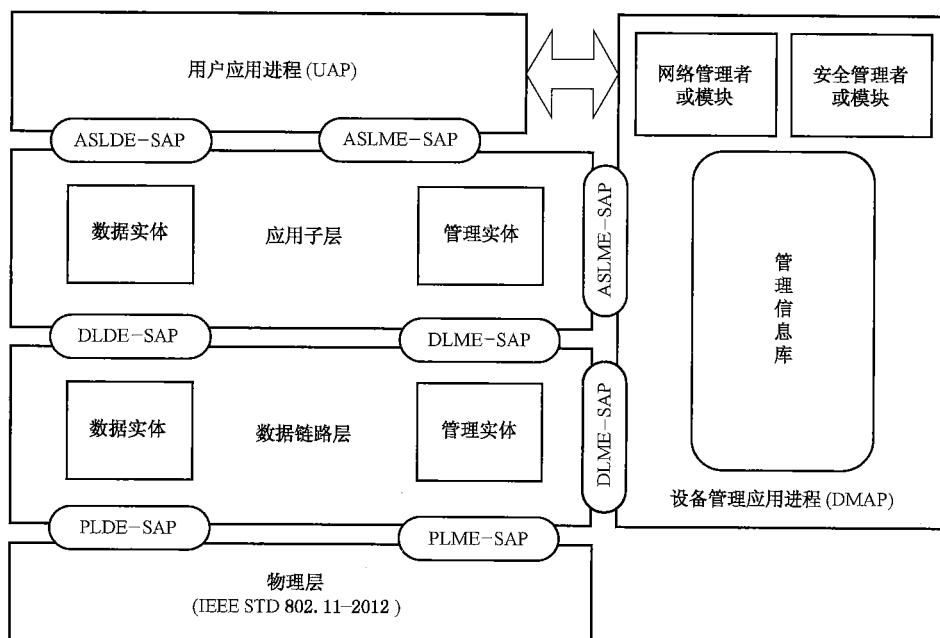


图 8 WIA-FA 协议栈结构

- 协议层：包括物理层、数据链路层、应用层（具体包括应用子层，用户应用进程 UAP 以及设备管理应用进程 DMAP）；
- 协议层内部实体：包括各层的数据实体 DLDE/ASLDE 和管理实体 DLME/ASLME；
- 协议层接口：包括层与层之间通信的接口 SAP，具体指数据实体 SAP(DLDE-SAP/ASLDE-SAP) 以及管理实体 SAP(DLME-SAP/ASLME-SAP)。

其中，DMAP 中包括网络管理者/模块、安全管理者/模块以及管理信息库 MIB。DMAP 是一类特殊的用户应用进程 UAP。DMAP 与 UAP 共用 ASLDE-SAP 和 ASLME-SAP 接口实现与 ASL 之间的交互。

WIA-FA 网络设备的数据流如图 9 所示，包括：

- 现场设备协议栈包括应用层、数据链路层和物理层；
- 接入设备协议栈包括数据链路层和物理层，接入设备与网关设备采用有线接入方式；
- 网关设备协议栈包括应用层（UAP 和 DMAP），也包括部分 DLL 层功能。网关设备上运行的应用包括：与 WIA-FA 网络应用层之间的交互；与现场网络应用层的交互；翻译 WIA-FA 网络与现场网络之间的信息。

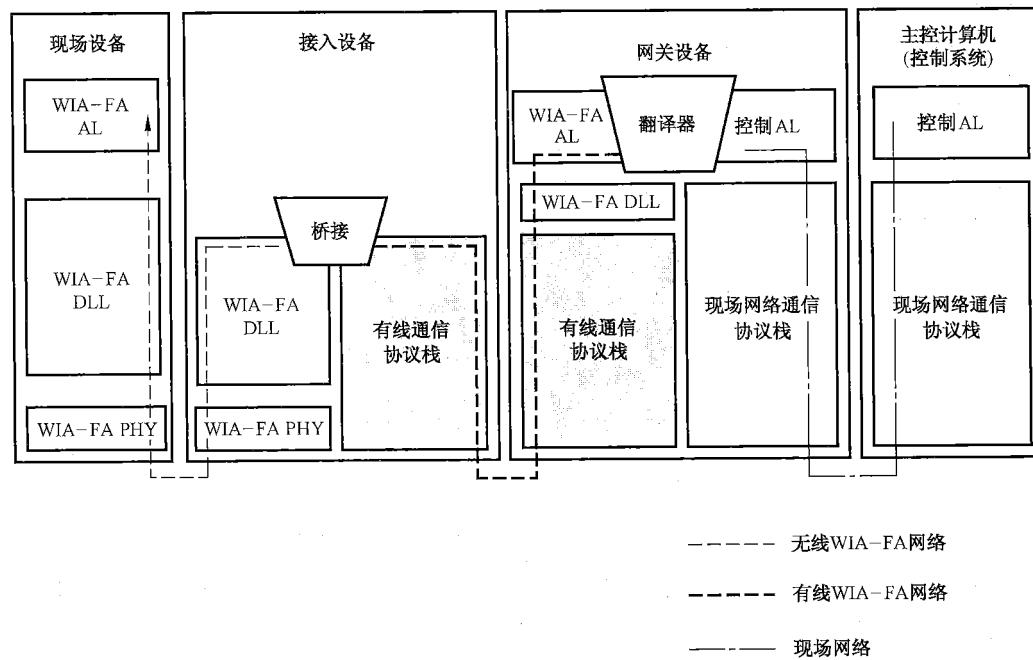


图 9 WIA-FA 数据流

## 6 系统管理

### 6.1 概述

WIA-FA 网络采用集中式的管理架构，如图 10 所示。系统管理功能由网关设备中的网络管理者 (NM)、安全管理者 (SM) 以及接入设备和现场设备的网络管理模块、安全管理模块共同完成。网络管理者和安全管理者在网关设备中实现，负责管理接入设备和现场设备。网络管理模块和安全管理模块在接入设备和现场设备中实现，配合网络管理者和安全管理者实现管理功能。

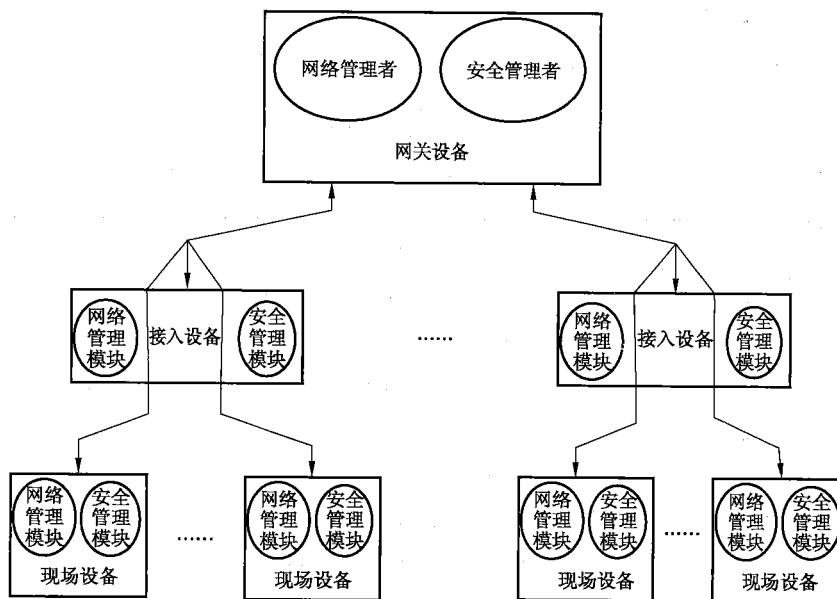


图 10 系统管理架构

## 6.2 设备管理应用进程

WIA-FA 网络的系统管理功能应由设备中的设备管理应用进程(DMAP)实现。DMAP 是特殊的用户应用进程,负责管理设备以及提供 MIB 访问服务。DMAP 在协议栈中的位置和构成如图 11 所示,灰色部分为 DMAP,其中的白色块为 DMAP 中的功能模块,具体包括:

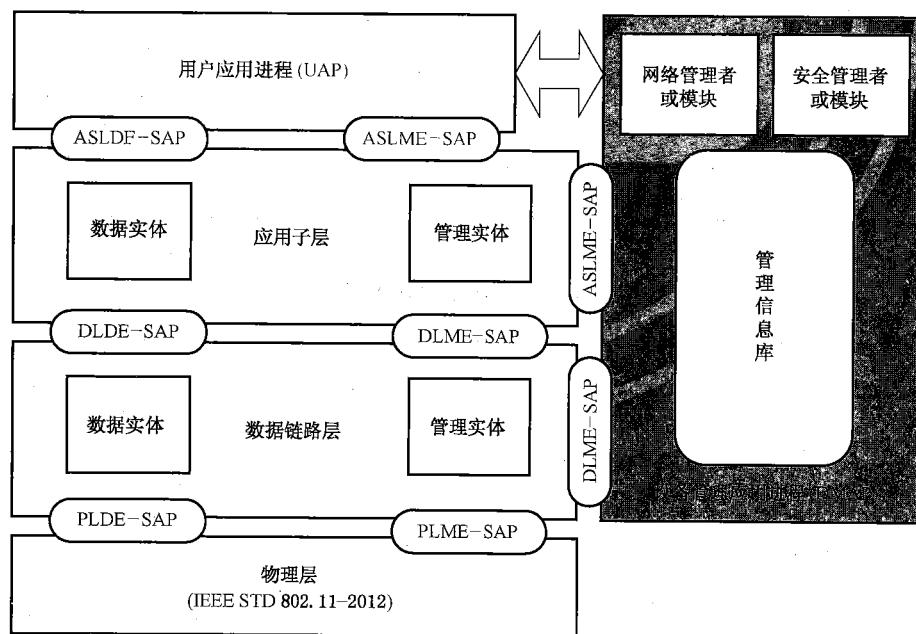


图 11 系统管理中的 DMAP

- 网关设备中的网络管理者,接入设备和现场设备中的网络管理模块;
- 网关设备中的安全管理者,接入设备和现场设备中的安全管理模块;
- 管理信息库,存储 WIA-FA 网络中用于网络管理和安全管理的全部属性。

DMAP 是一种特殊的 UAP,与 UAP 共用 ASLDE-SAP 接口,与 ASL 交互信息。WIA-FA 网络实现的网络管理功能如表 9 所示。安全管理功能如表 10 所示。

表 9 网络管理功能

网络管理功能	要求
网络建立	初始化: 初始化网络管理者与网络管理模块, 并启动网络
	时间源配置和系统时间服务: 一个 WIA-FA 网络应设置一个基准时间源, 这个基准时间源由网关设备充当。网络内的设备仅与网关设备进行时间同步
	设备加入过程管理: 设备入网前需要网络 ID(见表 15)。待加入设备发起加入过程, 由网络管理者 NM 返回加入响应
	网络地址分配: 网络中的所有设备均有一个称为“长地址”的 64 位全球惟一地址和一个称为“短地址”的 8 位或者 16 位的网络地址。所有设备的长地址在设备出厂时由厂商按 64 位扩展惟一标识符(EUI-64)生成并设置。网络中所有设备的网络地址由网络管理者 NM 分配
	拓扑管理: 形成和维护如图 6 所示的增强星型拓扑结构
网络调度和通信资源配置	网络配置管理: 维护通信资源、网络地址、网络属性等配置信息, 包括网络管理者分发给网络中所有设备的信息; 完成对 WIA-FA 信息库的配置
	超帧创建: 根据应用需求创建用于通信的超帧
	通信资源分配: 将超帧中通信资源分配给链路
网络诊断和性能监视	激活/去活: 根据应用过程激活和去活超帧
	信道管理: 信道汇报功能, 以及监视和维护信道的列表及状态
	设备健康状况: 监视和维护网络中每个设备的健康信息
离开	网络性能监视: 负责监视和维护网络性能
	管理设备离开过程: 现场设备的离开过程分为被动离开和异常离开。被动离开过程由网关设备发起, 现场设备收到离开请求后离开网络, 网关设备释放现场设备的通信资源。网关设备检测和处理异常离开

表 10 安全管理功能

安全管理功能	要求
安全的网络建立和配置	设备安全加入过程管理: 待加入设备发起安全加入过程, 通过安全管理者 SM 认证后, 由网络管理者 NM 返回安全加入响应
	密钥建立: 设备安全加入网络后, 安全管理者 SM 对其分发在正常运行过程中进行安全操作所使用的密钥, 包括 KEK、KEDB 以及 KEDU
密钥更新	密钥更新: 安全管理者对网络中正在使用的密钥在其生命周期结束前进行更新, 包括 KEK、KEDB 以及 KEDU
安全的性能监视	安全告警: 监视密钥的更新情况和密钥受攻击的次数

### 6.2.1 网络管理者

NM 在网关设备中实现网络管理功能, 管理网络中所有设备的信息。每个网络中有且只有一个运

行状态的网络管理者。

网络管理者主要完成如下功能：

- 为网络中的所有设备分配惟一的 8 位或 16 位短地址(详见 6.3)；
- 构建和维护增强星型拓扑；
- 分配 WIA-FA 设备通信所需的资源；
- 监视 WIA-FA 网络的性能,包括设备状态以及信道状况等。

#### 6.2.2 安全管理者

安全管理者(SM)在网关设备中实现安全功能,每个网络中有且只有一个运行状态的安全管理者。SM 直接与 NM 进行通信。

安全管理者主要完成如下功能：

- 认证试图加入 WIA-FA 网络的现场设备；
- 负责整个网络的密钥管理,包括密钥建立和密钥更新(详见 11.4)；
- 处理安全告警。

#### 6.2.3 网络管理模块

网络管理模块在现场设备和接入设备中实现网络管理功能,存储现场设备和接入设备通信所需的信息。

网络管理模块主要完成如下功能：

- 配合 NM 构建和维护增强星型拓扑；
- 配合 NM 分配 WIA-FA 设备通信所需的资源；
- 配合 NM 监视 WIA-FA 网络的性能,包括设备状态以及信道状况等。

#### 6.2.4 安全管理模块

安全管理模块在现场设备和接入设备中实现安全功能。

安全管理模块主要完成如下功能：

- 配合 SM 完成现场设备的安全加入；配合 SM 管理密钥管理；
- 配合 SM 汇报安全告警。

#### 6.2.5 DMAP 状态机

##### 6.2.5.1 网关设备 DMAP 状态机

网关设备状态机如图 12 所示,包括 Init 和 Active 两个状态。网关设备 DMAP 完成初始化后,由 Init 状态进入 Active 状态。

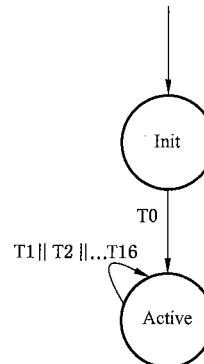


图 12 网关设备 DMAP 状态机

网关设备 DMAP 的状态转移如表 11 所示。

表 11 网关设备的 DMAP 状态转移表

编号	当前状态	事件/触发条件 => 动作	下一状态
T0	Init	IsDMAPIInitializationDone() == TRUE => ;	Active
T1    T2    ...T16	Active	T1    T2    ...T16 => 见表 12	Active

网关设备 DMAP 为每个现场设备维护一个状态机, 如图 13 所示。网关设备可以并行处理来自多个现场设备的多个报文。网关设备 DMAP 从 Init 状态转移到 Active 状态的触发条件是 T1~T16 的任意一条(如表 12 所示)。

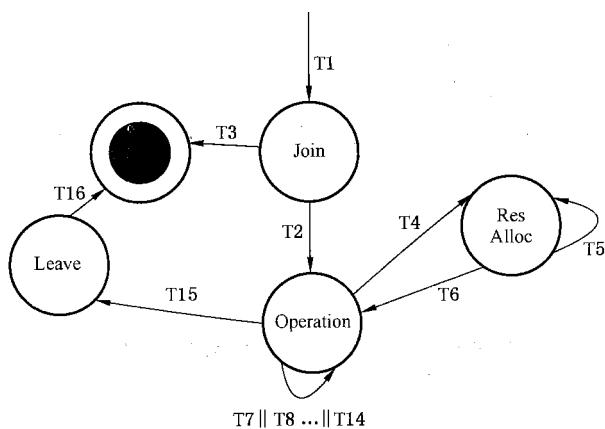


图 13 网关设备 DMAP 为每个现场设备维护的状态机

网关设备的 DMAP 所执行功能的状态转移如表 12 所示。

表 12 网关设备的 DMAP 状态转移表

编号	当前状态	事件/触发条件 => 动作	下一状态
T1	Active	DLME-JOIN.indication(PhyAddr, SecMaterial) =>	Join
T2	Join	Authentication(PhyAddr, SecMaterial) == SUCCESS && AllocateShortAddr(Addr) == SUCCESS => DLME-JOIN.response (Status := SUCCESS, ShortAddr);	Operation

表 12 (续)

编号	当前状态	事件/触发条件 => 动作	下一状态
T3	Join	<pre> Authentication(PhyAddr, SecMaterial) != SUCCESS    AllocateShortAddr(Addr) != SUCCESS =&gt; If ((Authentication(PhyAddr, SecMaterial) != SUCCESS) {     DLME-JOIN.response (Status := AUTH_FAILURE, ShortAddr); } else if(AllocateShortAddr(Addr) != SUCCESS) {     DLME-JOIN.response (Status := NETWORK_SCALE_ERROR,     ShortAddr); } </pre>	End
T4	Operation	<pre> IsHostComputerConfigureDone() == TRUE =&gt; AllocResult := ResAllocAgrithm (SuperframeList.LinkList); If (AllocResult == SUCCESS) {     DLME-INFO-SET.request (Handle, DstAddr, AttributeOption :     = 2, AttributeID := 131, MemberID := 12, FirstStoreIndex,     Count, AttributeValue := ALLOCATION);     DLME-INFO-SET.request (Handle, DstAddr, AttributeOption :     = 0, (AttributeID := 128)    (AttributeID := 129), MemberID,     FirstStoreIndex,     Count, AttributeValue); } </pre>	Res Alloc
T5	Res Alloc	<pre> DLME-INFO-SET.confirm() &amp;&amp; (AttributeID == 128    AttributeID == 129) =&gt; DLME-INFO-SET.request (Handle, DstAddr, AttributeOption : = 0, (AttributeID := 128)    (AttributeID := 129), MemberID, FirstStoreIndex, Count, AttributeValue); </pre>	Res Alloc
T6	Res Alloc	<pre> IsAllResAllocateDone() == TRUE =&gt; DLME-INFO-SET.request (Handle, DstAddr, AttributeOption : = 2, AttributeID := 131, MemberID := 12, MemberID, First- StoreIndex, Count, AttributeValue := OPERATION); </pre>	Operation

表 12 (续)

编号	当前状态	事件/触发条件 => 动作	下一状态
T7	Operation	DMAP-MIB-SET.request() => Status := WriteToMIB(Handle, ShortAddr, AttributeID, MemberID, FirstStoreIndex, Count, AttributeValue); DMAP-MIB-SET.confirm(Handle, Status);	Operation
T8	Operation	DMAP-MIB-GET.request() => Status := ReadFromMIB(Handle, ShortAddr, AttributeID, MemberID, FirstStroeIndex, Count); DMAP-MIB-GET.confirm (Handle, Status, Count, AttributeValue);	Operation
T9	Operation	IsHostComputerSet MIB() == TRUE => DLME-INFO-SET.request (Handle, DstAddr, AttributeOption, AttributeID, MemberID, FirstStoreIndex, Count, AttributeValue);	Operation
T10	Operation	DLME-INFO-SET.confirm() => IndicateSetMIBResult(Handle, Status);	Operation
T11	Operation	IsHostComputerGet MIB() == TRUE => DLME-INFO-GET.request (Handle, DstAddr, AttributeID, MemberID, FirstStoreIndex, Count);	Operation
T12	Operation	DLME-INFO-GET.confirm() => IndicateGetMIBResult (Handle, SrcAddr, Status, AttributeID, MemberID, FirstStoreIndex, Count, AttributeValue);	Operation
T13	Operation	DLME-CHANNEL-CONDITION.indication() => HandleChannelStatusReport(Addr, Count, ChannelConditonInfo);	Operation
T14	Operation	DLME-DEVICE-STATUS.indication() => HandleDeviceStatusReport(ShortAddr, PowerSupplyStatus);	Operation
T15	Operation	IsHostComputerRequestDeviceLeave() == TRUE => DLME-LEAVE.request (DeviceAddr); IndicateHostComputerLeaveResult(Addr);	Leave
T16	Leave	DLME-LEAVE.response() => ReleaseResources(Addr)	End

网关设备的 DMAP 具有以下状态。

#### ——Join 状态

处于 Join 状态的网关设备 DMAP 处理来自现场设备的加入请求,对现场设备执行入网认证和分配短地址。如果认证成功,则 NM 中的 DMAP 为现场设备分配短地址;如果认证或地址分配失败,则调用数据链路层的 DLME-JOIN.request 原语,通知现场设备入网失败结果。认证结束后,DMAP 状态机转到 End 状态。如果认证和地址分配都成功,则网关设备 DMAP 调用 DLME-Join.response 原语通知现场设备加入成功,DMAP 状态机转到 Operation 状态。

#### ——Operation 状态

在 Operation 状态,有以下事件可能发生:

- a) 主控计算机远程设置现场设备 MIB 属性,DMAP 调用 DLME-INFO-SET.request 原语,请求 DLL 生成远程配置属性请求命令帧(见 8.4.17);
- b) DLL 调用 DLME-INFO-SET.confirm 原语,返回远程配置现场设备的 MIB 属性的结果;
- c) 主控计算机远程获取现场设备 MIB 属性,DMAP 调用 DLME-INFO-GET.request 原语,请求 DLL 生成远程读属性请求命令帧(见 8.4.15);
- d) DLL 调用 DLME-INFO-GET.confirm 原语,返回远程读现场设备的 MIB 属性的结果;
- e) 主控计算机调用 DMAP-MIB-SET.request 原语本地设置网关设备 MIB 属性;DMAP 设置属性值后,调用 DMAP-MIB-SET.confirm 向主控计算机返回本地设置网关设备 MIB 属性的结果;
- f) 主控计算机调用 DMAP-MIB-GET.request 原语本地获取网关设备 MIB 属性,DMAP 获取属性值后,调用 DMAP-MIB-GET.confirm 原语返回本地获取网关设备 MIB 属性的结果;
- g) DLL 收到现场设备的设备状态报告后,调用 DLME-DEVICE-STATUS.indication 原语通知 DMAP;
- h) DLL 收到现场设备的信道状况报告后,调用 DLME-CHANNEL-CONDITION.indication 通知 DMAP;
- i) 主控计算机请求现场设备离开网络,DMAP 调用 DLME-LEAVE.request 原语,请求 DLL 生成双向时间同步请求命令帧(见 8.4.13)。

#### ——Res Alloc 状态

处于 Res Alloc 状态的网关设备的 DMAP 负责对现场设备进行资源分配。DMAP 调用 DLME-INFO-SET.request 原语向现场设备写超帧、链路等资源。如果所有资源都写入完成或分配资源失败,则将设备状态属性(见表 15 的 DeviceState 属性)设置为 Operation。之后 DMAP 状态机转到 Operation 状态。

#### ——Leave 状态

处于 Leave 状态的 DMAP 释放现场设备所占用的 MIB 属性、通信资源等。然后 DMAP 状态机转到结束状态。

### 6.2.5.2 现场设备 DMAP 状态机

现场设备的 DMAP 所执行功能状态机如图 14 所示。

现场设备的 DMAP 所执行功能状态转移如表 13 所示。

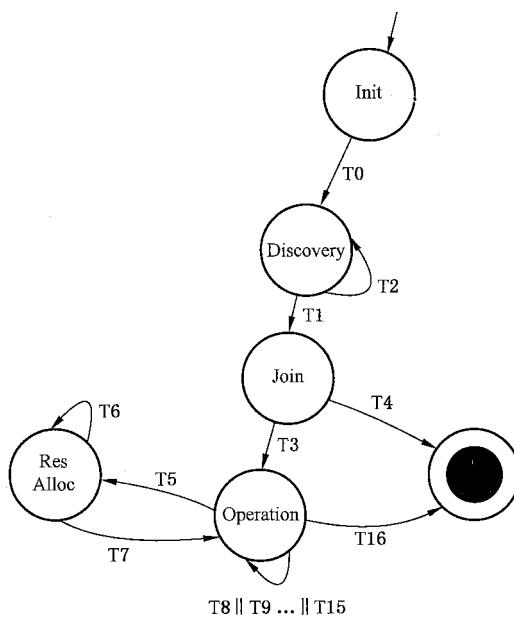


图 14 现场设备 DMAP 状态机

表 13 DMAP 状态转移表

编号	当前状态	事件/触发条件 => 动作	下一状态
T0	Init	IsDMAPInitializationDone() == TRUE => DLME-DISCOVERY.request(ScanChannels);	Discovery
T1	Discovery	DLME-DISCOVERY.confirm() && (Status == SUCCESS) => DLME-JOIN.request(NetworkID, Channel, PhyAddr, SecMaterial);	Join
T2	Discovery	DLME-DISCOVERY.confirm() && (Status == NO_BEACON) => DLME-DISCOVERY.request(ScanChannels);	Discovery
T3	Join	DLME-JOIN.confirm() && (Status == SUCCESS) => DeviceStruct.ShortAddr=ShortAddr;	Operation
T4	Join	DLME-JOIN.confirm() && (Status != SUCCESS) =>	End

表 13 (续)

编号	当前状态	事件/触发条件 => 动作	下一状态
T5	Operation	DLME-INFO-SET.indication() && (AttributeID == 131 && MemberID == 12) && (AttributeValue == ALLOCATION) => Status := WriteToMIB(Handle, ShortAddr, AttributeID := 131, MemberID := 12, FirstStoreIndex, Count, AttributeValue := ALLOCATION); DLME-INFO-SET.response(SrcAddr, AttributeOption, AttributeID := 131, MemberID := 12, FirstStroeIndex, Count, Status);	Res Alloc
T6	Res Alloc	DLME-INFO-SET.indication() && (AttributeID == 128    AttributeID == 129) => Status := WriteToMIB(Handle, ShortAddr, AttributeID, Mem- berID , FirstStoreIndex, Count, AttributeValue); DLME-INFO-SET.response(SrcAddr, AttributeOption, AttributeID, MemberID, FirstStroeIndex, Count, Status);	Res Alloc
T7	Res Alloc	DLME-INFO-SET.indication() && (AttributeID == 131 && MemberID == 12) && AttributeValue == OPERATION => Status := WriteToMIB(Handle, ShortAddr, AttributeID := 131, MemberID := 12, FirstStoreIndex, Count, AttributeValue := OPERATION); DLME-INFO-SET.response(SrcAddr, AttributeOption, AttributeID := 131, MemberID := 12, FirstStroeIndex, Count, Status);	Operation
T8	Operation	DLME-INFO-SET.indication() => Status := WriteToMIB(Handle, ShortAddr, AttributeID, Mem- berID , FirstStoreIndex, Count, AttributeValue); DLME-INFO-SET.response(SrcAddr, AttributeOption, AttributeID, MemberID, FirstStroeIndex, Count, Status);	Operation
T9	Operation	DLME-INFO-GET.indication() => Status := ReadFromMIB(Handle, ShortAddr, AttributeID, Mem- berID, FirstStroeIndex, Count); DLME-INFO-GET.response(DstAddr, Status, AttributeID, Mem- berID, FirstStoreIndex, Count, AttributeValue);	Operation

表 13 (续)

编号	当前状态	事件/触发条件 => 动作	下一状态
T10	Operation	DMAP-MIB-SET.request() => Status := WriteToMIB(Handle, ShortAddr, AttributeID, MemberID, FirstStoreIndex, Count, AttributeValue); DMAP-MIB-SET.confirm(Handle, Status);	Operation
T11	Operation	PrimitiveType == DMAP-MIB-GET.request => Status := ReadFromMIB(Handle, ShortAddr, AttributeID, Mem- berID, FirstStroeIndex, Count); DMAP-MIB-GET. confirm (Handle, Status, Count, AttributeVal- ue);	Operation
T12	Operation	DevStaRptCycle timeout => DLME-DEVICE-STATUS.request(PowerSupplyStatus);	Operation
T13	Operation	DLME-DEVICE-STATUS.confirm() =>	Operation
T14	Operation	ChaStaRptCycle timeout => DLME-CHANNEL-CONDITION.request(ChannelConditionInfo);	Operation
T15	Operation	DLME-CHANNEL-CONDITION.confirm() =>	Operation
T16	Operation	DLME-LEAVE.indication() => ReleaseResources(Addr);	End

现场设备的 DMAP 具有以下状态。

——Init 状态

处于 Init 状态的现场设备 DMAP 执行初始化过程, 初始化完成后进入 Discovery 状态。

——Discovery 状态

处于 Discovery 状态的现场设备的 DMAP 调用 DLL 的 DLME-DISCOVERY.request 原语扫描 WIA-FA 网络。DLL 调用 DLME-DISCOVERY.confirm 原语返回网络发现的结果。如果发现网络, 则转到 Join 状态; 否则转到自身 Discovery 状态, 重新进行网络发现过程。

——Join 状态

处于 Join 状态的现场设备的 DMAP 调用数据链路层的 DLME-JOIN.request 原语进行网络加入。DLL 调用 DLME-JOIN.confirm 返回加入网络的结果。如果加入成功, 则转到 Operation 状态; 否则, 转到 End 状态。

——Res Alloc 状态

Res Alloc 状态, DLL 接收到来自网关设备的远程配置属性请求(用于写超帧或者链路)后, 调用 DLME-INFO-SET.indication 原语。现场设备的 DMAP 将资源写入 MIB 库, 调用 DLME-INFO-SET.response 返回结果。如果网关设备将现场设备状态属性(见表 20 的 DeviceState 属性)设置为 Operation, DMAP 状态机转到 Operation 状态。

#### ——Operation 状态

在 Operation 状态, 有以下事件可能发生:

- a) DLL 接收到远程配置属性请求命令帧(见 8.4.17)后, 调用 DLME-INFO-SET.indication 原语, 通知 DMAP 配置现场设备的 MIB 属性。现场设备 DMAP 根据原语参数配置 MIB 属性, 并调用 DLME-INFO-SET.response 原语返回设置结果。
- b) DLL 接收到远程读属性请求命令帧(见 8.4.15)后, 调用 DLME-INFO-GET.indication 原语, 通知 DMAP 读取现场设备的 MIB 属性。现场设备 DMAP 根据原语参数读取 MIB 属性, 并调用 DLME-INFO-GET.response 原语返回 MIB 属性。
- c) ASL/DLL 调用 DMAP-MIB-GET.request 原语, 请求本地获取现场设备的 MIB 属性。现场设备 DMAP 根据原语参数获取 MIB 属性, 并调用 DMAP-MIB-GET.confirm 原语返回 MIB 属性。
- d) ASL/DLL 调用 DMAP-MIB-SET.request 原语, 请求本地设置现场设备的 MIB 属性。现场设备 DMAP 根据原语参数设置 MIB 属性, 并调用 DMAP-MIB-SET.confirm 原语返回设置结果。
- e) 每隔 DevStaRptCycle(见 6.7.1.2.1)时间, 设备状态报告定时器触发。现场设备 DMAP 调用 DLME-DEVICE-STATUS.request 原语, 向网关设备发送设备状态报告。
- f) 数据链路层调用 DLME-DEVICE-STATUS.confirm 原语, 通知 DMAP 发送设备状态报告的结果。
- g) 每隔 ChaStaRptCycle(见 6.7.1.2.1)时间, 信道状况报告定时器触发。现场设备 DMAP 调用 DLME-CHANNEL-CONDITION.request 原语, 向网关设备发送信道状况报告。
- h) DLL 调用 DLME-CHANNEL-CONDITION.confirm 原语, 通知 DMAP 发送信道状况报告的结果。
- i) DLL 接收到来自网关设备的离开请求命令帧后, 调用 DLME-LEAVE.indication 原语; 现场设备释放 MIB 属性、通信资源等, 现场设备 DMAP 状态机转到 End 状态。

#### 6.2.5.3 DMAP 状态机函数

网关设备状态机函数的定义见表 14。

表 14 DMAP 状态机函数

函数	输入	输出	函数描述
Authentication()	PhyAddr	AuthenResult	认证正在加入的现场设备 AuthenResult 取值包括: SUCCESS; FAILURE
AllocateShortAddr()	Addr	AllocateShortAddrResult	为现场设备分配短地址 AllocateShortAddrResult 取值包括: SUCCESS; FAILURE

表 14 (续)

函数	输入	输出	函数描述
IsHostComputer ConfigureDone()	—	HCCDResult	判断主控计算机的组态过程是否完成 HCCDResult 取值包括： TRUE； FALSE
ResAllocAgrithm()	SuperframeList LinkList	ResAllocResult	分配通信资源 ResAllocResult 取值包括： SUCCESS； NO_RESOURCE
IsHostComputer RequestDeviceLeave()	—	DeviceLeaveResult	判断主控计算机是否要求某个现场设备离开 WIA-FA 网络 DeviceLeaveResult 取值包括： TRUE； FALSE
IsHostComputer SetMIB()	—	HostComputer SetMIBResult	判断主控计算机是否要求远程配置现场设备的属性 HostComputerSetMIBResult 取值包括： TRUE； FALSE
IndicateSetMIB Result()	Handle Status	—	向主控计算机指示远程配置属性的结果
HandleChannel StatusReport()	Addr Count ChannelConditonInfo	—	处理信道状况报告
HandleDevice StatusReport()	ShortAddr PowerSupplyStatus	—	处理设备状态报告
IsHostComputer GetMIB()	—	HostComputer GetMIBResult	判断主控计算机是否远程读 MIB 属性 HostComputerGetMIBResult 取值包括： TRUE； FALSE
IndicateGet MIBResult()	Handle SrcAddr Status AttributeID MemberID FirstStoreIndex Count AttributeValue	—	向主控计算机指示远程读属性操作的结果

表 14 (续)

函数	输入	输出	函数描述
ReleaseResources()	Addr	—	释放某个现场设备占用的通信资源
IndicateHostComputerLeaveResult()	Addr	—	向主控计算机指示某个现场设备离开网络
WriteToMIB()	Handle ShortAddr AttributeID MemberID FirstStoreIndex Count AttributeValue	WriteToMIBResult	本地写入 MIB 属性 WriteToMIBResult 取值包括： SUCCESS; INVALIDATTRIBUTE; INVALIDATTRIBUTE_MEMBER; INVALID VALUE
ReadFromMIB()	Handle ShortAddr AttributeID MemberID FirstStroeIndex Count	ReadFromMIBResult	本地读取 MIB 属性 ReadFromMIBResult 取值包括： SUCCESS; INVALIDATTRIBUTE; INVALIDATTRIBUTE_MEMBER; INVALID RANGE
IsDMAPIInitializationDone()		DMAPIInitializationResult	判断 DMAP 初始化是否完成 DMAPIInitializationResult 取值包括： TRUE; FALSE
IsAllResAllocateDone()	—	AllResAllocateResult	判断所有的通信资源是否分配完成 AllResAllocateResult 取值包括： TRUE; FALSE

### 6.3 设备编址方法和地址分配

WIA-FA 网络中的现场设备、接入设备和网关设备都有一个全球惟一的 64 位长地址和一个 8 位或 16 位短地址(由 AddressTypeFlag 标志位指示,见表 15 定义)。当网络中现场设备数量小于 252 时,采用 8 位短地址;否则采用 16 位短地址。长地址由厂商按照 EUI-64 分配并设置,长地址的格式如图 15 所示。若网络采用 8 位短地址,则在管理信息库中的 16 位短地址属性,取其低 8 位存储。

网关设备采用 AdID(见 6.7.1.2.1)区分不同的接入设备。

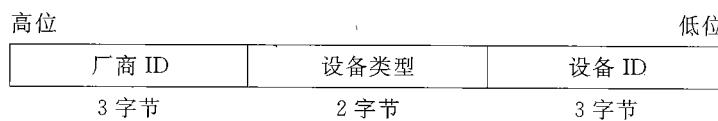


图 15 设备长地址

WIA-FA 网络中,网关设备、接入设备和现场设备的 8 位短地址具体设置如下:

- 网关设备的短地址为 0x01;
- 接入设备的短地址为 0x02;
- 现场设备的短地址取值范围为 0x03~0xFE;
- WIA-FA 网络内的广播地址为 0xFF。

WIA-FA 网络中,网关设备、接入设备和现场设备的 16 位短地址具体设置如下:

- 网关设备的短地址为 0x0001;
- 接入设备的短地址为 0x0002;
- 现场设备的短地址取值范围为 0x0003~0xFFFF;
- WIA-FA 网络内的广播地址为 0xFFFF。

设备出厂时的短地址为 0x00 或 0x0000,表示设备未被分配短地址。0x00 或 0x0000 不可以作为网络短地址使用。

## 6.4 通信资源分配

### 6.4.1 概述

通信资源包括时隙(见 3.1.31)和信道(见 3.1.8)。通信资源的分配是指根据数据优先级和通信资源占用方式,将超帧中的信道和时隙分配给接入设备和现场设备。链路属性 LinkList 包括链路 ID、链路类型、邻居设备短地址、相对时隙号、信道编号、超帧 ID(见 6.7.1.2.2)。

### 6.4.2 通信资源分配

#### 6.4.2.1 数据优先级

根据工业应用中数据的功能和要求,设置不同的数据优先级,具体包括以下五种数据优先级。

##### ——紧急数据(RT0)

RT0 数据具有最高优先级,是指对应用行为起关键作用,并要求及时传递的数据。RT0 数据通常包括:控制器制动执行器的命令、设备产生的故障/错误通知等紧急告警、主控计算机发出的时间紧迫的网络管理服务(如开始/停止命令)。RT0 数据的传输应采用 R/S 通信模型(详见 10.3.3)。

##### ——周期性过程数据(RT1)

RT1 数据具有第二优先级,是指具有严格实时性要求且周期性传输的过程数据。RT1 数据主要包括控制系统的物理测量和控制指令。RT1 数据的传输应采用 P/S 通信模型(详见 10.3.3)。

##### ——非周期性非紧急数据(RT2)

RT2 数据具有第三优先级,是指由事件驱动产生且非周期性传输的数据,如非紧急告警数据。RT2 数据的传输应采用 R/S 通信模型(详见 10.3.3)。

##### ——周期性管理数据(RT3)

RT3 数据具有第四优先级,是指具有一定实时性要求且周期性传输的设备和网络状态监视数据。RT3 数据主要包括设备状态、信道状况等。RT3 数据的传输应采用 P/S 通信模型(详见 10.3.3)。

##### ——非实时数据(NRT)

NRT 数据具有最低优先级,是指由网络操作产生的、没有严格实时性要求的数据。工业现场中,NRT 数据通常包括远程配置和管理等参数配置数据。NRT 数据的传输要求不对其他实时性数据的传输造成干扰。NRT 数据的传输应采用 C/S 通信模型(详见 10.3.3)。

上述五种优先级的数据,对应不同的通信模型,进而对应不同的 VCR。通信模型、VCR 类型及与数据的对应关系见 10.5.5.3。

#### 6.4.2.2 通信资源占用方式

不同优先级数据的通信资源占用方式包括通信资源的调度、抢占和竞争三种方式。

##### ——调度方式

调度方式用于传输周期性过程数据 RT1 和周期性管理数据 RT3。网络管理者负责整个网络的通信资源调度。接入设备和现场设备加入网络后，网络管理者为其分配固定的时隙，用以周期地传输和重传 RT1 和 RT3 数据。

##### ——抢占方式

抢占方法用于传输紧急数据 RT0，允许网关设备和现场设备直接利用周期性数据的时隙来传输紧急数据 RT0。此时，周期性数据推迟发送。

##### ——竞争方式

竞争方式利用共享发送链路传输 RT2 数据和 NRT 数据。

### 6.5 现场设备入网和离网过程

#### 6.5.1 现场设备加入过程

现场设备加入网络之前，手持设备通过有线维护端口预配置现场设备。预配置信息包括：

- 网络 ID；
- 安全等级；
- 加入密钥(安全等级不为 0 时)；
- 共享密钥(安全等级不为 0 时)。

预配置后的现场设备加入网络的过程如图 16 所示，包括：

- 接入设备周期性广播信标帧；
- 待加入网络的现场设备持续监听网络中的可用信道，获得接入设备的信标，并通过单向时间同步方法与网关设备完成时间同步(详见 8.1.4)；
- 待加入网络的现场设备选择发出信标的接入设备，根据该信标内的“共享起始相对时隙号”和“共享时隙数”(见 8.4.6)获得加入请求的共享管理时隙，并利用信标所在信道通过基于时隙的退避方式(见 8.1.7.5)竞争发送加入请求；
- 接入设备收到现场设备的加入请求后，交由网关设备的网络管理者进行处理；
- 网络管理者返回加入响应；如果加入成功，则在响应中置 Status=SUCCESS；如果存在错误，则根据错误类型，在加入响应中置 Status 值(Status 的定义见 8.3.4.3)；

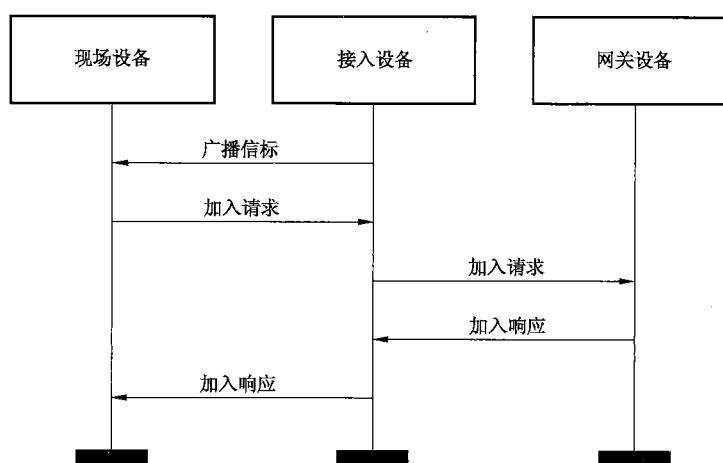


图 16 现场设备加入过程

注：安全相关内容见第 11 章。

- 接入设备发将加入响应发送给待加入网络的现场设备；
- 现场设备收到接入设备发送的加入响应；如果响应中 Status! = SUCCESS，则该现场设备重复加入过程；如果响应中 Status=SUCCESS，则该现场设备完成加入网络过程。

### 6.5.2 现场设备通信资源分配过程

现场设备加入 WIA-FA 网络后，主控计算机要求网关设备利用远程读属性服务（见 8.3.7）读取现场设备 UAO，并利用远程配置属性服务（见 8.3.8）向现场设备写入组态信息以及 VCR（见 10.5.5.3 和 10.5.5.4）。

网关设备上的网络管理者利用远程配置属性服务（详见 8.3.8）为现场设备分配通信资源，用于现场设备与接入设备之间的通信。如果现场设备的加入影响了接入设备的超帧结构（见 8.1.2），则对应的接入设备更新自身以及与其通信的现场设备的超帧属性和链路属性。

现场设备的通信资源分配过程如图 17 所示，具体包括以下过程：

- 网络管理者利用远程配置属性原语发出“远程配置属性”请求；
- 接入设备发送“远程配置属性”请求；
- 现场设备收到“远程配置属性”请求后，返回“远程配置属性”响应；
- 接入设备转发“远程配置属性”响应。

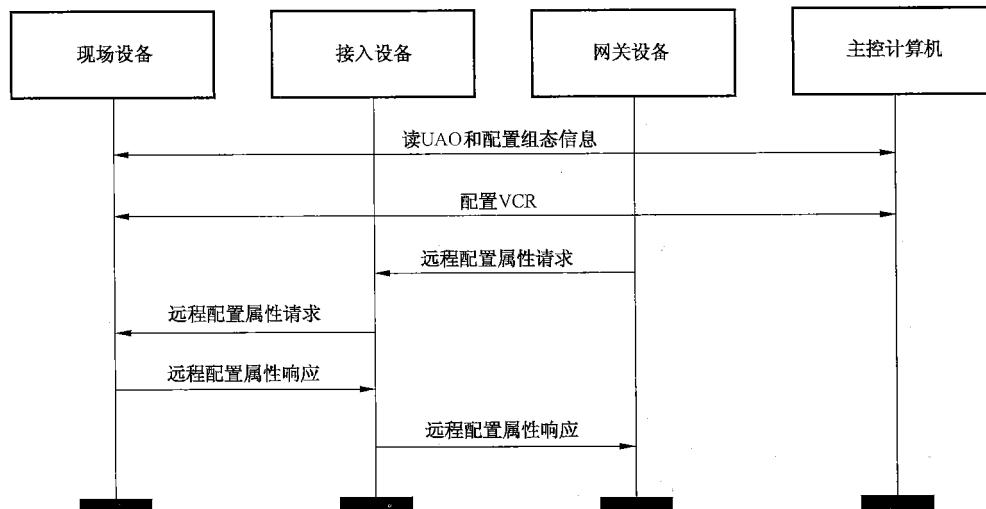


图 17 现场设备通信资源分配过程

### 6.5.3 现场设备离开过程

现场设备的离开过程包括异常离开和被动离开：

- 异常离开是指设备由于故障、失效、能量耗尽等原因无法与网络中的其他设备进行通信。当网关设备在 LossConnectDuration（见 6.7.1.2.1）时间内仍未收到现场设备的任何帧时，则认为现场设备异常离开，释放对应的现场设备所占用的短地址和通信资源。现场设备在 LossConnectDuration（见 6.7.1.2.1）时间内没有收到来自接入设备的任何帧，则认为已和接入设备失去连接，并做异常离开处理，即重启设备。
- 被动离开是指网关设备要求现场设备离开网络。

现场设备被动离开的过程如图 18 所示：

- 网关设备通过接入设备向某个现场设备发出离开请求，详见 8.4.9；

- 现场设备收到离开请求后,发出离开响应;
- 网关设备收到接入设备发来的现场设备离开响应后,由网关设备中的网络管理者处理离开响应,更新设备列表,并释放对应的现场设备所占用的短地址和通信资源;接入设备相应更新自身信息库。

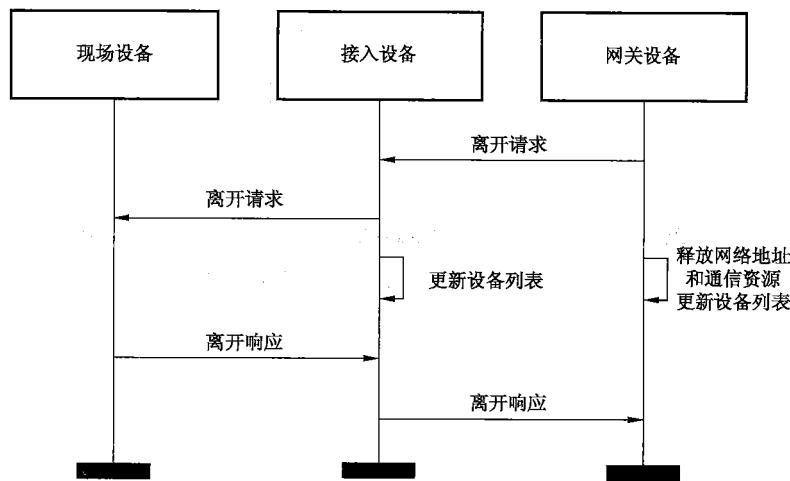


图 18 现场设备被动离开的过程

## 6.6 网络性能监视

### 6.6.1 设备状态报告

现场设备将自身的设备状态周期性汇报给网关设备中的网络管理者,设备状态详见 6.7.1.2.2 中的 DeviceList。网关设备收到现场设备的设备状态报告后,由网关设备中的网络管理者根据收到的设备状态报告,评估和诊断设备状态。设备状态报告用以检测网络内现场设备的异常情况,如电池电量不足等情况。现场设备状态报告的过程见图 19。

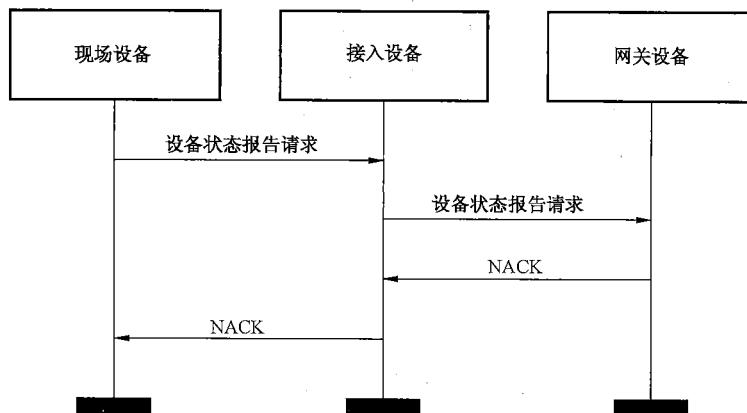


图 19 现场设备状态报告过程

### 6.6.2 信道状况报告

信道状况报告用于现场设备向网关设备中的网络管理者汇报信道的质量状况(信道状态详见 6.7.1.2.2 中的 ChannelConditionList)。信道状况报告的过程见图 20。

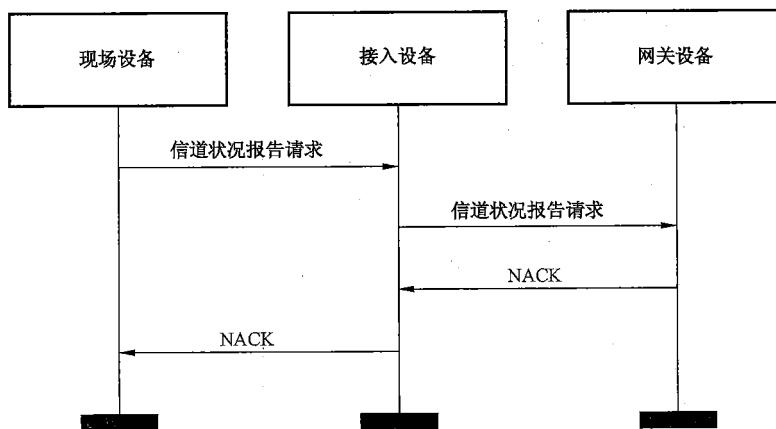


图 20 现场设备信道状况报告过程

## 6.7 管理信息库及其服务

### 6.7.1 管理信息库

#### 6.7.1.1 概述

管理信息库中的信息以属性的形式存在,用于监视和配置 WIA-FA 网络参数。属性可以被网络管理者配置、访问和更改。

按照属性的存储类型,管理信息库中属性分为以下三种:

- 常数属性 C(Constant): 指在设备运行过程中不可改变的属性。例如: 设备 64 位全球惟一地址。常数属性值在设备出厂时已被设置且不可改变。
- 静态属性 S(Static): 指在设备运行过程中改变次数很少且只能由 NM 设置的属性。例如: 设备状态汇报的周期。静态属性值在设备掉电或复位后保持之前的值。
- 动态属性 D(Dynamic): 指在设备运行过程中,在没有外部命令的情况下,设备根据自身运行状况改变的属性。动态属性值在设备掉电或复位后恢复为默认值。

按照属性的数据类型,管理信息库中的属性包括非结构化属性和结构化属性。

对管理信息库中属性进行操作的权限包括以下两种:

- 可读 R(Read): 指属性值可被网络上的其他设备读取;
- 可写 W(Write): 指属性可被网络上的其他设备修改。

按照属性执行的要求,管理信息库中的属性包括强制属性和可选属性。

管理信息库中的属性由属性标识符 AttributeID 标识。

#### 6.7.1.2 信息库属性

##### 6.7.1.2.1 非结构化属性

非结构化属性定义如表 15 所示。非结构化属性全网统一。

表 15 非结构化属性

属性标识符	属性名称	数据类型	取值范围	访问类型	存储类型	默认值	适用设备类型	描述
0	PriGwFailureTime	TimeData	0~(2 <sup>64</sup> -1)	R/W	S	10	网关设备	未收到心跳信号的最大时间 (以 μs 为单位)

表 15 (续)

属性标识符	属性名称	数据类型	取值范围	访问类型	存储类型	默认值	适用设备类型	描述
1	AddressTypeFlag	Unsigned8	0~255	R/W	S	0	所有设备	0=表示网络使用 8 位短地址 1=表示网络使用 16 位短地址；其余保留
2	MaxPayloadLength	Unsigned16	0~(2 <sup>16</sup> -1)	R/W	S	1000	所有设备	数据链路层的数据载荷的最大长度，以八位位组为单位，取值应符合 IEEE 802.11 物理层规定。
3	NACKCount	Unsigned8	0~255	R/W	S	1	所有设备	NACK 广播次数
4	NetworkID	Unsigned8	0~255	R/W	S	0	所有设备	网络的 ID 号，用于多个网络共存的情况下标识网络
5	BitMap	BitField24	—	R/W	S	0	所有设备	信道位图，用于表征物理层的调制方式和信道使用状态： 位 0~3=调制方式； 位 4~17=信道使用状态： 0 表示不使用； 1 表示使用； 其余位保留 BitMap 编码详见 7.3.3。
6	DevStaRptCycle	Unsigned16	0~65 535	R/W	S	10	网关设备 现场设备	设备状况汇报周期(以缺省超帧长度为单位)
7	ChaStaRptCycle	Unsigned16	0~65 535	R/W	S	10	网关设备 现场设备	信道状态汇报周期(以缺省超帧长度为单位)
8	LossConnectDuration	Unsigned24	0~(2 <sup>24</sup> -1)	R/W	S	30	网关设备 现场设备	如果在该时间之内没有收到来自邻居设备的任何帧，则认为邻居设备已经异常离开 WIA-FA 网络(以缺省超帧长度为单位)
9	KeyupdataDuration	Unsigned8	0~255	R/W	S	24	所有设备	设备的密钥更新周期(以 h 为单位)
10	TimeSlotDuration	Unsigned16	0~65 535	R/W	S	200	所有设备	时隙长度(以 μs 为单位)
11	TwoWayTimeSyn	Unsigned8	0~255	R/W	S	0	所有设备	是否使用双向时间同步， 0=使用单向时间同步； 1=使用双向时间同步； 其余保留

表 15 (续)

属性标识符	属性名称	数据类型	取值范围	访问类型	存储类型	默认值	适用设备类型	描述
12	TwoWayOverTime	Unsigned8	0~255	R/W	S	1	所有设备	表示双向时间同步超时的时间 (以缺省超帧长度为单位)
13	ADTeamNum	Unsigned8	0~255	R/W	S	1	网关设备	接入设备组数
14	TargetLossRate	Single Float	0~255	R/W	S	0	网关设备	WIA-FA 网络期望达到的丢包率指标, 取值为 0 到 1
15	LossRate	Single Float	0~255	R/W	S	0	网关设备	工业环境下, 当前的丢包率, 取值为 0 到 1
16	MaxRetry	Unsigned8	0~255	R/W	S	3	所有设备	设备的最大重传次数
17	SecLevel	Unsigned8	0~255	R/W	S	1	所有设备	表示数据链路层的数据包安全级别: 0=无 1=认证 2=认证 & MIC-32 3=认证 & MIC-64 4=认证 & MIC-128 5=认证 & 加密 6=认证 & 加密 & MIC-32 7=认证 & 加密 & MIC-64 8=认证 & 加密 & MIC-128 其余保留
18	AttackStatisDur	Unsigned16	0~65 535	R/W	S	60	所有设备	设备被攻击的统计周期(以分钟为单位)
19	MaxKeyAttackedNum	Unsigned8	0~255	R/W	S	5	所有设备	密钥的最大被攻击次数
20	AlarmRptDur	Unsigned8	0~255	R/W	S	1	所有设备	重复告警间隔(以缺省超帧为单位)
21	ChannelNum	Unsigned8	0~255	R/W	S	1	所有设备	一条链路可用信道数
22	AGGEnableFlag	Unsigned8	0, 1	R/W	S	0	所有设备	聚合和解聚功能启用标志: 0=不启用; 1=启用; 其余保留

### 6.7.1.2.2 结构化属性

结构化属性如表 16 所示。接入设备和现场设备应只存储和自己相关的结构化属性。网关设备应为每个设备维护其结构化属性。网关设备以设备短地址或者 AdID 为索引存储网络中每个设备的结构化属性。

表 16 结构化属性

属性标识符	属性名称	数据类型	访问类型	存储类型	适用设备类型	描述
128	SuperframeList	Superframe_Struct 结构体列表, 定义见表 17	R/W	D	所有设备	描述超帧的信息
129	LinkList	Link_Struct 结构体列表, 定义见表 18	R/W	D	所有设备	描述链路的信息
130	ChannelConditionList	ChanCon_Struct 结构体列表, 定义见表 19	R/W	D	网关设备 现场设备	记录信道状况统计的信息
131	DeviceList	Device_Struct 结构体列表, 定义见表 20	R/W	D	所有设备	记录 WIA-FA 设备的信息
132	KeyList	Key_Struct 结构体列表, 定义见表 21	R/W	D	所有设备	记录密钥的信息
133	VCRList	VcrEP_Struct 结构体列表, 定义见表 22	R/W	S	网关设备 现场设备	记录 VCR 的信息
134	SupUAOList	UAOClassDesc_Struct 结构体列表, 定义见表 23	R	S	网关设备 现场设备	现场设备支持的所有 UAO 类的描述列表
135	CfgUAOList	UAOInstDesc_Struct 结构体列表, 定义见表 25	R/W	S	网关设备 现场设备	现场设备所有被组态的 UAO 实例的描述列表

表 17 Superframe\_Struct 结构体定义

成员标识符	成员名称	数据类型	取值范围	描述
0	SuperframeID	Unsigned8	0~255	超帧的 ID 号, 由网络管理者指定
1	NumberSlots	Unsigned16	0~65 535	超帧大小(超帧中包含的时隙数)
2	ActiveFlag	Unsigned8	0~255	超帧激活标志: 0=不激活; 1=激活; 其余保留
3	ActiveSlot	Unsigned48	0~(2 <sup>48</sup> -1)	超帧激活的绝对时隙号(ASN), 等于 $\lfloor \frac{\text{Time-Value}}{\text{TimeSlotDuration}} \rfloor$

表 18 Link\_Struct 结构体定义

成员标识符	成员名称	数据类型	取值范围	描述
0	LinkID	Unsigned16	0~65 535	链路 ID 位 0 表示链路的传输类型： 0=单播； 1=广播。
1	LinkType	Unsigned8	0~255	位 1~2 表示链路的方向特性： 00=发送； 01=共享发送； 10=重传发送； 11=接收。 位 3~5 表示时隙的类型： 000=Beacon； 001=NACK； 010=GACK； 011=管理时隙； 100=数据时隙； 101=管理/数据时隙； 110~111=保留。 位 6~7 保留
2	ActiveSlot	Unsigned48	0~(2 <sup>48</sup> - 1)	链路激活的绝对时隙号(ASN), 等于 $\lfloor \text{Time-Value} / \text{TimeSlotDuration} \rfloor$
3	PeerAddr	Unsigned16	0~65 535	对端设备的短地址
4	RelativeSlotNumber	Unsigned16	0~65 535	相对时隙号
5	ChannelIndex	Unsigned8 列表		表示当前采用的信道编号, 目前使用 0~13, 其余保留。 ChannelIndex 的大小为 ChannelNum(见 6.7.1.2.1)。NM 负责分配 ChannelIndex。WIA-FA 现场设备在每个超帧的相同时隙内按照 ChannelIndex 的信道顺序切换信道
6	SuperframeID	Unsigned0	0~255	超帧 ID

表 19 ChanCon\_Struct 结构体定义

成员标识符	成员名称	数据类型	取值范围	描述
0	ChannelID	Unsigned8	0~255	信道编号, 目前使用 0~13, 其余保留
1	LinkQuality	Unsigned8	0~255	该信道上的 LQI 值
2	PacketLossRate	Single Float	0~1	该信道上的丢包率, 取值为 0 到 1
3	RetryNum	Unsigned8	0~255	该信道上的帧重传次数

表 20 Device\_Struct 结构体定义

成员标识符	成员名称	数据类型	取值范围	访问类型	存储类型	默认值	描述
0	Version	Unsigned16	0~(2 <sup>16</sup> -1)	R	C	2013	设备版本号,具体版本号因设备而异
1	LongAddress	Unsigned64	0~(2 <sup>64</sup> -1)	R	C	—	64 位全球惟一地址,其中第 4、5 八位位组(见 6.3)取值如下: 0=网关设备; 1=接入设备; 2=现场设备; 3=手持设备
2	AGGSupportFlag	Unsigned8	0~255	R	S	0	聚合和解聚功能支持标志(设备是否支持聚合和解聚功能): 0=不支持; 1=支持
3	NumOfSupUAO	Unsigned16	0~65 535	R	S	1	现场设备所支持的 UAO 类的个数
4	NumOfCfgUAO	Unsigned16	0~65 535	R/W	S	1	现场设备被组态的 UAO 实例的个数
5	TxDelay	Unsigned16	0~65 535	R/W	D	1200	设备发送帧的发送延迟时间(以 μs 为单位)
6	ProbeTime	Unsigned8	0~255	R/W	S	2	扫描一个信道的时间(以缺省超帧为单位)
7	TimeValue	TimeData	0~(2 <sup>64</sup> -1)	R/W	D	0	从零时刻开始计数的绝对时间(以 μs 为单位)
8	RedundantDevFlag	Unsigned8	0~255	R/W	S	0	指示设备是否为冗余设备: 0=否; 1=是; 其余保留
9	AdIID	Unsigned8	0~255	R/W	S	0	接入设备的 ID 号,对现场设备无效
10	DeviceShortAddress	Unsigned16	0~65 535	R/W	S	0	设备的短地址,详见 6.3
11	PowerSupplyStatus	Unsigned8	0~255	R/W	S	10	供电状态: 0=固定电源供电; 1~10=电池供电时的电量级别;10 表示电量最高,1 表示电量最低; 其余保留
12	DeviceState	Unsigned8	0~255	R/W	D	0	设备状态: 0=未加入网络; 1=正在加入网络; 2=安全认证; 3=正在组态; 4=资源分配; 5=运行; 其余保留

表 21 Key\_Struct 结构体定义

成员标识符	成员名称	数据类型	取值范围	描述
0	KeyID	Unsigned16	0~65 535	密钥标识符
1	PeerAddr	Unsigned16	0~65 535	配对地址,即邻居设备的短地址
2	KeyType	Unsigned8	0~255	密钥类型: 0=加入密钥 KJ; 1=共享密钥 KS; 2=密钥加密密钥 KEK; 3=单播数据加密密钥 KEDU; 4=广播数据加密密钥 KEDB; 其余保留
3	KeyDataValue	Octetstring	—	密钥值
4	KeyActiveSlot	Unsigned48	0~(2 <sup>48</sup> -1)	密钥激活的绝对时隙号(ASN),以微秒为单位
5	KeyAttackCnt	Unsigned16	0~65 535	密钥被攻击次数
6	AlarmFlag	Unsigned8	0~255	被检测到的该密钥的安全告警事件。当该密钥被检测到安全告警事件时,相应的位置 1。 位 0:密钥攻击告警; 位 1:密钥更新超时告警; 其余保留
7	KeyState	Unsigned8	0~255	密钥的状态: 0=备用(BACUP); 1=可用(USING); 2=过期(EXPIRED) 3=无效(INVALID); 其余保留

表 22 VerEP\_Struct 结构体定义

成员标识符	成员名称	数据类型	取值范围	描述
0	VCR_ID	Unsigned16	0~65 535	VCR 在设备内的唯一标识, VCR_ID=0 用于缺省的 C/S VCR, 其余由主控计算机组态配置
1	VcrEP_Type	Unsigned8	0~255	VCR 端点的类型, 取值如下: 0=CLIENT; 1=SERVER; 2=PUBLISHER; 3=SUBSCRIBER; 4=REPORT SOURCE; 5=REPORT SINK; 其余保留

表 22 (续)

成员 标识符	成员名称	数据类型	取值范围	描述
2	UAP_ID	Unsigned8	0~255	UAP 在设备内的惟一标识, UAP_ID = 0 用于 DMAP, 其余由主控计算机组态配置
3	PeerAddr	Unsigned16	0~65 535	对端现场设备或网关设备的段地址
4	VCRActiveTime	TimeData	0~(2 <sup>61</sup> −1)	仅对于 P/S VCR 有效, 指示 VCR 端点应被激活的绝对时间。UAP 的数据更新率 DataUpdateRate 应从该时刻开始。默认值为 0, 表示 VCR 端点应立即激活。 对于 C/S VCR 和 R/S VCR, 该值应被设为 0
5	DataUpdateRate	Unsigned32	0~(2 <sup>32</sup> −1)	仅对于 P/S VCR 有效, 指示 VCR 发布过程数据的周期(以 ms 为单位)。 对于 C/S VCR 和 R/S VCR, 该值应被设为 0
6	Deadline	Unsigned8	0~255	仅对于 P/S VCR 有效, 指示 VCR 端点自上一次接收数据后允许未接收到新数据的最大更新周期的倍数。 如果 VCR 端点在 DataUpdateRate × Deadline 时间内未接收到新数据, 则 UAP 应产生“过程数据未更新”的告警事件
7	WatchdogTime	Unsigned32	0~(2 <sup>32</sup> −1)	仅对于 C/S VCR 有效, 指示 VCR 端点应等待服务响应的最大时间(以 ms 为单位), 默认值为 100 ms。 如果 VCR 端点在该时间内未接收到服务响应, 则应返回“服务时间超时”的负响应

表 23 UAOClassDesc\_Struct 结构体定义

成员 标识符	成员名称	数据类型	取值范围	描述
0	ClassID	Unsigned8	0~255	UAO 类在设备内的惟一标识, 指示实例化 UAO 时所用的类模板
1	UAOTypte	Unsigned8	0~255	指示 UAO 类的类型, 取值如下: 0=AI; 1=AO; 2=DI; 3=DO; 其余保留
2	MaxInputDataLen	Unsigned16	1~988	UAO 类所支持的最大输入数据长度(以八位位组为单位) MaxInputDataLen ≤ 1 000—ASL 头长度(4 个八位位组)

表 23 (续)

成员标识符	成员名称	数据类型	取值范围	描述
3	MaxOutputDataLen	Unsigned16	1~988	UAO 类所支持的最大输出数据长度(以八位位组为单位) MaxOutputDataLen ≤ 1 000—ASL 头长度(4 个八位位组)
4	MinDataUpdateRate	Unsigned32	0~( $2^{32}$ - 1)	UAO 类所支持的最小过程数据更新率(以 ms 为单位)。
5	SuppInputType	ProDataDesc_Struct 结构体类型, 见表 24 定义		UAO 类的输入数据描述, 指示该 UAO 类所支持的所有输入数据的类型, 即代表 UAO 类所支持的输入数据类型的每一位都应被设为 1。如果该 UAO 类没有输入数据, 则所有位都应被设为 0
6	SuppOutputType	ProDataDesc_Struct 结构体类型, 见表 24 定义		UAO 类的输出数据描述, 指示该 UAO 类所支持的所有输出数据的类型。即代表 UAO 类所支持的输出数据类型的每一位都应被设为 1。如果该 UAO 类没有输出数据, 则所有位都应被设为 0

表 24 ProDataDesc\_Struct 结构体定义

成员标识符	成员名称	数据类型	数据长度(八位位组)	取值范围	描述
0	ParamDesc	BitField16	2	—	<p>描述 UAO 过程数据的数据类型, 以及包含相应数据类型的个数。</p> <p>位 10~15, 每位指示一种数据类型, 编码如下:</p> <ul style="list-style-type: none"> <li>位 15 指示 Unsigned8 数据类型;</li> <li>位 14 指示 Unsigned16 数据类型;</li> <li>位 13 指示 Unsigned32 数据类型;</li> <li>位 12 指示 Single Float 数据类型;</li> <li>位 11 指示 Double Float 数据类型;</li> <li>位 10 指示 Bitstring 数据类型。</li> </ul> <p>位 10~15 各位取值如下:</p> <ul style="list-style-type: none"> <li>0=不支持;</li> <li>1=支持。</li> </ul> <p>当该结构体用来描述 SupUAolist 中 UAO 类支持的 SuppInputType 或 SuppOutputType 时, 位 10~15 可能有多个位被置位, 指示该 UAO 类支持的所有数据类型。在此情况下, 位 0~9 无意义并应保留设为 0。</p>

表 24 (续)

成员标识符	成员名称	数据类型	数据长度 (八位位组)	取值范围	描述
0	ParamDesc	BitField16	2	—	<p>当该结构体用来描 CfgUAolist 中 UAO 实例被组态的 CfgInputDataList 或 CfgOutputDataList 时,位 10~15 应仅单个位被置位,指示该 UAO 实例被组态的一个输入或输出数据的数据类型。在此情况下,位 0~9 指示输入或输出数据具有相应数据类型数据的个数,取值如下:</p> <p>位 0~9=0,输入或输出数据具有 1 个相应数据类型的数据;</p> <p>位 0~9=1,输入或输出数据具有 2 个相应数据类型的数据;</p> <p>...</p> <p>位 0~9=1023,输入或输出数据具有 1024 个相应数据类型的数据</p>

表 25 UAOInstDesc\_Struct 结构体定义

成员标识符	成员名称	数据类型	取值范围	描述
0	UAO_ID	Unsigned8	0~255	UAO 在设备内的惟一标识, UAO_ID=0 用于 MIB
1	Class_ID	Unsigned8	0~255	UAO 类的类标识符,指示该 UAO 是 SuppUAolist 中标识符为 class_ID 的 UAO 类的一个实例
2	UAP_ID	Unsigned8	0~255	UAO 所属的 UAP 在设备内的标识符。如果一个 UAP 被分配了多个 UAO,则这些 UAO 具有相同的 UAP_ID
3	AckFlag	Unsigned16		该值应被赋给 UAO 的事件数据 EventData AckFlag。每一位的编码如 76 所示
4	NumInputData	Unsigned8	0~255	UAO 的输入数据个数
5	NumOutputData	Unsigned8	0~255	UAO 的输出数据个数
6	CfgInputDataList	ProDataDesc_Struct 结构体列表,见表 24 定义		UAO 输入数据的数据描述列表。列表中每个成员指示一个输入数据的数据类型以及该输入数据具有相应数据类型数据的个数,并且列表成员的顺序规定了应周期性传输的输入数据的顺序
7	CfgOutputDataList	ProDataDesc_Struct 结构体列表,见表 24 定义		UAO 输出数据的数据描述列表。列表中每个成员指示一个输出数据的数据类型以及该输出数据具有相应数据类型数据的个数,并且列表成员的顺序规定了应周期性传输的输出数据的顺序

## 6.7.2 管理信息库服务

### 6.7.2.1 概述

对管理信息库中属性的本地读写操作使用 DMAP 提供的读属性和写属性服务。

### 6.7.2.2 读属性原语

读属性请求原语用于设备各层请求读取 DMAP 管理信息库中的属性。

DMAP-MIB-GET.request(

```
Handle,  
ShortAddr,  
AttributeID,  
MemberID,  
FirstStoreIndex,  
Count
```

)

DMAP-MIB-GET.request 原语的参数说明如表 26 所示。

表 26 DMAP-MIB-GET.request 原语的参数

参数名称	数据类型	取值范围	描述
Handle	Unsigned8	0~255	调用读属性请求原语时分配的句柄
ShortAddr	Unsigned16	0~65 535	现场设备的 8 位或 16 位短地址,或者接入设备的 AdID (见 6.7.1.2.1)。短地址适用于网关设备读取现场设备或者接 入设备的 MIB 属性
AttributeID	Unsigned8	0~255	信息库中属性的标识符
MemberID	Unsigned8	0~255	属性成员标识符。如果该值为 255,则表示取全部的属性成 员。MemberID 对非结构化属性无效
FirstStoreIndex	Unsigned16	0~65 535	要读取属性或属性成员的第一个值的存储索引,FirstStore- Index 对非结构化属性无效
Count	Unsigned16	0~65 535	属性或属性成员的个数,用来表示需要读取属性或者属性成 员的数量。如果 Count=0,读取从 FirstStoreIndex 开始的所 有属性。该参数仅适用于读取结构化属性

读属性证实原语用来返回读取属性的结果。

DMAP-MIB-GET.confirm(

```
Handle,  
Status,  
Count,  
AttributeValue
```

)

DMAP-MIB-GET.confirm 原语的参数说明如表 27 所示。

表 27 DMAP-MIB-GET.confirm 原语的参数

参数名称	数据类型	取值范围	描述
Handle	Unsigned8	0~255	调用读属性请求原语时分配的句柄
Status	Unsigned8	0~255	获取信息库属性的结果： 0=SUCCESS; 1=INVALIDATTRIBUTE; 2=INVALID ATTRIBUTE MEMBER; 3=INVALID RANGE; 其余保留
Count	Unsigned16	0~65 535	属性或者属性成员的个数,用来表示返回属性或者属性成员的数量。如果 Count=0,读取从 FirstStoreIndex 开始的所有属性。仅当 Status=0 时,该值有效。该参数仅适用于读取结构化属性
AttributeValue	Octetstring	-	属性或者属性成员的属性值。仅当 Status =0 时,该值有效

如果读取管理信息库中的属性值成功,则 Status 返回“SUCCESS”,AttributeValue 参数有效;如果读取的属性标识符不在定义范围内,则返回“INVALID ATTRIBUTE”;如果读取的属性成员的标识符不在定义范围内,则 Status 返回“INVALID ATTRIBUTE MEMBER”;如果由 FirstStoreIndex 和 Count 计算出来的属性记录超限,则 Status 返回“INVALID RANGE”。

#### 6.7.2.3 写属性原语

写属性请求原语用于设备各层请求向 DMAP 管理信息库中写入属性值。

DMAP-MIB-SET.request(

```

        Handle,
        ShortAddr,
        AttributeID,
        MemberID,
        FirstStoreIndex,
        Count,
        AttributeValue
    )

```

DMAP-MIB-SET.request 原语的参数说明如表 28 所示。

表 28 DMAP-MIB-SET.request 原语的参数

参数名称	数据类型	取值范围	描述
Handle	Unsigned8	0~255	调用写属性请求原语时分配的句柄
ShortAddr	Unsigned16	0~65 535	现场设备的 8 位或 16 位短地址,或者接入设备的 AdID(见 6.7.1.2.1)。 短地址适用于网关设备向现场设备或者接入设备写入 MIB 属性
AttributeID	Unsigned8	0~255	信息库中属性的标识符。

表 28 (续)

参数名称	数据类型	取值范围	描述
MemberID	Unsigned8	0~255	属性成员标识符。如果该值为 255，则表示写入全部的属性成员。 MemberID 对非结构化属性无效
ValueStorIndex	Unsigned16	0~65 535	所写属性值的存储索引
FirstStoreIndex	Unsigned16	0~65 535	要写属性的第一个值的存储索引，FirstStoreIndex 对非结构化属性无效
Count	Unsigned16	0~65 535	属性或属性成员的个数，用来表示需要写属性或属性成员的数量。如果 Count=0，从 FirstStoreIndex 开始写所有属性。该参数仅适用于读取结构化属性
AttributeValue	Octetstring	—	表示写入管理信息库中的属性或属性成员的值

写属性证实原语用于返回写属性的结果。

DMAP-MIB-SET.confirm(

```
Handle,
Status
)
```

DMAP-MIB-SET.confirm 原语的参数说明如表 29 所示。

表 29 DMAP-MIB-SET.confirm 原语的参数

参数名称	数据类型	取值范围	描述
Handle	Unsigned8	0~255	调用写属性请求原语时分配的句柄
Status	Unsigned8	0~255	写管理信息库属性请求原语的结果： 0 = SUCCESS; 1 = INVALID ATTRIBUTE; 2 = INVALID ATTRIBUTE MEMBER; 3 = INVALID VALUE; 其余保留

如果写管理信息库中的属性值成功，则 Status 返回“SUCCESS”，AttributeValue 参数有效；如果所写的属性标识符不在定义范围内，则返回“INVALID ATTRIBUTE”；如果所写的属性成员的标识符不在定义范围内，则 Status 返回“INVALID ATTRIBUTE MEMBER”；如果所写 AttributeValue 超出有效范围，则 Status 返回“INVALID VALUE”。

## 7 物理层

### 7.1 概述

WIA-FA 物理层采用基于 IEEE STD 802.11-2012 的物理层进行通信，可采用不同调制解调方式，例如，FHSS/DSSS/OFDM。

## 7.2 WIA-FA 物理层一般性要求

表 30 所示为 WIA-FA 物理层采纳 IEEE STD 802.11—2012 物理层部分功能的列表。

表 30 物理层选择列表

章节	条目	是否保留	约束条件
7	PHY service specification		
7.1	Scope	是	
7.2	PHY functions	是	
7.3			
7.3.1	Scope and field of application	是	
7.3.2	Overview of the service	是	
7.3.3	Overview of interactions	是	
7.3.4			
7.3.4.1	General	是	
7.3.4.2	PHY-SAP peer-to-peer service primitives	是	
7.3.4.3	PHY-SAP sublayer-to-sublayer service primitives	是	
7.3.4.4	PHY-SAP service primitives parameters	是	
7.3.4.5	Vector descriptions	是	
7.3.5			
7.3.5.1	Introduction	是	
7.3.5.2	PHY-DATA.request	是	
7.3.5.3	PHY-DATA.indication	是	
7.3.5.4	PHY-DATA.confirm	是	
7.3.5.5	PHY-TXSTART.request	是	
7.3.5.6	PHY-TXSTART.confirm	是	
7.3.5.7	PHY-TXEND.request	是	
7.3.5.8	PHY-TXEND.confirm	是	
7.3.5.9	PHY-CCARESET.request	否	
7.3.5.10	PHY-CCARESET.confirm	否	
7.3.5.11	PHY-CCA.indication	否	
7.3.5.12	PHY-RXSTART.indication	是	
7.3.5.13	PHY-RXEND.indication	是	
7.3.5.14	PHY-CONFIG.request	是	
7.3.5.15	PHY-CONFIG.confirm	是	
7.4	PHY management	是	
14	Frequency-Hopping spread spectrum (FHSS) PHY specification for the 2.4GHz industrial, scientific, and medical (ISM) band	是	

表 30 (续)

章节	条目	是否保留	约束条件
15	Infrared (IR) PHY specification	否	
16	DSSS PHY specification for the 2.4GHz band designated for ISM applications	是	
17	High Rate direct sequence spread spectrum (HR/DSSS) PHY specification	是	
18	Orthogonal frequency division multiplex (OFDM) PHY specification	部分	2.4GHz
19	Extended rate PHY (ERP) specification	部分	2.4GHz
20	High Throughput (HT) PHY specification	部分	2.4GHz

### 7.3 WIA-FA 物理层附加要求

#### 7.3.1 概述

符合 WIA-FA 规范的设备可采用 IEEE STD 802.11-2012 2.4 GHz 定义的各种调制方式。本节定义 WIA-FA 设备采用的频段、信道、传输功率以及传输速率。

#### 7.3.2 频段

WIA-FA 网络使用 IEEE STD 802.11-2012 物理层中的 2.4 GHz 频段。根据 IEEE STD 802.11-2012 物理层,不同国家定义不同的 2.4 GHz 频段。例如,中国、美国和欧洲定义的 2.4 GHz 频段为 2.4 GHz~2.483 5 GHz;日本定义为 2.471 GHz~2.497 GHz。

#### 7.3.3 信道位图

WIA-FA 网络利用 BitMap(见 6.7.1.2.1)描述可用信道。BitMap 采用 3 个八位位组表示所采用的调制方式和信道索引,BitMap 的格式如图 21 所示。

位:0~3	位:4~17	位:18~23
调制方式	信道使用状态	保留

图 21 BitMap 格式

各域说明如下:

- 调试方式:长度为 4 个比特,其编码方式见表 31;
- 信道使用状态:位 4~17 对应如表 21 所示不同调制方式下使用的信道,见表 32。其中,位 4 对应信道编号为 1 的信道,位 4 的值为 0 表示对应信道 1 不可用,位 4 的值为 1 对应信道 1 可用;位 5~17 依次表示信道 2~信道 14 的使用状态。

表 31 调制方式编码

编码 位 3~位 0	调制方式
0000	FHSS
0001	DSSS
0010	HR/DSSS
0011	OFDM
0100	ERP-DSSS
0101	ERP-CCK
0110	ERP-OFDM
0111	ERP-PBCC
1000	DSSS-OFDM
1001	No-HT 模式
1010	mix 模式
1011	HT 模式

可用信道如表 32 所示。

表 32 信道编号

调制方式	信道编号
FHSS	—
DSSS	信道编号 1~14
HR/DSSS	信道编号 1~14
OFDM	无定义
ERP	ERP-DSSS ERP-CCZ ERP-OFDM ERP-PBCC DSSS-OFDM
HT	No-HT 模式 mix 模式 HT 模式

#### 7.3.4 传输功率

WIA-FA 设备的传输功率定义为等效各向同性辐射功率 EIRP。WIA-FA 设备的 EIRP 为 10 dBm±3 dBm。WIA-FA 设备的最大传输功率应低于所布设国家的调制要求。

#### 7.3.5 传输速率

WIA-FA 网络采用 IEEE STD 802.11-2012 2.4GHz 下的传输速率。WIA-FA 设备支持的传输速

率具体描述如表 33 所示。

表 33 传输速率

调制方式		传输速率(Mbps)
<b>FHSS</b>		1/1.5/2/2.5/3/3.5/4/4.5
<b>DSSS</b>		1/2
<b>HR/DSSS</b>		1/2/5.5/11
<b>OFDM</b>		5 MHz 信道间隔下的速率:1.5/2.25/3/4.5/6/9/ 12/13.5,其中,必须支持 1.5/3/6 传输速率
<b>ERP</b>	ERP-DSSS	1/2
	ERP-CCK	5.5/11
	ERP-OFDM	6/9/12/18/24/36/48/54
	ERP-PBCC	5.5/11/22/33
	DSSS-OFDM	6/9/12/18/24/36/48/54
<b>HT</b>	No-HT 模式	支持 ERP PHY 的速率
	mix 模式	无定义
	HT 模式	不支持

## 8 数据链路层

### 8.1 概述

WIA-FA 数据链路层(DLL)的主要功能是保证 WIA-FA 现场设备和接入设备间的实时、可靠、安全地传输。WIA-FA 的 DLL 包括:

- 数据链路层数据传输功能:采用基于超帧的 TDMA 机制,保证数据无碰撞、实时可靠地发送与接收;支持帧聚合/解聚。
- 数据链路层管理功能:定义设备加入、离开、时间同步、属性读写等功能。

#### 8.1.1 协议栈结构

图 22 所示为 WIA-FA 数据链路层的协议栈结构。WIA-FA 数据链路层为应用层提供服务接口。数据链路层包括数据链路层数据实体(DLDE)和数据链路层管理实体(DLME)。DLDE 负责提供数据服务接口 DLDE-SAP;DLME 提供加入、离开、时间同步、配置参数和监视 DLL 运行状态的管理服务等管理接口 DLME-SAP。

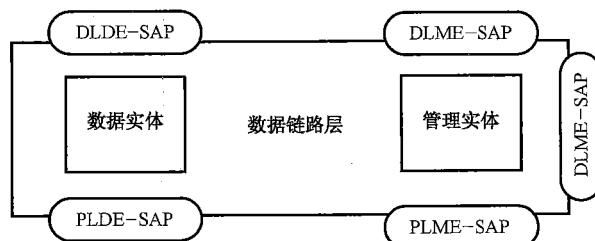


图 22 WIA-FA 数据链路层协议栈结构

### 8.1.2 WIA-FA 超帧

为了实现实时和可靠传输数据,WIA-FA 超帧采用 TDMA 接入机制。从时间的角度,WIA-FA 超帧由若干循环的等长时隙组成,每个时隙的结构如图 23 所示。时隙长度可配置,每个时隙只负责完成一个帧的传输。从通信资源的角度,WIA-FA 超帧由若干链路组成,每条链路包括时隙和信道两类通信资源。表 34 给出了时隙结构模板。

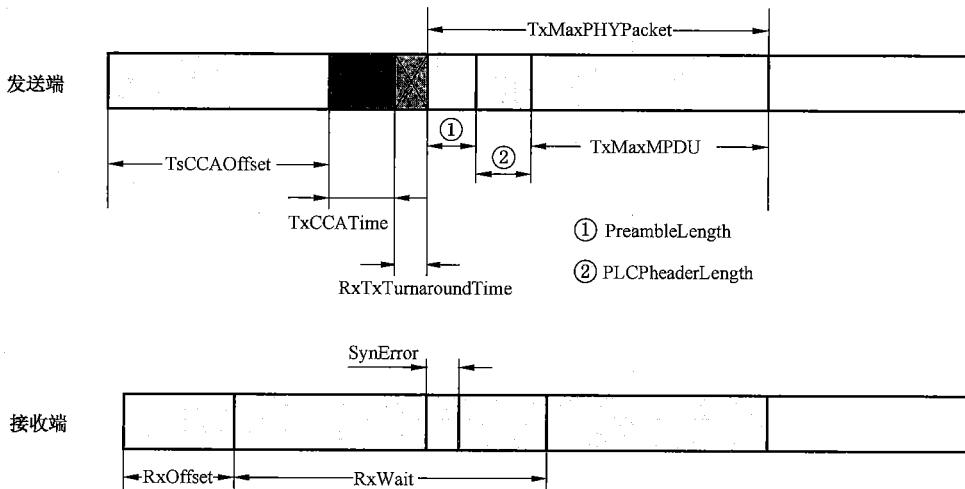


图 23 时隙结构模板

表 34 时隙结构模板参数

参数名称	描述
TsCCAOffset	从时隙开始,到开始执行 CCA 的时间(以 $\mu s$ 为单位)
TsCCATime	执行 CCA 的时间(8 symbols)
RxTxTurnaroundTime	收状态切换到发状态的最长时间(以 $\mu s$ 为单位)
PreambleLength	物理层前导码传输时间
PLCPheaderLength	PLCP 头的传输时间
TxMaxDPDU	传输最长 DLPDU 所需时间(以 $\mu s$ 为单位)
TxMaxPHYPacket	传输最长物理层报文所需时间,其值为 PreambleLength + PLCPheaderLength + TxMaxMPDU(以 $\mu s$ 为单位)
RxOffset	从时隙开始,到收发器开始侦听的时间(以 $\mu s$ 为单位)
RxWait	接收端等待报文传输开始的最短时间,取决于时间漂移(以 $\mu s$ 为单位)
SynError	报文传输的实际起始时间与理想时间的差值,即接收端与发送端之间的时间同步偏差(以 $\mu s$ 为单位)

WIA-FA 网络初始化后,网关设备首先维护一个初始超帧。初始超帧的结构如图 24 所示,由信标帧(Beacon)时隙、管理时隙和数据时隙组成。初始超帧的超帧表设置如下:

- superframeID 设置为 0;
- 初始超帧的长度默认值设置为 50 ms;
- ActiveFlag 和 ActiveSlot 的值设置为 0,可被用户修改。

初始超帧的结构由信标帧广播,信标帧格式见 8.4.6。

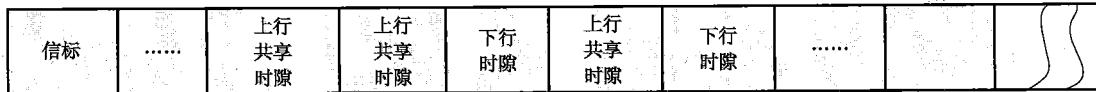


图 24 WIA-FA 初始超帧

初始超帧中各类时隙的功能如下:

- 接入设备在初始超帧的信标时隙发送信标(见 8.4.6),用于现场设备加入网络(见 8.3.4.5)。
- 现场设备和网关设备分别利用信标载荷域指定上行共享时隙和下行时隙(见 8.4.6)发送加入请求和加入响应(见 8.3.4.5)。
- 网关设备利用共享时隙中的下行时隙为现场设备组态。
- 网关设备利用共享时隙中的下行时隙,调用远程配置属性服务(见 8.3.8)为现场设备配置通信资源。

现场设备入网后,网络管理者可为设备配置多个超帧。超帧的数量和长度由现场设备中 UAO 的数据更新率以及现场设备的汇报周期决定。相同数据更新率和汇报周期的链路共用一个超帧,即 N 类数据更新率和汇报周期,则配置 N 个超帧。例如,如果现场设备 UAO 的数据更新率分别为 1 ms、2 ms、16 ms,设备状态和信道状况汇报周期分别为 2 ms 和 16 ms,加上现场设备的缺省超帧(长度为 50 ms),则现场设备将被配置 4 个超帧。网关设备维护网络中所有设备的超帧。接入设备的超帧数量和结构与所管理现场设备的超帧相同。

所配置超帧的结构如图 25 所示。

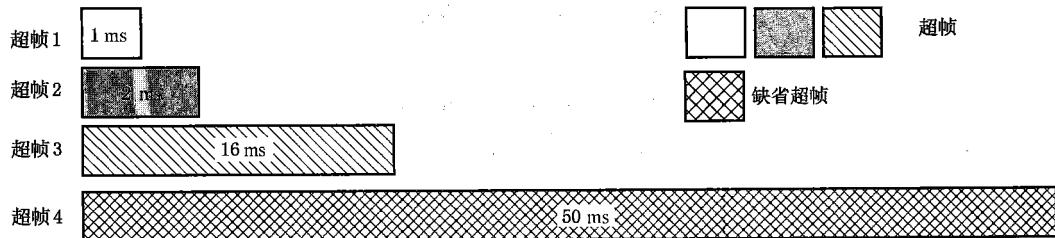


图 25 WIA-FA 超帧

WIA-FA 网络设备在各个不重叠的信道上采用 TDMA 接入机制进行通信,图 26 所示为 WIA-FA 设备利用多条信道进行通信的示例。

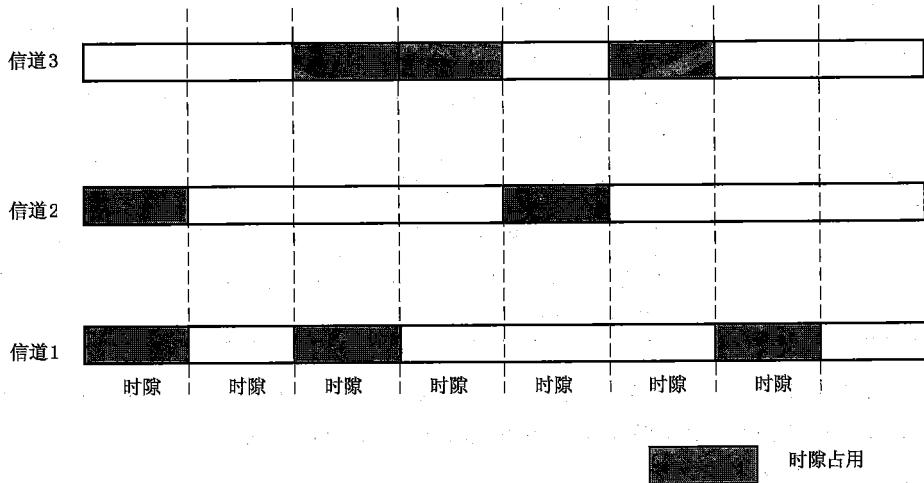


图 26 WIA-FA 设备多信道通信示例

### 8.1.3 基于多接入设备的通信

#### 8.1.3.1 基于多接入设备的信标帧传输

WIA-FA 网络中可存在多个接入设备,且多个接入设备与网关设备采用有线连接方式(见第 9 章)。网络管理者可根据当前可用信道数(见 BitMap,6.7.1.2.1)将这些接入设备分成多个集合,为每个集合分配一个用于发送信标帧的信道。对于每个集合中的接入设备,还可将其分成多组发送信标帧,分组的数目由 ADTeamNum(见 6.7.1.2.1)确定。NM 根据现场应用环境,确定分组。具体分组方法不在本部分定义范围内。

网关设备中的网络管理者为每个接入设备分配一个惟一的 AdID(见 9.2), WIA-FA 网络中的信标帧是由接入设备广播给现场设备的. 网络管理者采用 Beacon 分组发送策略, 具体过程包括以下步骤:

- 将初始超帧划分成  $ADTeamNum$  个阶段。其中，每个阶段对应一组接入设备；
  - 任意组对应阶段的开始预留信标时隙，预留时隙的数量等于组内接入设备的数量；
  - 计算每个组内的每个接入设备发送信标所需的时隙数  $TimslotCount$ ，见式(1)；

$$\text{TimeslotCount} = \text{SuperframeLength}/\text{ADTeamNum} \times \text{TeamID} + \text{InTeamID} \quad \dots \dots \dots \quad (1)$$

其中，

SuperframeLength:表示缺省超帧长度;

TeamID:表示组标识符,从0开始标识;

InTeamID: 表示每个组内的接入设备的标识符,从0开始标识。

对于每个现场设备，只要保证在初始超帧长度内接收任意一个接入设备广播的信标帧，即可完成时间同步。通过多个阶段的多次时间同步，可以提高时间同步精度。

如图 27 所示,图 27a) 中为单个接入设备时的一个 TDMA 超帧示例,此时超帧长度为 30,在超帧内的相对时隙号为 0 的时隙发送信标帧;图 27b) 为多个接入设备时的一个超帧示例,此时 WIA-FA 网络内有 6 个接入设备,将其分为两组,分组结果为:AD11、AD12、AD13 为第一组,AD21、AD22、AD23 为第二组。同时将原来的一个 TDMA 超帧划分为两个阶段,预留在每个阶段开始的前 3 个时隙,用于组内广播信标帧。

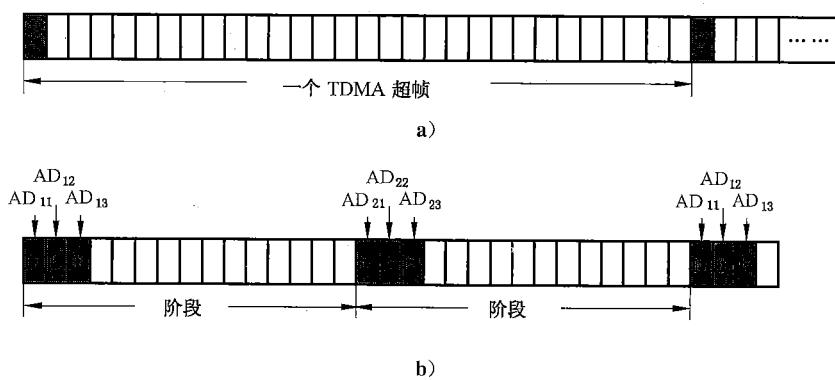


图 27 单个接入设备时的超帧示例

#### 8.1.3.2 基于多接入设备的其他帧传输

当现场设备按照网络管理者分配的链路发送帧时,可能有多个接入设备同时接收到该帧,网关设备通过帧序列号过滤掉重复帧。

当网关设备向现场设备发送帧时，网关设备应选择一个接入设备，利用网络管理者分配的链路发送

(具体的选择算法本部分不做定义)。

#### 8.1.4 时间同步

WIA-FA 网络规定将网关设备的内部时钟设为时钟源。遵照 GB/T 25931—2010,所有接入设备以有线连接方式与网关设备保持严格的时间同步。WIA-FA 网络具体支持两种时间同步方式:

- 若管理信息库中 TwoWayTimeSyn(见 6.7.1.2.1)为 0,则现场设备通过信标帧与接入设备进行单向时间同步。
- 若管理信息库中 TwoWayTimeSyn(见 6.7.1.2.1)为 1,则现场设备入网前通过信标帧与接入设备进行单向时间同步,加入网络后的首个超帧内与接入设备进行双向时间同步,之后超帧中执行单向时间同步。将双向时间同步中得到的帧发送时间记录在管理信息库 TxDelay 中(见 6.7.1.2.1)

单向时间同步,即接入设备周期性发送信标帧,现场设备接收到信标帧后,根据信标帧中的时戳值校准本地时间值,以达到全网时间同步的目的。单向时间同步流程如图 28 所示,信标帧格式详见8.4.6。

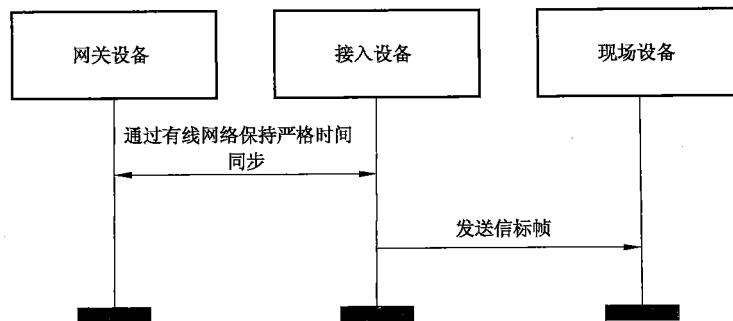


图 28 单向时间同步流程

双向时间同步,即现场收到信标帧后在上行共享时隙内向接入设备发送时间同步请求帧(详见 8.4.6)。接入设备随后在管理时隙上向现场设备发送双向时间同步响应帧,该响应帧的载荷包括来自现场设备的时间同步请求帧内的“现场设备发送时刻时间值”以及该接入设备收到时间同步请求帧时的时间值。现场设备根据收到的信标帧中的绝对时间值、收到信标帧时刻的时间值、响应帧中的“现场设备发送时刻时间值”及“接入设备接收到时间同步请求帧的时间值”进行时间同步,计算发送一个帧的发送时间 TxDelay(见表 20),并记录在管理信息库中。TxDelay 的计算公式如式(2):

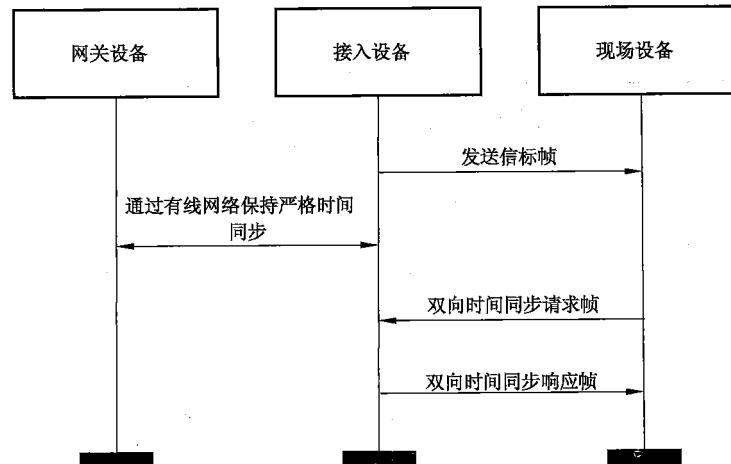


图 29 双向时间同步流程

$$\text{TxDelay} = [(\text{收到信标帧时刻的时间值} - \text{信标帧中的绝对时间值}) + (\text{接入设备接收到双向时间同步请求帧的时间值} - \text{现场设备发送时刻时间值})]/2$$

.....(2)

### 8.1.5 帧聚合/解聚

WIA-FA 数据链路层支持数据帧的聚合/解聚机制,以减少数据帧传输的次数。

帧聚合适用于 RT1 和 RT2 优先级,聚合帧的数据具有相同优先级,例如 RT1 或者 RT2 优先级。

帧聚合/解聚为数据链路层功能。接入设备支持帧聚合,现场设备支持帧解聚。当接入设备的数据链路层需要向现场设备发送 N 个数据帧时,接入设备计算聚合后的帧长度,若小于聚合帧最大长度 MaxPayloadLength(见 6.7.1.2.1),则接入设备的数据链路层聚合数据帧,每个现场设备收到聚合的数据帧后,对应地将发送给自己的数据帧解聚出来,这样可以减少接入设备发送给现场设备的数据帧数目,提高网络效率。

帧聚合由接入设备的数据链路层完成,聚合功能的配置过程如下:

- 主控计算机通过读取现场设备和接入设备聚合/解聚功能支持标志 AGGSupportFlag(见 6.7.1.2.1),判断设备是否支持聚合功能。当现场设备和接入设备的 AGGSupportFlag 都为 1 时,主控计算机继续下面的配置过程;否则将现场设备、接入设备和网关设备的 MIB 中的聚合/解聚功能启用标志 AGGEnableFlag(见 6.7.1.2.1)清零,即不启用聚合/解聚功能。
- 当选择启用帧聚合/解聚功能时,NM 将支持聚合功能的网关设备、接入设备和现场设备 MIB 中的帧聚合/解聚功能启用标志 AGGEnableFlag 置 1,启用接入设备和现场设备的帧聚合/解聚功能。

针对接入设备是否启用聚合功能,做以下处理:

- 如果接入设备管理信息库中的帧聚合/解聚功能启用标志 AGGEnableFlag 为 0,则该接入设备不启用帧聚合功能。
- 如果接入设备管理信息库中的帧聚合/解聚功能启用标志 AGGEnableFlag 为 1,则该接入设备启用帧聚合功能。接入设备按照图 30 所示的格式逐个来自多个现场设备的多个帧,并保证聚合后的帧长度小于 MaxPayloadLength(见 6.7.1.2.1);接入设备的 DLL 将数据链路层帧头的帧控制域中的帧类型(见图 48)设置为聚合帧,利用之前网络管理者为接入设备预先分配的广播时隙(对应的多个现场设备为接收时隙)发送该聚合帧。

	第 1 帧			...	第 N 帧		
1 八位位组	1 或 2 八位位组	2 八位位组	可变长度	...	1 或 2 八位位组	2 八位位组	可变长度
聚合数量	现场设备 地址	数据长度	数据	...	现场设备 地址	数据长度	数据

图 30 接入设备聚合帧载荷格式

图 30 中各参数的定义如下:

- 聚合数量:长度为 1 个八位位组,表示所聚合的发送给现场设备的帧数;
- 现场设备地址:长度为 1 或 2 个八位位组,表示其后被聚合数据的目的地址;
- 数据长度:长度为 2 个八位位组,表示其后被聚合的发送给现场设备的数据的长度,以八位位组为单位;
- 数据:可变长度,表示被聚合的发送给现场设备的数据。

现场设备解聚功能的设置和使用具体如下:

——若现场设备和接入设备的聚合/解聚功能支持标志 AGGSupportFlag 为 1,且网络管理者将现场设备和接入设备的 AGGEnableFlag 标识设置为 1 时,解聚功能与聚合功能同时启用,即接入设备启用帧聚合功能时,现场设备启用解聚功能。

——现场设备根据收到的 DLL 帧头决定是否解聚。如果 DLL 帧头的控制域的帧类型为聚合帧，则现场设备需要按照聚合帧载荷(图 30)格式解聚。

### 8.1.6 分段和重组

数据链路层负责数据单元的分段和重组。如果 APDU 长度大于 DLPDU 载荷允许的最大长度，则发送端的数据链路层根据需要进行分段。接收端收到分段的 DLPDU 后，要在数据链路层进行重组。具体的包格式见 8.4.1。

### 8.1.7 重传

#### 8.1.7.1 重传方式

WIA-FA 支持如下重传方式：

- a) 基于 NACK 的重传方式: 现场设备向网关设备发送周期性数据时, 采用基于否定应答帧(NACK)的重传方式。
  - b) 多次单播重传方式: 网关设备向现场设备周期性发送非聚合数据时, 采用多次单播重传方式。
  - c) 多次广播重传方式: 网关设备向现场设备周期性发送聚合数据时, 采用聚合帧广播重传方式。
  - d) 基于 GACK 的时隙退避重传方式: 现场设备向网关设备发送非周期性数据帧或管理帧时, 例如, 远程读属性、远程配置属性、双向时间同步, 现场设备根据网关设备广播的 GACK 帧采用基于时隙退避的重传方式。

网关设备向现场设备发送非聚合广播帧时，不需要回复确认，因此不需要重传。

此外，WIA-FA 网络需要多次广播 NACK 帧以及 GACK 帧，以进一步保证数据传输的可靠性。

#### 8.1.7.2 基于 NACK 的重传方式

基于 NACK 的重传方式的步骤如下：

——针对现场设备向网关设备发送周期性数据的需求，网络管理者在每个超帧周期内配置几组连续的重传时隙，用于现场设备重传周期性过程数据或周期性管理数据。重传时隙的组数和每组的时隙数由 WIA-FA 网络的可靠性指标 (TargetLossRate 属性定义见 6.7.1.2.1) 以及工业现场的信道丢包率 (Lossrate 属性定义见 6.7.1.2.1) 决定。

一种预留重传时隙的方法如下所述。

首先根据网络可靠性指标以及工业现场的信道丢包率,计算得到网络最小重传次数,见式(3):

则重传次数(`MaxRetry`属性定义见 6.7.1.2.1), 见式(4):

MaxRetry >= minRetryTime .....( 4 )

并计算得到第  $n$  ( $1 \leq n \leq \text{MaxRetry}$ ) 次重传的最小时隙数, 见式(5):

$$\min\text{RetrySlotNum}[n] \geq \text{FrameCount} \times \text{LossRate}^n$$

( 5 )

其中 FrameCount 为本超帧周期内现场设备需要传输给网关的周期性帧数目。

——多个现场设备依次利用自身的发送时隙发送周期性数据帧/管理帧给网关设备；

- 多个现场设备的周期性数据帧/管理帧发送完成后,网关设备统计发送失败的现场设备短地址,将其按一定顺序排列构造 NACK 帧(见 8.4.4),随后网关设备通过接入设备重复广播 NACK 帧,NACK 帧的广播次数等于 MaxRetry;
- 现场设备解析收到的 NACK 帧后,如果 NACK 帧中包含自身的短地址,则根据短地址在 NACK 帧中的排列顺序,确定占用重传时隙的顺序,并在对应的重传时隙号内重传周期性数据帧/管理帧;如果当次分配的重传时隙不足,则等待下一个 NACK 指示的重传机会。

如图 31 所示为基于 NACK 的重传方式的示例。假设工业现场环境中的丢包率为 0.1,可靠性指标为 0.01%,则 minRetryTime = 4,取 MaxRetry = 4。各个现场设备首先利用网络管理者为其分配的时隙向网关设备发送周期性数据帧,网关设备依据是否收到来自现场设备的帧,构造 NACK 帧,并重复广播 4 次给现场设备;现场设备根据 NACK 帧,在预留的重传时隙内重传数据帧给网关设备,以保障数据传输的可靠性。

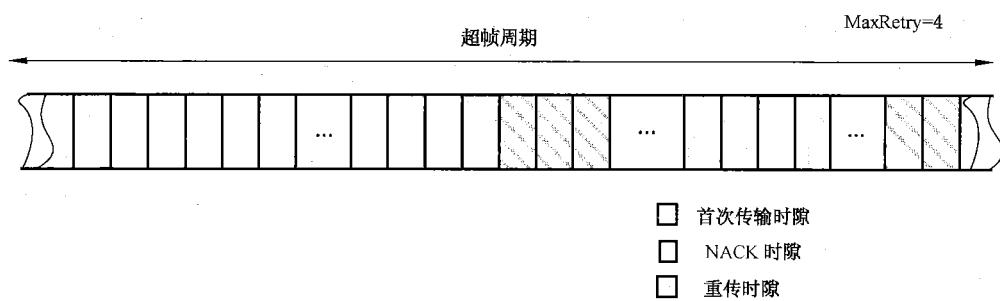


图 31 基于 NACK 的重传方式示例

#### 8.1.7.3 多次单播重传方式

网关设备向现场设备周期性发送非聚合数据帧或管理帧时,采用多次单播重传方式。根据 WIA-FA 网络的最大重传次数(MaxRetry 属性定义见 6.7.1.2.1),网关设备多次单播该周期性非聚合数据帧或管理帧,直至达到 MaxRetry 次数,如图 32 所示。

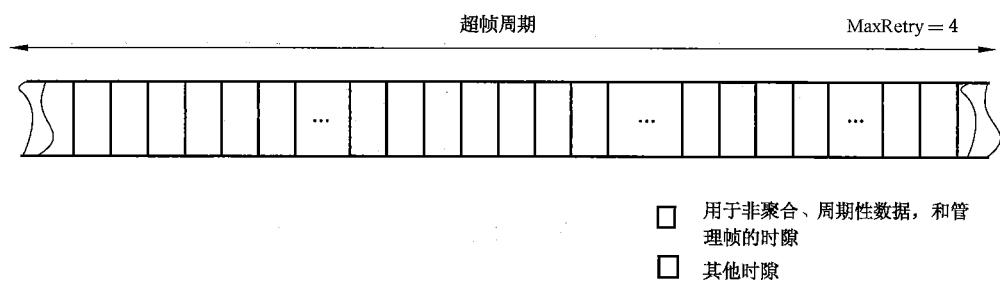


图 32 周期性非聚合数据多次单播重传

#### 8.1.7.4 多次广播重传

在支持帧聚合/解聚功能的 WIA-FA 网络中,当网关设备向现场设备发送聚合帧时,根据 WIA-FA 网络的最大重传次数(见 6.7.1.2.1MaxRetry),多次广播该聚合帧,直至达到 MaxRetry 次数,如图 33 所示。

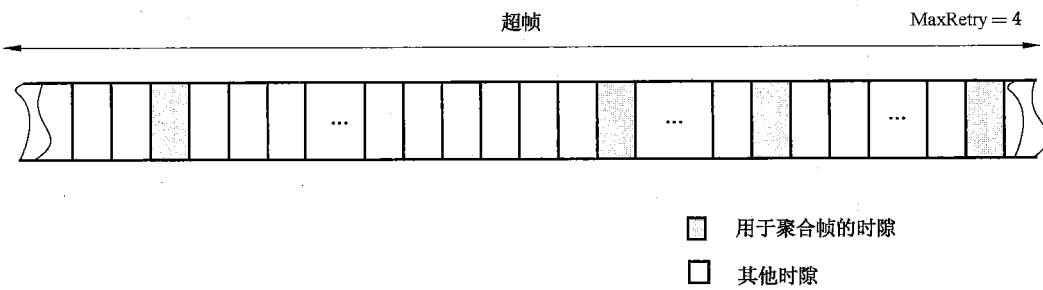


图 33 聚合帧多次广播重传

#### 8.1.7.5 基于 GACK 的时隙退避重传方式

现场设备向网关设备发送非周期性数据帧或管理帧时,例如,远程读属性、远程配置属性、双向时间同步,现场设备根据网关设备广播的 GACK 帧(见 8.4.5)采用基于时隙退避的重传方式。

多个现场设备的非周期性数据帧或管理帧发送完成后,网关设备统计接收到相关帧的现场设备短地址,构造 GACK 帧(见 8.4.5),随后网关设备通过接入设备重复广播 GACK 帧,GACK 帧的广播方式和次数同 NACK(见 8.1.7.2)。如果现场设备未收到 GACK 或接收到的 GACK 帧中未包含自身短地址,则采用基于时隙退避的方法竞争重传数据。

每个超帧周期内设有重传时隙(6.7.1.2.2LinkList),用于现场设备重传非周期性数据帧或管理帧,如图 34 所示。现场设备利用重传时隙以退避方式竞争发送非周期性数据帧或管理帧,在 GACK 广播时隙部分将链路置成接收链路并接收 GACK。如果现场设备未收到 GACK 或者 GACK 中未包含自身短地址,则竞争下一部分重传时隙,重传发送非周期性数据帧或管理帧,直至传输成功或者达到重传上限 MaxRetry(见 6.7.1.2.1),如图 34 所示。

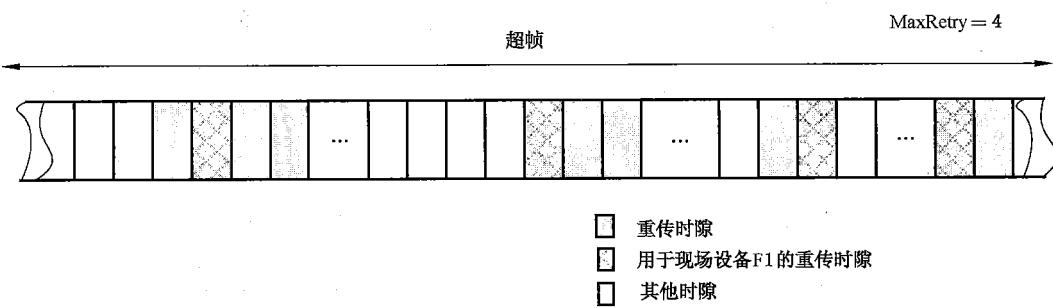


图 34 基于 GACK 的时隙退避重传方式示例

## 8.2 数据链路层数据服务

### 8.2.1 概述

数据链路层数据实体服务访问点(DLDE-SAP)支持接入设备与现场设备之间数据的点到点的传输。数据链路层数据服务原语包括数据请求(DLDE-DATA.request)、数据证实(DLDE-DATA.confirm)和数据指示(DLDE-DATA.indication)原语。

### 8.2.2 数据发送请求原语

应用子层调用 DLDE-DATA.request 原语发送数据。

DLDE-DATA.request 原语的语义如下:

DLDE-DATA.request(

DstAddr,  
VCR\_ID,  
DataType,  
Priority,  
PayloadLength,  
Payload  
)

DLDE-DATA.request 原语的参数说明如表 35 所示。

表 35 DLDE-DATA.request 原语参数

参数名称	数据类型	取值范围	描述
DstAddr	Unsigned16	0~65 535	目的设备的短地址
VCR_ID	Unsigned16	0~65 535	数据对应的 VCR 标识符,仅在 DataType = 0 时有效
DataType	Unsigned8	0~255	表示数据的类型: 0 = 数据帧; 1 = NACK; 2 = GACK; 其余保留
Priority	Unsigned8	0~255	载荷的优先级: 0 = RT0; 1 = RT1; 2 = RT2; 3 = RT3; 4 = NRT; 其余保留
PayloadLength	Unsigned16	0~65 535	表示 Payload 的长度(以八位位组 octet 为单位)
Payload	Octetstring	—	载荷内容

### 8.2.3 数据发送指示原语

DLDE-DATA.indication 原语用于向应用子层报告收到数据。

DLDE-DATA.indication(

SrcAddr,  
DataType,  
PayloadLength,  
Payload  
)

DLDE-DATA.indication 原语的参数说明如表 36 所示。

表 36 DLDE-DATA.indication 原语参数

参数名称	数据类型	取值范围	描述
SrcAddr	Unsigned16	0~65 535	发送设备的短地址
DataType	Unsigned8	0~255	表示数据的类型： 0 = 数据帧； 1 = NACK； 2 = GACK； 其余保留
PayloadLength	Unsigned16	0~65 535	表示载荷的长度(以八位位组为单位)
Payload	Octetstring	—	载荷内容

#### 8.2.4 数据服务时序

图 35、图 36 和图 37 为数据帧的发送、接收和确认的基本流程。

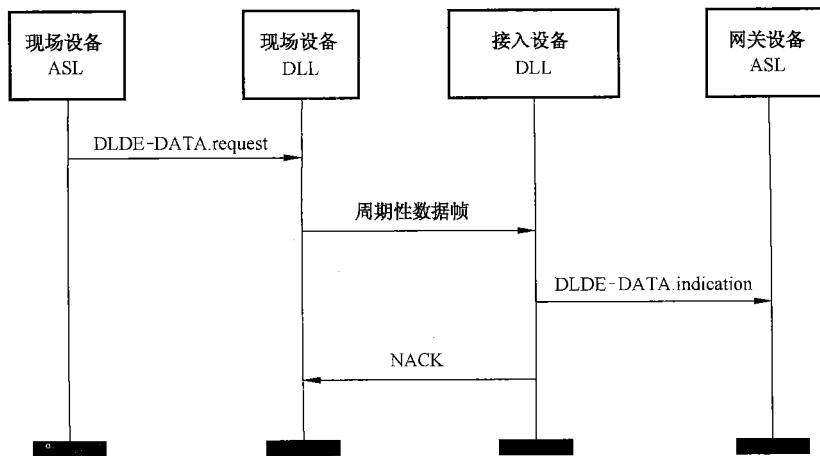


图 35 FD 到 GW 周期性数据服务时序

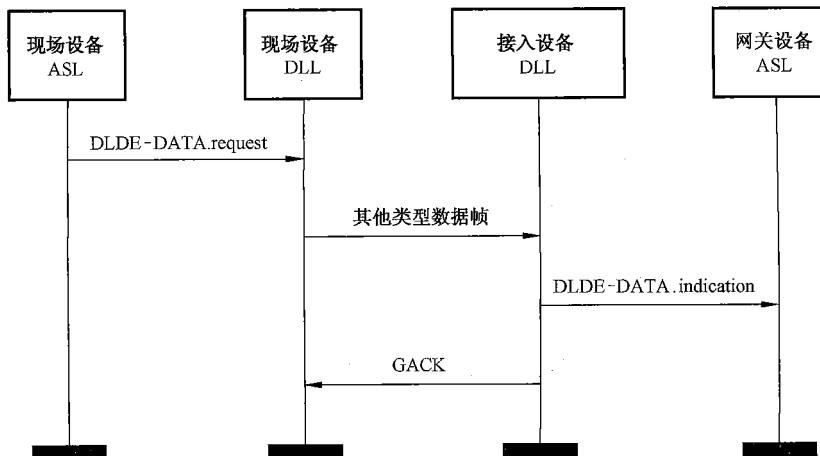


图 36 FD 到 GW 其他类型数据服务时序

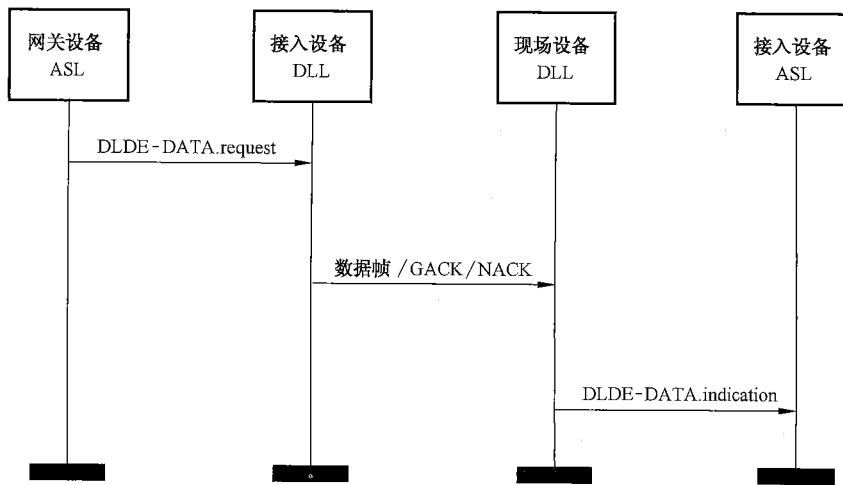


图 37 GW 到 FD 的数据服务时序

### 8.3 数据链路层管理服务

#### 8.3.1 概述

数据链路层管理实体服务访问点(DLME-SAP)支持上层和数据链路层管理实体之间管理命令的传输。表 37 列出了所有管理服务。

所有的请求服务用语 DMAP 调用 DLL 生成请求命令帧;指示服务用语 DLL 向 DMAP 报告收到一个命令帧;响应服务用于 DMAP 调用 DLL 生成响应命令帧;证实服务用语 DLL 向 DMAP 返回请求命令帧发送的结果。

表 37 管理服务原语

原语名称	请求	指示	响应	证实	用途
网络发现	8.3.2.1	—	—	8.3.2.2	用于现场设备发现 WIA-FA 网络
时间同步	8.3.3.1	8.3.3.2	8.3.3.3	8.3.3.4	用于现场设备与网关设备之间实现双向时间同步
设备加入	8.3.4.1	8.3.4.2	8.3.4.3	8.3.4.4	用于现场设备加入 WIA-FA 网络
设备状态报告	8.3.5.1	8.3.5.2	—	8.3.5.3	用于现场设备周期性汇报设备状态
信道状况报告	8.3.6.1	8.3.6.2	—	8.3.6.3	用于现场设备周期性汇报信道状况
远程读属性	8.3.7.1	8.3.7.2	8.3.7.3	8.3.7.4	用于网关设备远程读现场设备的 MIB 属性
远程配置属性	8.3.8.1	8.3.8.2	8.3.8.3	8.3.8.4	用于网关设备远程配置(增加,删除和更新)现场设备的 MIB 属性
设备离开	8.3.9.1	8.3.9.2	8.3.9.3	8.3.9.4	用于网关设备要求现场设备离开网络

#### 8.3.2 网络发现原语

##### 8.3.2.1 网络发现请求原语

现场设备加入网络前,其 DMAP 调用数据链路层的网络发现请求原语 DLME-DISCOVERY.request,用于请求执行信道扫描。

DLME- DISCOVERY.request(

    ScanChannels,

)

DLME- DISCOVERY.request 原语的参数说明如表 38 所示。

表 38 DLME- DISCOVERY.request 原语参数

参数名称	数据类型	取值范围	描述
ScanChannels	BitField24	BitMap(见 6.7.1.2.1, 表 15)	IEEE STD 802.11-2012 物理层的可用信道

### 8.3.2.2 网络发现证实原语

网络发现证实原语 DLME- DISCOVERY.confirm 用于响应 DLME- DISCOVERY.request 原语。

DLME-DISCOVERY.confirm(

    Status,

    BeaconCount,

    SuperframeLength,

    TimeslotDuration,

    FirstSharedTimeslotNum,

    SharedTimeslotCount,

    AbsoluteTimeValue,

    BeaconDescription

)

DLME-DISCOVERY.confirm 原语的参数说明如表 39 所示。

表 39 DLME-DISCOVERY.confirm 原语参数

参数名称	数据类型	取值范围	描述
Status	Unsigned8	0~255	扫描信道的结果： 0 = SUCCESS; 1 = NO_BEACON; 其余保留
BeaconCount	Unsigned8	0~255	接收到的 Beacon 的数量
SuperframeLength	Unsigned16	0~65 535	表示初始超帧的长度,以时隙为单位 (见 8.1.2)
TimeslotDuration	Unsigned16	0~65 535	表示初始时隙长度(6.7.1.2.1 TimeSlot Duration)
FirstSharedTimeslotNumber	Unsigned16	0~65 535	共享时隙在超帧中的开始位置
SharedTimeslotCount	Unsigned8	0~255	共享时隙的总数(格式见图 55)
AbsoluteTimeValue	TimeData	0~(2 <sup>64</sup> - 1)	表示发送信标帧的绝对时间值(见 6.7.1.2.1 TimeValue)
BeaconDescription	BeaconDescription_Struct 结构体列表	—	扫描得到的 Beacon 的信息,见表 40

表 40 BeaconDescription\_Struct 参数

参数名称	数据类型	取值范围	描述
信道索引 (ChannelIndex)	Unsigned24	0~(2 <sup>24</sup> -1)	表示接收到信标帧的信道索引(见 6.7.1.2.1, 表 15)
信标帧相对时隙号 (BeaconRelativeTimeslotNum)	Unsigned16	0~65 535	表示发送该信标帧时的相对时隙号
能量等级 (ED)	Unsigned8	0~255	接收到 Beacon 的能量等级

### 8.3.2.3 网络发现时序

网络发现时序如图 38 所示。

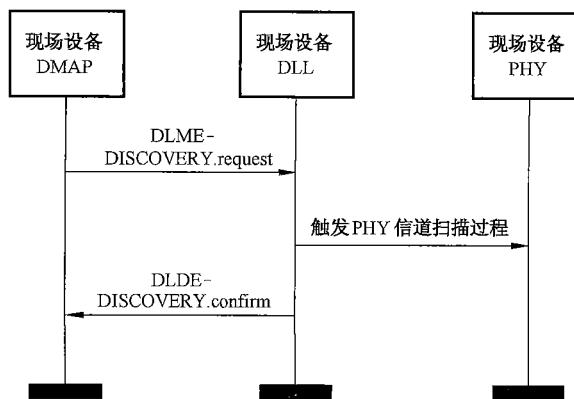


图 38 网络发现时序图

发送端 DMAP 调用 DLL 的 DLME-SCAN.request 原语, 指示物理层开始信道扫描(见 IEEE STD 802.11-2012, 10.1.4.3.3); 信道扫描结束后, DLL 层利用 DLDE-SCAN.confirm 向 DMAP 汇报信道扫描结果。

### 8.3.3 时间同步原语

#### 8.3.3.1 时间同步请求原语

时间同步请求原语用于现场设备 DMAP 请求 DLL 发送双向时间同步请求命令帧。

DLME-TIME-SYN.request(  
)

DLME-TIME-SYN.request 原语的参数为空, 指示物理层将绝对时间值写入双向时间同步请求命令帧中。

#### 8.3.3.2 时间同步指示原语

时间同步指示原语用于向接入设备的 DMAP 报告接收到时间同步请求命令帧。

DLME-TIME-SYN.indication(  
SrcAddr,  
FieldDeviceTimeValue

)

DLME-TIME-SYN.indication 原语的参数说明如表 41 所示。

表 41 DLME-TIME-SYN.indication 原语参数

参数名称	数据类型	取值范围	描述
SrcAddr	Unsigned16	0~65 535	现场设备短地址(见 6.7.1.2.1, DeviceShortAddress 属性)
FieldDeviceTimeValue	TimeData	0~(2 <sup>64</sup> -1)	现场设备发送时间同步请求命令帧时的时戳值(以 ms 为单位), 见 8.4.13

### 8.3.3.3 时间同步响应原语

时间同步响应原语用于对 DLME-TIME-SYN.indication 原语进行响应。

DLME-TIME-SYN.response(

DstAddr,  
FieldDeviceTimeValue,  
ReceiveTimeValue

)

DLME-TIME-SYN.response 原语的参数说明如表 42 所示。

表 42 DLME-TIME-SYN.response 原语参数

参数名称	数据类型	取值范围	描述
DstAddr	Unsigned16	0~65 535	现场设备地址(见 6.7.1.2.1, DeviceShortAddress 属性)
FieldDeviceTimeValue	TimeData	0~(2 <sup>64</sup> -1)	现场设备发送双向时间同步请求命令帧时的时戳值(以 μs 为单位), 见 8.4.13
ReceiveTimeValue	TimeData	0~(2 <sup>64</sup> -1)	接入设备接收到双向时间同步请求命令帧时刻的时间值(以 ms 为单位), 见 8.4.14

### 8.3.3.4 时间同步证实原语

时间同步证实原语用于返回双向时间同步响应命令帧(见 8.4.14)内的现场设备发送时刻的时间值以及接入设备接收到双向时间同步请求帧的时间值。

DLME-TIME-SYN.confirm(

Status,  
FieldDeviceTimeValue,  
ReceiveTimeValue

)

DLME-TIME-SYN.confirm 原语的参数说明如表 43 所示。

表 43 DLME-TIME-SYN.confirm 原语参数

参数名称	数据类型	取值范围	描述
Status	Unsigned8	0~255	表示双向时间同步请求的结果： 0 = 成功 SUCCESS； 1 = 超时 OVERTIME(当超过 TwoWayOverTime(见 6.7.1.2.1)后，则认为此次双向时间同步失败)； 其余保留
FieldDeviceTimeValue	TimeData	0~(2 <sup>64</sup> - 1)	现场设备发送双向时间同步请求命令帧时的时戳值(以 ms 为单位), 见 8.4.13
ReceiveTimeValue	TimeData	0~(2 <sup>64</sup> - 1)	接入设备接收到双向时间同步请求命令帧时刻时间值(以 ms 为单位), 见 8.4.14

### 8.3.3.5 时间同步时序

时间同步时序如图 39 所示。

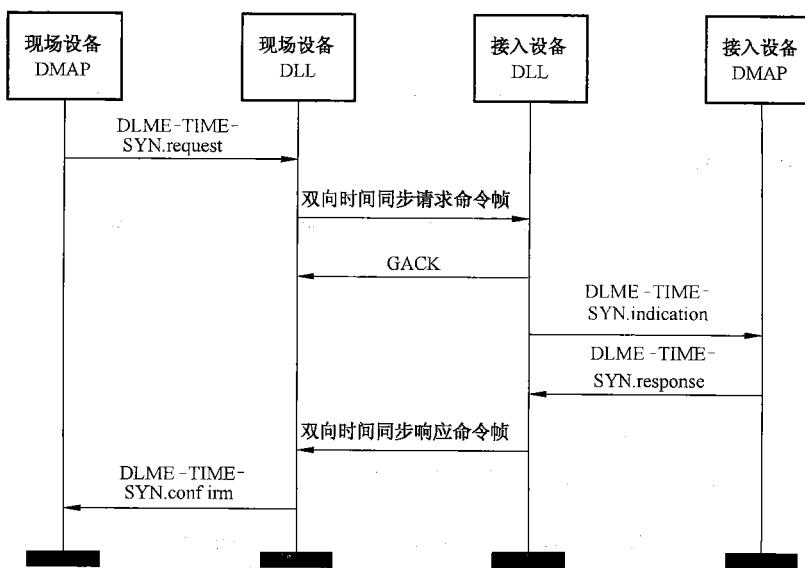


图 39 双向时间同步的时序图

现场设备 DMAP 调用 DLL 的 DLME-TIME-SYN.request 原语, 指示 DLL 发送双向时间同步请求命令帧; 接入设备 DLL 层接收到双向时间同步请求命令帧后, 返回 GACK, 并利用 DLME-TIME-SYN.indication 向网关设备 DMAP 汇报; 网关设备 DMAP 调用 DLME-TIME-SYN.response 原语, 指示接入设备 DLL 发送双向时间同步响应命令帧; 现场设备 DLL 接收到双向时间同步响应命令帧后, 并利用 DLME-TIME-SYN.confirm 向 DMAP 指示接收到双向时间同步响应命令帧。

### 8.3.4 设备加入原语

#### 8.3.4.1 设备加入请求原语

设备加入请求原语 DLME-JOIN.request 用于待加入网络的现场设备的 DMAP 请求 DLL 生成加

入请求命令帧。

```
DLME-JOIN.request(
    NetworkID,
    Channel,
    PhyAddr,
    SecMaterial
)
```

DLME-JOIN.request 原语的参数说明如表 44 所示。

表 44 DLME-JOIN.request 原语参数

参数名称	数据类型	取值范围	描述
NetworkID	Unsigned8	0~255	网络的 ID 号, 用于多个网络共存的情况下标识网络(见 6.7.1.2.1 表 15 属性)
Channel	BitField24	BitMap(见 6.7.1.2.1, 表 15)	发送加入请求的信道, 从物理层所支持的可用信道中选择
PhyAddr	Unsigned64	0~( $2^{64}$ - 1)	待加入设备的长地址(见 6.7.1.2.1, LongAddress 属性)
SecMaterial	Unsigned64	0~( $2^{64}$ - 1)	设备安全入网认证信息, 见 11.3。如果 SecMaterial = 0, 此参数无效

#### 8.3.4.2 设备加入指示原语

设备加入指示原语 DLME-JOIN.indication 用于向网关设备的 DMAP 指示已成功接收到一个现场设备的加入请求。

```
DLME-JOIN.indication(
    PhyAddr,
    SecMaterial
)
```

DLME-JOIN.indication 原语的参数说明如表 45 所示。

表 45 DLME-JOIN.indication 原语参数

参数名称	数据类型	取值范围	描述
PhyAddr	Unsigned64	0~( $2^{64}$ - 1)	待加入设备的长地址(见 6.7.1.2.1, LongAddress 属性)
SecMaterial	Unsigned64	0~( $2^{64}$ - 1)	设备安全入网认证信息, 见 11.3。如果 SecMaterial = 0, 此参数无效

#### 8.3.4.3 设备加入响应原语

设备加入响应原语 DLME-JOIN.response 是对加入请求命令的响应。

```
DLME-JOIN.response(
    Status,
    ShortAddr
)
```

)  
DLME-JOIN.response 原语的参数说明如表 46 所示。

表 46 DLME-JOIN.response 原语参数

参数名称	数据类型	取值范围	描述
Status	Unsigned8	0~255	表示加入请求的结果： 0 = SUCCESS; 1 = 网络 ID 不匹配; 2 = 认证失败; 3 = 网络规模超限; 其余保留
ShortAddr	Unsigned16	0~65 535	GW 分配给待加入设备的短地址(见 6.7.1.2.1, DeviceShortAddress 属性)。 Status 为 SUCCESS 时,该域有效

#### 8.3.4.4 设备加入证实原语

设备加入证实原语 DLME-JOIN.confirm 是对加入请求的响应。

DLME-JOIN.confirm(

Status,

ShortAddr

)

DLME-JOIN.confirm 原语的参数说明如表 47 所示。

表 47 DLME-JOIN.confirm 原语参数

参数名称	数据类型	取值范围	描述
Status	Unsigned8	0~255	表示加入请求的结果： 0 = SUCCESS; 1 = 网络 ID 不匹配; 2 = 认证失败; 3 = 网络规模超限; 其余保留
ShortAddr	Unsigned16	0~65 535	GW 分配给待加入设备的短地址(见 6.7.1.2.1, DeviceShortAddress 属性)。 Status 为 SUCCESS 时,该域有效

#### 8.3.4.5 设备加入时序

设备加入时序如图 40 所示。

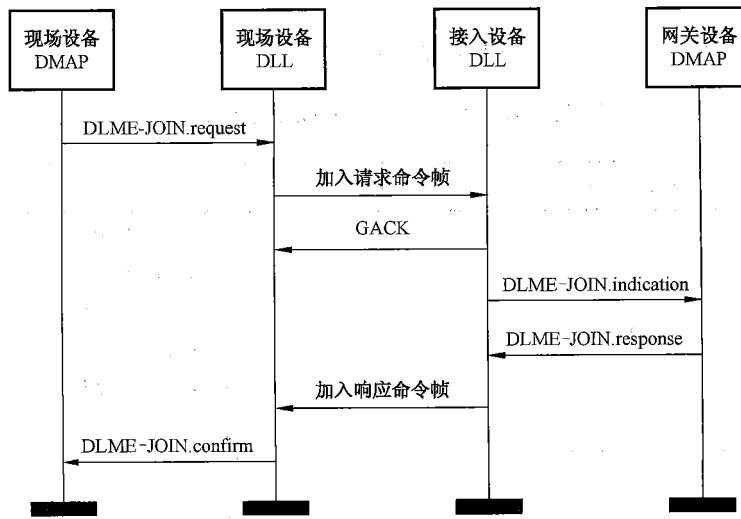


图 40 设备加入的时序图

现场设备 DMAP 调用 DLL 的 DLME-JOIN.request 原语, 指示 DLL 发送加入请求命令帧; 接入设备 DLL 层接收到加入请求命令帧后, 返回 GACK, 并利用 DLME-JOIN.indication 向网关设备 DMAP 汇报; 网关设备 DMAP 调用 DLME-JOIN.response 原语, 指示接入设备 DLL 发送加入响应命令帧; 现场设备 DLL 接收到加入响应命令帧后, 利用 DLME-JOIN.confirm 向 DMAP 指示接收到加入响应命令帧。

### 8.3.5 设备状态报告原语

#### 8.3.5.1 设备状态报告请求原语

设备状态报告请求原语用于现场设备向网关设备周期性汇报设备状态信息。

```

DLME-DEVICE-STATUS.request(
    PowerSupplyStatus
)

```

DLME-DEVICE-STATUS.request 原语的参数说明如表 48 所示。

表 48 DLME-DEVICE-STATUS.request 原语参数

参数名称	数据类型	取值范围	描述
PowerSupplyStatus	Unsigned8	0~255	设备的电量信息 (PowerSupplyStatus 见表 20 Device_Struct 结构体定义)

#### 8.3.5.2 设备状态报告指示原语

设备状态报告指示原语用于向 DMAP 指示接收到设备状态报告请求命令帧。

```

DLME-DEVICE-STATUS.indication(
    ShortAddr,
    PowerSupplyStatus
)

```

DLME-DEVICE-STATUS.indication 原语的参数说明如表 49 所示。

表 49 DLME-DEVICE-STATUS.indication 原语参数

参数名称	数据类型	取值范围	描述
ShortAddr	Unsigned16	0~65 535	现场设备的短地址(见 6.7.1.2.1, DeviceShortAddress 属性)
PowerSupplyStatus	Unsigned8	0~255	设备的电量信息(PowerSupplyStatus 见表 20 Device_Struct 结构体定义)

### 8.3.5.3 设备状态报告证实原语

设备状态报告证实原语用于返回设备状态报告请求的结果。

DLME-DEVICE-STATUS.confirm(

Status

)

DLME-DEVICE -STATUS.confirm 原语的参数说明如表 50 所示。

表 50 DLME-DEVICE-STATUS.confirm 原语参数

参数名称	数据类型	取值范围	描述
Status	Unsigned8	0~255	表示设备状态报告请求的结果： 0 = SUCCESS; 1 = FAILURE; 其余保留

### 8.3.5.4 设备状态报告时序

设备状态报告时序如图 41 所示。

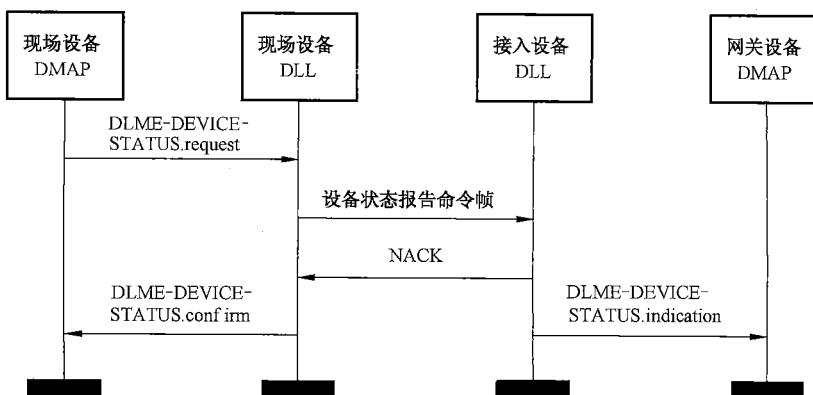


图 41 现场设备设备状态报告时序图

现场设备 DMAP 调用 DLL 的 DLME-DEVICE-STATUS.request 原语, 指示 DLL 发送设备状态报告命令帧; 接入设备 DLL 层接收到设备状态报告命令帧后, 返回 NACK, 并利用 DLME-DEVICE-STATUS.indication 向网关设备 DMAP 汇报; 现场设备 DLL 接收到接入设备的 NACK 后, 利用 DLME-DEVICE-STATUS.confirm 向 DMAP 指示成功发送设备状态报告命令帧。

### 8.3.6 信道状况报告原语

#### 8.3.6.1 信道状况报告请求原语

信道状况报告请求原语用于现场设备向网关设备汇报信道状况。

DLME-CHANNEL-CONDITION.request(

```
    Count,
    ChannelConditionInfo
)
```

DLME-CHANNEL-CONDITION.request 原语的参数说明如表 51 所示。

表 51 DLME-CHANNEL-CONDITION.request 原语参数

参数名称	数据类型	取值范围	描述
Count	Unsigned8	0~255	汇报的信道的数量
ChannelConditionInfo	ChanCon_Struct 结构体(见表 19)	—	信道状况属性的信息

#### 8.3.6.2 信道状况报告指示原语

信道状况报告指示原语用于向 DMAP 报告接收到信道状况报告请求命令帧。

DLME-CHANNEL-CONDITION.indication(

```
    SrcAddr,
    Count,
    ChannelConditionInfo
)
```

DLME-CHANNEL-CONDITION.indication 原语的参数说明如表 52 所示。

表 52 DLME-CHANNEL-CONDITION.indication 原语参数

参数名称	数据类型	取值范围	描述
SrcAddr	Unsigned16	0~65 535	现场设备短地址(见 6.7.1.2.1,DeviceShortAddress 属性)
Count	Unsigned8	0~255	汇报的信道的数量
ChannelConditionInfo	ChanCon_Struct 结构体(见表 19)	—	信道状况属性的信息

#### 8.3.6.3 信道状况报告证实原语

信道状况报告证实原语用于返回信道状况报告请求的结果。

DLME-CHANNEL-CONDITION.confirm(

```
    Status
)
```

DLME-CHANNEL-CONDITION.confirm 原语的参数说明如表 53 所示。

表 53 DLME-CHANNEL-CONDITION.confirm 原语参数

参数名称	数据类型	取值范围	描述
Status	Unsigned8	0~255	表示发送信道状况报告请求命令帧的结果： 0 = SUCCESS; 1 = FAILURE; 其余保留

#### 8.3.6.4 信道状况报告时序

信道状况报告时序如图 42 所示。

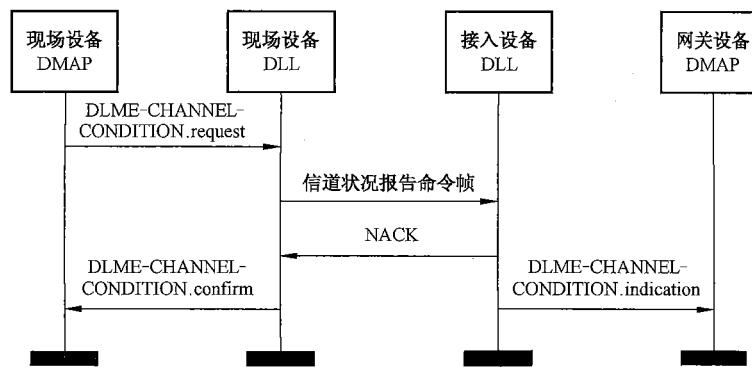


图 42 现场设备信道状态报告时序图

现场设备 DMAP 调用 DLL 的 DLME-CHANNEL-CONDITION.request 原语, 指示 DLL 发送信道状况报告命令帧; 接入设备 DLL 层接收到信道状况报告命令帧后, 返回 NACK, 并利用 DLME-CHANNEL-CONDITION.indication 向网关设备 DMAP 汇报; 现场设备 DLL 接收到接入设备的 NACK 后, 利用 DLME-CHANNEL-CONDITION.confirm 向 DMAP 指示成功发送信道状况报告命令帧。

#### 8.3.7 远程读属性原语

##### 8.3.7.1 远程读属性请求原语

远程读属性请求原语用于远程读取设备管理信息中的属性值。

DLME-INFO-GET.request(

Handle,  
DstAddr,  
AttributeID,  
MemberID,  
FirstStoreIndex,  
Count  
)

DLME-INFO-GET.request 原语的参数说明如表 54 所示。

表 54 DLME-INFO-GET.request 原语参数

参数名称	数据类型	取值范围	描述
Handle	Unsigned8	0~255	调用远程读配置请求原语时分配的句柄
DstAddr	Unsigned16	0~65 535	远程读属性请求的目的端 8 位或 16 位短地址
AttributeID	Unsigned8	0~255	信息库中属性的标识符
MemberID	Unsigned8	0~255	结构化属性成员标识符,用于获取 MIB 中的结构化属性。如果该值为 255,则表示获取全部的属性成员。该值对非结构化属性无效
FirstStoreIndex	Unsigned16	0~65 535	所获取结构化属性的第一个值的存储索引,FirstStoreIndex 对非结构化属性无效
Count	Unsigned16	0~65 535	记录的个数,用来表示需要读取记录的数量。如果 Count = 0,则给出从 FirstStoreIndex 开始的所有结构化记录的属性值

### 8.3.7.2 远程读属性指示原语

远程读属性指示原语用于通知 DMAP 成功接收到属性获取请求命令包。

DLME-INFO-GET.indication(

```
SrcAddr,  
AttributeID,  
MemberID,  
FirstStoreIndex,  
Count
```

)

DLME-INFO-GET.indication 原语的参数说明如表 55 所示。

表 55 DLME-INFO-GET.indication 原语参数

参数名称	数据类型	取值范围	描述
SrcAddr	Unsigned16	0~65 535	远程读属性请求的源端 8 位或 16 位短地址
AttributeID	Unsigned8	0~255	信息库中属性的标识符
MemberID	Unsigned8	0~255	结构化属性成员标识符,用于获取 MIB 中的结构化属性。如果该值为 255,则表示获取全部的属性成员。MemberID 对非结构化属性无效
FirstStoreIndex	Unsigned16	0~65 535	所获取结构化属性的第一个值的存储索引,FirstStoreIndex 对非结构化属性无效
Count	Unsigned16	0~65 535	记录的个数,用来表示需要读取结构化记录的数量。如果 Count = 0,则表示要获取从 FirstStoreIndex 开始的所有记录的属性值

### 8.3.7.3 远程读属性响应原语

远程读属性响应原语用于对属性获取请求原语进行响应。

DLME-INFO-GET.response(

```
DstAddr,
Status,
AttributeID,
MemberID,
FirstStoreIndex,
Count,
AttributeValue
```

)

DLME-INFO-GET.response 原语的参数说明如表 56 所示。

**表 56 DLME-INFO-GET.response 原语参数**

参数名称	数据类型	取值范围	描述
DstAddr	Unsigned16	0~65 535	远程读属性响应的目的端 8 位或 16 位短地址
Status	Unsigned8	0~255	表示属性获取请求的结果： 0 = SUCCESS; 1 = UNSUPPORTED_ATTRIBUTE; 其余保留
AttributeID	Unsigned8	0~255	信息库中属性的标识符
MemberID	Unsigned8	0~255	结构化属性成员标识符, 用于获取 MIB 中的结构化属性。如果该值为 255, 则表示获取全部的属性成员。MemberID 对非结构化属性无效
FirstStoreIndex	Unsigned16	0~65 535	所获取结构化属性的第一个值的存储索引, FirstStoreIndex 对非结构化属性无效
Count	Unsigned16	0~65 535	记录的个数, 用来表示返回记录的数量。如果 Count = 0, 则表示返回从 FirstStoreIndex 开始的所有记录的属性值
AttributeValue	Octectstring	—	需要读取的属性的值

如果获取管理信息库中的属性值成功, 则返回 "SUCCESS", AttributeValue 参数有效; 如果管理信息库中没有读取的属性, 则返回 "UNSUPPORTED\_ATTRIBUTE", AttributeValue 参数无效。

### 8.3.7.4 远程读属性证实原语

远程读属性证实原语用于返回属性获取请求原语的结果。

DLME-INFO-GET.confirm(

```
Handle,
SrcAddr
Status,
AttributeID,
MemberID,
```

```

        FirstStoreIndex,
        Count,
        AttributeValue
    )

```

DLME-INFO-GET.confirm 原语的参数说明如表 57 所示。

表 57 DLME-INFO-GET.confirm 原语参数

参数名称	数据类型	取值范围	描述
Handle	Unsigned8	0~255	调用远程读配置请求原语时分配的句柄
SrcAddr	Unsigned16	0~65 535	远程读属性响应的源端 8 位或 16 位短地址
Status	Unsigned8	0~255	表示远程读属性请求的结果： 0 = SUCCESS; 1 = UNSUPPORTED_ATTRIBUTE; 其余保留
AttributeID	Unsigned8	0~255	信息库中属性的标识符
MemberID	Unsigned8	0~255	结构化属性成员标识符, 用于获取 MIB 中的结构化属性。如果该值为 255, 则表示获取全部的属性成员。MemberID 对非结构化属性无效
FirstStoreIndex	Unsigned16	0~65 535	所读取结构化属性的第一个值的存储索引, FirstStoreIndex 对非结构化属性无效
Count	Unsigned16	0~65 535	记录的个数, 用来表示返回记录性的数量。如果 Count = 0, 则表示返回了从 FirstStoreIndex 开始的所有记录的值
AttributeValue	Octectstring	—	需要读取的属性的值

### 8.3.7.5 远程读属性时序

远程读属性时序如图 43 所示。

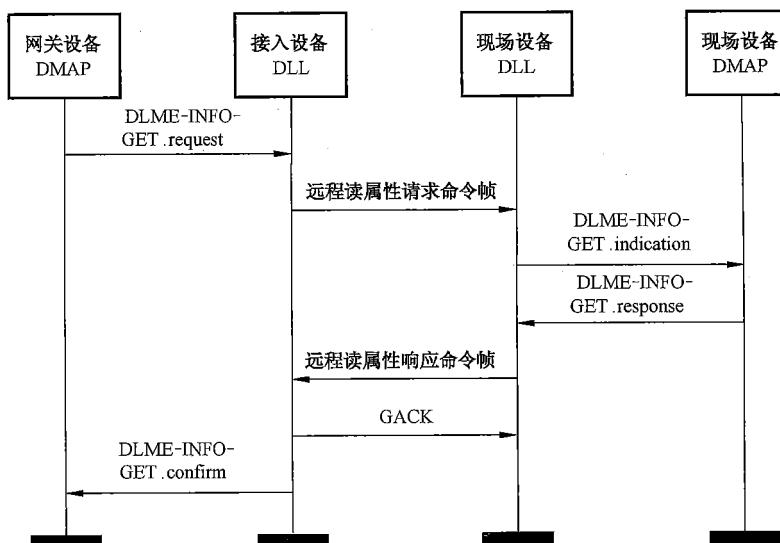


图 43 远程读属性时序图

网关设备 DMAP 调用接入设备 DLL 的 DLME-INFO-GET.request 原语,指示接入设备 DLL 发送远程读属性请求命令帧;现场设备 DLL 层接收到远程读属性请求命令帧后,并利用 DLME-INFO-GET.indication 向 DMAP 汇报;现场设备的 DMAP 调用 DLME-INFO-GET.response 原语,指示 DLL 发送远程读属性响应命令帧;接入设备 DLL 接收到远程读属性响应命令帧后,返回 GACK,并利用 DLME-INFO-GET.confirm 向网关设备 DMAP 指示接收到远程读属性响应命令帧。

### 8.3.8 远程配置属性原语

#### 8.3.8.1 远程配置属性请求原语

远程配置属性请求原语用于远程请求修改设备信息库中的属性。

DLME-INFO-SET.request(

```
    Handle,
    DstAddr,
    AttributeOption,
    AttributeID,
    MemberID,
    FirstStoreIndex,
    Count,
    AttributeValue
)
```

DLME-INFO-SET.request 原语的参数说明如表 58 所示。

表 58 DLME-INFO-SET.request 原语参数

参数名称	数据类型	取值范围	描述
Handle	Unsigned8	0~255	调用远程配置属性请求原语时分配的句柄
DstAddr	Unsigned16	0~65 535	远程配置属性目的端 8 位或 16 位短地址
AttributeOption	Unsigned8	0~255	远程写属性的具体操作: 0 = 增加; 1 = 删除; 2 = 更新
AttributeID	Unsigned8	0~255	信息库中属性的标识符
MemberID	Unsigned8	0~255	结构化属性成员标识符,用于修改 MIB 中的结构化属性。如果该值为 255,则表示写全部的属性成员。MemberID 对非结构化属性无效
FirstStoreIndex	Unsigned16	0~65 535	对结构化属性,FirstStoreIndex 是所配置第一条记录的存储索引;FirstStoreIndex 对非结构化属性无效
Count	Unsigned16	0~65 535	记录的个数,用来表示配置记录的数量。如果 Count = 0,则表示请求配置从 FirstStoreIndex 开始的所有记录的值
AttributeValue	Octetstring	—	需要设置记录的值。AttributeOption = 2 时,该值无效

### 8.3.8.2 远程配置属性指示原语

远程配置属性指示原语用于通知 DMAP 成功接收到一个远程配置属性请求命令包。

DLME-INFO-SET.indication(

```

        SrcAddr,
        AttributeOption,
        AttributeID,
        MemberID,
        FirstStoreIndex,
        Count,
        AttributeValue
    )

```

DLME-INFO-SET.indication 原语的参数说明如表 59 所示。

表 59 DLME-INFO-SET.indication 原语参数

参数名称	数据类型	取值范围	描述
SrcAddr	Unsigned16	0~65 535	远程配置属性的源端 8 位或 16 位短地址
AttributeOption	Unsigned8	0~255	远程写属性的具体操作： 0 = 增加； 1 = 删除； 2 = 更新
AttributeID	Unsigned8	0~255	信息库中属性的标识符
MemberID	Unsigned8	0~255	结构化属性成员标识符, 用于修改 MIB 中的结构化属性。如果该值为 255, 则表示配置全部的属性成员。MemberID 对非结构化属性无效
FirstStoreIndex	Unsigned16	0~65 535	所配置结构化属性的第一个值的存储索引, FirstStoreIndex 对非结构化属性无效
Count	Unsigned16	0~65 535	记录的个数, 用来表示配置记录的数量。如果 Count = 0, 则表示请求配置从 FirstStoreIndex 开始的所有记录的值
AttributeValue	Octetstring	—	需要设置属性的值。AttributeOption = 1 时, 该值无效

### 8.3.8.3 远程配置属性响应原语

远程配置属性响应原语用于对远程配置属性请求原语进行响应。

DLME-INFO-SET.response(

```

        SrcAddr,
        AttributeOption,
        AttributeID,
        MemberID,
        FirstStoreIndex,
        Count,
    )

```

Status

)

DLME-INFO-SET.response 原语的参数说明如表 60 所示。

表 60 DLME-INFO-SET.response 原语参数

参数名称	数据类型	取值范围	描述
DstAddr	Unsigned16	0~65 535	远程配置属性响应的目的端 8 位或 16 位短地址
AttributeOption	Unsigned8	0~255	远程写属性的具体操作： 0 = 增加； 1 = 删除； 2 = 更新
AttributeID	Unsigned8	0~255	信息库中属性的标识符
MemberID	Unsigned8	0~255	结构化属性成员标识符，用于修改 MIB 中的结构化属性。如果该值为 255，则表示配置全部的属性成员。MemberID 对非结构化属性无效
FirstStoreIndex	Unsigned16	0~65 535	所配置结构化属性的第一个值的存储索引，FirstStoreIndex 对非结构化属性无效
Count	Unsigned16	0~65 535	记录的个数，用来表示配置记录的数量。如果 Count = 0，则表示请求配置从 FirstStoreIndex 开始的所有记录的值
Status	Unsigned8	0~255	表示远程配置属性请求的结果： 0 = SUCCESS； 1 = UNSUPPORTED_ATTRIBUTE； 2 = INVALID_PARAMETER； 其余保留

如果配置属性成功，则返回“SUCCESS”；如果所配置属性的标识符不在定义范围内，则返回“UNSUPPORTED\_ATTRIBUTE”；如果配置的记录数不等于 MIB 库里的记录条数，则返回“INVALID\_PARAMETER”。

#### 8.3.8.4 远程配置属性证实原语

远程配置属性证实原语用于返回远程配置属性请求原语的结果。

DLME-INFO-SET.confirm (

Handle,

Status

)

DLME-INFO-SET.confirm 原语的参数说明如表 61 所示。

表 61 DLME-INFO-SET.confirm 原语参数

参数名称	数据类型	取值范围	描述
Handle	Unsigned8	0~255	调用远程配置属性请求原语时分配的句柄
Status	Unsigned8	0~255	表示远程配置属性请求的结果： 0 = SUCCESS; 1 = UNSUPPORTED_ATTRIBUTE; 2 = INVALID_PARAMETER; 其余保留

### 8.3.8.5 远程配置属性时序

远程配置属性时序如图 44 所示。

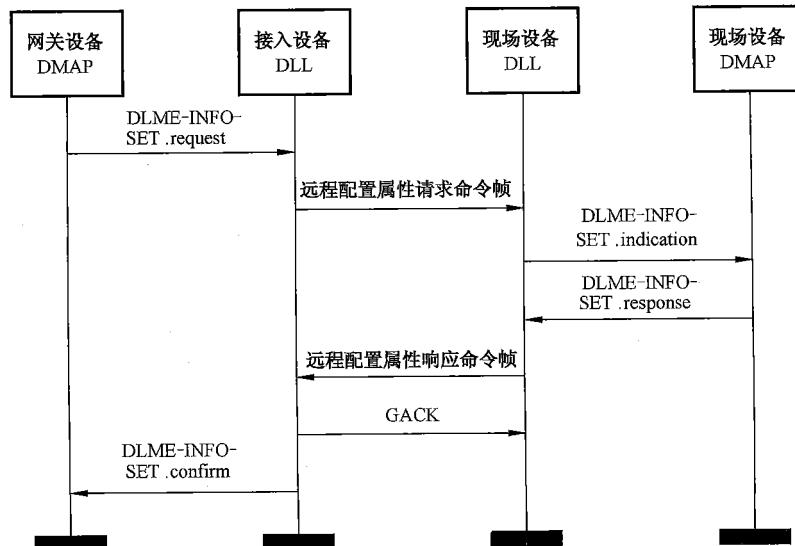


图 44 远程配置属性时序图

网关设备 DMAP 调用接入设备 DLL 的 DLME-INFO-SET.request 原语,指示接入设备 DLL 发送远程配置属性请求命令帧;现场设备 DLL 层接收到远程配置属性请求命令帧后,并利用 DLME-INFO-SET.indication 向 DMAP 汇报;现场设备的 DMAP 调用 DLME-INFO-SET.response 原语,指示 DLL 发送远程配置属性响应命令帧;接入设备 DLL 接收到远程配置属性响应命令帧后,返回 GACK,并利用 DLME-INFO-SET.confirm 向网关设备 DMAP 指示接收到远程配置属性响应命令帧。

### 8.3.9 设备离开原语

#### 8.3.9.1 设备离开请求原语

设备离开请求原语 DLME-LEAVE.request 用于网关设备要求现场设备离开网络。

DLME-LEAVE.request(

    ShortAddr

)

DLME-LEAVE.request 原语的参数说明如表 62 所示。

表 62 DLME-LEAVE.request 原语参数

参数名称	数据类型	取值范围	描述
ShortAddr	Unsigned8/ Unsigned16	0~255/ 0~65 535	被要求离开的现场设备的短地址(见 6.7.1.2.1, DeviceShortAddress 属性)

### 8.3.9.2 设备离开指示原语

设备离开指示原语 DLME-LEAVE.indication 用来向现场设备指示已接收到离开请求命令帧。

DLME-LEAVE.indication(  
)

DLME-LEAVE.indication 原语的参数为空。

### 8.3.9.3 设备离开响应原语

设备离开响应原语 DLME-LEAVE.response 用来通知网关设备已收到离开请求命令帧。

DLME-LEAVE.response(  
)

DLME-LEAVE.response 原语的参数为空。

### 8.3.9.4 设备离开证实原语

设备离开证实原语 DLME-LEAVE.confirm 用来返回 DLME-LEAVE.request 原语的执行结果。

DLME-LEAVE.confirm(  
Status  
)

DLME-LEAVE.confirm 原语的参数说明如表 63 所示。

表 63 DLME-LEAVE.confirm 原语参数

参数名称	数据类型	取值范围	描述
Status	Unsigned8	0~255	设备离开请求的结果： 0 = SUCCESS; 1 = FAILURE; 其余保留

### 8.3.9.5 设备离开时序

设备离开时序如图 45 所示。

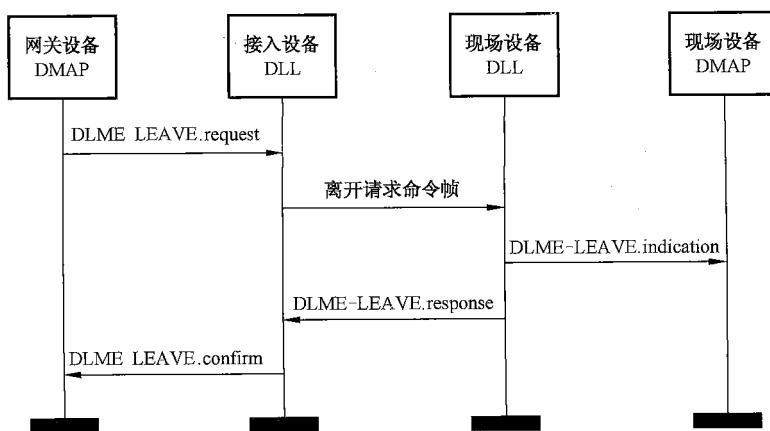


图 45 设备离开的时序图

网关设备 DMAP 调用接入设备 DLL 的 DLME-LEAVE.request 原语, 指示接入设备 DLL 发送离开请求命令帧; 现场设备 DLL 层接收到离开请求命令帧后, 利用 DLME-LEAVE.indication 向 DMAP 汇报; 接入设备利用 DLME-LEAVE.confirm 向网关设备 DMAP 指示成功发送离开请求命令帧。

## 8.4 数据链路层帧格式

### 8.4.1 数据链路层通用帧格式

数据链路层通用帧格式如图 46 所示。

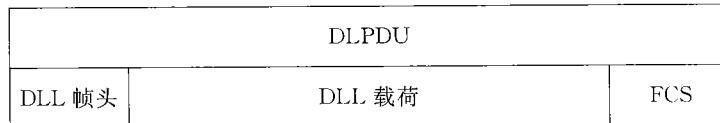


图 46 数据链路层通用帧格式

包括:

- WIA-FA DLL 帧头, 格式如图 47 所示;
- DLL 载荷;
- 帧校验序列(FCS)。

注: 安全部分的内容见第 11 章。

1 八位位组	1 八位位组	1/2/8 八位位组	2 八位位组	0/1 八位位组	0/1 八位位组	2 八位位组
帧控制	网络 ID	源或目的地址	序列号	分段数量	分段序列号	帧长度

图 47 DLL 帧头格式

各域说明如下:

- 帧控制, 格式如图 48 所示;
- 网络 ID: 长度为 1 个八位位组, 用于多个网络共存下区分网络;
- 源或目的地址: 长度为 1/2/8 个八位位组, 1/2 个八位位组表示 8/16 位短地址, 8 个八位位组表示 64 位长地址;
- 序列号: 长度为 2 个八位位组, 表示帧的序列号, 从 1 开始递增 1, 直至最大值后重置为 1;

- 分段数量：长度为 0/1 个八位位组，表示分段的总数；
- 分段序列号：长度为 0/1 个八位位组，用于指示第几个分段；
- 帧长度：长度为 2 个八位位组，用于表示 DLL 载荷的长度。

位:0~4	位:5	位:6	位:7
帧类型	分段标识	抢占标识	地址类型

图 48 帧控制格式

各子域具体说明如下：

- 帧类型：长度为 5 个比特，表示传输帧的类型，包括数据帧和命令帧，编码见表 64；

表 64 帧类型编码

位 0~4	帧类型
00000	信标帧
00001	数据帧
00010	聚合帧
00011	GACK
00100	NACK
00101	加入请求命令帧
00110	加入响应命令帧
00111	离开请求命令帧
01000	离开响应命令帧
01001	设备状态报告命令帧
01010	信道状况报告命令帧
01011	双向时间同步请求命令帧
01100	双向时间同步响应命令帧
01101	远程读属性请求命令帧
01110	远程读属性响应命令帧
01111	远程配置属性请求命令帧
10000	远程配置属性响应命令帧
10001	密钥建立请求命令帧(见 11.7.7)
10010	密钥建立响应命令帧(见 11.7.7)
10011	密钥更新请求命令帧(见 11.7.7)
10100	密钥更新响应命令帧(见 11.7.7)
10101	安全告警请求命令帧(见 11.7.7)
10110~11111	保留

- 分段标识：长度为 1 个比特，用于指示该帧是否为分段中的某一段，0 表示不是分段，1 表示为分段；如果分段标识值为 0，则 DLL 帧头中的分段数量子域和分段序列号子域无效；
- 抢占标识：长度为 1 个比特，用于指示该帧是否为抢占帧，0 表示为非抢占，1 表示抢占；
- 地址类型：长度为 1 个比特，用于表示 DLL 帧头中源或目的地址域的类型，编码见表 65。

表 65 地址类型编码

位 7	地址类型
0	64 位长地址
1	8/16 位长地址

#### 8.4.2 数据帧格式

DLL 的数据帧格式如图 49 所示。

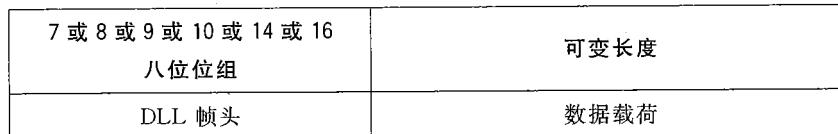


图 49 数据帧格式

各域说明如下：

- DLL 帧头，如图 47 所示；
- 数据载荷：可变长度，表示数据的内容，具体见 8.2.2 Payload。

#### 8.4.3 聚合帧格式

DLL 的数据帧格式如图 50 所示。

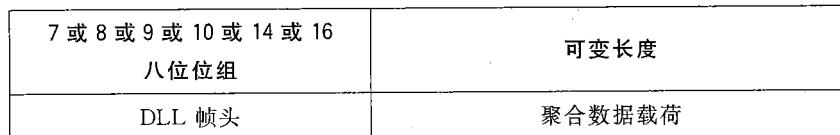


图 50 聚合帧格式

各域说明如下：

- DLL 帧头，如见图 47 所示；
- 数据载荷：可变长度，表示被聚合数据的内容，格式如图 30 所示。

#### 8.4.4 NACK 帧格式

DLL 的 NACK 帧格式如图 51 所示。

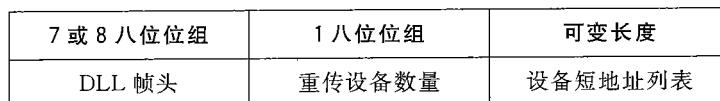


图 51 NACK 帧格式

各域说明如下：

- DLL 帧头，如图 47 所示；

- 重传设备数量:长度为1个八位位组,表示需要重传的现场设备的数量;
- 设备短地址列表:可变长度,表示需要重传的现场设备地址的列表。

#### 8.4.5 GACK 帧格式

DLL 的 GACK 帧格式如图 52 所示。

7 或 8 八位位组	1 八位位组	可变长度
DLL 帧头	设备数	GACK 信息

图 52 GACK 帧格式

各域说明如下:

- DLL 帧头,如图 47 所示;
- 设备数:长度为1个八位位组,用于指示发送给接入设备数据的现场设备的数量;
- GACK 信息:可变长度,如图 53 所示,包括设备短地址和帧序列号;
- 帧序列号:长度为2个八位位组,用于表示接收到现场设备的帧的序列号。

1 或 2 八位位组	2 八位位组
设备短地址	帧序列号

图 53 GACK 信息

#### 8.4.6 信标帧格式

信标帧格式如图 54 所示。

7 或 8 八位位组	2 八位位组	2 八位位组	2 八位位组	2 八位位组	1 八位位组	8 八位位组	可变长度
DLL 帧头	超帧长度	时隙长度	信标帧相 对时隙号	共享时隙 起始相对 时隙号	共享 时隙数	绝对 时间值	信标帧 载荷

图 54 信标帧格式

各域说明如下:

- DLL 帧头,格式如图 47 所示;
- 超帧长度:长度为2个八位位组,表示初始超帧的长度,以时隙为单位,详见 8.1.2;
- 时隙长度:长度为2个八位位组,表示网络设定的时隙长度,取值详见 6.7.1.2.1 TimeSlotDuration;
- 信标帧相对时隙号:长度为2个八位位组,表示发送该信标帧的相对时隙号,详见 8.1.2。
- 共享时隙起始相对时隙号:长度为2个八位位组,表示共享时隙在超帧中的开始位置,详见 8.1.2;
- 共享时隙数:长度为1个八位位组,表示共享时隙的总数,格式如图 55 所示;其中,位 0~3 表示上行共享时隙数,用于待加入网络的现场设备采用基于竞争方式向网关设备发送加入请求和返回读/配置属性响应;位 4~7 表示下行时隙数,用于网关设备向现场设备返回加入响应、远程读/配置属性请求以及对现场设备组态;详见 8.1.2;
- 绝对时间值:长度为8个八位位组,表示发送信标帧的绝对时间值,取值详见 6.7.1.2.1 TimeValue;

——信标帧载荷:可变长度,表示信标帧中载荷的内容。

位:0~3	位:4~7
上行共享时隙数	下行时隙数

图 55 共享时隙数格式

#### 8.4.7 加入请求命令帧格式

加入请求命令帧格式如图 56 所示。

14 八位位组	0/8 八位位组
DLL 帧头	安全材料

图 56 加入请求命令帧格式

各域说明如下:

——DLL 帧头,格式如图 47 所示;

——安全材料:长度为 8 个八位位组,表示设备入网的认证信息,取值详见 8.3.4.1 SecMaterial。

#### 8.4.8 加入响应命令帧格式

DLL 的加入响应命令帧格式如图 57 所示。

14 八位位组	1 八位位组	1 或 2 八位位组
DLL 帧头	加入状态	分配的短地址

图 57 加入响应命令帧格式

各域说明如下:

——DLL 帧头,格式如图 47 所示;

——加入状态:长度为 1 个八位位组,用于表示现场设备加入的状态,取值详见 8.3.4.3 Status;

——分配的短地址:长度为 1 或 2 个八位位组,用于表示新加入网络的现场设备的短地址,取值详见 8.3.4.3 ShortAddr。

#### 8.4.9 离开请求命令帧格式

离开请求命令帧格式如图 58 所示。

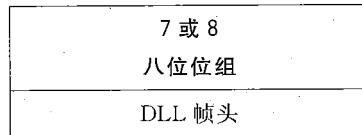


图 58 离开请求命令帧格式

离开请求命令帧仅包含 DLL 帧头,格式如图 47 所示。

#### 8.4.10 离开响应命令帧格式

离开响应命令帧格式如图 59 所示。



图 59 离开响应命令帧格式

离开响应命令帧仅包含 DLL 帧头, 格式如图 47 所示;

#### 8.4.11 设备状态报告命令帧格式

设备状态报告命令帧格式如图 60 所示。

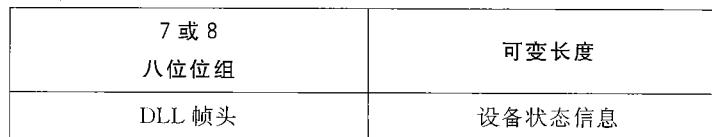


图 60 设备状态报告命令帧格式

各域说明如下:

——DLL 帧头, 格式如图 47 所示;

——设备状态信息: 可变长度, 表示报告的设备状态信息, 取值详见 8.3.5.1 PowerSupplyStatus。

#### 8.4.12 信道状况报告命令帧格式

信道状况报告命令帧格式如图 61 所示。



图 61 信道状况报告命令帧格式

各域说明如下:

——DLL 帧头, 格式如图 47 所示;

——信道状况信息: 可变长度, 表示报告的信道状况信息, 取值详见 8.3.6.1 ChannelConditionInfo。

#### 8.4.13 双向时间同步请求命令帧格式

双向时间同步请求命令帧格式如图 62 所示。

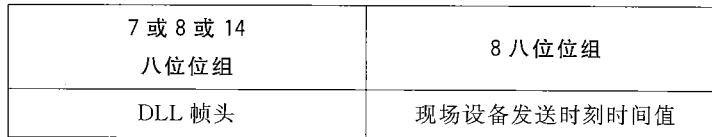


图 62 双向时间同步请求命令帧格式

各域说明如下:

——DLL 帧头, 格式如图 47 所示;

——现场设备发送时刻时间值: 长度为 8 个八位位组, 表示现场设备发送双向时间同步请求命令帧的时间值, 在物理层打入时戳, 取值详见 8.3.3.3 FieldDeviceTimeValue。

#### 8.4.14 双向时间同步响应命令帧格式

双向时间同步响应命令帧格式如图 63 所示。

7 或 8 或 14 八位位组	8 八位位组	8 八位位组
DLL 帧头	现场设备发送时刻时间值	接入设备接收时刻时间值

图 63 双向时间同步响应命令帧格式

各域说明如下：

- 帧控制, 格式如图 47 所示;
- 现场设备发送时刻时间值, 详见 8.4.13;
- 接入设备接收时刻时间值: 长度为 8 个八位位组, 表示接入设备接收到双向时间同步命令请求帧的时间值, 详见 8.3.3.3。

#### 8.4.15 远程读属性请求命令帧格式

远程读属性请求命令帧格式如图 64 所示。

7 或 8 八位位组	1 八位位组	1 八位位组	2 八位位组	2 八位位组
DLL 帧头	属性标识符	属性成员 标识符	多个属性值的 第一个存储索引	属性数目

图 64 远程读属性请求命令帧格式

各域说明如下：

- DLL 帧头, 格式如图 47 所示;
- 属性标识符: 长度为 1 个八位位组, 取值详见 8.3.7.1 AttributeID;
- 属性成员标识符: 长度为 1 个八位位组, 取值详见 8.3.7.1 MemberID;
- 多个属性值的第一个存储索引: 长度为 2 个八位位组, 取值详见 8.3.7.1 FirstStoreIndex;
- 属性数目: 长度为 2 个八位位组, 取值详见 8.3.7.1 Count。

#### 8.4.16 远程读属性响应命令帧格式

远程读属性响应命令帧格式如图 65 所示。

7 或 8 八位位组	1 八位位组	1 八位位组	1 八位位组	2 八位位组	2 八位位组	可变长度
DLL 帧头	执行结果	属性 标识符	属性成员 标识符	多个属性值 的第一个存 储索引	属性数目	属性值

图 65 远程读属性响应命令帧格式

各域说明如下：

- 帧控制, 格式如图 47 所示;
- 执行结果: 长度为 1 个八位位组, 取值详见 8.3.7.3 Status。
- 属性标识符: 长度为 1 个八位位组, 取值详见 8.3.7.3 AttributeID;

- 属性成员标识符:长度为1个八位位组,取值详见8.3.7.3 MemberID;
- 多个属性值的第一个存储索引:长度为2个八位位组,取值详见8.3.7.1 FirstStoreIndex;
- 属性数目:长度为2个八位位组,取值详见8.3.7.3 Count;
- 属性值:可变长度,取值详见8.3.7.3 AttributeValue。

#### 8.4.17 远程配置属性请求命令帧格式

远程配置属性请求命令帧格式如图66所示。

7或8 八位位组	1八位位组	1八位位组	1八位位组	2八位位组	2八位位组	可变长度
DLL帧头	远程属性操作	属性标识符	属性成员 标识符	多个属性值 的第一个 存储索引	属性数目	属性值

图66 远程配置属性请求命令帧格式

各域说明如下:

- DLL帧头,格式如图47所示;
- 远程属性操作:长度为1个八位位组,取值详见8.3.8.1 AttributeOption;
- 属性标识符:长度为1个八位位组,取值详见8.3.8.1 AttributeID;
- 属性成员标识符:长度为1个八位位组,取值详见8.3.8.1 MemberID;
- 多个属性值的第一个存储索引:长度为2个八位位组,取值详见8.3.8.1 FirstStoreIndex;
- 属性数目:长度为2个八位位组,取值详见8.3.8.1 Count;
- 属性值:可变长度,取值详见8.3.8.1 AttributeValue。

#### 8.4.18 远程配置属性响应命令帧格式

远程配置属性响应命令帧格式如图67所示。

7或8八位位组	1八位位组	1八位位组	1八位位组	2八位位组	2八位位组	1八位位组
DLL帧头	远程属性操作	属性标识符	属性成员 标识符	多个属性值 的第一个 存储索引	属性数目	执行结果

图67 远程配置属性响应命令帧格式

各域说明如下:

- 帧控制,格式如图47所示;
- 远程属性操作:长度为1个八位位组,取值详见8.3.8.3 AttributeOption;
- 属性标识符:长度为1个八位位组,取值详见8.3.8.3 AttributeID;
- 属性成员标识符:长度为1个八位位组,取值详见8.3.8.3 MemberID;
- 多个属性值的第一个存储索引:长度为2个八位位组,取值详见8.3.8.3 FirstStoreIndex;
- 属性数目:长度为2个八位位组,取值详见8.3.8.3 Count;
- 执行结果:长度为1个八位位组,取值详见8.3.8.3 Status。

### 8.5 数据链路层状态机

#### 8.5.1 接入设备 DLL 状态机

接入设备的数据链路层的状态机如图68所示。

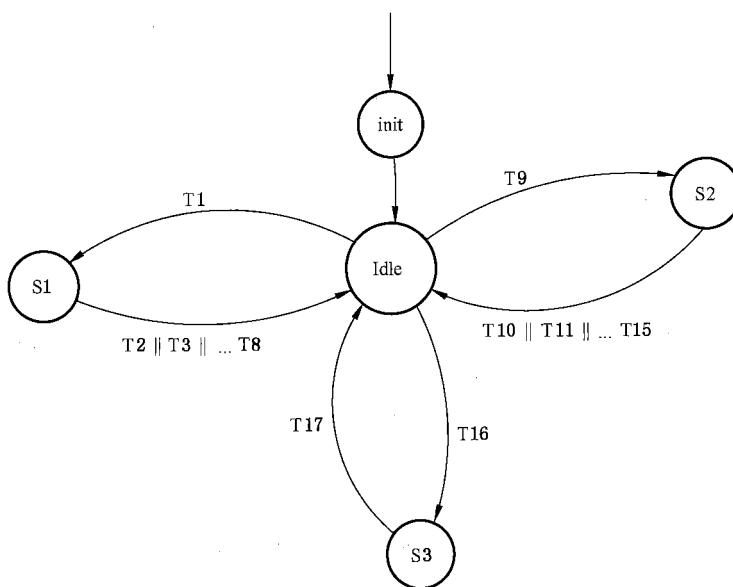


图 68 接入设备 DLL 状态机

接入设备的数据链路层的状态转移如表 66 所示。

表 66 接入设备 DLL 状态转移表

编号	当前状态	事件\条件 => 动作	下一状态
T1	Idle	PHY-DATA.indication() => FrameType = GetFrameType(DLPDU);	S1
T2	S1	FrameType == JoinRequest Command => PhyAddr = GetPhyAddr(DLPDU); DLME-JOIN.indication(PhyAddr, SecMaterial);	Idle
T3	S1	FrameType == Data => SrcAddr = GetSrcAddr(DLPDU); PayloadLength = GetPayloadLength(DLPDU); Payload = GetPayload(DLPDU); DLDE-DATA.indication(SrcAddr, DataType := Data, PayloadLength, Payload);	Idle
T4	S1	FrameType == RemoteAttributeSetResponse => DLME-INFO-SET.confirm(Handle, Status);	Idle
T6	S1	FrameType == DeviceStatusReport => DLME-DEVICE-STATUS.indication(PowerSupplyStatus)	Idle
T7	S1	FrameType == ChannelConditionReport => DLME-CHANNEL-CONDITION.indication(ShortAddr, Count, ChannelConditionInfo);	Idle

表 66 (续)

编号	当前状态	事件\条件 => 动作	下一状态
T8	S1	FrameType == TowWayTimeSynchronizationRequest => DLME-TIME-SYN.indication(ShortAddr, FieldDeviceTimeValue);	Idle
T9	Idle	APP or DMAP invokes primitives of DLL	S2
T10	S2	DLME-JOIN.response() => BuildFrame(FrameType := JoinResponse);	Idle
T11	S2	DLME-INFO-SET.request() => BuildFrame(FrameType := RemoteAttributeSetRequest);	Idle
T12	S2	DLME-INFO-GET.request() => BuildFrame(FrameType := RemoteAttributeGetRequest);	Idle
T13	S2	DLDE-DATA.request() => BuildFrame(FrameType := Data);	Idle
T14	S2	DLME-TIME-SYN.response() => BuildFrame(FrameType := TowWayTimeSynchronizationResponse);	Idle
T15	S2	DLME-LEAVE.request() => BuildFrame(FrameType := LeaveRequest);	Idle
T16	Idle	Slot timeout => If (LinkType == TRANSMIT_LINK) { Phy_set_RF_mode(TRANSMIT_MODE); DLPDU := GetDLPDUFromQueue(); PHY-DATA.request(DLPDU); } Else if (LinkType == RECEIVE_LINK) { Phy_set_RF_mode(RECEIVE_MODE); } Else if (LinkType == SHARED_TRANSMIT_LINK) { PHYSendWithBackoff(DLPDU); }	S3
T17	S3	(TransmissionCompleteISR()    (ReceiveCompleteISR())) =>	Idle

接入设备 DLL 具有以下状态。

——Init 状态

处于 Init 状态的现场设备 DLL 执行初始化过程, 初始化完成后进入 Idle 状态。

——Idle 状态

处于 Idle 状态的现场设备 DLL 有以下事件可能发生:

- a) 物理层收到帧, 调用 PHY-DATA.indication 原语将其转交给数据链路层; DLL 解包, 获得 DLPDU, DLL 状态机从 Idle 状态转移到 S1 状态;
- b) 应用层或 DMAP 调用数据链路层的原语, DLL 状态机从 Idle 状态转移到 S2 状态;
- c) 时隙触发, DLL 根据链路类型(LinkType 见表 18), 将 RF 设置为对应的接收或发送模式, 并执行发送或接收, DLL 状态机从 Idle 状态转移到 S3 状态。

——S1 状态

处于 S1 状态的 DLL 进一步根据解包后的帧类型(见 8.4.1), 触发不同条件, 进而执行不同的动作, 执行后进入 Idle 状态:

- a) 收到加入请求命令帧(见 8.4.7), DLL 调用 DLME-JOIN.indication 原语, DLL 状态机从 S1 状态转移到 Idle 状态;
- b) 收到数据帧(见 8.4.2), DLL 调用 DLDE-DATA.indication 原语, DLL 状态机从 S1 状态转移到 Idle 状态;
- c) 收到远程配置属性响应命令帧(见 8.4.18), DLL 调用 DLME-INFO-SET.confirm 原语, DLL 状态机从 S1 状态转移到 Idle 状态;
- d) 收到远程读属性响应命令帧(见 8.4.16), DLL 调用 DLME-INFO-GET.confirm 原语, DLL 状态机从 S1 状态转移到 Idle 状态;
- e) 收到双向时间同步请求帧(见 8.4.13), DLL 调用 DLME-TIME-SYN.indication 原语, DLL 状态机从 S1 状态转移到 Idle 状态;
- f) 收到设备状态报告命令帧(见 8.4.11), DLL 调用 DLME-DEVICE-STATUS.indication 原语, DLL 状态机从 S1 状态转移到 Idle 状态;
- g) 收到信道状况报告命令帧(见 8.4.12), DLL 调用 DLME-CHANNEL-CONDITION.indication 原语, DLL 状态机从 S1 状态转移到 Idle 状态;

——S2 状态

处于 S2 状态的 DLL 有以下事件可能发生:

- a) DMAP 调用 DLME-JOIN.response 原语, DLL 构造加入响应帧(见 8.4.8)并投到发送队列, DLL 状态机从 S2 状态转移到 Idle 状态;
- b) DMAP 调用 DLME-INFO-SET.request 原语, DLL 构造远程配置属性请求帧(见 8.4.17)并投到发送队列, DLL 状态机从 S2 状态转移到 Idle 状态;
- c) DMAP 调用 DLME-INFO-GET.request 原语, DLL 构造远程读服务请求帧(见 8.4.15)并投到发送队列, DLL 状态机从 S2 状态转移到 Idle 状态;
- d) DLL 调用 DLDE-DATA.request 原语, DLL 构造数据帧(见 8.4.2)并投到发送队列, DLL 状态机从 S2 状态转移到 Idle 状态;
- e) DMAP 调用 DLME-LEAVE.request 原语, DLL 构造离开请求帧(见 8.4.9)并投到发送队列, DLL 状态机从 S2 状态转移到 Idle 状态;
- f) DMAP 调用 DLME-TIME-SYN.response 原语, DLL 构造双向时间同步响应帧(见 8.4.14)并投到发送队列, DLL 状态机从 S2 状态转移到 Idle 状态。

——S3 状态

处于 S3 状态的 DLL 根据链路类型(LinkType 见表 18), 将 RF 设置为对应的接收或发送模式, 并执行发送或接收过程。当链路类型为接收时, 将 RF 设置为接收模式; 当链路类型为发送时, 将 RF 设置为发送模式, 从发送队列中获取帧, 并调用 PHY-DATA.request 原语发送; 当发送链路为共享发送时, 调用 PHYSendWithBackoff 函数采用退避方式发送。当接收或发

送完成中断处理函数执行完时,DLL 状态机从 S3 状态转移到 Idle 状态。

### 8.5.2 现场设备 DLL 状态机

现场设备的数据链路层的状态机如图 69 所示。

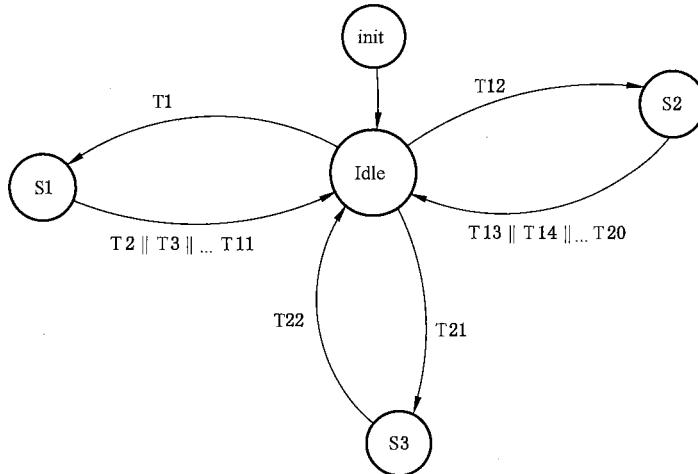


图 69 现场设备 DLL 状态机

现场设备的数据链路层的状态转移如表 67 所示。

表 67 现场设备 DLL 状态转移表

编号	当前状态	事件\条件 => 动作	下一状态
T1	Idle	PHY-DATA.indication() => FrameType := GetFrameType(DLPDU);	S1
T2	S1	FrameType == JoinResponse => DLME-JOIN.confirm(Status, ShortAddr);	Idle
T3	S1	FrameType == RemoteAttributeSetRequest => DLME-INFO-SET.indication(SrcAddr, AttributeOption, AttributeID, MemberID, FirstStoreIndex, Count, AttributeValue);	Idle
T4	S1	FrameType == RemoteAttributeGetRequest => DLME-INFO-GET.indication(SrcAddr, AttributeID, MemberID, FirstStroreIndex, Count);	Idle
T5	S1	FrameType == Data => DLDE-DATA.indication(SrcAddr, DataType, PayloadLength, Payload);	Idle
T6	S1	FrameType == LeaveRequest => DLME-LEAVE.indication();	Idle

表 67 (续)

编号	当前状态	事件\条件 => 动作	下一状态
T7	S1	(FrameType == NACK) && (NACK is for DeviceStatusReportCommand) => DLME-DEVICE-STATUS.confirm(Status);	Idle
T8	S1	(FrameType == NACK) && (NACK is for ChannelConditionReportCommand) => DLME-CHANNEL-CONDITION.confirm (Status);	Idle
T9	S1	(FrameType == Beacon) && (device Status == NOT_JOINED) => DoSynchronization(); DLME-DISCVOERY.confirm (Status := SUCCESS, BeaconCount, SuperframeLength, TimeslotDuration, FirstShareTimeslotNum, SharedTimeslotCount, AbsoluteTimeVaule, BeaconDescription);	Idle
T10	S1	(FrameType == Beacon) && (device Status != NOT_JOINED) => DoSynchronization();	Idle
T11	S1	FrameType == TowWayTimeSynchronizationResponse => DLME-TIME-SYN.confirm (Status, FieldDeviceTimeValue, ReceiveTimeValue);	Idle
T12	Idle	APP or DMAP invokes primitives of DLL	S2
T13	S2	DLME-JOIN.request() => BuildFrame(FrameType := JoinRequest);	Idle
T14	S2	DLME-INFO-SET.response() => BuildFrame(FrameType := RemoteAttributeSetResponse);	Idle
T15	S2	DLME-INFO-GET.response() => BuildFrame(FrameType := RemoteAttributeGetResponse);	Idle
T16	S2	DLDE-DATA.request() => BuildFrame(FrameType := Data);	Idle
T17	S2	DLME-DISCOVERY.request() => ScanChanel(Channels);	Idle
T18	S2	DLME-DEVICE-STATUS.request() => BuildFrame(FrameType := DeviceStatusReport);	Idle

表 67 (续)

编号	当前状态	事件\条件 => 动作	下一状态
T19	S2	DLME-TIME-SYN.request() => BuildFrame(FrameType == TowWayTimeSynchronizationRequest);	Idle
T20	S2	DLME-CHANNEL-CONDITION.request() => BuildFrame(FrameType == ChannelStatusReport);	Idle
T21	Idle	Slot timeout => If (LinkType == TRANSMIT_LINK) { Phy_set_RF_mode(TRANSMIT_MODE); DLPDU := GetDLPDUFromQueue(); PHY-DATA.request(DLPDU); } Else if (LinkType == RECEIVE_LINK) { Phy_set_RF_mode(RECEIVE_MODE); } Else if (LinkType == SHARED_TRANSMIT_LINK) { PHYSendWithBackoff(DLPDU); }	S3
T22	S3	TransmissionCompleteISR()    ReceiveCompleteISR() =>	Idle

现场设备 DLL 具有以下状态。

——Init 状态

处于 Init 状态的现场设备 DLL 执行初始化过程, 初始化完成后进入 Idle 状态。

——Idle 状态

处于 Idle 状态的现场设备 DLL 有以下事件可能发生:

- 物理层收到帧, 调用 PHY-DATA.indication 原语将其转交给数据链路层, DLL 解包, 获得 DLPDU, DLL 状态机从 Idle 状态转移到 S1 状态;
- 应用层或 DMAP 调用数据链路层的原语, DLL 状态机从 Idle 状态转移到 S2 状态;
- 时隙触发, DLL 根据链路类型, 将 RF 设置为对应的接收或发送模式, 并执行发送或接收, DLL 状态机从 Idle 状态转移到 S3 状态。

——S1 状态

处于 S1 状态的 DLL 进一步根据解包后的帧类型(见 8.4.1), 触发不同条件, 进而执行不同的动作, 但最后执行完都转回 Idle 状态:

- 收到加入响应命令帧(见 8.4.8), DLL 调用 DLME-JOIN.confirm 原语, DLL 状态机从 S1 状态

转移到 Idle 状态；

- b) 收到数据帧(见 8.4.2), DLL 调用 DLDE-DATA.indication 原语,DLL 状态机从 S1 状态转移到 Idle 状态；
- c) 收到远程配置属性请求命令帧(见 8.4.17), DLL 调用 DLME-INFO-SET.indication 原语,DLL 状态机从 S1 状态转移到 Idle 状态；
- d) 收到远程读属性请求命令帧(见 8.4.15), DLL 调用 DLME-INFO-GET.indication 原语,DLL 状态机从 S1 状态转移到 Idle 状态；
- e) 收到离开请求帧(见 8.4.9), DLL 调用 DLME-LEAVE.indication 原语,DLL 状态机从 S1 状态转移到 Idle 状态；
- f) 收到双向时间同步响应帧(见 8.4.14), DLL 调用 DLME-TIME-SYN.confirm 原语,DLL 状态机从 S1 状态转移到 Idle 状态；
- g) 收到用于回复设备状态报告命令帧的 NACK 帧(见 8.4.4), DLL 调用 DLME-DEVICE-STATUS.confirm 原语,DLL 状态机从 S1 状态转移到 Idle 状态；
- h) 收到用于回复信道状况报告命令帧的 NACK 帧(见 8.4.4), DLL 调用 DLME-CHANNEL-CONDITION.confirm 原语,DLL 状态机从 S1 状态转移到 Idle 状态；
- i) 收到信标帧(见 8.4.6)且现场设备处于未加入状态, DLL 进行时间同步,然后调用 DLME-DISCOVERY.confirm 原语,DLL 状态机从 S1 状态转移到 Idle 状态；

#### ——S2 状态

处于 S2 状态的 DLL 有以下事件可能发生：

- a) DMAP 调用 DLME-JOIN.request 原语,DLL 构造加入请求帧(见 8.4.7)并投到发送队列,DLL 状态机从 S2 状态转移到 Idle 状态；
- b) DMAP 调用 DLME-INFO-SET.response 原语,DLL 构造远程配置属性响应帧(见 8.4.18)并投到发送队列,DLL 状态机从 S2 状态转移到 Idle 状态；
- c) DMAP 调用 DLME-INFO-GET.response 原语,DLL 构造远程读服务响应帧(见 8.4.16)并投到发送队列,DLL 状态机从 S2 状态转移到 Idle 状态；
- d) DLL 调用 DLDE-DATA.request 原语,DLL 构造数据帧(见 8.4.2)并投到发送队列,DLL 状态机从 S2 状态转移到 Idle 状态；
- e) DMAP 调用 DLME-DISCOVERY.request 原语,DLL 设置 RF 为接收模式,根据参数扫描信道,监听 Beacon,DLL 状态机从 S2 状态转移到 Idle 状态；
- f) DMAP 调用 DLME-DEVICE-STATUS.request 原语,DLL 构造设备状态报告请求帧(见 8.4.11)并投到发送队列,DLL 状态机从 S2 状态转移到 Idle 状态；
- g) DMAP 调用 DLME-CHANNEL-CONDITION.request 原语,DLL 构造信道状况报告请求帧(见 8.4.12)并投到发送队列,DLL 状态机从 S2 状态转移到 Idle 状态。

#### ——S3 状态

处于 S3 状态的 DLL 根据链路类型,将 RF 设置为对应的接收或发送模式,并执行发送或接收过程。当链路类型为接收时,将 RF 设置为接收模式;当链路类型为发送时,将 RF 设置为发送模式,从发送队列中获取帧,并调用 PHY-DATA.request 原语发送;当发送链路为共享发送时,调用 PHYSendWithBackoff 函数采用退避方式发送。当接收或发送完成中断处理函数执行完时,DLL 状态机从 S3 状态转移到 Idle 状态。

### 8.5.3 DLL 状态机函数

DLL 状态机函数的定义见表 68。

表 68 DLL 状态机函数

函数	输入	输出	函数描述
GetFrameType()	DLPDU	Frametype	解析帧类型 Frametype 的取值包括： JoinRequest; Data; RemoteAttributeSetResponse; RemoteAttributeGetResponse; DeviceStatusReport; ChannelCondition Report; JoinResponse; RemoteAttributeSetRequest; RemoteAttributeGetRequest; LeaveRequest; LeaveResponse; NACK; GACK; Beacon
GetPhyAddr()	DLPDU	PhyAddr	解析物理地址
GetSrcAddr()	DLPDU	SrcAddr	解析源地址
GetPayloadLength()	DLPDU	PayloadLength	解析载荷长度
BuildFrame()	FrameType	—	构造帧后放入队列
Phy_set_RF_mode()	mode	—	设置 RF 模式
GetDLPDUFromQueue()	—	DLPDU	从传输队列取出 DLPDU
TransmissionCompleteISR()	—	—	传输结束后的中断函数
ReceiveCompleteISR()	—	—	接收结束后的中断函数
Do_Backoff()	—	ShareSendSlot	执行退避算法
PHY-CCA()	—	ChannelState	执行 PHY CCA ChannelState 的取值包括： BUSY; IDLE
DoSynchronization()	—	—	执行时间同步
ScanChanel()	Channels	—	扫描信道
PHYSendWithBackoff()	DLPDU	—	利用退避方式发送数据

## 9 接入设备与网关设备有线服务

### 9.1 概述

本部分定义了网关设备与接入设备之间的有线服务以及帧格式。接入设备与网关设备之间采用的有线通信方法不在本部分定义范围内。

## 9.2 接入设备加入网络

接入设备启动完成初始化后，并不执行网络发现。接入设备加入网络的过程包括：

- 待入网的接入设备利用服务标识符为 0 的通用帧(见 9.3)，向网关设备发送加入请求命令帧；
- 网关设备中的网络管理者利用服务标识符为 1 的通用帧(见 9.3)为待入网的接入设备返回加入响应，并利用服务标识符为 12 的通用帧(见 9.3)向接入设备中写入超帧、链路等通信资源。

服务标识符定义见 9.3。

接入设备的 8/16 位短地址全网惟一，见 6.3。

## 9.3 网关设备与接入设备有线连接的帧格式

网关设备与接入设备有线连接的通用帧格式如图 70 所示。

1 八位位组	0/1 八位位组	0/8 八位位组	2 八位位组	可变长度
服务标识符	AdID	AD 长地址	服务参数长度	服务参数

图 70 网关设备与接入设备有线连接的帧格式

各域说明如下：

- 服务标识符：长度为 1 个八位位组，用于区分网关设备与接入设备之间通信的内容，见表 64；
- AdID：长度为 1 个八位位组，用于区分 AD（见表 20），当服务标识符为 0 或 1 时，该域无效；
- AD 长地址：长度为 0 或 8 个八位位组，仅当服务标识符为 0 或 1 时，AD 长地址域的值表示接入设备的 64 位长地址；否则，该域无效；
- 服务参数长度：长度为 2 个八位位组，表示服务参数的长度；
- 服务参数：可变长度，表示网关设备与接入设备之间有线通信的不同服务数据（见表 69）。

网关设备与接入设备之间有线通信的服务及服务参数如表 69 所示。

表 69 网关设备与接入设备之间的有线服务

服务标识符	服务	服务参数	服务传输方向
0	接入设备加入请求	见表 70	接入设备→网关设备
1	接入设备加入响应	见表 71	网关设备→接入设备
2	网关设备指示接入设备发送 GACK	见表 72	网关设备→接入设备
3	网关设备指示接入设备发送 NACK	见表 74	网关设备→接入设备
4	数据发送请求	DLDE-DATA.request 参数(见 8.2.2)	网关设备→接入设备
5	数据发送指示	DLDE-DATA.indication 参数(见 8.2.3)	接入设备→网关设备
6	设备加入指示	DLME-JOIN.indicaiton 参数(见 8.3.4.2)	接入设备→网关设备
7	设备加入响应	DLME-JOIN.response 参数(见 8.3.4.3)	网关设备→接入设备
8	设备状态报告指示	DLME-DEVICE-STATUS.indication 参数(见 8.3.5.2)	接入设备→网关设备
9	信道状况报告	DLME-CHANNEL-CONDITION.indication 参数(见 8.3.6.2)	接入设备→网关设备

表 69 (续)

服务标识符	服务	服务参数	服务传输方向
10	远程读属性请求	DLME-INFO-GET.request 参数(见 8.3.7.1)	网关设备→接入设备
11	远程读属性证实	DLME-INFO-GET.confirm 参数(见 8.3.7.4)	接入设备→网关设备
12	远程配置属性请求	DLME-INFO-SET.request 参数(见 8.3.8.1)	网关设备→接入设备
13	远程配置属性证实	DLME-INFO-SET.confirm 参数(见 8.3.8.4)	接入设备→网关设备
14	设备离开请求	DLME-LEAVE.request 参数(见 8.3.9.1)	网关设备→接入设备
15	密钥建立请求	KEY-ESTABLISH.request 参数(见 11.2.2.1)	网关设备→接入设备
16	密钥建立证实	KEY-ESTABLISH.confirm 参数(见 11.2.2.4)	接入设备→网关设备
17	密钥更新请求	KEY-UPDATE.request 参数(见 11.2.3.1)	网关设备→接入设备
18	密钥更新证实	KEY-UPDATE.confirm 参数(见 11.2.3.4)	接入设备→网关设备
19	安全告警指示	SEC-ALARM.indication 参数(见 11.2.4.2)	接入设备→网关设备
22~255	保留	—	—

服务标识符为 0 和 1 对应的服务用于接入设备加入网络,服务参数如表 70 和表 71 所示;服务标识符为 2 和 3 的服务用于网关设备指定一个接入设备发送 GACK 和 NACK,服务参数如表 72 和表 74 所示。

表 70 接入设备加入请求服务参数

参数名称	数据类型	取值范围	描述
NetworkID	Unsigned8	0~255	网络的 ID 号,用于多个网络共存的情况下标识网络(见 6.7.1.2.1 表 15 属性)
PhyAddr	Unsigned64	0~(2 <sup>64</sup> - 1)	待加入接入设备的长地址(见 6.7.1.2.1, LongAddress 属性)

表 71 接入设备加入响应服务参数

参数名称	数据类型	取值范围	描述
Status	Unsigned8	0~255	表示加入请求的结果: 0=SUCCESS; 1=网络 ID 不匹配; 2=认证失败; 3=网络规模超限; 其余保留
AdID	Unsigned8	0~255	GW 分配给待加入接入设备的标识符(见 6.7.1.2.2 AdID 属性)。Status 为 SUCCESS 时,该域有效
ADAddr	Unsigned16	0~65 535	GW 分配给待加入接入设备的短地址

表 72 网关设备指示接入设备发送 GACK 服务参数

参数名称	数据类型	取值范围	描述
DeviceCount	Unsigned8	0~255	表示所有接入设备接收到的帧对应的现场设备的数量
GACKInformation	GACKInfo_Struct 结构体列表	—	表示 DeviceCount 个现场设备的信息, 表示为 GACK- Info_Struct 结构体, 见表 73

表 73 GACKInfo\_Struct 结构体参数

参数名称	数据类型	取值范围	描述
DstAddr	Unsigned8/ Unsigned16/ Unsigned64	0~255/ 0~65 535/ 0~(2 <sup>64</sup> - 1)	目的地址, 表示长地址或者短地址
SequenceNumber	Unsigned16	0~65 535	帧序列号

表 74 网关设备指示接入设备发送 NACK 服务参数

参数名称	数据类型	取值范围	描述
RetryDeviceCount	Unsigned8	0~255	需要重传的设备的数量
DstAddressList	Unsigned8/ Unsigned16	0~65 535	目的地址, 见 6.7.1.2.1 DeviceShortAddress

## 10 应用层

### 10.1 概述

WIA-FA 应用层为用户提供分布式应用, 定义了与工业过程交互的应用对象, 也定义了支持工业现场环境中分布式应用间的通信服务。WIA-FA 应用层包括用户应用进程(UAP)和应用子层(ASL)两部分。每个 UAP 由一个或多个用户应用对象(UAO)组成。设备管理应用进程(DMAP)是一种特殊的 UAP(见 6.2)。ASL 定义的通信服务支持不同设备上多个 UAP 之间的通信。

### 10.2 应用层协议栈

应用层在 WIA-FA 通信协议栈中的位置和构成如图 71 所示, 其中灰色部分表示应用层相关部分。

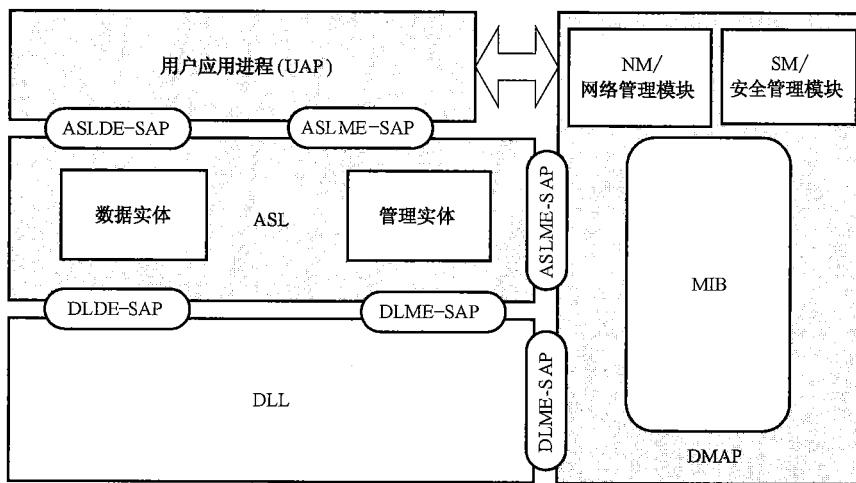


图 71 应用层在 WIA-FA 通信协议栈中的位置和构成

WIA-FA ASL 功能由 ASL 数据实体(ASLDE)和 ASL 管理实体(ASLME)实现。ASLDE 执行 ASL 数据传输功能(见 10.7.2),并在 ASLDE-SAP 上为 UAP 提供数据服务接口。ASLME 执行应用组态功能(见 10.5.5),并在 ASLME-SAP 上提供管理服务接口。

### 10.3 应用层功能

#### 10.3.1 数据功能

网关设备和现场设备之间传输三种类型的应用数据,包括周期性测量值(即输入数据)和设定值、控制值(即输出数据)、非周期性属性读写访问请求与响应,以及异常情况下的告警报告。为支持这些不同应用数据的使用和传输,WIA-FA 定义了相应的应用服务(见 10.6)和虚拟通信关系(见 10.3.3)。

WIA-FA 应用服务报文的最大长度受限于 DLL 资源和传输能力。MIB 中的 MaxPayloadLength 属性规定了 DLL 载荷的最大长度。

#### 10.3.2 管理功能

现场设备的输入数据和输出数据应按照预定义的周期而周期性传输。当一个现场设备加入 WIA-FA 网络时,主控计算机应对其进行组态,即规定该现场设备应使用的 UAO、这些 UAO 应进行周期性传输的输入数据和输出数据,以及相应的数据更新率(UAO 组态见 10.5.5.2)。

当现场设备上多个 UAO 被分配给一个 UAP 时,这些 UAO 应具有相同的数据更新率和使用相同的 P/S VCR。

#### 10.3.3 通信模型与虚拟通信关系

ASL 支持三种通信模型:客户机/服务器(Client/Server 或 C/S)模型、发布者/预订者(Publisher/Subscriber 或 P/S)模型,以及报告源/汇(Report source/Sink 或 R/S)模型。这些通信模型被用来传输相应优先级的应用数据。

- C/S 通信模型:适用于非周期、非实时的读/写访问和告警确认(NRT),通过单播方式传输;
- P/S 通信模型:适用于周期性过程数据发布(RT1),通过单播或广播方式传输;
- R/S 通信模型:适用于非周期的告警报告(RT2)和紧急命令(RT0),通过单播或广播方式传输。

网关设备和现场设备在不同通信模型中可担任角色如表 75 所示。

表 75 网关设备和现场设备之间的通信模型

网关设备	现场设备	通信方式	优先级	使 用
客户机	服务器	单播	NRT	网关设备读/写现场设备的 MIB 和 UAO 属性, 或者对现场设备进行告警确认
发布者	预订者	单播	RT1	网关设备向一个现场设备发布输出数据
		多播	RT1	网关设备向所有现场设备发布输出数据
预订者	发布者	单播	RT1	现场设备向网关设备发布输入数据
报告汇点	报告源点	单播	RT2	现场设备向网关设备报告告警
报告源点	报告汇点	单播	RT0	网关设备向一个现场设备发出启动和停止命令
		广播	RT1	网关设备向所有现场设备发出启动和停止命令

在 WIA-FA 中, 这 3 种通信模型由相应的虚拟通信关系(VCR)实现。VCR 定义了网关设备和现场设备之间的逻辑通信关系, 在设备中以担任相应角色的 VCR 端点表示。VCR 端点定义了 VCR 的通信相关属性, 在设备内通过 VCR\_ID 唯一标识。

## 10.4 应用数据

### 10.4.1 概述

在 WIA-FA 设备中, 供 UAO 使用的应用数据包括属性数据、过程数据和事件数据。属性数据可被进行非周期性读写访问, 包括 MIB 中的各结构化属性和非结构化属性(见 6.7.1.2), 以及 UAO 中与过程或工艺相关的各属性。过程数据是指在现场设备与网关设备之间周期性传输的 UAO 的输入数据和输出数据。事件数据是指现场设备向网关设备报告的告警。

### 10.4.2 过程数据

现场设备与网关设备之间进行周期性传输的过程数据包括输入数据和输出数据。输入数据是指传感器测量值或执行器反馈值, 输出数据是指执行器的设定值和控制值。

### 10.4.3 事件数据

WIA-FA 为现场设备定义了事件数据 EventData 来维护告警事件。当有异常情况发生时, 事件数据的相应事件标志位(EventFlag)应被置位。当异常状况消失时, 相应事件标志位(EventFlag)应被复位。确认标志位(AckFlag)用来指示该告警事件是否应被确认。事件数据 EventData 定义如表 76 所示。UAO 定义的事件如表 77 所示。

表 76 事件数据 EventData 定义

成员标识符	成员名称	数据类型	数据长度 (八位位组)	取值范围	描述
1	EventFlag	Bit Field	2	—	每位编码如下： 0=事件不存在 1=事件存在
2	AckFlag	Bit Field	2	—	该标志指示相应的事件是否需要被确认,每位编码如下： 0=不需要确认 1=需要确认

表 77 UAO 事件定义

位	事件类型	描述
0	CONFIGURATION_ERROR	组态错误
1	SENSOR_FAULT	传感器故障
2	ACTUATOR_FAULT	执行器故障
3	INPUT_EXCEEDS_UPPER_LIMIT	输入超出上限
4	INPUT_EXCEEDS_LOWER_LIMIT	输入超出下限
5	OUTPUT_EXCEEDS_UPPER_LIMIT	输出超出上限
6	OUTPUT_EXCEEDS_LOWER_LIMIT	输出超出下限
7	PROCESS_DATA_NOT_UPDATED	过程数据未更新
8	PROCESS_DATA_LENGTH_INCONSISTENT	过程数据长度不一致
9~15	Reserved	保留
16~23	Manufacturer specific events	制造商特定

## 10.5 用户应用进程

### 10.5.1 概述

WIA-FA 定义了分布式应用进程(DAP)来实现工业现场环境中的分布式应用。一个 DAP 可分布于一个或多个 WIA-FA 设备。DAP 在设备上的实现用 UAP 表示。UAP 在设备中通过 UAP\_ID 唯一标识。一个 WIA-FA 设备可支持一个或多个 UAP。图 72 给出了 WIA-FA 网络上 DAP 与 UAP 之间的关系。DMAP 是一种特殊类型的 UAP, 实现系统管理和安全管理功能(见 6.2)。每个 WIA-FA 设备应仅实现一个 DMAP。DMAP 的 UAP\_ID 应为 0。

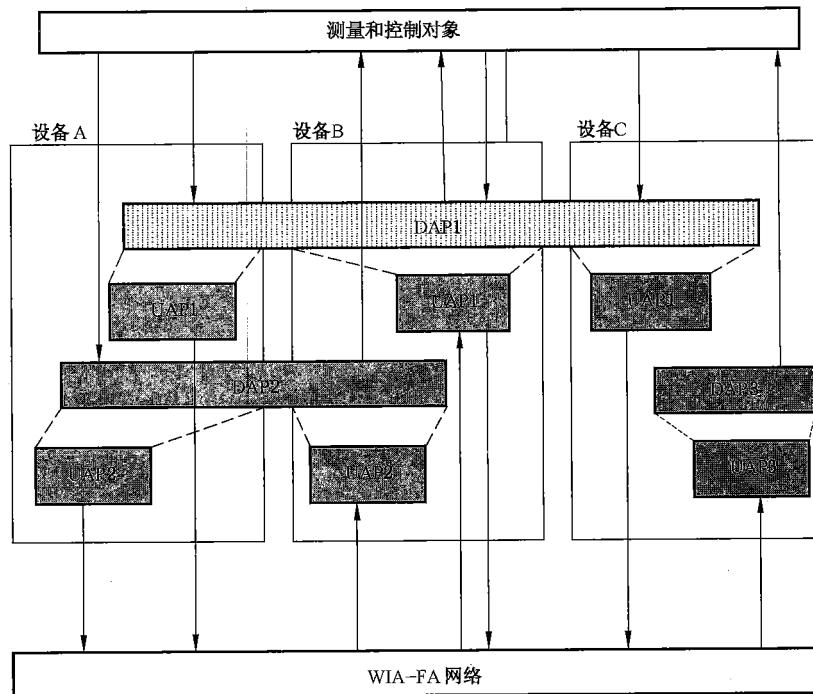


图 72 用户应用进程实现方式

### 10.5.2 用户应用对象 UAO

UAP 在现场设备内由一个或一组用户应用对象(UAO)构成,如图 73 所示。每个 UAO 管理并提供 WIA-FA 报文在网络上和在设备内的实时交换。

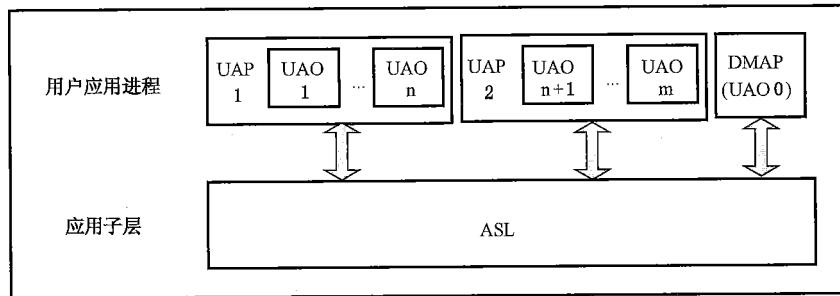


图 73 用户应用对象

按照工业应用需要,WIA-FA 支持 4 种 UAO 类型:模拟量输入(AI)、模拟量输出(AO)、数字量输入(DI),和数字量输出(DO),允许使用制造商特定的 UAO。现场设备实现哪些类型的 UAO 是可选的。UAO 属性定义不在本部分范围内。

UAO 在设备内通过 UAO 标识符(ObjectID)惟一标识。DMAP 是一种特殊类型的 UAP 且仅具有一个 UAO(UAO\_ID=0)。

### 10.5.3 网关设备上的 IO 数据映像

网关设备是否为现场设备实现相应的 IO 数据映像是可选的。当一个现场设备加入 WIA-FA 网络后,网关设备可为其分配一个 IO 数据映像。该数据映像用来缓存现场设备周期性传输的输入数据和输出数据。这样,当网关设备与另一个控制网络(如现场总线)互连时,可作为一个远程 I/O 设备与控

制网络上的其他设备通信。

网关设备应保存现场设备的组态数据,以解释该现场设备的输入数据和输出数据。

现场设备离开 WIA-FA 网络时,网关设备应释放该现场设备对应的 IO 数据映像(如实现)。

图 74 给出了网关设备实现 IO 数据映像的一个示例。IO 数据映像的实现是制造商特定的且不在本部分范围内。

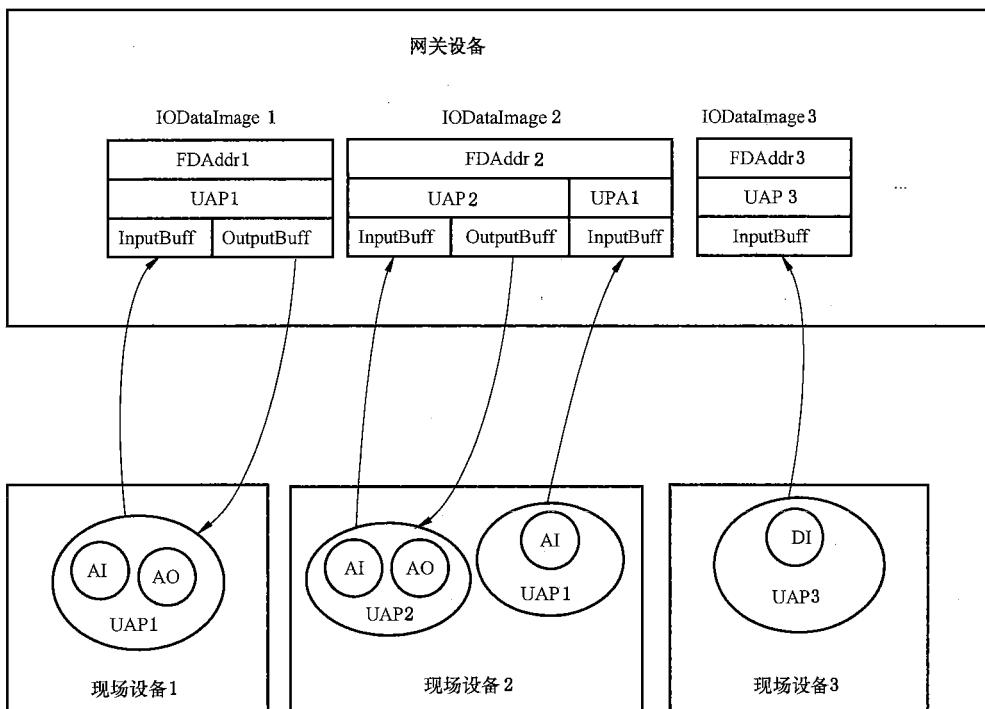


图 74 网关设备实现 IO 数据映像的示例

#### 10.5.4 告警机制

现场设备内每个 UAO 都维护一个事件数据(EventData 类型,见表 76 定义)。当有一个异常情况(告警事件)发生或消失时,事件数据的相应事件标志位(EventFlag)应被置位(发生)或复位(消失)。事件数据的 AckFlag 位指示对应的事件是否应被网关设备确认,其值可由网关设备组态设置。UAP 应将事件数据封装成 REPORT 请求报文,并将其发送给网关设备。

网关设备应为每个现场设备维护一个告警队列来保存接收到的告警事件。主控计算机(操作员)根据告警事件的 AckFlag 指示(AckFlag=1),可通过向现场设备返回一个 REPORT ACK 请求报文来对告警进行确认。

对于要求确认的告警事件,当现场设备接收到 REPORT ACK 请求报文时,UAP 应清除之前报告的 UAO 告警事件,即将 UAO 告警数据的 EvenFlag 位复位。如果现场设备在 AlarmRptDur(见表 15)时间内未接收到 REPORT ACK 请求报文,则 UAP 重复报告该告警事件。AlarmRptDur 值可由主控计算机组态设置。UAO 的告警事件是否需要确认也是可组态的。

现场设备可一次报告多个告警事件,网关设备可根据实际情况一次确认部分或全部告警事件。

#### 10.5.5 应用组态过程

##### 10.5.5.1 概述

网关设备对现场设备的应用组态过程应至少包括以下步骤:

——配置通信和应用相关的所有特性(MIB 中非结构化属性,见表 15);

——配置应用要使用的 UAO 实例;

——配置应用层通信所使用的 C/S VCR、P/S VCR 和 R/S VCR。

设备的非结构化属性是整个 WIA-FA 网络统一配置的,即所有设备应设置相同的值。主控计算机在组建 WIA-FA 网络初期,应首先配置网关设备自身 MIB 的非结构化属性。之后,每个现场加入网络后,网关设备都对其配置相同的值。该组态过程取决于实现,不在本标准范围内。

### 10.5.5.2 配置 UAO

现场设备加入 WIA-FA 网络后,网关设备应首先读取其 DeviceList 的 NumOfSupUAO 值(见表 20)来获取该现场设备支持的 UAO 类的个数,并读取 SupUAOList(UAOClassDesc\_Struct 结构体类型,见表 23 定义),来获取该现场设备实现的所有 UAO 类的描述,包括每个 UAO 类的 Class\_ID、UAO 类型、最小数据更新率,以及该 UAO 类支持的输入数据和输出数据的数据类型和长度。网关设备应将该信息传送给主控计算机。

根据实际应用需求,主控计算机应将现场设备支持的 UAO 类实例化为若干 UAO,并将这些 UAO 分配给一个或多个 UAP。属于同一 UAP 的 UAO 应具有相同的数据更新率。主控计算机应向现场设备写入 DeviceList 的 NumOfCfgUAOList 值(见表 20)来配置被组态的 UAO 实例的个数,并写入 CfgUAOList(UAOInstDesc\_Struct 结构体类型,见表 25 定义)来实例化要使用的所有 UAO。被组态 UAO 的所有输入/输出数据长度总和应小于该 UAO 类规定的最大输入/输出数据长度。在随后的周期性过程数据传输中,输入/输出数据的类型和顺序应与组态配置一致。

当现场设备离开 WIA-FA 网络时,应清除其 CfgUAOList。

### 10.5.5.3 配置 VCR

#### 10.5.5.3.1 概述

完成 UAO 组态后,网关设备还应根据 UAO 组态来配置现场设备的 VCR 列表。主控计算机应通过向现场设备写 VCRLIST 来设置其 P/S VCR 和 R/S VCR 的通信相关属性。在网关设备与现场设备之间始终存在一个默认的 C/S VCR(VCR\_ID=0)。C/S VCR 的组态是可选的。

表 78 给出了现场设备不同 VCR 类型的属性配置概要。VcrEP\_Struct 定义见表 22。

表 78 VCR 属性组态一览表

成员标识符	成员名称	VCR 端点类型			
		SERVER	PUBLISHER	SUBSCRIBER	REPORT SOURCE
0	VCR_ID	0	被组态	被组态	被组态
1	VcrEP_Type	1	2	3	4
2	UAP_ID	0	被组态	被组态	被组态
3	PeerAddr	网关设备短地址	网关设备短地址	网关设备短地址	网关设备短地址
5	VCRActiveTime	无效,应被设为 0	0 或被组态	0 或被组态	无效,应被设为 0
6	DataUpdateRate	无效,应被设为 0	被组态	被组态	无效,应被设为 0
7	Deadline	无效,应被设为 0	被组态	被组态	无效,应被设为 0
8	WatchdogTime	默认值为 100 ms, 并可被主控计算机改变,	无效,应被设为 0	无效,应被设为 0	无效,应被设为 0

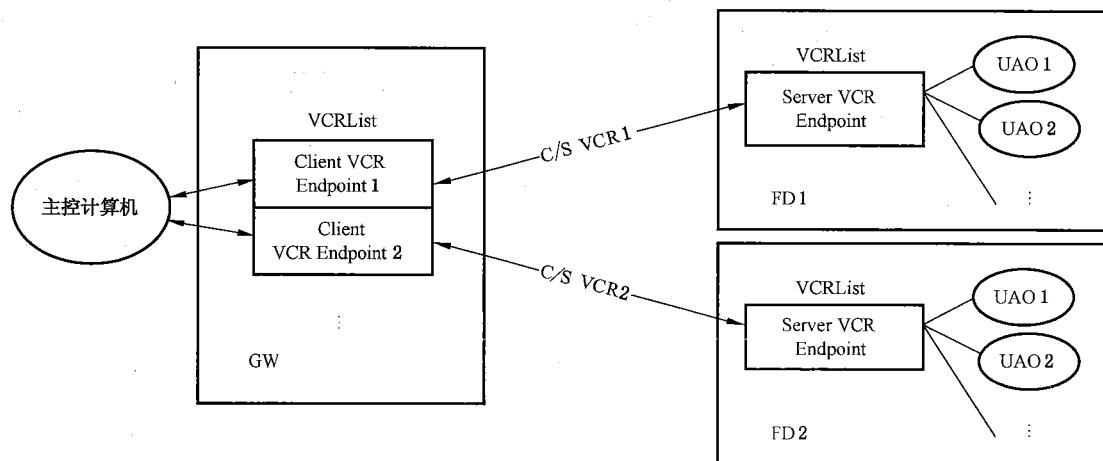
当对现场设备的 VCRLst 进行组态时,网关设备自身应生成相应的 VCR 端点以建立与现场设备之间的 VCR 连接。

当现场设备离开 WIA-FA 网络时,应清除除服务器 VCR 端点以外的 VCR 列表。

#### 10.5.5.3.2 C/S VCR 组态

网关设备与现场设备之间仅需要一个默认的 C/S VCR(VCR\_ID=0)。网关设备为客户机,现场设备为服务器。

图 75 给出了网关设备与现场设备之间的 C/S VCR 关系。



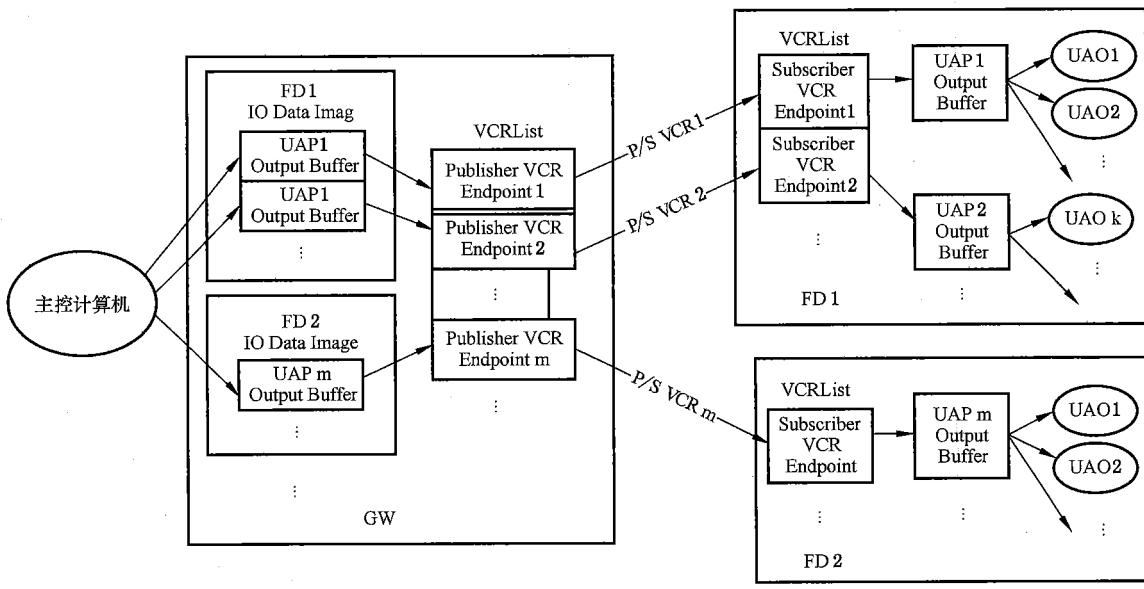
注:箭头指示数据传输方向。

图 75 网关设备与现场设备之间的 C/S VCR

#### 10.5.5.3.3 P/S VCR 组态

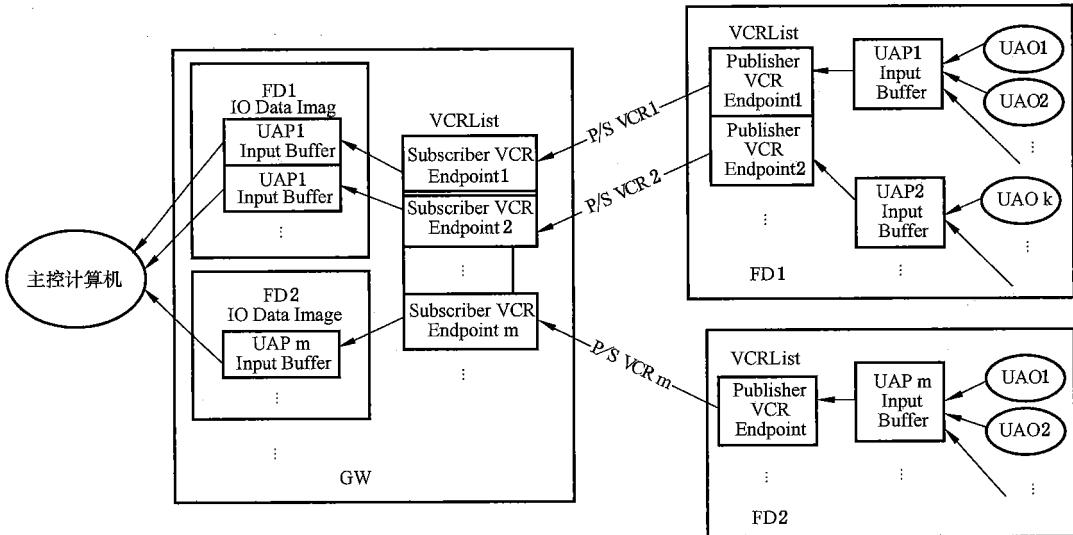
在现场设备上一个 UAP 可被分配多个 UAO。如果这些 UAO 具有输入数据,则应为其配置一个 Publisher VCR 端点,并且现场设备作为发布者。如果这些 UAO 具有输出数据,则应为其配置一个 Subscriber VCR 端点,并且现场设备作为预订者。每个 P/S VCR 端点应被分配一个缓冲区来保存输入数据或输出数据。

图 76 和图 77 给出了网关设备与现场设备之间的 P/S VCR 关系。



注：箭头指示数据传输方向。

图 76 网关设备与现场设备之间的 P/S VCR



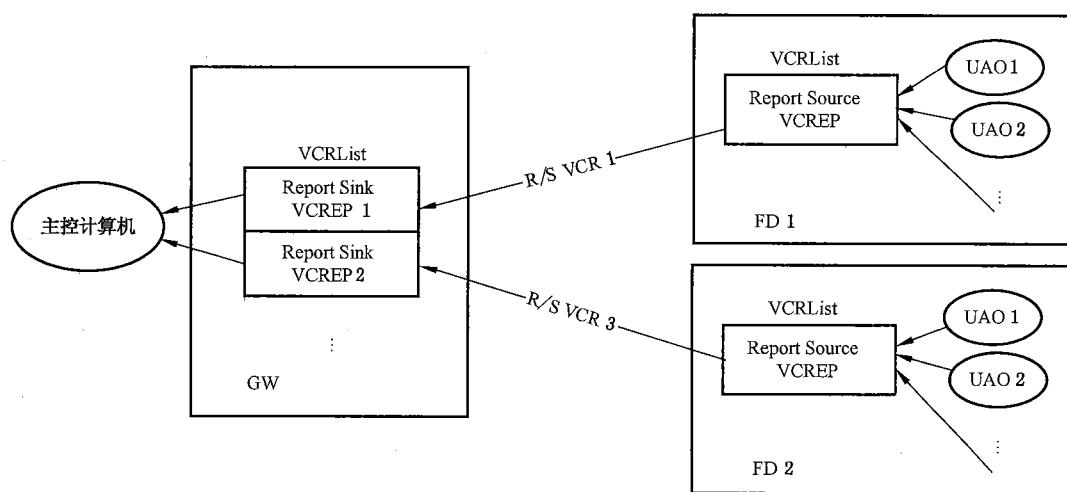
注：箭头指示数据传输方向。

图 77 现场设备与网关设备之间的 P/S VCR

#### 10.5.5.3.4 R/S VCR 组态

网关设备与现场设备之间应至少建立一个 R/S VCR。现场设备为报告源点，网关设备为报告汇点。

图 78 给出了网关设备与现场设备之间的 R/S VCR 关系。



注：箭头指示数据传输方向。

图 78 网关设备与现场设备之间的 R/S VCR

#### 10.5.5.4 组态过程

图 79 给出了网关设备对现场设备的组态过程序列。

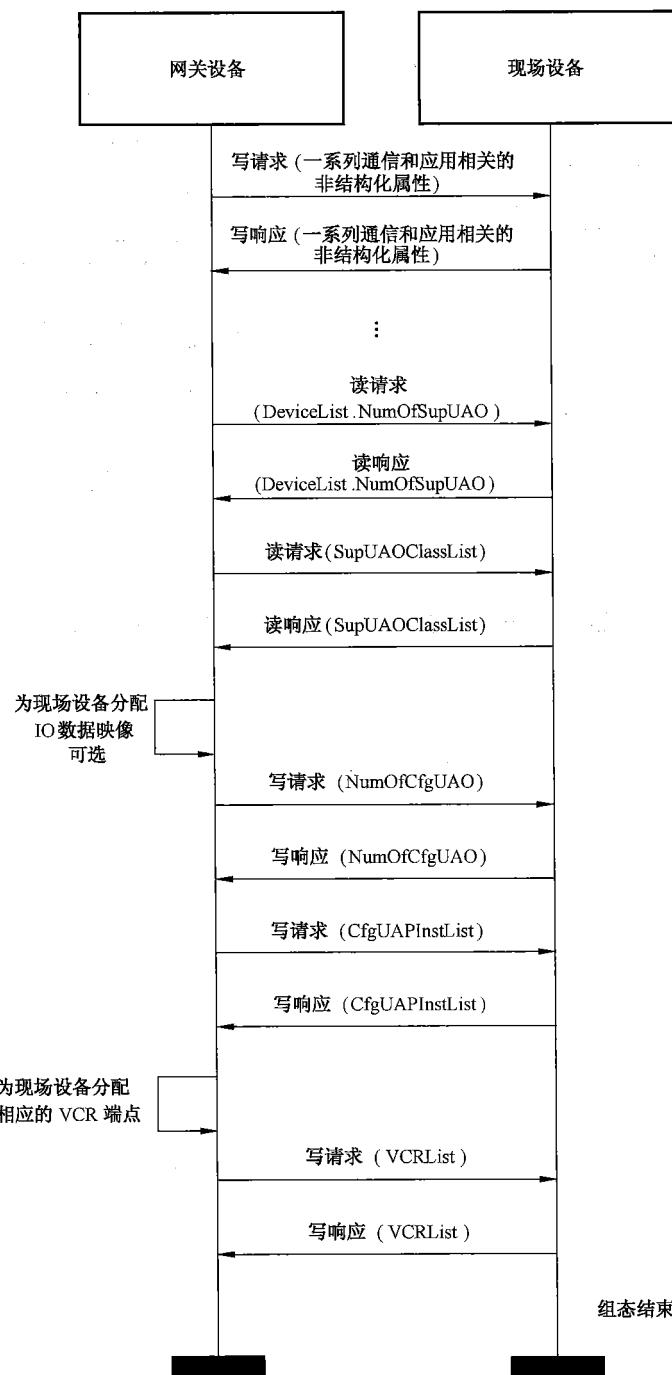


图 79 现场设备组态过程

#### 10.5.5.5 过程数据聚合与解聚

分配给同一 UAP 的所有 UAO 的数据在设备上应被聚合。作为发布者, UAP 应根据组态顺序依次获得所有 UAO 的过程数据, 并将这些过程数据形成一个 PUBLISH 请求报文。这就是应用层聚合过程。

作为预订者, UAP 应将接收到的 PUBLISH 请求报文进行解析, 并将解析出的过程数据依次传送给相应 UAO。这就是应用层解聚过程。

图 80 给出了一个 UAO 数据聚合和解聚过程的示例。

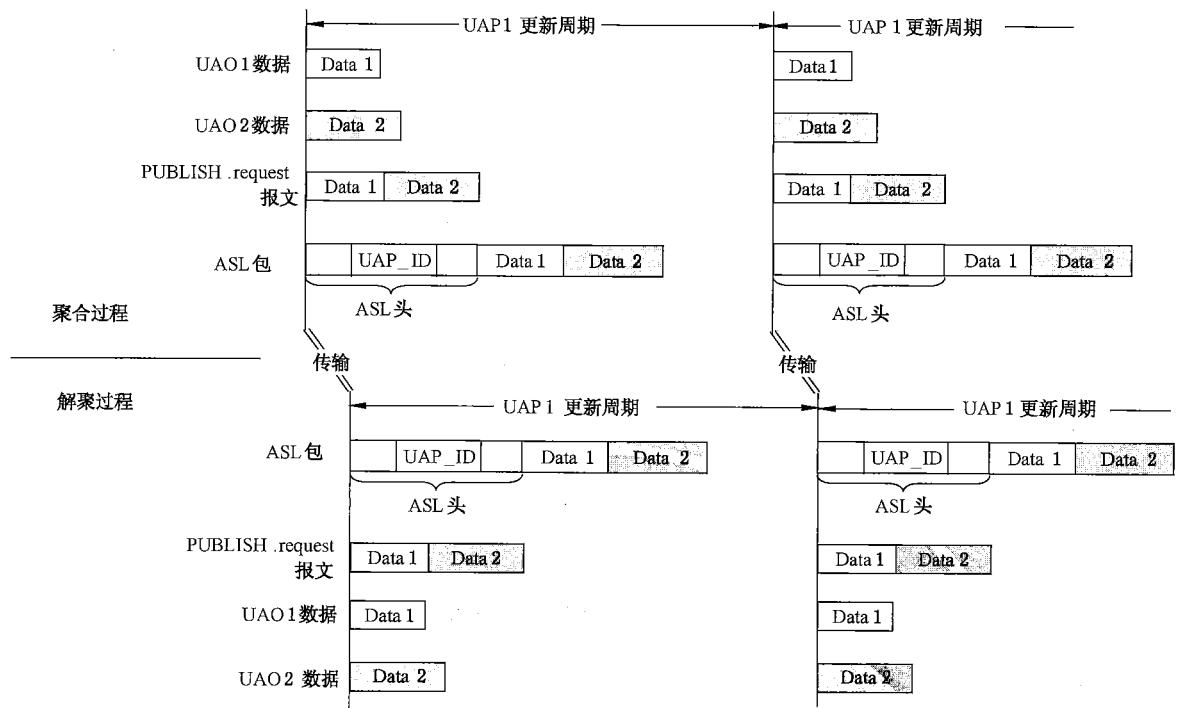


图 80 UAO 数据聚合和解聚过程

## 10.6 应用层服务

### 10.6.1 证实服务和非证实服务

WIA-FA 应用层定义了相应的应用服务来支持非周期性的读写访问、周期性过程数据发布,以及告警事件报告,如表 79 所示。

应用服务包括证实服务和非证实服务。证实服务于不同 UAP 之间双向的请求和响应传输,非证实服务于一个 UAP 到其他一个或多个 UAP 的单向数据传输。

表 79 UAP 支持的应用服务

服务名称	服务标识符	报文类型	描述
读(READ)	0x01	请求	请求读取某个 UAO 或 MIB 属性的值
		响应(+)	读 UAO 或 MIB 属性值成功,在响应中返回所请求的属性值
		响应(-)	读 UAO 或 MIB 属性值失败,在响应中返回失败原因
写(WRITE)	0x02	请求	请求写入某个 UAO 或 MIB 属性的值
		响应(+)	写 UAO 或 MIB 属性值成功
		响应(-)	写 UAO 或 MIB 属性值失败,在响应中返回失败原因
发布(PUBLISH)	0x03	请求	请求发布输入或输出过程数据值
报告(REPORT)	0x04	请求	请求报告一个或多个 UAO 事件
报告确认 (REPORT ACK)	0x05	请求	请求对收到的告警进行确认
		响应(+)	告警确认成功
		响应(-)	告警确认失败,在响应中返回失败原因

### 10.6.2 读(READ)服务

#### 10.6.2.1 报文格式

读请求报文的格式如图 81 所示。

1 八位位组	1 八位位组	2 八位位组	1 八位位组
UAO 标识符	属性标识符	存储索引	成员标识符

图 81 读请求报文格式

读正响应报文的格式如图 82 所示。

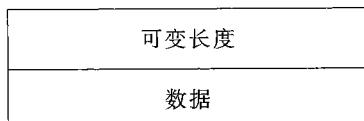


图 82 读正响应报文格式

读负响应报文的格式见图 83。

1 八位位组	1 八位位组
错误代码	附加信息

图 83 读负响应报文格式

读服务报文中各域说明如下：

- UAO 标识符：要读取 UAO 的 UAO\_ID；对于 MIB，UAO\_ID=0；
- 属性标识符：UAO 或 MIB 中某属性的 AttributeID；
- 存储索引：列表中某记录的索引，值 0xFFFF 指示要读取列表的所有记录，该域对于非结构化属性无效；
- 成员标识符：结构化属性某成员的 MemberID，值 0xFFFF 指示要读取结构化属性的所有成员，该域对于非结构化属性无效；
- 数据：读取的数据；
- 错误代码：失败原因代码，定义如表 80 所示；
- 附加信息：失败原因的附加信息，制造商特定。

表 80 读负响应报文的错误代码定义

取 值	定 义	含 义
1	SERVICE_EXPIRATION	服务超时
2	SERVICE_NOT_SUPPORTED	服务不被支持
3	UAO_NOT_EXISTENT	UAO 不存在
4	ATTRIBUTE_NOT_EXISTENT	属性不存在
5	STOREINDEX_NOT_EXISTENT	存储索引不存在

表 80 (续)

取 值	定 义	含 义
6	MEMBER_NOT_EXISTENT	成员不存在
7	LENGTH_TOO_LARGE <sup>a</sup>	长度太长 <sup>a</sup>
8	OTHERS	其他
9~255	Reserved for future use	保留供将来使用

<sup>a</sup> MaxPayLoadLength 描述 DLL 载荷的最大长度。当数据的长度大于 MaxPayLoadLength 时, 返回该错误代码。

### 10.6.2.2 服务规程

该证实服务通过 C/SVCR 传输 NRT 数据。网关设备使用该服务非周期地读取现场设备 UAO 或 MIB 的一个属性值或属性的某个成员值。当要读取的 UAP\_ID、AttributeID、MemberID 错误, 或者服务不被现场设备支持时, 应返回相应的错误代码。读服务规程如图 84 所示。

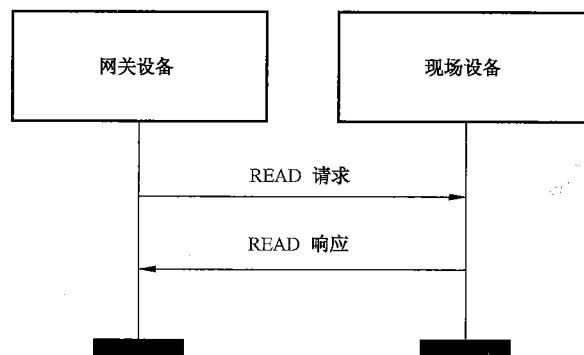


图 84 读服务规程

### 10.6.3 写(WRITE)服务

#### 10.6.3.1 报文格式

写请求报文的格式如图 85 所示。

1 八位位组	1 八位位组	2 八位位组	1 八位位组	可变长度
UAO 标识符	属性标识符	存储索引	成员标识符	数据

图 85 写请求报文格式

写正响应无报文格式。

写负响应报文的格式如图 86 所示。

1 八位位组	1 八位位组
错误代码	附加信息

图 86 写负响应报文格式

写服务报文中各域说明如下：

- UAO 标识符：要写入 UAO 的 UAO\_ID；对于 MIB，UAO\_ID=0；
- 属性标识符：UAO 或 MIB 中某属性的 AttributeID；
- 存储索引：列表中某记录的索引，值 0xFFFF 指示要写入列表的所有记录，该域对于非结构化属性无效；
- 成员标识符：结构化属性某成员的 MemberID，值 255 指示要写入结构化属性的所有成员，该域对于非结构化属性无效；
- 数据：写入的数据；
- 错误代码：失败原因代码，定义如表 81 所示；
- 附加信息：失败原因的附加信息，制造商特定。

表 81 写负响应错误代码定义

取 值	定 义	含 义
1	SERVICE_EXPIRATION	服务超时
2	SERVICE_NOT_SUPPORTED	服务不被支持
3	UAO_NOT_EXISTENT	UAO 不存在
4	ATTRIBUTE_NOT_EXISTENT	属性不存在
5	STOREINDEX_NOT_EXISTENT	存储索引不存在
6	MEMBER_NOT_EXISTENT	成员不存在
7	LENGTH_NOT_MATCH	长度不匹配
8	VALUE_EXCEED_SCOPE	值超出范围
9	OTHERS	其他
10~255	Reserved for future use	保留供将来使用

#### 10.6.3.2 服务规程

该证实服务通过 C/SVCR 传输 NRT 数据。网关设备使用该服务非周期地写入现场设备 UAO 或 MIB 的一个属性值或属性的某个成员值。当要写入的 UAP\_ID、AttributeID、MemberID 错误，或者值超出范围，或者服务不被现场设备支持时，应返回相应的错误代码。写服务规程如图 87 所示。

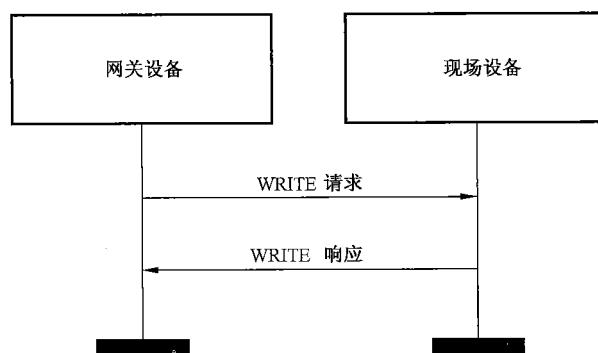


图 87 写服务规程

## 10.6.4 发布(PUBLISH)服务

### 10.6.4.1 报文格式

发布请求报文的格式如图 88 所示。

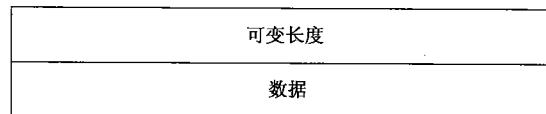


图 88 发布请求报文格式

发布服务是非证实服务，无响应报文。

发布请求报文仅包含数据(Data)域：UAP 要发布的所有输入数据或输出数据。

### 10.6.4.2 服务规程

该非证实服务通过 P/SVCR 传输 RT1 数据。现场设备或网关设备使用该服务以 DataUpdateRate 为周期来周期性发布过程数据。属于相同 UAP 的多个 UAO 的数据应聚合为一个 PUBLISH 服务进行发送。图 89 给出从现场设备到网关设备的发布规程，图 90 给出网关设备到现场设备的发布规程。

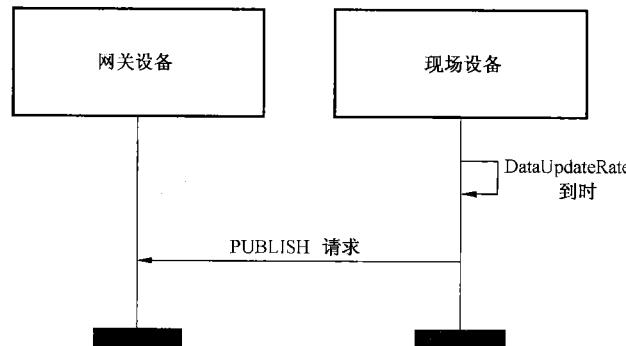


图 89 现场设备到网关设备的发布规程

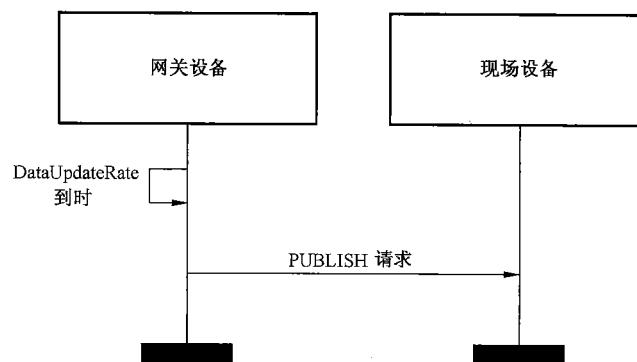


图 90 网关设备到现场设备的发布规程

## 10.6.5 报告(REPORT)服务

### 10.6.5.1 报文格式

报告请求报文的格式如图 91 所示。

1 八位位组	4 八位位组	1 八位位组
UAO 标识符	事件	附加信息

图 91 报告请求报文格式

报告服务是非证实服务,无响应报文。

报告服务报文中各域说明如下:

- UAO 标识符:报告告警事件的 UAO 的 UAO\_ID;
- 事件:报告的告警事件,EventData 结构体类型,定义如表 76 所示;
- 附加信息:制造商特定的附加信息。

#### 10.6.5.2 服务规程

该服务非证实服务通过 R/SVCR 传输 RT0 数据。现场设备应使用该服务向网关设备报告有一个或多个告警事件产生或消失。报告服务规程如图 92 所示。

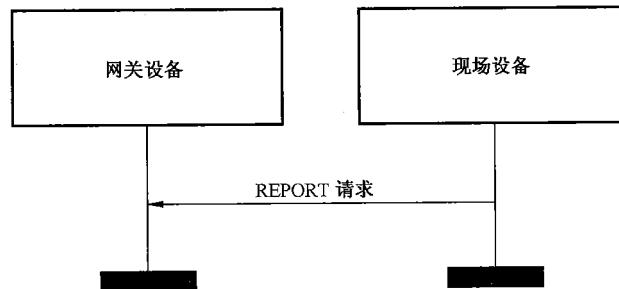


图 92 报告服务规程

#### 10.6.6 报告确认(REPORT ACK)服务

##### 10.6.6.1 报文格式

报告确认请求报文的格式如图 93 所示。

1 八位位组	2 八位位组
UAO 标识符	确认事件

图 93 报告确认请求报文格式

报告确认正响应报文的格式如图 94 所示。

2 八位位组
UAO 标识符

图 94 报告确认正响应报文格式

报告确认负响应报文的格式如图 95 所示。

1 八位位组	1 八位位组	1 八位位组
UAO 标识符	错误代码	附加信息

图 95 报告确认负响应报文格式

报告确认服务报文中各域说明如下：

- UAO 标识符：告警被确认的 UAO 的 UAO\_ID；
- 确认事件：被确认的事件。每位代表一个事件（见表 77 定义），值为 1 指示相应事件被确认；
- 错误代码：失败原因代码，定义如表 82 所示；
- 附加信息：制造商特定的附加信息。

表 82 报告确认负响应错误代码定义

取 值	定 义	含 义
1	SERVICE_EXPIRATION	服务超时
2	SERVICE_NOT_SUPPORTED	服务不被支持
3	UAO_NOT_EXISTENT	UAO 不存在
4	EVENT_NOT_EXISTENT	事件不存在
5	ACKNOWLEDGEMENT_NOT_REQUIRED	无需确认
6	OTHER	其他
7~255	Reserved for future use	保留供将来使用

#### 10.6.6.2 服务规程

该证实服务通过 C/SVCR 传输 NCR 数据。网关设备使用该服务对之前现场设备报告的告警事件进行部分或全部确认。报告确认服务规程如图 96 所示。

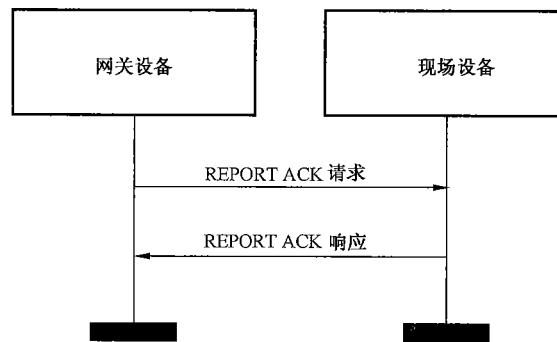


图 96 报告确认服务规程

### 10.7 应用子层

#### 10.7.1 概述

应用子层(ASL)为 UAP 与 DMAP 提供端到端的透明的数据传输服务。

### 10.7.2 ASL 数据服务

#### 10.7.2.1 概述

应用子层为 UAP 提供数据服务,以实现两个或多个 UAP 在 WIA-FA 网络上交换应用数据。

应用子层定义以下 4 种数据服务原语:

- 数据请求原语 ASLDE-DATA.request;
- 数据指示原语 ASLDE-DATA.indication;
- 数据响应原语 ASLDE-DATA.response;
- 数据证实原语 ASLDE-DATA.confirm。

#### 10.7.2.2 应用子层数据请求原语 ASLDE-DATA.request

UAP 调用 ASLDE-DATA.request 原语向 ASL 发送应用服务请求报文。根据 ASL 通用包格式(见 10.7.3.1),ASL 为请求报文增加 ASL 包头形成 APDU,并将其发送给 DLL。

应用子层数据请求原语 ASLDE-DATA.request 定义如下:

ASLDE-DATA.request(

```
DstAddr,  
ServiceID,  
Priority,  
AsduLength,  
Asdu
```

)

ASLDE-DATA.request 原语的参数说明如表 83 所示。

表 83 ASLDE-DATA.request 原语参数定义

参数名称	数据类型	取值范围	描述
DstAddr	Unsigned16	0~65 535	应用服务请求的目标地址
ServiceID	Unsigned8	0~5	应用服务的服务标识符,编码见表 79
UAP_ID	Unsigned8	0~255	UAP 在现场设备上的标识符
Priority	Unsigned8	0~255	应用数据的优先级,取值如下: 0=RT0; 1=RT1; 2=RT2; 4=NRT; 其余保留。优先级定义详见 6.4.2.1
AsduLength	Unsigned16	0~65 535	应用服务报文的长度
Asdu	Octetstring	—	应用服务报文

#### 10.7.2.3 应用子层数据指示原语 ASLDE-DATA.indication

ASL 从 DLL 接收到包含在 ASL 包中的应用服务请求时,调用 ASLDE-DATA.indication 原语来将请求报文传递给 UAP。

应用子层数据指示原语 ASLDE-DATA.indication 定义如下:

ASLDE-DATA.indication(

    ServiceID,  
    UAP\_ID,  
    AsduLength,  
    Asdu  
)

ASLDE-DATA.indication 原语的参数说明如表 84 所示。

表 84 ASLDE-DATA.indication 原语参数定义

参数名称	数据类型	取值范围	描述
ServiceID	Unsigned8	0~255	应用服务的服务标识符,编码见表 79
UAP_ID	Unsigned8	0~255	UAP 在现场设备上的标识符
AsduLength	Unsigned16	0~65 535	应用服务报文的长度
Asdu	Octetstring	—	应用服务报文

#### 10.7.2.4 应用子层数据响应原语 ASLDE-DATA.response

UAP 调用 ASLDE-DATA.response 原语向 ASL 发送应用证实服务的响应报文。根据 ASL 通用包格式(见 10.7.3.1),ASL 为响应报文增加 ASL 包头形成 APDU,并将其发送给 DLL。

应用子层数据响应原语 ASLDE-DATA.response 定义如下:

ASLDE-DATA.response(

    ServiceID,  
    MsgType,  
    UAP\_ID,  
    AsduLength,  
    Asdu  
)

ASLDE-DATA.response 原语的参数说明如表 85 所示。

表 85 ASLDE-DATA.response 原语参数定义

参数名称	数据类型	取值范围	描述
ServiceID	Unsigned8	0~255	应用服务的服务标识符,编码见表 79
MsgType	Unsigned8	0~255	应用服务的报文类型,取值如下: 0=REQUEST; 1=RESPONSE_P; 2=RESPONSE_N; 其余保留
UAP_ID	Unsigned8	0~255	UAP 在现场设备上的标识符
AsduLength	Unsigned16	0~65 535	应用服务报文的长度
Asdu	Octetstring	—	应用服务报文

### 10.7.2.5 应用子层数据证实原语 ASLDE-DATA.confirm

ASL 从 DLL 接收到包含在 ASL 包中的证实应用服务响应时, 调用 ASLDE-DATA.confirm 原语来将响应报文传递给 UAP。

应用子层数据证实原语 ASLDE-DATA.confirm 定义如下:

ASLDE-DATA.confirm(

```
SrcAddr,
ServiceID,
MsgType,
UAP_ID,
AsduLength,
Asdu
```

)

ASLDE-DATA.confirm 原语的参数说明如表 86 所示。

表 86 ASLDE-DATA.confirm 原语参数

参数名称	数据类型	取值范围	描述
SrcAddr	Unsigned16	0~65 535	应用服务响应的源地址
ServiceID	Unsigned8	0~255	应用服务的服务标识符, 编码见表 79
MsgType	Unsigned8	0~255	应用服务响应的报文类型, 取值如下: 1=RESPONSE_P; 2=RESPONSE_N; 其余保留
UAP_ID	Unsigned8	0~255	UAP 在现场设备上的标识符
AsduLength	Unsigned16	0~65 535	应用服务报文的长度
Asdu	Octetstring	—	应用服务报文

### 10.7.3 应用子层包格式

#### 10.7.3.1 通用包格式

每个应用子层包由以下两个部分组成:

- 应用子层包头(ASL Header), 包含包控制、UAP\_ID, 以及载荷长度域;
- 应用子层载荷(ASL Payload), 长度可变。

应用子层通用包格式如图 97 所示。

ASL 包头			ASL 载荷
包控制	UAP 标识符	载荷长度	载荷
1 八位位组	1 八位位组	2 八位位组	可变长度

图 97 应用子层通用包格式

### 10.7.3.2 ASL 包头

#### 10.7.3.2.1 包控制域

##### 10.7.3.2.1.1 概述

包控制域的数据类型为 Unsigned8, 长度为 1 八位位组(8 比特), 包括服务标识符、报文类型子域, 如图 98 所示。

位:0~2	位:3~5	位:6~7
服务标识符	保留	报文类型

图 98 包控制域定义

##### 10.7.3.2.1.2 服务标识符子域

服务标识符子域长度为 3 比特, 指示 AL 服务类型, 定义如表 87 所示。

表 87 服务标识符子域定义

位:0~2	定义
0b001	READ
0b010	WRITE
0b011	PUBLISH
0b100	REPORT
0b101	REPORT ACK

##### 10.7.3.2.1.3 报文类型子域

报文类型子域长度为 2 比特, 指示 AL 报文类型, 定义如表 88 所示。

表 88 报文类型子域定义

位:6~7	定义
0b00	REQUEST
0b01	RESPONSE_P
0b10	RESPONSE_N
0b11	Reserved

### 10.7.3.2.2 UAP 标识符域

UAP 标识符域的数据类型为 Unsigned8, 长度为 1 八位位组, 其值应为 UAP 在现场设备上的 UAP\_ID, 值 0 用于 DMAP。

#### 10.7.3.2.3 载荷长度域

载荷长度域的数据类型为 Unsigned16, 长度为 2 八位位组, 其值指示 ASL 载荷的八位位组长度, 不包括 ASL 包头。

### 10.7.3.3 ASL 载荷域

载荷域包含应用层服务报文,可变长度。不同服务定义了不同的报文格式,详见 10.6.2~10.6.6。

### 10.7.4 应用子层状态机

#### 10.7.4.1 概述

按照 VCR 端点类型不同,应用子层状态机(ASLM)可分为客户机状态机(AMCL)、服务器状态机(AMSV)、发布者状态机(AMPB)、预订者状态机(AMSB)、报告源状态机(AMRS),和报告汇状态机(AMRK)。

#### 10.7.4.2 ASL 与 UAP、DLL 交换的服务原语

##### 10.7.4.2.1 证实服务原语

图 99 和表 89 给出了 ASL 与一般 UAP、DLL 之间交换的证实服务原语。

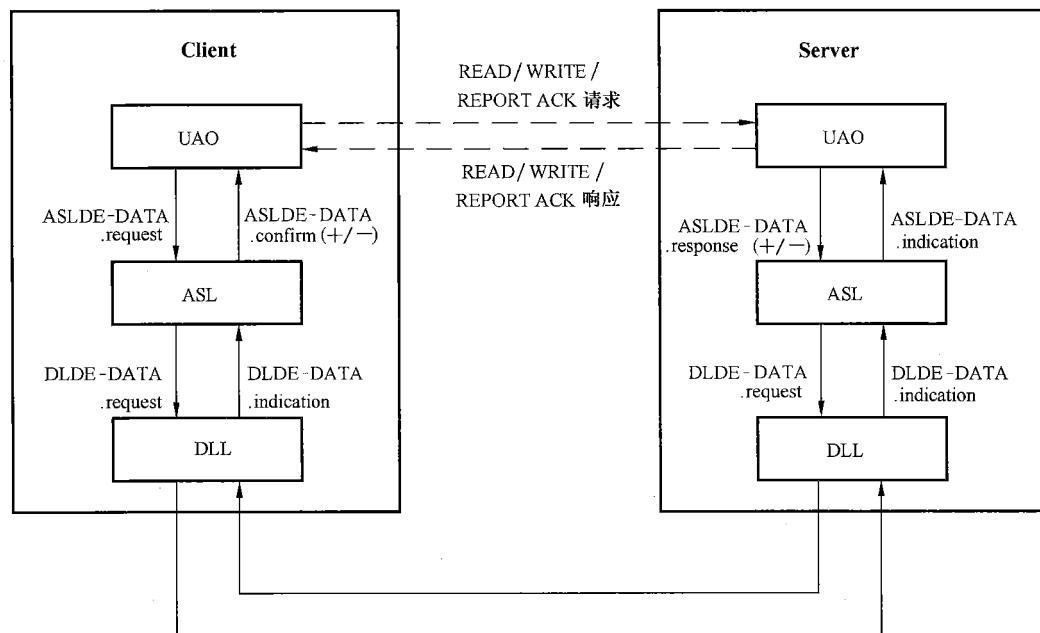


图 99 各层之间交换的证实服务原语

表 89 ASL 与其他层之间交换的证实服务原语

原语	来源	参数
ASLDE-DATA.request	UAP	DstAddr, ServiceID, UAP_ID, Priority, AsduLength, Asdu
ASLDE-DATA.indication	ASL	ServiceID, UAP_ID, AsduLength, Asdu
ASLDE-DATA.response	UAP	ServiceID, MsgType, UAP_ID, AsduLength, Asdu
ASLDE-DATA.confirm	ASL	SrcAddr, ServiceID, MsgType, UAP_ID, AsduLength, Asdu
DLDE-DATA.request	ASL	DstAddr, DataType, Priority, PayloadLength, Payload
DLDE-DATA.indication	DLL	SrcAddr, DataType, PayloadLength, Payload

#### 10.7.4.2.2 非证实服务原语

图 100 和表 90 给出了 ASL 与一般 UAP、DLL 之间交换的非证实服务原语。

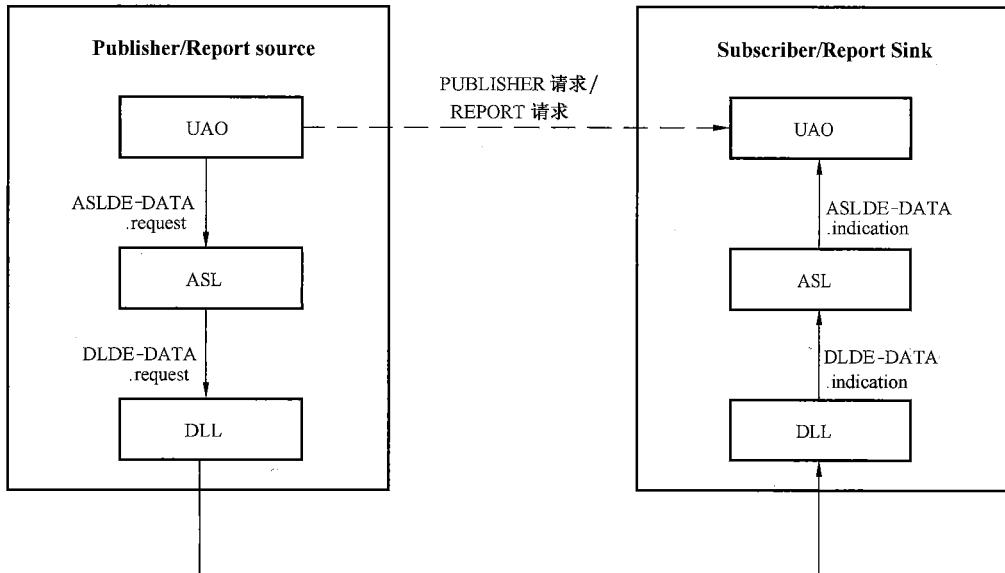


图 100 各层之间交换的非证实服务原语

表 90 ASL 与其他层之间交换的非证实服务原语

原语	来源	参数
ASLDE-DATA.request	UAP	DstAddr, ServiceID, UAP_ID, Priority, AsduLength, Asdu
ASLDE-DATA.indication	ASL	ServiceID, UAP_ID, AsduLength, Asdu
DLDE-DATA.request	ASL	DstAddr, DataType, Priority, PayloadLength, Payload
DLDE-DATA.indication	DLL	SrcAddr, DataType, PayloadLength, Payload

#### 10.7.4.3 ASL 与 DMAP、DLL 交换的服务原语

当网关设备读或写现场设备的 MIB 属性时, ASL 与 DMAP、DLL 之间交换的服务原语如图 101 和表 91 所示。

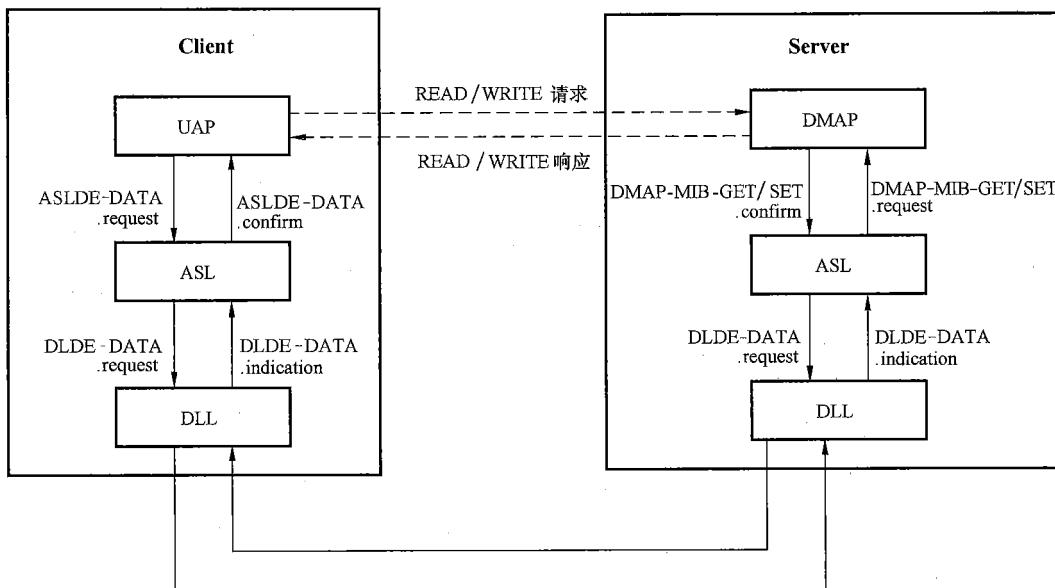


图 101 读写 MIB 时各层之间交换的服务原语

表 91 读写 MIB 时各层之间交换的服务原语

Primitives	Source	Parameters
ASLDE-DATA.request	UAP	DstAddr, ServiceID, UAP_ID, Priority, AsduLength, Asdu
ASLDE-DATA.confirm	ASL	SrcAddr, ServiceID, MsgType, UAP_ID, AsduLength, Asdu
DMAP-MIB-GET.request	ASL	Handle, ShortAddress AttributeID, MemberID, FirstStoreIndex, Count
DMAP-MIB-SET.request	ASL	Handle, ShortAddress AttributeID, MemberID, FirstStoreIndex, Count, AttributeValue
DMAP-MIB-GET.confirm	DMAP	Handle, Status, Count, AttributeValue
DMAP-MIB-SET.confirm	DMAP	Handle, Status
DLDE-DATA.request	ASL	DstAddr, DataType, Priority, PayloadLength, Payload
DLDE-DATA.indication	DLL	SrcAddr, DataType, PayloadLength, Payload

#### 10.7.4.4 客户机状态机

客户机状态机(AMCL)具有以下状态。

- Idle: 客户机 VCR 端点的初始状态和空闲状态。在该状态,VCR 端点等待由 ASLDE-DATA.request 原语传递的证实服务请求报文。当接收到 ASLDE-DATA.request 原语后,VCR 端点应将请求报文封装为 ASL 包,并调用 DLDE-DATA.req 原语将 ASL 包发送给 DLL,然后进入 Wait\_Cnf 状态以等待相应的服务响应。
- Wait\_Cnf: 在该状态,客户机 VCR 端点等待现场设备返回的 AL 服务响应,应执行以下状态转换之一:
  - 如果在 Watchdog 时间内接收到由 DLDE-DATA.indication 原语传递的包含证实服务请求报文的 ASL 包,则 VCR 端点应解析 ASL 包,并调用 ASLDE-DATA.confirm 原语将解析出的证实服务响应报文传递给 UAP,然后进入 Idle 状态;

- 如果在 Watchdog 时间内未接收到服务响应，则 VCR 端点应调用 ASLDE-DATA.confirm 原语向 UAP 返回“服务超时”的负响应报文，然后进入 Idle 状态。

Watchdog 时间通过 Watchdog 定时器监视，其值由 VCR 端点的 WatchdogTime 属性规定。

客户机状态机的状态转换如图 102 和表 92 所示。

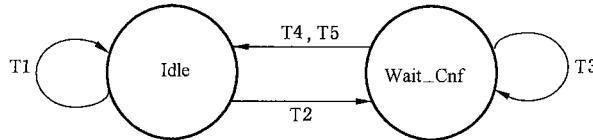


图 102 客户机状态机 AMCL 状态转换图

表 92 客户机状态机 AMCL 状态转换表

编号	当前状态	事件\条件 => 动作	下一状态
T1	Idle	(ASLDE-DATA.request() && ServiceID == (READ    WRITE    REPORT ACK))    ASLDE-DATA.response()    DLDE-DATA.indication() => Ignore;	Idle
T2	Idle	ASLDE-DATA.request() && ServiceID == (READ    WRITE    REPORT ACK) => VCR_ID := GetVcrID(DstAddr, CLIENT, UAP_ID); StartWatchdogTimer(VCR_ID); StoreSvrID(VCR_ID, ServiceID); MsgType := REQUEST; DLDE-DATA.request( DstAddr, VCR_ID, DataType := DATA, Priority, PayloadLength := AsduLength + 4, Payload := BuildAPDU(ServiceID, MsgType, UAP_ID, AsduLength, Asdu) ) ;	Wait_Cnf
T3	Wait_Cnf	ASLDE-DATA.request()    ASLDE-DATA.response()    (DLDE-DATA.indication() && (DataType != DATA    TakeServiceID(Payload) != (READ    WRITE    REPORT ACK))    TakeMsgType(Payload) != (RESPONSE_P    RESPONSE_N)) => Ignore;	Wait_Cnf
T4	Wait_Cnf	DLDE-DATA.indication() && DataType == DATA && TakeServiceID(Payload) == (READ    WRITE    REPORT ACK) && TakeMsgType(Payload) == (RESPONSE_P    RESPONSE_N) => UAP_ID := TakeUAPID(Payload); VCR_ID := GetVcrID(SrcAddr, CLIENT, UAP_ID);	Idle

表 92 (续)

编号	当前状态	事件\条件 => 动作	下一状态
T4	Wait_Cnf	<pre> StopWatchdogTimer(VCR_ID); ASLDE-DATA.confirm(     SrcAddr,     ServiceID:=TakeServiceID(Payload),     MsgType:=TakeMsgType(Payload),     UAP_ID,     AsduLength:=PayloadLength - 4,     Asdu:=TakeASLPayload(PayloadLength, Payload) ); </pre>	Idle
T5	Wait_Cnf	<pre> Watchdog Timer with VCR_ID expires =&gt; ASLDE-DATA.confirm(     SrcAddr:=GetPeerAddr(VCR_ID),     ServiceID:=RestoreSrvID(VCR_ID),     MsgType := RESPONSE_N,     UAP ID:=GetUAPID(VCR_ID),     AsduLength := 2,     Asdu:=BuildErrAsdu(SERVICE_EXPIRATION, 0) ); </pre>	Idle

#### 10.7.4.5 服务器状态机

服务器状态机(AMSV)具有以下状态。

——Idle: 服务器 VCR 端点的初始状态和空闲状态。在该状态, VCR 端点等待由 DLDE-DATA.indication 传递的包含证实服务请求报文的 ASL 包。当接收到 DLDE-DATA.indication 原语后, VCR 端点应解析 ASL 包, 并调用 ASLDE-DATA.indication 原语将解析出的请求报文发给 UAP, 然后进入 Wait\_Rsp 状态。

——Wait\_Rsp: 在该状态, 服务器 VCR 端点等待 UAP 返回 AL 服务响应报文, 应执行以下状态转移之一:

- 如果在 Watchdog 时间内接收到由 ASLDE-DATA.response 原语传递的服务响应报文, 则 VCR 端点应将响应报文封装为 ASL 包, 并调用 DLDE-DATA.req 原语将 ASL 包发送给 DLL, 然后进入 Idle 状态;
- 如果在 Watchdog 时间内未接收到响应报文, 则 VCR 端点应返回“服务超时”的负响应报文, 然后进入 Idle 状态。

Watchdog 时间通过 Watchdog 定时器监视, 其值由 VCR 端点的 WatchdogTime 属性规定。

服务器状态机的状态转换如图 103 和表 93 所示。

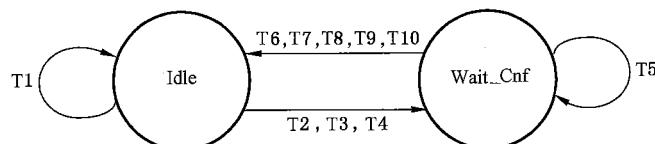


图 103 服务器状态机 AMSV 状态转换图

表 93 服务器状态机 AMSV 状态转换表

编号	当前状态	事件\条件 => 动作	下一状态
T1	Idle	<pre> ASLDE-DATA.request()    ASLDE-DATA.response()   (DLDE-DATA.indication() &amp;&amp;    (DataType != DATA    TakeMsgType(Payload) != REQUEST        TakeServiceID(Payload) != (READ    WRITE    REPORT ACK))) =&gt; Ignore; </pre>	Idle
T2	Idle	<pre> DLDE-DATA.indication() &amp;&amp; DataType == DATA &amp;&amp; TakeServiceID(Payload) == (READ    WRITE    REPORT ACK) &amp;&amp; TakeMsgType(Payload) == REQUEST &amp;&amp; TakeUAPID(Payload) != 0 =&gt; ServiceID := TakeServiceID(Payload); UAP_ID := TakeUAPID(Payload); VCR_ID := GetVcrID(SrcAddr, SERVER, UAP_ID); StartWatchdogTimer(VCR_ID); StoreSvrID(VCR_ID, ServiceID); ASLDE-DATA.indication(   ServiceID,   UAP_ID,   AsduLength := PayloadLength - 4,   Asdu := TakeASLPayload(PayloadLength, Payload) ); </pre>	Wait_Rsp
T3	Idle	<pre> DLDE-DATA.indication() &amp;&amp; DataType == DATA &amp;&amp; TakeServiceID(Payload) == READ &amp;&amp; TakeMsgType(Payload) == REQUEST &amp;&amp; TakeUAPID(Payload) == 0 =&gt; ServiceID := TakeServiceID(Payload); VCR_ID := GetVcrID(0, SERVER, 0); StartWatchdogTimer(VCR_ID); StoreSvrID(VCR_ID, ServiceID); DMAP-MIB-GET.request(   Handle,   AttributeID,   MemberID,   FirstStoreIndex,   Count ); </pre>	Wait_Rsp

表 93 (续)

编号	当前状态	事件\条件 => 动作	下一状态
T4	Idle	<pre> DLDE-DATA.indication() &amp;&amp; DataType == DATA &amp;&amp; TakeServiceID(Payload) == WRITE &amp;&amp; TakeMsgType(Payload) == REQUEST &amp;&amp; TakeUAPID(Payload) == 0 =&gt; ServiceID:=TakeServiceID(Payload); VCR_ID:=GetVcrID(0, SERVER, 0); StartWatchdogTimer(VCR_ID); StoreSrvID(VCR_ID, ServiceID); DMAP-MIB-SET.request(     Handle,     AttributeID,     MemberID,     FirstStoreIndex,     Count,     AttributeValue ); </pre>	Wait_Rsp
T5	Wait_Rsp	<pre> (ASLDE-DATA.response() &amp;&amp; ServiceID! =(READ    WRITE    REPORT ACK))    ASLDE-DATA.request()    DLDE-DATA.indication() =&gt; Ignore; </pre>	Wait_Rsp
T6	Wait_Rsp	<pre> ASLDE-DATA.response() &amp;&amp; ServiceID == (READ    WRITE    REPORT ACK) =&gt; VCR_ID:=GetVcrID(0, SERVER, VAP_ID); StopWatchdogTimer(VCR_ID); DLDE -DATA.request(     DstAddr:=GetPeerAddr(VCR_ID),     VCR_ID,     DataType:=DATA,     Priority:=NRT,     PayloadLength:=AsduLength + 4,     Payload:=BuildAPDU(ServiceID, MsgType, UAP_ID, AsduLength , Asdu) ); </pre>	Idle
T7	Wait_Rsp	<pre> DMAP-MIB-GET.confirm() &amp;&amp; Status == 0 =&gt; VCR:=GetVcrID(0, SERVER, 0); StopWatchdogTimer(VCR_ID); ServiceID:=READ; MsgType:=RESPONSE_P; </pre>	Idle

表 93 (续)

编号	当前状态	事件\条件 => 动作	下一状态
T7	Wait_Rsp	<pre> UAP_ID:=0; AsduLength:=Sizeof(AttributeValue); Asdu:=AttributeValue; DLDE-DATA.request(     DstAddr :=GetPeerAddr(VCR_ID),     VCR_ID,     DataType:=DATA,     Priority:=NRT,     PayloadLength:=AsduLength + 4,     Payload:=BuildAPDU(ServiceID, MsgType, UAP_ID, AsduLength , Asdu) ); </pre>	Idle
T8	Wait_Rsp	<pre> DMAP-MIB-SET.confirm() &amp;&amp; Status ==0 =&gt; VCR:=GetVcrID(0, SERVER, 0); StopWatchdogTimer(VCR_ID); ServiceID:=WRITE; MsgType:=RESPONSE_P; UAP_ID:=0; DLDE-DATA.request(     DstAddr :=GetPeerAddr(VCR_ID),     VCR_ID,     DataType:=DATA,     Priority:=NRT,     PayloadLength:=4,     Payload:=BuildAPDU( ServiceID, MsgType, UAP_ID, 0 , NULL) ); </pre>	Idle
T9	Wait_Rsp	<pre> (DMAP-MIB-GET.confirm()    DMAP-MIB-SET.confirm()) &amp;&amp; Status !=0 =&gt; VCR_ID:=GetVcrID(0, SERVER, 0); StopWatchdogTimer(VCR_ID); ServiceID:=RestoreSvrID(VCR_ID); UAP_ID:=0; MsgType:=RESPONSE_N; Asdu:=BuildErrAsdu(Status, 0); AsduLength:=2; DLDE-DATA.request(     DstAddr :=GetPeerAddr(VCR_ID),     VCR_ID,     DataType:=DATA,     Priority:=NRT,     PayloadLength:=AsduLength + 4,     Payload:=BuildAPDU(ServiceID, MsgType, UAP_ID, AsduLength, Asdu) ); </pre>	Idle

表 93 (续)

编号	当前状态	事件\条件 => 动作	下一状态
T10	Wait_Rsp	<pre> Watchdog Timer with VCR_ID expires =&gt; ServiceID := RestoreSrvID(VCR_ID); MsgType := RESPONSE_N; Asdu := BuildErrAsdu(SERVICE_EXPIRATION, 0); AsduLength := 2; DLDE-DATA.request(     DstAddr := GetPeerAddr(VCR_ID),     VCR_ID,     DataType := DATA,     Priority := NRT,     PayloadLength := AsduLength + 4,     Payload := BuildAPDU(ServiceID, MsgType, 0, AsduLength, Asdu) ); </pre>	Idle

#### 10.7.4.6 发布者状态机

发布者状态机(AMPB)具有以下状态。

——Init:发布者 VCR 端点的初始状态。在完成组态后,VCR 端点应执行以下状态转换之一:

- 如果 VCRActiveTime 的值为 0,则应进入 Active 状态;
- 如果 VCRActiveTime 的值不为 0,则应进入 NO\_Active 状态。

——No\_Active:在该状态,发布者 VCR 端点已完成组态但尚未被激活。当 VCRActiveTime 到时,则 VCR 端点进入 Active 状态;

——Active:发布者 VCR 端点的激活状态。在该状态,VCR 端点等待由 ASLDE-DATA.request 原语传递的 PUBLISHER 请求报文,应执行以下状态转换之一:

- 如果接收到 PUBLISH 请求报文,则应将报文缓存在缓冲区中,然后将请求报文封装成 ASL 包,并调用 DLDE-DATA.request 原语将 ASL 包发送给 DLL,保持在 Active 状态;
- 如果 DataUpdateRate 到时,应从缓冲区中取出数据并将其封装成 ASL 包,调用 DLDE-DATA.request 原语将 ASL 包发送给 DLL,保持在 Active 状态。

VCRActiveTime 时间通过 VCRActiveTime 定时器监视,其值由 VCR 端点的 VCRActiveTime 属性规定。

DataUpdateRate 时间通过 DataUpdateRate 定时器监视,其值由 VCR 端点的 DataUpdateRate 属性规定。

发布者状态机的状态转换如图 104 和表 94 所示。

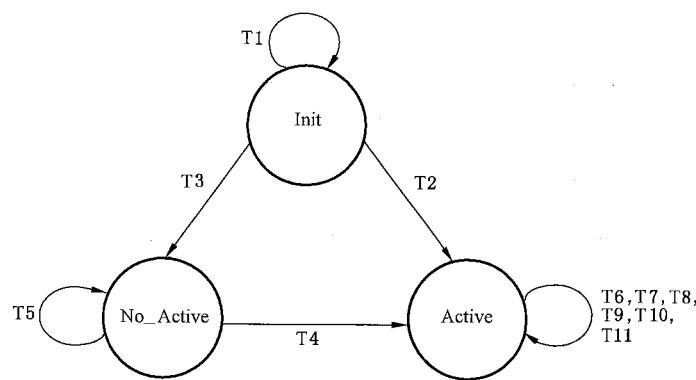


图 104 发布者状态机 AMPB 状态转换图

表 94 发布者状态机 AMPB 状态转换表

编号	当前状态	事件\条件 => 动作	下一状态
T1	Init	ASLDE-DATA.request()    ASLDE-DATA.response()    DLDE-DATA.indication() => Ignore;	Init
T2	Init	VCR with VCR_ID configured completely && VCRActiveTime == 0 => CreateBuffer(VCR_ID); StartDataUpdateRateTimer(VCR_ID); StartDeadlineTimer(VCR_ID);	Active
T3	Init	VCR with VCR_ID configured completely && VCRActiveTime != 0 => CreateBuffer(VCR_ID); StartActiveTimer(VCR_ID);	No_Active
T4	No_Active	VCRActiveTime with VCR_ID expires => StartDataUpdateRateTimer(VCR_ID); StartDeadlineTimer(VCR_ID);	Active
T5	No_Active	ASLDE-DATA.request()    ASLDE-DATA.response()    DLDE-DATA.indication() => Ignore;	No_Active
T6	Active	(ASLDE-DATA.request() && ServiceID != PUBLISH)    ASLDE-DATA.response()    DLDE-DATA.indication() => Ignore;	Active

表 94 (续)

编号	当前状态	事件\条件 => 动作	下一状态
T7	Active	<pre>(ASLDE-DATA.request() &amp;&amp; ServiceID == PUBLISH) &amp;&amp; CheckEvent(UAP_ID), PROCESS_DATA_NOT_UPDATED) =&gt; VCR_ID:=GetVcrID(DstAddr, PUBLISHER, UAP_ID); PutDataIntoBuffer(VCR_ID, AsduLength, Asdu); ServiceID:=PUBLISH; MsgType:=REQUEST; DLDE-DATA.request(     DstAddr :=GetPeerAddr(VCR_ID),     VCR_ID,     DataType:=DATA,     Priority:=RT1,     PayloadLength:=AsduLength + 4,     Payload:=BuildAPDU(ServiceID, MsgType, UAP_ID, AsduLength, Asdu) ); StartDataUpdateRateTimer(VCR_ID); StartDeadlineTimer(VCR_ID); SetEvent(UAP_ID, PROCESS_DATA_NOT_UPDATED, DISAPPEAR);</pre>	Active
T8	Active	<pre>(ASLDE-DATA.request() &amp;&amp; ServiceID == PUBLISH) &amp;&amp; ! CheckEvent(UAP_ID), PROCESS_DATA_NOT_UPDATED) =&gt; VCR_ID:=GetVcrID(DstAddr, PUBLISHER, UAP_ID); PutDataIntoBuffer(VCR_ID, AsduLength, Asdu); ServiceID:=PUBLISH; MsgType:=REQUEST; DLDE-DATA.request(     DstAddr :=GetPeerAddr(VCR_ID),     VCR_ID,     DataType:=DATA,     Priority:=RT1,     PayloadLength:=AsduLength + 4,     Payload:=BuildAPDU(ServiceID, MsgType, UAP_ID, AsduLength, Asdu) ); StartDataUpdateRateTimer(VCR_ID); StartDeadlineTimer(VCR_ID);</pre>	Active
T9	Active	<pre>DataUpdateRate timer with VCR_ID expires =&gt; UAP_ID:=GetUAPID(VCR_ID); ServiceID:=PUBLISH; MsgType:=REQUEST; Asdu:=GetDataFromBuffer(VCR_ID); AsduLength:=Sizeof(Asdu);</pre>	Active

表 94 (续)

编号	当前状态	事件\条件 => 动作	下一状态
T9	Active	<pre> DLDE-DATA.request(     DstAddr := GetPeerAddr(VCR_ID),     VCR_ID,     DataType:=DATA,     Priority:=RT1,     PayloadLength:=AsduLength + 4,     Payload:=BuildAPDU(ServiceID, MsgType, UAP_ID, AsduLength, Asdu) ); StartDataUpdateRateTimer(VCR_ID); </pre>	Active
T10	Active	<pre> Deadline timer with VCR_ID expires &amp;&amp; ! CheckEvent(GetUAPID(VCR_ID), PROCESS_DATA_NOT_UPDATED) =&gt; UAP_ID:=GetUAPID(VCR_ID); SetEvent(UAP_ID, PROCESS_DATA_NOT_UPDATED, APPEAR); StartDeadlineTimer(VCR_ID); </pre>	Active
T11	Active	<pre> Deadline timer with VCR_ID expires &amp;&amp; CheckEvent(GetUAPID(VCR_ID), PROCESS_DATA_NOT_UPDATED) =&gt; StartDeadlineTimer(VCR_ID); </pre>	Active

#### 10.7.4.7 预订者状态机

预订者状态机 AMSB 具有以下状态。

——Init: 预订者 VCR 端点的初始状态。在完成组态后, VCR 端点应执行以下状态转换之一:

- 如果 VCRActiveTime 的值为 0, 则应进入 Active 状态;
- 如果 VCRActiveTime 的值不为 0, 则应进入 NO\_Active 状态。

——No\_Active: 在该状态, 预订者 VCR 端点已完成组态但尚未被激活。当 VCRActiveTime 到时, 则 VCR 端点进入 Active 状态;

——Active: 预订者 VCR 端点的激活状态。在该状态, VCR 端点等待由 DLDE-DATA.indication 原语传递的包含 PUBLISH 请求报文的 ASL 包, 应执行以下状态转换之一:

- 如果接收到 PUBLISH 请求报文, 则应将请求中的数据缓存在缓冲区中, 然后调用 ASLDE-DATA.indication 原语将数据发送给 UAP; VCR 端点应检查属于 UAP 的所有 UAO 的事件数据 EventData, 如果事件数据的“过程数据未更新(PROCESS\_DATA\_NO\_UPDATED)”比特被置为 1, 则应将其复位为 0, 以产生“过程数据未更新”事件消失的告警; 保持处于 Active 状态;
- 如果 DataUpdateRate 到时, 应从缓冲区中取出数据, 调用 ASLDE-DATA.indication 原语将数据发送给 UAP; 保持处于 Active 状态。在此情况下, 数据为上一次缓存在缓冲区中的 PUBLISH 请求报文;
- 如果 Deadline 到时, 则应检查属于 UAP 的所有 UAO 的事件数据 EventData。如果事件数据的“过程数据未更新(PROCESS\_DATA\_NO\_UPDATED)”比特值为 0, 则应将其置位为 1, 以产生“过程数据未更新”事件发生的告警。保持处于 Active 状态。

VCRActiveTime 时间通过 VCRActiveTime 定时器监视, 其值由 VCR 端点的 VCRActiveTime 属性规定。

DataUpdateRate 时间通过 DataUpdateRate 定时器监视, 其值由 VCR 端点的 DataUpdateRate 属性规定。

Deadline 时间通过 Deadline 定时器监视, 其值由 VCR 端点的 Deadline 属性规定。

预定者状态机的状态转换如图 105 和表 95 所示。

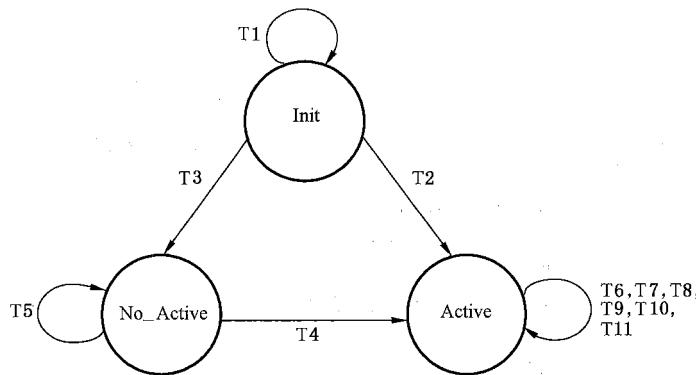


图 105 预订者状态机 AMSB 状态转换图

表 95 预订者状态机 AMSB 状态转换表

编号	当前状态	事件\条件 => 动作	下一状态
T1	Init	ASLDE-DATA.request()    ASLDE-DATA.response()    DLDE-DATA.indication() => Ignore;	Init
T2	Init	VCR with VCR_ID configured completely && VCRActiveTime == 0 => CreatBuffer(VCR_ID); StartDataUpdateRateTimer(VCR_ID); StartDeadlineTimer(VCR_ID);	Active
T3	Init	VCR with VCR_ID configured completely && VCRActiveTime != 0 => CreatBuffer(VCR_ID); StartActiveTimer(VCR_ID);	No_Active
T4	No_Active	VCRActiveTime with VCR_ID expires => StartDataUpdateRateTimer(VCR_ID); StartDeadlineTimer(VCR_ID);	Active
T5	No_Active	ASLDE-DATA.request()    ASLDE-DATA.response()    DLDE-DATA.indication() => Ignore;	No_Active

表 95 (续)

编号	当前状态	事件\条件 => 动作	下一状态
T6	Active	<pre> ASLDE-DATA.request()    ASLDE-DATA.response()    (DLDE-DATA.indication() &amp;&amp;    (TakeServiceID(Payload) != PUBLISH    TakeMsgType(Payload) != REQUEST        DataType != DATA)) =&gt; Ignore; </pre>	Active
T7	Active	<pre> DLDE-DATA.indication() &amp;&amp; DataType == DATA &amp;&amp; TakeServiceID(Payload) == PUBLISH &amp;&amp; TakeMsgType(Payload) == REQUEST &amp;&amp; GetUAPID(GetVcrID(SrcAddr, SUBSCRIBER, TakeUAPID(Payload))) ==      TakeUAPID(Payload) &amp;&amp; CheckEvent(TakeUAPID(Payload), PROCESS_DATA_NOT_UPDATED) =&gt; UAP_ID := TakeUAPID(Payload); VCR_ID := GetVcrID(SrcAddr, SUBSCRIBER, UAP_ID); ServiceID := TakeServiceID(Payload); AsduLength := PayloadLength - 4; Asdu := TakeASLPayload(PayloadLength, Payload); PutDataIntoBuffer(VCR_ID, AsduLength, Asdu); ASLDE-DATA.indication(   ServiceID,   UAP_ID,   AsduLength,   Asdu ); StartDataUpdateRateTimer(VCR_ID); StartDeadlineTimer(VCR_ID); SetEvent(UAP_ID, PROCESS_DATA_NOT_UPDATED, DISAPPEAR); </pre>	Active
T8	Active	<pre> DLDE-DATA.indication() &amp;&amp; DataType == DATA &amp;&amp; TakeServiceID(Payload) == PUBLISH &amp;&amp; TakeMsgType(Payload) == REQUEST &amp;&amp; GetUAPID(GetVcrID(SrcAddr, SUBSCRIBER, TakeUAPID(Payload))) ==      TakeUAPID(Payload) &amp;&amp; ! CheckEvent(TakeUAPID(Payload), PROCESS_DATA_NOT_UPDATED) =&gt; UAP_ID := TakeUAPID(Payload); VCR_ID := GetVcrID(SrcAddr, SUBSCRIBER, UAP_ID); ServiceID := TakeServiceID(Payload); </pre>	Active

表 95 (续)

编号	当前状态	事件\条件 => 动作	下一状态
T8	Active	<pre> AsduLength := PayloadLength - 4; Asdu := TakeASLPayload(PayloadLength, Payload); PutDataIntoBuffer(VCR_ID, AsduLength, Asdu); ASLDE-DATA.indication(     ServiceID,     UAP_ID,     AsduLength,     Asdu ); StartDataUpdateRateTimer(VCR_ID); StartDeadlineTimer(VCR_ID); </pre>	Active
T9	Active	<pre> DatyUpdateCycle Timer with VCR_ID expires =&gt; Asdu := GetDataFromBuffer(VCR_ID); AsduLength := Sizeof(Asdu); UAP_ID := GetUAPID(VCR_ID); ASLDE-DATA.indication(     ServiceID := PUBLISH,     UAP_ID,     AsduLength,     Asdu ); StartDataUpdateRateTimer(VCR_ID); </pre>	Active
T10	Active	<pre> Deadline timer with VCR_ID expires &amp;&amp; ! CheckEvent(GetUAPID(VCR_ID), PROCESS_DATA_NOT_UPDATED) =&gt; UAP_ID := GetUAPID(VCR_ID); SetEvent(UAP_ID, PROCESS_DATA_NOT_UPDATED, APPEAR); StartDeadlineTimer(VCR_ID); </pre>	Active
T11	Active	<pre> Deadline timer with VCR_ID expires &amp;&amp; CheckEvent(GetUAPID(VCR_ID), PROCESS_DATA_NOT_UPDATED) =&gt; StartDeadlineTimer(VCR_ID); </pre>	Active

#### 10.7.4.8 报告源状态机

报告源状态机(AMRS)仅具有 Active 状态,表示 VCR 端点处于激活。

报告源状态机的状态转换如图 106 和表 96 所示。

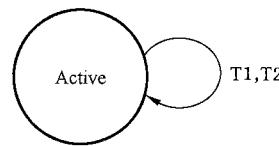


图 106 报告源状态机 AMRS 状态转换图

表 96 报告源状态机 AMRS 状态转换表

编号	当前状态	事件\条件 => 动作	下一状态
T1	Active	ASLDE-DATA.response()    (ASLDE-DATA.request() && ServiceID! = REPORT)    DLDE-DATA.indication() => Ignore;	Active
T2	Active	ASLDE-DATA.request() && ServiceID == REPORT => VCR_ID := GetVcrID(DstAddr, ServiceID, UAP_ID); Msgtype := REQUEST; DLDE-DATA.request( VCR_ID, DataType := DATA, Priority := Priority, PayloadLength := AsduLength + 4, Payload := BuildAPDU(ServiceID, Msgtype, UAP_ID, AsduLength, Asdu) );	Active

#### 10.7.4.9 报告汇状态机

报告汇状态机(AMSK)仅具有 Active 状态,表示 VCR 端点处于激活。

报告汇 VCR 端点状态转换如图 107 和表 97 所示。

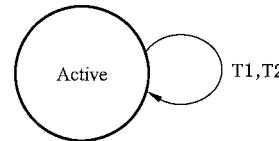


图 107 报告汇状态机 AMSK 状态转换图

表 97 报告汇状态机 AMSK 状态转换表

编号	当前状态	事件\条件 => 动作	下一状态
T1	Active	ASLDE-DATA.request()    ASLDE-DATA.response()              (DLDE-DATA.indication() &&           (DataType != DATA    TakeServiceID(Payload) != REPORT              TakeMsgType(Payload) != REQUEST))	Active
T2	Active	DLDE-DATA.indication()           && DataType == DATA           && TakeServiceID(Payload) == REPORT           && TakeMsgType(Payload) == REQUEST	Active
		=>           UAP_ID := TakeUapID(Payload);           VCR_ID := GetVcrID(SrcAddr, REPORT_SINK, UAP_ID);           ASLDE-DATA.indication(             ServiceID := TakeServiceID(Payload),             UAP_ID := TakeUAPID(Payload),             AsduLength := PayloadLength - 4,             Asdu := TakeASLPayload(PayloadLength, Payload)           );	

## 10.7.4.10 应用子层状态机函数

表 98 给出了应用子层状态机中使用的所有函数。

表 98 ASLM 中所有函数

函数	输入	输出	描述
BuildAPDU	ServiceID Msgtype UAP_ID AsduLength Asdu	Apdu	根据 ASL 通用包格式, 为 ASDU 增加 ASL 包头以构建 APDU
BuildErrAsdu	ErrorCode AddInfo	Asdu	构建 AL 服务负响应报文
CheckEvent	UAP_ID EventFlag	Result	对于属于 UAP 的所有 UAO 的事件数据, 检查其 EventFlag 比特的值是否被置位, Result 取值如下: TRUE=EventFlag 比特值为 1; FALSE=EventFlag 比特值为 0
CreatBuffer	VCR_ID		为 VCR 端点创建一个数据缓冲区, 以保存输入或输出数据

表 98 (续)

函数	输入	输出	描述
GetDataFromBuffer	VCR_ID	Asdu	从 VCR 端点的缓冲区中取出数据
GetPeerAddr	VCR_ID	PeerAddr	获得 VCR 端点的 PeerAddr 值
GetUAPID	VCR_ID	UAP_ID	获得 VCR 端点的 UAP_ID 值
GetVcrID	DstAddr VcrEpType UAP_ID	VCR_ID	根据 DstAddr、VcrEpType 和 UAP_ID, 在 VCRLIST 中查找相匹配的 VCR 端点，并返回 VCR_ID
PutDataIntoBuffer	VCR_ID AsduLength Asdu		将数据缓存在 VCR 端点的缓冲区
RestoreSvrid	VCR_ID	ServiceID	获取之前在 VCR 端点保存的 ServiceID 值
SetEvent	UAP_ID EventFlag AppearFlag		对于属于 UAP 的所有 UAO 的事件数据，将其 EventFlag 比特的值设为 AppearFlag
Sizeof	Data	Length	计算数据 Data 的长度
StartActiveTimer	VCR_ID		启动 VCR 端点的 VCRActiveTime 定时器
StartDataUpdateRateTime	VCR_ID		启动 VCR 端点的 DataUpdateRate 定时器
StartDeadlineTimer	VCR_ID		启动 VCR 端点的 Deadline 定时器
StartWatchdogTimer	VCR_ID		启动 VCR 端点的 Watchdog 定时器
StoreSvrid	VCR_ID ServiceID		在 VCR 端点保存 ServiceID 值
TakeASLPayload	DllPayloadLength DllPayload	Asdu	根据 ASL 通用包格式, 从 DLL 负载中解析出 ASDU
TakeMsgType	DllPayload	Msgtype	根据 ASL 通用包格式, 从 DLL 负载中解析出 Msgtype 值
TakeServiceID	DllPayload	ServiceID	根据 ASL 通用包格式, 从 DLL 负载中解析出 ServiceID 值
TakeUapid	DllPayload	UAP_ID	根据 ASL 通用包格式, 从 DLL 负载中解析出 UUAP_ID 值

## 11 安全

### 11.1 概述

#### 11.1.1 安全管理架构

综合面向工厂自动化无线网络的实时性、现场设备资源的有限性与安全管理问题,本标准定义分层的安全策略和安全措施,构成 WIA-FA 的安全体系架构,包括网关设备上的安全管理者、接入设备上的安全管理模块和现场设备上的安全管理模块,见图 108。附录 A 给出了 WIA-FA 网络的安全策略。

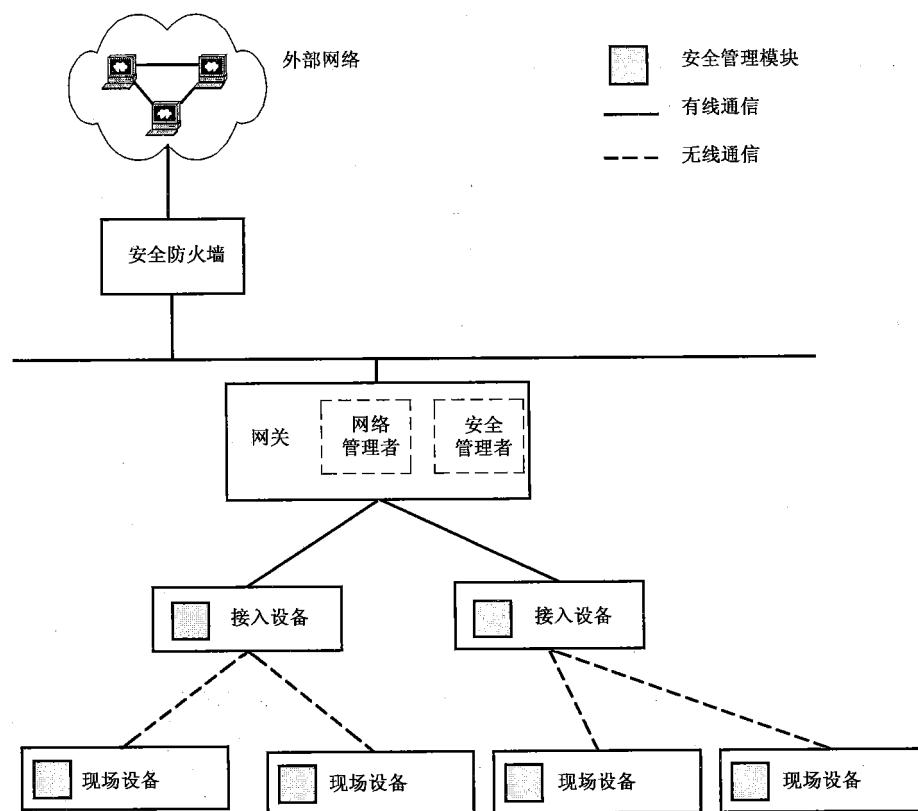


图 108 安全管理架构示意图

网关设备包含一个安全管理者,位于其 DMAP 中,功能主要包括:

- 根据具体应用,通过设置网关设备、接入设备和现场设备的 SecLevel 配置 WIA-FA 网络的安全措施(详见 A.4);
- 认证试图加入网络中的现场设备;
- 管理全网密钥,包括密钥产生、密钥建立和密钥更新;
- 接收并响应来自现场设备和接入设备的安全告警请求。

安全管理模块位于现场设备和接入设备的 DMAP 中,详见图 108(与系统管理的图一致),功能主要包含:

接入设备包含一个安全管理模块,位于 DMAP 中,功能主要包括:

- 维护数据链路层安全通信的密钥;
- 实现安全告警机制。

接入设备安全功能的实现在第 9 章中描述。

现场设备包含一个安全管理模块,位于其 DMAP 中,功能主要包括:

- 执行设备安全加入过程;
- 维护安全加入、安全通信以及密钥传输所需的密钥;
- 实现设备安全告警机制。

安全防火墙对整个 WIA-FA 网络实施边界保护,保证 WIA-FA 网络正常工作。安全防火墙不在本部分范围内。

### 11.1.2 安全功能

WIA-FA 网络提供以下几种安全功能:

- 设备认证功能；
- 数据完整性校验功能；
- 数据保密性功能；
- 防止重放攻击功能；
- 密钥管理功能；
- 安全告警功能。

其中，设备认证功能通过设备安全入网服务实现；数据完整性校验、数据保密和防止重放攻击通过数据链路层数据服务实现；密钥管理功能通过密钥建立服务和密钥更新服务实现；安全告警功能通过安全告警服务实现。

### 11.1.3 密钥

WIA-FA 网络的密钥长度均为 128 位，由安全管理者产生，以密钥结构体（详见表 21）的形式存放在每个设备 MIB 的密钥表中。

根据不同的安全功能，采用不同类型的密钥进行安全操作，具体密钥类型有：

- 共享密钥（KS）：全网统一的密钥，用于在设备未建立 KED 时的数据链路层安全通信密钥。该密钥在预配置阶段由安全管理者通过手持设备写入设备。
- 加入密钥（KJ）：用于试图加入 WIA-FA 网络的设备进行认证。该密钥在预配置阶段由安全管理者通过手持设备建立。
- 密钥加密密钥（KEK）：用于在密钥传输过程中对密钥结构体进行保护。该密钥在设备加入网络以后，由 WIA-FA 网络安全管理者建立。
- 单播数据加密密钥（KEDU）：用于保护接入设备与现场设备之间单播帧和聚合帧中每个现场设备数据的数据完整性和数据保密性。一个现场设备与接入设备通信时，同一组内的不同接入设备共享一个与该现场设备的单播数据加密密钥。该密钥在设备加入网络以后，由 WIA-FA 网络安全管理者建立。
- 广播数据加密密钥（KEDB）：用于保护接入设备向现场设备广播广播帧的数据完整性和数据保密性。该密钥在设备加入网络以后，由 WIA-FA 网络安全管理者建立。

为了保护密钥的安全性，不同密钥类型的生存周期不同，如图 109 所示。

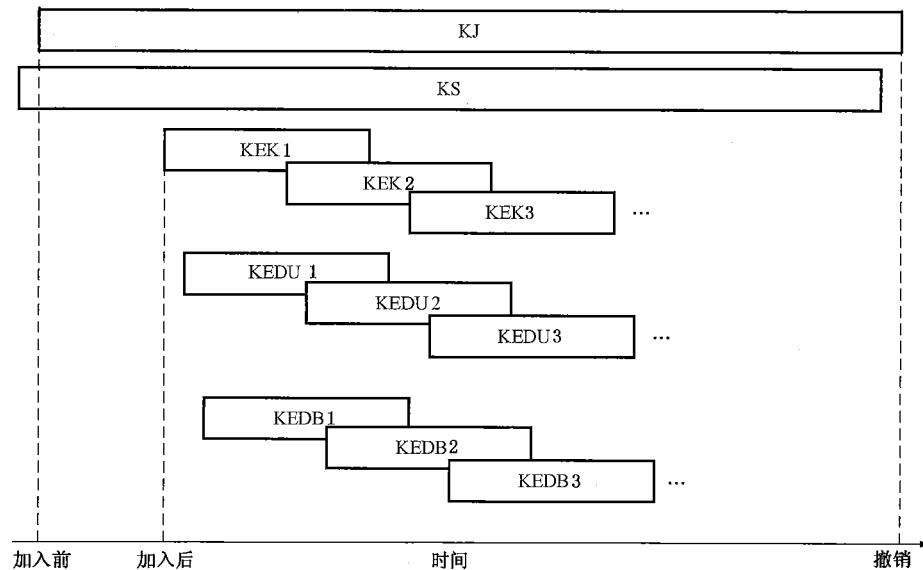


图 109 密钥生存周期

密钥在其生存周期内会经历以下状态：

- 备用(BACKUP)：密钥已经配置在 MIB 中，但它的 KeyActiveSlot(详见 6.7.1.2.1)未到；
- 可用(USING)：密钥的 KeyActiveSlot 到达，并且开始使用；
- 过期(EXPIRED)：密钥的 KeyActiveSlot+KeyUpdateDur 到达(详见 6.7.1.2.1)，但用于更新的新密钥尚未可用，此时该密钥继续使用；
- 无效(INVALID)：用于更新该密钥的新密钥开始使用时，该密钥进入此状态并停止使用。

对于已入网的现场设备，如果其 SecLevel 不等于 0 或 1，在同一时刻，每种密钥最多只有一个密钥处于可用或过期状态。

## 11.2 安全相关服务

### 11.2.1 概述

安全相关服务包含密钥建立服务、密钥更新服务、和安全告警服务。它们通过网关设备的安全管理者以及接入设备和现场设备的安全管理模块共同提供，并通过 DLME-SAP 访问。

接入设备的密钥建立服务、密钥更新服务和安全告警服务在第 9 章定义，本条只对现场设备的密钥建立服务、密钥更新服务和安全告警服务进行定义。

### 11.2.2 密钥建立服务

#### 11.2.2.1 密钥建立请求原语

密钥建立请求原语 KEY-ESTABLISH.request 用于网关安全管理者请求接入设备的数据链路层发送密钥建立请求帧。

```
KEY-ESTABLISH.request(
    DstAddr,
    KeyMaterial
)
```

KEY-ESTABLISH.request 原语的参数如表 99 所示。

表 99 KEY-ESTABLISH.request 原语的参数

参数名称	数据类型	取值范围	描述
DstAddr	Unsigned16	0~65 535	目标设备短地址
KeyMaterial	KeyMaterial_Struct	—	密钥的相关信息及其完整性校验码，见表 21

密钥分发时密钥材料的格式如表 100 所示。

表 100 KeyMaterial\_Struct 结构

参数名称	数据类型	取值范围	描述
KeyID	Unsigned16	0~65 535	密钥 ID
KeyType	Unsigned8	0~255	密钥类型，见表 21
KeyActiveSlot	Unsigned48	0~(2 <sup>48</sup> −1)	密钥启用的绝对时隙号
KeyDataValue	KeyData	—	密钥值，采用 KJ 或 KEK 和 CCM * (见 [IEEE 802.15.4-2006 B.4.1]) 加密
KeyMIC	Unsigned32	0~(2 <sup>32</sup> −1)	完整性校验码，用 KJ 或 KEK 和 CCM * (见 [IEEE 802.15.4-2006 B.4.1]) 产生，其保护范围包括 KeyID, KeyType, KeyActiveSlot 和 KeyDataValue

为了保护密钥传输中密钥相关信息的安全,网关中的安全管理者应对其采用 CCM \* (见[IEEE 802.15.4-2006 B.4.1])进行保护。在密钥建立时,使用目标现场设备的 KJ 进行保护;在密钥更新时,使用目标现场设备的 KEK 进行保护。

用于 CCM \* (见[IEEE 802.15.4-2006 B.4.1])的 13 个八位位组的 NONCE 如图 110 所示。

8 个八位位组	4 个八位位组	1 个八位位组
PhyAddress	TimeStamp	SecurityLevel

图 110 NONCE 结构

其中,PhyAddress 为现场设备的 64 位物理地址;TimeStamp 为 KeyActiveSlot 的低四个字节;SecurityLevel 设定为 5,即加密 &MIC-32。

#### 11.2.2.2 密钥建立指示原语

密钥建立指示原语 KEY-ESTABLISH.indication 用于现场设备的数据链路层指示接收到一个密钥建立请求帧,并将其发送给设备的安全管理模块。

KEY-ESTABLISH.indication(

KeyMaterial

)

KEY-ESTABLISH.indication 原语的参数如表 101 所示。

表 101 KEY-ESTABLISH.indication 原语的参数

参数名称	数据类型	取值范围	描述
KeyMaterial	KeyMaterial_Struct, 见表 100	—	密钥材料

当接收到该原语时,现场设备的安全管理模块应先用 KJ 对 KeyDataValue 进行解密,然后校验 KeyMIC。如果 KeyMIC 校验正确,为该密钥建立相应的 Key\_Struct 并存储在其 KeyList 中。

#### 11.2.2.3 密钥建立响应原语

密钥建立响应原语 KEY-ESTABLISH.response 为现场设备的安全管理模块产生,用于请求其数据链路层发送密钥建立响应帧。

KEY-ESTABLISH.response(

KeyID,

Status

)

KEY-ESTABLISH.response 原语的参数如表 102 所示。

表 102 KEY-ESTABLISH.response 原语的参数

参数名称	数据类型	取值范围	描述
KeyID	Unsigned16	0~65 535	所接收密钥的 ID
Status	Unsigned8	0~255	密钥接收结果: 0 = SUCCESS; 1 = FAILURE; 其余保留

### 11.2.2.4 密钥建立证实原语

密钥建立证实原语 KEY-ESTABLISH.confirm 用于接入设备的数据链路层通知网关的安全管理者密钥建立是否成功。

KEY-ESTABLISH.confirm(

DstAddr,  
KeyID,  
Status

)

KEY-ESTABLISH.confirm 原语的参数如表 103 所示。

表 103 KEY-ESTABLISH.confirm 原语的参数

参数名称	数据类型	取值范围	描述
DstAddr	Unsigned16	0~65 535	设备短地址
KeyID	Unsigned16	0~65 535	所接收密钥的 ID
Status	Unsigned8	0~255	密钥接收结果 0 = SUCCESS; 1 = FAILURE; 其余保留

### 11.2.2.5 密钥建立时序

密钥建立的时序如图 111 所示。

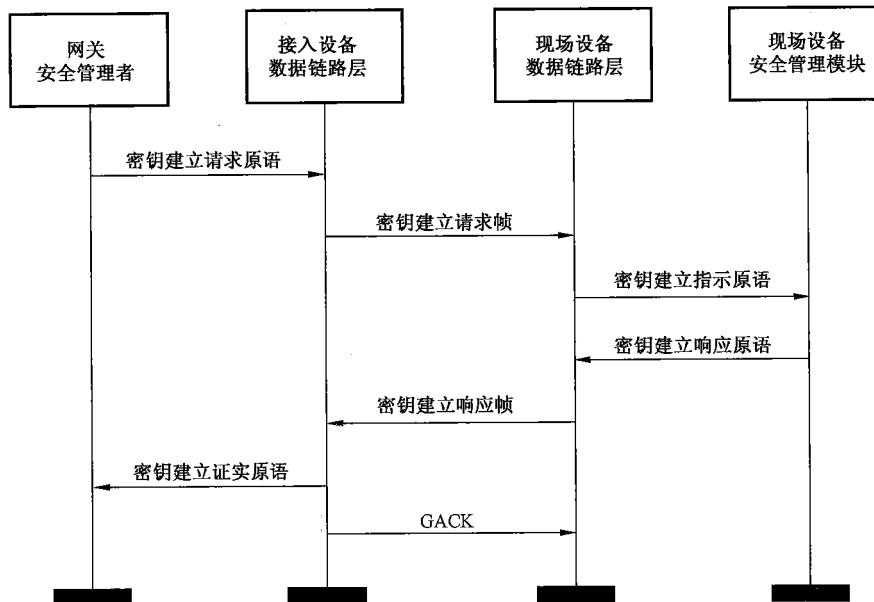


图 111 密钥建立时序图

### 11.2.3 密钥更新服务

#### 11.2.3.1 密钥更新请求原语

密钥更新请求原语 KEY-UPDATE.request 用于网关安全管理者请求接入设备的数据链路层发送密钥更新请求帧。

```
KEY-UPDATE.request(
    DstAddr,
    KeyMaterial
)
```

KEY-UPDATE.request 原语的参数如表 104 所示。

表 104 KEY-UPDATE.request 原语的参数

参数名称	数据类型	取值范围	描述
DstAddr	Unsigned16	0~65 535	目标设备短地址
KeyMaterial	KeyMaterial_Struct	—	密钥的相关信息及其完整性校验码, 见表 100

#### 11.2.3.2 密钥更新指示原语

密钥更新指示原语 KEY-UPDATE.indication 用于现场设备的数据链路层指示接收到一个密钥更新请求帧, 并将其发送给设备的安全管理模块。

```
KEY-UPDATE.indication(
    KeyMaterial
)
```

KEY-UPDATE.indication 原语的参数如表 105 所示。

表 105 KEY-UPDATE.indication 原语的参数

参数名称	数据类型	取值范围	描述
KeyMaterial	KeyMaterial_Struct, 见表 100	—	密钥材料

当接收到该原语时, 现场设备的安全管理模块应先用 KEK 对 KeyDataValue 进行解密, 然后校验 KeyMIC。如果 KeyMIC 校验正确, 为该密钥建立相应的 Key\_Struct 并存储在其 KeyList 中。

#### 11.2.3.3 密钥更新响应原语

密钥更新响应原语 KEY-UPDATE.response 由现场设备的安全管理模块产生, 用于请求其数据链路层发送密钥更新响应帧。

```
KEY-UPDATE.response(
    KeyID,
    Status
)
```

KEY-UPDATE.response 原语的参数如表 106 所示。

表 106 KEY-UPDATE.response 原语的参数

参数名称	数据类型	取值范围	描述
KeyID	Unsigned16	0~65 535	所接收密钥的 ID
Status	Unsigned8	0~255	密钥接收结果： 0 = SUCCESS; 1 = FAILURE; 其余保留

#### 11.2.3.4 密钥更新证实原语

密钥更新证实原语 KEY-UPDATE.confirm 用于接入设备的数据链路层通知网关的安全管理者密钥更新是否成功。

KEY-UPDATE.confirm(

KeyID,  
ShortAddr,  
Status

)

KEY-UPDATE.confirm 原语的参数如表 107 所示。

表 107 KEY-UPDATE.confirm 原语的参数

参数名称	数据类型	取值范围	描述
ShortAddr	Unsigned16	0~65 535	设备短地址
KeyID	Unsigned16	0~65 535	所接收密钥的 ID
Status	Unsigned8	0~255	密钥接收结果 0 = SUCCESS; 1 = FAILURE; 其余保留

#### 11.2.3.5 密钥更新时序

密钥更新的时序如图 112 所示。

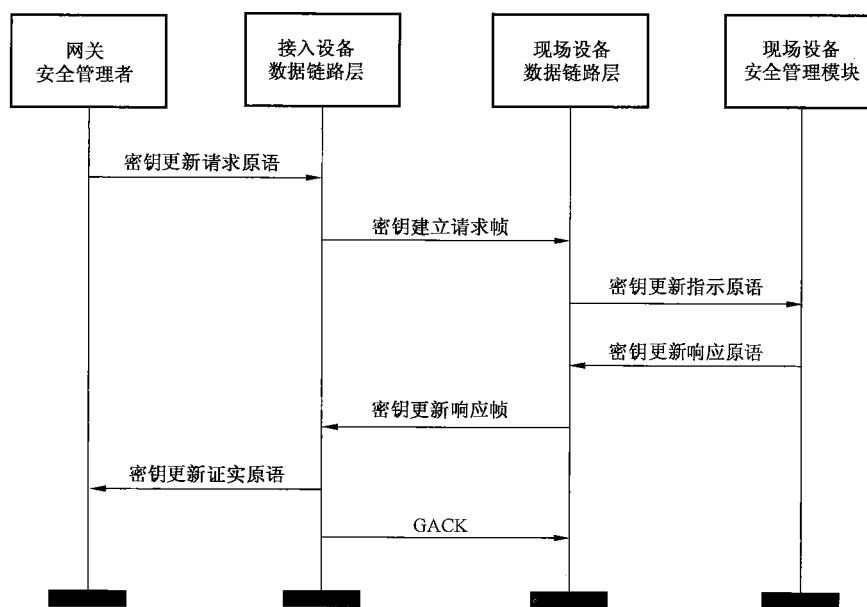


图 112 密钥更新时序图

#### 11.2.4 安全告警服务

##### 11.2.4.1 安全告警请求原语

安全告警请求原语 SEC-ALARM.request 用于现场设备的安全管理模块请求其数据链路层发送安全告警请求帧。

```

SEC-ALARM.request(
    SecAlarmCount,
    SecAlarmList
)
  
```

SEC-ALARM.request 原语的参数如表 108 所示。

表 108 SEC-ALARM.request 原语的参数

参数名称	数据类型	取值范围	描述
SecAlarmCount	Unsigned8	0~255	安全告警的数量
SecAlarmList	SecAlarm_Struct 列表	—	安全告警列表,格式见图 113

SecAlarm\_Struct 的格式如图 113 所示。

KeyID	AlarmFlag
2 个八位位组	1 个八位位组

图 113 SecAlarmt\_Struct 结构

SecAlarmList 中包含 n 个告警, n 由 SecAlarmCount 指出。表中每一个告警包含两个域:

——KeyID:与告警相关密钥的 ID;

——AlarmFlag:与告警相关的密钥的 AlarmFlag(详见表 21)。

### 11.2.4.2 安全告警指示原语

安全告警指示原语 SEC-ALARM.indication 用于接入设备的数据链路层指示其接收到一个安全告警请求帧，并将该帧发送给网关设备的安全管理者。

SEC-ALARM.indication(

```
SrcAddr,  
SecAlarmCount,  
SecAlarmList
```

)

SEC-ALARM.indication 原语的参数如表 109 所示。

表 109 SEC-ALARM.indication 原语的参数

参数名称	数据类型	取值范围	描述
SrcAddr	Unsigned16	0~65 535	发送安全告警的现场设备的短地址
SecAlarmCount	Unsigned8	0~255	安全告警的数量
SecAlarmList	SecAlarm_Struct 列表	—	安全告警列表, 格式见图 113

当接收到此原语时, 安全管理者应该对 SecAlarmList 中相关密钥进行更新。

### 11.2.4.3 安全告警时序

安全告警时序如图 114 所示。

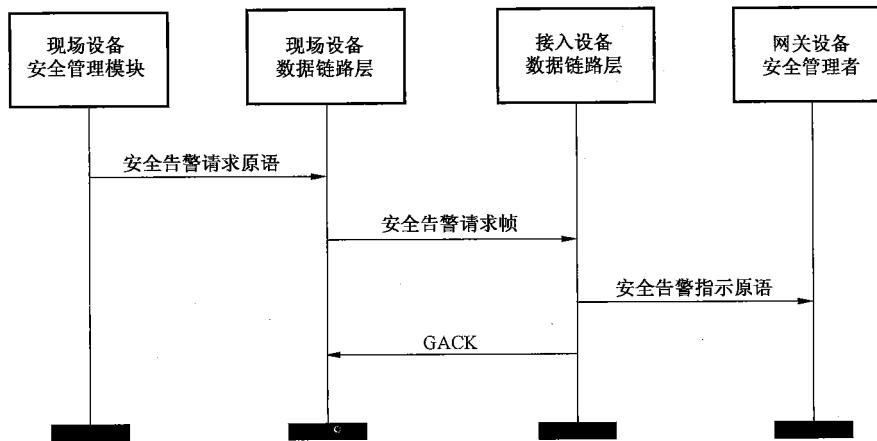


图 114 安全告警时序图

## 11.3 安全加入

### 11.3.1 概述

当 SecLevel 不等于 0 时, 安全管理者需要在现场设备的加入过程中对其进行认证。认证通过现场设备的物理地址和 KJ 完成。

在安全加入过程中, 现场设备和网关设备的中的网络管理者通过带有 SecMaterial 的 DLME-JOIN.request 和 DLME-JOIN.indication(见 8.3.4)完成安全加入过程, SecMaterial 计算如式(6)所示:

$$\text{SecMaterial} = \text{低 } 64 \text{ 位的 HMAC(KJ, PhyAddress)} \dots \dots \dots \dots \dots \dots \dots \quad (6)$$

其中, HMAC 基于 MD5 算法, HMAC 基于 MD5 算法, 见 IETF RFC1321。

### 11.3.2 设备安全加入流程

在安全入网前,现场设备需要预配置以下信息:

- 网络 ID;
- SecLevel;
- KJ 和 KS

预配置完成后,设备安全加入 WIA-FA 网络流程如图 115 所示,具体步骤如下:

- a) 接入设备周期性广播信标帧;
- b) 待入网的现场设备扫描信道并接收来自接入设备的信标帧,同时通过单向时间同步方法完成与网关设备的时间同步(详见 8.1.4);
- c) 现场设备从接收到的信标帧中选取一个信道,并获取用于发送加入请求的共享时隙和信道信息,在共享时隙内发送带有安全信息(SecMaterial 见 11.3.1)的加入请求帧。其中共享时隙由信标帧内的“First shared timeslot number”和“Shared timeslot count”(见 8.4.6)确定;
- d) 在接收到加入请求后,网关设备的网络管理者将现场设备的安全材料和物理地址转发给安全管理者;
- e) 安全管理者通过现场设备的安全材料、物理地址和 KJ 对现场设备进行认证,如果认证成功,向网络管理者回复认证成功,否则回复认证失败;
- f) 网络管理者向现场设备回复加入响应,当认证失败时,Status 设为 2(详见 8.3.4.3 中 Status 定义);
- g) 现场设备接收加入响应。如果加入响应中的 Status 为 2,现场设备重复上述过程。

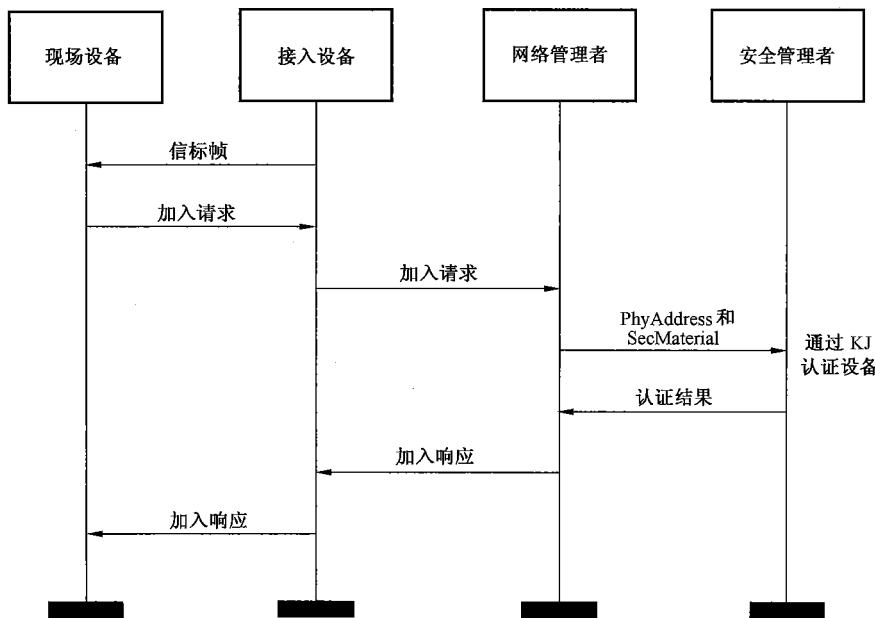


图 115 安全加入过程流程图

## 11.4 密钥管理

### 11.4.1 概述

密钥管理主要包含密钥建立和密钥更新。密钥建立用于安全管理者为现场设备建立新密钥;密钥

更新则是安全管理者根据密钥的生存周期对设备中已存在且即将过期的密钥进行更新。

#### 11.4.2 密钥建立流程

现场设备加入网络后,安全管理者应为其建立 KEK, KEDU 和 KEDB, 其流程如图 116 所示。

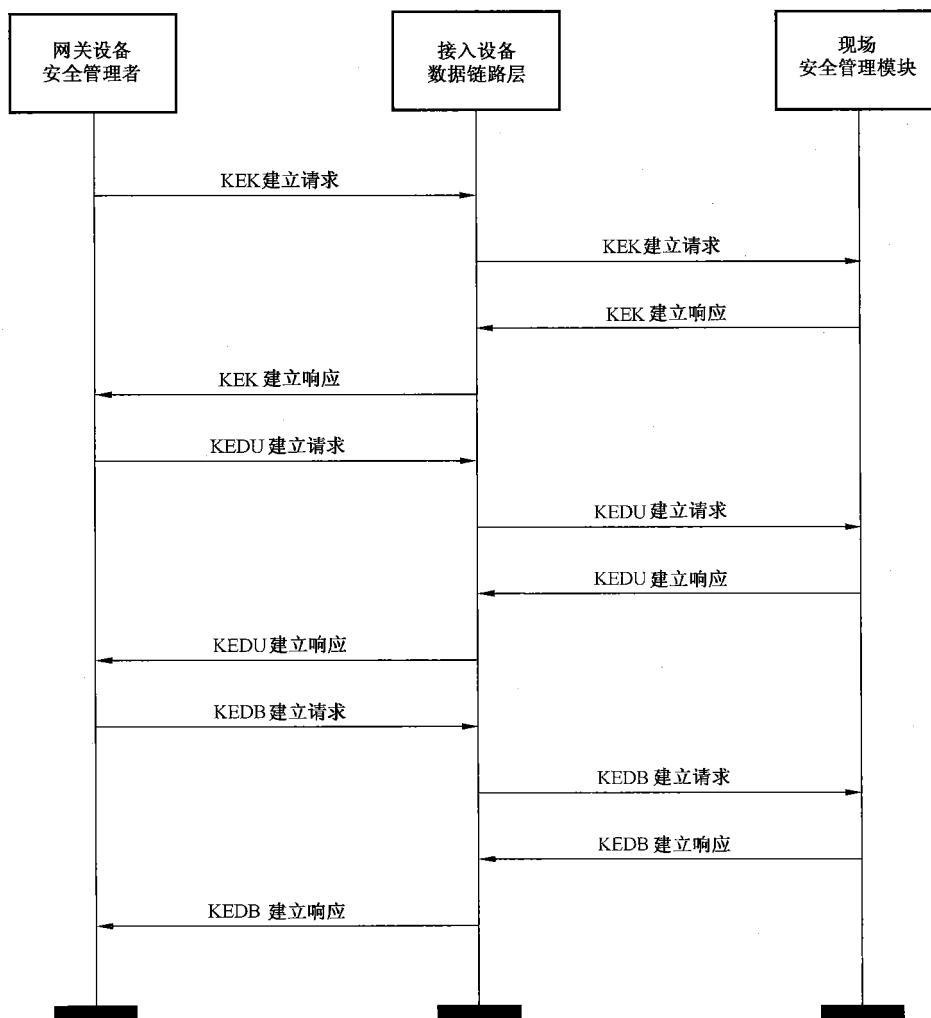


图 116 现场设备密钥建立流程

#### 11.4.3 密钥更新流程

安全管理者需要对网络中每一个正在使用的 KEK、KEDU 和 KEDB 在其 KeyActiveSlot + KeyUpdateDur 时间结束前进行更新。

对于现场设备中的 KEK、KEDU 和 KEDB, 定义正在使用的密钥为 Current\_key, 定义即将使用的密钥为 New\_key。

密钥更新的状态描述如表 110 所示, 状态机和状态转移分别如图 117 和表 111 所示。

表 110 密钥更新状态

状态名称	描述
ST1	Current_key:无, New_key:备用
ST2	Current_key:可用, New_key:无
ST3	Current_key:可用, New_key:备用
ST4	Current_key:过期, New_key 备用
ST5	Current_key:过期, New_key:无

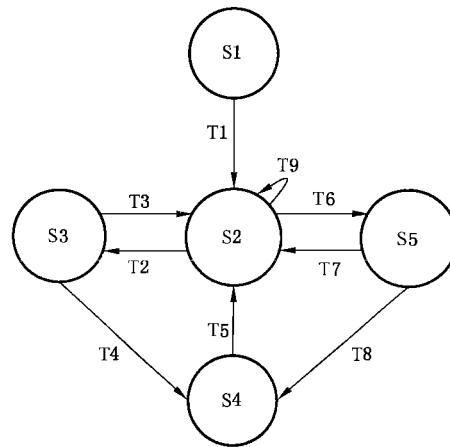


图 117 现场设备密钥更新状态机

表 111 密钥更新状态转移

编号	当前状态	事件\条件 =>动作	下一状态
T1	S1	ASN == New_key.KeyActiveSlot => Current_key:= New_key; Current_key.state:= USING;	S2
T2	S2	Reception of New_key && ASN < New_key.KeyActiveSlot => New_key.state:= BACKUP;	S3
T3	S3	ASN == New_key.KeyActiveSlot => Current_key.state:= INVALID; Current_key:= New_key; Current_key.state:= USING;	S2
T4	S3	ASN > Current_key.KeyActiveSlot + KeyUpdateDur && ASN < New_key.KeyActiveSlot => Current_key.state := EXPIRED;	S4

表 111(续)

编号	当前状态	事件\条件 =>动作	下一状态
T5	S4	ASN == New_key.KeyActiveSlot => Current_key.state := INVALID; Current_key := New_key; Current_key.state := USING;	S2
T6	S2	ASN > Current_key.KeyActiveSlot + KeyUpdateDur && No New_key => Current_key.state := EXPIRED;	S5
T7	S5	Reception of New_key && ASN ≥ New_key.KeyActiveSlot => Current_key.state := INVALID; Current_key := New_key; Current_key.state := USING;	S2
T8	S5	Reception of New_key && ASN < New_key.KeyActiveSlot => New_key.state := BACKUP;	S4
T9	S2	Reception of New_key && ASN ≥ New_key.KeyActiveSlot => Current_key.state := INVALID; Current_key := New_key; Current_key.state := USING;	S2

### 11.5 数据链路层安全通信

数据链路层应根据 MIB 中 SecLevel 的值对数据链路层帧进行加密/解密和完整性校验。数据链路层帧的加密/解密和完整性校验通过 CCM \* 实现(见 IEEE 802.15.4-2006 B.4.1), 安全数据链路层帧格式如图 118 所示。

当进行解密/解密和完整性校验时, 应根据表 112 所示使用不同的密钥。

表 112 数据链路层安全通信密钥使用

阶段	帧类型	密钥类型
KEDU 和 KEDB 建立前	所有需要安全保护的帧	KS
KEDU 和 KEDB 建立后	单播帧	KEDU
	非聚合广播帧	KEDB
	聚合广播帧	KEDU 用于聚合帧内发送给该现场设备数据的保护; KEDB 用于整个聚合帧的保护

在接入设备发送聚合广播帧前,其数据链路层应先采用聚合帧内每一个数据接收方对应的 KEDU 分别对每一个数据进行加密和/或计算完整性校验码,然后再采用 KEDB 对整个聚合帧进行加密和/或完整性校验码计算。在接收到聚合广播帧后,现场设备的数据链路层应先用 KEDB 对整个聚合帧进行解密和/或完整性校验,然后再使用自己的 KEDU 对帧内属于自己的数据进行解密和/或完整性校验。

## 11.6 安全告警

WIA-FA 定义两种安全告警:

- 密钥攻击告警:在一个 AttackStaticsDur(详见 6.7.1.2.1)内,密钥受到攻击的次数超过 Max-KeyAttackedNum(详见 6.7.1.2.1)。用某密钥计算 MIC(见 11.7.1)时,如果结果错误认为是该密钥受到攻击,MIC 计算错误的次数认为是密钥受到攻击的次数;
- 密钥更新超时告警:密钥的使用时间已超过 KeyUpdateDur(详见 6.7.1.2.1),但仍可用于更新的新密钥。

当现场设备的安全管理模块检测到安全告警事件发生时,应将与密钥相关的 AlarmFlag(见表 21)设置为 1,然后调用安全告警请求原语将安全告警事件及其他已存在的安全告警事件一起,根据 AlarmRptDur(见详见 6.7.1.2.1)周期性汇报给网关设备中的安全管理者。

当网关设备的安全管理者接收到安全告警请求后,应对安全告警请求中相应的密钥进行密钥更新。当有告警的密钥(AlarmFlag)不等于 0 被成功更新后,对应的 AlarmFlag 标志应清零。

## 11.7 安全相关帧格式

### 11.7.1 安全数据链路层通用帧格式

安全数据链路层通用帧格式如图 118 所示。

DLDPU			
7/8 八位位组	可变长度	0/4/8/16 八位位组	2 八位位组
DLL 帧头	DLL 载荷	MIC	FCS

图 118 带安全的数据链路层帧结构

安全数据链路层通用帧包含以下域:

- DLL 帧头:见 8.4.1;
- DLL 载荷:根据 MIB 中的 SecLevel 采用 CCM \*(见 IEEE 802.15.4-2006 B.4.1)和对应的 KEDU 或 KEDB 进行加密;

- MIC：根据 MIB 中的 SecLevel，若需要进行完整性校验，采用 CCM \*（见 IEEE 802.15.4-2006 B.4.1）和对应的 KEDU 或 KEDB 计算完整性码并填入 MIC，其保护范围为 DLL 帧头和 DLL 载荷；
- FCS，见 8.4.1。

表 113 为 WIA-FA 安全数据链路层帧所支持的安全等级。

表 113 安全数据链路层帧的安全等级

SecLevel	DLL 载荷加密	MIC 长度(位)
0/1	否	0
2	否	32(MIC-32)
3	否	64(MIC-64)
4	否	128(MIC-128)
5	是	0
6	是	32(MIC-32)
7	是	64(MIC-64)
8	是	128(MIC-128)

相关 NONCE 定义如图 110 所示，其中 PhyAddress 在单播通信中设置为现场设备的 64 位物理地址，在广播通信中设置为 0；TimeStamp 设置为该帧传输或接收所在时隙的 ASN；SecurityLevel 设置 MIB 中的 SecLevel。

### 11.7.2 安全聚合帧格式

安全聚合帧格式如图 119 所示。

		数据 1				...	数据 n				0/4/8/16 八位位组	0/4/8/16 八位位组	2 八位 位组
		1 八位 位组	2 八位 位组	1 八位 位组	可变长度		2 八位 位组	1 八位 位组	可变 长度	0/4/8/16 八位位组			
DLL 帧头	聚合 数量	现场设备 短地址	数据 长度	数据	sMIC	...	现场设备 短地址	数据 长度	数据	sMIC	MIC	FCS	

图 119 安全聚合帧格式

安全聚合帧包含以下域：

- DLL 帧头：见 8.4.1；
- 聚合数量：见 8.1.5；
- 现场设备短地址：见 8.1.5；
- 数据长度：见 8.1.5；
- 数据：见 8.1.5，根据 MIB 中 SecLevel 的值采用 CCM \*（见 IEEE 802.15.4-2006 B.4.1）和接收该数据的现场设备的 KEDU 进行加密；
- sMIC：根据 MIB 中 SecLevel 的值，如果需要进行完整新校验，则采用接收该数据的现场设备的 KEDU 和 CCM \*（见 IEEE 802.15.4-2006 B.4.1）计算完整性校验码，其保护范围为现场设备短地址、数据长度和数据，生成的完整性校验码填入 sMIC；
- MIC：见 11.7.1，密钥采用 KEDB；

——FCS:见 8.4.1。

对于聚合帧内部每一个数据的加密/解密和完整性校验,NONCE 中 PhyAddress 设置为接收该数据的现场设备的 64 位物理地址;而对于整个聚合帧的加密/解密和完整性校验,PhyAddress 则设置为 0。

### 11.7.3 密钥建立请求帧格式

密钥建立请求帧格式如图 120 所示。

7/8 八位位组	29 八位位组	04/8/16 八位位组	2 八位位组
DLL 帧头	KeyMaterial	MIC	FCS

图 120 密钥建立请求帧格式

密钥建立请求帧包含以下域:

——DLL 帧头:见 8.4.1;

——KeyMaterial:用于密钥建立的密钥材料,见表 100;

——MIC:见 11.7.1;

——FCS:见 8.4.1。

### 11.7.4 密钥建立响应帧格式

密钥建立响应帧格式如图 121 所示。

7/8 八位位组	29 八位位组	1 八位位组	04/8/16 八位位组	2 八位位组
DLL 帧头	KeyID	Status	MIC	FCS

图 121 密钥建立响应帧格式

密钥建立响应帧包含以下域:

——DLL 帧头:见 8.4.1;

——KeyID:所建立密钥的 ID;

——Status:密钥建立的结果,见表 102;

——MIC:见 11.7.1;

——FCS:见 8.4.1。

### 11.7.5 密钥更新请求帧格式

密钥更新请求帧格式如图 122 所示。

7/8 八位位组	29 八位位组	04/8/16 八位位组	2 八位位组
DLL 帧头	KeyMaterial	MIC	FCS

图 122 密钥更新请求帧格式

密钥更新请求帧包含以下域:

——DLL 帧头:见 8.4.1;

——KeyMaterialL:用于密钥更新的密钥材料,见表 100;

——MIC:见 11.7.1;

——FCS:见 8.4.1。

### 11.7.6 密钥更新响应帧格式

密钥更新响应帧格式如图 123 所示。

7/8 八位位组	29 八位位组	1 八位位组	04/8/16 八位位组	2 八位位组
DLL 帧头	KeyID	Status	MIC	FCS

图 123 密钥更新响应帧格式

密钥更新响应帧包含以下域：

- DLL 帧头：见 8.4.1；
- KeyID：所更新密钥的 ID；
- Status：密钥更新的结果，见表 106；
- MIC：见 11.7.1；
- FCS：见 8.4.1。

### 11.7.7 安全告警帧格式

安全告警帧格式如图 124 所示。

7/8 八位位组	1 八位位组	可变长度	0/4/8/16 八位位组	2 八位位组
DLL 帧头	SecAlarmCount	SecAlarmList	MIC	FCS

图 124 安全告警帧格式

安全告警帧包含以下域：

- DLL 帧头：见 8.4.1；
- SecAlarmCount：SecAlarmList 中包含安全告警的数量，见表 108；
- SecAlarmList：安全告警列表，见表 108；
- MIC：见 11.7.1；
- FCS：见 8.4.1。

附录 A  
(规范性附录)  
**WIA-FA 网络的安全策略**

#### A.1 WIA-FA 网络的风险分析

WIA-FA 作为一个开放系统,其潜在的安全风险是不可避免的。因此,必须采取必要的安全措施,以保证 WIA-FA 用户在这个开放的环境中能够安全地操作,保护系统内部的资源和维持正常的生产秩序。WIA-FA 网络安全的主要目标是保障系统的正常运行,或在受到攻击时能够迅速地发现并采取相应的安全措施,使系统的安全损失减少到最小,并在受到攻击后能够迅速地恢复。

WIA-FA 网络传输的数据可能受到入侵、毁坏数据或重放攻击等威胁,可分为恶意和非恶意两类。主要有利用网络对资源或者信息进行未授权访问、更改、拒绝三种操作。攻击来自 WIA-FA 网络内部以及 WIA-FA 网络外部。

#### A.2 WIA-FA 安全原则

由于 WIA-FA 网络的特点,其安全措施和安全服务应该遵循以下原则:

- 易于部署和使用;
- 尽可能减少与人有关的操作;
- 最大化延长电池寿命:包括减少包的大小和数目;利用基于硬件的加密技术等;
- 最大限度的使用已有的加密和鉴别技术和现有的标准。

#### A.3 WIA-FA 安全目标

WIA-FA 网络安全目标应该包括:

- 系统的可用性:合法用户根据需要可以随时访问系统资源。
- 数据的完整性:保证信息的一致性,防止非法用户对系统数据的篡改。
- 设备认证:对网络中的设备进行验证,证实其身份与其所声称的身份是否一致。
- 机密性:保证系统的硬件、软件和数据只能为合法用户所使用。
- 密钥管理:提供安全的密钥更新和管理的机制。

#### A.4 安全系统的分级

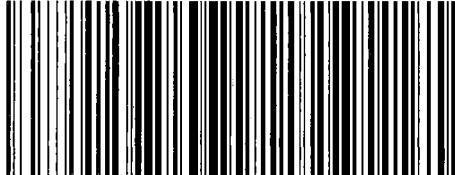
表 A.1 为 WIA-FA 网络所支持的安全分级及其对应安全措施,推荐使用 Grade 1。

**表 A.1 WIA-FA 网络的安全分级**

WIA-FA 网络安全级别	安全措施	安全等级
Grade 0	FCS	SecLevel = 0
Grade 1	FCS 设备认证	SecLevel = 1
Grade 2	FCS 设备认证 数据链路层安全	SecLevel = 2~8

## 参考文献

- [1] IEC 60559:1989, Binary floating-point arithmetic for microprocessor systems (previously designated IEC 559:1989).
- [2] IEEE Std. 1588—2002 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.
- [3] IEC 62657-2 Industrial Communication networks—Wireless Communication Networks—Part 2: Coexistence Management.
- [4] IETF RFC 1321 The MD5 Message-Digest Algorithm.
- [5] IEEE Std. 802.15.4-2016 Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 15.4: Wireless Medium Access Control (MAC) and physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).



GB/T 26790.2-2015

版权专有 偷权必究

\*

书号:155066·1-53418

定价: 106.00 元