

# Zigbee 协议栈中文说明

## 1.概述

### 1.1 解析 ZigBee 堆栈架构

ZigBee 堆栈是在 IEEE 802.15.4 标准基础上建立的，定义了协议的 MAC 和 PHY 层。ZigBee 设备应该包括 IEEE802.15.4(该标准定义了 RF 射频以及与相邻设备之间的通信)的 PHY 和 MAC 层，以及 ZigBee 堆栈层：网络层(NWK)、应用层和安全服务提供层。图 1-1 给出了这些组件的概况。

#### 1.1.1 ZigBee 堆栈层

每个 ZigBee 设备都与一个特定模板有关，可能是公共模板或私有模板。这些模板定义了设备的应用环境、设备类型以及用于设备间通信的簇。公共模板可以确保不同供应商的设备在相同应用领域中的互操作性。

设备是由模板定义的，并以应用对象(Application Objects)的形式实现(见图 1-1)。每个应用对象通过一个端点连接到 ZigBee 堆栈的余下部分，它们都是器件中可寻址的组件

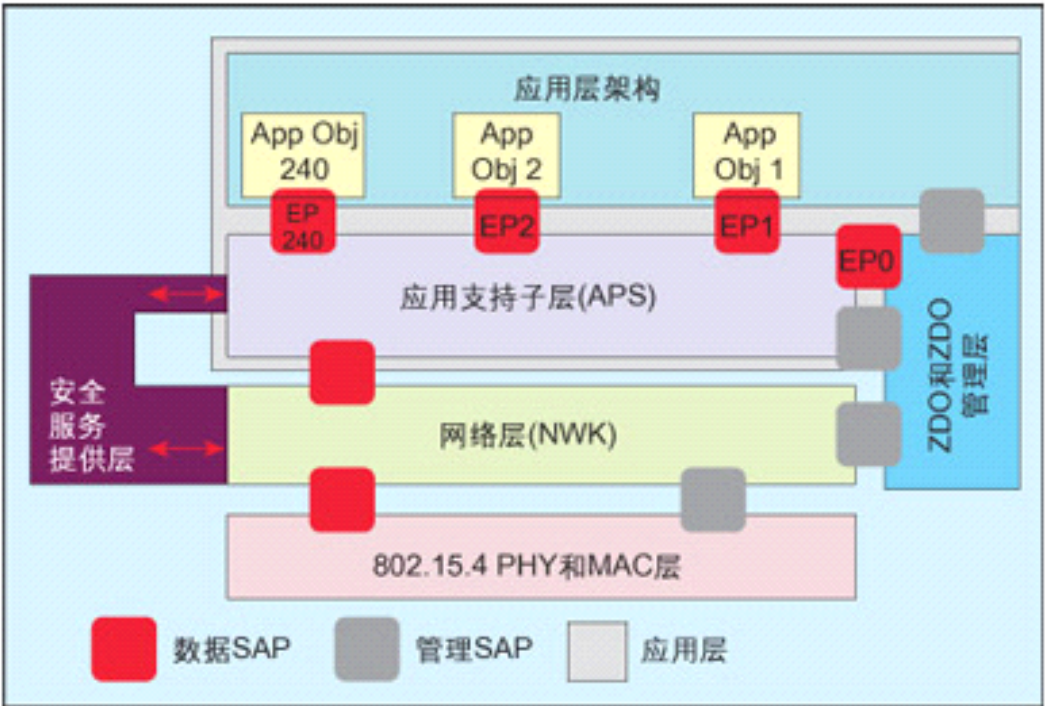


图 1-1 zigbe 堆栈框架

从应用角度看，通信的本质就是端到端的连接(例如，一个带开关组件的设备与带一个或多个灯组件的远端设备进行通信，目的是将这些灯点亮)。

端点之间的通信是通过称之为簇的数据结构实现的。这些簇是应用对象之间共享信息所需的全部属性的容器，在特殊应用中使用的簇在模板中有定义。图 1-1-2 就是设备及其接口的一个例子：

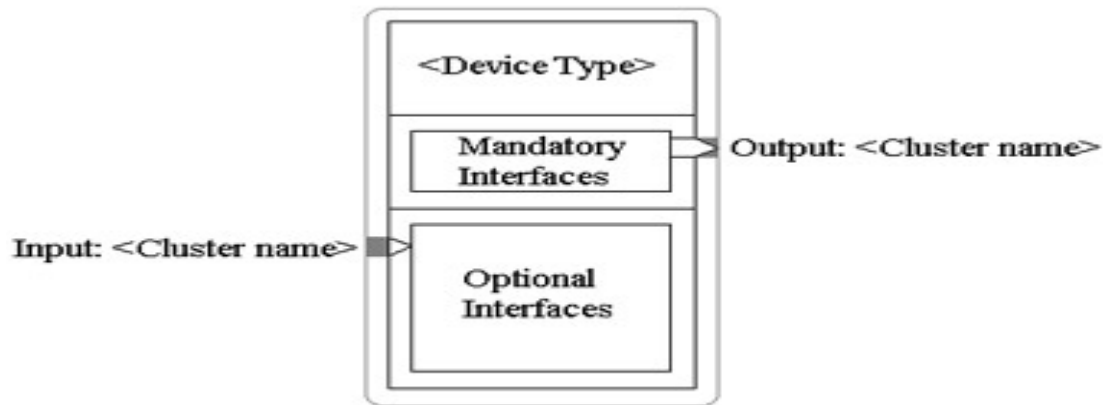


图 1-1-2

每个接口都能接收(用于输入)或发送(用于输出)簇格式的数据。一共有二个特殊的端点，即端点 0 和端点 255。端点 0 用于整个 ZigBee 设备的配置和管理。应用程序可以通过端点 0 与 ZigBee 堆栈的其它层通信，从而实现对这些层的初始化和配置。附属在端点 0 的对象被称为 ZigBee 设备对象(ZD0)。端点 255 用于向所有端点的广播。端点 241 到 254 是保留端点。

所有端点都使用应用支持子层(APS)提供的服务。APS 通过网络层和安全服务提供层与端点相接，并为数据传送、安全和绑定提供服务，因此能够适配不同但兼容的设备，比如带灯的开关。

APS 使用网络层(NWK)提供的服务。NWK 负责设备到设备的通信，并负责网络中设备初始化所包含的活动、消息路由和网络发现。应用层可以通过 ZigBee 设备对象(ZD0)对网络层参数进行配置和访问。

### 1.1.2 802.15.4 MAC 层

IEEE 802.15.4 标准为低速率无线个人域网(LR-WPAN)定义了 OSI 模型开始的两层。PHY 层定义了无线射频应该具备的特征，它支持二种不同的射频信号，分别位于 2450MHz 波段和 868/915MHz 波段。2450MHz 波段射频可以提供 250kbps 的数据速率和 16 个不同的信道。868/915MHz 波段中，868MHz 支持 1 个数据速率为 20kbps 的信道，915MHz 支持 10 个数据速率为 40kbps 的信道。

MAC 层负责相邻设备间的单跳数据通信。它负责建立与网络的同步，支持关联和去关联以及 MAC 层安全：它能提供二个设备之间的可靠链接。

### 1.1.3 关于服务接入点

ZigBee 堆栈的不同层与 802.15.4 MAC 通过服务接入点(SAP)进行通信。SAP 是某一特定层提供的服务与上层之间的接口。

ZigBee 堆栈的大多数层有两个接口：数据实体接口和管理实体接口。数据实体接口的目标是向上层提供所需的常规数据服务。管理实体接口的目标是向上层提供访问内部层参数、配置和管理数据的机制。

### 1.1.4 ZigBee 的安全性

安全机制由安全服务提供层提供。然而值得注意的是，系统的整体安全性是在模板级定义的，这意味着模板应该定义某一特定网络中应该实现何种类型的安全。

每一层(MAC、网络或应用层)都能被保护，为了降低存储要求，它们可以分享安全密钥。SSP 是通过 ZDO 进行初始化和配置的，要求实现高级加密标准(AES)。ZigBee 规范定义了信任中心的用途。信任中心是在网络中分配安全密钥的一种令人信任的设备。

### 1.1.5 ZigBee 堆栈容量和 ZigBee 设备

根据 ZigBee 堆栈规定的所有功能和支持，我们很容易推测 ZigBee 堆栈实现需要用到设备中的大量存储器资源。不过 ZigBee 规范定义了三种类型的设备，每种都有自己的功能要求：ZigBee 协调器是启动和配置网络的一种设备。协调器可以保持间接寻址用的绑定表格，支持关联，同时还能设计信任中心和执行其它活动。一个 ZigBee 网络只允许有一个 ZigBee 协调器。

ZigBee 路由器是一种支持关联的设备，能够将消息转发到其它设备。ZigBee 网络或树型网络可以有多个 ZigBee 路由器。ZigBee 星型网络不支持 ZigBee 路由器。

ZigBee 终端设备可以执行它的相关功能，并使用 ZigBee 网络到达其它需要与其通信的设备。它的存储器容量要求最少。然而需要特别注意的是，网络的特定架构会戏剧性地影响设备所需的资源。NWK 支持的网络拓扑有星型、树型和网格型。在这几种网络拓扑中，星型网络对资源的要求最低。

ZigBee 堆栈应该可以提供 ZigBee 规范要求的所有功能，因此制造商的重点工作是开发实际的应用。为了更加容易实现，如果制造商使用某种公共模板，那么可用大多数现成的配置。如果没有合适的公共模板，则可以充分利用其它模板已经做过的工作创建自己的模板。

ZigBee 协议栈体系包含一系列的层元件，其中有 IEEE802.15.4 2003 标准中的 MAC 层和 PHY 层，当然也包括 ZigBee 组织设计的 NWK 层。每个层的元件有其特定的服务功能。本说明描述内容涉及 ZigBee 协议栈的各层元件，但侧重于描述最具实际和理论探讨性的 APL 应用层和 NWK 网络层。图 1-1 为 ZigBee 栈结构框图。

## 2.APL 应用层介绍

### 2.1.1 应用层简介

如图 2-1 所示，ZigBee 应用层由三个部分组成，APS 子层、ZDO（包含 ZDO 管理平台）和制造商定义的应用对象。

Source: Zigbee Alliance 2004/10/15 **ZigBee Protocol Stack**

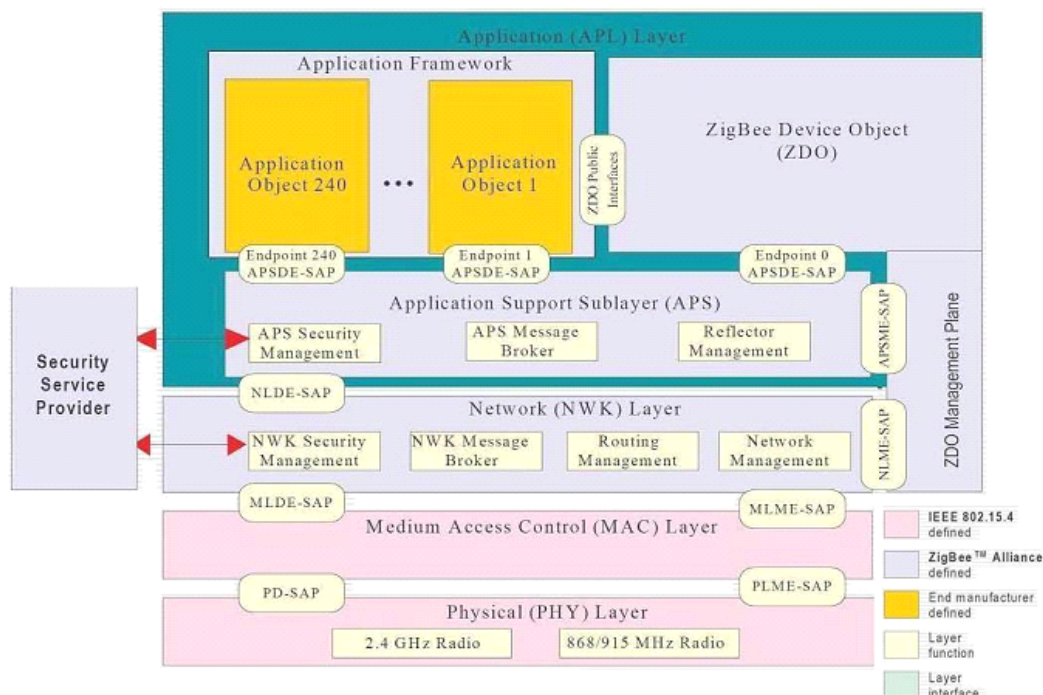


图 2-1 zigbee 协议堆栈分层结构

### 2.1.2 应用层框架

ZigBee 中的应用框架是为驻扎在 ZigBee 设备中的应用对象提供活动的环境。

最多可以定义 240 个相对独立的应用程序对象，且任何一个对象的端点编号都是从 1 到 240。此外还有两个附加的终端节点，为了 APSDE-SAP 的使用：端点号 0 固定用于 ZDO 数据接口；另外一个端点 255 固定用于所有应用对象广播数据的数据接口功能。端点 241-254 保留（留给未来扩展使用）。

#### 2.1.2.1 应用 Profiles

应用 profiles 是一组统一的消息，消息格式和处理方法，允许开发者建立一个可以共同使用的分布式应用程序，这些应用是利用驻扎在独立设备中的应用实体来实现的。这些应用 profiles 允许应用程序发送命令、请求数据和处理命令的请求。

#### 2.1.2.2 簇

簇标识符可用来区分不同的簇，簇标识符联系着从设备流出和向设备流入的数据。在特殊的应用 profiles 范围内，簇标识符是唯一的。

## 2.1.3 ZigBee 设备对象

ZigBee 设备对象(ZDO),描述了一个基本的功能函数,这个功能在应用对象、设备 profile 和 APS 之间提供了一个接口。ZDO 位于应用框架和应用支持子层之间。它满足所有在 ZigBee 协议栈中应用操作的一般需要。此外 ZDO 还有以下作用:

- (1) 初始化应用支持子层 (APS), 网络层 (NWK), 安全服务规范 (SSS)。
- (2) 从终端应用集合中配置的信息来确定和执行安全管理、发现、网络管理、以及绑定管理。

ZDO 描述了应用框架层中应用对象的公用接口以及控制设备和应用对象的网络功能。在终端节点 0, ZDO 提供了与协议栈中与低一层连接的接口, 如果是数据则通过 APSDE-SAP, 如果是控制信息则通过 APSME-SAP。Z D O 的具体描述在 2.5 节。

### 2.1.3.1 设备发现

设备发现是 ZigBee 设备为什么能发现其他设备的过程。这有两种形式的设备发现请求: IEEE 地址请求和网络地址请求。IEEE 地址请求是单播到一个特殊的设备且假定网络地址已经知道。网络地址请求是广播且携带一个已知的 IEEE 地址作为负载。

### 2.1.3.2 服务发现

服务发现是为什么一个已知设备被其他设备发现的能力的过程。服务发现通过在一个已知设备的每一个端点发送询问或通过使用一个匹配服务(广播或者单播)。服务发现方便定义和使用各种描述来概述一个设备的能力。

服务发现信息在网络中也许被隐藏,在这种情况下,设备提供的特殊服务便可能不在操作发生的时候到达。

## 2.2 ZigBee 应用支持子层 APS

APS 提供了这样的接口: 在 NWK 层和 APL 层之间, 从 ZDO 到供应商的应用对象的通用服务集。这服务由两个实体实现: APS 数据实体 (APSDE) 和 APS 管理实体 (APSME)。

(1)APSDE 提供在同一个网络中的两个或者更多的应用实体之间的数据通信。通过 APSDE 服务接入点(APSDE-SAP);

(2) APSME 提供多种服务给应用对象, 这些服务包含安全服务和绑定设备, 并维护管理对象的数据库, 也就是我们常说的 AIB。通过 APSME 服务接入点 (APSME-SAP)。

### 2.2.1 范围

这一小节描述了应用层部分提供的服务规范和生产商定义的应用对象与 ZigBee 设备对象之间的接口。规范定义了允许应用对象传输数据的数据服务和提供绑定机制的管理服务。另外, 它还定义了应用支持子层的帧格式和帧类型。如图 2-2



5		Sync Header	4 Preamble		ZigBee帧格式 更多内容见 <a href="http://www.c51rf.com">www.c51rf.com</a>			
			1 Start of Packet Delimiter					
1		PHY Header	7/8 Frame length					
			1/8 Reserve					
127	8	MHR MAC帧头包含当前的源和目标地址信息，注意如果在路途上，这不是确切的源和最终的目标。产生和应用这个帧头只为满足应用需要，应用程序并不受理这个数据区域	2	Frame contrl (FCF)	0~2	Frame Type		
					3	Security Enabled		
					4	Frame Pending		
					5	Acknowledge request		
					6	Intra PAN		
					7~9	Reserved		
					A~B	Destination addressing mode		
					C~D	Reserved		
					E~F	Source addressing mode		
					1 Sequence number			
					4	dst.panID		
					~	目标地址		
					20	src.panID		
					2或8	源地址		
127	8	NWK_HEADER (nwkCurrentFrame) NWK帧头包含确切的源和最终的目标地址信息，应用程序产生和应用这个帧头，还包含额外的源地址，可以通过其他宏定义识别源设备地址。	1	frameCONLSB	1	bits(Val)		
					0~1	type		
					2~5	version		
					6~7	discoverRoute		
					0			
					1	security		
					2~7			
					2	destAddr (最终目标)		
					2	srcAddr (确切的源)		
					1	broadcastRadius (允许的广播半径)		
					1	broadcastSequence		
					127	8	APS_HEADER (apsCurrentFrame) APS帧头包含当前信息的配置ID，集群ID和目标端口，宏定义提供了发送信息时创建此帧头的简化方法。处理接收的信息即可确定对应的端口。	1
2~3	deliveryMode							
4	indirectAddressMode							
5	security							
6	ackRequested							
7								
1 deliveryMode								
1 destEP								
1 clusterID (属性的集合)								
2 profileID (是对逻辑设备及其接口的简化描述)								
1 srcEP								
APSPayload								
2	MFR	Frame check Sequence						

图 2-2 zigbee 帧格式

## 2.2.2 目的

这节节的目的是定义 ZigBee 应用支持子层的功能。该功能建立在两个基础之上，一是正确运行 ZigBee 网络层的驱动功能，二是制造商定义的应用对象所需要的功能。

## 2.2.3 应用支持子层简介

应用支持子层给网络层和应用层通过 ZigBee 设备对象和制造商定义的应用对象使用的一组服务提供了接口，该接口提供了 ZigBee 设备对象和制造商定义的应用对象使用的一组服务。通过两个实体提供这些服务：数据服务和管理服务。APS 数据实体 (APSDE) 通过与之连接的 SAP，即 APSDE-SAP 提供数据传输服务。APS 管理实体 (APSME) 通过与之连接的 SAP，即 APSME-SAP 提供管理服务，并且维护一个管理实体数据库，即 APS 信息库 (NIB)。

### 2.2.3.1 应用支持子层的数据实体（APSDE）

APSDE 向网络层提供数据服务，并且为 ZDO 和应用对象提供服务，完成两个或多个设备之间传输应用层 PDU。这些设备本身必须在同一个网络。

APSDU 将提供如下服务：

生成应用层的协议数据单元（APDU）：APSDE 将应用层协议数据单元（PDU）加上适当的协议帧头生成应用子层的协议数据单元（PDU）。

绑定：两个设备服务和需求相匹配的能力。一旦两个设备绑定了，APSDE 将可以把从一个绑定设备接受到的信息传送给另一个设备。

组地址过滤：提供了基于终点组成员的过滤组地址信息的能力。

可靠传输：比从网络层仅仅通过端对端的传输增加了可靠性

拒绝重复：提供传送的信息不会被重复接收

支持大批量的传输：提供两个设备间顺序传输大批量的数据的能力。

碎片：当消息的长度大于单个网络层帧时，可以分割并重组消息。

流控制：APS 提供避免传输消息淹没接收者的措施。

阻塞控制：APS 层使用“尽力”原则，提供措施避免传输消息淹没中间网络。

### 2.2.3.2 应用支持子层的管理实体（APSME）

APSME 应提供管理服务支持应用程序符合堆栈。

APSME 应具有基于两个设备的服务和需求相匹配的能力。该服务称为绑定服务，APSME 应具有能力来构建和维护绑定表来存储这些信息。

另外，APSME 应提供如下服务：

- 1 应用层信息库管理：读取与设置设备应用层信息库属性的能力
- 2 安全：与其他设备通过使用安全密钥建立可信关系的能力

### 2.2.4 服务规范

应用支持子层为上层实体（NHLE）与网络层提供了一个接口。APS 层理论上包含一个管理实体称为 APS 层，管理实体（APSME）。这个实体通过调用子层的管理函数来提供服务接口。APSME 还负责维护一个关于 APS 子层管理实体的数据库。这是一个关于 APS 子层信息库（AIB）的数据库。图 2-3 描述了 APS 子层的构成和接口。

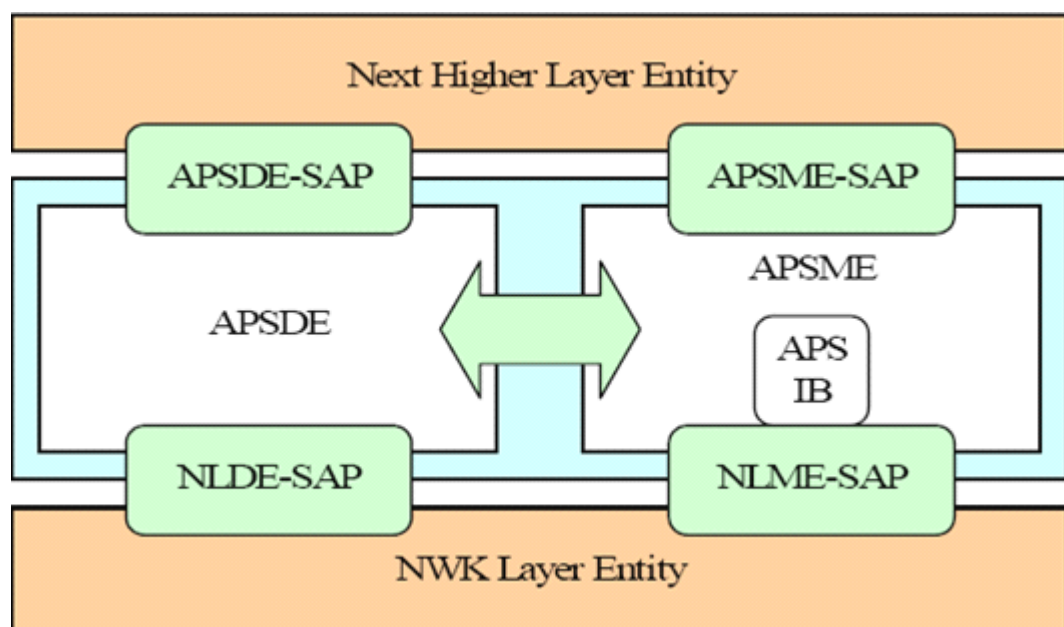


图 2-3 应用支持之层参考模型

APS子层通过两个服务指针（SAPs）提供两种服务。APS数据服务通过APS子层数据实体服务指针SAP（APSDE-SAP），APS管理服务通过APS子层管理实体服务指针SAP（APSME-SAP）。

这两个服务通过NLDE-SAP和NLME-SAP 接口（见3. 2小节）提供了NHLE和网络层之间的接口。网络层和APS子层之间的NLME-SAP接口只支持NLME-GET 和 NLME-SET原语，其他的NLME-SAP原语只可以通过ZDO实现（见2. 5小节）。除了这些外部接口以外，在APSME和APSDE之间还有一个内部的接口，支持APSME使用APS数据服务。

### 2.2.4.1 APS 数据服务

APS子层数据实体SAP（APSDE-SAP）支持在两个同等的的应用实体之间传输应用协议数据单元。表2-1列出了APSDE-SAP支持的原语。每一个原语将在下面的小节论述。

Table 2.1 APSDE-SAP Primitives

APSDE-SAP Primitive	Request	Confirm	Indication
APSDE-DATA	2.2.4.1.1	2.2.4.1.2	2.2.4.1.3

#### 2.2.4.1.1 APSDE-DATA.request

该原语请求从本地NHLE向一个同等的NHLE实体传输NHLE PDU (ASDU)。

##### 2.2.4.1.1.1 服务原语的语法

该原语的语法如下：

```
APSDE-DATA_request
{
    DstAddrMode
    DSTAddress
    DstEndpoint
    Profiled
    ClusterId
    SrcEndpoint
    asduLength
    asdu
    TxOptions
    RadiusCounter
}
```

表2. 2详细说明了APSDE-DATA.request原语的参数。



**Table 2.2 APSDE-DATA.request Parameters**

Name	Type	Valid Range	Description
DstAddrMode	Integer	0x00 – 0xff	The addressing mode for the source address used in this primitive and of the APDU to be transferred. This parameter can take one of the non-reserved values from the following list:  0x00 = DstAddress and DstEndpoint not present 0x01 = 16-bit group address for DstAddress and DstEndpoint not present 0x02 = 16-bit address for DstAddress and DstEndpoint present 0x03 = 64-bit extended address for DstAddress and DstEndpoint present 0x04 – 0xff = reserved
DstAddress	Address	As specified by the DstAddrMode parameter	The individual device address or group address of the entity to which the ASDU is being transferred
DstEndpoint	Integer	0x00 – 0xff	This parameter shall be present if, and only if, the DstAddrMode parameter has a value of 0x02 or 0x03 and, if present, shall be either the number of the individual endpoint or the broadcast endpoint (0xff) of the entity to which the ASDU is being transferred
ProfileId	Integer	0x0000 – 0xffff	The identifier of the profile for which this frame is intended
ClusterId	Integer	0x0000 – 0xffff	The identifier of the object to use in the binding operation if the DstAddrMode parameter has a value of 0x00. If the DstAddrMode parameter has any other value than 0x00 then this parameter is ignored
SrcEndpoint	Integer	0x00 – 0xfe	The individual endpoint of the entity from which the ASDU is being transferred
asduLength	Integer		The number of octets comprising the ASDU to be transferred

**Table 2.2 APSDE-DATA.request Parameters**

Name	Type	Valid Range	Description
Asdu	Set of octets	-	The set of octets comprising the ASDU to be transferred
TxOptions	Bitmap	0000 xxxx (Where x can be 0 or 1)	The transmission options for the ASDU to be transferred. These are a bitwise OR of one or more of the following:  0x01 = Security enabled transmission 0x02 = Use NWK key 0x04 = Acknowledged transmission 0x08 = Fragmentation permitted
Radius	Unsigned Integer	0x00-0xff	The distance, in hops, that a transmitted frame will be allowed to travel through the network

## 2.2.4.1.1.2 产生

当有一个数据PDU (ASDU) 由本地NHLE向一个同等的NHLE传输时, 由本地NHLE生成该原语。

## 2.2.4.1.1.3 2

当APS子层实体接收到该原语时, 便开始传输提供的ASDU。

如果DstAddrMode参数为0x00, 并且接收该原语的设备的APSDE支持绑定表, 那么在绑定表中根据参数SrcEndpoint和ClusterId所指定的endpoint和cluster identifiers寻找相关联的绑定表入口。如果没有绑定表入口, APSDE将发送状态参数为NO\_BOUND\_DEVICE 的语APSDE-DATA.confirm原语。如果找到了一个或多个绑定表入口, APSDE将构建APDU, 其endpoint信息从绑定表入口获得, 当通过网络层传输信息帧时, 其destination address信息从绑定表入口获得。如果存在多于一个绑定表入口, 当接收到相应的NLDE-DATA.confirm原语, 按上面描述的, APSDE将构建并向下一个绑定表入口传输APDU, 直到没有绑定表入口剩余。如果接收到该原语设备的APSDE不支持绑定表, 那么APSDE将发送状态参数为NOT\_SUPPORTED的APSDE-DATA.confirm原语。

如果DstAddrMode参数为0x02, DstAddress参数包含扩展的64位IEEE地址, 首次必须使用NIB (见表2. 24) 属性中的nwkAddressMap映射相应的16位网络地址。如果找不到相应的16位网络地址, 那么APSDE将发送状态参数为NO\_SHORT\_ADDRESS的APSDE-DATA.confirm原语。如果找到了相应的16位网络地址, 其值将被用在NLDE-DATA.request原语中, 参数DstEndpoint将被置在作为结果的APDU中。如果DstAddrMode参数为0x01, 表明为群地址, 参数DstAddress将被解释为16位的全地址。这个地址将被放置在APS头中的群地址域, 参数DstEndpoint将被忽略, APS头中的destination endpoint域将被省略。APS头中的帧控制域的delivery mode子域值在这种情况下为0x03。

如果DstAddrMode参数为0x02, DstAddress参数包含16位的网络地址, 并且提供参数DstEndpoint, 当目的网络地址用于应用响应, 并且网络地址部位后面的数据传输请求保留时, 上层只能使用DstAddrMode为0x02。

应用程序可以通过使用参数RadiusCounter来限制在网络中传输数据帧的跳数。如果参数RadiusCounter为0x00, 网络层在网络中传输信息帧没有约束。如果参数RadiusCounter为非零, 则网络层将允许信息帧在网络中传输存在最多RadiusCounter跳。

如果DstAddrMode参数为0x01, 表明为群地址, 或者DstAddrMode参数为0x00, 并且相应的绑定表入口包含哪一个群地址, 那么APSME将检查NIB (见表3. 42) 中的属性nwkUseMulticast值。如果属性值为FALSE, 那么输出帧的帧控制域中的delivery mode子域设为0b11, 16位的目的群地址将设置输出帧APS头中的group address域, 该帧将以广播方式传输。传输该帧的原语NLDE-DATA.request的DstAddr参数设置为值0xffffd, 广播给所有RxOnWhenIdle=TRUE的设备。如果属性nwkUseMulticast值为TRUE, 那么该帧将使用网络层多点传送方式传输, 群地址不用放置在输出帧的APS头中。

如果参数TxOptions指定使用安全传输, 则APS子层将使用安全服务为ASDU提供安全 (见4. 2. 4小节)。如果安全处理失败, 则APSDE发送状态参数为SECURITY\_FAIL的APSDE-DATA.confirm原语。

APSDE使用NLDE-DATA.request原语向网络层传输构造帧。当接收到NLDE-DATA.confirm原语, APSDE则发送APSDE-DATA.confirm原语, 其状态参数值域从网络层接收到的一致。

APSDE通过每次发送使NLDE-DATA.request原语的DiscoverRoute参数值为0x01确保网络层中的路由发现始终激活。

如果传输的ASDU大于合适的单个帧, 当没有请求确认传输或者在TxOptions域的

fragmentation permitted标志位设为0时，则放弃传输ASDU，APSDE将发送状态参数为INVALID\_REQUEST的APSDE-DATA.confirm原语。

如果传输的ASDU大于合适的单个帧，当请求确认传输并且在TxOptions域的fragmentation permitted标志位设为1时，ASDU将按照2.2.8.3.5小节所述分裂为多个APDU。如果请求传输和安全处理，那么每一个APDU都要进行处理。注意不要使用分裂处理，除非相应的上层文件或者相互明确表明帧的传输允许分裂处理，并且说明了块的数量和总共传输的大小。

### 2.2.4.1.2 APSDE-DATA.confirm

该原语报告从本地NHLE向一个同等的NHLE传输PDU数据的结果。

#### 2.2.4.1.2.1 服务原语的语法

该原语的语法如下：

```
APSDE-DATA.confirm  {
    DstAddMode
    DstAddress
    DstEndpoint
    SrcEndpoint
    Status
}
```

表2.3详细介绍了APSDE-DATA.confirm原语的参数。

**Table 2.3 APSDE-DATA.confirm Parameters**

Name	Type	Valid Range	Description
DstAddrMode	Integer	0x00 – 0xff	The addressing mode for the source address used in this primitive and of the APDU to be transferred. This parameter can take one of the non-reserved values from the following list:  0x00 = DstAddress and DstEndpoint not present 0x01 = 16-bit group address for DstAddress and DstEndpoint not present 0x02 = 16-bit address for DstAddress and DstEndpoint present 0x03= 64-bit extended address for DstAddress and DstEndpoint present 0x04 – 0xff = reserved
DstAddress	Address	As specified by the DstAddrMode parameter	The individual device address or group address of the entity to which the ASDU is being transferred
DstEndpoint	Integer	0x00 – 0xff	This parameter shall be present if, and only if, the DstAddrMode parameter has a value of 0x02 or 0x03 and, if present, shall be the number of the individual endpoint of the entity to which the ASDU is being transferred
SrcEndpoint	Integer	0x00 – 0xfe	The individual endpoint of the entity from which the ASDU is being transferred
Status	Enumeration	SUCCESS, NO_SHORT_ADDRESS , NO_BOUND_DEVICE, SECURITY_FAIL, NO_ACK or any status values returned from the NLDE-DATA.confirm primitive	The status of the corresponding request

### 2.2.4.1.2.2 产生

该原语有本地APS子层产生作为对APSDE-DATA.request原语的响应。该原语返回的状态参数值为SUCCESS，表明请求传输成功，或者为错误代码NO\_SHORT\_ADDRESS，NO\_BOUND\_DEVICE 或SECURITY\_FAIL或者为任何NLDE-DATA.confirm原语返回的状态值。这些状态值的路由在2.2.4.1.2小节中进行了详细的描述。

### 2.2.4.1.2.3 接收

接收到该原语，发起设备的上层被通报请求传输的结果。如果传输成功，状态参数值设置为SUCCESS。否则，状态参数表明错误。

### 2.2.4.1.3 APSDE-DATA.indication

该原语表明一个PDU数据向本地应用实体的APS子层传输。

#### 2.2.4.1.3.1 服务原语的语法

该原语的语法如下：

```
APSDE-DATA.indication
{
    DstAddrMode
    DSTAddress
    DstEndpoint
    SrcAddrMode
    SARCAAddress
    SrcEndpoint
    Profield
    ClusterId
    asduLength
    asdu
    WasBroadcast
    SecurityStatus
    LinkQuality
}
```

表2. 4详细描述了APSDE-DATA.indication原语的参数。



**Table 2.4 APSDE-DATA.indication Parameters**

Name	Type	Valid Range	Description
DstAddrMode	Integer	0x00 - 0xff	The addressing mode for the destination address used in this primitive and of the APDU that has been received. This parameter can take one of the non-reserved values from the following list:  0x00 = reserved 0x01 = 16-bit group address for DstAddress and DstEndpoint not present 0x02 = 16-bit address for DstAddress and DstEndpoint present 0x03 – 0xff = reserved
DstAddress	Address	As specified by the DstAddrMode parameter	The individual device address or group address to which the ASDU is directed
DstEndpoint	Integer	0x00 – 0xff	The target endpoint on the local entity from which the ASDU has been received
SrcAddrMode	Integer	0x00 – 0xff	The addressing mode for the source address used in this primitive and of the APDU that has been received. This parameter can take one of the non-reserved values from the following list:  0x00 = SrcAddress and SrcEndpoint not present 0x01 = reserved 0x02 = 16-bit short address for SrcAddress and SrcEndpoint present 0x03 = 64-bit extended address for SrcAddress and SrcEndpoint present 0x04 – 0xff = reserved
SrcAddress	Address	As specified by the SrcAddrMode parameter	The individual device address address of the entity from which the ASDU has been received
SrcEndpoint	Integer	0x00 – 0xfe	This parameter shall be present if, and only if, the SrcAddrMode parameter has a value of 0x02 or 0x03 and, if present, shall be the number of the individual endpoint of the entity from which the ASDU has been received
ProfileId	Integer	0x0000 - 0xffff	The identifier of the profile from which this frame originated

**Table 2.4 APSDE-DATA.indication Parameters**

Name	Type	Valid Range	Description
ClusterId	Integer	0x0000-0xffff	The identifier of the received object
asduLength	Integer		The number of octets comprising the ASDU being indicated by the APSDE
asdu	Set of octets	-	The set of octets comprising the ASDU being indicated by the APSDE
WasBroadcast	Boolean	TRUE or FALSE	TRUE if the transmission was a broadcast, FALSE otherwise
SecurityStatus	Enumeration	UNSECURED, SECURED_NWK_KEY, SECURED_LINK_KEY	UNSECURED if the ASDU was received without any security SECURED_NWK_KEY if the received ASDU was secured with the NWK key SECURED_LINK_KEY if the ASDU was secured with a link key
LinkQuality	Integer	0x00 - 0xff	The link quality indication delivered by the NLDE

### 2.2.4.1.3.2 产生

该原语由APS子层产生，当从本地网络层实体接收到适当地址的数据帧时，APS子层向上层发送该原语。如果ASDU头的帧控制域表明该帧安全保护，则按照4.2.4小节的描述进行安全处理。

该原语由APS子层产生，当通过NLDE-DATA.indication原语从网络层接收到适当地址的数据帧时，发送给上层实体。如果APDU头的帧控制域表明该帧安全保护，则按照4.2.4小节的描述进行安全处理。

接收到的帧的源地址必须通过NIB（见表2.24）中的属性nwkAddressMap映射为相应的扩展的64位IEEE地址。如果能找到相应的64位IEEE地址，则APSDE发送该原语，其参数SrcAddrMode设为0x02，SrcAddress参数设为相应的64位IEEE地址。如果找不到相应的64位IEEE地址，APSDE将发送该原语，其参数SrcAddrMode设为0x01，参数SrcAddress设为接收帧包含的16位源地址。

### 2.2.4.1.3.3 接收

接收到该原语，上层被通报有数据到达该设备。

## 2.2.4.2 APS管理服务

APS管理实体SAP (APSME-SAP) 支持上层和APSME层之间传输管理命令。表2.5总结了APSME通过APSME-SAP接口支持的原语。各原语的详细描述见下面小节。

**Table 2.5 Summary of the Primitives Accessed Through the APSME-SAP**

Name	Request	Indication	Response	Confirm
APSME-ADD-GROUP	2.2.4.5.1			2.2.4.5.2
APSME-BIND	2.2.4.3.1			2.2.4.3.2
APSME-GET	2.2.4.4.1			2.2.4.4.2
APSME-REMOVE-ALL-GROUPS	2.2.4.5.5			2.2.4.5.6
APSME-REMOVE-GROUP	2.2.4.5.3			2.2.4.5.4
APSME-SET	2.2.4.4.3			2.2.4.4.4
APSME-UNBIND	2.2.4.3.3			2.2.4.3.4

### 2.2.4.3 绑定原语

这组原语定义了设备上层如何将一个绑定记录加入（提交）其本地绑定表或将绑定记录从本地绑定表中移除。

只有支持绑定表或者绑定表存储器的设备支持这些原语。如果其他设备从上层接收到这些原语，那么这些原语将被忽略。

#### 2.2.4.3.1 APSME-BIND.request

该原语允许支持绑定的设备上层通过在本地绑定表中建立一个入口请求将两个设备绑定。

#### 2.2.4.6.1.1 服务原语的语法

该原语的语法如下：

APSME-BIND.request

{  
SrcAddr  
SrcEndpoint  
ClusterId  
DstAddrMode  
DstAddr  
DstEndpoint  
}

表2.6详细描述了APSME-BIND.request原语的参数。

**Table 2.6 APSME-BIND.request Parameters**

Name	Type	Valid Range	Description
SrcAddr	IEEE address	A valid 64-bit IEEE address	The source IEEE address for the binding entry
SrcEndpoint	Integer	0x01 – 0xff	The source endpoint for the binding entry
ClusterId	Integer	0x0000 – 0xffff	The identifier of the cluster on the source device that is to be bound to the destination device
DstAddrMode	Integer	0x00 – 0xff	The addressing mode for the source address used in this primitive. This parameter can take one of the non-reserved values from the following list:  0x00 = reserved 0x01 = 16-bit group address for DstAddr and DstEndpoint not present 0x02 = reserved 0x03 = 64-bit extended address for DstAddr and DstEndpoint present 0x04 – 0xff = reserved
DstAddr	Address	As specified by the DstAddrMode parameter	The destination address for the binding entry
DstEndpoint	Integer	0x01 – 0xff	This parameter will be present only if the DstAddrMode parameter has a value of 0x03 and, if present, will be the destination endpoint for the binding entry

### 2.2.4.3.1.2 产生

该原语由上层产生发送给APS子层，在支持绑定表的设备上发起绑定操作。

### 2.2.4.3.1.3 接收

一旦被当前没有加入到网络或不支持绑定表的设备接收到该原语，那么APSME将发送状态参数为ILLEGAL\_REQUEST的APSME-BIND.confirm原语。

如果支持绑定表的设备的APS子层从NHLE接收该原语，APSME将试图直接从其绑定表中建立指定的入口。如果可以建立入口，APSME将发送状态参数为SUCCESS的APSME-BIND.confirm原语。如果因为其绑定表缺乏能力而无法建立入口，APSME将发送状态参数为TABLE\_FULL的APSME-BIND.confirm原语。

### 2.2.4.3.2 APSME-BIND.confirm

该原语使设备得到其上层请求绑定两个设备的结果。

#### 2.2.4.3.2.1 服务原语的语法

该原语的语法如下：

```
APSME-BIND.confirm {
    Status
```

```

SrcAddr
SrcEndpoint
ClusterId
DstAddrMode
DstAddr
DstEndpoint
}

```

表2.7详细描述了APSME-BIND.confirm原语的语法。

**Table 2.7 APSME-BIND.confirm Parameters**

Name	Type	Valid Range	Description
Status	Enumeration	SUCCESS, ILLEGAL_DEVICE, ILLEGAL_REQUEST, TABLE_FULL, NOT_SUPPORTED	The results of the binding request
SrcAddr	IEEE address	A valid 64-bit IEEE address	The source IEEE address for the binding entry
SrcEndpoint	Integer	0x01 – 0xff	The source endpoint for the binding entry
ClusterId	Integer	0x0000 – 0xffff	The identifier of the cluster on the source device that is to be bound to the destination device
DstAddrMode	Integer	0x00 – 0xff	The addressing mode for the source address used in this primitive. This parameter can take one of the non-reserved values from the following list :  0x00 = reserved 0x01 = 16-bit group address for DstAddr and DstEndpoint not present 0x02 = reserved 0x03 = 64-bit extended address for DstAddr and DstEndpoint present 0x04 – 0xff = reserved
DstAddr	Address	As specified by the DstAddrMode parameter	The destination address for the binding entry
DstEndpoint	Integer	0x01 – 0xff	This parameter will be present only if the DstAddrMode parameter has a value of 0x03 and, if present, will be the destination endpoint for the binding entry

## 2.2.4.3.2.2 产生

该原语由APSME产生作为APSME-BIND.request原语的响应发送给NHLE。如果请求成功，那么状态参数将表明一个成功的绑定请求。否则，状态参数则为错误码ILLEGAL\_DEVICE、ILLEGAL\_REQUEST 或TABLE\_FULL。



2.2.4.3.2.3 接收

接收到该原语，上层就被通知其绑定请求的结果。如果绑定请求成功，状态参数设置为SUCCESS。否则，状态参数表明错误。

2.2.4.3.3 APSME-UNBIND.request

该原语允许支持绑定的设备上层通过在本本地绑定表中移除一个入口请求将两个设备解除绑定。

2.2.4.3.3.1 服务原语的语法:

```
APSME-UNBIND.request      {
                             SrcAddr
                             SrcEndpoint
                             ClusterId
                             DstAddrMode
                             DstAddr
                             DstEndpoint
                             }
```

表2. 8详细描述了APSME-UNBIND.request原语的参数。

Table 2.8 APSME-UNBIND.request Parameters

Name	Type	Valid Range	Description
SrcAddr	IEEE address	A valid 64-bit IEEE address	The source IEEE address for the binding entry
SrcEndpoint	Integer	0x01 – 0xff	The source endpoint for the binding entry
ClusterId	Integer	0x0000 – 0xffff	The identifier of the cluster on the source device that is bound to the destination device

**Table 2.8 APSME-UNBIND.request Parameters**

Name	Type	Valid Range	Description
DstAddrMode	Integer	0x00 – 0xff	The addressing mode for the source address used in this primitive. This parameter can take one of the non-reserved values from the following list :  0x00 = reserved 0x01 = 16-bit group address for DstAddr and DstEndpoint not present 0x02 = reserved 0x03 = 64-bit extended address for DstAddr and DstEndpoint present 0x04 – 0xff = reserved
DstAddr	Address	As specified by the DstAddrMode parameter	The destination address for the binding entry
DstEndpoint	Integer	0x01 – 0xff	This parameter will be present only if the DstAddrMode parameter has a value of 0x03 and, if present, will be the destination endpoint for the binding entry

### 2.2.4.3.3.2 产生

该原语有上层产生发送给APS子层，在支持绑定表的设备上发起解除绑定操作。

### 2.2.4.3.3.3 接收

一旦被当前没有加入到网络或不支持绑定表的设备接收到该原语，那么APSME将发送状态参数为ILLEGAL\_REQUEST的APSME-UNBIND.confirm原语。

如果支持绑定表的设备的APS子层从NHLE接收该原语，APSME将在绑定表中查找指定的入口。如果入口存在，APSME将移除这个入口并发送状态参数为SUCCESS的APSME-UNBIND.confirm原语（见2.2.4.3.4小节）。如果没有找到入口，APSME将发送状态参数为INVALID\_BINDING的APSME-UNBIND.confirm原语。如果该设备不在网络中，APSME将发送状态参数为ILLEGAL\_DEVICE的APSME-BIND.confirm原语。

### 2.2.4.3.4 APSME-UNBIND.confirm

该原语使设备得到其上层请求解除两个设备绑定的结果。

#### 2.2.4.3.4.1 服务原语的语法

该原语的语法如下：

```
APSME-UNBIND.confirm    {  
                           Status  
                           SrcAddr  
                           SrcEndpoint  
                           ClusterId  
                           DstAddrMode
```

```

DstAddr
DstEndpoint
}

```

表2.9详细描述了APSME-UNBIND.confirm原语的语法。

**Table 2.9 APSME-UNBIND.confirm Parameters**

Name	Type	ValidRange	Description
Status	Enumeration	SUCCESS, ILLEGAL_DEVICE, ILLEGAL_REQUEST, INVALID_BINDING	The results of the unbind request
SrcAddr	IEEE address	A valid 64-bit IEEE address	The source IEEE address for the binding entry
SrcEndpoint	Integer	0x01 – 0xff	The source endpoint for the binding entry
ClusterId	Integer	0x0000 – 0xffff	The identifier of the cluster on the source device that is bound to the destination device

**Table 2.9 APSME-UNBIND.confirm Parameters**

Name	Type	ValidRange	Description
DstAddrMode	Integer	0x00 – 0xff	The addressing mode for the source address used in this primitive. This parameter can take one of the non-reserved values from the following list:  0x00 = reserved 0x01 = 16-bit group address for DstAddr and DstEndpoint not present 0x02 = reserved 0x03 = 64-bit extended address for DstAddr and DstEndpoint present 0x04 – 0xff = reserved
DstAddr	Address	As specified by the DstAddrMode parameter	The destination address for the binding entry
DstEndpoint	Integer	0x01 – 0xff	The destination endpoint for the binding entry

## 2.2.4.3.4.2 产生

该原语由APSME产生作为APSME-UNBIND.request原语的响应发送给NHLE。如果请求成功，那么状态参数将表明一个成功的解除绑定请求。否则，状态参数则为错误码ILLEGAL\_DEVICE、ILLEGAL\_REQUEST 或INVALID\_BINDING。

2.2.4.3.4.3 接收

接收到该原语，上层就被通知其解除绑定请求的结果。如果解除绑定请求成功，状态参数设置为SUCCESS。否则，状态参数表明错误。

2.2.4.4 信息库的维护

这组原语定义了设备上层如何读取和写入AIB中的属性。

2.2.4.4.1 APSME-GET.request

该原语允许设备上层从AIB中读取属性值。

2.2.4.4.1.1 服务原语的语法

该原语的语法如下：

```
APSME-GET.request      {
                        AIBAttribute
                        }
```

表2. 10描述了该原语的参数。

Table 2.10 APSME-GET.request Parameters

Name	Type	Valid Range	Description
AIBAttribute	Integer	See Table 2.24	The identifier of the AIB attribute to read

2.2.4.4.1.2 产生

该原语由上层产生并发送给APSME来读取AIB中的属性。

2.2.4.4.1.3 接收

接收到该原语，APSME试图从数据库中得到AIB属性。如果在数据库中没有相应的AIB属性表标识符，APSME将发送状态参数为UNSUPPORTED\_ATTRIBUTE的APSME-GET.confirm原语。

如果成功得到了AIB属性，APSME将发送状态参数为SUCCESS，包含AIB属性标识符和属性值的APSME-GET.confirm原语。

2.2.4.4.2 APSME-GET.confirm

该原语向上层报告从AIB中读取属性值的结果。

2.2.4.4.2.1 服务原语的语法

该原语的语法如下：

```
APSME-GET.confirm      {
                        Status
                        AIBAttribute
                        AIBAttributeLength
                        }
```

```
AIBAttributeValue
}
```

表2. 11描述了该原语的参数。

**Table 2.11 APSME-GET.confirm Parameters**

Name	Type	Valid Range	Description
Status	Enumeration	SUCCESS or UNSUPPORTED_ATTRIBUTE	The results of the request to read an AIB attribute value
AIBAttribute	Integer	See Table 2.24	The identifier of the AIB attribute that was read
AIBAttributeLength	Integer	0x0000 - 0xffff	The length, in octets, of the attribute value being returned
AIBAttributeValue	Various	Attribute Specific (see Table 2.24)	The value of the AIB attribute that was read

2. 2. 4. 4. 2. 2 产生

该原语由APSME产生，发送给上层作为对APSME-GET.request原语的响应。该原语返回状态SUCCESS，表明请求读取AIB属性请求成功，或者返回错误码UNSUPPORTED\_ATTRIBUTE. 这些状态在2. 2. 4. 4. 1. 3小节进行了描述。

2. 2. 4. 4. 2. 3 接收

接收到该原语，上层得知读取AIB属性请求的结果。如果读取AIB属性请求成功，状态参数设置为SUCCESS。否则，状态参数表明错误。

2. 2. 4. 4. 3 APSME-SET.request

该原语允许设备上层将属性值写入AIB。

2. 2. 4. 4. 3. 1 服务原语的语法

该原语的语法如下：

```
APSME-SET.request      {
                        AIBAttribute
                        AIBAttributeLength
                        AIBAttributeValue
                        }
```

表2. 12描述了该原语的参数。



**Table 2.12 APSME-SET.request Parameters**

Name	Type	Valid Range	Description
AIBAttribute	Integer	See Table 2.24.	The identifier of the AIB attribute to be written.
AIBAttributeLength	Integer	0x0000 - 0xffff	The length, in octets, of the attribute value being set
AIBAttributeValue	Various	Attribute Specific (see Table 2.24).	The value of the AIB attribute that should be written.

## 2.2.4.4.3.2 产生

该原语由上层产生并发送给APSME在AIB中写入一个属性值。

## 2.2.4.4.3.3 接收

接收到该原语，APSME试图将给定的数据库中的值写入AIB属性。如果在数据库中没有AIB属性参数指定的属性，APSME将发送状态参数为UNSUPPORTED\_ATTRIBUTE的APSME-SET.confirm原语。如果AIB属性值参数给定的值超过了有效的属性范围，APSME将发送状态参数为INVALID\_PARAMETER的APSME-SET.confirm原语。

如果成功写入了AIB属性，APSME将发送状态参数为SUCCESS的APSME-SET.confirm原语。

## 2.2.4.4.4 APSME-SET.confirm

该原语向上层报告向AIB属性中写入属性值的结果。

## 2.2.4.4.4.1 服务原语的语法

该原语的语法如下：

```
APSME-SET.confirm      {  
                        Status  
                        AIBAttribute  
                        }
```

表2.13描述了该原语的参数。

**Table 2.13 APSME-SET.confirm Parameters**

Name	Type	Valid Range	Description
Status	Enumeration	SUCCESS, INVALID_PARAMETER or UNSUPPORTED_ATTRIBUTE	The result of the request to write the AIB Attribute
AIBAttribute	Integer	See Table 2.24.	The identifier of the AIB attribute that was written

## 2.2.4.4.4.2 产生

该原语由APSME产生，发送给上层作为对APSME-SET.request原语的响应。该原语返回状态SUCCESS，表明将属性值写入AIB属性的请求成功，或者返回错误码INVALID\_PARAMETER或

UNSUPPORTED\_ATTRIBUTE. 这些状态在2. 2. 4. 4. 3. 3小节进行了描述。

### 2. 2. 4. 4. 3 接收

接收到该原语，上层得知写入AIB属性请求的结果。如果写入AIB属性请求成功，状态参数设置为SUCCESS。否则，状态参数表明错误。

### 2. 2. 4. 5 组管理

这组原语允许上层在当前设备中通过在组表中添加和移除入口来管理每个端点的组关系。

#### 2. 2. 4. 5. 1 APSME-ADD-GROUP.request

该原语允许上层请求一个特定的组的组关系加入到特定的端点。

##### 2. 2. 4. 5. 1. 1 服务原语的语法

该原语的语法如下：

```
APSME-ADD-GROUP.request      {
                                GroupAddress
                                Endpoint
                                }
```

表2. 14描述了该原语的参数。

**Table 2.14 APSME-ADD-GROUP.request Parameters**

Name	Type	Valid Range	Description
GroupAddress	16-bit group address	0x0000 - 0xffff	The 16-bit address of the group being added
Endpoint	Integer	0x01 - 0xf0	The endpoint to which the given group is being added

##### 2. 2. 4. 5. 1. 2 产生

当上层要将一个特定组的关系加入一个端点时产生该原语，设置了组地址的帧将被传送给该端点。

##### 2. 2. 4. 5. 1. 3 接收

如果接收到该原语，其GroupAddress参数的值超出了有效范围，APSME将向上层发送状态参数为INVALID\_PARAMETER的APSME-ADD-GROUP.confirm原语。同样，如果Endpoint参数值为0x00或当前设备的其它没有执行的端点，APSME将发送状态参数为INVALID\_PARAMETER的APSME-ADD-GROUP.confirm原语。

完成上述参数检测后，APSME将检查组表中是否存在包含给定参数GroupAddress和Endpoint的入口。如果该入口已存在于组表中，APSME将向上层发送状态参数为SUCCESS的APSME-ADD-GROUP.confirm原语。如果没有该入口，表中还有入口空间，APSME将在组表中建

立一个新的入口,其参数为给定的GroupAddress和Endpoint值。入口加入到APS组表后,APSME将发送NLME-SET.request原语来确保相应的网络层组表中的nwkGroupIDTable属性与APS子层中的组表包含的组地址列表相一致。一旦两个表一致了,APSME将向上层发送状态参数为SUCCESS的APSME-ADD-GROUP.confirm原语。如果没有给定参数GroupAddress和Endpoint的入口并且组表中没有建立另一个入口的空间,APSME将向上层发送状态参数为TABLE\_FULL的APSME-ADD-GROUP.confirm原语。

### 2.2.4.5.2 APSME-ADD-GROUP.confirm

该原语使得设备得知其将一个组添加到端点的请求结果。

#### 2.2.4.5.2.1 服务原语的语法

该原语的语法如下:

```
APSME-ADD-GROUP.confirm      {
                                Status
                                GroupAddress
                                Endpoint
                                }
```

表2.15描述了该原语的参数。

Table 2.15 APSME-ADD-GROUP.confirm Parameters

Name	Type	Valid Range	Description
Status	Enumeration	SUCCESS, INVALID_PARAMETER or TABLE_FULL	The status of the request to add a group
GroupAddress	16-bit group address	0x0000 - 0xffff7	The 16-bit address of the group being added
Endpoint	Integer	0x01 - 0xf0	The endpoint to which the given group is being added

#### 2.2.4.5.2.2 产生

该原语由APSME产生并发送给上层作为对APSME-ADD-GROUP.request原语的响应。如果APSME-ADD-GROUP.request成功,那么状态参数值为SUCCESS。如果APSME-ADD-GROUP.request中的参数为无效值,那么状态产生设置为INVALID\_PARAMETER。如果APSME试图加入一个组表入口,但表中已没有加入其它入口的空间,状态参数设置为TABLE\_FULL。

#### 2.2.4.5.2.3 接收

上层接收到该原语,则得知添加组请求的结果。状态参数值如上面所述。

### 2.2.4.5.3 APSME-REMOVE-GROUP.request

该原语允许上层请求将一个特定的组的组关系从特定的端点中移除。

## 2.2.4.5.3.1 服务原语的语法

该原语的语法如下：

```
APSME-REMOVE-GROUP.request {
    GroupAddress
    Endpoint
}
```

表2.16描述了该原语的参数。

**Table 2.16 APSME-REMOVE-GROUP.request Parameters**

Name	Type	Valid Range	Description
GroupAddress	16-bit group address	0x0000 - 0xffff7	The 16-bit address of the group being removed
Endpoint	Integer	0x01 - 0xf0	The endpoint to which the given group is being removed

## 2.2.4.5.3.2 产生

当上层要将一个特定组的关系从一个端点中移除时产生该原语，设置了组地址的帧将不被传送给该端点。

## 2.2.4.5.3.3 接收

如果接收到该原语，其GroupAddress参数的值超出了有效范围，APSME将向上层发送状态参数为INVALID\_PARAMETER的APSME-REMOVE-GROUP.condirm原语。同样，如果Endpoint参数值为0x00或当前设备的其它没有执行的端点，APSME将发送状态参数为INVALID\_PARAMETER的APSME-REMOVE-GRRROUP.confirm原语。

完成上述参数检测后，APSME将检查组表中是否存在包含给定参数GroupAddress和Endpoint的入口。如果该入口已存在于组表中，该入口将被移除。APSME将发送NLME-SET.request原语来确保相应的网络层组表中的nwkGroupIDTable属性与APS子层中的组表包含的组地址列表相一致。一旦两个表一致了，APSME将向上层发送状态参数为SUCCESS的APSME-REMOVE-GROUP.confirm原语。如果没有该入口，APSME将向上层发送状态参数为SUCCESS的APSME-REMOVE-GROUP.confirm原语。

## 2.2.4.5.4 APSME-REMOVE-GROUP.confirm

该原语使得设备得知其将一个组从端点中移除的请求结果。

## 2.2.4.5.4.1 服务原语的语法

该原语的语法如下：

```
APSME-REMOVE-GROUP.confirm {
    Status
    GroupAddress
}
```

```
Endpoint
}
```

表2.17描述了该原语的参数。

**Table 2.17 APSME-REMOVE-GROUP.confirm Parameters**

Name	Type	Valid Range	Description
Status	Enumeration	SUCCESS or INVALID_PARAMETER	The status of the request to remove a group.
GroupAddress	16-bit group address	0x0000 - 0xfff7	The 16-bit address of the group being removed
Endpoint	Integer	0x01 - 0xf0	The endpoint which is to be removed from the group

## 2.2.4.5.4.2 产生

该原语由APSME产生并发送给上层作为对APSME-REMOVE-GROUP.request原语的响应。如果APSME-REMOVE-GROUP.request成功，那么状态参数值为SUCCESS。如果APSME-REMOVE-GROUP.request中有参数为无效值，那么状态产生设置为INVALID\_PARAMETER。

## 2.2.4.5.4.3 接收

上层接收到该原语，则得知移除组请求的结果。状态参数值如上面所述。

## 2.2.4.5.5 APSME-REMOVE-ALL-GROUP.request

当上层想要将所有组中的关系从端点中移除时产生该原语，因此，没有组地址的帧传送给端点。

### 2.2.4.5.5.1 服务原语的语法

该原语的语法如下：

```
APSME-REMOVE-ALL-GROUPS.request {
    Endpoint
}
```

表2.18描述了该原语的参数。

**Table 2.18 APSME-REMOVE-ALL-GROUPS.request Parameters**

Name	Type	Valid Range	Description
Endpoint	Integer	0x01 - 0xf0	The endpoint to which the given group is being removed

### 2.2.4.5.5.2 产生



当上层想要将所有组中的关系从端点中移除时产生该原语，因此，没有组地址的帧传送给端点。

### 2.2.4.5.5.3 接收

接收到该原语，如果Endpoint参数值为0x00或当前设备的其它没有执行的端点，APSME将发送状态参数为INVALID\_PARAMETER的APSME-REMOVE-ALL-GRROUP.confirm原语。

完成上述参数Endpoint检测后, APSME将从组表中移除所有与该端点相关的入口。APSME将发送NLME-SET.request原语来确保相应的网络层组表中的nwkGroupIDTable属性与APS子层中的组表包含的组地址列表相一致。一旦两个表一致了，APSME将向上层发送状态参数为SUCCESS的APSME-REMOVE-ALL-GROUP.confirm原语。

### 2.2.4.5.6 APSME-REMOVE-ALL-GROUP.confirm

该原语使得设备得知其从一个端点中移除所有组的请求结果。

#### 2.2.4.5.6.1 服务原语的语法

该原语的语法如下：

APSME-REMOVE-ALL-GROUPS.confirm	{ Status, Endpoint }
---------------------------------	-------------------------------

表2.19描述了该原语的参数。

Table 2.19 APSME-REMOVE-ALL-GROUPS.confirm Parameters

Name	Type	Valid Range	Description
Status	Enumeration	SUCCESS, INVALID_PARAMETER or TABLE_FULL	The status of the request to remove all group
Endpoint	Integer	0x01 - 0xf0	The endpoint which is to be removed from all groups

#### 2.2.4.5.6.2 产生

该原语由APSME产生并发送给上层作为对APSME-REMOVE-ALL-GROUP.request原语的响应。如果APSME-REMOVE-ALL-GROUP.request成功，那么状态参数值为SUCCESS。如果APSME-REMOVE-ALL-GROUP.request中有参数为无效值，那么状态产生设置为INVALID\_PARAMETER。

#### 2.2.4.5.6.3 接收

上层接收到该原语，则得知从端点中移除所有组请求的结果。状态参数值如上面所述。

### 2.2.5 帧格式

这小节描述了APS层的帧格式（APDU）。每一个APS帧包含如下的基本组成：

- 1、APS头，由帧控制和地址信息组成。
- 2、APS有效载荷，变长，包含帧类型指定的信息。

APS子层的帧作为有序域按照指定的顺序进行描述。这小节的所有帧格式都按照网络层的传输顺序进行描述，从左至右，最左的位最先传输。每个域中的长度为k位都从0（最左、最低）至k-1（最右、最高）排号。域中长度小于一个字节的值都按照从最低位至最高位的顺序向网络层传输。

### 2.2.5.1 常规的APDU帧格式

APS帧格式由一个APS帧头和APS有效载荷组成。APS帧头域有固定的顺序，在帧中可以不包含地址域。常规的APS帧格式如表2.2所示。

Octets: 1	0/1	0/2	0/2	0/2	0/1	1	Variable	Variable
Frame control	Destination endpoint	Group address	Cluster identifier	Profile Identifier	Source endpoint	APS counter	Extended header	Frame payload
	Addressing fields							
APS header								APS payload

Figure 2.2 General APS Frame Format

#### 2.2.5.1.1 帧控制域

帧控制域8比特长，包含定义的帧类型、地址域和其它控制标志信息。帧控制域如表2.3所示的格式。

<b>Bits: 0-1</b>	<b>2-3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
Frame type	Delivery mode	Reserved	Security	Ack. request	Extended header present

Figure 2.3 Format of the Frame Control Field

##### 2.2.5.1.1.1 帧类型子域

帧类型子域为2比特长，可设置为表2.20所列出的值。

Table 2.20 Values of the Frame Type Sub-field

Frame Type Value $b_1 b_0$	Frame Type Name
00	Data
01	Command
10	Acknowledgement
11	Reserved

##### 2.2.5.1.1.2 传输模式子域

传输模式子域2比特长，可设置为表2.21所列出的值。

**Table 2.21 Values of the Delivery Mode Sub-field**

<b>Delivery Mode Value b<sub>1</sub> b<sub>0</sub></b>	<b>Delivery Mode Name</b>
00	Normal unicast delivery
01	Reserved
10	Broadcast
11	Group addressing

如果值为0b00，帧将被发送给接收设备给定的端点。

如果值为0b10，消息为广播发送。在这种情况下，消息将被发送给所选择的使用的广播地址的所有设备和所有端点，见3.7.5小节。

如果值为0b11，将使用组地址，帧只被发送给APS头中组地址域所确定的在组中表示组成员的设备端点。注意，源设备的其它端点可能是输出帧组地址的成员。帧将被发送给指定组的成员，包括源设备的其它端点。

### 2.2.5.1.1.3 安全子域

安全服务提供者（见4章）管理安全子域。

### 2.2.5.1.1.4 确认请求子域

确认请求子域1比特长，指定了当前的传输是否要求接收者接收到帧后发送确认帧。如果该子域设置为1，确定接收的为有效帧后，接收者需要构建并向发起者发送确认帧。如果该子域为0，确定接收的为有效帧后，接收者不向发起者发送确认帧。

### 2.2.5.1.1.5 延长头存在

延长头存在子域为1比特长，指定在帧中是否包含延长头。如果该子域设置为1，那么延长头包含在帧中。否则，不包含在帧中。

### 2.2.5.1.2 目的端点域

目的端点8比特长，指定帧的最终接收端点。如果帧控制域中的传输模式子域为0b00（标准单播发送），那么帧中包含该域。

目的端点值为0x00，该帧的目的地址为每个设备的ZOD。目的端点值为0x01-0xf0，帧目的地址为操作的端点。目的端点值为0xff，帧目的地址为除了端点0x00的所有活跃的端点。端点（0xf1-0xfe）保留。

### 2.2.5.1.3 组地址域

组地址域16比特长，只有当帧控制中的传输模式子域为0b11时存在该域。在这种情况下，目的端点不存在。如果帧中的APS头包含组地址域，帧将被发送设备中组表中由组地址域确定的所有端点。

设备的nwkUseMukticast设置为TRUE，输出帧不设置组地址域。

### 2.2.5.1.4 簇标识符域

簇标识符16比特长，指定由请求中SrcAddr所指示的用于设备绑定操作的簇标识符。帧

控制域的帧类型子域指定簇标识符域是否存在。该域只用于数据帧，不用于命令帧。

### 2.2.5.1.5 Profile标识符域

Profile标识符2字节长，指定在传输帧的过程中，用于设备过滤消息和帧的Profile标识符。该域之用于数据帧和确认帧。

### 2.2.5.1.6 源端点域

源端点域8比特长，指定发起者帧的端点。源端点值为0x00，表明从每个设备的ZD0发起。源端点值为0x01-0xf0，表明帧从应用操作的端点发起。其它的端点（0xf1-0xfe）保留。

### 2.2.5.1.7 APS计数器

该域8比特长，用于防止接收重复帧，如2.2.8.4.2小节。每新传输一次该值加一。

### 2.2.5.1.8 延长头子域

延长头子域包含深层子域，格式如表2.4所示。

Octets: 1	0/1	0/1
Extended frame control	Block number	ACK bitfield

Figure 2.4    Format of the Extended Header Sub-frame

#### 2.2.5.1.8.1 延长帧控制域

延长帧控制域长8比特，包含使用分裂的定义信息。延长帧控制域的格式如表2.5所示。

Bits: 0-1	2-7
Fragmentation	Reserved

Figure 2.5    Format of the Extended Frame Control Field

分裂子域2比特长，值为表2.22所列出的任意值。

Table 2.22    Values of the Fragmentation Sub-field

Fragmentation Value b <sub>1</sub> b <sub>0</sub>	Decription
00	Transmission is not fragmented
01	Frame is first fragment of a fragmented transmission
10	Frame is part of a fragmented transmission but not the first part
11	Reserved

#### 2.2.5.1.8.2 块序号

块序号域为1字节长，用于如下所述的分裂控制：如果分裂子域的设置表示不是分裂传

输，那么子域中不包含块序号域。如果分裂域设置为01，那么子域中包含块序号域，并且该域表示在分裂传输中块的序号。如果分裂域设置为10，那么子域中包含块序号域，并且表示当前帧传输的块序号，用值0x02表示第二个分裂块，0x03表示第三个，等等。

### 2.2.5.1.8.3 应答位域

应答位域为1字节长，用于2.2.8.4.3小节所描述的APS确认，表示成功传输哪个ASDU分裂块。该域只有在帧类型域表明为确认帧并且分裂子域表明是分裂传输使才出现。

### 2.2.5.1.9 帧有效载荷域

帧有效载荷域为变长，包含各个帧类型指定的信息。

## 2.2.5.2 个别帧类型的格式

定义了三种帧类型：数据、APS命令和确认帧。每一个帧类型都在下面的小节进行讨论。

### 2.2.5.2.1 数据帧格式

数据帧的格式如表2.6所示。

Octets: 1	0/1	0/2	0/2	0/2	0/1	1	Variable	Variable
Frame control	Destination endpoint	Group address	Cluster identifier	Profile Identifier	Source endpoint	APS counter	Extended header	Frame payload
	Addressing fields							
APS header								APS payload

Figure 2.6 Data Frame Format

数据帧中域的顺序如表2.2所示的APS帧顺序。

#### 2.2.5.2.1.1 数据帧APS帧头域

数据帧的APS帧头域包含帧控制、簇标识符、Profile标识符、源端点和APS计数器域。数据帧是否包含目的端点和延长头域则各自按照帧控制域中的传输模式和延长头存在域的规定。

在帧控制域中，帧类型应包含如表2.20所示的表示数据帧的值。源端点存在域设置为1。所有其它域根据使用数据帧的意图设置。

#### 2.2.5.2.1.2 数据有效载荷域

对于输出的数据帧，数据有效载荷应包含部分或全部上层请求APS数据服务传输的字节序列。对于输入数据帧，数据有效载荷域应包含APS数据服务接收到的转发给目的设备或如果协调器是其中的目的地发送给上层字节序列。

### 2.2.5.2.2 APS命令帧格式

APS命令帧格式如表2.7所示。

Octets: 1	0/2	1	1	Variable
Frame control	Group Address	APS counter	APS command identifier	APS command payload
APS header			APS payload	

Figure 2.7 APS Command Frame Format.

APS命令帧中域的顺序如表2.7所示的APS帧顺序。

2.2.5.2.2.1 APS命令帧APS头域

APS命令帧的APS头域应包含帧控制和APS计数器域。如果帧控制域中的传输模式子域表明为组地址，则帧中应包含组地址域。在该版本的规范中，APS命令帧不能分裂，并且没有延长头域。

在帧控制域中，帧类型子域应包含表明是APS命令帧的值，如表2.20所示。APS命令有效载荷应根据使用APS命令帧的意图进行适当的设置。

2.2.5.2.2.2 APS命令标识符域

APS命令标识符域表明正在使用APS命令。

2.2.5.2.2.3 APS命令有效载荷域

APS命令帧的APS命令有效载荷域应包含APS命令本身。

2.2.5.2.3 确认帧格式

确认帧格式如表2.8所示。

Octets: 1	0/1	2	2	0/1	1	Variable
Frame control	Destination endpoint	Cluster Identifier	Profile identifier	Source endpoint	APS counter	Extended header
APS header						

Figure 2.8 Acknowledgement Frame Format

确认帧中域的顺序应与表2.8所示的APS帧中域顺序一致。

2.2.5.2.3.1 确认帧APS头域

确认帧的APS头域应包含帧控制、簇标识符、Profile标识符和APS计数器。源和目的端点都应包含在确认帧中。是否包含延长头域则按照帧控制域中延长头存在子域的要求设备。

在帧控制域中，帧类型子域应包含如表2020所示的表示为确认帧的值。延长头存在域应包含同样表明为确认帧的值。所有其它子域则根据使用确认帧的意图进行适当的设置。

确认帧源端点的值反映了要求进行确认的帧的目的端点的值。同样，确认帧目的端点的值反映了要求进行确认的帧的源端点的值。

APS计数器域包含与确认的帧相一致的值。

如果延长头域存在，延长的帧控制域的分裂域应包含域确认的帧一致的值。如果该帧使

用分裂，那么应包含块序号和请求域。如果传输的使分裂的第一个帧，那么块序号应为0，否则应包含域确认的帧一致的值。

### 2.2.6 命令帧

这部分规范没有命令帧。APS命令帧和原语的相关安全问题见4.5.9小节。

### 2.2.7 常数和PIB属性

#### 2.2.7.1 APS常数

APS子层常量的定义与描述见表2.23.

Table 2.23 APS Sub-layer Constants

Constant	Description	Value
apscMaxDescriptorSize	The maximum number of octets contained in a non-complex descriptor	64
apscMaxDiscoverySize	The maximum number of octets that can be returned through the discovery process	64
apscMaxFrameRetries	The maximum number of retries allowed after a transmission failure.	3
apscAckWaitDuration	The maximum number of seconds to wait for an acknowledgement to a transmitted frame	$0.05 * (2 * nwkcMaxDepth) + (\text{security encrypt/decrypt delay})$ , where the $(\text{security encrypt/decrypt delay}) = 0.1$  (assume 0.05 per encrypt or decrypt cycle)
apscMinDuplicateRejectionTableSize	The minimum required size of the APS duplicate rejection table.	1
apscMaxWindowSize	Fragmentation parameter - The maximum number of unacknowledged frames that can be active at once (see Sub-clause 2.2.8.4.5)	Set by stack profile (1-8 supported)
apscInterframeDelay	Fragmentation parameter - The standard delay between sending two blocks of a fragmented transmission (see Sub-clause 2.2.8.4.5)	Set by stack profile

#### 2.2.7.2 APS信息数据库

APS信息数据库包含管理设备APS层需要的属性。AIB属性如表2.24所示。AIB还包含一些管理安全服务的属性。这些属性在4.5.10小节列出。



Table 2.24 APS IB Attributes

Attribute	Identifier	Type	Range	Description	Default
apsBindingTable	0xc1	Set	Variable	The current set of binding table entries in the device (see sub-clause 2.2.8.2.1)	Null set
apsDesignatedCoordinator	0xc2	Boolean	TRUE or FALSE	TRUE if the device should become the ZigBee Coordinator on startup, FALSE otherwise	FALSE
apsChannelMask	0xc3	IEEE802.15.4 channel mask	Any legal mask for the PHY	The mask of allowable channels for this device to use for network operations.	All channels
apsUseExtendedPANID	0xc4	64-bit extended address	0x0000000000000000 to 0xfffffffffffffe	The 64-bit address of a network to form or to join.	0x0000000000000000
apsGroupTable	0xc5	Set	Variable	The current set of group table entries (see sub-clause 2.2.8.3)	Null set

2.2.8 功能描述



2.3.1 建立一个 ZigBee Profile

在 ZigBee 网络中两个设备之间通信的关键是统一一个 profile。

Profile 的一个例子就是智能家居。这个 ZigBee profile 允许一系列设备类型交换控制消息来构造一个无线智能家居应用。这些设备被设计成很好的交换已知信息来实现这些控制，如控制灯的开和关，发送一个亮度传感器测量给一个照明设备控制器或者如果已有的传感器检测到移动就发送一个警告信息。

Profile 另一个类型的例子是在连个 ZigBee 设备间定义了普通行为。为了举例说明，无线网络在网络中依靠自制设备的能力来同网络连接和发现其他设备和在设备上的服务。设备和服务发现是在设备的 profile 中支持的特性。

2.3.1.1 从 ZigBee 联盟获得的 Profile 标识符

ZigBee 在两个分开的等级定义 Profile，这两个等级是：私人的和公开的。这些等级的

精确定义和标准是在 ZigBee 联盟和在这个文件范围之外的一个管理问题。为了这个技术规范的目的，对 Profile 标识符标准是唯一的。到最后，对一个 Profile 标识符的应用程序，每一个 Profile 必须以向 ZigBee 联盟的一个请求开始。一旦获得 Profile 标识符，Profile 标识符允许 Profile 设计者有如小定义：

- (1) 设备描述
- (2) 簇标识符

Profile 标识符的应用的市场空间对从 ZigBee 联盟发行 Profile 标识符是一个关键的标准。Profile 需要覆盖一个足够宽的设备范围来允许互动性来发生在没有过度范围设备之间，且导致用来描述它们接口的一个簇标识符的不足。相反的。Profile 不能被定义的太狭窄导致很多被个人 Profile 标识符描述的设备导致 Profile 标识符寻址空间的浪费，且在描述设备如何接口时产生互操作性。在 ZigBee 联盟里的政策组将就如何定义 Profile 建立标准，且帮助请求者制作它们的 Profile 标识符请求。

### 2.3.1.2 定义设备描述和簇

Profile 标识符是在 ZigBee 协议中主要的主要枚举量。每一个唯一的 Profile 标识符定义了设备描述和簇标识符的一个联合的枚举量。例如，对 Profile 标识符“1”，存在一些被 16 位值描述的设备描述（就是说在每一个 Profile 中可能有 65536 个设备描述）和一些被 16 位值描述的簇标识符（就是说在每一个 Profile 中可能有 65536 个标识符）。每一个簇标识符也支持一些被 16 位值描述的属性。例如，每一个 Profile 标识符最多有 65536 格簇标识符且每一个这样的标识符最多又可以包含 65536 格属性。Profile 开发者的责任就是定义和分配设备描述，簇标识符和在它们已分配的 Profile 标识符里的属性。注意设备描述、簇标识符和属性标识符的定义必须很小心的采用以保证简单描述的有效建立和当交换消息时单一化处理。设备描述和簇标识符必须通过将被处理的已知的profile标识符来完成。在任何消息被定向到一个设备之前，ZigBee协议采用已经使用服务发现确定profile在设备和端点的支持。同样的，绑定处理采用相似的服务发现，且profile发生，由于作为结果的匹配提取到源地址、源端点、簇标识符、目的地址和目的端点。

### 2.3.1.3在端点配置profile

在一个单独的 ZigBee 设备也许包含许多的 profile 的维持，这些 profile 是由在这些 profile 定义的各种簇标识符的子集提供的，且维持多样的设备描述。在设备里使用一个分层寻址定义的能力如下：

- (1) 设备：设备是由有唯一的 IEEE 和网络地址的单个无线电来维持的。
- (2) 端点：这是一个 8 位的域，描述了不同的应用程序，这些应用都是由单个无线电来维持的。端点 0x00 用来寻址设备 profile，设备 profile 是每个 ZigBee 设备必须使用的；端点 0xff 用来寻址所有活动的端点（广播端点），且端点 0xf1-0xfe 保留。结果，一个单独的物理 ZigBee 无线电能维持最多 240 个应用程序在端点 0x01-0xf0。

应用程序决定关于如何造设备端点配置应用程序和哪个端点来广播（advertise）。唯一的要求是每个端点都建立简单的描述符，且这些描述符对于服务发现是有效的。

### 2.3.1.4激活安全发现

一旦设备被建立维护特殊的profile且同簇描述符使用一致，簇描述符使用是为在这些profile中的设备描述，那么应用程序能被配置。为了达到这一点，每一个应用程序被分配给个别的端点，且每一个都使用简单描述符来描述。通过简单描述和在 ZigBee 设被

profile中描述的其他服务发现机制，激活服务发现，设备的绑定被维持和在补充的设备间应用程序的通知。

重要的一点是服务发现是以profile标识符、输入簇标识符列表和输出簇标识符列表（设备描述很明显的丢失了）为基础构成的。设备描述是在表示profile的类型的设备里规定必选的和可选的簇标识符维持的一个简单的协定。另外，期望设备描述枚举在PDA里使用或者其他辅助的绑定设备提供设备能力的额外描述。

### 2.3.1.5混合标准和所有权Profile

一个例子，ZigBee设备能被建立带有一个为了一个标准而写的单独的端点应用程序，公开的ZigBee profile标识符“XX”。如果生产商想配置一个ZigBee设备支持的标准profile“XX”，且提供给卖主特殊的扩展名，这些扩展名将被advertised在一个孤立的端点。维持标准的profile标识符“XX”，但生产时没有卖主扩展名的设备将仅仅advertised维持单独的profile标识符“XX”，且不能使用卖主扩展名响应或者建立消息。

### 2.3.1.6激活相反的兼容性

在先前的例子中，使用一个标准建立一个设备，这个标准公布ZigBee profile标识符“XX”，它包含了标准的profile的最初版本。如果ZigBee联盟将更新这个标准profile来建立新的特性和加法（additions），修订本将组合成一个新的标准profile，这个新的标准profile有一个新的profile标识符（即“XY”）。有profile标识符“XX”的设备应域新设备兼容，这新的设备对于profile标识符“XX”和profile标识符“XY”有新设备advertised维持。以这种方式，新设备使用profile标识符“XX”与旧设备通信，然而，也可以使用profile标识符“XY”与旧设备通信在相同的应用程序里。在ZigBee中的服务发现特性激活网络中的设备来确定维持级别。

### 2.3.2ZigBee描述

ZigBee设备使用描述符数据结构来描述它们自己。包含在这些描述符里的实际数据被定义在个人的设备描述符里。有五个描述符：节点、节点电源、简单的、复杂的和使用者，如表2.25所示。

表2.25ZigBee描述符

描述符名称	状态	描述
Node	M	节点的类型和能力
Node power	M	节点电源特性
Simple	M	包含在节点里的设备描述
Complex	O	设备描述的进一步信息
User	O	定义的使用者的描述符

#### 2.3.2.1描述符传送

节点、节点电源、简单的和使用者描述符按它们出现在各自的表中的顺序传送，也就是，在表头的域第一个传送，表底的域最后传送。每一个individual域按第一章规定的顺序传送。复杂的描述符的格式和传送如图2.15所示。

字节：1	可变长	...	可变长
域计数器	域1	...	域n

图2.15复杂描述符的格式

包含在复杂标识符里的每一个域的格式如图2.16所示。

字节：1	可变量
压缩的XML标志	域数据

图2.16individual复杂描述符域的格式

### 2.3.2.1.1域计数器域

域计数器域长度为1字节，且规定包含在描述符里的域的数值，每一个格式描述如图。

### 2.3.2.1.1压缩的XML标志域

压缩的XML标志域长度为1字节，且规定当前域的XML标志。复杂标识符的压缩XML标志如表2.37所示。

### 2.3.2.1.1域数据域

域数据域是可变量且包含当前域的信息规定，如压缩XML标志域表明的。

### 2.3.2.2经由描述符发现

在ZDO管理实体设备中询问标识符信息，且使用ZigBee设备标识符请求原语的服务发现寻址到端点0。发现操作的详细描述见2.4.2.1节。信息通过ZigBee设备profile指示（indication）原语返回。

节点、节点电源、复杂和使用者标识符应用于完整节点。简单标识符必须为了每个被定义的端点在节点里而被规定。如果一个节点包含多个子组，这些将在孤立的端点上，且对于这些特殊的描述符通过在ZigBee设备profile里包含的相关的端点数来读取。

### 2.3.2.3复合设备（Composite Devices）

一个ZigBee节点包含分开的子组的数，每一个都有它自己的简单标识符。对于发现机制是在ZigBee设备profile发现部分描述。

### 2.3.4节点描述符

节点描述符包含ZigBee节点能力的信息，且对于每个节点都是必选的。在一个节点里仅仅有一个节点描述符。

节点描述符的域如表2.26所示，是按照传送的顺序。

表2.26节点描述符域

域名	长度（bit）
逻辑类型	3
有效复杂描述符	1
有效使用者描述符	1
保留	3
APS标志	3
频率组合（Frequency band）	5
MAC能力标志	8
生产商代码	16
最大缓冲值	8
最大转换值（Maximum transfer size）	16
服务器MASK	16

2.3.2.4.1逻辑类型域

节点的逻辑类型域是3个bit长，且规定ZigBee节点的设备类型，逻辑类型域设置为表2.27的一个非保留值。

表2.27逻辑类型域的值

逻辑类型域值b2b1b0	描述
000	ZigBee协调器
001	ZigBee路由器
010	ZigBee终端设备
011-111	保留

2.3.2.4.2有效复杂描述符域

节点描述符的有效复杂描述符域是1bit长，且规定一个复杂描述符在这个设备上是否有效。如果这个域设置为1，复杂描述符有效；如果这个域设置为0，复杂描述符无效。

2.3.2.4.3有效使用者描述符域

节点描述符的有效使用者描述符域是1bit长，且规定一个使用描述符在这个设备上是否有效。如果这个域设置为1，使用者描述符有效；如果这个域设置为0，使用者描述符无效。

2.3.2.4.4APS标志域

节点描述符的 APS 标志域是 3bit 长，且规定节点的应用支持子层的能力。  
这个域是普遍的不维持且设置为0。( This field is currently not supported and shall be set to zero.)

2.3.2.4.5频率组合域

节点描述符的频率组合域是5bit长，且规定节点使用的IEEE802.15.4支持的频率组合。对每一个IEEE802.15.4支持的频率组合，频率组合域都有相应位，如表2.28所示，使用哪个频率组合相应位设置为1，其他位设置为0。

表2.28频率组合域的值

频率组合域位数	支持的频率组合
0	868- 868.6 MHz
1	保留
2	902- 928 MHz
3	2400 -2483.5 MHz
4	保留

2.3.2.4.6MAC层能力标志域

MAC层能力标志域长度为8bit，且规定了节点的能力，是IEEE802.15.4MAC子层所要求的。MAC层能力标志域格式如图2.27所示。

比特： 0	1	2	3	4-5	6	7
可选的PAN协调器	设备类型	电源源	空闲时接收机开	安全能力	分配地址	

图2.17MAC层能力标志域格式

可选的PAN协调器子域长度是1位，且如果这个节点有成为PAN协调器的能力，该域设置为1。否则设置为0。

设备类型子域1位长，且如果这个节点是一个全功能设备（FFD），该域设置为1。否则设置为0，表明是一个简化功能设备（RFD）。

电源源子域长度是1位，且如果当前的电源源是主电源，该域设置为1。否则该域设置为0。这个信息是从节点电源（power）描述符的节点当前电源源域获得的。

空闲时接收机开子域长度是1位，且如果在空闲周期时设备使能它的接收机保存电源，该域设置为1。否则该域设置为0（参见2.3.2.5节）

安全能力子域长度是1位，且如果设备有使用【B1】规定的安全组使发送和接收帧安全的能力，该域设置为1。否则该域设置为0。

分配地址子域长度是1位，且总设置为1。

### 2.3.2.4.7生产商代码域

节点描述符生产商代码域长度是16位，且规定了一个由ZigBee联盟分配的生产商代码，与设备相关。

### 2.3.2.4.8最大缓冲值子域

节点描述符的最大缓冲域值长度8位，有效范围是0x00-0x7f，且规定了节点的应用支持子层（ASDU）的最大值，是以字节的方式。在分裂或者重新组合之前，这是要传输到应用层或者从应用层过来的数据或者命令的最大值。

这个域为了网络管理被作为高水平表示使用。

### 2.3.2.4.9最大转换值

节点描述符的最大转换值长度是16位，有效值范围是0x0000-0xffff，且以字节形式规定了转换到这个节点或从这个节点转换的最大值在一个单个消息转换里。这个值能超过节点最大缓冲值域的值（参见2.3.2.4.8）。

### 2.3.2.4.10服务Mask域

节点描述符的服务Mask域长度是16位，位设置表示这个节点的系统服务能力。系统里的其他节点使用这个使特殊系统服务发现便利。位设置如表2.29定义。

表2.29服务Mask位分配

位数	分配
0	主要信托中心
1	备份信托中心
2	主要绑定表高速缓冲存储器
3	备份绑定表高速缓冲存储器
4	主要发现高速缓冲存储器
5	备份发现高速缓冲存储器
6	网络管理
7-15	保留

### 2.3.2.5节点电源描述符

节点电源描述符给节点的电源状态一个动态表示，且对每一个节点都是必须有的。在一个节点里就只有一个节点电源描述符。

节点电源描述域如表2.30所示，按照传输的顺序。

表2.30节点电源描述域

域名	长度（bit）
当前电源模式	4
有效的电源源	4
当前的电源源	4
当前电源源级别	4

### 2.3.2.5.1当前电源模式域

节点电源描述符的当前电源模式域长4位，且规定了节点的当前休眠/省电模式。当前节点模式域设置为表2.31所列的一个非保留值。

表2.31当前电源模式域的值

当前电源模式值b3b2b1b0	描述
0000	接收机与节点描述符的空闲时接收机开子域同步
0001	接收机如节点电源描述符定义的那样周期性的开始
0010	当有激励是接收机开，举例来说是使用者按下按钮
0011-1111	保留

### 2.3.2.5.2有效电源源域

节点描述符的有效电源源域长度4位，且规定了在这个节点的有效电源源。对于每个节点支持的电源源，有效的电源源域的相应的位如表2.32所列，设置为1，其他位设置为0。

表2.32有效电源源域的值

有效电源源域位数	支持的电源源
0	持续的电源（主要的）（Constant (mains) power）
1	可充电电池
2	可任意使用的电池（Disposable battery）
3	保留

### 2.3.2.5.3当前电源源域

节点描述符的当前电源源域长度4位，且规定节点使用的当前电源源。对于所选择当前电源源，当前电源源域相应的位如表2.23所列设置为1.其他位设置为0。

表2.23当前电源源域的值

当前电源源域位数	当前电源源
0	持续的电源（主要的）（Constant (mains) pow
1	可充电电池
2	可任意使用的电池（Disposable battery）
3	保留

### 2.3.2.5.4当前电源源级别域

节点描述符的当前电源源级别域长度4位，且规定了电源源负荷的级别。当前电源源域设置成表2.34所列的非保留值之一。

表2.34当前电源源级别域的值

当前电源源级别域b3b2b1b0	负荷水平
0000	危急的（Critical）没有电？？
0100	33%



1000	66%
1100	100%
其他值	保留

### 2.3.2.6简单描述符

简单描述符包含节点里的每一个端点的特定信息。简单描述符在节点里存在的每一个端点是必选的。

简单描述符域如表2.35所示，是按照传输的顺序。这个描述符在整个空间进行传输，简单描述符的全部长度应小于等于 *maxCommandSize*。

表2.35简单描述符域

域名	长度（bits）
端点	8
应用profile标识符	16
应用设备标识符	16
应用设备版本	4
保留	4
应用输入簇计数器	8
应用输入簇列表器	16*i（i是应用输入簇计数器的值）
应用输出簇计数器	8
应用输出簇列表器	16*o（o是应用输出簇计器的值）

#### 2.3.2.6.1端点域

简单描述符的端点域长度是8位，且规定在这个描述相关的节点里的端点。应用只用端点1-240。

#### 2.3.2.6.2应用profile标识符域

简单描述符的应用profile标识符域长度是16位，且规定在这个端点上支持的profile。Profile标识符从ZigBee联盟处获得。

#### 2.3.2.6.3应用设备标识符域

简单描述符的应用设备标识符域长度是16位，且规定在这个端点上支持的设备描述符。设备描述符从ZigBee联盟处获得。

#### 2.3.2.6.4应用设备版本域

简单描述符的应用设备版本域长度是4位，且规定在这个端点上支持的设备描述符的版本。设备描述符的版本设置为表2.36所列的非保留值之一。

表2.36应用设备版本域的值

6应用设备版本域的值b3b2b1b0	描述
0000	版本1.0
0001-1111	保留

#### 2.3.2.6.5应用输入簇计数器域

简单描述符的应用输入簇计数器域长度是8位，且规定在这个端点上支持的输入簇数，将出现在应用输入簇列表域。如果这个域的是0，应用输入列表域不被包含。

3.2.2.6.6应用输入簇列表

简单描述符的应用输入簇列表长度为16*i*，*i*是应用输入簇计数器域的值，且规定了在这个端点上支持的输入列表，在绑定程序期间使用。

应用输入簇列表仅仅在输入簇计数器域的值大于0是才有。

2.3.2.6.7应用输出簇计数器域

简单描述符的应用输出簇计数器域长度是8位，且规定在这个端点上支持的输出簇数，将出现在应用输出簇列表域。如果这个域的是0，应用输出列表域不被包含。

3.2.2.6.6应用输出簇列表

简单描述符的应用输出簇列表长度为16*o*，*o*是应用输出簇计数器域的值，且规定了在这个端点上支持的输出列表，在绑定程序期间使用。

应用输出簇列表仅仅在输出簇计数器域的值大于0是才有。

2.3.2.7复杂描述符

复杂描述包含在节点里的每一个复杂描述符的扩展信息。复杂描述的使用是可选的。

由于在这个描述符里的扩展的和复杂的特性，它使用压缩的XML标志以XML格式存在。描述符的每个域如表2.37所示，可以以任何顺序传输。作为这个标识符需要在整个空间传输，复杂描述符的全部长度应小于等于*maxCommandSize*。

表2.37复杂描述符域

域名	XML标志	复杂XML标志值b3b2b1b0	数据类型
保留	-	0000	-
语言和字符设置	<语言代码>	0001	参见2.3.2.7.1
生产商名称	<生产商名称>	0010	字符串
模型名称	<模型名称>	0011	字符串
连续数	<连续数>	0100	字符串
设备URL	<设备URL >	0101	字符串
图标（Icon）	<图标>	0110	字节串
图标URL	<大纲 >	0111	字符串
保留	-	1000-1111	-

2.3.2.7.1语言和特性设置域

语言和字符设置域是3字节长，且规定了在复杂描述符里的字符字节串使用的语言和字符设置。语言和字符设置域的格式如图2.18所示。

字节：2	1
ISO639-1语言代码	字符设置标识符

图2.18语言和字符设置域格式

ISO639-1语言代码域是2字节长，且规定了为字符串使用的语言，如【B5】定义。

字符设置标识符子域长度是1字节，且规定了在字符设置里的字符使用的编码。这个子域设置为表2.38所列的非保留值之一。

表2.38字符设置标识符子域的值

字符设置标识符值	每个标识符的比特数	描述

0x00	8	ISO646, ASCII字符设置。每一个特性都适合一个字节的最没有意义的7 bit, 带有最有意义bit设置为0 (见【B6】)
0x01-0xff	-	保留

如果语言和字符设置都没有规定, 语言默认为英语(语言代码: “EN”)且字符设置为ISO 646。

### 2.3.2.7.2生产商名称域

生产商名称域是可变长, 且包含字符串表明设备生产商的名称。

### 2.3.2.7.3模型名称域

模型名称域是可变长, 且包含字符串表明设备生产商模型的名称。

### 2.3.2.7.4连续数域

连续数域是可变长, 且包含字符串表明设备生产商连续数。

### 2.3.2.7.5设备URL域

设备URL是可变长, 且包含字符串表明URL, 通过它更多的关于设备的信息可以获得。

### 2.3.2.7.6图标域

图标域是可变长, 且包含一个字节串, 这个字节串携带一个图标数据, 能表明在计算机、网关或者PDA上的设备。图标的格式是32\*32像素的PNG图像。

### 2.3.2.7.7图标URL域

图标URL域是可变长, 且包含字符串表明URL, 通过它可以获得设备的图标。

## 2.3.2.8使用者标识符

使用者标识符包含允许使用者使用user-friendly字符标识符来识别设备的信息, 这些字符串如“Bedroom TV”或者“Stairs light”。使用者标识符的使用是可选的。这个标识符包括一个单独的域, 使用ASCII字符设置, 且包含一个16个字符的最大值。

使用者标识符域如表2.39所示, 按照它们传输的顺序。

表2.39使用者标识符域

域名	长度(字节)
使用者标识符	16

## 2.3.3功能描述

### 2.3.3.1接受和拒绝

应用程序框架能通过APS子层的数据服务过滤到达的帧, 且仅存在对在每个活动的(active)端点上执行的应用有影响的帧。

应用程序框架通过APSDedata.indication原语从APS子层接收数据, 且被标定为一个特殊的端点(DstEndpoint参数)和一个特殊的profile(ProfileId参数)。

如果应用程序框架为一个不活动的端点接收一个帧, 丢弃该帧。否则, 应用程序框架应确定是否规定profile标识符与在规定的端点上执行的profile标识相匹配。如果profile标识符不匹配, 那么应用程序框架拒绝该帧。反之, 应用程序框架应传递接收到的帧的载荷到执行在端点的应用。

## 2.5 ZigBee设备对象（ZDO）

### 2.5.1 范围

本小节介绍在ZigBee应用支持子层和网络层顶端执行ZigBee设备对象应用需要的概念、结构和原语。

ZDO是使用网络和应用支持层原语执行ZigBee终端设备、路由器和协调器的一个应用。

ZDO Profile使用簇来描述它的原语。ZigBee设备Profile簇不使用属性，且同在消息传输协议里的消息类似。在ZigBee设备中使用簇标识符来列举在ZDO中使用的消息。

ZDO也使用配置属性。这些属性不是任何簇的元素。在ZDO中的配置属性是由应用或者是栈Profile设置的配置参数。虽然配置属性和ZigBee设备Profile都由ZDO来使用，但是配置属性和ZigBee设备Profile无关。

### 2.5.2 设备对象描述

ZDO 是应用解决方案，驻扎在 ZigBee 协议栈中的 APL 层和 APS 层之上，如图 1.1 所示。

ZDO 有以下功能：

（1）初始化应用支持子层（APS），网络层（NWK），安全服务提供（SSP）和任何其他 ZigBee 设备层而不是驻扎在端点 1-240 的终端应用。

（2）从终端应用中集合配置信息来确定和执行下节描述的功能。

#### 2.5.2.1 最初的发现高速缓冲器设备操作（Primary Discovery Cache）

最初的发现高速缓冲器设备是通过设备的配置和在节点描述符里的 advertisement 来指定的。最初的发现高速缓冲器设备操作作为一个状态机，这个状态机是关于客户机希望使用最初的发现高速缓冲器。如下的状态和操作，如图 2.99 描述的，应被最初的发现高速缓冲器设备支持：

1. 未发现的：

客户使用有限的半径广播到所有的 RxOnWhenIdle 设备消息 Discovery Register 请求来定位在请求提供的半径范围内的 Primary Discovery Cache 设备

2. 发现的：

客户使用单播发现高速缓存器请求，这个请求是定向到 Discovery Cache 设备，这个设备包含它愿意存储的发现高速缓存器信息的大小。Discovery Cache Device 将响应，参数是 SUCCESS 或者 TABLE\_FULL。

3. 已注册的：

当客户从 Discovery Cache 设备接收到 SUCCESS 状态，这个状态就从先前的 Discovery Cache 请求处到达。客户现在必须使用节点描述符 (NodeDescriptor) 存储请求、电源描述符存储请求、活动的端点存储请求和简单描述符存储请求上载它的发现信息来

激活Primary Discovery Cache设备为了它自己的利益来充分的响应。

4.未注册的:

客户（或任何其他设备）也许请求不被注册。移动节点高速缓存器（Remove Node Cache）请求移动设备从Primary Discovery Cache设备。

Primary Discovery Cache设备响应设备和它支持的所有注册的客户的发现请求。Find Node Cache请求被想定位设备和为了已给设备的服务发现请求的客户使用。注意如果发现信息被设备本身保持，设备也必须响应来确认它自己作为发现信息的储藏。见图2.99为状态机处理Primary Discovery Cache设备的详细信息。

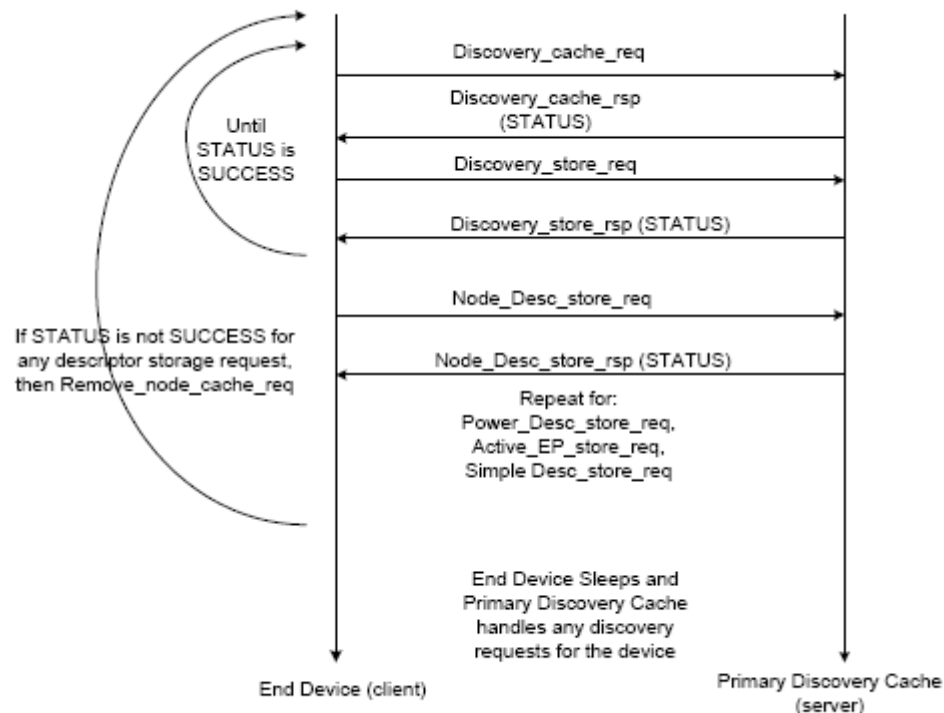


Figure 2.99 Primary Discovery Cache State Machine

## 2.5.2.2 设备和服务发现

在一个单独的PAN里，这个功能将支持设备和服务发现。另外，对于ZigBee协调器、ZigBee路由器和ZigBee终端设备类型，这个功能将做如下处理：

(1) 在每一使用休眠的ZigBee终端设备、ZigBee路由器（或ZigBee协调器）的网络，必须被设计作为如它们的节点描述符描述的Primary Discovery Cache Devices。这些Primary Cache Devices 是它们自己可发现的，且提供服务器服务来上载和存储代表休眠的ZigBee终端设备的发现信息。另外Primary Cache Devices响应代表休眠Zigbee]终端设备的发现请求。每一个Primary Discovery Cache Device是ZigBee路由器或者ZigBee协调器。

(2) 对于被:Config\_Node\_Power,设备和服务发现指示想要休眠的ZigBee终端设备将管理被ZigBee终端设备选择的Primary Discovery Cache设备上的网络地址、IEEE地址、活动节点、简单描述符、节点描述符和电源描述符的上载和存储来允许在这些休眠设备上的设备和服务发现操作。

(3) 对于被设计作为Primary Discovery Cache Device的ZigBee协调器和ZigBee路由器，这个功能将代表休眠ZigBee终端设备响应发现请求，这些终端设备已经注册和上载了它们的发现信息。

(4) 对于所有的ZigBee设备、设备和服务发现将支持设备和从其他设备过来的服务发现请求，且允许从其他本地的应用对象过来的请求的产生。注意设备和服务发现服务是由Primary Discovery Cache设备代表其他ZigBee终端设备提供的。万一Primary Discovery Cache Device是请求的目标，那么NWKAddrOfInterest或者Interest域的设备将被请求和/或响应填满来区分从设备来的请求的目标，这个设备是发现的目标。将支持下边的发现特性：

(1) 设备发现：

——以ZigBee协调器或者路由器IEEE地址的一个单播询问为基础，被请求设备的IEEE地址，随机的，所有联合设备的网络地址将被返回。

——以ZigBee终端设备的IEEE地址的一个单播询问为基础，被请求的设备的IEEE地址被返回。

——以ZigBee协调器或者带有一个已经提供的IEEE地址的路由器网络地址的一个多播询问（任何广播地址类型）为基础，被请求的设备的网络地址，随机的，所有联合设备的网络地址将被返回。

——以带有已经提供的IEEE地址的ZigBee终端设备的网络地址的广播查询（任何广播地址类型）为基础。被请求设备的网络地址被返回。响应的设备将使用APS层为单播响应已知的服务来广播查询。

(2) 服务发现：以如下的输入为基础，相应的响应被提供：

——网络层地址加上 (plus) 活动的端点查询类型——指定设备将返回在那个设备里的所有应用的端点数。

——网络层地址或广播地址（任何广播地址类型）加上服务匹配，这些匹配包括Profile ID和随意的，输入和输出簇——指定的设备匹配带有所有活动的端点的Profile ID来确定一个匹配。如果没有输入或者输出簇被规定，匹配请求的端点被返回。如果那些匹配的输入和/或输出簇在请求里被提供，且任何匹配在带有提供匹配的设备上的端点列表的响应里被提供。响应的设备应该使用APS层已知的服务，这服务是为了单播响应到广播查询的。万一应用profiles想列举输入簇和它们的带有相同簇标识符的响应输出簇，应用profile将仅仅在为服务发现目的的简单标识符里列出输入簇。在这些情况下它将被采用，应用profile提供关于输入和响应输出的簇标识符的使用的细节。

——网络层地址加上节点标识符或标识符查询类型——指定的地址将为设备返回联合端点的简单标识符。

——随意的，网络层地址加上复杂或者使用者标识符查询类型——如果支持，指定的地址将为设备返回复杂或者使用者标识符。

### 2.5.2.3安全管理

这个功能确定是否使能安全，如果使能，将做如下处理：

建立钥匙

传输钥匙

请求钥匙

更新设备

移动设备

转换钥匙

安全管理功能按安全服务规范执行。安全管理由ZDO发出APSME原语来执行，步骤如下：

与信托中心通信（假定是ZigBee协调器）来获得Master Key，在设备和信托中心之间（如果设备是ZigBee协调器或者信托中心的Master Key被重新分配这一步忽略）。这一步使用传



输钥匙原语。

与信托中心建立一个Link Key。这一步使用APSMEEstablish-Key原语。

从信托中心获得网络钥匙使用安全的通信与信托中心。这一步使用 APSME-TRANSPORT-KEY原语。

作为必须的，建立Link Key和Master Key与在网络中被确定为消息的目的的指定的设备。这步使用APSMEESTABLISH-KEY和/或APSME-REQUEST-KEY原语。

使用APSMEDevice-UPDATE通知任何一个设备的信托中心连接网络。这个功能只有设备是ZigBee路由器时才执行。

允许设备使用APSMEREQUEST-KEY原语从信托中心获得钥匙。

允许信托中心从网络中移动设备，使用APSME-REMOVE-DEVICE原语。

允许信托中心转换active的网络钥匙，使用APSMESWITCH-KEY原语。

## 2.5.2.4网络管理

这个功能将执行ZigBee协调器、ZigBee路由器或者ZigBee终端设备逻辑设备类型根据已确定的配置设置，通过程序应用或者在安装期间。如果设备类型是一个ZigBee协调器或者Zigbee终端设备，这个功能将提供选择一个存在的PAN来加入和如果网络通信断开执行允许设备重新加入的程序的能力。如果设备类型是ZigBee协调器或者是Zigbee路由器，这个功能将提供为一个新的PAN建立选择一个未用的信道。注意在没有一个设备是预先指定为协调器的情况下，配置一个网络是可能的，这时，第一个全功能设备 (FFD) 被确定为ZigBee协调器的角色。网络管理做如下处理：

允许为网络信道列表的规定扫描程序。缺省值是规定在已选择的操作联合的所有信道的使用。

管理网络扫描程序来确定邻居网络和它们协调器和路由器的一致性。

允许一个信道的选择来启动一个PAN（ZigBee协调器）或者一个存在的PAN的选择来连接（ZigBee路由器或者Zigbee终端设备）。

支持孤点和扩展的程序来重新连接网络，包括支持可携带的内部PAN。

也许支持直接连接。对于ZigBee协调器和ZigBee路由器，直接连接的一个本地版本被支持来使能设备通过孤点或者重新连接流程来加入网络。

## 2.5.2.5绑定管理

绑定管理执行下列任务：

为绑定表建立一个资源值。这个资源值是通过程序应用或通过一个在安装期间定义的配置参数确定的。

从APS绑定表增加或者减少实体处理绑定请求。

从外部应用支持绑定和解绑定命令，如那些是主机在一个PDA上来支持协助绑定。绑定和解绑定命令将通过ZigBee设备Profile（见2.4节）被支持。

对于ZigBee协调器，支持终端设备绑定，这绑定允许以按钮按压或其他手动菜单为基础的绑定。

## 2.5.2.6节点管理

对于Zigbee协调器和路由器，节点管理功能执行以下步骤：

允许遥控操作命令来执行网络发现

提供遥控操作命令来重新获得路由表

提供遥控操作命令来重新获得绑定表



提供一个遥控操作命令来使一个设备离开网络或者是命令另一个设备离开网络  
提供一个遥控操作命令来重新获得LQI，是为这个遥远的设备的邻居获得的。  
允许源设备向一个初始化绑定表高速缓冲寄存器登记的能力来保持他们自己绑定表  
允许配置工具把一个设备换成另一个设备，这个设备是在所有的绑定表入口中，这个入口涉及到他。  
允许初始化绑定表高速缓冲寄存器备份和恢复个人绑定入口或者入口绑定表或者保持他们自己绑定表的源设备的表  
提供一个遥控操作命令来允许或者禁止连接一个特殊的路由器；或者通常允许或者禁止通过信托中心连接

### 2.5.3层接口描述

不像对于应用居住的上述的端点1-240的其他设备描述，Zigbee设备对象(ZDO)接口除了APSDE-SAP之外，通过APSME-SAP到APS，通过NLME-SAP到NWK。ZDO在端点0上通信像所有其他应用一样通过Profiles使用APSDE-SAP。ZDO使用的Profile是ZigBee 设备Profile（见2.4节）

### 2.5.4系统使用方法

标题在协议版本发布的图表的同一页。

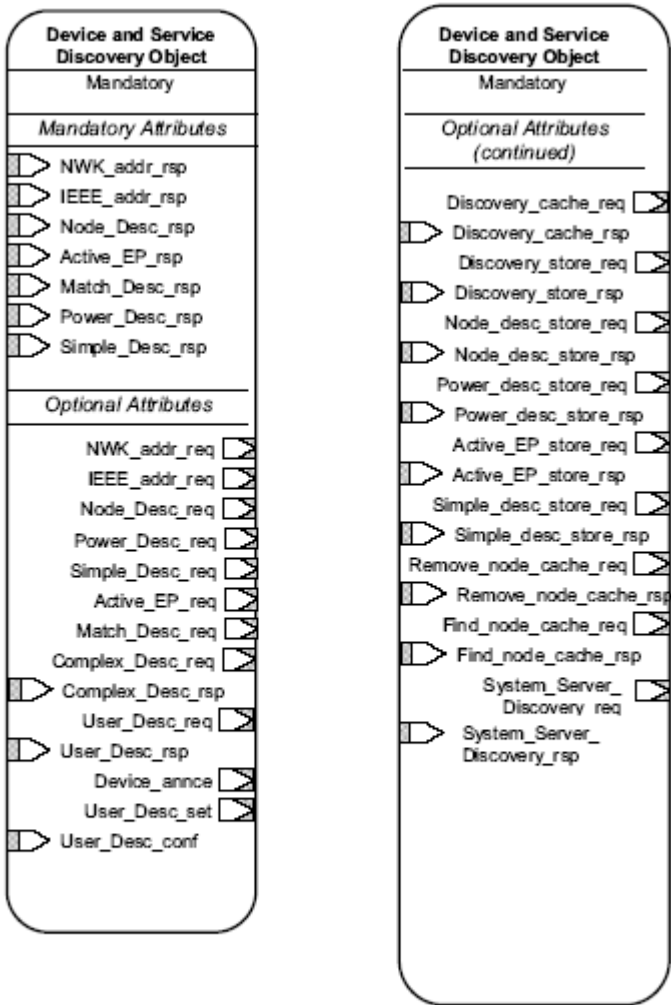


Figure 2.100 System Usage ZigBee Device Object Details

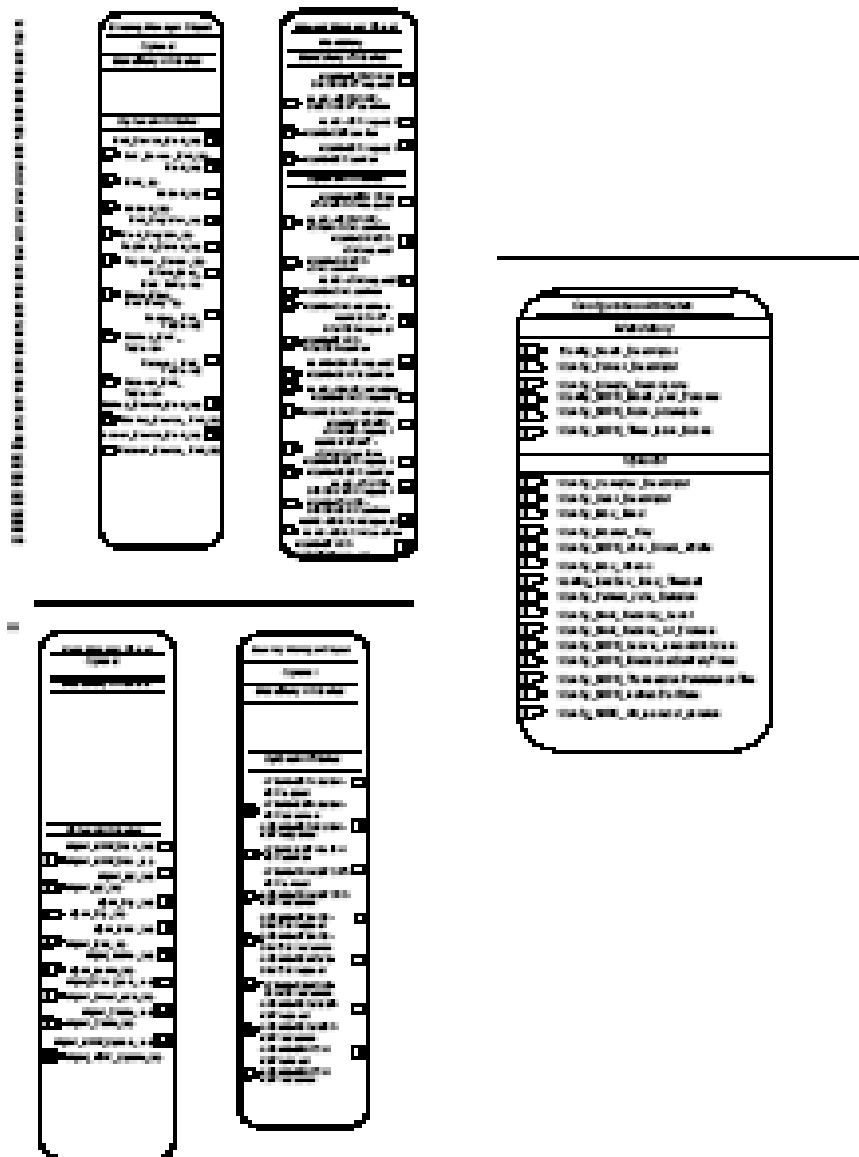


Figure 2.101 System Usage ZigBee Device Object Details.

## 2.5.5对象定义和行为

### 2.5.5.1对象概述

ZigBee设备对象包括五个对象：

设备和服务发现

网络管理

绑定管理

安全管理

节点管理

表2.132描述这些ZigBee设备对象

表2.132 ZigBee设备对象

对象		描述
名称	状	

	态	
:Device_and_Service_Discovery	M	处理设备和安全发现
:Network_Manager	M	处理网络行为，如网络发现，断开/加入网络，重新设置一个网络连接和建立一个网络
:Binding_Manager	O	处理终端设备绑定，绑定和解绑行为
:Security_Manager	O	处理安全服务，如钥匙装载，钥匙建立，钥匙传输和认证
:Node_Manager	O	处理操作功能

## 2.5.5.2可选的和强制的对象和属性

作为强制的列出的对象将在所有ZigBee设备中存在。然而，对于确定的ZigBee逻辑类型，对于所有ZigBee设备作为可选的列出的对象对于特殊的逻辑设备类型也许是强制的。例如，在网络管理对象中的NLME-NETWORK-FORMATION.request原语是强制对象且是可选属性，尽管对于Zigbee协调器逻辑设备类型属性是必需的。每一个设备类型部分的介绍将详细说明逻辑设备类型的对象和属性支持的必要条件。

## 2.5.5.3安全钥匙使用方法

ZigBee设备对象也许为了由ZigBee设备Profile原语建立的数据包使用安全。这些在端点使用APSDE的应用数据包将使用网络钥匙，不使用个人连接钥匙。

## 2.5.5.4公共的和私人的方法

能够到达设备的任何端点应用的方法叫做公共方法。私人方法是仅仅可以到达端点0的设备应用，且不是到达终端设备（运行在端点1到240）

## 2.5.5.5状态机功能描述

### 2.5.5.5.1ZigBee协调器

#### 2.5.5.5.1.1初始化

在执行中应该安排供应（**Provision**）来提供期望网络配置参数（:Config\_NWK\_Mode\_and\_Params）的一个单独复制到ZigBee设备对象的网络对象。另外，安排供应来提供配置元素来描述节点描述符，电源描述符，简单描述符为每一个活动的端点，和**应用加上活动端点的列举**。？？？这些配置将包括在:Config\_Node\_Descriptor,:Config\_Power\_Descriptor 和 :Config\_Simple\_Descriptors里，如果:Config\_Node\_Descriptor配置对象表明这个设备是Primary Discovery Cache设备，那么这个设备将被配置成处理服务器命令，是为了联合请求Primary Discovery Cache 的ZigBee设备Profile。且将会根据2.5.2.1节提供的状态机描述符来处理。

如果支持，将安排供应为复杂描述符，使用者描述符，绑定入口最大值和master key 提供配置元素。这些元素将包含在

:Config\_Complex\_Descriptor ,  
:Config\_User\_Descriptor, :Config\_Max\_Bind 和:Config\_Master\_Key里。

设备应用使用NLME-NETWORK-DISCOVERY.request原语，其中:Config\_NWK\_Mode\_and\_Params的ChannelList部分能扫描指定的信道。作为结果的NLME-NETWORK-DISCOVERY.confirm原语将提供一个网络清单，这个网络清单详细描述了在这个范围内的

活动的PANs。设备应用将比较信道清单和网络清单，且选择一个未使用过的信道。

未使用信道选择的运算法则的规定将留给工具（implementer?）。一旦未使用的信道被确定，设备应用将设置*nwkSecurityLevel*和*nwkSecureAllFrames*的NIB属性，是根据设备使用的堆栈 Profile 里的规定建立的 值来设置的。它将使用 NLME-NETWORK-FORMATION.request原语，原语使用在:Config NWK Mode and Params规定的参数来建立一个PAN在那个信道。在*nwkExtendedPANID*里，扩展的PANID域将被设置。设备应用将通过 NLME-NETWORK-FORMATION.confirm原语来核对返回的状态来检查PAN的成功建立。:Config\_Permit\_Join\_Duration将根据使用NLME-PERMIT-JOINING.request原语提供的缺省参数值来设置。另外，*nwkNetworkBroadcastDeliveryTime*和*nwkTransactionPersistenceTime*网络信息块参数将分别的设置 为 with :Config NWK BroadcastDeliveryTime 和:Config NWK TransactionPersistenceTime（参见第三章）

应该安排供应来确保从端点0到240的终端设备的APS原语命令返回合适的错误状态值，是在ZigBee设备对象的初始化状态完成之前，且转换正常的操作状态。

### 2.5.5.1.2正常的操作状态

在这个状态中，Zigbee协调器将处理直接连接地址清单，是在:Config\_NWK\_Join\_Direct\_Addrs里，通过为每个仔清单里包含的地址产生NLME-DIRECTJOIN.request原语。直接连接地址处理过程将使用:Config\_Max\_Assoc参数来测试在:Config\_NWK\_Join\_Direct\_Addrs里是否成功处理直接连接地址。

ZigBee协调器将响应任何的设备发现或者服务发现操作，是由它自己设备请求的，且如果他指定作为一个Primary Discovery Cache设备，也将代表注册的设备响应，这些设备已经存储了发现信息。设备应用将确保绑定入口数不超过:Config\_Max\_Bind属性。

ZigBee协调器将支持NLME-PERMIT-JOINING.request原语和NLME-PERMIT-JOINING.confirm原语允许网络连接处理的应用控制。

ZigBee协调器将支持NLME-LEAVE.request和NLMELEAVE.indication原语，原语使用:Config\_NWK\_Leave\_removeChildren属性，在这个属性里适当允许在应用控制下的联合设备的移除。导致移除的条件也许包括缺少安全信任，设备的移除是通过一个有特权的应用或者是例外的发现。

ZigBee协调器应包含当前联合的设备的清单，且方便了孤点扫描的支持，并且重新连接处理使能先前的联合设备来重新连接到网络。ZigBee协调器也许为设备维持直接包含在网络中的能力，是通过NLME-DIRECTJOIN.request和NLME-DIRECT-JOIN.confirm原语。这个特性应允许ZigBee IEEE 地址清单被提供给Zigbee协调器，因为那些地址被包含作为先前的联合设备。对于由这些地址的ZigBee设备通过孤点或者重新连接程序而不是联合的直接的连接到网络是可能的。

ZigBee协调器应处理End\_Device\_Bind\_req从ZigBee路由器和终端设备。一旦接收到一个End\_Device\_Bind\_req，ZigBee协调器将使用属性中的:Config\_EndDev\_Bind\_Timeout的值，且等待第二个End\_Device\_Bind\_req的到来。第二个指示在timeout期间到达，ZigBee协调器将在两个指示之间匹配Profile ID。如果在两个指示中的Profile IDs不匹配，一个适当的错误状态将通过End\_Device\_Bind\_req返回到每个设备，。如果Profile IDs匹配，ZigBee协调器将匹配两个指示里的AppInClusterLists和AppOutClusterLists。第一个指示的AppInClusterLists的Cluster IDs和第二个指示里的AppOutClusterLists的Cluster IDs匹配将被保存在一个清单里为指示End\_Device\_Bind\_req。

ZigBee协调器将处理从其它ZigBee设备来的Device\_annce信息。一旦接收到Device\_annce，ZigBee协调器将检查所有的内部表，这些内部表是为PAN中设备维持64位

IEEE地址为了与在Device\_annce信息中提供的地址相匹配。如果匹配存在，ZigBee协调器将根据匹配的64位IEEE地址更新它的nwkAddressMapNIB属性来反映包含在Device\_annce中的更新的16位网络地址。

### 2.5.5.5.1.3 信托中心操作

当网络中安全使能时，ZigBee协调器将行使一个信托中心的功能。

信托中心被网络中的新设备通知，是通过APSMEDevice-UPDATE.indication原语。信托中心也能选择允许设备保持在网络中或者是强迫他离开网络，是通过APSMEREMOVE-DEVICE.request原语。这个选择是使用网络控制原则制定的，是在这个协议范围外的。

如果信托中心决定允许设备保持在网络中，他将和设备建立一个master key，是通过APSME-TRANSPORTKEY.request原语，除非master key已经在两个设备中可用，且信托中心使用 out-of-band 机制来保证安全和认证。一旦交换了 master key，信托中心将使用 APSMEESTABLISH-KEY.request 原语与设备建立一个 link key，且将使用 APSMEESTABLISHKEY.response原语相应link key建立的请求。

信托中心将提供给设备网络钥匙，是通过使用APSMETRANSPORT-KEY.request原语。一旦通过APSME-REQUEST-KEY.indication原语从设备接收到一个请求，他将提供网络钥匙。

信托中心再任何两个设备间将支持link keys的建立，是通过提供给他们一个共同的钥匙。一旦接收到APSMEREQUEST-KEY.indication原语请求一个应用钥匙，信托中心将建立一个master key或者link key，并且使用the APSMETRANSPORT-KEY.request原语传输它到两个设备。

信托中心将周期性的更新网络钥匙，是根据一个原则，这个原则的详细内容在本协议范围外。网络中的所有设备将被更新新的网络钥匙，是通过APSME-TRANSPORT-KEY.request原语。

### 2.5.5.5.2 ZigBee路由器

#### 2.5.5.5.2.1 初始化

在执行中应该安排供应（Provision）来提供期望网络配置参数（:Config\_NWK\_Mode\_and\_Params）的一个单独复制到ZigBee设备对象的网络对象。如果:Config\_Node\_Descriptor配置对象表明这个设备是Primary Discovery Cache设备，设备将被配置成处理服务器命令为联合了请求Primary Discovery Cache的ZigBee设备Profile，且将根据2.5.2.1节提供的状态机描述来操作。

如果支持，将安排作为复杂描述符，使用者使用描述符，为绑定入口最大值和master key提供配置元素。这些元素将被包含在:Config\_Complex\_Descriptor, :Config\_User\_Descriptor, 以及:Config\_Max\_Bind 和:Config\_Master\_Key里。

设备应用将使用带有:Config\_NWK\_Mode\_and\_Params的ChannelList选项的NLME-NETWORK-DISCOVERY.request原语，然后使用NLME-NETWORK-DISCOVERY.request原语属性来扫描特殊的信道。

作为结果的NLME-NETWORK-DISCOVERY.confirm原语将提供一个详细描述在那个范围内的活跃的PANs的网络清单（NetworkList）。NLME-NETWORKDISCOVERY.request程序将被:Config\_NWK\_Scan\_Attempts执行，每一个被:Config\_NWK\_Time\_btwn\_Scans及时的分离。重复NLME-NETWORK-DISCOVERY.request原语的目的是提供一个更正确的邻居列表和到网络层的联合的连接质量指示。设备应用将比较信道清单（ChannelList）

和网络清单（NetworkList），且选择一个存在的PAN来连接。PAN选择的运算法则的规范将留给profile 描述，而且也许包含扩展的PAN ID。

### 3.1 网络层状态值

网络层确认原语通常都包括一个参数，这个参数记录回答请求原语的状态。网络层状态参数值如表 3.1 所示。

表 3.1

名称	值	描述
SUCCESS	0x00	请求执行成功
INVALID_PARAMETER	0xc1	从高层发出的原语无效或者超出范围
INVALID_REQUEST	0xc2	考虑到网络层目前的状态，高层发送的请求原语无效或者不能执行
NOT_PERMITTED	0xc3	NLME-JOIN.request原语不被接受
STARTUP_FAILURE	0xc4	NLME-NETWORK-FORMATION.request原语启动网络失败
ALREADY_PRESENT	0xc5	产生NLMEDIRECT-JOIN.request原语的设备的邻居表中已经存在有地址设备提供的NLMEDIRECT-JOIN.request原语
SYNC_FAILURE	0xc6	用来表明在MAC层NLME-SYNC.request原语失败
NEIGHBOR_TABLE_FULL	0xc7	NLME-JOIN-DIRECTLY.request失败，因为邻居表没有更多的空间
UNKNOWN_DEVICE	0xc8	NLME-LEAVE.request原语失败，因为产生原语的设备地址不在邻居表中的参数列表中
UNSUPPORTED_ATTRIBUTE	0xc9	NLME-GET.request or NLME-SET.request原语产生带有未知的属性标识符
NO_NETWORKS	0xca	没有检测到网络环境产生NLME-JOIN.request原语
LEAVE_UNCONFIRMED	0xcb	设备确认从网络出发失败
MAX_FRM_CNTR	0xcc	因为帧计数器达到最大值，所以输出帧安全处理失败
NO_KEY	0xcd	输出帧尝试安全处理且失败，因为对于处理没有有效的钥匙
BAD_CCM_OUTPUT	0xce	输出帧尝试安全处理且失败，因为安全设计产生一个错误的输出
NO_ROUTING_CAPACITY	0xcf	由于缺少路由表或者发现路由表能力，尝试发现路由失败
ROUTE_DISCOVERY_FAILED	0xd0	尝试发现路由失败，由于缺少路由能力
ROUTE_ERROR	0xd1	由于发送设备的路由失败，NLDE-DATA.request原语失败
BT_TABLE_FULL	0xd2	由于没有足够的空间在BTT，尝试发送一个广播帧或成员模式多点传送失败



FRAME_NOT_BUFFERED	0xd3	一个非成员多点传送帧丢弃未决路由发现
--------------------	------	--------------------

## 3.2 概况描述

### 3.2.1 网络层概述

ZigBee 网络层的主要功能就是提供一些必要的函数，确保 ZigBee 的 MAC 层（IEEE 802.15.4-2003）正常工作，并且为应用层提供合适的服务接口。为了向应用层提供其接口，网络层提供了两个必须的功能服务实体，它们分别为数据服务实体和管理服务实体。网络层数据实体（NLDE）通过网络层数据服务实体服务接入点（NLDE-SAP）提供数据传输服务，网络层管理实体（NLME）通过网络层管理实体服务接入点（NLME-SAP）提供网络管理服务。网络层管理实体利用网络层数据实体完成一些网络的管理工作，并且，网络层管理实体完成对网络信息库（NIB）的维护和管理，下面分别对它们的功能进行介绍。

#### 3.2.1.1 网络层数据实体（NLDE）

网络层数据实体为数据提供服务，在连个或者更多的设备之间传送数据时，将按照应用协议数据单元（APDU）的格式进行传送，并且这些设备必须在同一个网络中，即在同一个内部个域网中。

网络层数据实体提供如下服务：

- （1）生成网络层协议数据单元（NPDU）：网络层数据实体通过增加一个适当的协议头，从应用支持层协议数据单元中生成网络层的协议数据单元。
- （2）指定拓扑传输路由，网络层数据实体能够发送一个网络层的协议数据单元到一个合适的设备，该设备可能是最终目的通信设备，也可能是在通信链路中的一个中间通信设备。
- （3）安全：确保通信的真实性和机密性。

#### 3.2.1.2 网络层管理实体（NLME）

网络层管理实体提供网络管理服务，允许应用与堆栈相互作用。网络层管理实体应该提供如下服务：

- （1）配置一个新的设备：为保证设备正常工作的需要，设备应具有足够的堆栈，以满足配置的需要。配置选项包括对一个 ZigBee 协调器或者连接一个现有网络设备的初始化的操作。
- （2）初始化一个网络：使之具有建立一个新网络的能力。
- （3）连接和断开网络。具有连接或者断开一个网络的能力，以及为建立一个 ZigBee 协调器或者路由器，具有要求设备同网络断开的的能力。
- （4）寻址：ZigBee 协调器和路由器具有为新加入网络的设备分配地址的能力。
- （5）邻居设备发现：具有发现、记录和汇报有关一跳邻居设备信息的能力。
- （6）路由发现：具有发现和记录有效地传送信息的网络路由的能力。
- （7）接收控制：具有控制设备接收状态的能力，即控制接收机什么时间接收、接收时间的长短，以保证 MAC 层的同步或正正常接收等。



3.3 网络层服务协议

图 3.1 给出了网络层各组成部分和接口。

网络层通过两种服务接入点提供响应的两种服务。它们分别是网络层数据服务和网络层管理服务。网络层数据服务通过网络层数据实体服务接入点接入，网络层管理服务通过网络层管理实体服务接入点接入。这两种服务通过 MCPS-SAP 和 MLME-SPA 接口为 MAC 层提供接口。除此之外，在 NLME 和 NLDE 间还有一个接口使得 NLME 可以使用网络层数据服务。

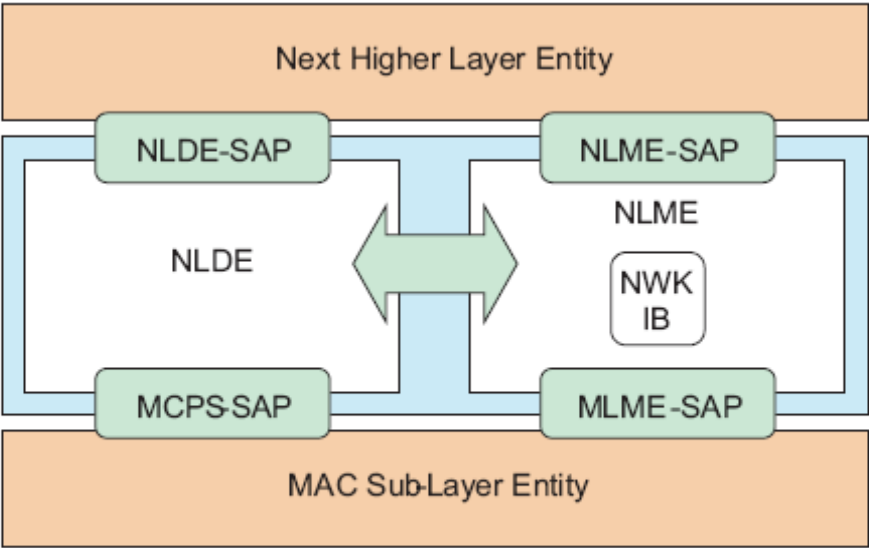


Figure 3.1 The NWK Layer Reference Model

3.3.1 网络层数据服务

网络层数据实体服务接入点支持对等应用实体之间的应用协议数据单元的传输。表 3.2 列出了网络层数据实体服务接入点支持的原语，下面小节就是对这些原语的讨论。

Table 3.2 NLDE-SAP Primitives

NLDE-SAP Primitive	Request	Confirm	Indication
NLDE-DATA	3.3.1.1	3.3.1.2	3.3.1.3

3.3.1.1 NLDE-DATA.request 原语

该原语请求从本地应用支持层实体到单个或者多个对等的网络支持层实体的协议数据单元传输。

3.3.1.1.1 服务原语的语法

该服务原语的语法如下所示：

NLDE-DATA.request	<pre> {     DstAddrMode,     DstAddr,     NsduLength,     Nsdu,     NsduHandle,     Radius,     NonmemberRadius,     DiscoverRoute,     SecurityEnable } </pre>
-------------------	---

表 3. 3 描述了 NLDE-DATA.request 函数原语的参数

**Table 3.3 NLDE-DATA.request Parameters**

Name	Type	Valid Range	Description
DstAddrMode	Integer	0x01 or 0x02	The type of destination address supplied by the DstAddr parameter; This may have one of the following two values:  0x01=16-bit multicast group address 0x02=16-bit NWK address of a device or a 16-bit broadcast address
DstAddr	16-bit Address	0x0000-0xFFFF	Destination address
NsduLength	Integer	< aMaxMACFrameSize - (nwkcMACFrameOverhead + nwkcMinHeaderOverhead	The number of octets comprising the NSDU to be transferred. This has been modified from the aMaxMACFrameSize limit specified in the IEEE 802.15.4 specification to take into account that the Zigbee network layer does not use the extended addressing modes. The effect of this is to free the unused portion of the header to be used for payload.
Nsdu	Set of Octets	-	The set of octets comprising the NSDU to be transferred
NsduHandle	Integer	0x00 – 0xff	The handle associated with the NSDU to be transmitted by the NWK layer entity
Radius	Unsigned Integer	0x00 – 0xff	The distance, in hops, that a frame will be allowed to travel through the network

**Table 3.3 NLDE-DATA.request Parameters (Continued)**

Name	Type	Valid Range	Description
NonmemberRadius	Integer	0x00 – 0x07	The distance, in hops, that a multicast frame will be relayed by nodes not a member of the group; A value of 0x07 is treated as infinity
DiscoverRoute	Integer	0x00 – 0x01	The DiscoverRoute parameter may be used to control route discovery operations for the transit of this frame (see sub-clause 3.7.3.5):  0x00 = suppress route discovery 0x01 = enable route discovery
SecurityEnable	Boolean	TRUE or FALSE	The SecurityEnable parameter may be used to enable NWK layer security processing for the current frame; If the security level specified in the NIB is 0, meaning no security, then this parameter will be ignored; Otherwise, a value of TRUE denotes that the security processing specified by the security level will be applied and a value of FALSE denotes that no security processing will be applied

#### 3.3.1.1.2 产生

当一个 NSDU 要传送到一个对等的应用支持层实体时，本地应用支持层实体就会生成该原语。

#### 3.3.1.1.3 接收

当一个不与网络连接的设备接收到该原语时，该设备网络层将发出一个状态参数为 INVALID-REQUEST 的 NLDE-DATA.confirm 原语。

网络层数据实体在接受到该原语时，为传送 NSDU 包，需要构造一个 NPDU 包。在处理过程中，如果网络层数据实体在发送 NSDU 包之前，先发送了 NLDE-DATA.confirm 原语，则将发起所有的后续处理。在构造新的 NPDU 过程中，网络层头的目的地址域设置为参数 DstAddr 所提供的值，源地址域设置为 MAC PIB 中属性 macShortAddress 的值。网络层帧头帧控制域中的路由发现域设置为 DiscoverRoute 参数的值。如果提供的 Radius 参数不为 0，那么它将设置在网络层帧头的 radius 域，如果值为 0，那么网络层帧头中的 radius 域设置 NWK IB 中 nwkMaxDepth 属性值的二倍。网络层将会生成一个如 3.7.2.1 小节所描述的系列号。这个序列号可以插入到网络层帧头的 sequence number 域。帧头的多点发送标志位将根据 DstAddrMode 的值设置。如果 DstAddrMode 的参数值为 0x01，网络层帧头将包含 multicast control 域，该域的设置如下：

- (1) 如果该节点是 DstAddr 参数所包含的节点，那么 multicast mode 域置为 0x01
- (2) 否则，multicast mode 域设为 0x00
- (3) non-member radius 和 max non-member radius 域按照 NonmemberRadius 的值设置

一旦构造好 NSDU 包，如果是单播，将按照 3.7.3.3 小节所描述的过程为 NSDU 包确定传输路由；如果是广播，则参见 3.7.4 小节；如果是多点通信，则参见 3.7.5.2 小节。当确定了 NSDU 包传输路由后，通过 MCPS-DATA.request 原语来发送 NSDU 包，在该原语中参数 SrcAddrMode 和 DstAddrMode 都设置为 0x02，表明适应 16 位的网络地址。参数 SrcPANId 和 DstPANId 应设置

为MAC PIB中的macPANId值。SrcAddr参数值设置为MAC PIB中的macShortAddr值。DstAddr参数值为由路由程序所决定的下一跳地址。当TxOptions与0x01相与时，该参数值应为非零值，表示发送需要确认。在接收到MCPS-DATA.confirm原语时，网络层数据试题发送NLDE-DATA.confirm原语，该原语中的参数状态为MAC层所接收到的状态。

如果在网络层信息数据库（NIB）中所确定的网络安全级别标准为一个非零值，并且SecurityEnable值为TRUE，则在帧发送之前，按照4.4小节所描述对帧进行安全处理。否则，网络层不对该帧进行安全处理。如果安全处理已经进行了，但是由于某些原因而失败了，那么，将丢弃该帧，并且网络层数据实体将发送NLDE-DATA.confirm原语，该原语的状态参数为安全方案所返回的值。

3.3.1.2 NLDE-DATA.confirm 原语

该原语提供了从本地应用支持层实体到一个对等应用支持成实体传送NSDU包请求原语的结果。

3.3.1.2.1 服务原语的语法

该原语的语法如下所示：

NLDE-DATA.confirm	{ NsduHandle, Status TxTime }
-------------------	---

表3.4详细描述了NLDE-DATA.confirm原语的参数。

Table 3.4 NLDE-DATA.confirm Parameters

Name	Type	Valid Range	Description
NsduHandle	Integer	0x00 – 0xff	The handle associated with the NSDU being confirmed
Status	Status	INVALID_REQUEST, MAX_FRM_COUNTER, NO_KEY, BAD_CCM_OUTPUT, ROUTE_ERROR, BT_TABLE_FULL, FRAME_NOT_BUFFERED or any status values returned from security suite or the MCPS-DATA.confirm primitive (see [B1])	The status of the corresponding request
TxTime	Integer	Implementation specific	A time indication for the transmitted packet based on the local clock. The time should be based on the same point for each transmitted packet in a given implementation. This value is only provided if <i>nwkTimeStamp</i> is set to TRUE

3.3.1.2.2 产生

该原语为本地网络层数据实体对接收到NLDE-DATA.request原语而产生的响应。

Status域将反映相应的请求结果，详见3.3.1.2.3小节。

3.3.1.2.3 接收

接收到该原语，开始设备的APS子层将被通知传输请求的结果。如果传输成功了，那么status参数为SUCCESS。否则，status参数表明传输的错误。

3.3.1.3 NLDE-DATA.indication原语

该原语表示一个NSDU包从网络层到本地应用支持层实体的传送。

3.3.1.3.1 服务原语的语法

该原语的语法如下：

NLDE-DATA.indication	{ DstAddrMode, DstAddr, SrcAddr, NsduLength, Nsdu, LinkQuality RxTime }
----------------------	---

表3.5描述了NLDE-DATA.request原语的参数。

Table 3.5 NLDE-DATA.indication Parameters

Name	Type	Valid Range	Description
DstAddrMode	Integer	0x01 or 0x02	The type of destination address supplied by the DstAddr parameter; This may have one of the following two values:  0x01=16-bit multicast group address 0x02=16-bit NWK address of a device or a 16-bit broadcast address
DstAddr	16-bit Address	0x0000-0xFFFF	The destination address where the NSDU is sent
SrcAddr	16-bit Device address	Any valid device address except a broadcast address	The individual device address from which the NSDU originated

**Table 3.5 NLDE-DATA.indication Parameters (Continued)**

NsduLength	Integer	$\leq \text{aMaxMACFrameSize} - (\text{nwkMACFrameOverhead} + \text{nwkMinHeaderOverhead})$	The number of octets comprising the NSDU being indicated. This has been modified from the aMaxMACFrameSize limit specified in the IEEE 802.15.4 specification to take into account that the Zigbee network layer does not use the extended addressing modes. The effect of this is to free the unused portion of the header to be used for payload.
Nsdu	Set of octets	–	The set of octets comprising the NSDU being indicated
LinkQuality	Integer	0x00 – 0xff	The link quality indication delivered by the MAC on receipt of this frame as a parameter of the MCPS-DATA.indication primitive (see [B1])
RxTime	Integer	Implementation specific	A time indication for the received packet based on the local clock. The time should be based on the same point for each received packet on a given implementation. This value is only provided if <i>nwkTimeStamp</i> is set to TRUE

#### 3.3.1.3.2 产生

当本地MAC层实体接收到一个适当地址的数据帧时，就生成该原语，并发送给应用支持层。

#### 3.3.1.3.3 接收

当应用支持层接收到该原语时，则被通知一个数据帧到达设备，就可得到设备所接收的数据。

#### 3.3.1.3.4 网络管理服务

网络层管理实体服务接入点为其上层和网络层管理实体之间传送管理命令提供接口。表3.6列出了NLME所支持的NLME-SPA原语，下面的小节详细介绍了这些原语。

Table 3.6 Summary of the Primitives Accessed Through the NLME-SAP

Name	Sub-clause Number in This Specification			
	Request	Indication	Response	Confirm
NLME-NETWORK-DISCOVERY	3.3.2.1			3.3.2.2
NLME-NETWORK-FORMATION	3.3.3.1			3.3.3.2
NLME-PERMIT-JOINING	3.3.4.1			3.3.4.2
NLME-START-ROUTER	3.3.5.1			3.3.5.2
NLME-ED-SCAN	3.3.6.1			3.3.6.2
NLME-JOIN	3.3.7.1	3.3.7.2		3.3.7.3
NLME-DIRECT-JOIN	3.3.8.1			3.3.8.2
NLME-LEAVE	3.3.9.1	3.3.9.2		3.3.9.3
NLME-RESET	3.3.10.1			3.3.10.2
NLME-SYNC	3.3.11.1	3.3.11.2		3.3.11.3
NLME-GET	3.3.12.1			3.3.12.2
NLME-SET	3.3.12.3			3.3.12.4
NLME-ROUTE-ERROR		3.3.13.1		
NLME-ROUTE-DISCOVERY	3.3.14.1			3.3.14.2
NLME-START-BACKOFF	3.3.15.1			

3.3.2 网络发现

网络层管理实体服务接入点支持运行网络的发现。采用NLME-NETWORK-DISCOVERY原语来发现网络。

3.3.2.1 NLME-NETWORK-DISCOVERY.request原语

该原语支持网络层上层应用该原语来发现在POS范围内正在运行的网络。

3.3.2.1.1 服务原语的语法

该原语的语法如下：

NLME-NETWORK-DISCOVERY.request	{ ScanChannels, ScanDuration }
--------------------------------	---

表3.7详细描述了NLME-NETWORK-DISCOVERY.request原语的参数。



**Table 3.7 NLME-NETWORK-DISCOVERY.request Parameters**

Name	Type	Valid Range	Description
ScanChannels	Bitmap	32-bit field	The five most significant bits (b27,..., b31) are reserved; the 27 least significant bits (b0, b1,... b26) indicate which channels are to be scanned (1 = scan, 0 = do not scan) for each of the 27 valid channels (see [B1])
ScanDuration	Integer	0x00 – 0x0e	A value used to calculate the length of time to spend scanning each channel;  The time spent scanning each channel is ( <i>aBaseSuperframeDuration</i> * (2n + 1)) symbols, where n is the value of the ScanDuration parameter; for more information on MAC sub-layer scanning (see [B1])

#### 3.3.2.1.2 产生

该原语由ZigBee设备网络层上层产生，发送给它的网络层管理实体，请求网络层发现当前在POS正在运行的网络。

#### 3.3.2.1.3 接收

网络层在接收到该原语后，将通过检查ScanChannels参数确定的信道以及ScanDuration参数所确定的扫描时间，发现在POS中正在运行的网络。通过MLME-SCAN.request原语进行扫描。

在接收到MLME-SCAN.confirm原语后，网络层管理实体发送NLME-NETWORK-DISCOVERY.confirm原语，其原语参数为发现网络信息以及随MLME-SCAN.confirm原语返回的状态参数值。

#### 3.3.2.2 NLME-NETWORK-DISCOVERY.confirm 原语

该原语返回网络发现操作的结果。

##### 3.3.2.2.1 服务原语的语法

该原语的语法如下：

---

NLME-NETWORK-DISCOVERY.confirm	{
	NetworkCount,
	NetworkDescriptor,
	Status
	}

---

表3.8详细描述了NLME-NETWORK-DISCOVERY.confirm原语的参数。

**Table 3.8 NLME-NETWORK-DISCOVERY.confirm Parameters**

Name	Type	Valid Range	Description
NetworkCount	Integer	0x00 – 0xff	Gives the number of networks discovered by the search
NetworkDescriptor	List of network descriptors	The list contains the number of elements given by the NetworkCount parameter	A list of descriptors, one for each of the networks discovered; Table 3.9 gives a detailed account of the contents of each item
Status	Status	Any Status value returned with the MLME-SCAN.confirm primitive	See [B1]

表3.9给出了NetworkDescriptor参数中网络描述符所包含的具体内容。

**Table 3.9 Network Descriptor Information Fields**

Name	Type	Valid Range	Description
PanID	Integer	0x0000 – 0x3fff	The 16-bit PAN identifier of the discovered network; the 2 highest-order bits of this parameter are reserved and shall be set to 0
ExtendedPanID	Integer	0x0000000000000001 - 0xfffffffffffffffe	The 64-bit PAN identifier of the network.
LogicalChannel	Integer	Selected from the available logical channels supported by the PHY (see [B1])	The current logical channel occupied by the network

**Table 3.9 Network Descriptor Information Fields (Continued)**

Name	Type	Valid Range	Description
StackProfile	Integer	0x00 – 0x0f	A ZigBee stack profile identifier indicating the stack profile in use in the discovered network
ZigBeeVersion	Integer	0x00 – 0x0f	The version of the ZigBee protocol in use in the discovered network
BeaconOrder	Integer	0x00 – 0x0f	This specifies how often the MAC sub-layer beacon is to be transmitted by a given device on the network; for a discussion of MAC sub-layer beacon order (see [B1])
SuperframeOrder	Integer	0x00 – 0x0f	For beacon-oriented networks, that is, beacon order < 15, this specifies the length of the active period of the superframe; for a discussion of MAC sub-layer superframe order (see [B1])
PermitJoining	Boolean	TRUE or FALSE	A value of TRUE indicates that at least one ZigBee router on the network currently permits joining; That is, its NWK has been issued an NLME-PERMIT-JOINING primitive and, the time limit if given, has not yet expired

#### 3.3.2.2.2 产生

当NLME-NETWORK-DISCOVERY.request原语执行完成后，网络层管理实体生成该原语，并发送给网络上层。

#### 3.3.2.2.3 接收

其上层接收到该原语后，就可得到网络的搜索结果。

### 3.3.3 网络的形成

本小节原语定义了一个设备的应用层如何初始化，使其自身成为一个新的ZigBee网络协调器。

#### 3.3.3.1 NLME-NETWORK-FORMATION.request 原语

该原语允许高层使用该原语请求设备发起一个新的ZigBee网络。并将其自身作为ZigBee协调器。

##### 3.3.3.1.1 服务原语的语法

该原语的语法如下：

NLME-NETWORK-FORMATION.request	{
	ScanChannels,
	ScanDuration
	}

表3.10详细描述了NLME-NETWORK-FORMATION.request原语的参数。

**Table 3.10 NLME-NETWORK-FORMATION.request Parameters**

Name	Type	Valid Range	Description
ScanChannels	Bitmap	32-bit field	The five most significant bits (b27,..., b31) are reserved. The 27 least significant bits (b0, b1,... b26) indicate which channels are to be scanned in preparation for starting a network (1=scan, 0=do not scan) for each of the 27 valid channels (see [B1])
ScanDuration	Integer	0x00 – 0x0e	A value used to calculate the length of time to spend scanning each channel;  The time spent scanning each channel is ( $aBaseSuperframeDuration * (2^n + 1)$ ) symbols, where $n$ is the value of the ScanDuration parameter (see [B1])

### 3.3.3.1.2 产生

该原语由具有ZigBee协调器能力设备的应用层生成，发送给它的网络层管理实体，请求初始化设备，使之成为一个新网络的协调器。

### 3.3.3.1.3 接收

在网络中，当一个没有ZigBee协调器能力的设备接收到该原语时，网络层管理实体就会返回状态参数为INVALID-REQUEST的NLME-NETWORK-FORMATION.confirm原语。

如果设备被初始化为ZigBee协调器，网络层管理实体请求MAC层首先执行一个能量检测扫描，然后在所指定的信道上执行主动扫描。为了执行扫描任务，网络层管理实体将向MAC发送ScanType参数设置为能量检测扫描的MLME-SCAN.request原语；然后，再发送ScanType为主动扫描的MLME-SCAN.request原语。在主动扫描完成以后，网络层管理实体从MAC层接收到MLME-SCAN.confirm原语，并且选择一个合适的信道。网络层将选择一个个域网标识符，并且确保其不会与所选择信道的现有网络个域网标识符参数产生冲突。一旦合适的信道和个域网标识符PANId确定后，网络层管理实体将选择0x0000作为16位的短MAC地址，并且告知MAC层。为了实现该目的，网络层管理实体将向MAC层大宋MLME-SET.request原语来设置MAC PIB中的macShortAddress属性。如果NIB中的属性nwExtendedPANId值为0x0000000000000000，那么该属性将被设置为MAC层的aExtendedAddress值。如果不能找到合适的信道和个域网标识符PANId，网络层管理实体将会发出状态参数为START\_FAILURE的NLME-NETWORK-FORMATION.confirm原语。

如果在上层的请求中只提供了一个信道，那么网络层管理实体在开始形成网络前不需要进行能量检测扫描。主动扫描仍需要进行，确保所选择的个域网标识符不与现有网络中的标识符发生冲突。

开始一个新的网络，网络层管理实体向MAC层发送MLME-START.request原语。MLME-START.request原语的PANCoordinator参数设置为TRUE。MLME-START.request原语中的BeaconOrder和SuperframeOrder参数都设置为15，表明没有超帧信标。MLME-START.request中的参数CoordRealignment设置为False。在接收到相应的MLME-START.confirm原语时，网络层管路实体将会向其上层发送NLME-NETWORK-FORMATION.confirm原语，其中的状态参数为MLME-START.confirm原语所返回的状态参数值。

### 3.3.3.2 NLME-NETWORK-FORMATION.confirm 原语

该原语返回在网络中初始化一个ZigBee协调器请求的执行结果。

#### 3.3.3.2.1 服务原语的语法

该原语的语法如下：

NLME-NETWORK-FORMATION.confirm	{ Status }
--------------------------------	------------------

表3. 11详细描述了NLME-NETWORK-FORMATION.confirm原语的参数。

**Table 3.11 NLME-NETWORK-FORMATION.confirm Parameters**

Name	Type	Valid Range	Description
Status	Status	INVALID_REQUEST, STARTUP_FAILURE or any status value returned from the MLME-START.confirm primitive	The result of the attempt to initialize a ZigBee coordinator

3. 3. 3. 2. 2 产生

该原语由网络层管理实体生成，作为对NLME-NETWORK-FORMATION.request原语的响应，发送给其上层。该原语返回的状态为INVALID\_REQUEST、STARTUP\_FAILURE或者MLME-START.confirm原语所返回的状态。3. 3. 3. 1. 3描述了在那些条件下返回这些值。

3. 3. 3. 2. 3 接收

接收到该原语，上层就可得知初始化一个ZigBee协调器的执行结果。如果成功执行了请求原语，则状态参数设置为SUCCESS。否则，状态参数为错误状态。

3. 3. 4 允许设备连接

该原语定义了ZigBee协调器或路由器的上层如何设置其设备允许其他设备同其网络连接。

3. 3. 4. 1 NLME-PERMIT-JOINING.request 原语

该原语允许ZigBee协调其或路由器上层设定其MAC层连接许可标志，在一定期间内，允许其他设备同网络连接。

3. 3. 4. 1. 1 服务原语的语法

该原语的语法如下：

NLME-PERMIT-JOINING.request	{ PermitDuration }
-----------------------------	--------------------------

表3. 12详细描述了NLME-PERMIT-JOINING.request原语的参数。

**Table 3.12 NLME-PERMIT-JOINING.request Parameters**

Name	Type	Valid Range	Description
PermitDuration	Integer	0x00 – 0xff	The length of time in seconds during which the ZigBee coordinator or router will allow associations; The value 0x00 and 0xff indicate that permission is disabled or enabled, respectively, without a specified time limit

3. 3. 4. 1. 3 产生

当ZigBee协调器或路由器上层希望其他设备加入或阻止加入其网络时，将生成该原语，并传送给网络层管理实体。

3.3.4.1.3 接收

仅允许 ZigBee 协调器或路由器的上层发送该原语。如果 ZigBee 终端设备的网络层管理实体收到该原语，则将返回状态为 INVALID\_REQUEST 的 NLME-PERMIT-JOINING.confirm 原语。

一旦网络层管理实体接收到参数 PermitDuration 的值为 0x00 的原语，则通过向 MAC 层发送 MLME-SET.request 原语将 MAC 层的 PIB 的 macAssociationPermit 属性设置为 FALSE。一旦收到 MLME-SET.confirm 原语，则网络层管理实体发送 NLME-PERMITJOINING.Confirm 原语，将其状态值设置为从 MAC 层所收到的状态。

一旦网络层管理实体接收到参数 PermitDuration 的值为 0xff 的原语，则通过向 MAC 层发送 MLME-SET.request 原语将 MAC 层的 PIB 的 macAssociationPermit 属性设置为 TRUE。一旦收到 MLME-SET.confirm 原语，则网络层管理实体发送 NLME-PERMITJOINING.Confirm 原语，将其状态值设置为从 MAC 层所收到的状态。

如果收到参数 PermitDuration 的值为除 0x00 或 0xFF 外的值，则网络层管理实体 MAC 层的 PIB 的 macAssociationPermit 属性设置为 TRUE。当网络层管理实体收到 MLME-SET.confirm 原语后，将会启动一个计时器，在 PermitDuration 秒后，停止计时。一旦计时器启动，网络层管理实体将发送 NLME-PERMIT-JOINING.confirm 原语，其状态值设置为从 MAC 层所得到的状态值。如果计时器超时，网络层管理实体将发送参数 macAssociationPermit 为 FALSE 的 MLME-SET.request 原语。

任何一个由上层发出的 NLME-PERMIT-JOINING.request 原语，可以取代所有一切的请求。

3.3.4.2 NLME-PERMIT-JOINING.confirm 原语

该原语向 ZigBee 协调器或路由器的上层返回允许设备连接网络请求原语的执行结果。

3.3.4.2.1 服务原语的语法

该原语的语法如下：

NLME-PERMIT-JOINING.confirm	{
	Status
	}

表 3.13 详细描述了 NLME-PERMIT-JOINING.confirm 原语的参数。

Table 3.13 NLME-PERMIT-JOINING.confirm Parameters

Name	Type	Valid Range	Description
Status	Status	INVALID_REQUEST or any status returned from the MLME-SET.confirm primitive (see [B1])	The status of the corresponding request

3.3.4.2.2 产生

该原语由 ZigBee 协调器或路由器初始化的网络管理实体生成，并且向上层发送作为对 NLME-PERMIT-JOINING.request 原语的确认。其状态参数既可以为 MAC 层所收到的状态，也可以 INVALID\_REQUEST 的出错代码。这些状态值的原因详见 3.3.4.1 小节。

3.3.4.2.3 接收

当接收到该原语后，所初始化的设备上层即可得知允许其他设备连接网络请求原语的执行结果。

3.3.5 路由器初始化

该原语允许一个新加入网络的 ZigBee 路由器开始参加 ZigBee 路由器的活动，包括数据

帧的路由、路由发现、接收其他设备加入网络的请求。

3.3.5.1 NLME-START-ROUTER.request 原语

该原语允许一个 ZigBee 路由器的上层发起路由。

3.3.5.1.1 服务原语的语法

该原语的语法如下：

NLME-START-ROUTER.request	{
	}

3.3.5.1.2 产生

该原语由新设备的网络层管理实体上层生成，并发出给网络管理实体要求将设备初始化为 ZigBee 路由器。

3.3.5.1.3 接收

如果不是作为网络 ZigBee 路由器的设备接收到该原语后，网络层管理实体将返回状态参数为 INVALID\_REQUEST 的 NLME-START-ROUTER.confirm 原语。

为初始化一个路由，网络层管理实体向 MAC 层发送 MLME-START.request 原语，MLME-START.request 原语中的 BeaconOrder 和 SuperframeOrder 参数值设置为 15，表明 beaconless 操作。MLME-START.request 原语的 CoordRealignment 参数设置为 FALSE。

当网络层管理实体收到相应的 MLME-START.confirm 原语，将向上层发送 NLME-START-ROUTER.confirm 原语，其中其状态值与 MLME-START.confirm 原语中的状态值一样。只有当 MLME-START.confirm 原语返回的状态值为 SUCCESS 时，设备开始作为 ZigBee 路由器开始工作，包括数据帧的路由、路由发现、接收设备加入网络的请求。否则，设备不允许做这些工作。

3.3.5.2 NLME-START-ROUTER.confirm 原语

该原语返回执行 ZigBee 路由器配置初始化的结果。

3.3.5.2.1 服务原语的语法

该原语的语法如下：

NLME-START-ROUTER.confirm	{
	Status
	}

表 3.14 描述了 NLME-START-ROUTER.confirm 原语的参数。

Table 3.14 NLME-START-ROUTER.confirm Parameters

Name	Type	Valid Range	Description
Status	Status	INVALID_REQUEST or any status value returned from the MLME-START.confirm primitive	The result of the attempt to initialize a ZigBee router

3.3.5.2.2 产生

该原语由网络层管理实体生成，在接收到 NLME-START-ROUTER.request 原语时，向上层发送该原语作为响应。该原语返回的参数值为 INVALID\_REQUEST 或者为 MLME-START.confirm 所返回的任何状态值。3.3.5.1.3 小节描述了在哪些条件下返回这些值。

3.3.5.2.3 接收

接收到该原语，上层就得到 ZigBee 路由器初始化请求的结果。如果网络层管理实体已



经成功设置，其返回的参数状态为 SUCCESS，否则，参数状态为出错信息。

3.3.6 能量扫描

该原语定义了设备的上层如何操作能量扫描

3.3.6.1 NLME-ED-SCAN.request 原语

该原语允许上层请求本地信道进行能量扫描。

3.3.6.1.1 服务原语的语法

原语的语法如下：

NLME-ED-SCAN.request	{ ScanChannels, ScanDuration, }
----------------------	--

表 3.15 详细描述了服务原语的参数。

Table 3.15 NLME-ED-SCAN.request Parameters

Name	Type	Valid Range	Description
ScanChannels	Bitmap	32-bit field	The five most significant bits (b27,..., b31) are reserved. The 27 least significant bits (b0, b1,... b26) indicate which channels are to be scanned (1=scan, 0=do not scan) for each of the 27 valid channels (see [B1])
ScanDuration	Integer	0x00-0x0e	A value used to calculate the length of time to spend scanning each channel; The time spent scanning each channel is ( <i>aBaseSuperframeDuration</i> * ( $2^n + 1$ )) symbols, where <i>n</i> is the value of the ScanDuration parameter [B1]

3.3.6.1.2 产生

上层产生该原语要求：

- 管理信道的能量扫描

3.3.6.1.3 接收

如果是连接到网络的设备接收到该原语，设备将停止接收任何新的 NLDE-DATA.request 原语，返回错误代码 INVALID REQUEST。完成未解决的 NLDE-DATA.request 原语。一旦完成了未解决的 NLDE-DATA.request 原语，设备将临时停止网络操作，进行能量扫描。网络层管理实体向 MAC 层发送参数 ScanType 表示为能量扫描，参数 ScanChannels 和 ScanDuration 根据网络层管理实体请求设置的 MLME-SCAN.request 原语。

3.3.6.2 NLME-ED-SCAN.confirm 原语

该原语返回上层请求能量扫描的结果。

3.3.6.2.1 服务原语的语法

该原语的语法如下：

NLME-ED-SCAN.confirm	{ Status ScanChannels, EnergyDetectList }
----------------------	---

表 3.16 详细描述了该原语的参数。

**Table 3.16 NLME-ED-SCAN.confirm**

Name	Type	Valid Range	Description
Status	Status	SUCCESS, or any valid code from the MAC	The status of the request.
ScannedChannels	Bitmap	32 bit field	The five most significant bits (b27,..., b31) are reserved. The 27 least significant bits (b0, b1,... b26) indicate which channels are to be scanned (1=scan, 0=do not scan) for each of the 27 valid channels (see [B1])
EnergyDetectList	List of integers	0x00-0xff for each integer	The list of energy measurements in accordance with B1, one for each channel.

### 3.3.6.2.2 产生

该原语由 ZigBee 设备的网络层管理实体生成,作为对 NLME-ED-SCAN.request 原语的响应。其状态表明从 MAC 层收到的 MLME-SCAN.confirm 原语所返回的状态。ScannedChannels 表明那个信道被扫描了 (1=信道已扫描)。EnergyDetectList 包含信道扫描的结果 (0x00-0xff)。其值与 MAC 层硬件表示为[dBm]无关。(e.g. [-185 dBm .. 70dBm])参考IEEE802.15.4-2003。

### 3.3.6.2.3 接收

接收到该原语,上层得到能量扫描的结果。

### 3.3.7 设备同网络连接

该原语给定了设备同网络连接的方式:

- (1) 通过联合方式请求连接网络
- (2) 直接请求连接网络
- (3) 如果成为孤点设备,请求重新连接网络

### 3.3.7.1 NLME-JOIN.request 原语

该原语允许设备上层通过该原语以直接或间接方式请求连接网络,或者当设备为孤点设备时,请求重新连接网络。或者在一个网络中为设备改变操作的信道。

#### 3.3.7.1.1 服务原语的语法

该原语的语法如下:

NLME-JOIN.request	{ ExtendedPANId, JoinAsRouter, RejoinNetwork, ScanChannels, ScanDuration, PowerSource, RxOnWhenIdle }
-------------------	---

表 3.17 详细描述了 NLME-JOIN.request 原语的参数。

**Table 3.17 NLME-JOIN.request Parameters**

Name	Type	Valid Range	Description
ExtendedPANId	Integer	0x0000000000000000 001 – 0xfffffffffffffffe	The 64-bit PAN identifier of the network to join
JoinAsRouter	Boolean	TRUE or FALSE	The parameter is TRUE if the device is attempting to join the network in the capacity of a ZigBee router; Otherwise, it is FALSE; The parameter is valid in requests to join through association and ignored in requests to join directly or to re-join through orphaning
RejoinNetwork	Integer	0x00 – 0x02	This parameter controls the method of joining the network.  The parameter is 0x00 if the device is requesting to join a network through association.  The parameter is 0x01 if the device is joining directly or rejoining the network using the orphaning procedure.  The parameter is 0x02 if the device is joining the network using the NWK rejoining procedure.  The parameter is 0x03 if the device is to change the operational network channel to that identified in the ScanChannel parameter.
ScanChannels	Bitmap	32-bit field	The five most significant bits (b27,..., b31) are reserved. The 27 least significant bits (b0, b1,... b26) indicate which channels are to be scanned (1=scan, 0=do not scan) for each of the 27 valid channels (see [B1])

**Table 3.17 NLME-JOIN.request Parameters (Continued)**

Name	Type	Valid Range	Description
ScanDuration	Integer	0x00-0x0e	A value used to calculate the length of time to spend scanning each channel; The time spent scanning each channel is $(aBaseSuperframeDuration * (2^n + 1))$ symbols, where $n$ is the value of the ScanDuration parameter [B1]
PowerSource	Integer	0x00 – 0x01	This parameter becomes a part of the CapabilityInformation parameter passed to the MLME-ASSOCIATE.request primitive that is generated as the result of a successful executing of a NWK join. The values are:  0x01 = Mains-powered device 0x00 = other power source (see [B1])
RxOnWhenIdle	Boolean	TRUE or FALSE	This parameter indicates whether the device can be expected to receive packets over the air during idle portions of the CAP. The values are:  TRUE = The receiver is enabled when the device is idle FALSE = The receiver may be disabled when the device is idle  RxOnWhenIdle shall have a value of TRUE for ZigBee coordinators and ZigBee routers.

### 3.3.7.1.2 产生

设备的上层使用该原语请求：

- (1) 通过使用 MAC 层连接过程请求同新网络连接
- (2) 直接使用 MAC 层孤点过程请求连接网络
- (3) 在成为孤点设备后，完成设备位置确定，并且重新连接网络
- (4) 为连接到网络的设备改变操作信道

### 3.3.7.1.3 接收

如果收到该原语的设备已经同网络连接，并且RejoinNetwork参数为0x00，则网络管理实体将返回参数状态为INVALID\_REQUEST的NLME-JOIN.confirm原语。

如果收到该原语的设备目前还没有同网络连接，并且RejoinNetwork参数为0x00，则设备尝试连接由参数ExtendedPANId所指定的网络。网络层管理实体发送MLME-ASSOCIATE.request原语，其中参数CoordAddress设置为在它的邻居表中的路由器地址，满足如下条件：

- (1) 路由器属于参数CoordAddress所标识的网络
- (2) 路由器对连接请求开发，is advertising capacity of the correctdevice type
- (3) 当按照3.7.3.1小节所描述的计算方法，所计算连接成本最大为3时，设备收到帧的链路质量。

如果设备存在于邻居表中，且满足上述条件，原语MLME-ASSOCIATE.request中的LogicalChannel参数设置为邻居表中的地址，该地址与协调器地址的潜在父节点地址相对应。CapabilityInformation参数的位字段如表3.18所示。这里所收集的性能信息作为网络

信息库的属性nwkCapabilityInformation存储起来（见表3.42）。如果多台设备满足上述要求，网络信息库中的nwkAddrAlloc属性为TRUE，则连接设备将选择最小深度的父节点。

**Table 3.18 Capability Information Bit-fields**

Bit	Name	Description
0	Alternate PAN coordinator	This field will always have a value of 0 in implementations of this specification
1	Device type	This field will have a value of 1 if the joining device is a ZigBee router and the JoinAsRouter parameter has a value of TRUE; It will have a value of 0 if the device is a ZigBee end device or else a router-capable device that is joining as an end device
2	Power source	This field shall be set to the value of lowest-order bit of the PowerSource parameter passed to the NLME-JOIN-request primitive; The values are: 0x01 = Mains-powered device 0x00 = other power source
3	Receiver on when idle	This field shall be set to the value of the lowest-order bit of the RxOnWhenIdle parameter passed to the NLME-JOIN.request primitive. 0x01 = The receiver is enabled when the device is idle 0x00 = The receiver may be disabled when the device is idle

**Table 3.18 Capability Information Bit-fields (Continued)**

Bit	Name	Description
4 – 5	Reserved	This field will always have a value of 0 in implementations of this specification
6	Security capability	This field will always have a value of 0 in the implementations of this specification indicating MAC security is disabled.
7	Allocate address	This field will always have a value of 1 in implementations of this specification, indicating that the joining device must be issued a 16-bit short address

如果在邻居表中不存在符合条件的设备，则网络层发送状态为NOT\_PERMITTED的NLME-JOIN.confirm原语。否则，网络层管理实体发送状态与收到MLME-ASSOCIATE.confirm原语中的状态相一致的NLME-JOIN.confirm原语。

如果 RejoinNetwork 参数的值为 0x00，且参数 JoinAsRouter 的值为 TRUE，则设备将作为一个 ZigBee 路由器运行。如果参数 JoinAsRouter 的值为 FALSE，则设备作为终端设备，不参与路由选择。

如果设备接收到该原语，且参数 RejoinNetwork 值为 0x01，则发送 MLME-SCAN.request 原语，其参数 ScanType 设置为孤点扫描，ScanChannels 和 ScanDuration 参数与 NLME-

JOIN.request 原语的参数一致。网络层管理实体接收到 MLME-SCAN.confirm 原语，则发送 NLME-JOIN.confirm 原语，如果设备没有能力找到要连接的网络，其状态值为 NO\_NETWORKS，否则参数状态为 MLME-SCAN.confirm 原语所返回的状态值。

如果没有同网络连接的设备接收到该原语，并且 RejoinNetwork 参数值为 0x02，则网络层管理实体发送状态参数为 INVALID\_REQUEST 的 NLME-JOIN.confirm 原语。

如果当前与网络连接的设备接收到该原语，其 RejoinNetwork 参数值为 0x02，则设备试图重新与当前的网络连接。在这种情况下，当下面情况为真时，网络层管理实体通过向它的邻居表中的路由地址发送重新建立网络连接请求命令初始化重新建立网络连接：

- 1. 路由器有能力接收 JoinAsRouter 参数定义的设备类型
- 2. 当按照 3.3.7.1 小节描述的计算方法，所计算连接成本最大为 3 时，设备收到帧的链路质量。
- 3. 如果网络信息库中的属性 nwkAddrAlloc 值为 0x00，并且由多于一个满足上述两个条件的潜在的父节点存在，则连接设备将选择最小深度的树根。

如果设备存在于邻居表中，且满足上述条件，重新连接请求命令的目的地址设置为潜在的父节点的网络地址。参数 CapabilityInformation 位如表 3.18 所示。这里的能力信息如网络信息库中属性 nwkCapabilityInformation 所示。（见表 3.42）。

如果在邻居表中不存在符合条件的设备，则网络层发送状态为 NOT\_PERMITTED 的 NLME-JOIN.confirm 原语。否则，网络层管理实体发送状态与收到重新连接响应命令状态参数值一致的 NLME-JOIN.confirm 原语。

一旦设备成功同网络连接，它将把网络信息库中属性 nwkExtendedPANID 的值设置为连接网络的 PAN 标识符。

如果设备接收到该原语，且参数 RejoinNetwork 值为 0x03，设备试图把操作信道改变为参数 ScanChannel 所提供的信道。如果在参数 ScanChannel 中，提供多个信道，网络层管理实体将发送状态参数为 INVALID\_REQUEST 的 NLME-JOIN.confirm 原语。否则，网络层管理实体发送 NLME-JOIN.confirm 原语，其状态参数为从转换的信道接收的状态参数值。

3.3.7.2 NLME-JION.indication 原语

当一个新设备通过联合方式或者按照 3.7.1.3.3 小节所描述的重新连接的方式连接网络成功后，就发送该原语通知 ZigBee 协调器或路由器的上层。

3.3.7.2.1 服务原语的语法

该原语的语法如下：

NLME-JOIN.indication	{
	ShortAddress,
	ExtendedAddress,
	CapabilityInformation,
	SecureJoin
	}

表 3.19 详细描述了 NLME-JION.indication 原语的参数。

**Table 3.19 NLME-JOIN.indication Parameters**

Name	Type	Valid Range	Description
ShortAddress	Network address	0x0001 – 0xffff7	The network address of an entity that has been added to the network
ExtendedAddress	64-bit IEEE address	Any 64-bit, IEEE address	The 64-bit IEEE address of an entity that has been added to the network
CapabilityInformation	Bitmap	See [B1]	Specifies the operational capabilities of the joining device
SecureJoin	Boolean	TRUE or FALSE	This parameter will be TRUE if the join was performed in a secure manner; Otherwise this parameter will be FALSE

#### 3.3.7.2.2 产生

在通过如表 3.31 所示的 MAC 层的联合方式成功的将一个新的设备连接到网络或如表 3.36 所示的网络层管理实体的重新连接方式将设备重新连接网络成功后，ZigBee 协调器和路由器的网络层管理实体生成该原语，并向其上层传送。

#### 3.3.7.2.3 接收

设备上层收到该原语就可得知一个新的设备已经成功地连接到本网络。

#### 3.3.7.3 NLME-JOIN.confirm 原语

设备上层通过该原语可得知其请求连接网络的结果。

##### 3.3.7.3.1 服务原语的语法

该原语的语法如下：

NLME-JOIN.confirm	{ ShortAddress, ExtendedPANId, HaveNetworkKey ActiveChannel Status }
-------------------	--

表 3.20 详细描述了 NLME-JOIN.confirm 原语的参数



**Table 3.20 NLME-JOIN.confirm Parameters**

Name	Type	Valid Range	Description
ShortAddress	Integer	0x0001 – 0xFFFF	The 16-bit short address that was allocated to this device; This parameter will be equal to 0xFFFF if the join attempt was unsuccessful
ExtendedPAN Id	Integer	0x0000000000000001 – 0xfffffffffffffffe	The 64-bit extended PAN identifier for the network of which the device is now a member.
HaveNetwork Key	Boolean	TRUE or FALSE	The parameter is TRUE if this device and its parent are known to have the same network key sequence number and FALSE otherwise.
ActiveChannel	Integer	0x00 -0xff	The channel used when joining the network.
Status	Status	INVALID_REQUEST, NOT_PERMITTED, NO_NETWORKS  or any status value returned from the MLME-ASSOCIATE.confirm primitive or the MLME-SCAN.confirm primitive	The status of the corresponding request

### 3.3.7.3.2 产生

网络层管理实体接收到 NLME-JOIN.request 时，对其 NLME 进行初始化，并生成该原语，发送给其上层作为对网络连接请求原语的响应。如果连接请求成功，则状态参数为 SUCCESS，否则，状态参数为错误代码。如 INVALID\_REQUEST、NOT\_PERMITTED、NO\_NETWORKS 或者为 MLME-ASSOCIATE.confirm 和 MLME-SCAN.confirm 原语所返回的状态值。这些状态值的情况如 3.3.7.1.3 小节所述。

### 3.3.7.3.3 接收

正在初始化设备的上层接收到该原语后，就可得到各种连接方式请求的执行结果，连接方式为联合方式，直接连接方式或古典连接方式。

## 3.3.8 直接将设备同网络连接

该原语定义了 ZigBee 协调器或路由器上层利用直接请求的方式，将另一个设备同自身网络连接。

### 3.3.8.1 NLME-DIRECT-JOIN.request 原语

该原语给出了 ZigBee 协调器或路由器的上层如何请求直接把另一个设备连接到自己的网络中。3.3.8.1.1 服务原语的语法

该原语的语法如下：

NLME-DIRECT-JOIN.request	{ DeviceAddress, CapabilityInformation }
--------------------------	---

表 3.21 详细描述了 NLME-DIRECT-JOIN.request 原语的参数。

**Table 3.21 NLME-DIRECT-JOIN.request Parameters**

Name	Type	Valid Range	Description
DeviceAddress	64-bit IEEE address	Any 64-bit IEEE address	The IEEE address of the device to be directly joined
CapabilityInformation	Bitmap	See Table 3.18	The operating capabilities of the device being directly joined

### 3.3.8.1.2 产生

ZigBee 协调器或路由器生成该原语把新设备直接连接到自己的网络。这个过程不需要任何传输。

### 3.3.8.1.3 接收

网络层管理实体接收到此原语后，将会尝试把参数 DeviceAddress 所给定地址的设备连接到邻居表中，而参数 CapabilityInformation 设定了加入网络后设备的运行能力。在执行协议中，alternate PAN coordinator 位为 0。如果设备作为 ZigBee 路由器，那么其 device type 位为 1，如果为终端设备则为 0。如果设备的电源为交流电源，则 power source 位置为 1，否则为 0。如果设备在空闲期间，设备接收器打开，则 receiver on when idle 位置为 1，否则置为 0。如果设备具有安全操作能力，则 security capability 位置为 1，否则为 0。

如果网络层管理实体成功地把连接设备加入其邻居表，则发送状态参数为 SUCCESS 的 NLME-DIRECT-JOIN.confirm 原语。如果网络层管理实体发现所要加入的设备已在其邻接表中，则发送状态参数为 ALREADY\_PRESENT 的 NLME-DIRECT-JOIN.confirm 原语。如果网络层管理实体不能将新的设备加入到邻接表中，则发送状态参数为 NEIGHBOR\_TABLE\_FULL 的 NLME-DIRECT-JOIN.confirm 原语。

### 3.3.8.2 NLME-DIRECT-JOIN.confirm 原语

该原语向 ZigBee 协调器或路由器上层通告直接把一设备加入网络请求原语的执行结果。

#### 3.3.8.2.1 服务原语的语法

该原语的语法如下：

---

NLME-DIRECT-JOIN.confirm	{ DeviceAddress, Status }
--------------------------	------------------------------------

---

表 3.22 详细描述了 NLME-DIRECT-JOIN.confirm 原语的参数。

**Table 3.22 NLME-DIRECT-JOIN.confirm Parameters**

Name	Type	Valid Range	Description
DeviceAddress	64-bit IEEE address	Any 64-bit, IEEE address	The 64-bit IEEE address in the request to which this is a confirmation
Status	Status	SUCCESS, ALREADY_PRESENT, NEIGHBOR_TABLE_FULL	The status of the corresponding request

### 3.3.8.2.2 产生

在接收到NLME-DIRECT-JOIN.request原语后，网络层管理实体生成该原语，并向上层发送作为对请求原语的响应。如果请求成功，则参数表示连接成功，否则，状态参数为错误代码，即为ALREADY\_PRESENT或 NEIGHBOR\_TABLE\_FULL。这些状态值的理由如3.3.8.1.3小节所述。

### 3.3.8.2.3 接收

正在初始化设备的上层接收到该原语后，即可得到其直接把一设备加入网络的请求原语执行结果。

## 3.3.9 断开网络

本小节介绍了设备上层请求自身或其他设备同网络断开连接的原语，同时也介绍了当设备成功地同网络断开后，向 ZigBee 协调器上层报告时所采用的原语。

### 3.3.9.1 NLME-LEAVE.request 原语

设备上层利用该原语请求自身或者其他设备同网络断开连接。

#### 3.3.9.1.1 服务原语的语法

该原语的语法如下：

NLME-LEAVE.request	{ DeviceAddress, RemoveChildren, Rejoin }
--------------------	---

表 3.23 详细描述了 NLME-LEAVE.request 原语的参数。

**Table 3.23 NLME-LEAVE.request Parameters**

Name	Type	Valid Range	Description
DeviceAddress	Device address	Any 64-bit, IEEE address	The 64-bit IEEE address of the entity to be removed from the network or NULL if the device removes itself from the network
RemoveChildren	Boolean	TRUE or FALSE	This parameter has a value of TRUE if the device being asked to leave the network is also being asked to remove its child devices, if any. Otherwise it has a value of FALSE.
Rejoin	Boolean	TRUE or FALSE	This parameter has a value of a TRUE if the device being asked to leave from the current parent is requested to rejoin the network; Otherwise, the parameter has a value of FALSE

#### 3.3.9.1.2 产生

当设备上层需要同网络断开连接，或者 ZigBee 协调器或路由器上层准备将一个设备同网络断开连接时，生成该原语。

#### 3.3.9.1.3 接收

如果设备没有同网络连接，而设备的网络层管理实体接收到该原语，则网络层管理实体发送状态参数为 INVALID\_REQUEST 的 NLME-LEAVE.confirm 原语。当一个网络连接设备的网络层管理实体接收到该原语，并且其 DeviceAddress 参数为 NULL，RemoveChildren 参数为 FALSE 时，网络层管理实体将按照 3.7.1.8.1 小节所述将自身与网络断开连接。网络层管理

实体将清除路由表入口参数，并向 MAC 层发送 MLME-RESET.request 原语。如果网络层管理实体收到 MLME-RESET.confirm 原语其状态参数不为 SUCCESS, 时，网络层管理实体可能会选择重发复位请求。网络层管理实体也将把相对于父节点的邻居表入口的 relationship 域设置为 0x03，表明没有关系。如果接收到的 NLME-LEAVE.request 原语的 DeviceAddress 参数为 NULL, RemoveChildren 参数为 TRUE，那么网络层管理实体将试图如 3.7.1.8.3 小节所述，移除其子节点。

当 ZigBee 协调器或路由器接收到该原语，且原语的设备地址参数不为 NULL，则网络层管理实体将判断所指定设备是否存在于邻居表中，。如果所请求设备不存在于邻居表中，则网络层管理实体将发送状态值为 UNKNOWN\_DEVICE 的 NLME-LEAVE.confirm 原语。如果所请求的设备存在于邻居表中，网络层管理实体将按照 3.7.1.8.3 小节所述将设备从网络移除。如果 RemoveChildren 参数为 TRUE，将请求移除该设备的子节点。移除结束，网络层管理实体将发送 NLME-LEAVE.confirm 原语，其参数 DeviceAddress 为移除设备的 64 位 IEEE 地址，状态参数为 MCPS-DATA.confirm 原语所返回的状态值。然后对应于移除设备的邻居表的 relationship 域将被更新。Relationship 域按照 NLME-LEAVE.request 原语的 Rejoin 参数进行更新。如果 Rejoin 域的为 FALSE，那么 relationship 域为 0x03，表明没有关系。如果 Rejoin 域为 TRUE，那么 relationship 域为 0x04，表明节点属于上一层的子节点。

3.3.9.2 NLLME-LEAVE.indication 原语

3.3.9.2.1 服务原语的语法

该原语的语法如下：

NLME-LEAVE.indication	{ DeviceAddress, Rejoin }
-----------------------	------------------------------------

表 3.24 详细描述了 NLLME-LEAVE.indication 原语的参数。

Table 3.24 NLME-LEAVE.indication Parameters

Name	Type	Valid Range	Description
DeviceAddress	64-bit IEEE address	Any 64-bit, IEEE address	The 64-bit IEEE address of an entity that has removed itself from the network or NULL in the case that the device issuing the primitive has been removed from the network by its parent
Rejoin	Boolean	TRUE or FALSE	This parameter has a value of TRUE if the device being asked to disassociate from the current parent is requested to rejoin the network; Otherwise, this parameter has a value of FALSE

3.3.9.2.2 产生

当与 ZigBee 协调器或路由器所连接的设备同网络断开时，协调器或路由器的网络层管理实体生层该原语，并且发送到 ZigBee 协调器或路由器的上层。该原语也可由 ZigBee 路由器或终端设备的网络层管理实体生层，并发送给设备上层以表明该设备已同该设备所连接的 ZigBee 协调器或路由器成功地断开连接。

3.3.9.2.3 接收

ZigBee 协调器或路由器上层一旦收到该原语，就可得到与其连接的设备已离开网络的消息。ZigBee 路由器或终端设备上层也由该原语可得到它与所连接的 ZigBee 协调器或路由器断开的通告消息。

如果参数 Rejoin 值为 TRUE, 那么上层期望按照 3. 7. 1. 3 小节所述的 NLME-JOIN.request 原语重新与网络连接。如果参数 Rejoin 值为 FALSE，离开的设备将不能自动的与网络重新连接，尽管可能在上层的指导下与网络重新连接。

3. 3. 9. 3 NLME-LEAVE.confirm 原语

该原语向一个设备上层通告请求设备自身或其他设备离开连接网络的结果。

3. 3. 9. 3. 1 服务原语的语法

该原语的语法如下：

NLME-LEAVE.confirm	{ DeviceAddress, Status }
--------------------	------------------------------------

表 3. 25 详细描述了 NLME-LEAVE.confirm 原语的参数

Table 3.25 NLME-LEAVE.confirm Parameters

Name	Type	Valid Range	Description
DeviceAddress	64-bit IEEE address	Any 64-bit, IEEE address	The 64-bit IEEE address in the request to which this is a confirmation or null if the device requested to remove itself from the network
Status	Status	SUCCESS, INVALID_REQUEST, UNKNOWN_DEVICE or any status returned by the MCPS-DATA.confirm primitive	The status of the corresponding request

3. 3. 9. 3. 2 产生

该原语向一个设备上层通告请求设备自身或者其他设备离开连接网络的结果。如果请求断开连接原语成功执行，则该状态参数值表明为成功地断开连接；否则，状态参数为 INVALID\_REQUEST或UNKNOWN\_DEVICE或为MCPS-DATA.confirm原语所返回的任意状态值。这些状态值的原因如3. 3. 9. 1. 3小节所述。

3. 3. 9. 3. 3 接收

正在初始化中的设备上层收到此原语，就可得到请求自身或其他设备同网络断开的执行结果。

3. 3. 10 重新复位设备

该原语介绍了一个设备应用层如何请求重新复位它的网络层。

3. 3. 10. 1 NLME-RESET.request 原语

设备应用层采用该原语请求网络层执行重新复位操作。

3. 3. 10. 1. 1 服务原语的语法

该原语的语法如下：

NLME-RESET.request	{ }
--------------------	--------

该原语无参数。

3.3.10.1.2 产生

该原语由设备应用层生成，并且发送到该设备的网络层管理实体，用来请求网络层重新复位网络层，以使它恢复到初始状态。

3.3.10.1.3 接收

网络层管理实体一旦收到该原语，就行 MAC 层发送 SetDefaultPIB 参数置为 TRUE 的 MLME-RESET.request 原语。网络层一旦收到所对应的 MLME-RESET.confirm 原语，将清除设备所有的内部变量和路由表入口参数，并将所有的 NIB 属性设为默认值，从而重新复位网络层。在网络层重新复位后，网络层管理实体将发出 MLME-RESET.confirm 原语；并且当 MAC 层成功地重新复位时，原语的状态参数设置为 SUCCESS，否则状态参数设置为 DISABLE\_TRX\_FAILURE。

如果此原语发送到一个已连接网络设备的网络层管理实体，任何使用 NLME-LEAVE.request 原语请求断开连接的企图都由上层进行优先判断。

3.3.10.2 NLME-RESET.confirm 原语

该原语用来向设备应用层报告请求重新复位网络层的执行结果。

3.3.10.2.1 服务原语的语法

该原语的语法如下：

NLME-RESET.confirm	{ Status }
--------------------	------------------

表 3.26 详细描述了 NLME-RESET.confirm 原语的参数

Table 3.26 NLME-RESET.confirm Parameters

Name	Type	Valid Range	Description
Status	Status	Any status value returned from the MLME-RESET.confirm primitive (see [B1])	The result of the reset operation

3.3.10.2.2 产生

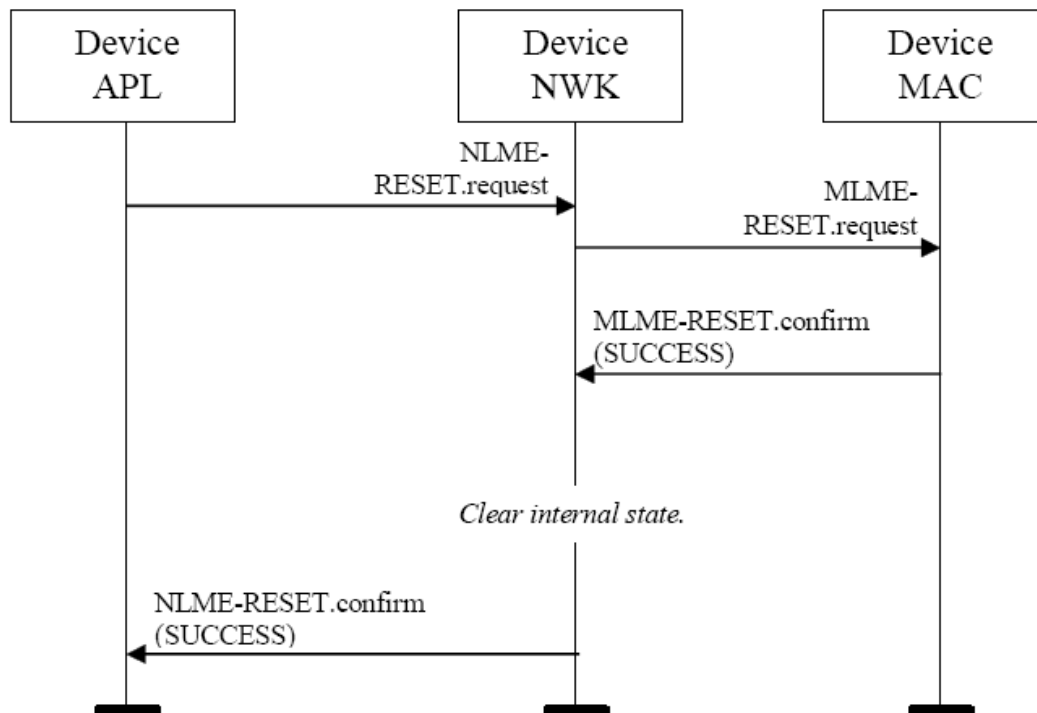
该原语由设备的网络层管理实体生成，并发送到它的上层用开对 NLME-RESET.request 原语进行确认。如果请求成功，则状态参数表明进行了一次成功的重新复位操作。否则，状态参数为 DISABLE\_TRX\_FAILURE 的错误代码。这些状态值的原因如 3.3.10.1.3 小节所述。

3.3.10.2.3 接收

在收到该原语后，该设备应用层就会得到它请求网络层重新复位的执行结果。

3.3.10.3 网络层重新复位的顺序图

图 3.2 描述了重新复位网络层所必须的顺序图



**Figure 3.2** Message Sequence Chart for Resetting the Network Layer

### 3.3.11 接收机同步

该原语介绍了一个设备应用层如何使它得接收机与 ZigBee 协调器或路由器同步，并从网络层中得到它的数据。

#### 3.3.11.1 NLME-SYNC.request 原语

设备应用层使用该原语与 ZigBee 协调器或路由器进行同步，或从 ZigBee 协调器或路由器中得到它的数据。

##### 3.3.11.1.1 服务原语的语法

该原语的语法如下：

NLME-SYNC.request	{
	}

##### 3.3.11.1.2 产生

无论何时，设备应用层要与 Zigbee 协调器或路由器实现同步，或查询在协调器或路由器中是否存在它的数据时，都可生成该原语。

##### 3.3.11.1.3 接收

接收到该原语，网络层管理实体将向 MAC 层发送 MLME-POLL.request 原语，并将它的参数 TrackBeacon 置为 FALSE。在收到相应的 MLME-POLL.confirm 原语后，网络层管理实体将发送 NLME-SYNC.confirm 原语，其状态参数与 MLME-POLL.confirm 原语的状态参数一致。

#### 3.3.11.2 NLME-SYNC.indication 原语

该原语向设备的应用层通告 MAC 层丢失网络同步信号。

##### 3.3.11.2.1 服务原语的语法

该原语的语法如下：



NLME-SYNC.indication	{
	}

该原语无参数。

### 3.3.11.2.2 产生

网络层管理实体通过 MLME-SYNC-LOSS.indication 原语从 MAC 层得到丢失同步信号通知后，就会生成该原语。该原语跟随着 NLME-SYNC.request 原语之后，发送到网络层管理实体。

### 3.3.11.2.3 接收

接收到该原语，其应用层就可得到设备的 MAC 层丢失了网络的同步信标。

### 3.3.11.3 NLME-SYNC.confirm 原语

该原语用来向设备应用层报告它所请求网络同步的执行结果，或报告请求从 ZigBee 协调器或路由器中所得到的数据的结果。

#### 3.3.11.3.1 服务原语的语法

该原语的语法如下：

NLME-SYNC.confirm	{
	Status
	}

表 3.27 详细描述了 NLME-SYNC.confirm 原语的参数。

**Table 3.27 NLME-SYNC.confirm Parameters**

Name	Type	Valid Range	Description
Status	Status	SUCCESS, SYNC_FAILURE, INVALID_PARAMETER or any status value returned from the MLME_POLL.confirm primitive (see [B1])	The result of the request to synchronize with the ZigBee coordinator or router

### 3.3.11.3.2 产生

该原语由正在初始化中的网络层管理实体生成，并发送到它的应用层用以对 NLME-SYNC.request 原语的确认。如果请求原语成功执行，状态参数表明为一次成功的状态改变尝试。否则，状态参数为错误代码。这些状态的原因如 3.3.11.1.3 小节所述。

### 3.3.11.3.3 接收

设备应用层收到此原语后，就可得到请求同步或请求从 ZigBee 协调器或路由器取得数据原语的执行结果。如果请求执行成功，则状态参数置为 SUCCESS；否则，状态参数为错误状态代码。

### 3.3.12 信息库维护

该原语介绍了设备上层如何读写网络信息库的属性

### 3.3.12.1 NLME-GET.request 原语

设备上层应用该原语请求读取网络信息库中某一属性值。

#### 3.3.12.1.1 服务原语的语法

该原语的语法如下：

NLME-GET.request	{ NIBAttribute }
------------------	------------------------

表 3.28 详细描述了 NLME-GET.request 原语的参数。

**Table 3.28 NLME-GET.request Parameters**

Name	Type	Valid Range	Description
NIBAttribute	Integer	See Table 3.42	The identifier of the NIB attribute to read

### 3.3.12.1.2 产生

该原语由设备网络层管理实体的上层生成,并发送给网络层管理实体以便从网络信息库中读取所指定的属性值。

### 3.3.12.1.3 接收

网络层管理实体一旦接收到该原语,就试图从它的数据库中获取所请求的属性值。如果在数据库中没有找到所指定的属性标识符,则发送状态为 UNSUPPORTED\_ATTRIBUTE 的 NLME-GET.confirm 原语。

如果网络层管理实体成功地获取了所请求的属性值,则发送状态参数为 SUCCESS 以及 NIB 属性标识符和属性值的 NLME-GET.confirm 原语。

### 3.3.12.2 NLME-GET.confirm 原语

该原语报告了从网络信息库中读取属性值的执行结果。

#### 3.3.12.2.1 服务原语的语法

该原语的语法如下:

NLME-GET.confirm	{ Status, NIBAttribute, NIBAttributeLength, NIBAttributeValue }
------------------	--

表 3.29 详细描述可该原语的参数。

**Table 3.29 NLME-GET.confirm Parameters**

Name	Type	Valid Range	Description
Status	Enumeration	SUCCESS or UNSUPPORTED_ATTRIBUTE	The results of the request to read a NIB attribute value
NIBAttribute	Integer	See Table 3.42	The identifier of the NIB attribute that was read.
NIBAttributeLength	Integer	0x0000 – 0xffff	The length, in octets, of the attribute value being returned.
NIBAttributeValue	Various	Attribute Specific (see Table 3.42)	The value of the NIB attribute that was read.

### 3.3.12.2.2 产生

该原语由网络层管理实体生成,并发送给它的上层,作为对 NLME-GET.request 原语的

确认。该原语返回的状态参数为 SUCCESS，则表明成功地读取了所请求的 NIB 属性值，或者为 UNSUPPORTED\_ATTRIBUTE 的错误代码。

3.3.12.2.3 接收

网络层管理实体上层一旦接收到该原语，就可得知读取 NIB 属性请求原语的执行结果。如果成功地执行了请求原语，则状态参数置为 SUCCESS。否则，状态参数为错误代码。

3.3.13.3 NLME-SET.request 原语

网络层管理实体上层使用该原语向网络信息库写入所指定的属性值。

3.3.13.3.1 服务原语的语法

该原语语法如下：

NLME-SET.request	{ NIBAttribute, NIBAttributeLength, NIBAttributeValue }
------------------	---

表 3.30 详细描述了该原语的参数。

Table 3.30 NLME-SET.request Parameters

Name	Type	Valid Range	Description
NIBAttribute	Integer	See Table 3.42	The identifier of the NIB attribute to be written
NIBAttributeLength	Integer	0x0000 – 0xffff	The length, in octets, of the attribute value being set
NIBAttributeValue	Various	Attribute Specific (see Table 3.42)	The value of the NIB attribute that should be written

3.3.12.3.2 产生

该原语由网络层管理实体上层生成，并发送给网络层管理实体，以此向网络信息库写入所给定的属性值。

3.3.12.3.3 接收

网络层管理实体一旦接收到该原语，就试图向它的数据库中写入所指定的属性值。如果所指定的属性参数在数据库中不存在，则网络层管理实体将发送状态参数为 UNSUPPORTED\_ATTRIBUTE 的 NLME-SET.confirm 原语。如果所指定的属性值超出了所给定属性的正常范围，则网络层管理实体将发送状态为 INVALID\_PARAMETER 的 NLME-SET.confirm 原语。

如果成功地写入了 NIB 属性，网络层管理实体将发送状态为 SUCCESS 的 NLME-SET.confirm 原语。

3.3.12.4 NLME-SET.confirm 原语

该原语报告了尝试向网络信息库中写入属性值的执行结果。

3.3.12.4.1 服务原语的语法

该原语的语法如下：

NLME-SET.confirm	{ Status, NIBAttribute }
------------------	-----------------------------------

表 3.31 详细描述了该原语的参数。

**Table 3.31 NLME-SET.confirm Parameters**

Name	Type	Valid Range	Description
Status	Enumeration	SUCCESS, INVALID_PARAMETER or UNSUPPORTED_ATTRIBUTE	The result of the request to write the NIB Attribute
NIBAttribute	Integer	See Table 3.42	The identifier of the NIB attribute that was written

3.3.12.4.2 产生

该原语由网络层管理实体生成，并向其上层发送，作为对 NLME-SET.request 原语的确认。如果该原语返回的状态为 SUCCESS，则表明所指定的属性值已经成功地写入所指定的 NIB 属性中，或者状态为 INVALID\_PARAMETER 或 UNSUPPORTED\_ATTRIBUTE。这些状态值的情况如 3.3.12.3.3 小节所述。

3.3.12.4.3 接收

网络层管理实体上层一旦接收到该原语，就可得知请求写入 NIB 属性值原语的执行结果。如果成功执行，则状态参数为 SUCCESS；否则，状态参数为出错代码。

3.3.13 路由错误报告

该原语用来描述设备网络层通知其上层发生路由失败，结果是至少一个单播或多播帧发送失败或通过该设备转发信息帧失败。广播帧的路由错误是帧发送到如表 3.23 所示的广播地址没有报告。

3.3.13.1 NLME-ROUTE-ERROR.indication 原语

该原语向设备上层通告网络通信失败。

3.3.13.1.1 服务原语的语法

该原语的语法如下：

NLME-ROUTE-ERROR.indication	{ ShortAddr, Status }
-----------------------------	--------------------------------

表 3.32 详细描述了该原语的参数。

**Table 3.32 NLME-ROUTE-ERROR.indication Parameters**

Name	Type	Valid Range	Description
ShortAddr	Integer	0x0000 – 0xFFF7	The 16-bit network address of the destination device associated with the routing failure
Status	Status	Any route error status code (see Table 3.40)	The error code associated with the routing failure

3.3.13.1.2 产生

当如下情况发生时，设备的网络层用该原语通知设备的上层：

- (1) 设备发现或重修 ShortAddr 参数所给地址的路由发生错误
- (2) 因为表 3.40 所给出的原因，设备向参数 ShortAddr 所给出的 16 位网络地址的终端子设备发送数据帧失败

- (3) 设备收到该设备的路由错误命令帧。在这种情况下，参数 ShortAddr 域将反映目的地址的值和命令帧的错误码域。

接收

设备上层通过该原语被通告与确定地址的通信失败。

3.3.14 路由发现

该原语用来定义设备上层如何初始化路由发现，如单播路由发现、多点传送路由发现和多对一路由发现，并被通知路由发现的结果信息。

3.3.14.1 NLME-ROUTE-DISCOVERY.request 原语

该原语允许设备上层初始化路由发现。

3.3.14.1.1 服务原语的语法

该原语的语法如下：

NLME-ROUTE-DISCOVERY.request	{ DstAddrMode, DstAddr, Radius MemoryConstrained }
------------------------------	---

表 3.33 详细描述了该原语的参数。

Table 3.33 NLME-ROUTE-DISCOVERY.request Parameters

Name	Type	Valid Range	Description
DstAddrMode	Integer	0x00 – 0x02	A parameter specifying the kind of destination address provided; The DstAddrMode parameter may take one of the following three values:  0x00 = No destination address 0x01 = 16-bit NWK address of a multicast group 0x02 = 16-bit NWK address of an individual device

**Table 3.33 NLME-ROUTE-DISCOVERY.request Parameters (Continued)**

Name	Type	Valid Range	Description
DstAddr	16-bit NWK Address	Any NWK address or multicast address	<p>The destination of the route discovery.</p> <p>If the DstAddrMode parameter has a value of 0x00 then no DstAddr will be supplied. This indicates that the route discovery will be a many-to-one discovery with the device issuing the discovery command as a target.</p> <p>If the DstAddrMode parameter has a value of 0x01, indicating multicast route discovery then the destination shall be the 16-bit NWK address of a multicast group.</p> <p>If the DstAddrMode parameter has a value of 0x02, this indicates unicast route discovery. The DstAddr will be the 16-bit NWK address of a device to be discovered.</p>
Radius	Integer	0x00 – 0xff	This optional parameter describes the number of hops that the route request will travel through the network.
MemoryConstrained	Boolean	TRUE or FALSE	Whether or not there is expected to be sufficient memory to store route record for each correspondent

#### 3.3.14.1.2 产生

该原语由 ZigBee 协调器或路由器的上层产生并发送给网络层管理实体请求初始化路由发现。

#### 3.3.14.1.3 接收

如果是 ZigBee 终端设备的网络层管理实体接收到该原语，那么网络层管理实体将向上层发送状态参数为 INVALID\_REQUEST 的 NLME-ROUTE-DISCOVERY.confirm 原语。

如果该原语的 DstAddrMode 参数不为 0x00，并且 DstAddr 参数不为广播地址，那么网络层管理实体将向上层发送状态值为 INVALID\_REQUEST 的 NLME-ROUTE-DISCOVERY.confirm 原语。

如果接收到该原语的 ZigBee 路由器或协调器没有路由能力，并且 DstAddrMode 参数为 0x01 或 0x02，那么网络层管理实体将向上层发送状态值为 NO\_ROUTING\_CAPACITY 的 NLME-ROUTE-DISCOVERY.confirm 原语。

如果接收到该原语的 ZigBee 路由器或协调器有路由能力，并且参数 DstAddrMode 的值为 0x02，网络层将开始发现从当前设备到参数 DstAddr 所示的 16 位网络地址设备的路由。初始化发现单播路由的详细描述见 3.7.3.5.1 小节。

如果接收到该原语的 ZigBee 路由器或协调器有路由能力，并且参数 DstAddrMode 的值为 0x01，网络层管理实体将通过查看在 nwkGroupIDTab1 入口中是否有与目的地址相对应的入口来检查该设备是否是 DstAddr 参数所确定的多点传输组标识符的一员。如果设备时多点传输组中的一员，则网络层管理实体将立即发送状态值为 SUCCESS 的 NLME-ROUTE-DISCOVERY.confirm 原语，终止 NLME-ROUTEDISCOVERY.request 原语的后续处理。如果设备不是多点传输组的一员，网络层管理实体将初始化一个从当前设备到由 DstAddr 参数确定的多点传输组的单播路由发现。初始化发现单播路由的详细描述见 3.7.3.5.1 小节。

如果接收到该原语的 ZigBee 路由器或协调器的 DstAddrMode 参数为 0x00，网络层管理

实体将初始化多对一路由发现。初始化多对一路由发现的过程详见 3.7.3.5.1 小节。

在任何一个上述的三中路由发现情况下，网络层管理实体将使用 MAC 层的 MCPS-DATA.request 原语试图传输一个路由发现命令帧来初始化路由发现。如果提供了可选参数 Radius，那么该值将出现在输出帧的网络层帧头的 radius 域；如果没有提供该值，那么如果要传输数据帧，网络层帧头的 radius 域将设置为网络层信息库中 nwkMaxDepth 参数的二倍。如果 MAC 层因为某些原因传输路由请求命令帧失败，那么网络层管理实体将向上层发送 ROUTE-DISCOVERY.confirm 原语，其状态参数与 MCPS-DATA.confirm 原语返回的状态值一致。如果路由发现命令帧发送成功，并且参数 DstAddrMode 值为 0x00，表明为多对一路由发现，网络层管理实体将立即发送状态值为 SUCCESS 的 ROUTE-DISCOVERY.confirm 原语。否则，网络层管理实体将等待直到接收到路由响应命令帧或路由发现操作超时，如 3.7.3.5 小节所示。如果在路由发现操作时间结束前接收到路由响应命令帧，网络层管理实体将向上层发送状态值为 SUCCESS 的 NLME-ROUTE-DISCOVERY.confirm 原语。如果操作时间超时，将发送状态值为 ROUTE\_DISCOVERY\_FAILED 的 NLME-ROUTE-DISCOVERY.confirm 原语。

3.3.14.2 NLME-ROUTE-DISCOVERY.confirm 原语

该原语用来向设备上层报告初始化路由发现操作的结果。

3.3.14.2.1 服务原语的语法

该原语的语法如下：

NLME-ROUTE-DISCOVERY.confirm	{ Status }
------------------------------	------------------

表 3.34 详细描述了 NLME-ROUTE-DISCOVERY.confirm 原语的参数。

Table 3.34 NLME-ROUTE-DISCOVERY.confirm Parameters

Name	Type	Valid Range	Description
Status	Status Value	INVALID_REQUEST, NO_ROUTING_CAPACITY, ROUTE_DISCOVERY_FAILED or any status value returned by the MCPS-DATA.confirm primitive	The status of an attempt to initiate route discovery

3.3.14.2.2 产生

该原语由网络层管理实体产生，并作为初始化路由发现的结果发送给设备上层。

3.3.14.2.3 接收

设备上层通过该原语得知初始化路由发现的执行结果。可能的状态参数和它们在何种情况下产生如 3.3.14.1.3 小节所述。

3.3.15 网络回退

该原语用来定义设备上层如何在改变信道前初始化网络回退。

3.3.15.1 NLME-START-BACKOFF.request 原语

设备上层利用该原语用来请求初始化网络回退。

3.3.15.1.1 服务原语的语法

该原语的语法如下：





网络层帧报头	有效载荷
--------	------

图 3.3 通用网络层帧格式

#### 3.4.1.1 帧控制域

帧控制域为 16 位，包含所定义的帧类型、地址和序列域以及其他控制标记。帧控制域格式如图 3.4 所示。

比特 0-1	2-5	6-7	8	9	10	11	12	13-15
帧类型	协议版本	发现路由	多播标记	安全	源路由	目的 IEEE 地址	源 IEEE 地址	保留

图 3.4 帧控制域格式

##### 3.4.1.1.1 帧类型子域

帧类型子域为 2bit，其值为表 3.36 中所列的非保留值。

表 3.36 帧类型子域值

帧类型值 b1 b0	帧类型名
00	数据
01	网络层命令
10, 11	保留

##### 3.4.1.1.2 协议版本子域

协议版本子域为 4bit，设置值反应了所使用的 ZigBee 网络层协议版本号特定设备上所使用的协议版本应像固定网络层协议版本号一样。

##### 3.4.1.1.3 发现路由子域

发现路由子域用根据帧的传送控制路由发现操作。（见 3.7.3.5）

对于网络层命令帧，路由发现子域设置为 0x00 表明抑制路由发现。

表 3.37 发现路由子域值

发现路由子域值	域意义
0x00	抑制路由发现
0x01	使能路由发现
0x02	强制路由发现
0x03	保留

##### 3.4.1.1.4 多播标志域

多播标志域为 1bit，如果是单播或者广播帧，值为 0，如果为多播帧值为 1。

##### 3.4.1.1.5 安全子域

安全子域值为 1 时，该帧才具有网络层安全操作能力。如果该帧的安全由另一层来完成或者完成被禁止，则该值是 0。

##### 3.4.1.1.6 源路由子域

源路由子域值为 1 时，源路由子帧才在网络报头中存在。如果源路由子帧不存在则源路由子域值为 0。

##### 3.4.1.1.7 目的 IEEE 地址子域

目的 IEEE 地址是 1 时，网络帧报头包含整个目的 IEEE 地址。

##### 3.4.1.1.8 源 IEEE 地址子域

源 IEEE 地址是 1 时，网络帧报头包含整个源 IEEE 地址。

#### 3.4.1.2 目的地址域

在网络层帧中必须有目的地址域，其长度是2字节。如果帧控制域的多播标志子域值是0，那么目的地址域值是16位的目的设备网络地址或者为广播地址（见表）。如果多播标志子域

值是1，目的地址域是16位目的多播组的Group ID。值得注意的是设备的网络地址与IEEE802.15.4-2003协议中的MAC层16位短地址相同。

3.4.1.3源地址域

在网络层帧中必须有源地址域，其长度是2字节，其值是源设备的网络地址。值得注意的是设备的网络地址与在IEEE802.15.4-2003协议中的MAC层16位短地址相同。

3.4.1.4半径域

在网络层帧中必须有半径域，其长度是1字节，并且限定了传输半径范围。每个设备接收一次该帧，则该值减以。

3.4.1.5序列号域

在每个帧中都包含序列号域，其长度是1字节。每发送一个新的帧序列号值加1。帧的源地址和序列号子域是一对，在限定了序列号1字节的长度内是唯一的标识符。关于使用序列号的更多信息，见3.7.2节。

3.4.1.6目的IEEE地址域

如果存在目的IEEE地址域，则包含与包含在网络层地址头中的目的地址域的16位网络地址相对应的64位IEEE地址。如果该16位网络地址是广播或者多播地址那么目的IEEE地址不存在。

3.4.1.7源IEEE地址

如果存在源IEEE地址域，则包含与包含在网络层地址头中的源地址域的16位网络地址相对应的64位IEEE地址。

3.4.1.8多播控制域

多播控制域是1字节长度且只有多播标志子域值是1时存在。它分成3个子域如图3.5所示。

比特： 0-1	2-4	5-7
多播模式	非成员半径	最低非成员半径

图3.5多播控制域帧格式

3.4.1.8.1多播模式子域

多播模式子域表明无论是使用成员或非成员模式传输该帧。成员模式在目的组成员设备中使用传送多播帧。非成员模式是从不是多播组成员设备到是多播组成员设备换算多播帧。

表3.38多播模式子域值

多播模式域值b0b1	域意义
00	非成员模式
01	成员模式
10	保留
11	保留

3.4.1.8.2非成员半径子域

当不是目的组成员设备转播时，非成员半径域表明成员模式多播范围。接收设备是目的组成员将设置该子域值是最大非成员半径（MaxNonmemberRadius）域的值。如果NonmemberRadius field的值是0，接收设备不是目的组成员时将丢弃该帧，且如果NonmemberRadius域的值是在0x01到0x06范围内，那么将耗尽此域。如果NonmemberRadius域值是0x07表明无限的范围且不能被耗尽。

3.4.1.8.3最大非成员半径（MaxNonmemberRadius）子域

该帧的非成员半径域的最大值。

3.4.1.9源路由子帧域

如果帧控制域的源路由子域的值是1，才存在源路由子帧域。它分成三个子域如图3.6所示。

字节：1	1	可变
应答 计数器	应答索引	应答列表

图3.6源路由子帧格式

3.4.1.9.1应答计数器子域

应答计数器子域表明包含在源路由子帧转发列表里的应答的数值。

3.4.1.9.2转发索引

应答索引子域表明传输的数据包的应答列表子域的下一转发的索引。这个域被数据包的发送设备初始化为0，且每转发一次就加1。

3.4.1.9 应答列表子域

应答列表子域是节点的2字节短地址的列表，这个域用来为源路由数据包的目的转发。地址是最无意义字节格式（formatted least significant byte first, ???）且在源路由中有顺序的出现。

3.4.1.10帧有效载荷域

帧有效载荷的长度是可变的，包含了各种帧类型的具体信息。

3.4.2各种帧类型的格式

定义了两类类型的网络层帧，它们分别是数据帧和网络层命令帧。在下面将对这两种帧类型进行讨论。

3.4.2.1数据帧格式

数据帧格式如图3.7所示。

字节：2	可变长	可变长
帧控制	路由域	数据载荷
网络层帧头		网络层载荷

图3.7数据帧格式

数据帧各部分的顺序与图3.3所示的通用网络层帧格式的顺序相同。

3.4.2.1.1数据帧网络层报头域

数据帧的网络层报头域有控制域和根据需要适当组合而得到的路由域组成。

如表3.36所示，在帧控制域中，帧类型子域应表示数据帧的值。根据数据帧的用途，对其他所有的子域进行设置。

根据帧控制域中的设置（参见图3.4），路由为地址域和广播域经过适当组合得到的。

3.4.2.1.2数据的有效载荷域

数据帧的数据有效载荷域包含字节的序列，该序列为网络层上层要求网络层传送的数据。

3.4.2.2网络层命令帧格式

网络层命令帧格式如图3.8所示。

字节：2	可变长	1	可变长
帧控制	路由域	网络层命令标识符	网络层命令载荷
网络层帧报头		网络层载荷	

图3.8网络层命令帧格式

网络层命令帧各部分的顺序与图3.3所示的通用网络层帧格式的顺序相同。

3.4.2.2.1网络层命令帧中的网络层帧报头域

网络层命令帧中的网络层帧报头域由帧控制域和根据需要适当组合得到的路由域组成。

如表3.36所示，在帧控制域中，帧类型子域应表示网络层命令帧的值。根据网络层命令

帧的用途，对其他所有的子域进行设置。

根据帧控制域中的设置，路由为地址域和广播域经过适当组合得到的。

3.4.2.2网络成命令标识符

网络层命令标识符域表明所使用的网络层命令，其值如表3.39所列的非保留值之一。

3.4.2.2.3网络层命令的有效载荷域

网络层命令帧的网络层命令载荷域包含网络层命令本身。

3.5命令帧

网络层定义的命令帧，如表3.39所示。本小节详细介绍网络层管理实体如何构造要传递的各种命令。

表3.39网络层命令帧

命令帧标识符	命令名称	
0x01	路由请求	3.5.1节
0x02	路由应答	3.5.2节
0x03	路由错误	3.5.3节
0x04	断开	3.5.4节
0x05	路由记录	3.5.5节
0x06	重新连接请求	3.5.6节
0x07	重新加入响应	3.5.7节
0x08	连接状态	3.5.8节
0x09	网络报告	3.5.9节
0x0A	网络更新	3.5.10节
0x0B- 0xFF	保留	——

3.5.1路由请求命令

设备使用路由请求命令来请求在其无线通信范围内的其他设备发现到达目的设备的路由，以便在网络中建立一条稳定的使信息更快更经济地到达目的设备的路由。路由请求命令的载荷格式如图3.9所示。

字节：1	1	1	2	1
命令帧标识符（见表3.39）	命令选择	路由请求标识	目的地址	路由开销
网络层载荷				

图3.9路由请求命令帧格式

3.5.1.1MAC数据服务请求

为了利用MAC层数据服务（在IEEE 802.15.4-2003【B1】）来传输这个命令，在MAC层帧报头包含如下信息：

- （1）目的PAN标识符设置为发送路由请求命令设备的PAN标识符。
- （2）目的地址必须为广播地址0xffff。
- （3）源MAC地址和PAN标识符设置为发送路由请求命令设备的地址和PAN标识符，该设备不一定是命令源发送设备。
- （4）因为任何来自于网络层的可靠帧都使用网络层的安全协议，帧控制域将禁止MAC层对MAC层数据帧使用安全功能。由于该帧为广播帧，因此不需要确认。地址模式以及内部PAN标记设置为支持在这里所描述的地址域。

3.5.1.2网络层帧报头域

为了传送路由请求命令帧到它的目的设备，且为了路由发现过程正确完成，应提供如下信息：

- （1）网络层帧头的源地址域设置成发送设备的地址。

(2) 网络层帧头的目的地址域设置成设备的广播地址，*macRxOnWhenIdle*的值等于TRUE（参见表3.51）

(3) 网络层帧头的发现路由子域设置成抑制路由发现（参见表3.36）

(4) 作为一个网络层命令帧，帧控制域的源IEEE地址子域设置成1，且网络层帧头的源IEEE地址域存在且包含帧发送此帧设备的64位IEEE地址。如果试图发现的设备的64位IEEE地址已经知道，那么帧控制域的目的IEEE地址子域的值是1，且网络层帧头的目的IEEE地址域存在且包含设备的64位IEEE地址。

3.5.1.3网络层帧有效载荷域

网络层帧的载荷载包含命令标识符域、命令选择域、路由请求标识符域、目的地址和最新的路由总开销域。

命令帧标识符应包含表明路由请求命令帧的值。

3.5.1.3.1命令选择域

8位的命令选择域格式如图3.10所示。

比特：0-4	5	6	7
保留	多播	保留	

图3.10路由请求命令选择域

3.5.1.3.1.1多播子域

多播子域是1位。只有命令帧请求多播组路由时，它的值是1，在这个情况下，目的地址域包含期望组的Group ID。

3.5.1.3.2路由请求标识符

路由请求标识符为一个8bit的路由请求序列号，在特定设备的网络层每发送一次路由请求，该标识符增加1。

3.5.1.3.3目的地址

目的地址长2字节，标识路由请求命令帧的目的地址。

3.5.1.3.4路由开销

路由开销域长度为8bit，常常用来积累路由请求命令帧在网络中传送的开销信息。（参见3.7.3.5.2节）

3.5.2路由应答命令

路由应答命令的目的设备使用路由应答命令来通知路由请求的源设备已接收到请求命令。ZigBee路由请求所经路由器建立一种能使帧更快捷地从源地址路由到目的地址的状态路由应答命令的载荷格式如图3.11所示。

字节：1	1	1	2	2	1
命令标识符（参见表3.39）	命令选择	路由请求标识符	源地址	响应地址	路由开销
网络层载荷					

图3.11路由应答命令帧格式

3.5.2.1MAC层数据服务请求

根据802.15.4协议标准，为了利用MAC层数据服务来传输该命令，在MAC层帧报头中应包含一下信息。

(1) 目的MAC层地址和PAN标识符分别设置为路由请求命令帧的发起端路由中第一跳的网络地址和PAN标识符。目的PAN标识符必须与命令起始端的PAN标识符相同。

(2) 源MAC地址和PAN标识符设置为发送路由应答命令的设备的地址和PAN标识符，该设备不一定为源命令发送的设备。

(3) 帧控制域设置为禁止MAC数据帧使用MAC层安全功能，因此任何来自于

网络层的可靠的帧都使用网络层的安全协议。传输选择应设置为请求命令确认。地址模式以及内部PAN标记设置为支持这里所描述的地址域。

3.5.2.2网络层帧报头域

为了传送路由应答到目的地，并且使路由发现进程正确无误，必须提供下列信息：

- (1) 网络层帧控制域中的帧类型子域应设置为表明此帧为网络层命令帧。
- (2) 在网络层帧报头中的目的地址域应设置为回到相应路由请求的发起端路由的第一跳的网络地址。
- (3) 网络层帧报头中的源地址应设置为传送此帧设备的网络层的16位网络地址。
- (4) 网络层帧报头中的发现路由子域设置成抑制路由发现（参见表3.36）
- (5) 作为一个网络层命令帧，帧控制域的源IEEE地址子域设置成1，且网络层帧头的源IEEE地址域存在且包含帧发送此帧设备的64位IEEE地址。如果试图发现的设备的64位IEEE地址已经知道，那么帧控制域的目的IEEE地址子域的值是1，且网络层帧头的目的IEEE地址域存在且包含所应答的路由请求命令帧的发送端的64位IEEE地址。

3.5.2.3网络层有效载荷域

网络层有效载荷域包含命令标识符、命令选择域、路由请求标识符域、源地址和应答地址和最新的路由开销总和。

命令标识符域包含表明路由应答命令帧的值。

3.5.2.3.1命令选择域

8bit的命令选择域的格式如图3.12所示。

比特： 0-5	6	7
保留	多播域	保留

图3.12路由应答命令选择域格式

3.5.2.3.1.1多播子域

多播子域是1位。只有命令帧应答多播组路由时，它的值是1，在这个情况下，响应者地址域包含期望组的Group ID。

3.5.2.3.2路由请求标识符

路由请求标识符长度是8位的所应答的路由请求帧序列号。

3.5.2.3.3源地址

源地址长度为2字节，包含所应答的路由请求命令帧发起端的16位的网络地址。

3.5.2.3.4响应地址

响应地址长度为2字节，且与相应的路由请求命令帧的目的地址域的值相同。

3.5.2.3.5路由成本

路由成本用来收集当路由应答命令帧穿梭于网络时链路成本。

3.5.3路由错误命令

当设备无法向前传送数据时，便使用路由错误命令。该命令通知发送数据帧源设备，在传送数据帧时出现错误。路由错误命令的载荷格式如图3.13所示。

字节： 1	1	2
命令帧标识符（参见表3.39）	错误代码	目的地址

图3.13路由错误命令帧格式

3.5.3.1MAC层数据服务要求

为了利用MAC层数据服务来传输该命令，根据802.15.4协议标准，应提供下列信息。

- (1) 目的MAC层地址和PAN标识符应分别设置为出现传送故障的数据帧的发



起端所经路由中第一跳的地址和PAN标识符。

(2) 源MAC层地址和PAN标识符应设置为发送路由错误命令的设备地址和PAN标识符。

(3) 帧控制域应设为使MAC数据帧禁止使用MAC安全功能，因此任何来自于网络层的可靠的帧都使用网络层的安全协议。是否执行发送该路由错误命令须取决于是否需要确认。

(4) 地址模式和内部PAN标记应设置为这里所描述的支持地址域。

3.5.3.2网络层帧报头域

为了传送路由错误命令帧，网络层帧报头中的地址域应为与出现传送错误数据帧的发起端地址相同。

网络层帧报头中的源地址应设置为发送路由错误命令的设备地址。

网络层帧报头中的发现路由子域应设置为抑制路由发现（参见表3.36）

3.5.3.3网络层载荷域

路由错误命令帧的网络层帧载荷域包含如下描述的命令标识符域、错误代码域和目的地址域。命令帧标识符域设置为为了描述如表3.39所定义的路由错误代码命令帧。

3.5.3.3.1错误代码

错误代码为表3.40所示的非保留值之一。

表3.40路由错误命令帧的错误代码

值	错误代码
0x00	无有效路由
0x01	树状态链路失败
0x02	非树状态链路失败
0x03	低电池电压
0x04	无路由能力
0x05	无间接能力
0x06	间接传送终止
0x07	目的设备没有获得
0x08	目的地址没有获得
0x09	父设备链路失败
0x0a	有效路由
0x0b	源路由失败
0x0c	多对一路由失败
0x0d	地址冲突
0x0e	校验地址
0x0f	PAN标识符更新
0x10-0xff	保留

这些错误代码是表示错误代码命令帧的错误代码域的值和NLME-ROUTEERROR.indication原语的状态参数值。简单的解释如下：

- (1) 无有效路由：路由发现和/或已经尝试修复和到目的地址没有发现路由。
- (2) 树状态链路失败：帧沿树进行路由失败时，路由失败发生。
- (3) 非树状态链路失败：尝试沿树路由的结果没有失败。
- (4) 低电池电压：因为应答设备工作在低电压状态下，所以帧没有应答。
- (5) 无路由能力：因为应答设备没有路由能力所以失败发生。

- (6) 无间接能力: 休眠终端子设备缓存该帧和应答设备没有足够的缓冲能力, 所以失败发生。
- (7) 间接传送终止: 该帧代表休眠子设备time-out的结果缓存。
- (8) 目的设备没有获得: 响应设备的终端子设备由于一些原因没有获得。
- (9) 目的地址没有获得: 该帧是响应设备的终端子设备不存在的地址。
- (10) 父设备链路失败: RF链路到设备的父设备失败。
- (11) 有效路由: 在目的地址域中多播路由标识符有效。
- (12) 源路由失败: 源路由失败可能表明源路由链路中的一个链路失败。
- (13) 多对一路由失败: 多对一的路由请求失败。
- (14) 地址冲突: 目的地址域的地址被两个或者更多设备使用。
- (15) 校验地址: 源地址在源IEEE地址域里有IEEE地址, 且如果目的IEEE地址域存在, 它包含的目的的期望的IEEE地址。
- (16) PAN标识符更新: 设备的操作网络PAN标识符已经更新。

### 3.5.3.3.2目的地址

目的地址长度为2字节, 包含出现传送错误数据帧的目的地址。

### 3.5.4断开命令

网络层管理实体用断开命令通知网络中的其他设备设备正在离开网络或者请求一个设备离开网络。断开命令帧的载荷格式如图3.14所示。

字节: 1	1
命令帧标识符 (参见表3.39)	命令选择

图3.14断开命令帧格式

### 3.5.4.1MAC数据服务请求

为了利用MAC层数据服务来传输该命令, 根据802.15.4协议标准, 应提供下列信息。

- (1) 目的MAC层地址和PAN标识符应分别设置为该帧要发送到的邻居设备的地址和PAN标识符。
- (2) 源MAC层地址和PAN标识符应设置为发送断开命令的设备地址和PAN标识符。
- (3) 帧控制域应设为使MAC数据帧禁止使用MAC安全功能, 因此任何来自于网络层的可靠的帧都使用网络层的安全协议。请求被应答。
- (4) 地址模式和内部PAN标记应设置为这里所描述的支持地址域。

### 3.5.4.3网络层帧报头域

为了发送网络层断开命令帧, 如果请求子域设置成1, 那么在网络层帧头中的目的地址子域设置成请求断开子设备的网络地址, 帧控制域中的目的IEEE地址设置成请求离开的设备的IEEE地址。如果请求子域设置成0, 那么网络层帧报头中的目的地址域设置成0xffffd以表明被带有*macRxOnWhenIdle*的值等于TRUE的所有设备接收, 帧控制域的源IEEE地址子域设置成1, 且源IEEE地址应设置为断开网络设备的IEEE地址。网络层帧头的半径域设置成1。

网络层帧报头中的发现路由子域应设置为抑制路由发现 (参见表3.36)

### 3.5.4.3网络层载荷域

断开命令帧的网络载荷域包含一个命令帧标识符和一个命令选择域。命令帧标识符域与表3.39描述的断开命令帧一样。

### 3.5.4.3.1命令选择域

8bit的命令选择域的格式如图3.15所示。

比特: 0-4	5	6	7
保留	重新连接	请求	断开子设备

图3.15断开命令帧选择子域

3.5.4.3.1.1重新连接子域

重新连列子域是1bit在比特5的位置上。如果这个子域的值是1，同它目前父设备断开的设备重现连接到网络。如果该子域值是0，设备将不重新连接网络。

3.5.4.3.1.2请求子域

请求子域长度是1bit在bit6位置上。如果该子域的值是1，那么断开命令帧请求另一个设备离开网络。如果该子域值是0，那么断开命令帧表明发送设备准备断开网络。

3.5.4.3.1.3断开子设备子域

断开子设备子域是1bit长度在bit7的位置。如果该子域的值是1，那么断开设备的子设备也断开网络。

3.5.5路由记录命令

路由记录命令允许穿梭于网络中的单播数据包在命令载荷中记录路由，且传送到目的地。路由记录命令的载荷格式如图3.16所示。

字节：1	1	可变长
命令帧标识符	应答计数器	应答列表
网络层载荷		

图3.16路由记录命令帧格式

3.5.5.1MAC层数据服务请求

- 为了利用MAC层数据服务来传输该命令，根据802.15.4协议标准，应提供下列信息。
- (1) 目的MAC层地址和PAN标识符应分别设置为该帧要发送到的邻居设备的地址和PAN标识符。
  - (2) 源MAC层地址和PAN标识符应设置为发送路由记录命令的设备地址和PAN标识符。
  - (3) 帧控制域应设为使MAC数据帧禁止使用MAC安全功能，因此任何来自于网络层的可靠的帧都使用网络层的安全协议。请求被应答。
  - (4) 地址模式和内部PAN标记应设置为这里所描述的支持地址域。

3.5.5.2网络层帧报头域

网络层帧控制域的帧类型子域应社这为表示该帧是网络层命令帧。网络层帧头的源地址域和目的地址域分别设置成发起设备和目的设备的地址。帧控制域源路由子域应设置为0。

网络层帧头的发现路由子域应设置为抑制路由发现（参见表3.36）

3.5.5.3网络层有效载荷

网络层帧载荷包含命令标识符、应答计数器域和应答列表域。命令帧标识符域包含的值表明路由记录命令帧。

3.5.5.3.1应答计数器域

这是个长度1字节的域，包含路由记录命令的应答列表域的应答数。发起设备把它初始化为0，且每接收一个应答加1。

3.5.5.3.2应答列表域

应答列表域是应答数据包的节点的2字节的短地址的列表。地址是最少的有意义的格式。在发送一个数据包之前接收节点附加它们的短地址给列表。

3.5.6重新连接请求命令

重新连接请求命令允许设备重新连接它的网络。通常是响应通信失败才这么做，例如当终端设备不能同它的发起父设备通信。

字节：1	1
命令帧标识符（参见表3.39）	能力信息
网络层载荷	

图3.17重新请求命令帧格式

3.5.6.1MAC层数据服务请求

- 为了利用MAC层数据服务来传输该命令，根据802.15.4协议标准，应提供下列信息。
- (1) 目的MAC层地址和PAN标识符应分别设置为预期的父设备的地址和PAN标识符。
  - (2) 源MAC层地址和PAN标识符应设置为先前的短地址和传送重新请求命令帧的PAN标识符。
  - (3) 传送选择应设置为请求确认。
  - (4) 地址模式和内部PAN标记应设置为这里所描述的支持地址域。

3.5.6.2网络层帧报头域

当发送一个重新连接请求命令帧，网络层管理实体将设置网络层帧头的源地址域是先前的传送帧的设备的短地址，帧控制域的源IEEE地址子域将设置为1，且源IEEE地址域将设置为发送请求设备的IEEE地址。半径域设置为1。

网络层帧报头中的发现路由子域应设置为抑制路由发现（参见表3.36）

3.5.6.3网络层有效载荷域

网络层载荷域包含一个命令标识符域和一个能力信息域。命令帧标识符包含表明重新连接请求命令帧的值。

3.5.6.3.1能力信息域

是一个1字节的域，这个域包含在连接请求命令中的能力信息域的格式，在表3.18中有描述。

3.5.7重新连接响应命令

设备发送重新连接响应命令来通知它的短地址的子设备和重新连接状态。

字节：1	2	1
命令帧标识符（参见表3.39）	短地址	重新连接状态
网络层载荷		

图3.18重新连接响应命令帧格式

3.5.7.1MAC层数据服务请求

- 为了利用MAC层数据服务来传输该命令，根据802.15.4协议标准，应提供下列信息。
- (1) 目的MAC层地址和PAN标识符应分别设置为请求重新连接网络的设备的地址和PAN标识符。
  - (2) 源MAC层地址和PAN标识符应设置为接收和处理重新请求命令帧的网络地址和PAN标识符。
  - (3) 请求确认。
  - (4) 地址模式和内部PAN标记应设置为这里所描述的支持地址域。如果包含在重新请求命令的'Capability Information'字节的'Receiver on when idle'位的值等于0x00，那么TX操作将请求“间接传送”。反之使用'directtransmission'。

3.5.7.2网络层帧报头域

当发送一个重新连接响应命令帧，网络层管理实体将设置网络层帧头的目的地址域是先前的连接设备的短地址，帧控制域的源IEEE地址子域将设置为1，且网络层帧头的源IEEE地址域存在且包含发送响应的父设备的64位IEEE地址。帧控制域的目的IEEE地址这也是1，且网络层帧头的目的地址子域存在且包含响应重新连接请求命令帧的源的子设备的64位IEEE地址。

网络层帧报头中的发现路由子域应设置为抑制路由发现（参见表3.36）。

3.5.7.3网络层载荷域

当发送一个重新连接响应命令帧，网络层管理实体将设置网络层帧头的目的地址域是先前的重新连接设备的短地址。

3.5.7.3.1短地址域

如果重新连接成功，那么这个2字节域包含一个新的被指定的重新连接设备的短地址。如果重新连接没成功，这个域包含广播地址（0xffff）。

3.5.7.3.2重新连接状态域

这是1字节的域，包含在【B1】中规定的非保留连接状态值之一。

3.5.8连接状态命令

连接状态命令帧允许邻居路由器之间通信，直到它们彼此的输入链路成本如3.7.3.4描述。链路状态命令载荷格式如图3.19所示。

字节：1	1	可变长
命令帧标识符	命令选择	链路状态列表
网络层载荷		

图3.19链路状态命令帧格式

8bit的命令选择域的格式如题3.20所示。

比特：0-4	5	6	7
入口计数器	第一帧	最后帧	保留

图3.20链路状态命令选择域

在链路状态列表中的一个入口的格式如图3.21所示。

字节：2	1
邻居设备网络层地址	链路状态

图3.21链路状态入口

链路状态入口的链路状态域的格式如下：

比特：0-2	3	4-6	7
输入成本	保留	输出成本	保留

3.5.8.1MAC层数据服务请求

为了利用MAC层数据服务来传输该命令，根据802.15.4协议标准，应提供下列信息。

- （1）目的MAC层地址和PAN标识符应分别设置为发送链路状态命令设备的地址和PAN标识符。
- （2）目的地址必须设置成广播地址0xffff。
- （3）源MAC层地址和PAN标识符应设置为发送状态命令设备的地址和PAN标识符。
- （4）帧控制域应设为使MAC数据帧禁止使用MAC安全功能，因此任何来自于网络层的可靠的帧都使用网络层的安全协议。
- （5）地址模式和内部PAN标记应设置为这里所描述的支持地址域。

3.5.8.2网络层帧报头域

为了发送一个链路状态命令帧，网络层帧头的源地址域应设置为发送设备的地址。

网络层帧报头的目的地址设置成仅仅是路由器的广播地址。（参见表3.51）

网络层帧报头中的发现路由子域应设置为抑制路由发现（参见表3.36）。

3.5.8.3网络层有效载荷域

每一个链路状态入口白含路由器邻居设备的网络地址，最没有意义字节在链路状态字节之后。输入成本域包含设备为邻居设备估计的链路成本，其值在1到7之间。输出链路成本域包含邻居表的输出成本域的值。

链路状态入口按网络地址的上升顺序存储。如果所有的路由器邻居设备不适合一个单帧，多针发送。当发送多帧时，对于帧N的链路状态表中最后网络地址等于帧N+1的链路状态表的第一个网络地址。

命令选择域的入口计数器子域表明链路状态表中的目前链路状态入口的值。如果是发送

者的链路状态的第一帧那么第一帧子域值设置为1。如果是发送者的链路状态的最后帧那么最后帧子域值设置为1。如果发送这状态适合单帧，第一帧和最后帧位都设置为1。

链路状态帧作为一个没有重发的单跳广播传输。

3.5.9网络层报告命令

网络层报告命令允许设备报告网络事件给协调器。可以报告的事件是无限电通信信道条件和PAN ID冲突。网络层报告命令载荷格式如图3.22所示。

字节： 1	1	8	可变长
命令帧标识符（参见表3.39）	命令选择（参见图3.23）	EPID	记录信息
网络层载荷			

图3.23网络层记录命令帧载荷

3.5.9.1MAC层数据服务请求

为了利用MAC层数据服务来传输该命令，根据802.15.4协议标准，应提供下列信息。

（1）目的MAC层地址和PAN标识符应分别设置为发送网络记录命令设备的地址和PAN标识符。

（2）目的地址必须设置成NIB中的nwkManagerAddr

（3）源MAC层地址和PAN标识符应设置为发送网络记录命令设备的地址和PAN标识符，这个设备不一定是发出命令的设备

（4）帧控制域应设为使MAC数据帧禁止使用MAC安全功能，因此任何来自于网络层的可靠的帧都使用网络层的安全协议。传送选择需要设置请求确认。

3.5.9.2网络层帧报头域

为了是网络记录命令到达包含NIB中nwkManagerAddr参数的地址指定的设备，必须提供下列信息。

网络层帧控制域的帧类型子域设置成表明这个帧是网络层命令帧。

网络层帧报头域的目的地址域应设置成包含NIB中nwkManagerAddr参数的16位网络地址。

网络层帧报头域的源地址域应设置成发送此帧的设备的16位网络地址。

网络层帧报头中的发现路由子域应设置为抑制路由发现（参见表3.36）。

3.5.9.3网络层有效载荷域

网络层帧载荷包含一个命令标识符域、一个命令选择域、一个EPID和一个记录信息载荷。

命令标识符域包含表明一个网络记录命令帧的值。

3.5.9.3.1命令选择域

8bit的命令选择域的格式如图3.23所示。

比特： 0-4	5-7
记录信息计数器	记录命令标识符（参见图3.24）

图3.23网络层记录命令选择子域

3.5.9.3.1.1记录信息计数器子域

记录命令标识符子域包含一个目标表明记录信息命令的类型。图3.24包含能插入这个域的值。

命令标识符值	记录类型
0x00	PAN标识符出土
0x01	本地冲突记录
0x02-0x07	保留

图3.24 记录命令标识符子域



### 3.5.9.3.2 EPID子域

EPID域包含64位EPID，定义一个网络，记录设备是这个网络中的一个成员。

### 3.5.9.3.3 记录信息

记录信息域根据记录命令标识符子域的值提供正在记录的信息，域的格式。

### 3.5.9.3.3.1 PAN标识符冲突记录

如果记录命令标识符子域的值表示一个PAN标识符冲突记录，那么记录信息域由一个16位PAN标识符列表组成，这个标识符是在记录设备的邻居运行的。记录信息计数器将设置等于包含在记录信息域中的PAN标识符的数。

### 3.5.9.3.3.2 本地信息记录

如果记录命令标识符子域的值表示一个本地信息记录那么记录信息域是有能量扫描值列表（每个信道1字节）和所有邻居设备的Tx失败总和（2字节）组成。记录信息计数器包含正在记录的信道的数。

### 3.5.10 网络更新命令

网络更新命令允许由NIB中的nwkManagerAddr参数确定的设备广播配置信息的改变到网络中的所有设备。例如广播网络将改变短PAN标识符。

网络更新命令帧的载荷格式如图3.25所示。

字节：1	1	8	2
命令帧标识符	命令选择（参见图3.23）	EPID	更新信息
网络层载荷			

图3.23网络更新命令帧格式

### 3.5.10.1 MAC层数据服务请求

为了利用MAC层数据服务来传输该命令，根据802.15.4协议标准，MAC层帧报头包含如下信息。

- （1）为了使命令帧能到达没有接收到更新命令的网络设备，目的PAN标识符将设置成ZigBee协调器的老PAN标识符。目的地址必须设置成广播地址0xffff。
- （2）源MAC地址和PAN标识符设置成发送网络记录命令设备的网络地址和老的PAN标识符，这个设备不一定是发送这个命令的设备。
- （3）帧控制域应设为使MAC数据帧禁止使用MAC安全功能，因此任何来自于网络层的可靠的帧都使用网络层的安全协议。传送选择需要设置请求确认。

### 3.5.10.2 网络层帧报头域

为了发送一个网络更新命令帧，网络层帧头的源地址域应设置为发送设备的地址。

网络层帧报头的目的地址设置成广播地址0xFFFF。

网络层帧报头中的发现路由子域应设置为抑制路由发现（参见表3.36）。

### 3.5.10.3 网络层有效载荷域

网络层帧载荷包含一个命令标识符域、一个命令选择域、一个EPID和一个更新命令可变域。

命令标识符域包含表明一个网络更新命令帧的值。

### 3.5.10.3.1 命令选择域

8bit的命令选择域的格式如图3.26所示。

比特：0-4	5-7
更新信息计数器	更新命令标识符（参见图3.27）

图3.23网络层更新命令选择子域

### 3.5.10.3.1.1 更新信息计数器域

更新信息计数器子域包含一个整数表明包含在更新信息域里的记录数。记录的大小根据



更新命令标识符的值。

### 3.5.10.3.1.2更新命令标识符子域

更新命令标识符子域包含一个整数表明更新信息命令的类型。图3.27包含能插入这个域的值。

命令标识符值	记录类型
0x00	PAN标识符更新
0x01	网络更新
0x02-0x07	保留

图3.27更新命令标识符子域

### 3.5.10.3.2EPID域

EPID包含64位EPID确定网络在更新。

### 3.5.10.3.3更新信息

更新信息域根据更新命令标识符子域的值提供正在更新的信息，域的格式。

#### 3.5.10.3.3.1PAN标识符更新

如果更新命令标识符子域的值表示一个PAN标识符更新，那么更新信息域由一个单独的16位PAN标识符组成，这个标识符是网络正在使用的新的标识符。更新信息计数器将设置等于1，仅仅只有一个单独的PAN标识符包含在更新信息域中的。

#### 3.5.10.3.3.2网络更新

如果更新命令标识符子域的值表示一个网络更新，那么更新信息由潜在的网络使用的有效信道的32位bitmask，使用的信道的32位bitmask和nwkManager Addr组成。

## 3.6常量和NIB属性

### 3.6.1网络层常量

网络层所定义的特性常量如表3.41所示。

常量	描述	值
<i>nwkCoordinatorCapable</i>	布尔标记，表明设备是否具有成为ZigBee协调器的能力。其中0x00表明设备不具备这样的能力；0x01表明设备有成为ZigBee协调器能力	在初始化时设定
<i>nwkDefaultSecurityLevel</i>	使用的缺省安全级别（参见第4章）	ENC-MIC-64
<i>nwkDiscoveryRetryLimit</i>	路由发现重试的最大次数	0x03
<i>nwkMaxDepth</i>	一台设备拥有的最大深度（离ZigBee协调器的最小逻辑跳数）	0x0f
<i>nwkMinHeaderOverhead</i>	由网络层加到载荷中的最大字节数	0x08
<i>nwkProtocolVersion</i>	设备中ZigBee网络层协议的版本	0x02
<i>nwkWaitBeforeValidation</i>	在接收路由应答和发送有效路由信息之间，多播路由请求发送者持续时间（单位：毫秒）	0x500
<i>nwkRepairThreshold</i>	链路失败和网络层启动维修机制后，所能允许的最大通信错误数	0x03
<i>nwkRouteDiscoveryTime</i>	直到路由发现终止，所需的持续事件（单位：毫秒）	0x2710
<i>nwkMaxBroadcastJitter</i>	最大广播不稳定时间（单位：毫秒）	0x40
<i>nwkInitialRREQRetries</i>	路由请求命令帧的第一个广播传输的重试次数	0x03
<i>nwkRREQRetries</i>	中间ZigBee路由器或协调器请求命令帧广播重传次数	0x02
<i>nwkRREQRetryInterval</i>	广播路由请求命令帧重传的间隔（单位：毫秒）	0xFF

<i>nwkMinRREQJitter</i>	路由请求命令帧重传的最小不稳定(2毫秒间隔)	0xFE
<i>nwkMaxRREQJitter</i>	路由请求命令帧重传的最大不稳定(2毫秒间隔)	0x40
<i>nwkMACFrameOverhead</i>	ZigBee网络层使用的MAC层帧头的大小。ZigBee协议栈Profile能增加这个常量值以保证与IEEE802.15.4 2003协议兼容	0x0b

### 3.6.2网络层信息库

网络层信息库（NIB）由管理设备网络层所需要的属性组成。每一个属性都可以分别使用NLME-GET.request和NLME-SET.request原语进行读写。NIB属性如表3.42所示。

表3.42网络层信息库属性

属性	代码	类型	有效值范围	描述	缺省值
<i>nwkSequenceNumber</i>	0x81	整型	0x00-0xff	加到输出帧上的序列号	范围内的随机值
<i>nwkPassiveAck-Timeout</i>	0x82	整型	0x00-0x0a	父设备与所有子设备重传广播信息的最长持续时间（单位：秒，被动确认超时）	0x03
<i>nwkMaxBroadcast-Retries</i>	0x83	整型	0x00-0x05	广播帧传送失败后最大重传次数	0x03
<i>nwkMaxChildren</i>	0x84	整型	0x00-0xff	现有网络上所能拥有的最大子设备数	0x07
<i>nwkMaxDepth</i>	0x85	整型	0x01- <i>nwkMaxDepth</i>	设备拥有的蛇深度	0x05
<i>nwkMaxRouters</i>	0x86	整型	0x01-0xff	设备能接入的路由器数。网络中所有设备的此值都是由ZigBee协调器来决定	0x05
<i>nwkNeighborTable</i>	0x87	设置	可变	设备中现有的邻居表	未设置
<i>nwkNetworkBroadcast-DeliveryTime</i>	0x88	整型	( <i>nwkPassiveAckTimeout</i> * <i>nwkBroadcastRetries</i> )-0xff	广播信息漫布整个网络的持续时间（单位：秒）	<i>nwkPassiveAckTimeout</i> * <i>nwkBroadcastRetries</i>
<i>nwkReportConstantCost</i>	0x89	整型	0x00-0x01	如果设置为0，则网络层将使用MAC层所报告的LQI值计算所有邻居节点链路的成本。否这它将报告一个常量值	0x00
<i>nwkRouteDiscovery-</i>	0x8a	整	0x00-0x03	在失败路由请求	Nwkc

<i>RetriesPermitted</i>		型		之后允许重试的最大次数	Discovery RetryLimit
<i>nwkRouteTable</i>	0x8b	设置	可变	设备的现有路由表	未设置
<i>nwkSymLink</i>	0x8e	布尔型	TRUE or FALSE	现有的路由对称设置TRUE，表示路由默认由对称链路组成。路由发现期间建立了前向和后向路由，并且二者是相同的。FALSE表示路由不是由对称链路组成。在路由发现期间只有前向路由被保留	FALSE
<i>nwkCapabilityInformation</i>	0x8f	比特组	参见表3.18	包含网络连接期间建立的设备能力信息	0x00
<i>nwkAddrAlloc</i>	0x90	整型	0x00 - 0x02	确定分配地址方法的值：0x00=使用分布式地址分配，0x01=保留，0x02=使用随机地址分配	0x00
<i>nwkUseTreeRouting</i>	0x91	布尔型	TRUE orFALSE	确定网络层是否有使用分等级路由（树形路由）能力的标志：TRUE=有使用分等级路由能力；FALSE=不使用分等级路由	TRUE
<i>nwkManagerAddr</i>	0x92	整型	0x0000 -0xffff7	指定的网络管理地址	0x0000
<i>Reserved</i>	0x93				
<i>Reserved</i>	0x94				
<i>nwkTransaction-PersistenceTime</i>	0x95	整型	0x0000 -0xffff	协调器存储处理的最大时间（在超帧周期）且表明在它的信标内。这个属性反应了MAC PIB属性中的 <i>Transaction- Persi</i>	0x01f4

				<i>stenceTime</i> 的值且高层对此值的任何改变将反应在MAC PIB属性值中	
<i>nwkShortAddress</i>	0x96	整型	0x0000 -0xfff7	设备使用的PAN通信的16位地址。这个属性反应了MAC PIB属性中的 <i>macShortAddress</i> 的值且高层对此值的任何改变将反应在MAC PIB属性值中	0xffff
<i>nwkStackProfile</i>	0x97	整型	0x00-0x0f	设备中使用的ZigBee协议栈的profile标识符	0
<i>Reserved</i>	0x98				
<i>nwkGroupIDTable</i>	0x99	设置	可变	设备是这个网络组的成员，标识符的设置范围在0x00 -0xff	未设置
<i>nwkExtendedPANID</i>	0x9A	64位扩展地址	0x00000000-0xfffffffffffffe	设备是这个网络的成员的扩展PAN标识符。 0x000000000-0000000标识扩展的PAN标识未知	0x0000000000000000
<i>nwkUseMulticast</i>	0x9B	布尔型	TRUE or FALSE	确定多播信息在哪一层发生的标志。TRUE=多播发生在网络层；FALSE=多播发生在应用子层且使用应用子层帧报头	TRUE
<i>nwkRouteRecordTable</i>	0x9C	设置	可变	路由记录表	未设置
<i>nwkSetConcentrator</i>	0x9D	布尔型	TRUE or FALSE	确定设备是否是集中器的标志。TRUE=是被是集中器；FALSE=设备不是集中器	FALSE

<i>nwkConcentratorRadius</i>	0x9E	整型	0x00 - 0xff	协调器路由发现的跳计数器被半径	0x0000
<i>nwkConcentrator-DiscoveryTime</i>	0x9F	整型	0x00 - 0xff	两个协调器路由发现的间隔时间（单位：秒）如果设置成0x0000只有高层在启动时发现路由	0x0000
<i>nwkLinkStatusPeriod</i>	0xA0	整型	0x00 - 0xff	链路状态命令帧之间的间隔（单位：秒）	0x0F
<i>nwkRouterAgeLimit</i>	0xA1	整型	0x00- 0xff	在链路成本复位到0前，丢失的链路状态命令帧个数	3
<i>nwkUniqueAddr</i>	0xA2	布尔型	TRUE or FALSE	确认网络层是否检测和改正冲突地址的标志： TRUE=所用地址是唯一的； FALSE=所用地址不唯一	TRUE
<i>nwkAddressMap</i>	0xA3	设置	可变	64位IEEE当前的设置到16位网络地址地图	未设置
<i>nwkTimeStamp</i>	0xA4	布尔型	TRUE or FALSE	确定输入输出数据包是否提供一个时间表示的标志。TURE=提供了时间表示； FALSE=未提供了时间表示	FALSE

表3.43路由记录表入口帧格式

域名	域类型	有效值范围	相关信息
Network Address	整型	0x0000-0xffff7	路由记录的目的短地址
Relay Count	整型	0x0000 - 0xffff	从协调器到目的设备的应答节点的计数器
Path	网络地址设置		短地址的设置表示从协调器到目的设备的路由顺序

表3.44网络地址地图

64位IEEE地址	16位网络地址
一个有效的64位IEEE地址或者如果未知就是NULL	0x0000 - 0xffff7

### 3.7 功能描述

#### 3.7.1 网络 and 设备的维护

所有的 ZigBee 设备都具有以下功能：

1. 连接网络。
2. 断开网络。

ZigBee 协调器和路由器都具有以下附加功能：

1. 允许设备用如下方式与网络连接：
  - ① MAC 层的连接命令。
  - ② 应用层的连接请求命令。
2. 允许设备以如下方式断开网络：
  - ① MAC 层的断开命令。
  - ② 应用层的断开命令。
  - ③ 对逻辑网络地址进行分配。
  - ④ 维护邻居设备表。

ZigBee 协调器应具有建立一个新网络的功能。

ZigBee 路由器和终端设备在一个网络中应提供轻便支持。

##### 3.7.1.1 建立一个新的网络

设备通过 NLME-NETWORK-FORMATION.request 原语来启动一个新的网络的建立过程。仅仅当具有 ZigBee 协调器能力，且当前还没有与网络连接的设备才可以尝试着去建立一个新的网络。如果该过程由其他设备开始，则网络层管理实体将终止该过程，并向其上层发出非法请求的报告。该步骤通过发出状态参数为 INVALID\_REQUEST 的 NLME-NETWORKFORMATION.confirm 原语来完成。

当建网过程开始后，网络层将首先请求 MAC 层对协议所规定的信道，或由物理层所默认的有效信道进行能量检测扫描，以检测可能的干扰。为实现能量检测扫描，设备网络层通过发送扫描类型（ScanType）参数设置为能量检测扫描的 MLME-SCAN.request 原语到 MAC 层进行信道能量检测扫描，扫描结果通过 MLME-SCAN.confirm 原语返回。

当网络层管理实体收到成功的能量检测扫描结果后，将以递增的方式对所测量的能量值进行信道排序，并且抛弃那些能量值超出了可允许能量水平的信道，选择可允许能量水平的信道有待进一步处理。此后，网络层管理实体将通过发送 MLME-SCAN.request 原语执行主动扫描，其中该原语的 ScanType 参数设置为主动扫描，ChannelList 参数设置为可允许信道的列表，搜索其他的 ZigBee 设备。为了决定用于建立一个新网络的最佳通道，网络层管理实体将检查 PAN 描述符，并且所查找的第一个信道为网络的最小编号。

如果网络层管理实体找不到适合的信道，就将终止建网过程，并且向应用层发出启动失

败信息，即通过发送参数状态为STARTUP\_FAILURE的NLME-NETWORK-FORMATION.confirm原语向其上层通告。

如果网络层管理实体找到了合适的信道，则将为这个新网络选择一个PAN标识符。为了选择一个PAN标识符，设备将选择一个随机的PAN标识符值小于等于3fff没有在已选择信道里使用的。一旦网络层管理实体做出了选择，则它通过发出MLME-SET.request原语将这个值写为MAC层macPANId属性。

如果选择不出唯一的标识符，网络层管理实体将终止程序，并且通过发送状态参数为STARTUP\_FAILURE的NLME-NETWORK-FORMATION.confirm原语向其上层通告。

网络层管理实体一旦选择了一个PAN标识符，将选择一个等于0x0000的16位网络地址，并且设置MAC层的macShortAddress PIB属性，使其等于所选择的网络地址。

一旦选择了网络地址，网络层管理实体核对PIB属性的*extendedPANId*的值。如果这个值是0x0000000000000000这个属性以MAC常量*aExtendedAddress*初始化。

一旦*nwkExtendedPANId*的值核对，网络层管理实体通过MLME-START.request原语给MAC层开始新的PAN操作。MLME-START.request原语的参数根据NLME-NETWORK-FORMATION.request原语来施舍，即根据信道扫描和所选择的PAN标识符来设置。PAN的启动状态通过MLME-START.confirm原语返回到网络层。

当网络层管理实体收到PAN的启动状态后，将向启动ZigBee协调器请求状态的上层报告，即通过发出the NLME-NETWORK-FORMATION.confirm原语向其上层报告，其原语的状态参数为从MAC层的MLMESTART.confirm原语所返回的值。

成功启动一个新网络的信息流程如图3.28所示。



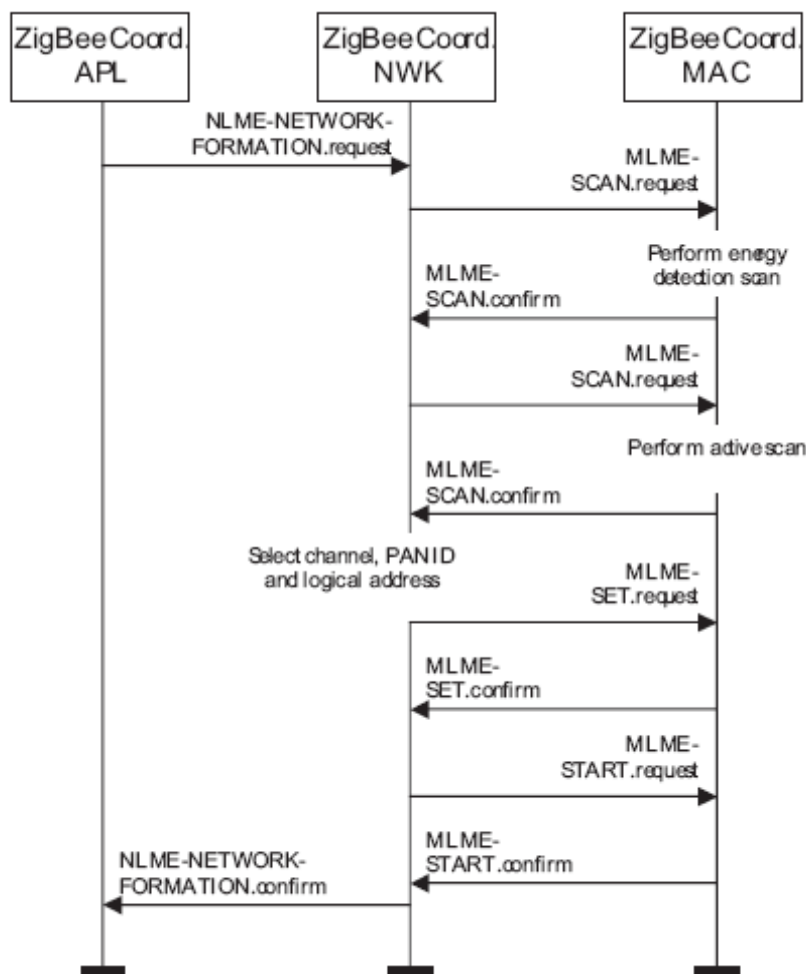


Figure 3.28 Establishing a New Network

### 3.7.1.2 允许设备与网络连接

通过NLME-PERMIT-JOINING.request原语来允许设备与网络连接。仅仅只有设备为ZigBee协调器或者路由器时，才能企图允许设备与网络连接。

当此过程开始时，若设置PermitDuration参数为0x00则启动该过程，并且网络层管理实体把在MAC层的*macAssociationPermit* PIB属性设置为FALSE。MAC层的属性设置通过MLME-SET. Request原语来完成。

当此过程开始时，若设置PermitDuration参数为一个0x01和0xFE之间的值时，则网络层管理实体将把在MAC层中的*macAssociationPermit* PIB属性设置为TRUE，并且网络层管理实体将启动一个定时器，用来对一个特定的时间进行计时，达到该时间时，定时器停止计时，在该定时器停止时，网络层管理实体将把MAC层中的*macAssociationPermit* PIB属性设置为FALSE。

当此过程开始时，若设置PermitDuration参数设置为0xFF，则网络层管理实体将把在MAC层中的*macAssociationPermit* PIB属性设置为TRUE，以表示无限定时间，除非发送另一个NLME-PERMITJOINING.request原语。

允许设备同网络连接的过程如图3.29所示。

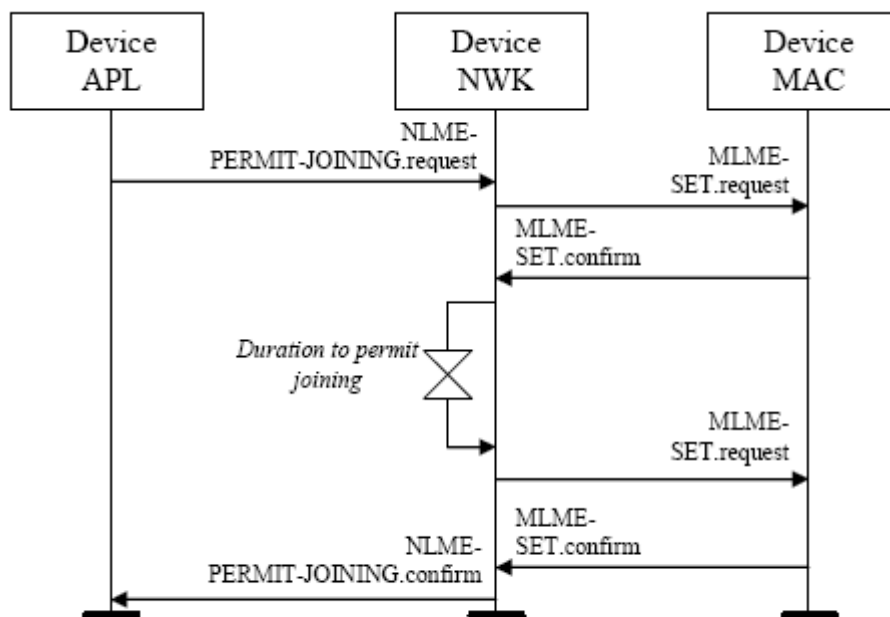


Figure 3.29 Permitting Devices to Join a Network

### 3.7.1.3连接网络

在一个网络中具有从属关系的设备允许一个新设备连接时，它就与新连接的设备形成了一个父子关系。新设备成为子设备，而第一个设备为父设备。一个子设备通过一下两个方法加入到网络中：

- ① 子设备用MAC连接程序来加入网络；
- ② 在设备直接同一个预先所指定的父设备连接来加入网络。

#### 3.7.1.3.1通过联合方式加入网络

本小节详细介绍了一个子设备同一个网络连接的过程，以及一个ZigBee协调器或路由器（父设备）在接收到连接请求命令后所采取的措施。

##### 3.7.1.3.1.1子设备流程

通过MAC层连接程序连接网络的流程为：首先，应用层发送NLME-NETWORK-DISCOVERY.request原语，其中扫描参数（ScanChannels）设置为网络将要扫描的信道，扫描持续时间参数（ScanDuration）设置为扫描每个信道所需要的时间。网络层接收到该原语后，将发送MLME-SCAN.request原语请求MAC层执行一个主动扫描。

扫描设备的MAC层在扫描过程中一旦接收到有效长度不为零的信标帧时，将向其网络层发送MLME-BEACON-NOTIFY.indication原语。该原语中包括的信息为信标设备地址、是否允许连接和信标载荷。（见【B1】参数完整列表）扫描设备的网络层将检查信标载荷中的协议标识符域的值，并验证它是否与ZigBee协议识别符匹配。如果不匹配，则忽略该信标。反之，设备将从接收到的信标中，将相关的信息（见图3.42信标载荷结构）复制到的的邻居表中（见表3.45邻居表条目内容）。

一旦MAC层完成对信道的扫描，在向网络层管理实体发送MLME-SCAN.confirm原语后，网络层将发送NLME-NETWORK-DISCOVERY.confirm原语，其参数包括扫描得到的网络描述参数。这些描述参数为ZigBee版本号、堆栈结构、扩展个域网网标识符（PANId）、个域网网标识符（PANId）、逻辑信道和是否允许连接的信息（见表3.9）。

其上层收到NLME-NETWORK-DISCOVERY.confirm原语，就可得到目前邻居网络的信息。以便发现更多的网络或者其他原因，上层可以选择重新执行网络发现命令。如果不重新执行，它将从所发现的网络中选择一个网络进行连接，即通过发送NLME-JOIN.request原语

进行连接，其中RejoinNetwork参数设置为0x00，且JoinAsRoute参数设置为设备是否同网络连接。

仅仅只有那些还没有同网络连接的设备才能执行该连接流程。如果任何其他设备执行这个流程，则网络管理实体将终止这个流程，并且向上层发送状态参数为INVALID\_REQUEST的NLME-JOIN.confirm原语。

对于一个还没有同网络连接的设备，NLME-JOIN.request原语将使得网络层在邻居表中搜索一个合适的父设备。一个合适的父设备必须具备2个条件：允许连接；链路成本最大为3（见3.7.3.1链路成本的详细计算）。如果在邻居表中存在潜在的父设备子域，则该子域设置为1。

如果邻居表中不包括合适的父设备，网络层管理实体将发送参数状态为NOT\_PERMITTED的NLME-JOIN.confirm原语。如果邻居表中包括不只一个合适的父设备，则选择具有到ZigBee协调器最小深度的设备。如果存在多个到ZigBee协调器最小深度的设备，则可在它们之间任意地选择一个。

一旦选择了一个合适的父设备，网络层管理实体将向MAC层发送MLMEASSOCIATE.

Request原语，其原语的地址参数为在邻居表中所选择的设备地址，并通过MLMEASSOCIATE.confirm原语将连接的状态返回到网络层管理实体。

如果试图连接网络没有成功，网络层将收到从MAC层发送来的MLMEASSOCIATE.

Confirm原语，其状态参数为错误代码。如果状态参数表明拒绝与邻居设备连接（即PAN容量或者PAN接入拒绝），则尝试连接的设备将把邻居表中潜在的父设备子域设置为0，以表示尝试连接失败。潜在的父设备子域为0使得网络层将不会发送另一个连接请求原语去尝试连接该邻居设备。每次发送MLMESCAN.request原语，将邻居表中的潜在的父设备子域设置为1。

如果潜在的父设备不允许连接新的路由器（路由器的最大数，已经连接设备的最大路由器），并且要连接的设备将JoinAsRouter参数设置为TRUE，则连接请求也可能不成功。在这种情况下，NLMEJOIN.confirm原语将给出NOT\_PERMITTED的状态，子设备应用层将希望再次尝试连接，但只能作为一个终端设备，将发送另一个NLME-JOIN.request原语，且原语的JoinAsRouter参数设置为FALSE。

如果尝试连接网络失败，网络层管理实体将试图从邻居表中找寻一个合适的父设备。如果不存在这样的设备，网络管理实体将发出NLME-JOIN.confirm原语，其状态参数值为MLME-ASSOCIATE.confirm原语所返回的值。

如果尝试连接失败，并且存在第二个邻居的设备，该设备可以作为合适的父设备，则网络层启动连接第二个设备的MAC层连接程序。网络层将不断重复这个过程，知道直到成功的与网络连接或者已尝试所有可能连接的网络。

如果设备不能成功的连接由上层所指定的网络，网络管理实体将通过NLME-JOIN.confirm原语来终止该过程，其原语的状态参数为最后接收到的MLME-ASSOCIATE.confirm原语所返回的值。在这种情况下，设备将不接收有效的逻辑地址，也不允许在网络中通信。

如果尝试连接网络成功，网络层收到MLME-ASSOCIATE.confirm原语，该原语中将包括16位的逻辑地址，该逻辑地址在网络中是唯一的，并且该子设备在未来的通信中将使用这个逻辑地址。然后，网络层将设置相对应的邻居表的关系域，以表示邻居设备为它的父设备。此时，父设备将把新连接的设备增加到它的邻居表中。而且网络层将更新NIB中nwkShortAddress的值。

如果设备试图同一个安全网络连接且它是路由器，则在发送信标前必须等待父设备对它进行验证，验证之后就可以进行连接。因而，该设备将等待上层发送来的NLME-START-

ROUTER.request原语。如果设备为一个路由器，当它的网络层管理实体接收到该原语，就发送MLME-START.request原语。如果NLME-STARTROUTER.request原语由一个终端设备发出，则网络层将发出NLME-START-ROUTER.confirm原语，其原语状态参数设置为INVALID\_REQUEST。

当设备成功的同网络连接，如果设备是路由器且上层将发出NLME-START-ROUTER.request原语，则网络层将向MAC层MLME-START.request原语。PANId、LogicalChannel、BeaconOrder和SuperframeOrder参数设置将设置为它所对应的父设备在邻居表中的所对应的参数值。而PANCoordinator和CoordRealignment参数都会设置为FALSE，网络层接收到MLME-START.confirm原语后，将发送具有相同状态的NLME-START-ROUTER.confirm原语。

图3.30为通过联合方式同网络连接的流程。

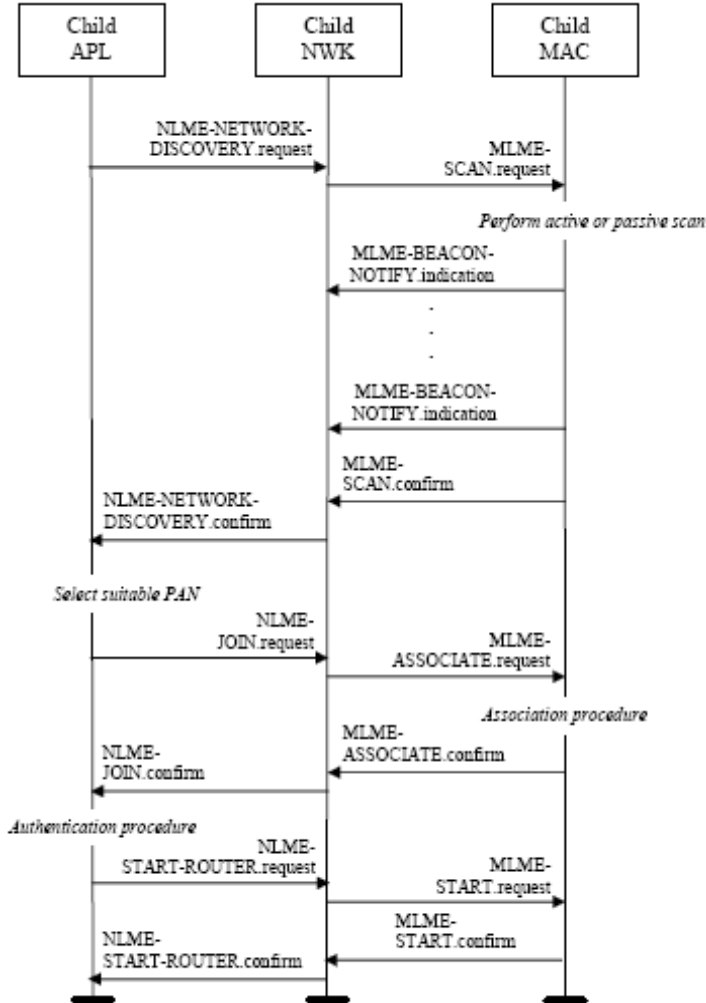


Figure 3.30 Procedure for Joining a Network Through Association

3.7.1.3.1.2父设备流程

ZigBee协调器或者路由器使用MAC层将一个设备同它所在的网络进行连接，其流程来自于MAC层的MLMEASSOCIATE.indication原语来进行初始化。仅仅当这些设备为协调器或者路由器，并且允许同网络连接的设备时，才能执行这个流程。如果设备为其他设备，网络层管理实体将终止这个流程。

当这个流程开始后，潜在父设备的网络层管理实体首先将要确定设备是否愿意同已经存在的网络连接。为了确定这一点，网络层管理实体将会搜索的邻居表以确定是否能找到一个

匹配的64位扩展地址。如果搜索到相匹配的地址，则网络层管理实体将检查在邻居表中给定的设备能力是否匹配设备类型。如果设备类型也匹配则网络层管理实体将得到一个相应的16位网络地址，并且向MAC层发送连接响应。如果设备类型不匹配，网络管理实体将移除邻居表中设备的所有记录且重新启动MLME-ASSOCIATION.indication。如果搜索不到相匹配的地址，如果可能，网络管理实体将分配一个16位的网络地址给这个新设备。见3.7.1.5节和3.7.1.6节地址分配解释。

如果潜在的父设备没有能力接受更多的子设备（用完了它的分配地址空间），则网络管理实体将终止该流程，然后向MAC层发出MLME-ASSOCIATE.response原语对其响应。该原语的状态参数将表明PAN的能力。

如果同意连接请求，则父设备的网络管理实体将使用设备所提供的信息在它的邻居表中为子设备创建一个新的入口。并且随后向MAC层发送表明连接成功的MLME-ASSOCIATE.response原语。MLME-COMMSTATUS.indication原语将传送给子设备的响应状态回到网络层。

如果传送不成功（即MLME-COMM-STATUS.indication原语状态参数不为SUCCESS），则网络层管理实体将终止程序。如果传送成功，网络层管理实体将通过向上层发送NLME-JOIN.indication原语，表明子设备已经成功地同网络连接。

成功的将设备同网络连接的流程如图3.31。

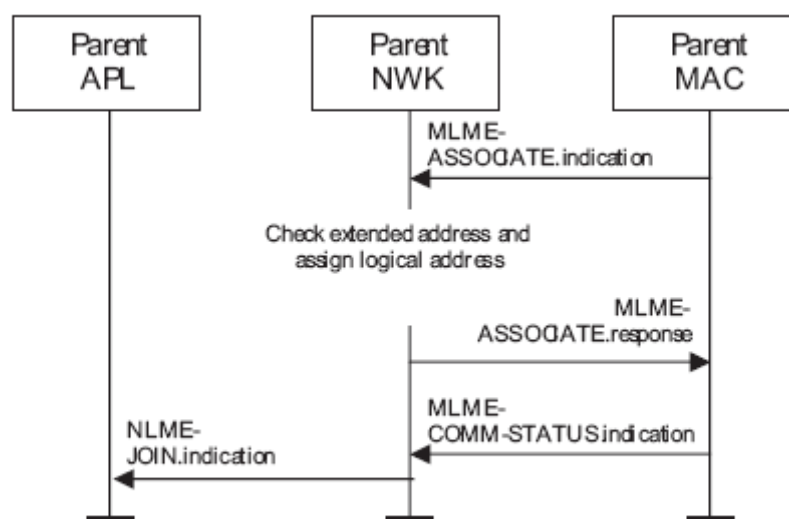


Figure 3.31 Procedure for Handling a Join Request

### 3.7.1.3.2重新连接网络

已经同网络失去所有联系的设备，例如一个ZED同它的父设备失去联系，能通过使用网络层重新连接请求和网络层重新连接响应命令来重新加入到网络。重新加入过程同前面所说的联合过程是一样的，除了MAC层联合过程被包括重新加入请求和重新加入响应命令的交流所代替。且因为网络层命令使用网络层安全，没有认证步骤执行。使用这些命令代替MAC层程序允许设备重新连接网络而不是一个新设备连接。

#### 3.7.1.3.2.1子设备流程

重新连接一个网络的流程是通过发出NLME-JOIN.request原语来启动网络层重新连接流程，如图3.32所示，原语中RejoinNetwork参数设置为0x02且ExtendedPANId参数设置成网络冲连接的ExtendedPANId。JoinAsRouter参数设置将表明设备是否愿意作为路由设备连接网络。

ScanChannels参数将设置成表示哪些信道被扫描来定位这个网络且ScanDuration参数设

置来表示扫描每一个信道的时间长度。

一旦接收到这条原语，网络层发送MLME-SCAN.request原语请求MAC层进行主动扫描。

扫描设备的MAC层在扫描过程中接收到有效长度不为零的信标帧时，将向其网络管理实体发送MLME-BEACON-NOTIFY.indication原语。扫描设备的网络管理实体检查有效信标中包含的ExtendedPANId看它是否是正确的值。如果不正确，则忽视该信标。反之，设备将从所接收到的信标中，将相关信息复制到它的邻居表中。

一旦MAC层完成对信道的扫描，在向网络层管理实体发送MLME-SCAN.confirm原语，网络层将为合适的父设备搜索邻居表。一个合适的父设备将advertise在JoinAsRoute参数的请求类型的设备能力且链路成本最大应是3。如果邻居表没有合适的父设备，网络管理实体将用状态参数是NOT\_PERMITTED的NLME-JOIN.confirm原语来响应。如果有不止一个设备是合适的父设备且nwAddrAlloc NIB属性值是0x00，则选择具有到ZigBee协调器最小深度的设备。一旦选择了一个合适的父设备，网络层管理实体将构造一个网络层重新请求命令帧。在重新连接命令中的地址参数将设置成包含从邻居中选择的设备的地址信息。

在用数据服务网络层重新连接请求命令成功传送后，将加载一个倒计时的定时器，它的值是aResponseWaitTime（【B1】），如果RxOnWhenIdle参数值是FALSE，当定时器停止时，网络层将产生MLME-POLL.request原语给潜在的父设备，来重新获得重新连接响应命令。

一旦接收到重新响应命令帧，在以上步骤之后或者任何其他时间，设备将检查有效命令帧的重新连接设备的IEEE地址域和父设备的IEEE地址域。如果重新连接设备的IEEE地址域不等于接收到设备的IEEE地址的值或者如果父设备的IEEE地址域不等于重新请求命令帧发送给的最新的潜在父设备的IEEE地址值，或者当前父设备是主动提供重新连接响应的情况，那么丢弃重新请求命令帧没有进一步处理。

如果重新请求响应命令帧的重新连接状态域表明拒绝允许重新连接邻居设备（也就是，PAN在能力上或PAN访问被拒绝），那么设备试图重新连接应该在相应的邻居表入口设置潜在的父设备位为0来表明企图连接失败。设置潜在父设备位为0确保网络层不发送其他的请求来重新连接一个相同的邻居设备。如果试图连接失败，那么网络层管理实体将试图在邻居表中寻找另一个合适的父设备。如果没有找到这样的设备，网络管理实体将发出状态参数设置是NOT\_PERMITTED的NLME-JOIN.confirm的原语。如果试图连接失败且有第二个邻居设备是合适的父设备，网络层将发出网络层重新连接程序与这第二个设备。网络层将不断重复这个过程直到它重新连接PAN成功或者是耗尽它重新连接PAN的选择权。如果其高层说明这个设备重新连接PAN没有成功，那么网络层管理实体将使用状态参数设置是NOT\_PERMITTED的NLMEJOIN.confirm原语来终止这个过程。在这种情况下，设备不会收到一个有效的逻辑地址且不允许在这个网络传送。如果试图重新连接成功，网络层将收到包含一个在网络中唯一的16位逻辑地址的网络层重新连接响应命令，子设备在未来的传送中要使用这个地址。网络层设置相应的邻居表入口的相关域来表明这个邻居是它的父设备。此时，父设备增加这个新设备到它的邻居表中。另外，网络层将更新在NIB中nwShortAddress的值。



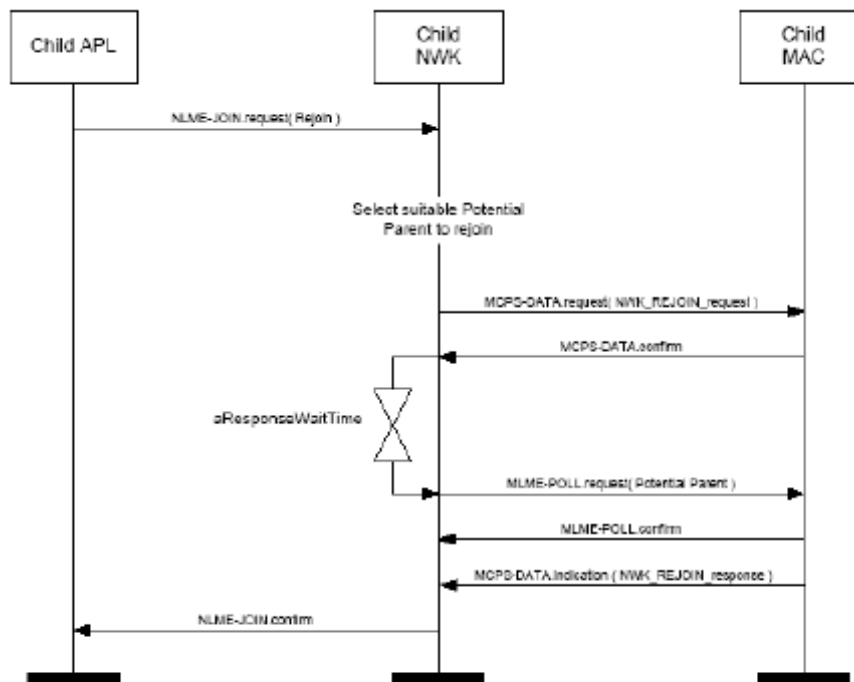


Figure 3.32 Child Rejoin Procedure

#### 3.7.1.3.2.2父设备流程

ZigBee协调器或者路由器重新连接一个设备到它的网络的流程是使用通过MAC数据服务使网络层重新请求命令帧的到达来执行网络层重新连接流程。仅仅当这些设备为ZigBee协调器或者路由器时才可以执行这个流程。如果是其他设备执行该流程，那么网络管理实体将终止该流程。当该流程开始时，潜在父设备的网络层管理实体首先确定它是否已经知道请求设备。为了确定这一点，网络层管理实体将搜索它的邻居表来确定是否有一个匹配的64位扩展地址。如果搜索到匹配的扩展地址，网络层管理实体将检查提供的设备能力是否匹配邻居表中记录的设备类型。如果设备类型也匹配，网络管理实体将考虑连接企图成功且用这个在邻居表中搜索到的16位网络地址作为正在连接设备的网络地址。如果设备类型不匹配，那么网络管理实体将移除在它的邻居表中的设备所有记录，且重新开始网络层重新连接命令流程。如果没有搜索到匹配的扩展地址，如果可能，网络管理实体给这个新设备分配一个16位的网络地址。见3.7.1.5节和3.7.1.6节地址分配机制详述。如果潜在的父设备没有能力接受连接设备，网络管理实体将终止这个流程，且表明这个事实在后来的重新响应命令。命令的状态参数将表明PAN的能力。如果允许重新连接请求，父设备的网络层管理实体将在邻居表中为子设备创建一个新的入口，或者如果已经存在这个入口就用提供的设备信息修改这个现有的入口，且通过使用网络层重新响应命令回答请求设备来表明重新连接成功。网络层管理实体发送NLME-JOIN.indication原语通知其上层子设备已经重新连接这个网络。设备成功重新连接网络的流程如图3.33。



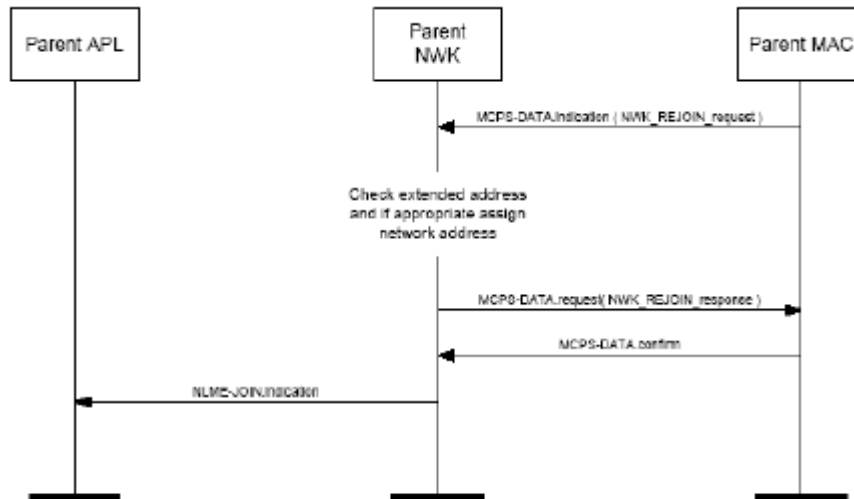


Figure 3.33 Parent Rejoin Procedure

### 3.7.1.3.3 直接方式连接网络

本小节将介绍子设备如何通过预先分配的父设备（ZigBee协调器或路由器）直接同网络连接的过程。在这种情况下，父设备将为子设备预先分配一个64位地址。下面将描述如何使用这个优先地址来建立父子网络关系。

ZigBee协调器或者路由器直接将一个设备加入它所在的网络的流程是从发送NLME-DIRECT-JOIN.request原语开始的，其原语的DeviceAddress参数设置为要连接网络的设备地址。仅仅只有设备为ZigBee协调器或者路由器时，才可以执行这个流程。如果其他设备执行这个流程，网络层管理实体将终止该流程，并且将发送NLME-DIRECT-JOIN.confirm原语向其上层通告其非法请求，其原语的状态参数设置为INVALID\_REQUEST。

当该流程开始后，父设备的网络层管理实体将首先确定所指定的设备是否存在于网络中。为完成这个过程，网络层管理实体将搜索它的邻居表，以确定是否有一个相匹配的64位扩展地址。如果存在一个相匹配的64位地址，则网络层管理实体将终止该流程，并发送NLME-DIRECT-JOIN.confirm原语向其上层通告该设备已经存在于网络设备列表中，其原语的状态参数设置为ALREADY\_PRESENT。

如果不存在一个相匹配的64位地址，如果可能，网络层管理实体将为此新设备分配一个16位网络地址和一个新的邻居表入口。见3.7.1.5节和3.8.1.6节地址分配机制。如果父设备在邻居表中没有足够的空间，那么网络层管理实体将终止该流程，并且发送NLME-DIRECT-JOIN.confirm原语向其上层通告空间不够，其原语状态参数设置为NEIGHBOR\_TABLE\_FULL。如果存在足够的空间，则网络层管理实体将发送NLME-DIRECT-JOIN.confirm原语向其上层通告设备已经同网络连接，其原语状态参数设置为SUCCESS。

一旦父设备将子设备同网络连接，子设备为了建立父子网络关系必须与父设备进行通信。子设备将启动孤点流程来完成这个请求，古典流程将在3.7.1.3.3.1描述。

父设备成功地将子设备直接同它的网络连接流程如图3.34。在这个流程中他们不在空中交换任何信息。

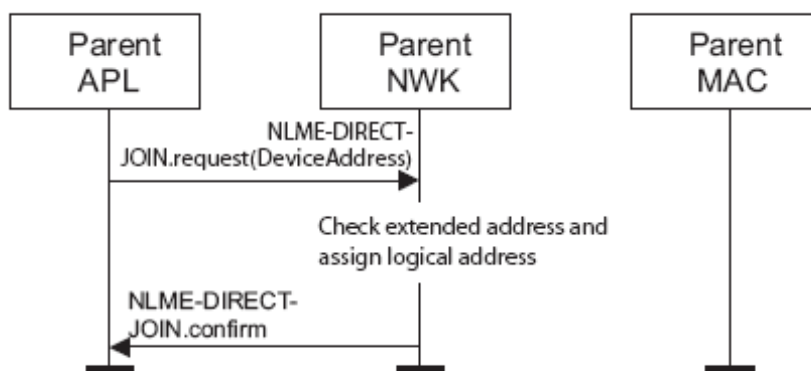


Figure 3.34 Joining a Device to a Network Directly

### 3.7.1.3.3.1通过孤点方式连接或重新连接网络

本小节介绍了一个已经直接同网络连接的设备（通过孤点方式）或者一个以前同网络连接的设备，但目前没有和它的父设备失去联系，它将（通过孤点方式重新连接）如何执行孤点连接流程同网络连接。

一个已经同网络连接的设备为了完成建立它与其父设备的关系，应开始执行孤点流程，设备的应用层将决定是否开始该流程，如果开始，则应用层将通过网络层打开电源。

如果一个以前已经同网络连接的设备，其网络层管理实体将不断地接收到来自于MAC层发送的通信失败通知，则它将开始执行孤点流程。

### 3.7.1.3.3.2子设备流程

子设备通过发送NLME-JOIN.request原语来开始执行孤点方式同网络连接，其原语的RejoinNetwork参数设置为0x01。

当开始执行流程时，首先，网络层管理实体请求MAC层对ScanChannels参数给定的信道进行孤点扫描。通过向MAC层发送MLMESCAN.request原语开始进行孤点扫描，其扫描的结果通过MLME-SCAN.confirm原语返回到网络层管理实体。

如果孤点扫描成功（即子设备扫描到父设备），网络层管理实体将通过发送NLME-JOIN.confirm原语向其上层通告请求连接或者重新连接网络已成功执行，其原语状态参数设置为SUCCESS。

注意如果子设备是第一次连接或者以前已经同网络连接但是保持树形深度信息失败（树形深度在3.4.6.1中规定），它有可能在网络中不能正确操作，对于消息的恢复超出了规定的范围。

如果孤点扫描不成功（即没有扫描到父设备），网络层管理实体将终止该流程，并通过发送NLME-JOIN.confirm原语向其上层通告没有扫描到网络，其原语的状态参数设置为NO\_NETWORKS。

子设备通过孤点方式连接网络或者重新连接网络的流程如图3.35。

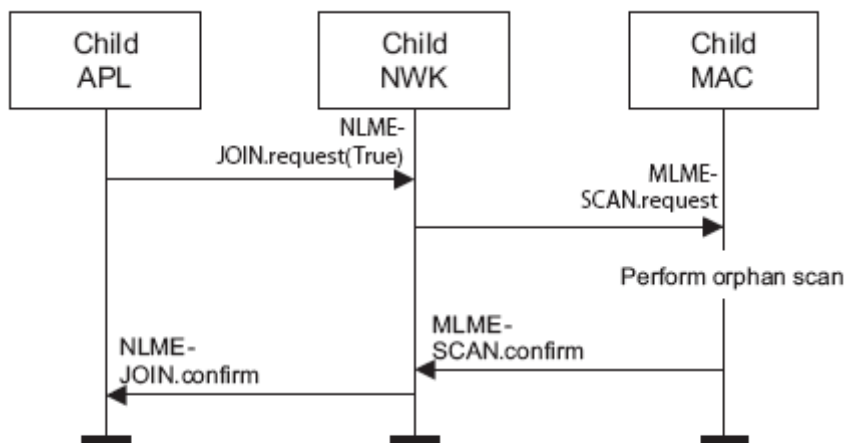


Figure 3.35 Child Procedure for Joining or Re-joining a Network Through Orphaning

### 3.7.1.3.3父设备流程

一个父设备收到来自于MAC层发送来的MLME-ORPHAN.indication原语时，就可得知存在一个孤点设备。仅仅当设备为Zigee协调器或者路由器（也就是具有父设备能力）时，才能执行该连接流程；否这其他设备执行该流程时，网络层管理实体将终止该流程的执行。

该流程开始执行时，网络层管理实体首先判断该孤点是否是它的子设备。为了对其进行判断，需要将孤点设备的扩展地址和邻居表中所记录的子设备地址向比较，如果存在向匹配的的地址（即古典设备是它的子设备），则网络层管理实体将得到其相对应的16位网络地址以及它随后对MAC层的孤点响应状态。网络层管理实体通过向MAC层发送MLME-ORPHAN.response原语对其孤点进行响应，并且通过MLME-COMM-STATUS.indication原语得到其传输状态。

如果不存在相匹配的地址（即孤点设备不是它的子设备），流程终止且不通知上层。

父设备连接或者重新连接孤点设备的流程如图3.36。

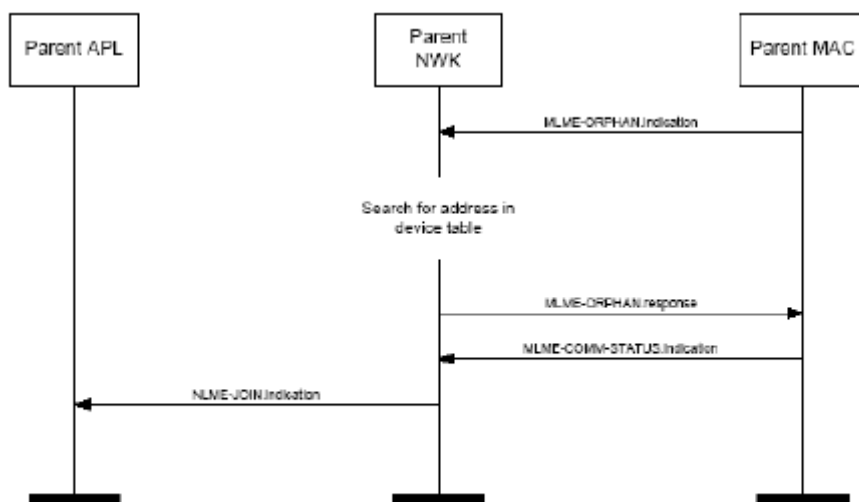


Figure 3.36 Parent Procedure for Joining or Re-joining a Device to its Network Through Orphaning

### 3.7.1.4邻居表

一个设备的的邻居表中应包含在其传输范围中每一个设备的信息。

邻居表是很有用的。首先，它用在网络发现或者重新连接来存储路由相关的信息（在RF

接收范围内)能成为候选父设备。其次,在设备连接到网络之后,它存储在这个网络中的邻居设备的关系和链路状态信息。一个设备从相应的邻居表中接收到任何帧时表入口都要更新。

输出成本域包含邻居表中测试的链路成本。这个值是从邻居表接收到的最新链路状态命令帧处得到的。如果值是0表明接收到的设备没有链路状态命令表。

年龄域表明nwkLinkStatusPeriod间隔的数值是从已经收到的最后链路状态命令帧直到nwkRouterAgeLimit的最大值。(p376)

在正常网络操作中固定的和可选的数据如表3.45所示。

表3.45邻居表入口格式

域名	域的类型	有效值范围	描述
Extended address	整型	一个扩展的64位IEEE地址	每个设备的唯一的64位IEEE地址。如果邻居设备是父设备或子设备,在存在该子域
Network address	网络地址	0x0000-0xffff	邻居设备的16位网络地址。在每一个邻居表中都存在该子域
Device type	整型	0x00-0x02	邻居设备的类型: 0x00为ZigBee协调器 0x01为ZigBee路由器 0x02为ZigBee终端设备 在每一个邻居表中都存在该子域
RxOnWhenIdle	布尔型	TRUE or FALSE	表示邻居设备接收机在超帧活动期的空闲期是否工作。 TRUE为接收机关 FALSE为接收机开
Relationship	整型	0x00-0x04	邻居设备和当前设备的关系: 0x00=邻居设备为父设备 0x01=邻居设备为子设备 0x02=邻居设备为同属设备 0x03=不为上述设备 0x04=先前的子设备 在每个邻居表中都存在该子域
Transmit Failure	整型	0x00-0xff	表明以前设备的传送是否成功。值越大则表明越失败。在每个邻居表中都存在该子域
LQI	整型	0x00-0xff	估计RF传输链路质量。在每个邻居表中都存在该子域
Outgoing Cost	整型	0x00-0xff	邻居设备计算的输出链路成本。值是0表示输出成本有效。该项为选择项
Age	整型	0x00-0xff	接收到链路状态命令后的nwkLinkStatusPeriod间隔的值。该项为选择项
Incoming beaco	整	0x0000000-0xffffffff	从邻居表中接收到的最后一个信标帧的时间

n timestamp	型		标记。这个值等于当就收到信标帧时采用的 timestamp。该项为选择项
Beacon transmission time offset	整 型	0x000000-0xfffff	邻居设备信标与它的父设备的信标之间的传 输时间差。从响应的输入信标时标减去该偏差 就可得计算出邻居父设备传送信标时间。该项 为选择项

在网络发现和重新连接用到的信息如上描述如表3.46所示。所有域都是可选的且在网络层管理实体选择连接网络之后就不能保持。不是所选择网络的设备的邻居表的入口被丢弃。

表3.46附加邻居表域

域名	域类 型	有效值范围	描述
Extended PAN ID	整型	0x0000000000000001 - 0xfffffffffffffffe	设备属于的网络的64位唯一的标识 符
Logical channel	整型	PHY支持的可用逻辑信 道	网络工作的逻辑信道
Depth	整型	0x00- <i>nwkMaxDepth</i>	邻居设备的树状深度
Beacon order	整型	0x00-0x0f	设备的IEEE802.15.4的信标顺序
Permit joining	布尔 型	TRUE or FALSE	表示设备是否接受连接请求。 TRUE=设备正在接受连接请求 FALSE=设备没有接受连接请求
Potential parent	整型	0x00-0x01	表示是否排除设备为潜在的父设 备。 0x00表示设备不是潜在的父设备。 0x01表示设备是潜在的父设备。

### 3.7.1.5分布式地址分配机制

NIB属性*nwkAddrAlloc*的缺省值是0x00，采用分布式地址分配方案来分配网络地址，即该方案为每一个父设备分配一个有限的网络地址段。这些地址在一个特殊的网络中是唯一的，并且由它的父设备分配给它的子设备。ZigBee协调器决定在其网络中允许连接的子设备的最大个数。对于这些子设备，参数*nwkMaxRouters*为路由器最大个数，而剩下的设备数为终端设备数。每一个设备具有一个连接深度，即连接深度表示仅仅采用父子关系的网络中，一个传送帧传送到ZigBee协调器所传递的最小跳数。ZigBee协调器自身深度为0，而它的子设备深度为1。ZigBee协调器决定网络的最大深度。

假定父设备拥有子设备数量的最大值为*nwkMaxChildren (Cm)*，网络的最大深度为*nwkMaxDepth (Lm)*，父设备将路由器最为它的子设备的最大数为*nwkMaxRouters (Rm)*，则可计算函数*Cskip(d)*，该函数为在给定网络深度d和路由器以及子设备个数的条件下，父设备所能分配子区段地址数为：

$$Cskip(d) = \begin{cases} 1 + Cm(Lm - d - 1), & \text{if } Rm = 1 \\ \frac{1 + Cm - Rm - Cm \cdot Rm^{Lm-d-1}}{1 - Rm}, & \text{otherwise} \end{cases}$$

如果一个设备的*Cskip(d)*的值为0，则它没有接收子设备的能力，并且将这样的设备看作作为一个ZigBee网络的终端设备。该设备的网络层管理实体将发送MLME-SET.request原语，将MAC层的个域网信息库中的*macAssociationPermit*属性设置为FALSE，且对将来所接收到

参数PermitDuration的值等于或大于0x01的NLME-PERMIT-JOINING.request原语进行响应，其响应是状态参数为INVALID\_REQUEST的NLME-PERMIT-JOINING.confirm原语。然后终止这个允许连接的流程。

如果父设备的Cskip(d)的值大于0，则可以接受子设备，并且将根据子设备是否具有路由能力来向子设备分配不同的地址。

利用Cskip(d)作为偏移，向具有路由能力的子设备分配网络地址。父设备为它的第一个路由子设备分配一个比自己大1的地址，随后分配给路由子设备的地址将以Cskip(d)为间隔，以此类推为所有的路由器分配地址。nwkMaxRouters的最大值将分配这样的地址。第n个终端设备的网络地址将按照如下的公式进行分配：

$$A_n = A_{parent} + Cskip(d) \cdot Rm + n$$

其中1≤n≤（Cm-Rm），Aparent为父设备地址。

图3. 37给出了一个具有最大子设备数（nwkMaxChildren）为4，最大路由数（nwkMaxRouters）为4，网络最大深度（nwkMaxDepth）为4的ZigBee网络，则利用上述公式计算出的Cskip(d)值如表3.47表示。

表3.47深度与偏差

网络深度d	偏移Cskip（d）
0	21
1	5
2	1
3	0

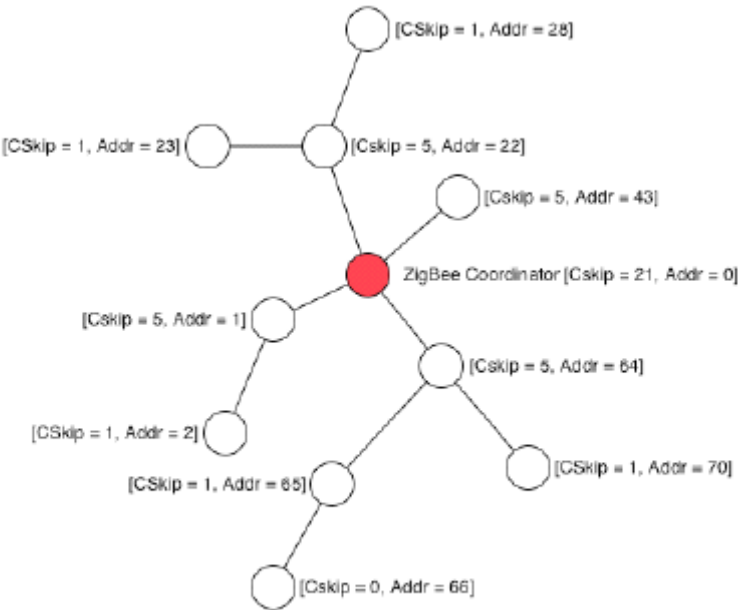


Figure 3.37 Address Assignment in an Example Network

由于在设备之间不能共享一个地址段，因此，当第二层的父设备所具有的地址不用时，则第一层的父设备有可能用尽它的所有地址。一个不具备可用地址的父设备将不允许新设备加入该网络。在这种情况下，新设备将寻找另一个父设备，如果在其传输范围内设备找不到有效的父设备，则该设备将不能加入到该网络，除非物理移动它或者网络有一些其他的变化。

3.7.1.6随机地址分配机制

当NIB属性的nwkAddrAlloc是0x02时，随机选择地址。当设备连接到网络它的父设备选

择一个随机地址，这个随机地址不是已经出现在父设备的NIB终端的任何入口。ZigBee协调器，它没有任何父设备，但仍然有地址0x0000。

### 3.7.1.7安装和寻址

可以清楚的看出 $nwkMaxDepth$ 大致决定了从网络树根到最远的终端设备之间的距离。从理论上来说， $nwkMaxDepth$ 也决定了整个网络的直径。在特殊情况下，对于一个以ZigBee协调器为网络中心的理想网络结构来说，如图3.37所示，其网络直径为 $2 * nwkMaxDepth$ 。在实际情况中，应用驱动布置方式与布置的顺序可以减小网络的直径。在这种情况下， $nwkMaxDepth$ 为网络直径的下界，而 $2 * nwkMaxDepth$ 为网络直径的上界。

最后，由于树形的事实，当 $nwkAddrAlloc$ 值是0x00时，不能动态的平衡网络树，在一些实际的网络分配中，如按照线性方式分布的网络设备，可能在远远没有到达真正的网络容量时，其网络地址已经全部分配。

在随机地址分配中，没有网络树的建立因此 $nwkMaxDepth$ 和网络跳数有关。在网络中用随机地址分配没有限定的值。

### 3.7.1.8断开网络

本小节介绍两种断开网络流程，即设备自己从网络中离开和父设备请求子设备离开。这两种情况，断开网络的设备的子设备也断开网络。

#### 3.7.1.8.1设备自己从网络中断开的方法

本小节描述设备在响应从高层接收到NLME-LEAVE.request原语如何从网络中断开，如图3.38

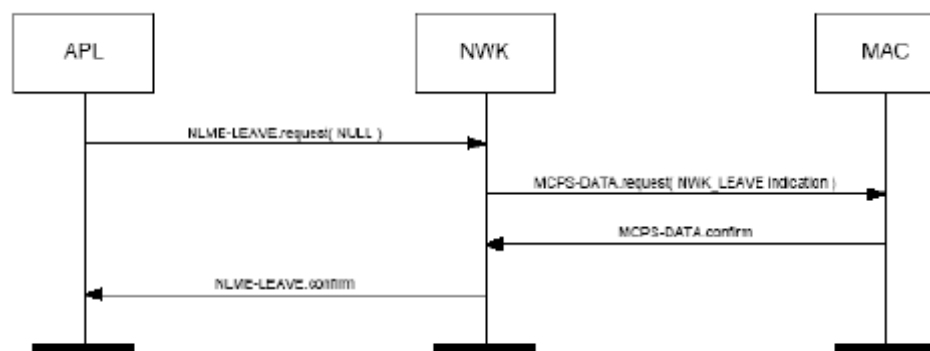


Figure 3.38 Initiation of the Leave Procedure

ZigBee协调器或者路由器的网络层接收到NLME-LEAVE.request原语之后，其DeviceAddress参数等于NULL（表明设备自己断开网络），设备将使用MCPS-DATA.request原语发送断开命令帧，其DstAddr参数设置0xffff表明是MAC广播。断开命令帧的命令选择域的请求子域设置为0。断开命令帧的命令选择域的断开子设备子域的值应反应NLMELEAVE.request原语的RemoveChildren参数值，且断开命令的重连接子域的值反应NLMELEAVE.request原语的Rejoin参数值。在传送断开命令帧之后，它将发送一个NLME-LEAVE.confirm原语给上层，其DeviceAddress参数设置是NULL。如果断开命令帧传送成功，那么状态参数设置为SUCCESS。反之，NLME-LEAVE.confirm原语的状态参数值与the MCPS-DATA.confirm原语返回的状态参数值相同。

如果接收到NLME-LEAVE.request原语的设备是ZigBee终端设备，那么设备将使用MCPSDATA.request原语发送一个断开命令帧，其原语DstAddr参数设置为它父设备的16位网络地址，表明是MAC单播。断开命令帧的命令选择域的请求和断开子设备子域应设置为0。断开命令帧传送之后，它将发送NWK-LEAVE.confirm原语给高层，其DeviceAddress参数设置等于NULL。如果断开命令帧传送成功，那么状态参数设置SUCCESS。反之，NLME-



LEAVE.confirm原语的状态参数与MCPS-DATA.confirm原语返回的状态参数相同。

### 3.7.1.8.2父设备将子设备断开网络的方法

本小节介绍父设备在从高层接收到NLME-LEAVE.request原语后如何让其一个子设备同网络断开。如图3.39所示。

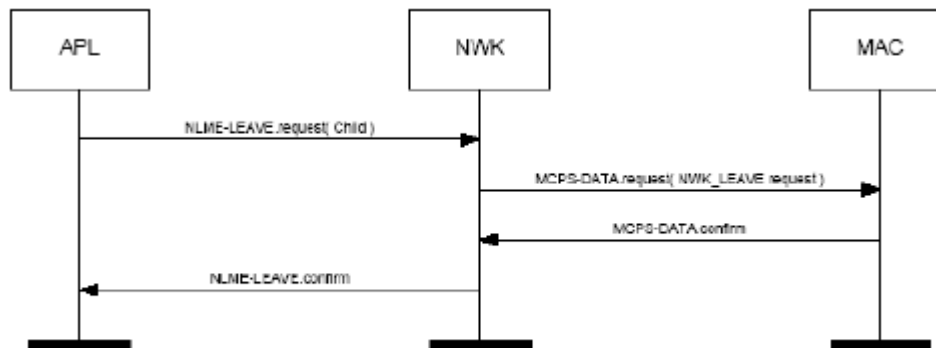


Figure 3.39 Procedure for a Device to Remove Its Child

ZigBee协调器或者路由器的网络层在接收到NLME-LEAVE.request原语之后，其原语参数设置等于子设备的64位IEEE地址，设备将使用MCPS-DATA.request原语发送一个网络断开命令帧，其原语DstAddr参数设置为子设备的16位网络地址。断开命令帧的命令选择域的请求子域设置为1，表明断开网络请求。断开命令帧的命令选择域的断开子设备子域的值反应NLME-LEAVE.request原语的RemoveChildren参数值，且断开命令帧的Rejoin子域的值反应NLME-LEAVE.request的Rejoin参数值。

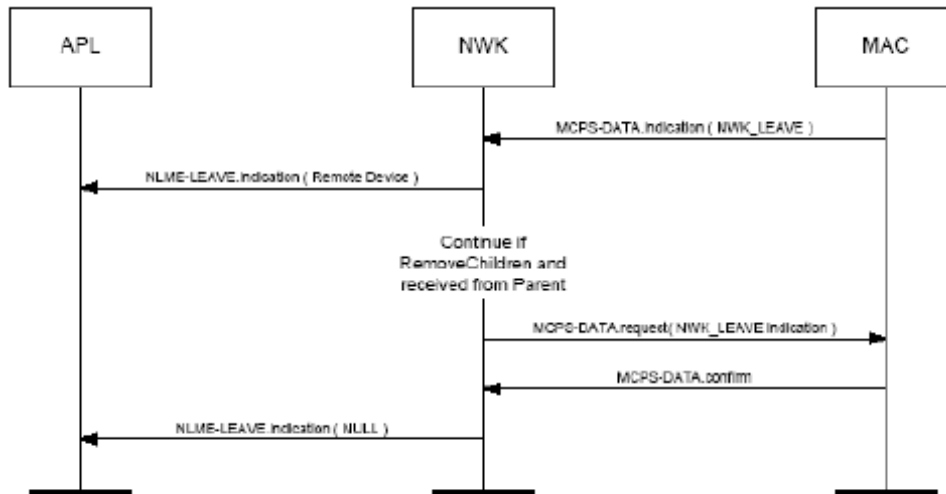
断开命令帧传送之后，如果断开命令帧传送成功，网络层将发送NLME-LEAVE.confirm原语，其DeviceAddress参数设置为正在断开网络的子设备的64位IEEE地址和SUCCESS状态。反之，NLME-LEAVE.confirm状态参数与MCPS-DATA.indication返回的状态参数一致。

在子设备已经断开之后，父设备的网络层将修改它的邻居表，和其他任何和子设备相关的数据结构，来表明设备不再网络中。设备已经断开之后对于高层寻址和传送帧是错误码。

ZigBee终端设备没有子设备来断开且不应该收到带有参数是non-NULL DeviceAddress的NLME-LEAVE.request原语。

### 3.7.1.8.3收到断开命令帧之后

网络层通过MCPSDATA.indication原语收到断开命令帧之后，设备将检查命令帧的命令选择域的请求子域的值。如图3.40所示。如果请求子域是0，那么网络层将发送NLME-LEAVE.indication原语给高，其原语设备地址参数等于断开命令帧的源IEEE地址子域的值。设备将修改它的邻居表和与正在离开网络设备相关的任何其他相关数据结构，来表明它不再在网络中。设备离开网络之后对于高层寻址和传送帧给这个设备是错误码。



**Figure 3.40 On Receipt of a Leave Command**

如果ZigBee协调器的网络一旦收到如上描述的断开命令帧，命令帧传送的MCPS-DATA.indication原语的SrcAddr参数是接收者的父设备的16位网路地址，且命令选择子域的请求子域的值是1或者命令选择域的断开子设备子域的值是1那么接收者将使用MCPS-DATA.request原语发送断开命令帧，其原语的DstAddr的参数设置为0xffff，表明MAC广播。断开命令帧的命令选择域的请求子域的值设置为0。

输出断开命令帧的命令选择域的断开子设备子域的值反应输入断开命令帧的相同域的值。传送断开命令帧之后它将发送一个NLME-LEAVE.indication原语给高层，其DeviceAddress参数设置为NULL。

如果请求子域的值是1，且离开命令帧的源是一个设备而不时接收者的父设备，帧被丢弃。

如果ZigBee终端设备接收一个如上所述的断开命令帧，且传送命令帧的MCPS-DATA.indication原语SrcAddr参数是接收者的父设备的16位网络地址，接收者将发送一个NLME-LEAVE.indication原语给高层，其DeviceAddress参数等于NULL。

网络层也许使用重发技术，如3.7.4描述，开加强断开网络过程的可靠性，但是除了这一点，这些机制是在协议范围外的。

### 3.7.1.9设备复位

设备的网路层将在如下3中情况下对设备进行复位，即（1）在上电后；（2）在企图连接网络前；（3）在企图同网络断开连接后。这个过程在其他任何时间都不执行。设备通过向上层网络管理实体发送NLME-RESET.request原语来执行设备复位，并且通过NLME-RESET.confirm原语返回执行复位的结果信息。复位流程将清楚设备的路由表。

一些设备在ROM存储单元中可能存储了网络层信息，在复位后将恢复这些存储信息。但设备在复位后，将丢弃它的网络地址。这些设备搜索网络连接网络并从它的父设备得到一个网络地址。新网络地址可能与旧的网络地址不同。在这种情况下，任何设备想同复位后的设备进行通信的时候，都必须使用上层协议来重新发现该设备。

### 3.7.1.10管理PANID冲突

由于PANID不是唯一的数字所以PANID有可能冲突。下一节介绍如何通过网络层Report和Update命令帧来更新一个网络的PANID。

#### 3.7.1.10.1检测PANID冲突

网络中任何一个操作的设备接收到MLME-BEACONNOTIFY.indication原语，这里超帧的标识符与他们自己的PAN标识符，但是EPID的值不等于nwkExtendedPANID的值将考虑检

测PAN标识符冲突。

检测到PAN标识符冲突的节点将构造一个PAN标识符冲突的网络Report命令帧。Report信息域将包含一个在本地邻居表中使用的所有16位PAN标识符的列表。在协议范围之外如何建立这个列表，然而从ACTIVE类型的MLME-SCAN.request原语结果构建时推荐使用。

#### 3.7.1.10.2接收到网络Report命令帧之后

被16位网络地址标识的设备，这个地址包含有NIB的*nwkManagerAddr*参数，这个设备将是PAN标识符冲突的网络Report命令帧的接收者。

一旦接收到指定的网络管理将选择为这个网络选择一个新的16位网络标识符。这个新的网络标识符是随机选择的，但是需要检测确认这个选择的网络标识符在本地邻居表中没被使用，且不包含在网络Report命令帧的Report信息域中。

一旦选择了新的PAN标识符，指定的网络管理将构造一个PAN标识符更新的网络Update命令帧。Update信息域应设置为新的PAN标识符的值。

在它发送这个命令帧之后，指定的网络管将启动一个定时器，定时器的值等于*nwkNetworkBroadcastDeliveryTime*秒。当定时器终止，ZigBee协调器将改变它当前的PANID选择一个新的。

传送网络Update命令帧之后，指定的网络管理将建立NLME-ROUTE-ERROR.indication原语，其ShortAddr参数设置是0，且状态参数设置是PAN标识符更新。

#### 3.7.1.10.3接收到网络Update命令帧之后

接收到PAN标识符更新的网络Update命令帧之后，设备启动一个定时器，定时器的值等于*nwkNetworkBroadcastDeliveryTime*秒。当定时器终止时，设备改变它当前的PAN标识符为包含在Update信息域的值。

在网络Update命令帧传送之后，设备将创建一个带有ShortAddr参数设置为0，和PAN标识符Update的状态参数的NLME-ROUTE-ERROR.indication原语。

### 3.7.2发送和接收

#### 3.7.2.1发送

仅仅只有已经与网络连接的设备可从网络层发送数据帧。如果未连接设备收到传输帧的请求命令，将丢弃该帧，并向其上层发送状态为INVALID\_REQUEST的NLDE-DATA.confirm原语，通报其错误状态。

在网络层传送或生成的所有帧，都将根据图3.3的通用帧结构进行构造，并采用MAC层数据服务进行传送。

另外源地址域和目的地址域，所有网络层传输帧都包含一个半径域和序列号域。对于高层数据帧的初始化，使用NLDE-DATA.request原语的Radius参数值提供的半径域的值。如果没有该值，那么网络层头的半径域将设置成NIB中的*nwkMaxDepth*参数值的二倍。每个设备的网络层都保持一个序列号，这个序列号是随机值。每一次网络层构建一个新的网络层帧，和从高层传输一个新的网络帧的求情结果，或者是当它需要一个新的网络层命令帧时，序列号加1，增加完之后，序列号的值将插入到帧的头的序列号域。

当构造好网络协议数据单元后，如果对该帧需要进行安全处理，则将根据安全方案对它进行安全处理。如果NLDE-DATA.request的安全允许参数SecurityEnable等于FALSE，则不需要安全处理。如果网络层安全级别参数*nwkSecurityLevel*等于0，或者如果在高层帧经过初始化且*nwkSecureAllFrames* NIB属性设置为0x00，那么帧控制域的安全子域总是设置成0。

当安全处理成功后，将返回该帧，并由网络层进行传输。经处理的帧将附加一个校验帧头。如果帧的安全处理失败，并且此帧为数据帧，则将通过NLDE-DATA.confirm原语的状态向上层进行通报。如果帧的安全处理失败，且此帧为网络命令帧，则将丢弃该帧，不进行进一步处理。

当构造好一个帧，并且已准备好传输该帧时，通过向MAC层发送MCPSDATA.request原语请求发送网络层协议数据单元，将该帧传送到MAC数据服务单元，其传送的结果将通过MCPS-DATA.confirm原语返回。

### 3.7.2.2接受和拒绝

为了接受数据，设备必须打开其接收机。上层使用NLME-SYNC.request原语初始化设备，打开其接收机。NLME-SYNC.request原语将会引起网络层使用MLME-POLL.request原语对其父设备进行轮询。

ZigBee协调器或者路由器的网络层必须在最大程度上保证无论什么时候接收机总是处于接受状态。

一旦接收机处于接受状态，网络层将通过MAC数据服务来接受数据帧。每一帧在接收之后，网络层头的半径域减1。如果值减到0，任何情况下，该帧都不能转发。然而，它可能传输到高层或者作为协议的列出的其他地方的网络层处理。使用NLDE-DATA.indication原语把如下叙述的数据帧传送到高层：

- (1) 有广播地址的帧，此广播地址匹配一个广播组，设备是这个广播组的成员。
- (2) 目的地址匹配网络地址的单播数据帧和源地址数据帧。
- (3) 多播数据帧，它的组ID是在*nwkGroupIDTable*列出来的。

如果接收机是ZigBee协调器或者是正在操作的路由器，也就是，路由器已经调用NLME-START-ROUTER.request原语，它将按如下步骤处理数据帧：

- (1) 根据3.7.4和3.7.5节列出的过程来转播广播和多播数据帧。
- (2) 有目的地址的单播数据帧，目的地址和设备的网络地址不匹配，将根据3.7.3节列出的过程来转发该帧。（在任何其他情况下，单播数据帧应立刻丢弃）
- (3) 有目的地址的源路由数据帧，目的地址和设备的网络地址不匹配，将根据3.7.3.3.2来转发该帧。
- (4) 处理路由请求命令帧的过程是根据3.7.3.5.2节。

处理目的地址与设备网络地址匹配的路由转发命令帧是根据3.7.3.5.3节。

- (5) 目的地址与设备网络地址不匹配的路由转发命令帧被丢弃。路由错误命令帧和数据帧的处理方法相同。

网络层将使用NLDE-DATA.indication原语向其高层表明所接收到的数据帧。

一旦接收到帧信息，网络层数据实体将会检查帧控制域中的安全子域的值。如果该值不为0，则网络层数据实体将把该帧传送到安全服务提供单元，并根据所指定的安全标准对其进行安全处理。如果安全子域设置为0，那么NIB中的*nwkSecurityLevel*属性不为0，且输入帧是网络层命令帧，则网络层数据实体丢弃该帧。如果安全子域设置为0，那么NIB中的*nwkSecurityLevel*属性不为0，且输入帧是网络层数据帧，网络层数据实体将检查*nwkSecureAllFrames* NIB属性值。如果属性值设置为0x01，网络层数据实体将只接收帧如果是发给自己也就是说它不需要转发给其他设备。

### 3.7.3 路由选择

ZigBee 协调器和路由器应提供如下的功能：

- (1) 为上层中继数据帧
- (2) 为其他 ZigBee 路由器中继数据帧
- (3) 参与路由发现，为后续数据帧建立路由
- (4) 为终端设备参与路由选择
- (5) 参与路由修复
- (6) 在路由发现中，使用所规范的 ZigBee 路由成本进行度量

ZigBee 协调器活路由器还可能提供如下功能：

- (1) 为记录下最佳的有效路由，维护路由表
- (2) 为上层初始化路由选择
- (3) 为其他的 ZigBee 路由器，初始化路由选择
- (4) 初始化路由修复

### 3.7.3.1 路由成本

在路由选择和维护时，ZigBee 的路由算法使用了路由成本的度量方法来比较路由的好坏。成本，即众所周知的链路成本，与路由中的每一条链路相关，组成路由的链路成本之和为路由成本。

假定一个长度为 L 的路由 P，由一系列设备  $[D_1, D_2, \dots, D_L]$  组成，每一个链路为长度为 2 的子路由  $[D_i, D_{i+1}]$  组成，则路由 P 的成本为

$$C\{P\} = \sum_{i=1}^{L-1} C\{[D_i, D_{i+1}]\}$$

其中：C{[  $D_i, D_{i+1}$ ]} 为链路成本。链路成本 {[1]} 为链路 1 的函数，且其值为集合 [0, ..., 7]，函数的表达式为

$$C\{1\} = \begin{cases} 7 \\ \min\left(7, \text{round}\left(\frac{1}{p_l^4}\right)\right) \end{cases}$$

其中， $p_l$  为链路 1 中发送数据包的概率。

因而，链路成本为常数 7，或者与链路接受概率  $p_l$  相关的值，即为接受概率  $p_l$  的倒数，该数为每次使用该链路预期从该链路得到数据包的请求次数。设备利用网络层信息库的 `nwkReportConstantCost` 属性设置为 TRUE 的方法，强迫设备报告链路成本。

然而，主要问题是怎样测量或估计  $p_l$ 。 $p_l$  可通过实际计算收到的信标和数据帧来进行估计，即通过观察帧的相应序列号来检测丢失的帧，这就通常被认为最准确地测量接受概率的方法。但是，对于所有的方法来说，最直接和最有效的方法就是基于 IEEE 802.15.4 的 MAC 层和 PHY 层所提供的每一帧的 LQI 通过平均所计算的值。即使使用其他的方法，最初的成本估计值也是基于平均的 LQI 值。常常使用驱动函数表来映射平均 LQI 值于 C{1} 值的关系。在实际应用中，应注意实际的硬件对驱动函数表的影响，不准确的成本对 ZigBee 的路由算法将造成影响。

### 3.7.3.2 路由表

ZigBee 路由器和协调器可对路由表进行维护。存储在路由表中的信息如表 3.48 所示。旧的和退休的路由表入口想要在命令操作中重新收回表空间入口的操作不在协议的规定范围内。

**Table 3.48 Routing Table**

Field Name	Size	Description
Destination address	2 bytes	The 16-bit network address or Group ID of this route; If the destination device is a ZigBee router or ZigBee coordinator, this field shall contain the actual 16-bit address of that device; If the destination device is an end device, this field shall contain the 16-bit network address of that device's parent
Status	3 bits	The status of the route. See Table 3.49 for values
Memory Constrained	1 bit	A flag indicating that the device is a memory constrained concentrator
Many-to-one	1 bit	A flag indicating that the destination is a concentrator that issued a many-to-one route request
Route record required	1 bit	A flag indicating that a route record command frame should be sent to the destination prior to the next data packet
GroupID flag	1 bit	A flag indicating that the destination address is a Group ID
Next-hop address	2 bytes	The 16-bit network address of the next hop on the way to the destination

表3. 49枚举了路由状态所对应的值。

**Table 3.49 Route Status Values**

Numeric Value	Status
0x0	ACTIVE
0x1	DISCOVERY_UNDERWAY
0x2	DISCOVERY_FAILED
0x3	INACTIVE
0x4	VALIDATION_UNDERWAY
0x5 – 0x7	Reserved

这部分描述了路由算法。“路由表能力”常用来描述一个设备使用自身的路由表来建立一条到达指定目的地址设备的路由能力。如果设备满足以下条件，则设备具有路由表能力：

- (1) 为ZigBee路由器或协调器
- (2) 具有路由表维护能力
- (3) 具有一个空闲的路由表入口，或者已经具有一个到目的地的路由表入口

如果ZigBee路由器或协调器维护一个路由表，则它同样应维护一个路由选择表，该表所包含的信息如表3. 50所示。路由表入口是长期存在的和不变的，而路由选择表的入口仅仅是在路由选择的过程中存在，并且可重新生成。



**Table 3.50 Route Discovery Table**

Field Name	Size	Description
Route request ID	1 byte	A sequence number for a route request command frame that is incremented each time a device initiates a route request
Source address	2 bytes	The 16-bit network address of the route request's initiator
Sender address	2 bytes	The 16-bit network address of the device that has sent the most recent lowest cost route request command frame corresponding to this entry's Route request identifier and Source address; This field is used to determine the path that an eventual route reply command frame should follow
Forward Cost	1 byte	The accumulated path cost from the source of the route request to the current device
Residual cost	1 byte	The accumulated path cost from the current device to the destination device
Expiration time	2 bytes	A countdown timer indicating the number of milliseconds until route discovery expires; The initial value is <i>mwkcRouteDiscoveryTime</i>

如果设备满足以下两个条件，则设备具有路由选择表能力：

- (1) 具有维护路由选择表能力
- (2) 在它的路由选择表中，具有一个空闲的入口

如果一个设备既具有路由表能力又具有路由选择表能力，则称该设备具有“路由能力”。

如果一个设备有发起多对一路由请求的能力，那么它还应该拥有一个源路由表。

### 3.7.3.3 接收到数据帧

网络层接收到数据后，不管是从MAC层或者是从高层接收到的，将按下列的流程发送该数据帧。

如果接收设备为ZigBee协调器或者路由器，并且帧的目的地址为该设备的终端子设备，则发送MSDE-DATA.request原语，如3.7.2.1所述。将该数据帧直接发送到目的地址设备，并且将下一跳的目的地址设置为最终目的地址。否则，为了后续的讨论，如果他是一个路由器或协调器，则定义设备的路由地址为它的短地址，或者终端设备父节点的短地址。定义一个帧的路由地址为网络层定义的帧的路由地址。注意，ZigBee地址的分配是得设备可以从自己的地址中得好路由地址。详见3.7.1.5。

ZogBee路由器或协调器可以通过邻居表检测帧的路由目的地址的响应入口。如果有相对应的入口，设备就可以通过MCPS-DATA.request原语直接按照路由发送该帧，详见3.7.2.1小节。

具有路由能力的设备首先检查与目的地址相对应的路由表入口。如果存在该入口，并且如果该入口路由状态域的值为ACTIVE或VALIDATION\_UNDERWAY, 设备将使用MCPS\_DATA.request原语转发该帧，如果路由状态域仍没有值则将其设为ACTIVE。

如果路由表入口的路由记录请求域设置为真，或者目的地没有被记录，或者被中继的帧是由本地生成的，或者是由终端子设备产生的，那么设备可以根据3.7.3.5.4小节的描述向目的地址初始化一个路由记录命令帧。

当转发一个数据帧时，MCPS-DATA.request原语的SrcAddrMode 和 DstAddrMode参数都要设置为0x02，表明使用16位地址。SrcPANId 和 DstPANId参数都应设置为转发设备的MAC PIB中的macPANId属性。参数SrcAddr设置为转发设备MAC PIB 的 macShortAddress ，并且DstAddr参数应设置为路由表入口中相对于路由目的地址的下一跳地址。TxOptions参数应设置为与0x01按位与为非零的值，表明确认传输。如果设备有一个与帧的路由目的地址相对应



的路由表入口，但是入口的路由状态值为DISCOVERY\_UNDERWAY，则应按照3.7.3.5.1小节所描述的，该帧应该按照路由发现进行初始化。那么，该帧应该放到路由未决缓冲器，如果NIB属性中的nwUseTreeRouting参数为真，则按照树型分级路由。如果帧沿树型路由，那么网络层帧头控制域中的路由发现域应设置为0x00。

如果设备对应于路由目的地址有一个相对应的路由表入口，但是入口的路由状态域的值为DISCOVERY\_FAILED 或 INACTIVE，如果NIB属性的nwUseTreeRouting值为真，那么设备将沿着树使用等级路由，如果设备没有与路由目的地址相对应的值为ACTIVE的路由表入口，并且帧是从上层接收到的，它将与路由目的地址相对应的源路由表入口，如果存在这样的入口，设备将按照3.7.3.3.1小节所描述的使用源路由传送该帧。如果设备没有对应于路由目的地址的路由表入口，并且帧不是使用源路由产生，它将检测网络层帧头的控制域的路由发现子域，如果路由发现子域的值0x01，设备将初始路由发现，详见3.7.3.5.1小节。如果路由发现子域的值0，NIB属性的nwUseTreeRouting值为真，那么设备沿树使用分级路由。如果路由发现子域的值0，NIB属性的nwUseTreeRouting值为假，没有与路由目的地址向对应的路由表入口，则帧将被丢弃，NLDE将发送状态为ROUTE\_ERROR的NLDE-DATA.confirm原语。

没有路由能力的设备，如果它的NIB 属性中的 nwUseTreeRouting为真，那么人它将沿树使用分级路由。

对应分级路由，如果目的地是该设备的子设备，设备直接将帧给适合的子设备。如果目的地是子设备，并且还是终端设备，那么可能由于子设备的macRxOnWhenIdle状态使得传送失败。如果子设备的macRxOnWhenIdle值为假，则使用IEEE 802.15.4-2003[B1]中多描述的方法间接传送。如果目的地址不是子节点，设备将按照父节点进行路由。

网络中的其他设备都是ZigBee协调器的子设备，在网络中不存在ZigBee终端设备的子设备。对于一个地址为A，深度为d的ZigBee路由器，如果下述表达式成立，则具有地址为D的目的地址设备为子设备。

$$A < D < A + Cskip(d-1)$$

其中Cskip的定义见3.7.1.5小节。

如果确定目的地址为接收设备的子设备，则下一跳设备的地址N为

$$N = D$$

对于ZigBee终端设备来说， $D > A + Rm \cdot Cskip(d)$ ，否则

$$N = A + 1 + \left\lceil \frac{D - (A + 1)}{Cskip(d)} \right\rceil \cdot Cskip(d)$$

如果ZigBee路由器或ZigBee协调器的网络层发送单播或多播数据帧因为某种原因失败后，路由器或协调器要发送失败信息。可以以两种形式报告，如果转发失败是因为上层请求失败，则网络层向上层发送NLME-ROUTEERROR.indication原语。其中原语的16位短地址参数应为帧的目的地址。如果帧是利用另一个设备转发，那么转发设备应该发送NLME-ROUTE-ERROR.indication原语，并且发送路由错误帧给数据帧的源地址。路由错误命令帧的目的地址域应为发送失败数据帧的目的地址。

在任何一种情况下，失败的原因按照表3.40所列的情况。

### 3.7.3.3.1 产生源路由数据帧

从上一层接收到数据帧，在3.7.3.3小节中描述的情况下，设备将使用源路由机制发送数据帧。

可以通过源路由表得到源路由。

如果没有中间的转发节点，使用DstAddr 参数值标示路由目的地址的 MCPS-DATA.request 原语，不用源路由，直接发送到路由目的地址。

如果有至少一个转发节点，应该设置网络层头帧控制域的源路由标志，并且要有网络层源路由域。源路由中的转发计数器的值英语转发表中的值相等。转发序号的值应该等于转发数值减1。转发表应该至少包含从最低位开始的转发地址。最首先列出最靠近目的地址的地址。源地址列在最后。

设备使用MCPS-DATA.request原语转发数据帧。DstAddr参数应设置为转发表中的转发最终地址。

#### 3.7.3.3.2 转发一个源路由数据帧

转发一个从MAC层接收到的源路由数据帧，详见3.7.3.3小节，设备寻找网络层头源路由中的转发表作为它的短地址。如果没找到短地址，或者如果转发表中的索引与转发索引值不一致，那么帧被丢弃，不再做任何处理。

如果转发索引值为0，设备将使用MCPS-DATA.request原语将数据帧直接转发给网络层头目的地址。

如果转发索引的值不是0，则将转发索引减1，并且立即将数据帧转发到在转发表中先于自己地址的地址。

#### 3.7.3.4 链路状态信息

无线的连接可以是不对称的，也就是说，它们在一个方向可能工作良好，但向另一个方向却不然。因此，在路由请求中的链路发现返回应答时会发生错误。

对于多对一路由和双向的路由发现(nwkSymLink = TRUE)，要求双向的路由发现都是可靠的。为达到这个目的，路由器要周期性的和它们的邻居进行单跳广播传输用来传输链路状态帧以交换计算的链路成本。然后在路由发现中用得到的链路成本来确保路由发现在双向中都使用高的链路质量。

##### 3.7.3.4.1 初始化链路状态命令帧

当一个ZigBee路由器或协调器加入到网络中后，它将周期性的每隔wkLinkStatusPeriod秒发送一个单跳的广播帧，用来发送链路状态。如果要求的话可以更频繁的发送。可以加入随机抖动值来避免与其它节点同步。链路状态命令帧的格式见3.5.8小节。终端设备不发送链路状态命令帧。

##### 3.7.3.4.2 接收到链路状态命令帧

ZigBee路由器或协调器接收到链路状态命令帧后，对应于传输设备的邻居表入口的age域将被置为0。帧所覆盖的地址范围由链路状态表中的首地址和末地址和命令选项域的首帧和末帧决定。如果接收设备的网络地址超出了帧覆盖的范围，那么该帧将被丢弃，该过程终止。如果接收设备的网络地址在帧覆盖的范围内，那么收集链路状态表。如果找到了接收设备的地址，邻居表中对应于发送设备的输出成本域设置为链路状态入口的输入成本值。如果没有找到接收设备的地址，输出成本域设为0。

终端设备不处理链路状态命令帧。

##### 3.7.3.4.3 完善邻居表

邻居表中的age域每隔时间nwkLinkStatusPeriod增加一次。如果该值超出了参数值nwkRouterAgeLimit，邻居表入口的输出成本域置为0。也就是说，如果设备从邻居路由器连续接收nwkRouterAgeLimit链路状态消息失败，旧的输出链路成本将被丢弃。在这种情况下，邻居表入口将被认为时陈旧的，如果由新的邻居，则该入口将被重新使用。

#### 3.7.3.5 路由发现

路由发现是在网络中的设备互相合作条件下选择，并建立路由的一个流程，该流程通常与特定的源地址和目的地址相对应。多点传送路由是执行关于一个特殊的源设备和多点传送组的路由。多对一路由发现是一个源设备与所有的ZigBee路由器和协调器在radius范围内与自身建立路由的过程。根据3.7.3.5小节的描述，目的地址可以是16位的广播地址，或是一

个设备的16位网络地址，或作为多点传送组ID的16位多点传送地址。如果一个路由请求命令的目的地址是一个特殊设备的路由地址，它的路由请求选项域没有设置多点传送域，则请求为一个单播路由请求。一个路由请求命令，它的路由请求选项域有多点传送位，则设置为多点传送路由请求。多点传送路由请求的目的地址域应设置为多点传送组的ID。一个目的地址域为广播地址（见表）的路由请求命令的载荷应为多对一路由请求。多对一路由请求的路由请求选项的多点传送位应置为0。

### 3.7.3.5.1路由发现的初始化

在网络层收到其上层发送来的NLDE-ROUTEDISCOVERY.request原语，其中DstAddrMode参数值为0x00，在路由表中没有与目的地址DstAddr相对应的入口，或在MAC层接收到的帧控制域中的路由搜索子域值为0x01，或在网络层帧报头中的目的地址不是当前设备地址或者为广播地址，或没有与网络层帧报头中的目的地址相对应的路由选择表入口。在其他情况下，如果设备没有路由选择能力，并且NIB属性中nwkUseTreeRouting的值为真，则有问题的数据帧将按照分级的路由方法，沿树搜索路由。如果设备没有路由能力，并且NIB属性中nwkUseTreeRouting值为假，那么该帧将被丢弃，网络层按照3.7.3.7小节中描述的产生NLME-ROUTEERROR.indication原语或路由错误命令帧。

多点传送的路由发现过程可以通过网络层初始化，或者通过从高层接收到NLME-ROUTEDISCOVERY.request原语，其中DesAddrMode参数值为0x01，或者通过下面3.7.5.2.2小节所描述的过程完成。

如果设备具有路由选择能力，则建立一个路由选择表入口和路由搜索表入口，并且该入口的状态设置为DISCOVERY-UNDERWAY。如果已经存在一个与目的地址相对应的路由选择表入口，并且状态值为ACTIVE或VALIDATION\_UNDERWAY，那么这个入口可以被使用，并且入口的状态域保存为当前值。如果存在路由表入口，但状态值不是ACTIVE，那么入口可以被使用，并且入口的状态值设置为DISCOVERY\_UNDERWAY。如果不存在入口，那么设备将建立一个相对应的路由发现表入口。

每一个发送路由请求命令帧的设备维护一个生成路由请求标识符的计数器。当生成一个新的路由请求命令帧时，该计数器加1，并且将该值保存在设备路由搜索表的路由请求标识符中。路由选择表和路由搜索表中的其他域将根据3.7.3.2小节进行设置。

网络层缓存所接收到的待处理路由搜索帧，或者，如果为单播帧并且NIB树型中的nwkUseTreeRouting值为真，将网络层帧报头中帧控制域的路由选择子域设置为0，然后沿树向前发送数据帧。

一旦设备创建了路由搜索表和路由选择表入口，则将按照图3.10所示的节后创建哟个载有有效载荷的路由请求命令帧，帧中的各子域的设置如下所述：

- (1) 命令帧标识符域设置为路由请求帧，详见表3.39
- (2) 路由请求标识符设置为路由选择表入口
- (3) 多点传送标志位和目的地址域应该设置为与要被发现的地址向一致
- (4) 路由成本域设为0

在构造完成所广播的路由搜索命令帧后，网络层使用MCPS-DATA.request原语将其传送到MAC层。

在路由搜索的开始阶段，广播一个路由命令请求帧时，网络层在开始广播后，将重播nwkInitialRREQRetries次，因此，最大的广播次数为nwkInitialRREQRetries+1次，每次重播的时间间隔为nwkRREQRetryInterval毫秒。

ZigBee路由器或协调器的网络层管理实体在接收到上层发送的DstAddrMode 参数值为0x00的NLME-ROUTE-DISCOVERY.request原语后，初始化多对一路由发现。在这种情况下，设备不是必须建立路由表入口。如果路由请求命令帧的目的地址参数为0xffff9，那么就如该小

节描述的建立并广播。

设备不是必须建立发现表或路由表入口。一个路由请求命令帧可以如3.7.3.5.1小节所示建立并广播。NLME-ROUTE-DISCOVERY.request原语的参数MemoryConstrained为真，那么路由请求命令帧的目的地址域置为0xfff8。否则，置为0xfff9。

#### 3.7.3.5.2 接收到路由请求命令帧

在接收到路由请求命令帧后，如果设备是一个终端设备，那么丢弃该帧。否则，判断是否有路由能力。

如果设备没有路由能力，并且是多点传送路由请求或多对一路由请求，那么路由请求将被丢弃并且路由请求处理被终止。

如果设备没有路由选择能力，并且路由请求时单播路由请求，则将判断所接收的帧是否来自于有效的路由。所谓有效路由是指所接收的帧来自于设备的子设备，并且全设备为孩子设备的后裔设备；或者来自于设备的父设备，并且，源设备不是设备的子设备。如果路由请求命令帧不是来自于有效路由，则将丢弃该帧。否则，将检查设备是否为预期的目的设备。同样，通过路由请求命令帧有效载荷中的目的地址与它的每一个终端子设备地址比较，检查命令帧的目的地址时都为设备的一个终端子设备。如果路由请求命令帧的目的地址为设备本身或者为设备的一个子设备，则它将用一个路由请求应答命令帧进行应答。当设备用一个路由请求应答命令帧应答路由请求时，将构造一个类型域为0x01的帧。路由请求应答的源地址应设置为创建请求应答设备的16位网络地址，在考虑到路由请求的发起者为最终目的地址的情况下，将帧的目的地址域设置为所计算出来的下一跳的地址。计算下一跳设备到挡墙设备的链路成本，其计算方法详见3.7.3.1小节，并将该成本附着在路由应答帧的路由成本域中。通过发送MCPS-DATA.request原语，将路由应答命令帧单播到下一跳设备。

如果设备不是路由请求命令帧的目的地址，则计算从前一个设备到本设备传送该帧的链路成本，其方法如3.7.3.1节中所述。其成本数值将加到路由请求命令帧的路由成本值中，然后，使用MCPS-DATA.request服务原语，向目的地址单播该路由请求命令帧。同数据帧一样，采用有效载荷中的目的地址域标识来判断设备地址的方法，决定单播传输的下一跳地址。

如果设备没有路由能力，接收的请求是一个单播的路由请求，设备将路由请求命令帧载荷的目的地址与自身的地址比较来判断该设备是否是目的地址。它还将路由请求命令帧载荷的目的地址与它的终端子设备地址进行比较。判断命令帧的目的地址是否是自己的终端子设备。如果该设备或设备的终端子节点是路由请求命令帧的目的节点，设备将决定在路由发现表（见表3.50）入口中是否存在相同的路由请求标识符和源地址。如果没有相对应的入口存在，将建立一个入口。

如果设备没有路由能力，并且接收到的路由请求帧的路由请求命令选项域标示为多点传送路由请求，设备将判断在nwkGroupIDTable中是否已经存在其GroupID域与目的地址相配的入口。如果相配的入口存在，设备将判断是否存在一个有相同路由请求标识符和源地址的路由发现表（见表3.50）入口。

对于多对一路由请求，和常规的路由请求，如果参数nwkSymLink的值为TRUE，接收到一个路由请求命令帧后，设备将在邻居表中寻找对应于传输设备的入口。如果没有对应的入口或该入口的输出成本域的值为0，则丢弃该帧，路由请求过程终止。邻居设备的最大输出和输入成本时是用来计算路径成本而不是输入成本。这包括增加重发前一个路由请求帧的路径成本。

当建立了路由发现表入口，则其值设置为与路由请求命令帧相对应的值。唯一例外的是前向成本域，该域是利用前面的发送者的命令帧计算的链路成本，详见3.7.3.1小节。将它加到路由请求命令帧的路径成本中。上述计算的结果保存在新建立的路由发现表入口的前向成本域。



如果nwkSymLink属性值为真，设备将建立一个路由表入口，其目的地址域为路由请求命令帧的源地址，下一跳设置为传输命令帧的前一个设备的地址。状态域置为ACTIVE。然后设备向路由请求命令帧的源地址发送路由应答命令帧。如果设备对于源地址和路由请求标识符对已经有一个路由发现表入口，设备判断在路由请求命令帧中的路径成本是否小于存储在路由发现表入口中的前向成本。该比较通过计算发送该帧的前向设备的链路成本，如3.7.3.1小节所述，加上路由请求命令帧的路径成本。如果该值比路由发现表入口的值大，将丢弃该帧，没有后续的处理。否则，路由发现表中的前向成本和发送地址域将被更新为路由请求命令帧的新的成本和前向设备地址。

如果属性nwkSymLink的值为TRUE，接收到的路由请求命令帧是一个单播路由请求，设备仍将建立一个路由表入口，其目的地址设置为路由请求命令帧的源地址，下一跳域设置为传输命令帧的前一个设备的地址。状态域设置为ACTIVE。然后设备将响应一个路由回复命令帧。在任何一种情况下，如果设备是代表它的一个终端子设备响应，那么路由回复命令帧载荷的响应地址应与终端子设备的地址一致，而不是响应设备的地址。

当一个具有路由选择能力的设备不是接收到的路由请求命令帧的目的设备时，则判断在路由选择表（见表3.50）中是否存在一个有相同的路由请求标识符和源地址域入口。如果入口不存在，则创建一个入口。路由请求定时器终止时间设置为nwkcRouteDiscoveryTime毫秒。如果相对应与目的地址的路由地址的路由表入口存在，并且其状态值不为ACTIVE，那么将其设置为DISCOVERY\_UNDERWAY。如果不存在这样的入口，并且该帧为单播路由请求，将建立一个入口，其状态值为DISCOVERY\_UNDERWAY。如果nwkSymLink属性为TRUE或者是多对一路由请求帧，则设备也将建立一个路由表入口，并且它的目的地址设置为路由请求命令帧的源地址，下一跳的地址设置为上一个传送该命令帧设备的网络地址。如果是多对一的路由请求帧，则多对一域和路由表入口应设置为真，如果目的地址域是0xffff8则MemoryConstrained标志位设为真否则设为假；如果下一跳域改变了，路由记录要求域则设为真。状态域设置为ACTIVE。当路由请求定时器终止时，设备将从路由选择表中删除该路由请求入口。在这种情况下，如果它的状态域的值DISCOVERY\_UNDERWAY并且为目的地址在路由发现表中没有其他的入口，则与路由表入口对应的目的路由地址也要被删除。如果在路由选择表中存在一个入口，则路由请求命令帧中的路由成本将与路由选择表入口的前向成本相比较，比较是通过计算先前设备的链路成本，详见3.7.3.1，加上路由请求帧中的路由成本。如果路由成本更大，则丢弃该路由强求命令帧，不对其进行处理；否则，路由选择表中的前向成本和发送者地址域将更新为路由请求命令帧中的新的成本和上一个发送该帧的设备地址。此外，路由请求命令帧中的路由成本域的值新的计算结果。如果nwkSymLink属性为TRUE，则设备也将更新它的路由表入口，目的地址将更新为路由请求命令帧的源地址，下一跳的地址将更新为上一个传送该命令帧的设备网络地址。状态域设置为ACTIVE，然后，设备将使用MCPS\_DATA.request原语，重新广播该路由请求命令帧。

当重新广播路由请求命令帧时，网络层将使用下面的公式，用一个随机不稳定值计算出重传的时延。

$$2 \cdot R[\text{nwkcMinRREQJitter}, \text{nwkcMaxRREQJitter}]$$

其中R[a, b]是在[a, b]参数区间的随机函数，不稳定值的单位为毫秒。设备可调整这个不稳定值，以使接收到的路由成本大的路由请求命令帧比路由成本小的延时更大。网络层将在初始中继后重试广播nwkcRREQRetries次，以至于每次中继为最大的nwkcRREQRetries+1次。当相当的源和路由请求标识符的帧比等候重传帧所花费路由成本小时，设备可能丢弃等待重传的路由选择命令帧。

设备根据有效载荷中的目的地址，将相对应的路由表入口的状态域设为DISCOVERY-UNDERWAY。如果不存在这样的入口，则重新建立一个入口。

当对一个路由请求帧进行应答时，具有与路由请求的源地址、路由请求标识符相对应的路由选择表的入口的设备将构建一个帧类型为0x01的命令帧。网络头的源地址域设置为当前设备的16位网络地址，目的地址域设置为对应路由选择表入口中的发送者地址域。构造路由应答的设备将按照下述方法在组成载荷域。网络命令标识符设置为路由应答，路由请求标识符域的值设置为与路由请求命令帧的路由请求标识符域中的值相同，起始者域设置为路由请求命令帧中的网络头中的源地址。利用路由请求命令帧中的网络帧报头中的源地址与它相对应的路由选择表入口的起始者地址，并根据3.7.3.1小节中描述的来计算链路成本。该链路成本设置在路由成本域中，然后，利用MCPS-DATA.request原语，将路由请求应答帧单播到目的地址，从路由选择表中所得到的发送地址作为下一跳地址。

#### 3.7.3.5.3接收到路由应答命令帧

一旦设备接收到路由应答命令帧，将按照下述所描述的流程，对应答帧进行处理。

如果接收设备不具有路由选择能力，它的NIB属性中的nwkUseTreeRouting的值为真，则利用树型路由转发路由应答。如果接收设备不具有路由选择能力，并且它的NIB属性中的nwkUseTreeRouting的值为假，则丢弃该命令帧。在转发路由应答命令帧之前，根据3.7.3.1小节中所介绍的方法，计算从下一跳设备到它本身的链路成本，将该链路成本与载荷中的路由成本域中的值相加，并将其结果更新载荷中的路由成本域，得到新的路由成本。

如果接受设备具有路由能力，则将设备地址同路由应答命令帧载荷的始发者地址域的内容比较，来判断设备是否为路由应答命令帧的目的设备。如果是该应答命令帧的目的设备，则在路由搜索表中搜索与路由应答命令帧载荷中的路由请求标识符相对应的入口。如果不存在这样的入口，将丢弃路由应答命令帧，并终止对路由应答帧的处理。如果存在有这样的路由选择表入口，则设备将在路由选择表搜索一个与路由应答命令帧中响应地址相对应的入口。如果不存在这样的路由选择表入口，则丢弃路由应答命令帧，且即使对应于路由应答命令帧中的路由请求标识符的路由搜索表入口存在，也要删除该应答帧，并且，终止对路由应答帧的处理流程。如果路由选择表入口和路由搜索表入口都存在，且路由选择表入口的状态域为DISCOVERY-UNDEERWAY，如果路由表入口的GroupId标志位为真，则需要改为ACTIVE，并且路由选择表中的下一跳域应设置为向前发送路由应答命令帧的前一个设备的地址。将路由搜索表入口中的剩余的成本域值设置到路由应答载荷的路由成本域中。

如果状态域已经设置为ACTIVE，则设备对路由应答命令帧中的路由成本域路由搜索表入口中的剩余路由成本进行比较，并且如果路由应答命令帧中的成本更小，则更新在路由选择表中的剩余路由成本域和下一跳域。如果路由应答命令帧中的成本更大或相同，则丢弃该路由应答帧，部队该帧进行处理。

如果接收到路由应答的设备不是目的地址设备，则设备搜索与路由应答命令帧载荷中的始发者的地址和路由请求标识符相对应的路由搜索表入口。如果不存在这样的路由搜索表入口，则将丢弃该路由应答命令帧。如果存在这样的路由搜索表入口，则对路由应答命令帧中的路由成本与路由搜索表入口中的剩余路由成本进行比较。如果路由搜索选择表入口的值更小，则将丢弃路由应答命令帧。否则，设备将搜索与路由应答命令帧中的发送者地址相对应的路由选择表入口。如果路由搜索表入口存在，但没有相对应的路由选择表入口，此时就是产生一个错误，则应丢弃该路由应答命令帧。利用向前发送该路由应答帧设备地址替代下一跳地址的方法，对路由选择表入口进行更新。同时，利用路由应答命令帧中的成本替代剩余成本的方法，更新路由搜索表入口。

当接收的路由应答使得响应路由表入口的下一跳地址更改，路由表入口的GroupId标志为TRUE，如果设备是路由应答帧的目的地址，那么设备将响应的路由发现表入口的expiration time域设置为nwkWaitBeforeValidation毫秒，如果不是，则设置为nwkRouteDiscoveryTime毫秒。

在更新本身的路由入口后,设备将向目的地址发送路由应答。在向前发送路由应答帧前,需要更新路由成本。发送者通过在路由搜索表中搜索与路由请求标识符、源地址以及所提取的发送者地址相应入口的方法,找到下一跳到路由应答的目的地址。利用下一跳地址,根据3.7.3.1小节的计算方法,计算链路成本,并将该成本加到路由应答的路由成本域中。在命令帧网络头中的目的地址应设置为下一跳地址,并且通过MCPS-DATA.request原语向下一跳设备单播发送。

#### 3.7.3.5.4 初始化和处理路由记录命令帧

如3.7.3.3小节所述,如果设备正在处理一个单播数据信息,并且目的地址的路由表入口的路由记录请求域为TRUE,它将首先检测路由表入口的Memory Constrained域。如果目的地址没有强制记忆,那么设备将向目的地址发送单播路由记录命令,如果向下一跳传输成功,则路由表入口的路由记录请求域置为FALSE。如果目的地址强制记录,并且要转发的消息由本地生成,或者接收于一个终端子设备,则向目的地址单播路由记录命令,否则不发该命令。

每一个接收到路由记录命令的转发节点应将它的网络地址、转发计数器增量、前向消息附加到其命令载荷。如果路由记录命令成功传输到下一跳,目的地址不是强制记忆,则目的地址路由表入口的路由记录请求域为FALSE。如果没有下一跳地址,或者若果向下一跳发送失败,或者对于id载荷没有足够的空间,则命令帧将被丢弃,网络层管理实体将通过NLME-ROUTE-ERROR.indication原语通知上层。

目的地址接收到路由记录命令后,路由将被存储在源路由表中。任何一个现有的消息路由的路线或中间节点都将被新的路由信息取代。

#### 3.7.3.6 结束路由发现表入口

当建立一个路由发现表入口,它的终止时间应设置为nwkcRouteDiscoveryTime毫秒。对于GroupId标志为TRUE的入口,当接收到的路由回复引起下一跳改变,则响应的路由发现表入口的终止时间域也发生改变。如果设备是路由回复的目的地址,则将终止时间域设置为nwkcWaitBeforeValidation毫秒,否则设置为nwkcRouteDiscoveryTime毫秒。当时间终止了,设备将从路由发现表中删除路由请求入口。如果设备是路由请求的始发设备,对应于目的地址的路由表入口的状态域值为VALIDATION\_UNDERWAY,那么设备将通过一个有效的路由传输消息。该消息或者为缓冲的未决路由发现,或者为路由错误命令,其错误代码为0x0a(有效路由)。如果对应于目的地址的路由表入口的状态值不是ACTIVE,并且在路由发现表中没有相同目的地址的入口,则路由表入口也将被删除。

#### 3.7.3.7 路由维护

每一个设备的网络层为每一个邻居设备维护一个失效计数器,该邻居设备具有一条输出链路,即要求发送一个数据帧。如果输出链路失效计数器的值超过了nwkcRepairThreshold,则设备根据如下所述方法,开始路由维护。可选择一个简单的失效计数方案来生成这个失效计数器值,或者使用一个更加准确的时间窗口方案。需要注意的是,由于修复操作涉及在整个网络,可能导致其他通信中断,因此,不可能经常对路由进行修复。退休的链路和终止查找失败的程序超出了该协议的范围。

##### 3.7.3.7.1 路由修复

当设备正在处理一个路由表入口的many-to-one域为TRUE的单播数据帧时,如果链路失败,则生成一个错误码为Many-to-one route failure的路由错误信息。路由错误命令帧网络帧头的目的地址域应与导致错误的帧的网络层帧头的目的地址域一致。路由错误命令的目的地址域应与导致错误的帧的网络层帧头的源地址一致。利用MCPS-DATA.request原语向它的随机路由器邻居单播路由错误信息。因为是多对一路由,所有的邻居都被认为由对应于目的地址的路由表入口。接收到路由偶错误帧后,如果没有对应于目的地址的路由表入口,或者向路由表入口的下一跳传输路由错误信息失败,那么将利用MCPS-DATA.request原语随机的



向它的路由器邻居发送路由错误信息。网络层帧头的半径将限制路由错误转发的最大次数。目的地址接收到路由错误帧，它将利用NLME-ROUTE-ERROR.indication原语通知上层路由错误。网络层不能自动的重新发现多对一路由。

如果设备在使用正常的单播路由处理单播数据帧时遇到链路失败，设备将向源设备发送返回一个路由错误命令帧，其错误代码标示失败的原因（见表3.40），并向上层发送NLME-ROUTE-ERROR.indication原语。

如果指定终端目的地址的父节点接收到路由错误命令帧，其命令帧载荷的错误代码值为0x01或0x02，表明链路失败，如果存在命令帧载荷中的目的地址相对应的路由表入口，网络层将移除该入口。然后向终端设备转发该帧。

如果终端设备接收到路由错误命令帧，网络层将向上层发送NLME-ROUTE-ERROR.indication原语。

如果是终端设备并且简化功能设备与父节点发生链路失败，终端设备将向上层发送NLME-ROUTE-ERROR.indication原语，其状态参数值为0x09，表明父节点连接失败（见表3.40）。同样，如果没有路由能力的ZigBee路由器与其父节点链路失败，其nwkUseTreeRouting值为TRUE，它将向上层发送状态参数为0x09的NLME-ROUTE-ERROR.indication原语，表明父节点链路失败。

3.7.4 广播通信

本节将介绍在一个ZigBee网络中如何实现广播传输。这种机制用来广播网络层所有的数据帧。网络中的任何设备都可以向同属该网络的其他设备进行广播。本地的应用层实体通过NLDE-DATA.request原语来进行广播传输，其中原语的DstAddr参数设置为广播地址，如表3.51所示。

表3.51 广播地址

广播地址	目的地组
0xffff	个域网的所有设备
0xfffe	保留
0xfffd	macRxOnWhenIdle=TRUE
0xfffc	所有的路由器和协调器
0xfff-0xfffb	保留

为发送一个广播MSDU，ZigBee路由器或协调器的网络层发送一个MCPS-DATA.request原语，其中DstAddrMode参数设置为0x02（16位网络地址），DstAddr参数设置为广播网络地址0xffff。作为ZigBee的终端设备，一个广播帧的MAC目的地址应该与终端设备的父节点的16位网络地址一致。其PANId参数应该设置为该ZigBee网络的PANID。这个协议不支持多个网络广播。广播传输不采用MAC层确认方式，使用被动的确认方式。被动确认机制是指每一个ZigBee路由器和协调器跟踪它的邻居设备以确认是否成功之间广播传输。将TxOption参数的确认传输标志设置为FALSE，则禁止MAC层的确认机制。其他所有TxOption参数的标志应该按照网络结构来设置。

ZigBee协调器、每一个路由器和终端设备，都要保存任何新的广播事务记录，不管该广播事务是从本地开始，还是从邻居设备接收来。该记录称为广播事务处理记录（BTR），它至少应该包含广播序号和广播数据帧的源地址。该广播事务处理记录存储在广播事务处理表（BTT）中。

当一个设备从邻居设备收到一个广播数据帧时，将数据帧中的广播序号和源地址与该设备的BTT表中的记录相比较。如果目的地址与如表3.52所示的接收者设备类型不一致，则丢

弃该帧。如果目的地址与接收者的设备类型相同，设备将广播帧中的序列号和源地址与BTR中的记录相比较。如果该设备存在一个BTR域这个广播数据帧相匹配，他就更新该BTR，标示邻居设备中继该广播数据帧。然后，抛弃该数据帧。如果不存在这样的记录，则在BTR中创建一个新的BTR记录，标示邻居设备中继该广播数据帧，网络层将向上层表明接收到一个新的广播数据帧，并将网络帧报头中的radius域减1，如果该值大于0，或者设备不是终端设备，则该设备将中继该数据帧，否则抛弃该数据帧。在转发之前，该设备将随机等候一段时间，该随机时间成为广播抖动，由nwkMaxBroadcastJitter属性值限定。如果ZigBee终端设备的macRxOnWhenIdle值为假时，那么该设备将不转发广播帧，并且不需要维护为广播帧产生的广播事务处理表。

如果接收到一个广播帧，而网络层发现它的广播事务处理表已经满了，并且没有终止的入口，那么这个帧将被忽略。在这种情况下，该帧既不会被转发，也不会被传到上一层。

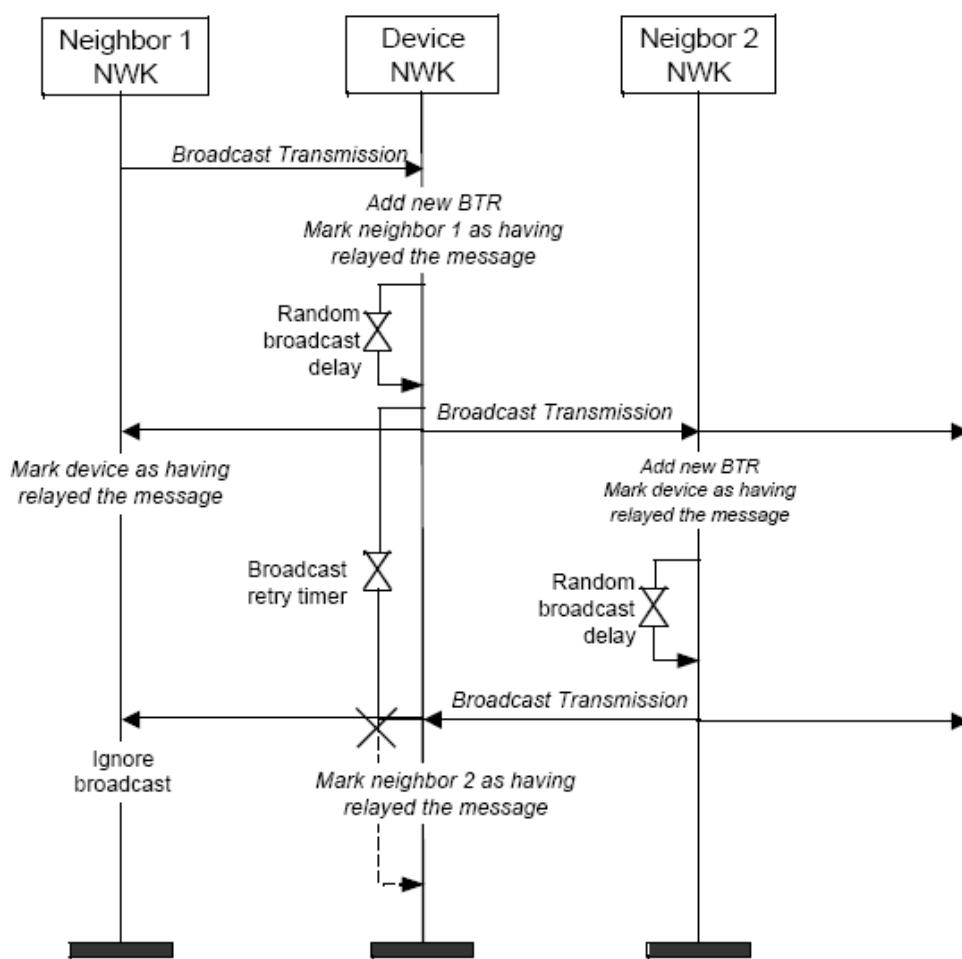
在转发了nwkMaxBroadcastRetries次后，ZigBee协调器或路由器将重发先前的广播帧。如果设备不支持被动确认机制，那么要转发该帧nwkMaxBroadcastRetries次。如果设备支持被动确认机制，在nwkPassiveAckTimeout 秒内，没有任何邻居设备中继广播数据帧，一个设备最多转发一个数据包nwkPassiveAckTimeout次。

当一个广播事务处理记录已经存在了nwkNetworkBroadcastDeliveryTime秒后，设备就可以更改这个广播事务处理记录入口的状态，如果接收到新的广播帧，那么这个入口将被更改。

当在MAC的PIB中的macRXOnWhenIdle属性设置为FALSE的ZigBee路由器接收到一个广播数据帧时，将使用与上述不相同的流程进行处理。它将使用MAC层单播法师，毫不延迟地向它的每一个邻居设备转发。利用MAC层的单播，将MCPS-DATA.request原语中的DstAddr参数设置为接收设备的地址，而不是广播地址。同样，一个拥有一个或者多个macRxOnWhenIdle MAC PIB 属性设置为FALSE的邻居路由器，而它自身的macRxOnWhenIdle MAC PIB 属性设置为TRUE时，如果它的目的地址为0xffff，那么人它除了按照上段所描述的长队程序外，还将利用MAC层单播一次向各个邻居转发该广播数据帧。为了确保这些单播到达它们的目的设备，可采用802.15.4-2003[B1]所介绍的间接传输的方式。

为有利于广播转发，每个ZigBee路由器必须具有在网络层中缓冲至少一个数据帧的能力。

图3.41描述了一个设备与它的两个邻居设备间的广播传输。



**Figure 3.41** Broadcast Transaction Message Sequence Chart

### 3.7.5多播通信

本小节描述在ZigBee网络中如何完成多播通信。多播提供多对多路由。多播使用16位多播组代码寻址。多播消息发送给一个特定的目的组且被注册为这个组的成员的所有设备接收。仅仅数据帧是多播；没有多播命令帧。多播帧有一个模式标志，表示它们是成员模式还是非成员模式。成员模式在目的组的成员减传输多播帧。只要它们是广播成员模式多播就在BTT里被记录。使用非成员模式传送多播帧从不是多播组成员的设备到是多播组成员的设备。在目的多播组中的成员之间进行的多播传送都是成员模式。如果它们先前已经被一个组成员传送那么在非成员之间进行的多播传送是成员模式，否则是非成员模式。多播消息被终端设备初始化但没有被参数`RxOnWhenIdle`等于FALSE的设备发送。

#### 3.7.5.1GROUP ID表

设备的网络层也许维持一个组代码表，`nwkGroupIDTable`，NIB中的一个属性，参见表3.4.2。如果`nwkGroupIDTable`的NIB属性存在，那么设备将包含一个16位组代码的设置，这个组是设备所在的组。

注意可选的`nwkGroupIDTable`的NIB属性与托管的APS组表有一个功能的交叠，参见表

2.18。如果设备包含两个表，且因此期望使用网络层多播作为一种接收组地址帧的方法，它必须确保每一个16位组代码出现在APS组表中和网络层组表中。

还要注意，从执行的方面讲，在每一层间复制该表是很浪费的，且假设执行将发现合并应用支持子层和网络层组表的一个方法来避免浪费。

### 3.7.5.2从高层接收到多播帧之后

如果网络层从高层接收到的数据帧，且多播控制域是0x01，那么网络层将确定该帧的组代码域和目的组代码域匹配的the *nwkGroupIDTable*的入口是否存在。如果匹配入口存在，网络层将根据3.7.5.2.1节的流程多播该帧。如果匹配入口不存在，该帧将作为一个非成员模式多播使用3.7.5.2.2节的流程来初始化。

#### 3.7.5.2.1初始化一个成员模式多播

网络层将设置多播控制域的多播模式子域的值是0x01（成员模式）。如果BTT表是满的且包含没有终止的入口，消息将不发送，且网络层数据实体将发送NLDEDATA.confirm原语，其状态参数值是BT\_TABLE\_FULL。如果BTT没满或者包含一个终止的BTR，一个新的BTR被建立，这个新的BTR带有作为源的本地节点和多播帧的序列号。消息将格局3.7.5.3节描述的流程来传送。

#### 3.7.5.2.2初始化一个非成员模式多播

网络层将设置多播控制域的多播模式子域的值是0x00（非成员模式）。那么网络层将为该帧相应的组代码目的的入口检测它的路由表。如果有这样一个入口，网络层将检查入口状态域。如果状态值是ACTIVE，那么设备将（重）传送该帧。如果状态值是VALIDATION\_UNDERWAY，那么状态将改变成ACTIVE，设备将根据3.7.5.4节描述的流程传送该帧，且网络层数据实体将发送NLDEDATA.confirm原语，其状态参数值为MCPSDATA.confirm原语返回的值。如果没有相应该帧组代码目的的路由表入口，且DiscoverRoute参数值是0x00（抑制路由发现），该帧丢弃，且网络层数据实体将发送NLDEDATA.confirm，其状态参数值是ROUTE\_DISCOVERY\_FAILED。如果DiscoverRoute参数值是0x01（使能路由发现），且有相应该帧组代码目的的路由表入口，那么设备将立即初始化路由发现如3.7.3.5.1节描述。该帧将随意的缓冲未决路由发现。如果没有缓存，该帧丢弃，且网络层数据实体将发送NLDEDATA.confirm原语，其状态参数值为FRAME\_NOT\_BUFFERED。

### 3.7.5.3在接收到成员模式多播帧之后

当设备从邻居设备接收到一个成员模式多播帧时，将数据帧中的序列号值和源地址与该设备的BTT表中的记录相比较。如果该设备存在在BTT中的这个多播帧的一个BTR，设备丢弃该帧。如果没有记录且BTT是满的和没有包含终止入口，那么设备丢弃该帧。如果没有记录且BTT不是满的或者包含终止BTR，设备建立一个新的BTR，且按照下一节的流程处理消息。当从邻居设备接收到一个成员模式多播帧且加到BTT中，网络层将确定是否在*nwkGroupIDTable*中存在入口，表的组代码域与该帧的目的组代码域想匹配。如果匹配入口存在，消息将传送到高层，多播控制域将设置成0x01（成员模式），NonmemberRadius域的值设置为maximum NonmemberRadius域的值，且消息将按下一节流程传送。如果匹配入口不存在，网络层将检测帧的多播NonmemberRadius域。如果多播NonmemberRadius域的值是0，丢弃消息，连同新增加的BTR。否则，如果值小于0x07，NonmemberRadius将消耗，且帧按照下一节流程传送。每一个成员模式多播消息能传送*nwkMaxBroadcastRetries*次。成员模式多播帧不在本地设备开始，最开始的传送将延迟一个随机时间，这个随机时间由*nwkMaxBroadcastJitter*属性值限定。设备在两个特殊成员模式帧转发之间将延迟一个周期*nwkPassiveAckTimeout*秒。不像广播，对于多点传送没有被动确认。ZigBee终端设备将不参与多播帧的转发。为了传送一个成员模式多播协议数据单元（MSDU），网络层发送MCPS-

DATA.request原语到MAC层，其DstAddrMode参数设置为0x02（16位网络地址），和DstAddr参数设置为0xffff——广播网络地址。PANId参数将设置为ZigBee网络的PANId。成员模式多播传送不使用MAC层确认或者广播使用的被动确认。MAC层确认通过设置TxOptions参数的确认传送标志为FALSE来不使能。TxOptions参数的所有其他标志以网络配置为基础。

#### 3.7.5.4在接收到非成员模式多播帧之后

当设备从邻居设备接收到一个非成员模式多播帧时，网络层将确定是否在 *nwkGroupIDTable* 中存在入口，表的组代码域与该帧的目的组代码域想匹配。如果匹配入口存在，多播控制域将设置成0x01（成员模式），且多播帧就按接收到的是成员模式多播处理。如果没有匹配的 *nwkGroupIDTable* 入口存在，设备将为该帧相应的组代码目的地的入口检测它的路由表。如果没有这样一个路由表入口，将丢弃消息。如果没有这样一个入口，网络层将检查入口状态域。如果状态值是ACTIVE，那么设备将（重）传送该帧。如果状态值是 VALIDATION\_UNDERWAY，那么状态将改变成ACTIVE，且设备将（重）传送该帧。为了传送一个非成员模式多播协议数据单元（MSDU），网络层发送MCPS-DATA.request原语到MAC层，其DstAddrMode参数设置为0x02（16位网络地址），和DstAddr参数设置为从匹配路由表中确定的下一跳。PANId参数将设置为ZigBee网络的PANId。MAC层确认通过设置TxOptions参数的确认传送标志为TRUE来使能。TxOptions参数的所有其他标志以网络配置为基础。

#### 3.7.6 MAC层信标中网络层的信息

本小节介绍了网络层怎样利用MAC层信标帧中的信标载荷将它的信息传送到邻居设备。

当设备的信标帧中的超帧规范域中的链接允许子域，如IEEE 802.15.4-2003[B1]所定义，设置为1时，标示设备允许连接。次年表在和所包含的信息如表3.52所示。这样，使网络层向新设备提供附加的信息来执行网络搜索任务，使这些新设备更有效地选择网络和特定的邻居设备，有关于网络搜索流程的详细信息，参见3.7.1.3.1.1。当设备信标的超帧域中的连接允许子域设置为0时，即禁止连接时，在信标载荷中不再需要这些信息。

**Table 3.52 NWK Layer Information Fields**

Name	Type	Valid Range	Description
Protocol ID	Integer	0x00 – 0xff	This field identifies the network layer protocols in use and, for purposes of this specification shall always be set to 0, indicating the ZigBee protocols; The value 0xff shall also be reserved for future use by the ZigBee Alliance
Stack profile	Integer	0x00 – 0x0f	A ZigBee stack profile identifier
nwkProtocolVersion	Integer	0x00 – 0x0f	The version of the ZigBee protocol
Router capacity	Boolean	TRUE or FALSE	This value is set to TRUE if this device is capable of accepting join requests from router-capable devices and is set to FALSE otherwise.
Device depth	Integer	0x00 – <i>nwkMaxDepth</i>	The tree depth of this device. A value of 0x00 indicates that this device is the ZigBee coordinator for the network
End device capacity	Boolean	TRUE or FALSE	This value is set to TRUE if the device is capable of accepting join requests from end devices seeking to join the network and is set to FALSE otherwise
<i>nwkExtendedPANID</i>	64-bit Extended address	0x0000000000000001 – 0xfffffffffffffffe	The globally unique ID for the PAN of which the beaconing device is a member. By default, this is the 64-bit IEEE address of the ZigBee coordinator that formed the network, but other values are possible and there is no required structure to the address

ZigBee协调器的网络层在网络形成后，将立即更新信标载荷。其他的所有ZigBee设备在连接完成时，或者网络配置（表3.11中的参数）发生任何编号时立即更新信标载荷。利用NLME-SET.request 原语 将 信 标 载 荷 写 入 到 MAC 层 个 域 网 信 息 数 据 库 中 ， 其 中 MacBeaconPayloadLength属性设置为信标载荷的长度，代表信标载荷的字节序列写入到 macBeaconPayload属性中。信标载荷的字节序列如图3.38所示。

Bits: 0 – 7	8 – 11	12 – 15	16 – 17	18	19 – 22	23	24 – 87
Protocol ID	Stack profile	nwkProtocol Version	Reserved	Router capacity	Device depth	End device capacity	<i>nwk Extended PANID</i>

**Figure 3.42** Format of the MAC Sub-Layer Beacon Payload

### 3.7.6.1 稳定数据

设备的操作可以通过手动或者编程人员通过程序进行重新设置。或者可以通过局部或网络范围内的能量短缺，或者在正常工作中的的电池更换，或者碰撞等原因进行更改。至少以下信息在重新设置的过程中为了网络操作需要进行保留：



- (1) 设备的PANId和扩展的PANId
- (2) 设备的16位网络地址
- (3) 每一个连接的终端子设备的64位IEEE地址和16位的网络地址，如果 *nwkAddrAlloc* 的值为0，则还包含每一个连接的路由器子设备。
- (4) 对于终端设备，还要记录它的父设备的16位网络地址
- (5) 被使用的堆栈结构
- (6) 设备深度

这些数据被保存的方法不再本协议的范围。

### 3.7.7地址冲突

当同一个网络的两个设备有相同的 *nwkShortAddress* 值时地址冲突发生。预防所有这样的冲突，例如使用树形地址分配和禁止已经分配的地址重复使用，是不实际的。本小节描述如何检测和修改地址冲突。如果NIB属性 *nwkUniqueAddr* 的值是FALSE地址冲突检测激活。

注意在路由消息中使用的网络地址在路由发现处理中是被校验。在发现的时候校验仅仅应用在设备、链路和目前信息。校验能在其他时间到达，如发送直接单播给邻居设备以前，是通过发送带有 *verify addresses* 错误码的路由错误命令。

#### 3.7.7.1获得地址信息

网络层从输入消息获得地址信息，输入信息包括网络层命令和ZDO数据消息。从ZDO数据消息处获得的地址信息通过加到NIB中的地址表通过网络层。

检测地址冲突的能力是通过增加一个或两个消息的网络层帧的目的IEEE地址和源IEEE地址域来加强的。当 *nwkUniqueAddr* 是FALSE是，所有的网络层命令消息将包含源IEEE地址和目的IEEE地址，如果它被源设备直到。其他设备不必包含任一IEEE地址域。

当 *nwkUniqueAddr* 是FALSE，路由请求命令将包含发送者的源地址域在发送IEEE地址域。这确保设备知道它们的邻居设备的IEEE地址。

#### 3.7.7.2检测地址冲突

网络层将把输入地址信息和本地设备本身的IEEE和网络地址、NIB中的地址和邻居表做比较，输入信息由特定的设备IEEE地址和网络地址组成的。

如果输入网络地址与 *nwkShortAddress* 匹配，但输入IEEE地址不等于 *aExtendedAddress*，网络层已经检测到一个带有 *nwkShortAddress* 的冲突。

如果输入网络地址与邻居表或者地址表的入口的网络地址相匹配，输入IEEE地址域入口的IEEE地址不匹配，且表入口的IEEE不是空IEEE地址（0x00...00），那么设备已经检测到一个在网络其他地方的冲突。

如果输入网络地址与邻居表入口的网络地址匹配，且表入口的IEEE地址是空IEEE地址（0x00...00），输入IEEE地址将代替表入口IEEE地址。没有冲突被检测到。

#### 3.7.7.3解决地址冲突

如果设备确定有一个或者多个设备使用它自己的网络地址，或者如果设备接收到带有 *address conflict* 错误代码和在目的域它自己的地址的路由错误命令，那么设备将获得一个新的地址。如果使用树形连接，请求一个新地址的设备将断开或重新连接网络。如果使用随机地址分配，请求一个新地址的设备将随机的选取一个新的地址，这个新的随机地址避免已经在NIB入口出现的所有地址。选择一个新的地址的终端设备将断开网络且那么使用新地址重新连接。

如果一个设备确定有多个设备使用一个地址，这个地址不是它自己的，它将使用带有 *address conflict* 错误码和在目的地址域厌恶的地址的广播路由错误命令通知网络。

如果一个父设备检测或被带有 *RxOnWhenIdle=FALSE* 子设备地址的冲突通知，父设备将



为子设备选取一个新地址,且发送一个主动重新连接响应命令帧来通知这个有新地址的子设备。