基于OAS的扩展MME及TLV格式定义

Contents

[1 缆桥终端业务接入管理系统体系结构示意图 5](#_Toc349694206)

[2 基本型缆桥终端互通管理流程 6](#_Toc349694207)

[3 家庭网关型缆桥终端管理通道建立流程 7](#_Toc349694208)

[4 Oas扩展MME 定义 9](#_Toc349694209)

[4.1 Oas Extended MME Type 9](#_Toc349694210)

[4.1.1 Processed by OasClient 9](#_Toc349694211)

[4.1.2 Transparent MME type 9](#_Toc349694212)

[4.2 Oas Extended MME Format 10](#_Toc349694213)

[4.3 Action code 13](#_Toc349694214)

[4.4 Extend MME definition 14](#_Toc349694215)

[4.4.1 Set RequestExtended MME 14](#_Toc349694216)

[4.4.2 Set ConfirmExtended MME 15](#_Toc349694217)

[4.4.3 Get RequestExtended MME 16](#_Toc349694218)

[4.4.4 Get Confirm Extended MME 17](#_Toc349694219)

[4.4.5 Indication Extended MME 20](#_Toc349694220)

[4.4.6 Response Extended MME 22](#_Toc349694221)

[5 Oas扩展MME TLV 定义 23](#_Toc349694222)

[5.1 MME TLV格式定义 23](#_Toc349694223)

[5.1.1 扩展MME的Main Type值 23](#_Toc349694224)

[5.2 操作对象实例索引TLV的格式 23](#_Toc349694225)

[5.2.1 实例索引TLV Subtype域的取值: 24](#_Toc349694226)

[5.3 扩展属性的TLV的格式 25](#_Toc349694227)

[5.3.1 CNU的基本配置 25](#_Toc349694228)

[5.3.2 Switch的基本配置 26](#_Toc349694229)

[5.3.3 Switch UNI端口的基本配置 28](#_Toc349694230)

[5.3.4 Switch 的VLAN配置管理 37](#_Toc349694231)

[5.3.5 Switch 的QoS相关配置 38](#_Toc349694232)

[5.3.6 Switch 的组播功能相关配置 41](#_Toc349694233)

[5.3.7 终端主动上报消息类型 43](#_Toc349694234)

[5.4 缆桥终端管理对象 43](#_Toc349694235)

[5.4.1 缆桥终端Capability的Variable Container格式 43](#_Toc349694236)

[5.5 终端家庭网关芯片管理对象 44](#_Toc349694237)

[5.5.1 家庭网关管理配置参数 44](#_Toc349694238)

[5.5.2 家庭网关芯片重启操作 47](#_Toc349694239)

[5.5.3 缆桥终端整机重启操作 47](#_Toc349694240)

[5.5.4 家庭网关芯片管理系统启动状态 48](#_Toc349694241)

[5.5.5 家庭网关无线SSID名称设置 48](#_Toc349694242)

[5.5.6 家庭网关处理MME结果上报消息 48](#_Toc349694243)

Figures

[Figure 1‑1缆桥终端业务接入管理系统体系结构示意图 7](#_Toc340678957)

[Figure 2‑1基本型缆桥终端互通管理流程 8](#_Toc340678958)

[Figure 3‑1家庭网关型缆桥终端管理通道建立流程 10](#_Toc340678959)

[Figure 4‑1 Set Request Extended MME消息结构 17](#_Toc340678960)

[Figure 4‑2 Set Confirm Extended MME消息结构 18](#_Toc340678961)

[Figure 4‑3Get Request Extended MME消息结构 19](#_Toc340678962)

[Figure 4‑4 Get Confirm Extended MME消息结构 20](#_Toc340678963)

[Figure 4‑5 Get Confirm Extended MME消息举例 21](#_Toc340678964)

[Figure 4‑6 Indication Extended MME消息结构 23](#_Toc340678965)

[Figure 4‑7 Response Extended MME消息结构 24](#_Toc340678966)

[Figure 5‑1 TLV的格式 25](#_Toc340678967)

Tables

[Table 4‑1 OasClient Processed MME type 11](#_Toc340678968)

[Table 4‑2 OasClient Transparent MME type 12](#_Toc340678969)

[Table 4‑3VS\_EXTENDED\_MME.request （0xAE08） 12](#_Toc340678970)

[Table 4‑4 VS\_EXTENDED\_MME.confirm（0xAE09） 13](#_Toc340678971)

[Table 4‑5 VS\_EXTENDED\_MME.indication（0xAE0A） 13](#_Toc340678972)

[Table 4‑6 VS\_EXTENDED\_MME.response（0xAE0B） 13](#_Toc340678973)

[Table 4‑7VS\_TRANSPARENT\_MME.request （0xAE0C） 14](#_Toc340678974)

[Table 4‑8 VS\_TRANSPARENT\_MME.confirm（0xAE0D） 14](#_Toc340678975)

[Table 4‑9 VS\_TRANSPARENT\_MME.indication（0xAE0E） 14](#_Toc340678976)

[Table 4‑10 VS\_TRANSPARENT\_MME.response（0xAE0F） 15](#_Toc340678977)

[Table 4‑11 Action Code 15](#_Toc340678978)

[Table 4‑12 Extended Get Response和Extended Set Response中附加的回复码 21](#_Toc340678979)

[Table 5‑1扩展MME管理对象的Main type值 25](#_Toc340678980)

[Table 5‑2扩展MME实例索引TLV的Main type值 25](#_Toc340678981)

[Table 5‑3操作对象的实例索引TLV的格式 26](#_Toc340678982)

[Table 5‑4实例索引TLV的Subtype值 26](#_Toc340678983)

[Table 5‑5扩展MME管理对象的Main type值 27](#_Toc340678984)

[Table 5‑6扩展操作的Subtype值 27](#_Toc340678985)

[Table 5‑7 CNU的基本配置对应的Main Type 和 Subtype 27](#_Toc340678986)

[Table 5‑8 Switch的基本配置对应的Main Type 和 Subtype 28](#_Toc340678987)

[Table 5‑9Switch UNI端口的基本配置对应的Main Type 和 Subtype 29](#_Toc340678988)

[Table 5‑10Switch 的VLAN配置管理对应的Main Type 和 Subtype 39](#_Toc340678989)

[Table 5‑11Switch 的QoS相关配置对应的Main Type 和 Subtype 40](#_Toc340678990)

[Table 5‑12Switch 的组播功能相关配置对应的Main Type 和 Subtype 43](#_Toc340678991)

[Table 5‑13终端主动上报消息的main type和subtype 45](#_Toc340678992)

[Table 5‑14缆桥终端Capability 45](#_Toc340678993)

[Table 5‑15 家庭网关管理 46](#_Toc340678994)

[Table 5‑16家庭网关管理通道参数 47](#_Toc340678995)

[Table 5‑16家庭网关业务WAN连接参数 48](#_Toc340678996)

[Table 5‑17 家庭网关芯片重启操作 49](#_Toc340678997)

[Table 5‑18 缆桥终端整机重启操作 49](#_Toc340678998)

# 缆桥终端业务接入管理系统体系结构示意图



Figure 1‑1缆桥终端业务接入管理系统体系结构示意图

# 基本型缆桥终端互通管理流程



Figure 2‑1基本型缆桥终端互通管理流程

* 1. OAS Client运行在EoCSlave芯片，缆桥终端启动后OASClient从终端EoC系统的Flash中读取终端的Capability属性保存到custommodule中，跟之前的MDIO commend分开保存,以便跟以前兼容。
  2. OAS Client调用API向EoCSlave芯片注册扩展MME消息。扩展MME消息注册后，所有注册过的扩展MME都转发给OAS Client处理，其中4个属于OAS\_EXTENDED类型的扩展MME由OAS Client自己处理，而4个属于TRANSPARENT类型的扩展MME由OAS Client替换目的MAC地址为家庭网关MAC地址后，发给UNI通道。反之，当OASClient收到家庭网关上报的TRANSPARENT类型扩展MME后，将源MAC地址替换为EoCSlave芯片的MAC后，发给EoC通道。
  3. HS收到EoC终端上线消息后，发送扩展MME(类型为0xAE08)查询终端的Capability属性。（注：HS可根据终端的UserHFID来区分终端类型，若是早期已支持互通的终端型号则不必采用查询终端Capability属性的方式，而是可以直接走原有的终端业务认证流程）
  4. 终端OASClient发送扩展MME(类型为0xAE09)回复终端的Capability属性。
  5. HS向NMS发送终端上线认证信息，认证信息中包括缆桥终端的EoC Slave芯片MAC。
  6. NMS和BOSS完成对该终端认证通过后，利用原有的配置文件下发方式，将缆桥终端配置参数下发给HS。
  7. HS收到配置后，解析配置文件，将配置参数保存在HS本地内存。若是对终端EoC Slave芯片的配置，则直接通过标准MME下发，若是针对缆桥终端交换芯片的配置，则封装到扩展MME(0xAE08)发送给EoC终端。（注：若HS未收到配置，则按照原有流程定期向NMS发送终端上线认证信息）
  8. 终端OAS Client收到扩展MME(类型为0xAE08)后，根据消息内容，对交换芯片进行配置，并将配置结果封装成扩展MME(类型为0xAE09)回应HS。

# 家庭网关型缆桥终端管理通道建立流程



Figure 3‑1家庭网关型缆桥终端管理通道建立流程

1. 同基本型缆桥终端互通管理流程（1）。
2. 同基本型缆桥终端互通管理流程（2）。
3. 同基本型缆桥终端互通管理流程（3）。
4. OAS Client发送扩展MME(类型为0xAE09)回复终端的Capability属性，Capability属性中包括家庭网关芯片的MAC地址。
5. HS向NMS发送终端上线认证信息，认证信息中包括EoC Slave芯片的MAC和家庭网关芯片的MAC。
6. NMS和BOSS完成对该终端认证通过后，利用原有的配置文件下发方式，将家庭网关的管理VLAN和管理IP(含子网掩码、IP网关、ACS URL、DNS IP)设置下发给HS。
7. HS收到配置后，解析配置文件并将家庭网关的配置参数封装成扩展MME(0xAE0C)发送给EoC终端 (此MME注册为透传模式)。（注：若HS未收到配置，则按照原有流程定期向NMS发送终端上线认证信息）
8. 终端的OASClient芯片收到扩展MME(类型为0xAE0C)后替换目的MAC地址为家庭网关的MAC地址，源地址改为Slave的MAC地址，然后发送到家庭网关芯片。
9. 家庭网关收到扩展MME后解析报文内容，配置管理VLAN和管理IP(含子网掩码、IP网关、ACS URL、DNS IP)等相关参数，并将配置结果发送扩展MME(类型为0xAE0D)回应HS。
10. 从家庭网关发回给HS的扩展MME(类型为0xAE0D)报文，在终端EoC芯片的OASClient处将报文中源MAC地址由家庭网关MAC替换为终端EoCSlave芯片的MAC。
11. HS收到家庭网关配置成功结果后，在本地建立EoC终端和家庭网关的关联关系，并保存家庭网关的相关配置信息。
12. 家庭网关在获得配置管理VLAN和管理IP(含子网掩码、IP网关、ACS URL、DNS IP)等相关参数后，即可开始与ACS之间建立TR069的连接。

# Oas扩展MME 定义

## Oas Extended MME Type

对MME进行扩展，使MME能够完成对OAMSet/Get的对象索引TLV，Variable Descriptor/Container TLV以及OAM event/alarm的封装。由于开放给客户的并通过OasClient来处理的扩展MME必须是在0xAC00之后，且最大支持512个扩展MME，考虑到之前有客户已经扩展了一部分，为了防止冲突，我们定义从0xAE08开始。定义两组扩展MME，第一组直接交由OasClient处理，第二组从Slave的UNI口直接透传出去。

### Processed by OasClient

| **MMTYPE** | **Interpretation** | **Description** | **Comment** |
| --- | --- | --- | --- |
| 0xAE08 | VS\_OAS\_EXTENDED\_MME.request | Encapsulate extended MME Set/Get request |  |
| 0xAE09 | VS\_OAS\_EXTENDED\_MME.confirm | Encapsulate extended MME Set/Get response |  |
| 0xAE0A | VS\_OAS\_EXTENDED\_MME.indication | Encapsulate extended MME event/alarm |  |
| 0xAE0B | VS\_OAS\_EXTENDED\_MME.response | Encapsulate extended MME event/alarm response |  |

Table 4‑1OasClientProcessed MME type

### Transparent MME type

| **MMTYPE** | **Interpretation** | **Description** | **Comment** |
| --- | --- | --- | --- |
| 0xAE0C | VS\_TRANSPARENT\_MME.request | Encapsulate transparent MME Set/Get request | Host to home gateway |
| 0xAE0D | VS\_TRANSPARENT\_MME.confirm | Encapsulate transparent MME Set/Get response | Home gateway to host |
| 0xAE0E | VS\_TRANSPARENT\_MME.indication | Encapsulate transparent MME event/alarm | Home gateway to host |
| 0xAE0F | VS\_TRANSPARENT\_MME.response | Encapsulate transparent MME event/alarm response | Host to home gateway |

Table 4‑2OasClient Transparent MME type

## Oas Extended MME Format

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Size** | **Field Name** | **Description** |
| 0x0000 | 6 | ODA | Original Destination Address |
| 0x0006 | 6 | OSA | Original Source Address |
| 0x000C | 4 | VLAN | VLAN Tag (optional) |
| 0x0010 | 2 | MTYPE | Ethertype (Home Plug AV =0xe188) |
| 0x0012 | 1 | MMV | MME Version (=0) |
| 0x0013 | 2 | MMTYPE | 0xAE08 (Request) |
| 0x0015 | 3 | OUI | Intellon (0x00, 0xb0, 0x52) |
| 0x0018 | 1 | Action code | 0x01(Set), 0x02(Get), 0x03(Reset) |
| 0x0019 | X | MME\_TLV | Encapsulated OAM set/get request TLVs. |

Table 4‑3VS\_EXTENDED\_MME.request（0xAE08）

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Size** | **Field Name** | **Description** |
| 0x0000 | 6 | ODA | Original Destination Address |
| 0x0006 | 6 | OSA | Original Source Address |
| 0x000C | 4 | VLAN | VLAN Tag (optional) |
| 0x0010 | 2 | MTYPE | Ethertype (Home Plug AV =0xe188) |
| 0x0012 | 1 | MMV | MME Version (0) |
| 0x0013 | 2 | MMTYPE | 0xAE09 (Confirm) |
| 0x0015 | 3 | OUI | Intellon OUI (0x00, 0xb0, 0x52) |
| 0x0018 | 1 | MSTATUS | Status(0x00=Success, 0x01=Fail) |
| 0x0019 | 1 | Action code | 0x01(Set), 0x02(Get) |
| 0x001A | X | MME\_TLV | Encapsulated OAMset/get confirm TLVs. |

Table 4‑4VS\_EXTENDED\_MME.confirm（0xAE09）

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Size** | **Field Name** | **Description** |
| 0x0000 | 6 | ODA | Original Destination Address |
| 0x0006 | 6 | OSA | Original Source Address |
| 0x000C | 4 | VLAN | VLAN Tag (optional) |
| 0x0010 | 2 | MTYPE | Ethertype (Home Plug AV =0xe188) |
| 0x0012 | 1 | MMV | MME Version (0) |
| 0x0013 | 2 | MMTYPE | 0xAE0A (indication) |
| 0x0015 | 3 | OUI | Intellon OUI (0x00, 0xb0, 0x52) |
| 0x0018 | 1 | Action code | 0x00(No action) |
| 0x0019 | X | MME\_TLV | Encapsulated OAMevent/alarm TLVs |

Table 4‑5VS\_EXTENDED\_MME.indication（0xAE0A）

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Size** | **Field Name** | **Description** |
| 0x0000 | 6 | ODA | Original Destination Address |
| 0x0006 | 6 | OSA | Original Source Address |
| 0x000C | 4 | VLAN | VLAN Tag (optional) |
| 0x0010 | 2 | MTYPE | Ethertype (Home Plug AV =0xe188) |
| 0x0012 | 1 | MMV | MME Version (0) |
| 0x0013 | 2 | MMTYPE | 0xAE0B (resonse) |
| 0x0015 | 3 | OUI | Intellon OUI (0x00, 0xb0, 0x52) |
| 0x0018 | 1 | MSTATUS | Status(0x00=Success, 0x01=Fail) |
| 0x0019 | 1 | Action code | 0x00(No action) |
| 0x001A | X | MME\_TLV | Encapsulated OAMevent/alarm response TLVs |

Table 4‑6VS\_EXTENDED\_MME.response（0xAE0B）

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Size** | **Field Name** | **Description** |
| 0x0000 | 6 | ODA | Original Destination Address |
| 0x0006 | 6 | OSA | Original Source Address |
| 0x000C | 4 | VLAN | VLAN Tag (optional) |
| 0x0010 | 2 | MTYPE | Ethertype (Home Plug AV =0xe188) |
| 0x0012 | 1 | MMV | MME Version (=0) |
| 0x0013 | 2 | MMTYPE | 0xAE0C (Request) |
| 0x0015 | 3 | OUI | Intellon (0x00, 0xb0, 0x52) |
| 0x0018 | 1 | Action code | 0x01(Set), 0x02(Get), 0x03(Reset) |
| 0x0019 | X | MME\_TLV | Encapsulated OAMset/get request TLVs. |

Table 4‑7VS\_TRANSPARENT\_MME.request（0xAE0C）

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Size** | **Field Name** | **Description** |
| 0x0000 | 6 | ODA | Original Destination Address |
| 0x0006 | 6 | OSA | Original Source Address |
| 0x000C | 4 | VLAN | VLAN Tag (optional) |
| 0x0010 | 2 | MTYPE | Ethertype (Home Plug AV =0xe188) |
| 0x0012 | 1 | MMV | MME Version (0) |
| 0x0013 | 2 | MMTYPE | 0xAE0D (Confirm) |
| 0x0015 | 3 | OUI | Intellon OUI (0x00, 0xb0, 0x52) |
| 0x0018 | 1 | MSTATUS | Status(0x00=Success, 0x01=Fail) |
| 0x0019 | 1 | Action code | 0x01(Set), 0x02(Get) |
| 0x001A | X | MME\_TLV | Encapsulated OAMset/get confirm TLVs. |

Table 4‑8VS\_TRANSPARENT\_MME.confirm（0xAE0D）

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Size** | **Field Name** | **Description** |
| 0x0000 | 6 | ODA | Original Destination Address |
| 0x0006 | 6 | OSA | Original Source Address |
| 0x000C | 4 | VLAN | VLAN Tag (optional) |
| 0x0010 | 2 | MTYPE | Ethertype (Home Plug AV =0xe188) |
| 0x0012 | 1 | MMV | MME Version (0) |
| 0x0013 | 2 | MMTYPE | 0xAE0E (indication) |
| 0x0015 | 3 | OUI | Intellon OUI (0x00, 0xb0, 0x52) |
| 0x0018 | 1 | Action code | 0x00(No action) |
| 0x0019 | X | MME\_TLV | Encapsulated OAMevent/alarm TLVs |

Table 4‑9VS\_TRANSPARENT\_MME.indication（0xAE0E）

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset** | **Size** | **Field Name** | **Description** |
| 0x0000 | 6 | ODA | Original Destination Address |
| 0x0006 | 6 | OSA | Original Source Address |
| 0x000C | 4 | VLAN | VLAN Tag (optional) |
| 0x0010 | 2 | MTYPE | Ethertype (Home Plug AV =0xe188) |
| 0x0012 | 1 | MMV | MME Version (0) |
| 0x0013 | 2 | MMTYPE | 0xAE0F (resonse) |
| 0x0015 | 3 | OUI | Intellon OUI (0x00, 0xb0, 0x52) |
| 0x0018 | 1 | MSTATUS | Status(0x00=Success, 0x01=Fail) |
| 0x0019 | 1 | Action code | 0x00(No action) |
| 0x001A | X | MME\_TLV | Encapsulated OAMevent/alarm response TLVs |

Table 4‑10VS\_TRANSPARENT\_MME.response（0xAE0F）

## Action code

每个MME都对应有一种操作类型，该类型通过Action Code来区分是Set还是Get操作，indication和response MME对应的action code为0x00。

Table 4‑11Action Code

|  |  |  |
| --- | --- | --- |
| **Action Code** | **值** | **描述（Description）** |
| No Action | 0x00 | 用于Extended Indication/Response消息。 |
| Set | 0x01 | 用于Extended Set Request/Response消息。 |
| Get | 0x02 | 用于Extended Get Request/Response消息。 |
| Reset | 0x03 | 用于执行网关芯片或者终端整机的重启操作 |

## Extend MME definition

### Set RequestExtended MME

Set Request Extended MME用于Master向Slave设置相关属性和操作，具体消息结构如Figure4‑1所示。每个实例索引TLV后可以包含一个或多个Variable Container TLV，这些Variable Container TLV均是对这个实例索引TLV的各种属性和操作进行设置。Set Request Extended MME消息可以包含一个或多个实例索引TLV。

如果Set Request Extended MME中的某个Variable Container TLV中携带的Value的长度大于127，则应将该Variable Container TLV进行分段传输，如Figure4‑5所示。Slave接收时，如果检测到连续的Variable Container TLV中含有同样的MainType和Subtype字段，应认为这是同一个Variable Container TLV。

说明：

1. MME的数据部分以一个字节Pad=0x00作为结束标志。
2. 所有的报文都采用Little-endian字节序



Figure 4‑1Set Request Extended MME消息结构

### Set ConfirmExtended MME

Set Confirm Extended MME用于用于Slave向Master回复其设置的相关属性和操作的执行结果，具体消息结构如Figure4‑2所示。Set Confirm Extended MME应分别对Extended Set Request消息的每个实例对象的每个属性或操作进行回复，具体回复码的取值见Table 4‑12的规定，回复码放在TLV的length字段中返回，Set Confirm Extended MME消息的VariableContainer不携带Value域。如果所有的TLV对应的操作都成功，则“MSTATUS”域等于0，如果MSTATUS大于0，表示有操作失败，MSTATUS的值表示操作失败的个数。



Figure 4‑2Set Confirm Extended MME消息结构

### Get RequestExtended MME

Get RequestExtendedMME用于Master向Slave查询相关属性和操作，具体消息结构如Figure4‑3所示。每个实例索引TLV后可以包含一个或多个Variable Descriptor TLV，这些Variable Descriptor TLV均是对这个实例索引TLV的各种属性和操作进行查询。Get Request Extended MME 对应的Variable Descriptor TLV不包含length和value。Extended Get Request扩展OAMPDU消息可以包含一个或多个实例索引TLV。



Figure 4‑3Get Request Extended MME消息结构

### Get Confirm Extended MME

Get Confirm Extended MME用于Slave向Master回复其查询的相关属性和操作，具体消息结构如Figure4‑4所示。Get Confirm Extended MME应分别对Extended Get Request消息的每个实例对象的每个属性或操作进行回复；即每个实例索引TLV后可以包含一个或多个Variable Container TLV，与Get Request Extended MME消息中的Variable Descriptor TLV一一对应；Get Confirm Extended MME消息可以包含一个或多个实例索引TLV，与Get Request Extended MME消息中的实例索引TLV一一对应。



Figure 4‑4Get Confirm Extended MME消息结构

考虑到Variable Container TLV中的Variable Width的取值范围是1～127，如果某个Variable Container TLV中携带的Value的长度大于127，则应将该Variable Container TLV进行分段传输，如Figure4‑5所示。接收时，如果检测到连续的Variable Container TLV中含有同样的MainType和Subtype字段，应认为这是同一个Variable Container TLV。



Figure 4‑5Get Confirm Extended MME消息举例

如果Get Confirm Extended MME消息中回复码的取值与Table4‑12相同的话(0x80除外)，Get Confirm Extended MME消息的VariableContainer不携带Value域，回复码放在length中返回。如果所有的TLV对应的操作都成功，则“MSTATUS”域等于0，如果MSTATUS大于0，表示有操作失败，MSTATUS的值表示操作失败的个数。

Table 4‑12Extended Get Response和Extended Set Response中附加的回复码

|  |  |  |
| --- | --- | --- |
| **附加的回复码**  **（Additional Variable Indication）** | **值** | **描述（Description）** |
| SetOK | 0x80 | 设置请求（Extended Set Request）的属性（Attribute）或操作（Action）操作成功，用于Extended Set Response消息。 |
| Set Fail | 0x81 | 设置请求（Extended Set Request）的属性（Attribute）或操作（Action）操作失败，用于Extended Set Response消息。 |
| VarBadParameters | 0x86 | 请求的属性（Attribute）或操作（Action）的参数无效，用于Extended Get Response和Extended Set Response消息。 |
| VarNoResource | 0x87 | 请求的属性（Attribute）或操作（Action）的参数有效，但Switch的当前状态使该请求或操作无法完成，用于Extended Get Response和Extended Set Response消息。 |

### Indication Extended MME

Indication Extended MME用于Slave向Master上报相关的状态信息，具体消息结构如Figure4‑6所示。



Figure 4‑6Indication Extended MME消息结构

考虑到Variable Container TLV中的Variable Width的取值范围是1～127，如果某个Variable Container TLV中携带的Value的长度大于127，则应将该Variable Container TLV进行分段传输，如Figure4‑5所示。接收时，如果检测到连续的Variable Container TLV中含有同样的MainType和Subtype字段，应认为这是同一个Variable Container TLV。

### Response Extended MME

Response Extended MME用于Master对slave的indication消息的回应，具体消息结构如Figure4‑7所示。



Figure 4‑7Response Extended MME消息结构

考虑到Variable Container TLV中的Variable Width的取值范围是1～127，如果某个Variable Container TLV中携带的Value的长度大于127，则应将该Variable Container TLV进行分段传输，如Figure4‑5所示。接收时，如果检测到连续的Variable Container TLV中含有同样的MainType和Subtype字段，应认为这是同一个Variable Container TLV。

# Oas扩展MME TLV 定义

## MME TLV格式定义

对switch的某个或者多个属性做查询或者配置操作时，都必须有一个管理对象TLV来指明所要管理的对象。在一个扩展MME中，一个操作对象实例索引之后的全部Variable Descriptor或Variable Container均为针对该实例的属性和操作，直至该MME的数据部分结束或出现该操作对象的另外一个实例索引或者另外一个操作对象的Variable Descriptor/Container



Figure 5‑1TLV的格式

其中Type第一个byte表示Main type，后两个byte表示Subtype；Length表示数据value的长度；value表示TLV的数据部分。

### 扩展MME的Main Type值

Table 5‑1扩展MME管理对象的Main type值

|  |  |  |
| --- | --- | --- |
| **值（hex）** | **Maintype的定义** | **描述** |
| 0xF7 | 实例索引 | 表示该TLV为操作对象的实例索引 |
| 0xF8 | 扩展属性  （Extended Attribute） | 扩展的属性，可以执行Get(或)Set命令以及Reset命令 |

## 操作对象实例索引TLV的格式

Table 5‑2扩展MME实例索引TLV的Main type值

|  |  |  |
| --- | --- | --- |
| **值（hex）** | **Maintype的定义** | **描述** |
| 0xF7 | 实例索引 | 表示该TLV为操作对象的实例索引 |

Table 5‑3操作对象的实例索引TLV的格式

|  |  |  |
| --- | --- | --- |
| **字节数** | **字段** | **描述** |
| 1 | Main Type | 值为0xF7，表示本TLV为操作对象的实例索引 |
| 2 | Subtype | 操作对象的类型(见Table 5‑4) |
| 1 | Length | 值为0x04 |
| 4 | Value | 表示具体的操作对象的实例的编号 |

每个实例索引TLV对应一个操作对象，每个Extended MME可以包含多个索引TLV；同时每个索引下面的每个操作TLV对应该操作对象的一个管理操作，每个索引TLV下面可以有一个或多个操作TLV，对应该管理对象的一个或多个管理操作。

其中Type第一个byte表示Main type，后两个byte表示Subtype。如果Main type == 0xF7，则表示该TLV为索引TLV；如果操作对象是Switch 的Port，Subtype=0x0001；如果操作对象是Switch，Subtype = 0x0002；如果操作对象是Gateway， Subtype=0x0003；如果操作对象是Gateway的Port，Subtype=0x0004；如果操作对象是EOC Slave 终端设备，Subtype=0x0005。

Value域表示该操作对象的实例编号。针对不同的操作对象，Value域的取值如下所示

1. 如果操作的对象是端口，第一个字节：保留。第二个字节：表示端口的编号。0x00表示上联口，0x01表示端口号为1，0x02表示端口号为2，以此类推，端口编号范围为0x01～0xFF。注意该端口号为逻辑端口号。第三个字节：表示本次操作是否还有Extended MME报文，其中0x01表示还有Extended MME报文， 0x00表示该MME报文是本次操作的最后一个MME报文；第四个字节：表示Extended MME报文的序列号，计数从0开始，当达到255时，然后回转从0开始。
2. 如果操作对象是switch/gateway/slave，则前2个字节保留，第三个字节：表示本次操作是否还有Extended MME报文，其中0x01表示还有Extended MME报文， 0x00表示该MME报文是本次操作的最后一个MME报文；第四个字节：表示Extended MME报文的序列号，计数从0开始，当达到255时，然后回转从0开始。

### 实例索引TLV Subtype域的取值:

Table 5‑4实例索引TLV的Subtype值

|  |  |  |
| --- | --- | --- |
| **对象（object）** | **Subtype值（Value）** | **描述** |
| Switch Port | 0x0001 | 操作对象为switch端口 |
| Switch | 0x0002 | 操作对象为Switch |
| Gateway（终端家庭网关芯片） | 0x0003 | 操作对象为终端家庭网关芯片（扩展MME透传给家庭网关处理） |
| Gateway Port | 0x0004 | 操作对象为终端家庭网关端口 |
| EOC Slave 终端设备 | 0x0005 | 操作对象为缆桥终端设备本身（扩展MME是OAS Client处理） |
|  | 其他 | 保留，在接收时忽略处理 |

## 扩展属性的TLV的格式

Table 5‑5扩展MME管理对象的Main type值

|  |  |  |
| --- | --- | --- |
| **值（hex）** | **Maintype的定义** | **描述** |
| 0xF8 | 扩展属性  （Extended Attribute） | 扩展的属性，可以执行Get(或)Set命令以及Reset命令 |

其中每个subtype对应一个管理操作。

Table 5‑6扩展操作的Subtype值

|  |  |
| --- | --- |
| **描述（Description）** | **Subtype** |
| CNU的基本配置（Capability etc.） | 0x0001－0x000F |
| Switch 的基本配置 | 0x0011－0x001F |
| Switch UNI端口的基本配置（端口的打开/关闭，端口镜像,以太网端口的Flow Control等） | 0x0021－0x003F |
| Switch 的VLAN配置管理 | 0x0041－0x005F |
| Switch 的QoS相关配置 | 0x0061－0x007F |
| Switch 组播功能 | 0x0081－0x009F |
| 终端主动上报消息类型 | 0x0101－0x010F |
| 家庭网关管理通道参数配置 | 0x1001－0x100F |
| 重启操作 | 0x2001－0x200F |
| 家庭网关基本配置 | 0x3001－0x3FFF |

### CNU的基本配置

Table 5‑7CNU的基本配置对应的Main Type和Subtype

| **属性/操作名称** | **操作对象** | **Main Type** | **Subtype** | **HomePlug AV Slave** | **QCA MME** | **Get**  **（action code= 0x02）** | **Set**  **（action code = 0x01）** | **备注** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| eocCnuCapability | EOC Slave | 0xF8 | 0x0001 | current | New | ○ |  | 获取缆桥终端的产品形态 (见Table 5‑14) |
| eocCnuCPEIPAddressList | EOC Slave | 0xF8 | 0x0002 | optional | New | ○ |  | EOC终端设备连接CPE IP地址列表  EOC终端设备连接CPE IP地址列表，该参数为可选项。以连续4字节OCTET STING类型表示。如检测到2个CPE连接，IP地址  分别为10.0.0.1和10.0.0.2，则返回值为0x0a 0x00 0x00 0x01 0x0a 0x00 0x00 0x02，返回值长度8。 |

### Switch的基本配置

Table 5‑8Switch的基本配置对应的Main Type和Subtype

| **属性/操作名称** | **操作对象** | **Main Type** | **Subtype** | **HomePlug AV Slave** | **QCA MME** | **Get**  **（action code= 0x02）** | **Set**  **（action code = 0x01）** | **备注** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| eocPortLoopDetectEn | Switch | 0xF8 | 0x0011 | current | New | ○ | ○ | EOC终端环路检测端口使能（1字节）  Looping check timer:  0: Disable  1: 1ms  2: 10ms  3: 100ms  4: 500ms |
| eocCnuCPEMacAddressList | Switch | 0xF8 | 0x0012 | optional | New | ○ |  | EOC终端设备连接CPE MAC地址列表  EOC终端设备连接CPE MAC地址列表，该参数为可选项。以连续8字节OCTET STRING类型表示。其中前6字节为MAC地址，第七字节表示该MAC是静态配置的还是动态学习到的，  1: static  0: dynamic  第八字节表示该MAC对应的目的端口，bit 0对应端口0，bit 1 对应端口1，以此类推。如果其中一个bit置1，表示该端口即为该MAC对应的目的端口. |
| eocAgingTime | Switch | 0xF8 | 0x0013 | current | New | ○ | ○ | EOC终端aging time（老化时间）周期，单位:秒 ，4字节 |
| eocSlaveSwitchMibEn | Switch | 0xF8 | 0x0014 | Current | New | ○ | ○ | Switch 流量统计使能：  1: Enable  0: disable |
| eocSlaveLogicalToPhysicalPortMap | Switch | 0xF8 | 0x0015 | Current | New | ○ | ○ | 配置switch逻辑端口跟物理端口的映射关系((port number \* 2) +1 bytes)  Byte 1: 端口个数  紧接着每两个byte对应一个映射关系，其中第一个byte表示逻辑端口号，第二个byte表示该逻辑端口对应的物理端口号 |
| eocCnuMaxUpBandwidthLimit | Switch | 0xF8 | 0x0016 | current | New | ○ | ○ | 终端最大上行限速配置，（4bytes）  分别为：  64 64Kbps  128 128Kbps  256 256Kbps  512 512Kbps  1024 1024Kbps  2048 2048Kbps  4096 4096Kbps  6144 6144Kbps  8192 8192Kbps  10240 10240Kbps  32000 32000Kbps  64000 64000Kbps  128000 128000Kbps  不设置上限时设置值为0 |

### Switch UNI端口的基本配置

Table 5‑9Switch UNI端口的基本配置对应的Main Type和Subtype

| **属性/操作名称** | **操作对象** | **Main Type** | **Subtype** | **HomePlug AV Slave** | **QCA MME** | **Get**  **（action code= 0x02）** | **Set**  **（action code = 0x01）** | **备注** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| eocMirrorPortIngress | Switch port | 0xF8 | 0x0021 | current | New | ○ | ○ | EOC终端端口镜像使能（1字节）  复制该端口的入栈流  1: enable  0: disable |
| eocMirrorPortEgress | Switch port | 0xF8 | 0x0022 | current | New | ○ | ○ | EOC终端端口镜像使能（1字节）  复制该端口的出栈流  1: enable  0: disable |
| eocPortMirrorDestPortSet | Switch port | 0xF8 | 0x0023 | current | New | ○ | ○ | 配置EOC设备端口镜像目地端口使能(1字节)  1: enable  0: disable |
| eocCnuPortLinkStatus | Switch port | 0xF8 | 0x0024 | current | New | ○ |  | EOC终端设备以太网口连接状态（1字节）  1: 有CPE连接  0: 无CPE连接 |
| eocFlowControlEn | Switch port | 0xF8 | 0x0025 | current | New | ○ | ○ | EOC终端流控制使能（1字节）  1: Enable  0: Disable |
| eocPortStateCtl | Switch port | 0xF8 | 0x0026 | current | New | ○ | ○ | 端口状态控制（1字节）  1: blocking  2: listening  3: learning  4: forwarding  0: disable |
| eocPortDuplexMode | Switch port | 0xF8 | 0x0027 | current | New | ○ | ○ | 配置端口双工模式（1字节）  1: full duplex  0: half duplex |
| eocPortSpeedMode | Switch port | 0xF8 | 0x0028 | current | New | ○ | ○ | 配置端口速率模式（1字节）  0: 10M  1: 100M  2: 1000M |
| eocPortAutoNeg | Switch port | 0xF8 | 0x0029 | current | New | ○ | ○ | 使能端口自协商功能（1字节）  1: enable  0: disable |
| eocSlavePortUnknownUnicastPacketsStormControlEnable | Switch port | 0xF8 | 0x002A | current | New | ○ | ○ | 未知单播风暴抑制（1字节）  For unknown unicast packets  0: disable  1: enable |
| eocSlavePortUnknownMulticastPacketsStormControlEnable | Switch port | 0xF8 | 0x002B | current | New | ○ | ○ | 未知组播风暴抑制（1字节）  For unknown multicast packets  0: disable  1: enable |
| eocSlavePortBroadcastPacketsStormControlEnable | Switch port | 0xF8 | 0x002C | current | New | ○ | ○ | 广播风暴抑制（1字节）  For broadcast packets  0: disable  1: enable |
| eocSlavePortUnknownUnicastPacketsStormLimitRate | Switch port | 0xF8 | 0x002D | current | New | ○ | ○ | 未知单播风暴抑制门限（4字节）  Forunknown unicast packets Storm control rate:  0:storm control disable  1000 frame per second  2000 frame per second  4000 frame per second  8000 frame per second  16000 frame per second  32000 frame persecond  64000 frame per second  ……  1024000 frame per second. |
| eocSlavePortUnknownMulticastPacketsStormLimitRate | Switch port | 0xF8 | 0x002E | current | New | ○ | ○ | 未知组播风暴抑制门限（4字节）  Forunknown multicast packets Storm control rate:  0:storm control disable  1000 frame per second  2000 frame per second  4000 frame per second  8000 frame per second  16000 frame per second  32000 frame persecond  64000 frame per second  ……  1024000 frame per second. |
| eocSlavePortBroadcastPacketsStormLimitRate | Switch port | 0xF8 | 0x002F | current | New | ○ | ○ | 广播风暴抑制门限（4字节）  Forbroadcast packets  Storm control rate:  0:storm control disable  1000 frame per second  2000 frame per second  4000 frame per second  8000 frame per second  16000 frame per second  32000 frame persecond  64000 frame per second  ……  1024000 frame per second. |
| eocPortMaxDownBandwidthLimit | Switch port | 0xF8 | 0x0030 | current | New | ○ | ○ | 下行端口最大出口限速（4bytes）,分别为：  64: 64Kbps  128: 128Kbps  256: 256Kbps  512: 512Kbps  1024: 1024Kbps  2048: 2048Kbps  4096: 4096Kbps  6144: 6144Kbps  8192: 8192Kbps  10240: 10240Kbps  32000: 32000Kbps  64000: 64000Kbps  128000: 128000Kbps  不设置上限时设置值为0 |
| eocPortMaxUpBandwidthLimit | Switch port | 0xF8 | 0x0031 | current | New | ○ | ○ | 上行端口最大入口限速（4bytes）,分别为：  64 64Kbps  128 128Kbps  256 256Kbps  512 512Kbps  1024 1024Kbps  2048 2048Kbps  4096 4096Kbps  6144 6144Kbps  8192 8192Kbps  10240 10240Kbps  32000 32000Kbps  64000 64000Kbps  128000 128000Kbps  不设置上限时设置值为0 |
| eocSlavePhyPowerCtl | Switch port | 0xF8 | 0x0032 | Current | New |  | ○ | Set the phy power on or off  0: power off  1: power on |
| eocSlaveSwitchMibInfor | Switch port | 0xF8 | 0x0033 | Current | New | ○ |  | 获取switch每个端口的流量统计信息 (42\*4 bytes)  4 bytes:The number of good broadcast framesreceived  4 bytes: The number of PAUSE frames received  4 bytes: The number of good multicast frames received  4 bytes: The total number of frames received with a valid length, but an invalid FCS and an integral number of octets  4 bytes: The total number of frames received with a balid length that do not have an integral number of octets and an invalid FCS  4 bytes: The number of frames received that are less than 64 bytes long and have a bad FCS  4 bytes: The number of frames recdeived that are less than 64 bytes long and have a bad FCS  4 bytes: The number of frames received that are exactly 64 bytes long including those with errors  4 bytes: The number of frames received whose length is between 65 and 127 bytes, including those with errors  4 bytes: The number of The number of frames received whose length is between 128and 255 bytes, including those with errors  4 bytes: The number of frames received whose length is between 256 and 511 bytes, including those with errors  4 bytes: The number of frames received whose length is between 512 and 1023 bytes, including those with errors  4 bytes: The number of frames received whose length is between 1024 and 1518 bytes, including those with errors  4 bytes: The number of frames received whose length is between 1519 and maxlength, including those with errors (Jumbo)  4 bytes: The number of frames received whose length exceeds maxlength including those with FCS errors  4 bytes: The low 32 bits of the total data octets received in a frame with a valid FCS. All frame sizes are included.  4 bytes: The high 32 bits of the total data octets received in a frame with a valid FCS. All frame sizes are included.  4 bytes: The low 32 bits of the total data octets received in frame with an invalid FCS. All frame sizes are included. Pause frame is included with a valid FCS.  4 bytes: The high 32 bits of the total data octets received in frame with an invalid FCS. All frame sizes are included. Pause frame is included with a valid FCS.  4 bytes: Total valid frames received that are discarded due to lack of buffer space  4 bytes: Port disabled and unknown VID  4 bytes: Total good frames tranmitted with a broadcast Destination address  4 bytes: Total good PAUSE frames transmitted  4 bytes: Total good frames transmitted with a multicast Destination address  4 bytes: Total valid frames discarded that were not transmitted due to transmit FIFO buffer underflow  4 bytes: Total frames tranmitted with a length of exactly 64 bytes, including errors  4 bytes: Total frames tranmitted with a length between 65 and 127 bytes, including those with errors  4 bytes: Total frames tranmitted with a length between 128 and 255 bytes, including those with errors  4 bytes: Total frames tranmitted with a length between 256 and 511 bytes, including those with errors  4 bytes: Total frames tranmitted with a length between 512 and 1023 bytes, including those with errors  4 bytes: Total frames transmitted with length between 1024 and 1518, including those with errors (Jumbo)  4 bytes: Total frames transmitted with length between 1519 and Maxlength, including those with errors (Jumbo)  4 bytes: Total frames over Maxlength but transmitted truncated with bad FCS  4 bytes: The low 32 bits of the total data octets transmitted from counted, including those with a bad FCS  4 bytes: The high 32 bits of the total data octets transmitted from counted, including those with a bad FCS  4 bytes: Total collisions experienced by a port during packet transmission  4 bytes: Total number of frames not transmitted becase the frame experienced 16 transmission attempts and was discarded  4 bytes: Total nuber of successfully transmitted fromes that experienced more than one collision  4 bytes: Total number of successfully transmitted frames that experienced exactly one collision  4 bytes: The number of frames that defered for an excessive period of time  4 bytes: Total frames whose transmission was delayed on its first attempt because the medium way was busy  4 bytes: Total number of times a collision is detedted later than 512 bit-times into the transmission of a frame |

### Switch 的VLAN配置管理

Table 5‑10Switch 的VLAN配置管理对应的Main Type和Subtype

| **属性/操作名称** | **操作对象** | **Main Type** | **Subtype** | **HomePlug AV Slave** | **QCA MME** | **Get**  **（action code= 0x02）** | **Set**  **（action code = 0x01）** | **备注** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| eocVlanVersionNumber | Switch | 0xF8 | 0x0041 | current | New | ○ |  | VLAN协议版本 (1 byte)  1 – IEEE 802.1Q |
| eocVLANMaxVID | Switch | 0xF8 | 0x0042 | current | New | ○ |  | Get the max VID（2 bytes） |
| eocVlanMaxSupportedVlans | Switch | 0xF8 | 0x0043 | current | New | ○ |  | 获取最大能支持的VLAN数目（2 bytes） |
| eocVLANCreatedVLANNumber | Switch | 0xF8 | 0x0044 | current | New | ○ |  | 已创建的VLAN数目（2bytes） |
| eocVlanVIDList | Switch port | 0xF8 | 0x0045 | current | New | ○ |  | 获取端口的VID列表:  以连续的2字节表示,每2个字节表示一个VID |
| eocVLANNextFreeVID | Switch | 0xF8 | 0x0046 | current | New | ○ |  | 下一个可用的VID（2 bytes） |
| eocSlavePortVlanConfig | Switch port | 0xF8 | 0x0047 | current | New | ○ | ○ | 配置端口VLAN id (2 bytes) |
| eocSlaveVlanDelete | Switch | 0xF8 | 0x0048 | current | New |  | ○ | 删除某个VLAN id (2 bytes) |
| eocSlavePortDefaultPVID | Switch port | 0xF8 | 0x0049 | current | New | ○ | ○ | 配置端口默认PVID  (2 bytes） |
| eocSlavePortDefaultTPID | Switch port | 0xF8 | 0x004A | optional | New | ○ | ○ | 配置端口默认的TPID（2 bytes） |
| eocVlanPortPriority | Switch port | 0xF8 | 0x004B | optional | New | ○ | ○ | Set/Get the port’s default priority (1 byte)  0 ~ 7 |
| eocSlavePortVlanMode | Switch port | 0xF8 | 0x004C | current | New | ○ | ○ | 配置出端口VLAN模式  (1 byte)  0: unmodified  1: without vlan tag  2: with vlan tag  3: hybrid mode |
| eocSlavePortDot1qMode | Port | 0xF8 | 0x004D | Current | New | ○ | ○ | 配置switch端口802.1q模式(1 byte)  0: 802.1Q disable. Use port base VLAN only.  1: fallback. Enable 802.1Q for all received frames. Don't discard ingress membership violation and use the port base VLAN if the frame's VID isn't contained in VLAN Table.  2: check. Enable 802.1Q for all received frames. Don't discard ingress membership violation but discard frames which VID isn't contained in VLAN Table.  3: secure. Enable 802.1Q for all received frames. Discard frames with ingress membership violation or whose VID isn't contained in the VLAN Table. |
| eocSlaveVlanFlush | Switch | 0xF8 | 0x004E | Current | New |  | ○ | 一次性清除switch中所有已创建的VLAN |
| eocSlaveVlanRemovePort | Switch port | 0xF8 | 0x004F | Current | New |  | ○ | 将端口从某个VLAN中移除（2 bytes）  Vlan ID |
| eocSlavePortBasedVlan | Switch | 0xF8 | 0x0050 | Current | New | ○ | ○ | 端口隔离：  1: enable  0: disable |

### Switch 的QoS相关配置

Table 5‑11Switch 的QoS相关配置对应的Main Type和Subtype

| **属性/操作名称** | **操作对象** | **Main Type** | **Subtype** | **HomePlug AV Slave** | **QCA MME** | **Get**  **（action code= 0x02）** | **Set**  **（action code = 0x01）** | **备注** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| eocSlaveQosScheduleMode | Switch port | 0xF8 | 0x0061 | Current | New | ○ | ○ | 配置switch的端口队列调度模式(1 byte)  0: strict priority  1: only highest queue use strick priority, others use weighted fair queuing schme  2: the highest two queues use strick priority, other two queues use weighted fair queuing schme.  3: all queues use weighted fair queuing schme 8:4:2:1. |
| eocSlaveQosAssignMode | Switch port | 0xF8 | 0x0062 | Current | New | ○ | ○ | 配置switch的端口QOS信任模式  第一个byte表示QoS的信任模式，第二个byte表示该模式是使能还是关闭(2 bytes).  第一个byte表示QoS模式:  0: qos assignment basedon destination macaddress  1: qos assignment based on 802.1p field in vlan tag  2: qos assignment based on dscp field in ip header  3: qos assignment based on port  第二个byte表示打开或关闭该模式：  1: enable  0: disable |
| eocSlaveQosAssignModePriority | Switch port | 0xF8 | 0x0063 | Current | New | ○ | ○ | Set priority of one particular qos mode on one particular port(2 bytes).  第一个byte表示QoS模式:  0: qos assignment basedon destination macaddress  1: qos assignment based on 802.1p field in vlan tag  2: qos assignment based on dscp field in ip header  3: qos assignment based on port  第二个byte表示该QoS模式的优先级 |
| eocSlaveQosCosMapDscpQueue | Switch | 0xF8 | 0x0064 | Current | New | ○ | ○ | Set dscp map to queue  (2 bytes)  Byte 1: dscp  Byte 2: queue id |
| eocSlaveQosCosMapToQueue | Switch | 0xF8 | 0x0065 | Current | New | ○ | ○ | Set cos map to queue  (2 bytes)  Byte 1: 802.1p priority  Byte 2: queue id |

### Switch 的组播功能相关配置

Table 5‑12Switch 的组播功能相关配置对应的Main Type和Subtype

| **属性/操作名称** | **操作对象** | **Main Type** | **Subtype** | **HomePlug AV Slave** | **QCA MME** | **Get**  **（action code= 0x02）** | **Set**  **（action code = 0x01）** | **备注** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| eocSlaveIGMPStatus | Switch port | 0xF8 | 0x0081 | Current | New | ○ | ○ | Set igmp/mld packets snooping status on a particular port.(1 byte)  0: disable  1: enable |
| eocSlaveIGMPMLDCmd | Switch | 0xF8 | 0x0082 | Current | New | ○ | ○ | Set igmp/mld packets forwarding command on a particular device.(1 byte)  2: copy to CPU  3: redirect to CPU |
| eocSlaveIGMPMLDJoin | Switch port | 0xF8 | 0x0083 | Current | New | ○ | ○ | Set igmp/mld join packets hardware acknowledgement status on particular port.(1 byte)  0: disable  1: enable |
| eocSlaveIGMPMLDLeave | Switch port | 0xF8 | 0x0084 | Current | New | ○ | ○ | Set igmp/mld leavepackets hardware acknowledgement status on a particular port.(1 byte)  0: disable  1: enable |
| eocSlaveIGMPMLDRouterPort | Switch | 0xF8 | 0x0085 | Current | New | ○ | ○ | Set igmp/mld router ports.After enabling igmp/mld join/leave feature on a particular port, igmp/mldjoin/leave packets received on this port will be forwarded to router ports.(1 byte)  Port bit map: bit 0对应port0，bit 1对应port1，以此类推。对应的bit位置1表示join/leave会发送到相应的端口。 |
| eocSlaveIGMPMLDEntryCreate | Switch | 0xF8 | 0x0086 | Current | New | ○ | ○ | Enable hardware add new address to ARL table when received IGMP/ MLD join frame and remove address from ARL when received IGMP/MLD leave frame.(1 byte)  0: disable  1: enable |
| eocSlaveIGMPMLDEntryStatic | Switch | 0xF8 | 0x0087 | Current | New | ○ | ○ | 将硬件学习到的muticast entry配置成静态entry。(1 byte)  0: disable  1: enable |
| eocSlaveIGMPMLDEntryLeaky | Switch | 0xF8 | 0x0088 | Current | New | ○ | ○ | IGMP join address leaky vlan enable. (1 byte)  1: igmp join address should be set the leaky\_en bit in ARL  0: igmp join address should be clear the leaky\_en bit in ARL table |
| eocSlaveIGMPv3Mldv2 | Switch | 0xF8 | 0x0089 | Current | New | ○ | ○ | Hardware acknowledge IGMP v3 frame and MLD v2 frame, and multicast address can be hardware join or leave. (1 byte)  0: disable  1: enable |

### 终端主动上报消息类型

该类消息由终端主动产生并上报给管理服务器，MME type=0xAE0A，其TLV对应的main type 和subtype见Table 5‑13

Table 5‑13终端主动上报消息的main type和subtype

| **属性/操作名称** | **操作对象** | **Main Type** | **Subtype** | **备注** |
| --- | --- | --- | --- | --- |
| eocSlaveLoopDetect | Switch | 0xF8 | 0x0101 | Switch 检测到有环路产生时，主动上报该消息（2 bytes）  Byte 1： 环路产生时，对应新的端口号  Byte 2： 环路产生时，对应旧的端口号 |

## 缆桥终端管理对象

### 缆桥终端Capability的Variable Container格式

Table 5‑14缆桥终端Capability

|  |  |  |
| --- | --- | --- |
| **字节数** | **字段** | **描述** |
| 1 | Main Type(0xF8) | 扩展属性操作 |
| 2 | Subtype(0x0001) | 获取终端 Capability属性操作（见Table 5‑7） |
| 1 | Variable Width | 值可变； |
| 1 | EoC Type | 枚举类型，终端设备形态：  0x00:基本型缆桥终端；  0x01:家庭网关型缆桥终端 |
| 1 | Num of Interface type | 接口类型数量 |
| 1 | Interface Type | 枚举类型，终端支持的接口类型：  0x00表示千兆以太网GE接口  0x01表示百兆以太网FE接口（最高速率为100M，不含GE口）  0x02表示VoIP接口  0x06表示WLAN  0x07表示USB口  0x08表示CATV RF口  其他待定 |
| 1 | Num of Port | 该接口类型支持的端口总数。 |
| … | … | … |
| 1 | Interface Type | 枚举类型，网关支持的接口类型：  0x00表示千兆以太网GE接口  0x01表示百兆以太网FE接口（最高速率为100M，不含GE口）  0x02表示VoIP接口  0x06表示WLAN  0x07表示USB口  0x08表示CATV RF口  其他待定 |
| 1 | Num of Port | 该接口类型支持的端口总数。 |
| 6 | Home gateway MAC | 家庭网关CPU的MAC地址（若EoC Type值为0x01时，存在该字段）。 |

注：Home gateway MAC信息应保存在EoC系统Flash的Custommodule区域，该区域对应的Custom module ID为0x1000，custom module Sub-ID为11，最多存放128字节的数据。

## 终端家庭网关芯片管理对象

### 家庭网关管理配置参数

Table 5‑15家庭网关管理

| **属性/操作名称** | **操作对象** | **Main Type** | **Subtype** | **Get** | **Set** | **备注** |
| --- | --- | --- | --- | --- | --- | --- |
| eocGwMngPara | Gateway | 0xF8 | 0x1001 | √ | √ | 建立缆桥终端家庭网关的三层管理通道，用于TR069远程管理的VLAN、IP地址、子网掩码、缺省网关、ACS、DNS等参数 |
| eocGwPortConfig | Gateway | 0xF8 | 0x1002 | √ | √ | 创建和配置缆桥终端家庭网关WAN口的业务参数 |
| eocGwWanDelete | Gateway | 0xF8 | 0x3001 |  | √ | 删除缆桥终端家庭网关WAN连接 |

Table 5‑16家庭网关管理通道参数

|  |  |  |
| --- | --- | --- |
| **字节数** | **字段** | **描述** |
| 1 | Main Type(0xF8) | 扩展属性操作 |
| 2 | Subtype(0x1001) | 家庭网关管理通道WAN连接参数 |
| 1 | Variable Width | 值可变； |
| 1 | WAN Name length | 值可变； |
| Variable Width | WAN Name | Wan连接名称（最大长度不超过32，含字符串结尾的‘\0’） |
| 2 | VLAN | 管理VLAN |
| 1 | 802.1p pBits | VLAN priority |
| 1 | User Name length | 值可变 |
| Variable Width | User Name | For ACS Access（最大长度不超过32，含字符串结尾的‘\0’） |
| 1 | Password length | 值可变 |
| Variable Width | Password | For ACS Access（最大长度不超过32，含字符串结尾的‘\0’） |
| 1 | ACS URL length | 值可变 |
| Variable Width-18 | ACS URL | ACS的URL字符串（最大长度不超过256，含字符串结尾的‘\0’） |
| 1 | Request User Name length | 值可变 (远程反向连接) |
| Variable Width | Request User Name | For ACS Romote Access（最大长度不超过32，含字符串结尾的‘\0’） |
| 1 | Request Password length | 值可变 |
| Variable Width | Request Password | For ACS Romote Access（最大长度不超过32，含字符串结尾的‘\0’） |
| 1 | Request ACS URL length | 值可变 |
| Variable Width-18 | Request ACS URL | ACS的URL字符串（最大长度不超过256，含字符串结尾的‘\0’） |
| 1 | IP Assigned Mode | DHCP/Static IP/PPPOE/ (1:DHCP,2:Static,3:PPPOE) |
| 1 | Connection Protocol | IPv4/IPv6/IPv4&IPv6 (1:IPv4,2:IPv6,3:IPv4&IPv6) |
| 1 | User Name length | 值可变 |
| Variable Width | User Name | For PPPoE, if Connection Mode is PPPoE（最大长度不超过32，含字符串结尾的‘\0’） |
| 1 | Password length | 值可变 |
| Variable Width | Password | For PPPoE, if Connection Mode is PPPoE（最大长度不超过32，含字符串结尾的‘\0’） |
| 4 | MngIPv4Addr | IPv4地址 |
| 4 | MngIPv4Mask | IPv4地址掩码 |
| 4 | Default IPv4GW | IPv4的默认网关 |
| 16 | MngIPv6Addr | IPv6地址 |
| 1 | MngIPv6Mask | IPv6地址掩码 |
| 16 | Default IPv6GW | IPv6的默认网关 |
| 4 | DNS IPv4 | DNS server IPv4地址 |
| 4 | DNS IPv4 | 备份DNS server IPv4地址 |
| 16 | DNS IPv6 | DNS server IPv6地址 |
| 16 | DNS IPv6 | 备份DNS server IPv6地址 |

Table 5‑17家庭网关业务WAN连接参数

|  |  |  |
| --- | --- | --- |
| **字节数** | **字段** | **描述** |
| 1 | Main Type(0xF8) | 扩展属性操作 |
| 2 | Subtype(0x1002) | 家庭网关业务WAN连接参数 |
| 1 | Variable Width | 值可变； |
| 1 | WAN Name length | 值可变； |
| Variable Width | Name | Wan连接名称（最大长度不超过32，含字符串结尾的‘\0’） |
| 2 | VLAN | 连接VLAN |
| 1 | 802.1p pBits | VLAN priority |
| 1 | Connection Mode | Route/Bridge(1:Route,2:Bridge,3:Route&Bridge) |
| 1 | IP Assigned Mode | DHCP/Static IP/PPPOE/ (1:DHCP,2:Static,3:PPPOE) |
| 1 | Service Type | Internet/VOD/VOIP (1:Internet,2:VOD,3:VOIP) |
| 1 | Connection Protocol | IPv4/IPv6/IPv4&IPv6 (1:IPv4,2:IPv6,3:IPv4&IPv6) |
| 4 | Bind Interface | Wan绑定的LAN侧接口  （0x00010000:SSID-1,  0x00100000:SSID-2，  0x01000000:SSID-3，  0x10000000:SSID-4,  0x00000001:LAN1,  0x00000010:LAN2,  0x00000100:LAN3,  0x00001000:LAN4,  0x00000000未绑定） |
| 1 | User Name length | 值可变 |
| Variable Width | User Name | For PPPoE, if Connection Mode is PPPoE（最大长度不超过32，含字符串结尾的‘\0’） |
| 1 | Password length | 值可变 |
| Variable Width | Password | For PPPoE, if Connection Mode is PPPoE（最大长度不超过32，含字符串结尾的‘\0’） |
| 4 | ServeIPv4Addr | IPv4地址 |
| 4 | ServeIPv4Mask | IPv4地址掩码 |
| 4 | Default IPv4GW | IPv4的默认网关 |
| 16 | ServeIPv6Addr | IPv6地址 |
| 1 | ServeIPv6Mask | IPv6地址掩码 |
| 16 | Default IPv6GW | IPv6的默认网关 |
| 4 | DNS IPv4 | DNS server IPv4地址 |
| 4 | DNS IPv4 | 备份DNS server IPv4地址 |
| 16 | DNS IPv6 | DNS server IPv6地址 |
| 16 | DNS IPv6 | 备份DNS server IPv6地址 |

Table 5‑18 家庭网关WAN连接删除

|  |  |  |
| --- | --- | --- |
| **字节数** | **字段** | **描述** |
| 1 | Main Type(0xF8) | 扩展属性操作 |
| 2 | Subtype(0x3001) | 家庭网关WAN连接删除操作 |
| 1 | Variable Width | 值可变； |
| 1 | WAN Name length | 值可变； |
| Variable Width | Name | Wan连接名称（最大长度不超过16，含字符串结尾的‘\0’） |

### 家庭网关芯片重启操作

Table 5‑19 家庭网关芯片重启操作

| **属性/操作名称** | **操作对象** | **Main Type** | **Subtype** | **备注** |
| --- | --- | --- | --- | --- |
| eocGwReset | Gateway | 0xF8 | 0x2001 | 家庭网关收到该扩展MME后，仅重启网关芯片 |

该消息无Variable Container，终端在收到网关重启的扩展MME请求后，无需向HS发送确认（Confirm）扩展MME消息，而是立即重启动家庭网关芯片。

### 缆桥终端整机重启操作

Table 5‑20 缆桥终端整机重启操作

| **属性/操作名称** | **操作对象** | **Main Type** | **Subtype** | **备注** |
| --- | --- | --- | --- | --- |
| eocCnuReset | Gateway | 0xF8 | 0x2002 | 家庭网关收到该扩展MME后，将网关芯片和EoC芯片都重启 |

该消息无Variable Container，终端在收到终端重启的扩展MME请求后，无需向HS发送确认（Confirm）扩展MME消息，而是立即重启动整机。

### 家庭网关芯片管理系统启动状态

Table 5‑21家庭网关芯片管理系统启动状态

| **属性/操作名称** | **操作对象** | **Main Type** | **Subtype** | **备注** |
| --- | --- | --- | --- | --- |
| eocGwStatus | Gateway | 0xF8 | 0x3002 | 获取家庭网关管理系统启动状态 |

该消息无Variable Container，终端在收到网关查询启动状态的扩展MME请求后，向HS发送确认（Confirm）扩展MME消息，通知HS网关终端的管理系统已准备就绪。

### 家庭网关无线SSID名称设置

Table 5‑22 家庭网关无线SSID名称设置

|  |  |  |
| --- | --- | --- |
| **字节数** | **字段** | **描述** |
| 1 | Main Type(0xF8) | 扩展属性操作 |
| 2 | Subtype(0x3003) | 家庭网关无线SSID名称设置 |
| 1 | Variable Width | 值可变； |
| 1 | SSID Num | 1:SSID-1, 2:SSID-2, 3:SSID-3, 4:SSID-4 |
| 1 | SSID Name length | 值可变； |
| Variable Width | Name | SSID名称（最大长度不超过32，含字符串结尾的‘\0’） |

### 家庭网关处理MME结果上报消息

Table 5‑23 家庭网关处理MME结果上报消息

|  |  |  |
| --- | --- | --- |
| **字节数** | **字段** | **描述** |
| 1 | Main Type(0xF8) | 扩展属性操作 |
| 2 | Subtype(0x3004) | 家庭网关处理MME 结果 上报消息 |
| 1 | Variable Width | 值可变； |
| 1 | MSTATUS | 0：success 1: fail |
| 1 | Process Type | 处理MME type(1:wan连接设置，2：SSID 名称设置)； |
| 1 | Wan Name length | 值可变； |
| Variable Width | Wan NAME | wan名称（最大长度不超过16，含字符串结尾的‘\0’） |
| 1 | SSID num | 设置的SSID num |