

# GY

## 中华人民共和国广播电视行业标准

GY/T 322.3—2019

---

### 网络音频应用的开放式控制架构 第3部分：用于TCP/IP网络的协议

Audio applications of networks - open control architecture—  
Part 3: Protocol for TCP/IP networks

2019 - 04 - 28 发布

2019 - 04 - 28 实施

---

国家广播电视总局 发布

## 目 次

前言 .....	II
引言 .....	III
0.1 概述 .....	III
0.2 文档约定 .....	III
1 范围 .....	1
2 规范性引用文件 .....	1
3 术语、定义和缩略语 .....	1
3.1 术语和定义 .....	1
3.2 缩略语 .....	1
4 最小实现 .....	1
5 协议细节 .....	1
5.1 初始化 .....	2
5.2 设备发现 .....	2
5.3 设备监管 .....	3
5.4 设备复位 .....	4
5.5 约定 .....	4
5.6 协议数据单元 .....	5
5.7 协议特定数据类型 .....	15
附录 A (资料性附录) 数据类型索引 .....	17
附录 B (资料性附录) 协议数据单元 (PDU) 的 UML 描述 .....	18
参考文献 .....	19

## 前 言

GY/T 322《网络音频应用的开放式控制架构》分为以下三部分：

- 第1部分：框架；
- 第2部分：类结构；
- 第3部分：用于TCP/IP网络的协议。

本部分为GY/T 322的第3部分。

本部分按照GB/T 1.1—2009给出的规则起草。

本部分是参照 AES70-3-2015《网络音频应用的开放式控制架构 第3部分：用于TCP/IP网络的协议》编制的。

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

本部分由全国广播电影电视标准化技术委员会(SAC/TC 239)归口。

本部分起草单位：中央广播电视总台、国家广播电视总局广播电视科学研究院、国家广播电视总局广播电视规划院、江苏省广播电视总台、浙江广播电视集团、苏州市福川科技有限公司、北京英夫美迪科技股份有限公司、北京众和传新科技有限公司、杭州联汇科技股份有限公司、上海佰贝科技发展有限公司、北京捷成世纪科技股份有限公司、苏州大学。

本部分主要起草人：钱岳林、朱峰、罗攀、潘宇、张磊、王兰岚、庞超、唐峰、张伟、邓向冬、董升来、何晶、孙岩君、李维民、陈武、董晓坡、陈沁、唐卫平、陈立德、赵崇峰、肖仲喆。

# 引 言

## 0.1 概述

本部分支持在TCP/IP网络中实现符合开放式控制架构的媒体设备的远程监控。

开放式控制架构的第1部分是参照AES70-1-2015《网络音频应用的开放式控制架构 第1部分：框架》编制的，英文原文可从<http://www.aes.org/publications/standards/search.cfm?docID=101>下载。

开放式控制架构的第2部分定义了用于媒体网络监控的开放式控制架构的类结构。第2部分是参照AES70-2-2015《网络音频应用的开放式控制架构 第2部分：类结构》编制的，英文原文可从<http://www.aes.org/publications/standards/search.cfm?docID=102>下载。

开放式控制架构的第3部分是参照AES70-3-2015《网络音频应用的开放式控制架构 第3部分：用于TCP/IP网络的协议》编制的，英文原文可从<http://www.aes.org/publications/standards/search.cfm?docID=103>下载。

## 0.2 文档约定

本部分涉及的通用数据类型适用于所有符合开放式控制架构的协议，特定数据类型只适用于本部分。为了便于区分，通用数据类型的名称使用“Oca”前缀，而特定数据类型的名称使用“Ocp1”前缀。

# 网络音频应用的开放式控制架构

## 第3部分：用于TCP/IP网络的协议

### 1 范围

GY/T 322的本部分规定了用于TCP/IP网络的协议。  
本部分适用于网络音频应用的监控。

### 2 规范性引用文件

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

GY/T 322.1—2019 网络音频应用的开放式控制架构 第1部分：框架

GY/T 322.2—2019 网络音频应用的开放式控制架构 第2部分：类结构

IETF RFC 3927 IPv4本地链路地址动态配置 (Dynamic Configuration of IPv4 Link-Local Addresses)

IETF RFC 4862 IPv6无状态地址自动配置 (IPv6 Stateless Address Autoconfiguration)

IETF RFC 6335 互联网数字分配机构 (IANA) 为服务名称和传输协议端口号注册管理程序 (Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry)

IETF RFC 6762 组播DNS (Multicast DNS)

IETF RFC 6763 基于DNS的服务发现 (DNS-Based Service Discovery)

### 3 术语、定义和缩略语

#### 3.1 术语和定义

下列术语和定义适用于本文件。

##### 3.1.1

**开放式控制协议** open control protocol; OCP

依据开放式控制架构定义的网络协议。

#### 3.2 缩略语

下列缩略语适用于本文件。

PDU 协议数据单元 (Protocol Data Unit)

### 4 最小实现

每个符合开放式控制架构的设备应实现本部分的全部内容。  
本部分包含的特定可选项见第5章。

### 5 协议细节

## 5.1 初始化

### 5.1.1 IP 地址初始化

在设备初始化OcaNetwork或OcaStreamNetwork对象（见GY/T 322.2—2019）时，应进行5.1.2、5.1.3、5.1.4所述的初始化步骤。

上述对象的ControlProtocol属性值应为“OCP01”。

### 5.1.2 IP 地址分配方法

符合开放式控制架构的设备至少应实现IPv4或IPv6网络寻址标准。在本部分，实现IPv4网络寻址的符合开放式控制架构的设备称为IPv4设备。实现IPv6网络寻址的符合开放式控制架构的设备称为IPv6设备。设备可同时实现IPv4和IPv6，即它可同时是IPv4设备和IPv6的设备。

每个IPv4设备宜实现DHCP客户端并使用DHCP服务器。每个IPv6设备宜实现DHCPv6客户端并使用DHCPv6服务器。在下文中，这些客户端和服务器将分别统称为IP地址客户端和IP地址服务器。

如果设备属于多个IP子网，它宜为每个子网设置一个IP地址客户端。当设备连接一个子网时，设备宜启动对应的IP地址客户端。

如果在地址分配超时时段内IP地址客户端连接到IP地址服务器，则该设备应使用该服务器分配的地址。当在地址分配超时时段内未发现IP地址服务器，或设备未实现IP地址客户端：

- a) IPv4 设备宜使用在 IETF RFC 3927 中定义的 IPv4 本地链路地址；
- b) IPv6 设备宜使用在 IETF RFC 4862 中定义的由 IPv6 自动生成的 IPv6 本地链路地址。

当设备未实现本地链路地址时，应手工设置IP地址。

### 5.1.3 套接字和端口

获得IP地址后，设备应建立一个TCP监听套接字接收不安全的符合本部分的会话，或一个TCP监听套接字接收安全的符合本部分的会话，或两者同时打开。

设备应使用标准IANA动态端口范围（49152到65535，见IETF RFC 6335）内的TCP端口号。在该范围内，设备可以把不安全的监听套接字绑定到任何可用的TCP端口，以及安全的监听套接字绑定到其他可用的TCP端口。这些端口应公告，见5.2.2。

## 5.2 设备发现

### 5.2.1 概述

设备发现，是连接到网络的符合开放式控制架构的设备使自身被公共访问目录服务获知的机制，也是网络中的其他设备可使用目录服务查找和寻址设备的机制。

符合开放式控制架构的设备的发现进程应具备服务发现架构，其中符合开放式控制架构的设备应到网络服务目录中自行注册，需要获悉该设备IP地址的网络实体可通过此目录获取到。

服务发现应通过基于DNS的服务发现实现（见IETF RFC 6763）。

注：“发现”一词的另一个常见用法是指设备功能的发现。在开放式控制架构中，功能发现是由设备的根块和内部块（如果有）枚举的方法实现。这样的枚举是正常开放式控制架构的命令响应序列，对网络类型没有特殊的依赖性。因此，它们不在本部分的范围内。有关详细信息，见GY/T 322.1—2019和GY/T 322.2—2019的OcaBlock类的详细说明。

### 5.2.2 服务发现

如果一个符合开放式控制架构的设备打开了一个监听套接字来建立符合本部分的不安全连接，应注册成以下服务类型：

`_oca._tcp`

如果一个符合开放式控制架构的设备打开了一个监听套接字来建立符合本部分的安全连接，应注册成以下服务类型：

\_ocasec.\_tcp

对安全和不安全的服务，注册的服务名称应与连接所用的OcaNetwork或OcaStreamNetwork对象NameAdvertised属性相同。如果该名称变更，设备应注销旧服务，并使用新名称注册新服务。

注册可在任何期望的域中进行，在大多数应用中，建议在本地域注册。

在本地域注册应使用组播DNS（mDNS）协议（见IETF RFC 6762）。

当在本地域注册，服务名称冲突是由DNS组播协议自动解决。当通过组播DNS改变服务名称以避免冲突时，服务名称已更改的设备应自动更新NameAdvertised属性。

注：在非本地域注册时，名称冲突不会自动解决。因此，如果符合开放式控制架构的设备可能在非本地域注册，宜选择唯一的缺省服务名称。

为服务注册的端口应与该设备在5.1.3中选择的端口相一致。

遵循IETF RFC 6763第6章版本标签的建议，不安全和安全注册的TXT记录应至少包含两个键/值对。第一个键/值对应为：

txtvers=1 [OCA service registration version]

第二键/值对包含开放式控制架构的版本，应按照以下格式：

protovers=x

x是设备OcaDeviceManager对象指定的开放式控制架构版本十进制数（见GY/T 322.2—2019）。

TXT记录可包含更多的数据，只要记录包含上面提到的两个键/值对，且数据符合在IETF RFC 6763第6章阐述的规则。TXT记录的开始字段如图1所示。如果开放式控制架构的十进制版本大于9，第二长度字段应为0C<sub>16</sub>。

09 <sub>16</sub>	txtvers=1	0B <sub>16</sub>	protovers=x
------------------	-----------	------------------	-------------

图1 TXT 记录起始字段

控制器可以在所需域内通过执行一个DNS-SD服务浏览发现网络中的符合开放式控制架构的设备，查找“\_oca.\_tcp”服务或“\_ocasec.\_tcp”服务，或两者。

在本地域中浏览应使用组播DNS（见IETF RFC 6762）。

### 5.3 设备监管

#### 5.3.1 概述

设备监管意味着对网络上的设备可用性进行相对持续（通常是周期性的）验证。符合本部分定义的设备监管机制，可用来监管连接的符合开放式控制架构的设备。

#### 5.3.2 规范

每个符合开放式控制架构的设备均应实现监管机制；按照应用要求，控制器可以启用或禁用该机制。

在设备建立安全或不安全的连接后的任何时刻，控制器可通过向设备发送KeepAlive消息（见5.6.5）启动设备监管进程。从那一刻直到断电或设备复位，该设备和控制器应使用HeartbeatTime值以确保设备和控制器每心跳时间（HeartbeatTime）发送一个消息。该消息可以是KeepAlive消息或其他消息。

HeartbeatTime值应在KeepAlive消息中指定，并可随时变更。设备应为不同的连接支持不同HeartBeatTime值。

一旦监管进程启动，对已建立的连接，控制器和设备应记录收到符合本部分的消息之间的时间间隔。如果控制器或设备未接收消息的时间长度等于三倍HeartBeatTime值，应认为连接丢失，控制器或设备应关闭该连接。关键开放式控制架构的应用应使用KeepAlive机制，而不是靠TCP/IP检测连接丢失。

示例：

如果控制器发送的第一个KeepAlive消息中HeartBeatTime值为2秒，控制器和设备必须每两秒发送一个消息。如果6秒没有收到消息，设备和控制器将认为连接丢失。

注：如果控制器在建立连接后未发送 KeepAlive 消息，则该连接不启动设备监管机制。在这种情况下，除非检测到 TCP 保持激活机制，否则不会对空闲连接进行连接丢失检测（即连接没有控制流量）。典型参数设置条件下的 TCP 保持激活机制的检测超时时长超长，是不可接受的，往往需要数小时。此外，并非所有的 TCP/IP 协议栈都能正确地实现保持激活机制。

当连接丢失时，如果可能的话，控制器和设备应执行适当的终止处理。应删除连接上的锁和订阅，并清除连接信息。

## 5.4 设备复位

### 5.4.1 概述

符合开放式控制架构的设备可实现设备复位机制。

### 5.4.2 未实现复位

如果设备未实现复位机制，它应以NotImplemented状态响应SetResetKey消息，且不执行其他操作。否则，根据5.4.3规定的操作。

### 5.4.3 实现复位

上电复位后，设备应禁用复位机制。若要启用复位机制，控制器应先调用设备的OcaDeviceManager对象的SetResetKey方法。当调用SetResetKey时，控制器应传递以下参数：

ResetKey：一个128bit设备复位校验码；

ResetAddress：设备应监听DeviceReset消息（见5.6.6）的OcaNetworkAddress（见5.7.2）。该地址应包含一个UDP端口号的数据类型，以及，可选的一个IPv4或IPv6组播地址。

当收到SetResetKey的消息，该设备应配置相关机制，在ResetAddress参数指定的端口打开UDP套接字。如果ResetAddress参数同时指定一个组播地址，设备应加入指定的组播组。

如果收到多个SetResetKey消息，应使用最新的SetResetKey消息给出的参数。

一旦复位机制配置完毕，设备应监听用于获得包含给定设备复位校验码的DeviceReset消息的UDP套接字。如果接收到这样一个复位消息，设备应执行上电复位。如果给定的ResetAddress不包含一个组播地址，复位消息将通过指定的端口直接发送到设备上。如果ResetAddress包含一个组播地址，复位消息将直接发送到该设备的指定端口或组播组的指定端口。

如果收到的DeviceReset消息包含指定消息以外的复位校验码值，消息应被忽略并不执行复位操作。

设备复位或关机后，设备应解除其复位机制。该机制可由上述过程重新配置。

注：如果设备的复位校验码是通过不安全的开放式控制协议（OCP）连接发送，可能会被截获并被恶意使用，导致系统破坏。当需要在有破坏威胁的环境下使用设备复位机制时，SetResetKey消息只宜通过安全的OCP连接发送。

实现设备复位机制的设备宜提供手动方式（例如：开关、跳线或面板命令）来禁用该功能。

## 5.5 约定

### 5.5.1 字节顺序

对于基本的符合开放式控制架构的整型数据类型和字长超过1个字节的浮点数据类型，本部分应使用网络字节顺序（即大端或高位优先）。

### 5.5.2 封装规则

OcaBoolean类型应封装为单字节，其中值0表示的布尔值false，所有其他的值表示的布尔值true。

在GY/T 322.2—2019中定义的复合数据类型（即转化为网络字节流）的单条属性应使用GY/T 322.2—2019中的数据类型定义进行封装。

在GY/T 322.2—2019中定义的接口适用于以下封装规则：

——类方法输入参数的数量应通过Ocp1Command中的Ocp1Parameters结构中的parameterCount属性传递，见5.6.2；



- 方法的输入参数应按照 GY/T 322.2—2019 规定的顺序, 封装在 Ocp1Command 中的 Ocp1Parameters 结构中, 见 5.6.2;
- 类方法输出参数的数量应通过 Ocp1response 中的 Ocp1Parameters 结构中的 parameterCount 属性传递, 见 5.6.3;
- 方法输出参数应通过 GY/T 322.2—2019 规定的顺序, 封装在 Ocp1Response 中的 Ocp1Parameters 结构中, 见 5.6.3;
- 事件参数的数量应加到 Ocp1Notification 中的 Ocp1Ntfparams 结构中 parameterCount 属性;
- 事件参数应通过 GY/T 322.2—2019 规定的顺序, 封装在 Ocp1Notification 中的 Ocp1NtfParams 结构中的 Ocp1EventData 结构中, 见 5.6.4。

注: 本部分使用的数据类型索引参见附录 A。

### 5.5.3 示例

考虑 OcaClassIdentification 复合数据类型, 在 OCC 定义如下:

```
OcaClassIdentification = { OcaClassID ClassID,
                          OcaClassVersionNumber
                          ClassVersion }
OcaClassID = { OcaUint16 FieldCount, OcaUint16[] Fields }
OcaClassVersionNumber = { OcaUint16 Value }
```

现在考虑一个具有以下值的特定类标识实例:

```
OcaClassIdentification classIdentification = {
    ClassID = { 2, { 1, 3 } },
    ClassVersion = 1 }
```

通过上述封装规则, 在网络中实例将表示为以下字节序列 (十进制值, 左侧的先发送):

0, 2, 0, 1, 0, 3, 0, 1

## 5.6 协议数据单元

### 5.6.1 消息格式

#### 5.6.1.1 概述

每个符合本部分的消息格式如图2所示。

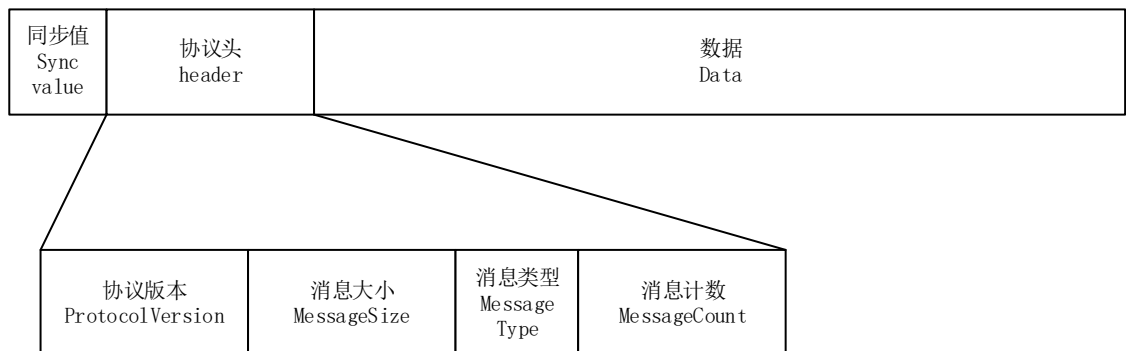


图2 通用消息格式

消息协议数据单元的C语言风格定义如下:

```
struct {
    OcaUint8    syncVal;    // 消息同步值
```

```
Ocp1Header    header;    // OCP 包头
OcaUint8      data[];    // 消息数据
} Ocp1MessagePdu;
参数定义见表1。
```

表1 消息协议数据单元参数定义

参数	定义
syncVal	消息同步值指示一个新符合本部分的消息的开始。同步值应为常量 3B <sub>16</sub>
header	本部分包头包含通用消息字段
data	存放实际消息数据的数据数组

接收器应始终检查每一个以同步值开始的符合本部分的消息。如果接收到的消息不以标准同步值开始，则接收器应关闭与发送器的连接。

注：连接断开之后，所涉及的控制器的可以选择重启连接。开放式控制架构没有定义标准的恢复程序。

协议数据单元可由GB/T 28167—2011中定义的XML元数据交换（XMI）2.1格式的通用建模语言（UML）文档定义。相关文件的获取途径参见附录B。

### 5.6.1.2 包头

符合本部分的数据结构的C语言风格定义如下：

```
struct {
    OcaUint16    protocolVersion;    // 版本号
    OcaUint32    pduSize;            // PDU 字节长度
    OcaMessageType pduType;          // PDU 类型
    OcaUint16    messageCount;       // 消息数量
} Ocp1Header;
参数定义见表2。
```

表2 包头参数定义

参数	定义	
protocolVersion (长度为 2 字节)	本部分的协议版本号。开放式控制架构的类有自己的版本号，因此只有在本部分中描述的本部分的协议更新时，协议版本才更新	
pduSize (长度为 4 字节)	完整 PDU 的字节长度，包括包头，但不包括同步值	
pduType (长度为 1 字节)	表示 PDU 的类型；即，包含的 PDU 消息类型。应使用下列消息类型之一（值在括号之间）：	
	OcaCmd (0)	命令-不需要响应
	OcaCmdRrq (1)	命令-需要响应
	OcaNtf (2)	通知
	OcaRsp (3)	响应（对应命令或通知）
messageCount (长度为 2 字节)	消息数量，显示在数据字段有多少消息（pduType 类型）。此消息数量应至少为 1。如果 pduType 等于 OcaKeepAlive，消息数量应为 1	

## 5.6.2 命令消息

### 5.6.2.1 格式

命令消息格式如图3所示。

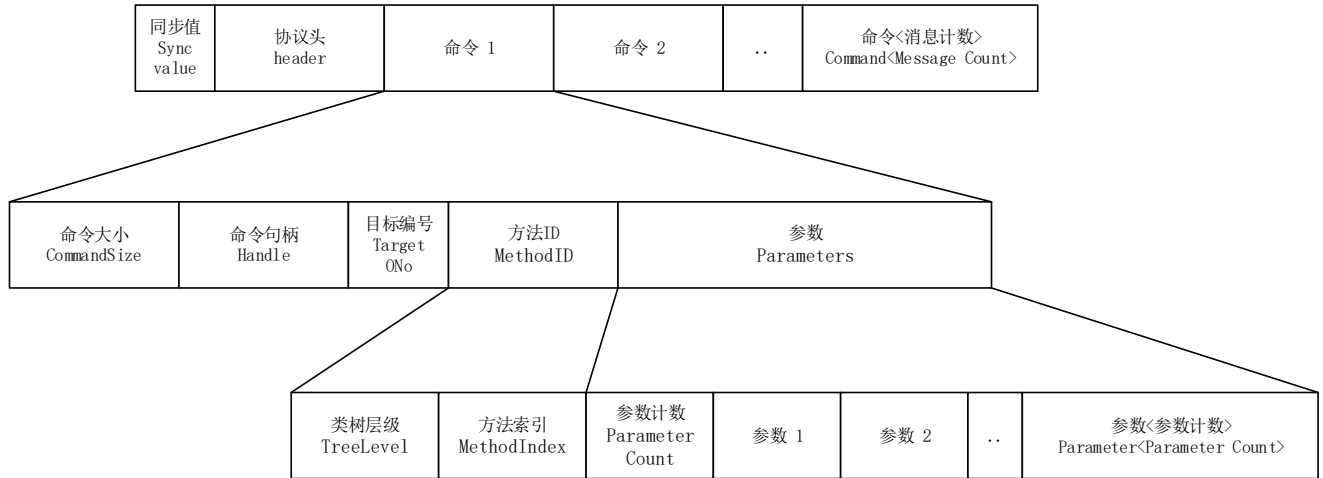


图3 命令消息

命令消息协议数据单元的C语言风格定义如下：

```

struct {
    OcaUInt8      syncVal;           // 消息同步值
    Ocp1Header    header;           // OCP 包头
    Ocp1Command   commands[messageCount]; // 命令数组
} Ocp1CommandPdu;
    
```

参数定义见表3。

表3 命令消息参数定义

参数	定义
syncVal (长度为 1 字节)	消息同步值 (见 5.6.1.1)
header (长度为 9 字节)	通用消息字段。见 5.6.1.2 的 Ocp1Header 定义
commands (可变长度)	命令 (messageCount) 数组。命令格式由 Ocp1Command 类型定义，见 5.6.2.2

5.6.2.2 Ocp1Command

Ocp1Command数据结构的C语言风格定义如下：

```

struct {
    OcaUInt32      commandSize;     // 单条命令的长度
    OcaUInt32      handle;          // 命令句柄
    OcaONo         targetONo;       // 在控制器设备中的目的对象编号
    OcaMethodID    methodID;        // 被调用方法的 ID
    Ocp1Parameters parameters;      // 被调用消息的输入参数
}Ocp1Command;
    
```

参数定义见表4。

表4 Ocp1Command 参数定义

参数	定义
commandSize (长度为 4 字节)	单条命令的长度，以字节为单位。应包含 commandSize 字段完整的 Ocp1Command 结构的长度
handle (长度为 4 字节)	强制使用 32bit 作为对于命令引用“handle”。响应应使用同样的 handle 值。控制器可自由地为命令分配 handle 值，设备应在对应的响应中匹配这些 handle 值。在控制器和设备之间，对于每个 TCP 会话应用私有 handle
target0No (长度为 4 字节)	在控制器设备中的目的对象编号(Oca0No)。Oca0No 应使用 4 字节（即一个 OcaUint32）
methodID (长度为 4 字节)	被调用方法的 ID (OcaMethodID)（即目标对象调用的方法），见 5.6.2.3
parameters (可变长度)	被调用消息的输入参数。如果被调用的方法没有任何参数，该结构应保留

5.6.2.3 OcaMethodID

OcaMethodID数据结构的C语言风格定义如下：

```
struct {
    OcaUint16 treeLevel; //类树层级
    OcaUint16 methodIndex; //方法的索引
} OcaMethodID;
元素定义见表5。
```

表5 OcaMethodID 元素定义

元素	定义
treeLevel (长度为 2 字节)	在本方法中定义的类树层级 (1=root)
methodIndex (长度为 2 字节)	方法的索引。定义了方法的类应决定索引值。对于每个树层级，索引应起始于 1

类元素分类详见GY/T 322.1—2019。

5.6.2.4 Ocp1Parameters

Ocp1Parameters数据结构的C语言风格定义如下：

```
struct {
    OcaUint8 parameterCount; //参数数量
    OcaUint8 parameters[]; //参数数组
} Ocp1Parameters;
元素定义见表6。
```

表6 Ocp1Parameters 元素定义

元素	定义
parameterCount (长度为 1 字节)	在参数数组中的参数数量。若没有参数数据, 该值可为 0
parameters (可变长度)	命令实际(封装后)参数数组。参数应配置特定的调用方法。对于每个类方法专门的参数定义见 GY/T 322.2—2019

### 5.6.3 响应消息

#### 5.6.3.1 格式

响应消息格式如图4所示。

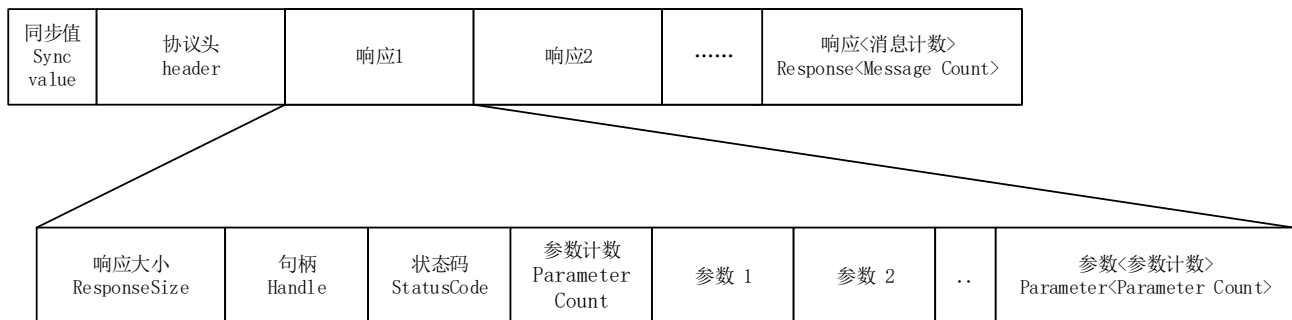


图4 响应消息

响应消息协议数据单元的C语言风格定义如下:

```
struct {
    OcaUInt8      syncVal;           // 消息同步值
    Ocp1Header    header;           // OCP 包头
    Ocp1Response  responses[messageCount]; // 响应数组
} Ocp1ResponsePdu;
元素定义见表7。
```

表7 响应消息元素定义

元素	定义
syncVal (长度为 1 字节)	消息同步值, 见 5.6.1.1
header (长度为 9 字节)	通用消息字段, 见 5.6.1.2
responses (可变长度)	响应数组 (messageCount)。响应格式由 Ocp1Response 数据类型定义, 见 5.6.3.2

#### 5.6.3.2 Ocp1Response

Ocp1Response数据结构的C语言风格定义如下:

```
struct {
    OcaUInt32      responseSize;    // 单条响应的长度
    OcaUInt32      handle;         // 响应句柄
}
```

```

    OcaStatus      statusCode;    // 状态代码表示响应所述方法的调用结果
    Ocp1Parameters parameters;    // 状态参数
} Ocp1Response;

```

元素定义见表8。

表8 Ocp1Response 元素定义

元素	定义
responseSize (长度为4字节)	单条响应的长度(以字节为单位)。应包含 responseSize 字段完整的 Ocp1Response 结构的长度
handle (长度为4字节)	强制使用 32bit 作为对于命令引用“handle”。响应应使用与触发该响应命令同样的 handle 值
statusCode (长度为1字节)	状态代码表示响应所属方法调用结果。状态代码值定义见 GY/T 322.2—2019
parameters (可变长度)	状态参数(调用方法的输出参数)。若方法没有任何输出参数, parameterCount 应为 0。Ocp1Parameters 参数数据类型定义见 5.6.2.4

#### 5.6.4 通知消息

##### 5.6.4.1 格式

通知消息格式如图5所示。

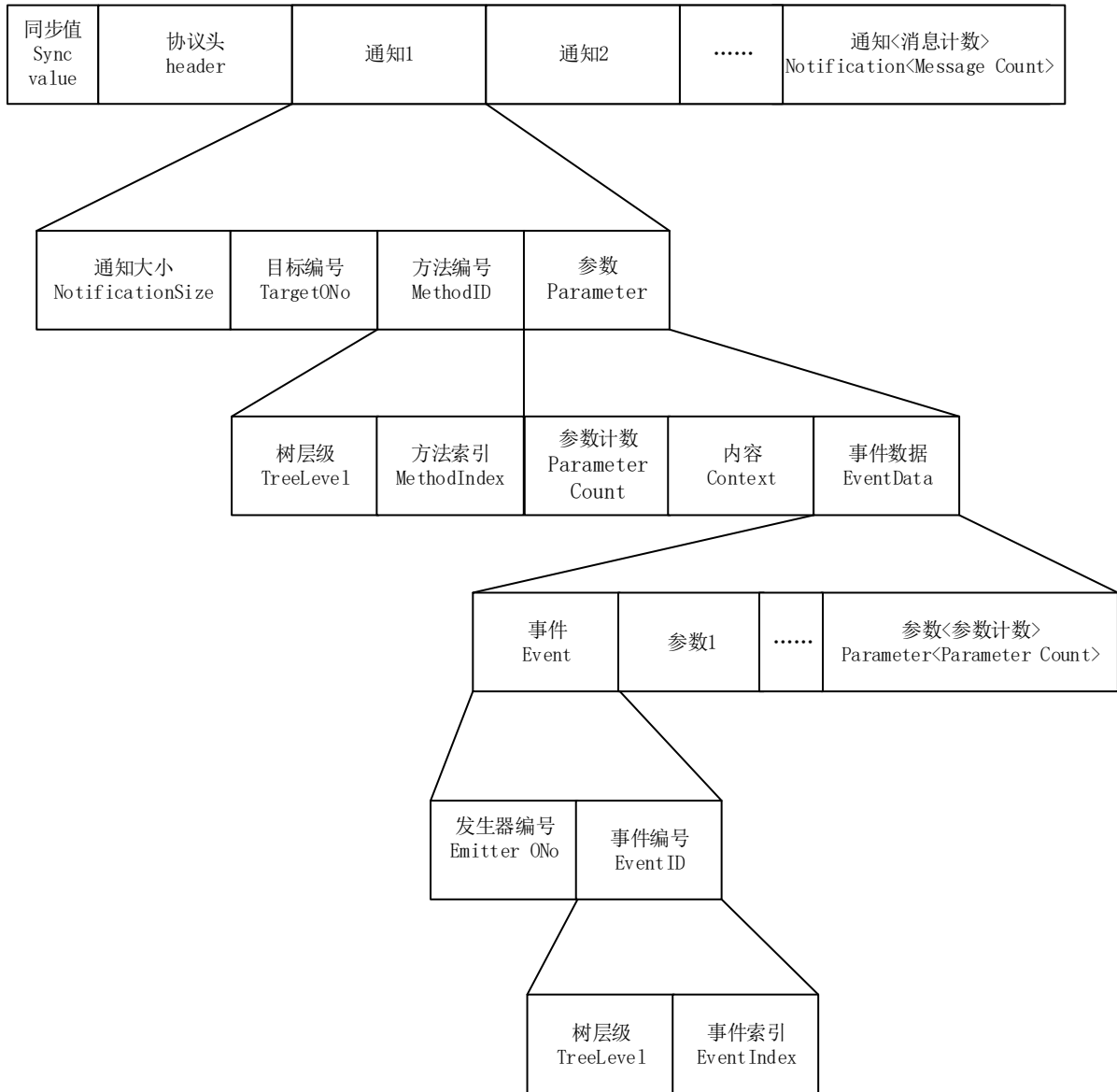


图5 通知消息

通知消息协议数据单元的C语言风格定义如下：

```

struct {
    OcaUint8          syncVal;           // 消息同步值
    Ocp1Header        header;           // OCP 包头
    Ocp1Notification notifications[messageCount]; // 通知数组
} Ocp1NotificationPdu;

```

元素定义见表9。

表9 通知消息元素定义

元素	定义
syncVal (长度为1字节)	消息同步值, 见 5.6.1.1
header (长度为9字节)	通用消息字段, 见 5.6.1.2
notifications (可变长度)	通知(messageCount)数组, 通知格式由 Ocp1Notification 数据类型定义, 见 5.6.4.2

5.6.4.2 Ocp1Notification

Ocp1Notification 数据结构的C语言风格定义如下:

```

struct {
    OcaUInt32    notificationSize; // 单条通知的长度
    OcaONo      targetONo        // 目标对象编号
    OcaMethodID methodID;        // 调用对应回调函数的方法 ID
    Ocp1NtfParams parameters;    // 事件的参数
}Ocp1Notification;
    
```

元素定义见表10。

表10 Ocp1Notification 元素定义

元素	定义
notificationSize (长度为4字节)	单条通知长度(以字节为单位), 应包含 notificationSize 字段的完整 Ocp1Notification 结构长度
targetONo (长度为4字节)	目标对象编号, 定义回调函数的事件处理程序对象编号
methodID (长度为4字节)	当事件触发时, 调用对应回调函数的方法 ID。见 5.6.2
parameters (可变长度)	事件的参数。每个通知消息至少应包含一个参数。参数格式由 Ocp1NtfParams 数据类型定义, 见 5.6.4.3

5.6.4.3 Ocp1NtfParams

Ocp1NtfParams 数据结构的C语言风格定义如下:

```

struct {
    OcaUInt8    parameterCount; // 参数数量
    OcaBlob     context;        // 订阅该事件时传递的任意值
    Ocp1EventData eventData;   // 事件的数据
} Ocp1NtfParams;
    
```

元素定义见表11。



表11 Ocp1NtfParams 元素定义

元素	定义
parameterCount (长度为 1 字节)	出现在参数数组的参数数量。每个通知消息应至少包含触发该事件的 OcaEvent (见 5.6.4.5), 该数量至少应为 1。如果事件包含额外事件参数, 数量应超过 1
context (长度为 4 字节)	订阅器在订阅该事件时传递的任意值。该值应按原值传回
eventData (可变长度)	事件的数据, 见 5.6.4.4

#### 5.6.4.4 eventData

Ocp1EventData数据结构的C语言风格定义如下:

```
struct {
    OcaEvent    event;           // OcaEvent 表示触发该通知的事件
    OcaUInt8    eventParameters[]; // 用于保存事件的其他参数的参数数组
} Ocp1EventData;
元素定义见表12。
```

表12 eventData 元素定义

元素	定义
event	OcaEvent 表示触发该通知的事件, 见 5.6.4.5
eventParameters	用于保存事件的其他参数(如果有)的参数数组。通过触发通知事件的类型参数被确定, 并由 GY/T 322.2—2019 定义 如果事件没有任何参数, 该数组不存在, 并且在 Ocp1NtfParams 的 parameterCount 值应为 1

#### 5.6.4.5 OcaEvent

OcaEvent数据结构的C语言风格定义如下:

```
struct {
    OcaONo    emitterONo; // 触发事件的发送器对象编号
    OcaEventID eventID;   // 在该消息中通知事件 ID
} OcaEvent;
元素定义见表13。
```

表13 OcaEvent 元素定义

元素	定义
emitterONo (长度为 4 字节)	触发事件的发送器对象编号
eventID (长度为 4 字节)	在该消息中通知事件的事件 ID。事件 ID 的格式由 OcaEventID 数据类型定义, 见 5.6.4.6

#### 5.6.4.6 OcaEventID

OcaEventID数据结构的C语言风格定义如下:

```
struct {
    OcaUInt16 treeLevel; // 类树层级
```

```
OcaUInt16 eventIndex; // 事件的索引
} OcaEventID;
元素定义见表14。
```

表14 OcaEventID 元素定义

元素	定义
treeLevel (长度为2字节)	在本方法中定义的类树层级 (1=root)
eventIndex (长度为2字节)	事件的索引。在事件中定义类事件应定义索引值

类元素分类见GY/T 322.1—2019。

### 5.6.5 Keep-Alive 消息

Keep-Alive消息格式如图6所示。



图6 保持激活消息

Keep-Alive消息协议数据单元C语言风格定义如下：

选项1:

```
struct {
    OcaUInt8    syncVal;           // 消息同步数值
    Ocp1Header  header;           // 包头 (见 5.6.1.2)
    OcaUInt16   heartBeatTime;    // 消息时间间隔 (以秒为单位)
} Ocp1KeepAlivePdu;
```

选项2:

```
struct {
    OcaUInt8    syncVal;           // 消息同步数值
    Ocp1Header  header;           // 包头 (见 5.6.1.2)
    OcaUInt32   heartBeatTime;    // 消息时间间隔 (以毫秒为单位)
} Ocp1KeepAlivePdu;
```

元素定义见表15。

表15 Keep-Alive 消息元素定义

元素	定义
syncVal (长度为1字节)	消息同步数值, 见 5.6.1.1
header (长度为9字节)	通用消息字段, 见 5.6.1.2
heartBeatTime	在该符合本部分的链路发送 keepAlive 消息时间间隔, 见 5.3
选项 1 (长度为2字节)	以秒 (s) 为单位
选项 2 (长度为4字节)	以毫秒 (ms) 为单位

设备应通过测试消息长度区分选项1和选项2。

### 5.6.6 设备复位消息

DeviceReset消息格式如图7所示。

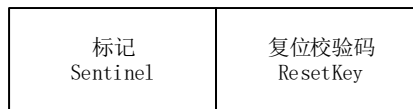


图7 DeviceReset 消息

Sentinel应为一个十六进制64bit常量:

DEAF DADA CAFE BABE

复位校验码长度应为128bit (8字节)。复位校验码数值应由设备管理器的方法设置, 见GY/T 322.1—2019。

## 5.7 协议特定数据类型

### 5.7.1 概述

5.7描述了用于符合本部分的协议特定数据类型。在GY/T 322.2—2019中, 协议特定数据类型用于定义通用OcaBlob元素。5.7定义了这些OcaBlob元素在本部分特定数据类型的映射。

GY/T 322.2—2019定义的OcaBlob仅可拥有一个命名为Data的属性。5.7描述如何理解在本部分中Data属性的值。

### 5.7.2 OcaNetworkAddress

对于本部分中OcaNetworkAddress数据类型应理解为:

```
struct {
    OcaString    ipAddress; // IPv4 或 IPv6 地址
    OcaUint16    port;      // IP 端口
} Ocp1NetworkAddress;
```

字符串类型ipAddress表示IPv4 或 IPv6地址。端口应为16bit无符号整型, 表示网络地址的端口。

IPv4地址应使用十进制, 点为分隔符 (如'192.168.1.1')。

IPv6地址应使用十六进制, 冒号为分隔符 (如'2001:0ba0::')。

### 5.7.3 OcaNetworkHost

对于本部分中OcaNetworkHost数据类型应理解为：

```
struct {
    OcaString  hostName;    // 网络主机的主机名
} Ocp1NetworkHost;
```

其中hostName应为字符串格式，表示网络主机的主机名称。

#### 5.7.4 OcaNetworkSystemInterfaceID

对于本部分中OcaNetworkSystemInterfaceID数据类型的SystemInterfaceHandle属性应理解为：

```
struct {
    OcaUInt32      interfaceIndex;    // 系统接口的索引
    OcaUInt8       subnetMaskLength;  // 系统接口的子网掩码的长度
    OcaString      defaultGateway;    // 系统接口的缺省网关地址
    OcaString      dnsServer;         // 系统接口 DNS 服务器地址
    OcaString      dnsDomainName;    // 系统接口 DNS 域名
    OcaBoolean     linkUp;           // 指示系统接口链路是否在线
    OcaUInt64      adapterSpeed;     // 系统接口适配器速率
    Ocp1IPParametersType parametersType; // 信息源
    OcaBlobFixedLen<6> macAddress;    // MAC 地址
} Ocp1NetworkSystemInterfaceID;
```

元素定义见表16。

表16 OcaNetworkSystemInterfaceID 元素定义

元素	定义	
interfaceIndex	系统接口的索引	
subnetMaskLength	系统接口对应网络 IP 地址的子网掩码长度	
defaultGateway	系统接口缺省网关（IP 地址为字符串）	
dnsServer	系统接口 DNS 服务器（IP 地址为字符串）	
dnsDomainName	系统接口 DNS 域名	
linkUp	指示系统接口（以太网）链路是否在线	
adapterSpeed	当前系统接口适配器速率，以比特每秒（bps）为单位（如 1000000 对应为 1Gbps 链路）	
parametersType	指示 IP 参数类型，信息源，应使用以下至少一个参数类型（数值在括号中给出）：	
	Unknown (0)	信息源未知
	LinkLocal (1)	网络信息通过链路本地寻址分配
	DHCP (2)	网络信息通过 DHCP 分配
	Static (3)	网络信息通过静态分配
macAddress	6 字节二进制大对象表示系统接口的 MAC 地址	

附 录 A  
(资料性附录)  
数据类型索引

表A.1列出了开放式控制架构使用的每一个数据类型，并给出了简介和在本部分章条编号。

表A.1 数据类型

数据类型	简介	章条编号
Ocp1MessagePdu	通用消息协议数据单元	5.6.1.1
Ocp1Header	每个本部分消息的包头	5.6.1.2
Ocp1CommandPdu	命令消息协议数据单元（可包含多个单条命令）	5.6.2.1
Ocp1Command	单条命令，可是 Ocp1CommandPdu 的一部分	5.6.2.2
Ocp1Parameters	包含命令/响应消息参数的参数结构。对一个命令是输入方法参数的集合。对于一个响应是输出方法参数的集合	5.6.2.4
Ocp1ResponsePdu	响应消息协议数据单元（可包含多个单条响应）	5.6.3.1
Ocp1Response	单条响应，可为 Ocp1ResponsePdu 的一部分	5.6.3.2
Ocp1NotificationPdu	通知消息协议数据单元（可包含多个单条通知）	5.6.4.1
Ocp1Notification	单条通知，可为 Ocp1NotificationPdu 的一部分	5.6.4.2
Ocp1NtfParams	通知消息参数结构（包含事件的参数）。该参数结构至少包含事件自身。如果事件包含额外的参数，那也将是该结构的一部分	5.6.4.3
Ocp1EventData	每个数据结构包含事件自身（这样事件接收器知道是哪个事件触发的通知）和事件参数（如果有）	5.6.4.4
Ocp1KeepAlivePdu	Keep-alive 消息协议数据单元	5.6.5
Ocp1NetworkAddress	本部分对更为通用的 OcaNetworkAddress 的解读	5.7.2
Ocp1NetworkHost	本部分对更为通用的 OcaNetworkHost 的解读	5.7.3
Ocp1NetworkSystemInterfaceID	本部分对更为通用的 OcaNetworkSystemInterfaceID 的解读	5.7.4

附 录 B  
(资料性附录)  
协议数据单元 (PDU) 的 UML 描述

本附录的内容是外部XMI 2.1文档。它可通过以下链接下载:

[www.aes.org/standards/models/aes70-3-annexb-151112-tcpip-protocol-1.xmi](http://www.aes.org/standards/models/aes70-3-annexb-151112-tcpip-protocol-1.xmi)

参 考 文 献

- [1] GB/T 28167—2011 信息技术 XML 元数据交换 (XMI)
-

中 华 人 民 共 和 国  
广 播 电 视 行 业 标 准  
**网络音频应用的开放式控制架构**  
**第 3 部分：用于 TCP/IP 网络的协议**  
GY/T 322.3—2019

\*

国家广播电视总局广播电视规划院出版发行

责任编辑：王佳梅

查询网址：[www.abp2003.cn](http://www.abp2003.cn)

北京复兴门外大街二号

联系电话：(010) 86093424 86092923

邮政编码：100866

**版权专有 不得翻印**