目 次

1 范围 ................................................................................ 1

2 引用文件 ............................................................................ 1

3 术语、定义和缩略语 .................................................................. 2

3.1 术语和定义 ........................................................................ 2

3.2 缩略语 ............................................................................ 3

4 信令方法要求 ........................................................................ 3

* 1. REGISTER 方法 3
  2. INVITE 方法 4
  3. ACK 方法 4
  4. BYE 方法 4
  5. CANCEL 方法 4
  6. OPTIONS 方法 4
  7. INFO 方法 4
  8. UPDATE 方法 4
  9. MESSAGE 方法 4
  10. SUBSCRIBE 方法 4
  11. PUBLISH 方法 5
  12. NOTIFY 方法 5

5 接入要求 ............................................................................ 5

5.1 SIP URI 编码规则 5

5.2 接入认证 .......................................................................... 5

5.3 漫游接入 .......................................................................... 5

6 传输交换要求 ........................................................................ 5

6.1 信令传输 .......................................................................... 5

6.2 媒体传输 .......................................................................... 5

6.3 媒体协商 .......................................................................... 6

6.4 端口复用 .......................................................................... 6

6.5 信令路由 .......................................................................... 6

6.6 媒体路由 .......................................................................... 6

7 控制要求 ............................................................................ 6

7.1 注册 .............................................................................. 6

7.2 点播和呼叫 ........................................................................ 6

7.3 设备控制 .......................................................................... 6

7.4 媒体存储、检索与调阅 .............................................................. 7

7.5 业务控制 .......................................................................... 7

7.6 信息同步 .......................................................................... 7

8 信道适应要求 ........................................................................ 7

8.1 会话保链 .......................................................................... 7

I

8.2 编码调整 .......................................................................... 7

8.3 I 帧申请 8

8.4 控制及信息同步命令分包传输及丢包重传 .............................................. 8

8.5 媒体丢包重传 ...................................................................... 8

8.6 定时器配置 ........................................................................ 8

9 信令流程要求 ........................................................................ 8

9.1 注册和注销 ........................................................................ 8

9.2 点播和呼叫 ....................................................................... 10

9.3 设备控制 ......................................................................... 28

9.4 媒体存储、检索与调阅 ............................................................. 29

9.5 业务控制 ......................................................................... 36

9.6 信息同步 ......................................................................... 43 附 录 A (规范性附录) 控制及信息同步命令集 ............................................ 47

A.1 命令格式 47

A.2 基础命令 48

* + 1. 命令集 48
    2. 丢包重传 48
    3. 操作反馈 50

A.3 设备控制命令 51

* + 1. 命令集 51
    2. 云镜控制 51

A.3.3 预置位操作 57

* + 1. 字幕设置 59
    2. I 帧间隔配置 66
    3. I 帧申请 67

A.4 媒体控制命令 68

* + 1. 命令集 68
    2. 媒体推送 69
    3. 建立呼叫 72
    4. 媒体存储 73
    5. 媒体检索 76
    6. 媒体传输 82
    7. 媒体下载 86

A.4.8 文件结尾通知 88

A.5 信息同步命令 89

A.5.1 命令集 89

A.5.2 设备状态通知 89

A.5.3 链路状态通知 91

A.5.4 状态信息同步 92

A.5.5 路由信息同步 96

A.5.6 授权信息通知 98

A.5.7 预置位请求 102

A.5.8 预置位响应 102

II

A.5.9 权限信息请求 104

A.5.10 权限信息响应 105

A.5.11 业务组信息请求 107

A.5.12 业务组信息响应 108

A.6 业务控制命令 111

A.6.1 命令集 111

A.6.2 静态组信息同步 112

A.6.3 分组业务基础控制 116

A.6.4 会议业务控制 145

A.6.5 ZH业务控制 148

附 录 B (规范性附录) 静态资源信息管理 157

B.1 协议要求 157

B.2 目录结构 157

B.3 数据描述 157

B.3.1 组织机构信息 158

B.3.2 用户信息 158

B.3.3 编解码设备信息 159

B.3.4 服务节点信息 160

B.3.5 服务设备信息 160

附 录 C (规范性附录) 扩展 SDP 定义 162

附 录 D (规范性附录) 方法响应命令集 165

D.1 常用响应命令集 165

D.2 自定义响应 166

* + 1. 488 Not Acceptable Here 166
    2. 451 XML Error 166
    3. 452 Not Accpted Label 166
    4. 453 Error Value 166
    5. 454 Unknown Label 166
    6. 455 Incomplete Label 166
    7. 456 Control Authorization Required 166
    8. 457 Bad Content Type 166

附 录 E (规范性附录) 媒体调阅控制命令 167

E.1 命令名称和说明 167

E.2 命令定义 167

E.2.1 媒体播放命令 167

E.2.2 暂停播放命令 167

E.2.3 快进／慢进命令 167

E.2.4 随机拖放命令 167

E.2.5 停止命令 167

E.2.6 应答命令 167

附 录 F (规范性附录) 优先级控制方法 168

附 录 G (建议性附录) 端口使用建议表 169

附 录 H (建议性附录) 媒体协商方法 170

H.1 H.264 媒体协商 170

H.2 H.265 媒体协商 170

H.3 AVS 媒体协商 171

H.3.1 SDP 参数 171

H.3.2 协商过程 174

1 范围

### xx视频ZH系统技术规范（1.0 版） 第 2 部分：信令控制技术要求

2 引用文件

#### 3 术语、定义和缩略语

##### 3.1 术语和定义

* + 1. 背靠背用户代理 back to back user agent

背靠背用户代理是逻辑功能实体，它作为用户代理服务器接收请求消息并处理该消息，同时，为了

判决该请求消息如何应答，它也作为用户代理客户端来发送请求消息。

* + 1. 代理服务器 proxy server

代理服务器是逻辑功能实体，通过它把来自用户代理客户端的请求转发到用户代理服务器，并把用

户代理服务器的响应消息转发回用户代理客户端。一个请求消息有可能通过若干个代理服务器来传送， 每一个代理服务器独立地确定路由，响应消息沿着请求消息相反的方向传递。

* + 1. 对话 dialog

对话是两个用户代理之间持续一段时间的对等SIP关系，它使用户代理之间的消息变得有序，同时

给出请求消息的正确路由。

* + 1. 第三方呼叫控制 third party call control

第三方控制者在另外两方或者更多方之间发起、建立会话以及释放会话的操作，负责会话方之间的

媒体协商。

* + 1. 呼叫 call

呼叫是指源代理与目标代理之间的一个通信行为，用于建立多媒体会话。

* + 1. 会话 session

会话是通信参与方及它们之间的媒体流的集合。

* + 1. 记录地址 address-of-record

记录地址（AOR）是一个SIP URI，指向一个具有定位服务的主机，这个主机可以把URI映射成用户

的具体位置。记录地址是一个用户的公开地址。

* + 1. 临时响应 provisional response

临时响应是标识服务器处理消息进展情况的响应，但是该响应并不终止对应的SIP事务。1xx响应是临时响应，其它响应都是最终响应，标志事务的结束。

* + 1. 全局路由表 global routing table

全局路由表是根据静态资源信息中服务节点关系生成的路由信息表。

* + 1. 事务 transaction

事务是用户代理之间由请求及其最终响应的所有消息构成。

* + 1. 请求 request

由客户端向服务器发送的SIP消息，用于执行特定功能。

* + 1. 响应 response

由服务器向客户端发送的SIP消息，用于标识请求处理的进展情况。

* + 1. 用户代理 user agent

用户代理是一个逻辑实体，能够发起SIP请求，也能够响应（接受、拒绝或者重定向）SIP请求。

* + 1. 源代理/目标代理 source agent/destination agent

源代理是主动发起SIP请求的一方，目标代理是最终响应SIP请求的一方。

* + 1. 最终响应 final response

最终响应是终止SIP事务的响应，和事务的临时响应相反。所有2xx、3xx、4xx、5xx和6xx响应都是最终响应。

##### 3.2 缩略语

下列缩略语适用于本标准。

AOR——record of address，记录地址； B2BUA——back to back user agent，背靠背用户代理

IP——internet protocol，因特网协议；

LDAP——lightweight directory access protocol，轻量级目录访问协议； MD5——message digest 5，信息摘要算法5；

RTP——real-time transport protocol，实时传输协议；

RTCP——real-time transport control protocol，实时传输控制协议； RTSP——real-time streaming protocol，实时流传输协议；

SIP——session initiation protocol，会话初始协议； SDP——session description protocol，会话描述协议；

SHA——secure hash algorithm，安全哈希算法； TCP——transmission control protocol，传输控制协议；

UDP——user datagram protocol，用户数据报协议； URI——uniform resource identifier，统一资源标识；

UUID——universally unique identifier，通用唯一识别码； XML——extensible markup language，可扩展标记语言。

#### 4 信令方法要求

* 1. REGISTER 方法

REGISTER方法应用于用户代理向信令控制服务单元进行注册和注销，创建、删除或更新AOR与网络

具体位置的绑定关系。注册认证应采用HTTP DIGEST认证机制，应支持MD5散列算法，宜支持SHA-1、SHA- 256、AKAv1-MD5、AKAv2-MD5等散列算法。采用MD5、SHA-1、SHA-256算法认证应符合IETF RFC 7616的

规范和要求，采用AKAv1-MD5算法认证应符合IETF RFC 3310的规范和要求，采用AKAv2-MD5算法认证应

符合IETF RFC 4169的规范和要求。另外，应具备扩展支持采用XX加密算法进行注册认证的能力。信 令控制服务单元负责对用户代理身份合法性进行验证，同时完成对REGISTER内容映射的读写及映射关系

的存储。

REGISTER方法宜采用Channel头域字段对信道特征进行描述，具体格式如下：

Channel:type=satellite/mobile/microwave/ultrashortwave/fixed;bandwidth=xxx 其中，satellite/mobile/microwave/ultrashortwave/fixed分别表示卫星、移动、微波、超短波、

固定网络等信道类型，可根据实际场景扩展Channel字段对于信道类型的定义；bandwidth表示信道带宽， 单位是kbps。

REGISTER方法和内容应符合IETF RFC 3261的规范和要求。

* 1. INVITE 方法

INVITE方法应用于发起会话和修改当前会话，信令控制服务单元应采用背靠背用户代理或第三方呼

叫控制等方式建立会话，不应采用代理服务器的方式建立会话。 INVITE消息应包含Subject头域，用于携带本次会话主叫方的SIP URI，例如：

示例：Subject:<sip:用户或设备编号@设备 IP 地址:端口> INVITE消息应支持优先级控制，具体优先级控制方法见附录F。 INVITE方法和内容应符合IETF RFC 3261的规范和要求。

* 1. ACK 方法

ACK方法应用于对INVITE请求产生的最终响应进行确认。

ACK方法和内容应符合IETF RFC 3261的规范和要求。

* 1. BYE 方法

BYE方法应用于结束已建立的会话。

BYE方法和内容应符合IETF RFC 3261的规范和要求。

* 1. CANCEL 方法

CANCEL方法应用于强制结束正在建立中的会话，同时会话被叫方应产生487响应作为本次会话的最

终响应。

CANCEL方法和内容应符合IETF RFC 3261的规范和要求。

* 1. OPTIONS 方法

OPTIONS方法应用于查询用户代理能力。

OPTIONS方法和内容应符合IETF RFC 3261的规范和要求。

* 1. INFO 方法

INFO方法应用于在会话过程中沿信令路径传递控制信息，INFO方法不应改变会话状态及参数。

INFO方法和内容应符合IETF RFC 6086的规范和要求。

* 1. UPDATE 方法

UPDATE方法应用于会话更新。当UPDATE方法用于会话更新时，不携带消息体。

UPDATE方法和内容应符合IETF RFC 3311的规范和要求。

* 1. MESSAGE 方法

MESSAGE方法应用于在服务节点内或服务节点间传递控制命令。

MESSAGE消息应包含Subject头域，用于携带命令发起方的SIP URI，例如：

示例：Subject:<sip:命令发起方编号@设备 IP 地址:端口> MESSAGE消息应支持优先级控制，具体优先级控制方法见附录F。 MESSAGE方法和内容应符合IETF RFC 3428的规范和要求。

* 1. SUBSCRIBE 方法

SUBSCRIBE方法应用于下级服务节点应用服务单元向上级服务节点应用服务单元订阅设备、用户和

路由等状态信息。

SUBSCRIBE方法和内容应符合IETF RFC 6665的规范和要求。

* 1. PUBLISH 方法

PUBLISH方法应用于下级服务节点应用服务单元向上级服务节点应用服务单元发布设备、用户和路

由等状态信息。

PUBLISH方法和内容应符合IETF RFC 3903的规范和要求。

* 1. NOTIFY 方法

NOTIFY方法应用于上级服务节点应用服务单元向已订阅下级服务节点应用服务单元发布设备、用户

和路由等状态信息。

NOTIFY方法和内容应符合IETF RFC 6665的规范和要求。

#### 5 接入要求

5.1 SIP URI 编码规则

SIP URI编码应采用如下格式：

sip:resourcenumber@domain:port 其中，resourcenumber表示设备或用户编号，由11位数字字符组成，具有全局唯一性，应对系统内

编/解码单元、应用服务单元、信令控制服务单元、信令网关服务单元等功能实体及通过信令网关服务 单元接入系统的编解码资源进行统一编号，宜对媒体处理服务单元、媒体存储服务单元等功能实体进行

统一编号；domain表示设备或用户的归属域，应采用设备或用户归属信令控制服务单元的IP地址，不应 采用域名；port表示信令传输端口，宜采用5060。

5.2 接入认证

接入系统的设备应支持向信令控制服务单元进行注册，信令控制服务单元负责对接入设备的注册请

求进行认证鉴权，设备注册的具体要求见7.1。 系统应支持用户通过应用控制单元登录系统，由应用服务单元负责对用户的登录请求进行认证鉴权。

5.3 漫游接入

系统应支持用户和设备的漫游接入。当用户或设备在非归属域服务节点接入时，该服务节点依据资

源目录服务单元存储的用户或设备信息对接入请求进行认证鉴权，并通过附录A.5.4节规定的状态信息 同步命令将用户或设备实际接入的服务节点ID同步至全网所有服务节点。漫游用户和设备使用接入服务

节点提供的服务，对于漫游用户和设备的控制信令的目的地址应指向其实际接入的服务节点。用户漫游 登录时，该用户实际登录的服务节点与用户归属的服务节点间应通过附录A.5.9节和附录A.5.10节规定

的权限信息请求和响应命令进行该用户权限信息的同步，通过附录A.5.11节和附录A.5.12节规定的分组 信息请求和响应命令进行该用户业务组信息的同步。

#### 6 传输交换要求

6.1 信令传输

信令的传输层协议应支持TCP和UDP，宜优先选用UDP协议。信令传输端口应可配置，TCP和UDP应配置相同的信令传输端口，宜采用5060。

6.2 媒体传输

媒体的传输层协议应支持UDP，会话层协议应支持RTP/RTCP。视音频码流的数据封装格式应符合《XX视频ZH系统技术规范（1.0版） 第3部分：视音频编解码技术要求》的要求。 RTP和RTCP协议应符合IETF RFC 3550、IETF RFC 4571的规定。

服务节点间媒体应通过媒体处理服务单元单播传输，不应使用组播。服务节点内部，媒体处理服务 单元与解码单元之间媒体传输应支持组播，编码单元与媒体处理服务单元或解码单元之间不宜使用组播。

服务节点应具备媒体流的复用能力，在会话建立过程中，若呼叫路由路径上的某一服务节点已经接 收了被叫编码单元的媒体流，则该服务节点应直接响应此呼叫，并通过媒体处理服务单元进行媒体流的

复用，而无需向路由路径的下一跳转发呼叫请求。

6.3 媒体协商

应支持会话发起方作为请求方的媒体协商过程，宜支持会话发起方作为应答方的媒体协商过程。媒

体协商过程中会话属性应采用扩展SDP进行描述，具体见附录C 单播协商时，请求方和响应方通过媒体描述的方向属性在呼叫双方之间建立发送或接收的媒体流。

组播协商时，请求方和响应方媒体描述的方向属性均为只发送（或只接收），代表呼叫双方（请求 方和响应方）只能进行媒体的发送（或接收）操作。

媒体协商的语法和流程应符合IETF RFC 3264的规定。

6.4 端口复用

媒体传输应支持端口复用，是否启用应可配置，并通过SDP的rtpport-mux参数协商，宜默认开启，

具体协商方式见附录C。在端口复用模式下，媒体码流应采用H.460.19协议定义的RTP端口复用机制进行 传输，流复用标识应通过SDP的muxid参数进行协商，由媒体接收方生成，同一媒体接收方不同会话的流

复用标识不应重复，同一会话相同方向的视频与音频码流使用同一流复用标识，媒体接收方根据RTP净

载类型进行解码。在端口复用模式下，RTP数据包的接收与发送端口应一致，宜采用19000、19002、19004 或19006，对应RTCP数据包的接收与发送端口宜采用19001、19003、19005或19007。

6.5 信令路由

服务节点内和服务节点间的信令交互应通过信令控制服务单元进行路由与转发。具备上下级关系或

关联关系的服务节点所属的信令控制服务单元在相互注册后应能够进行直接的信令交互。若信令控制服 务单元未互相注册或服务节点间不具备上下级关系或关联关系，则服务节点所属的信令控制服务单元间

不能直接进行SIP信令交互，需通过其它服务节点的信令控制服务单元转发。

SIP信令在跨服务节点路由时，若指定路由路径，应在SIP信令的Route头域携带所经过服务节点信 令控制服务单元的地址信息，各服务节点应严格按照所指定的路由路径传递信令；若不指定路由路径，

各服务节点应根据全局路由表和动态路由信息自行选择下一跳路由，全局路由表应根据资源目录服务单 元中节点信息表生成，动态路由信息应采用A.5.4节定义的路由信息同步命令在全网同步。

6.6 媒体路由

服务节点内和服务节点间的媒体交换应通过媒体处理服务单元进行转发。

会话内媒体路由所经服务节点应与信令路由所经服务节点保持一致。

#### 7 控制要求

7.1 注册

编/解码单元、应用服务单元、信令网关服务单元在接入系统前应向所属服务节点的信令控制单元进行注册；媒体处理服务单元、媒体存储服务单元在接入系统前宜向所属服务节点的信令控制单元进行 注册；具有上下级关系或关联关系的服务节点在建立连接前其所属的信令控制服务单元之间应进行相互注册；信令网关服务单元应代理由其接入的编解码资源向信令控制服务单元进行注册。 编/解码单元应具备双节点注册功能，当编/解码单元的归属服务节点信令控制服务单元无法提供服务时，应能自动切换至备用服务节点的信令控制服务单元进行注册，编/解码单元向非归属服务节点注 册应符合5.3节漫游接入的规定。

注册的信令交互流程见9.1节。

7.2 点播和呼叫

应用控制单元应支持对指定设备或用户进行视音频的实时点播，建立单向视音频通信。 应用控制单元应支持对指定用户进行视音频的实时呼叫，建立双向视音频通信。 点播和呼叫的信令交互流程见9.2节。

7.3 设备控制

系统应支持对指定编码单元进行云镜控制、预置位控制和I帧间隔配置，对指定解码单元进行字幕

设置。

应支持跨服务节点的设备控制操作。 设备控制的信令交互流程见9.3节。

7.4 媒体存储、检索与调阅

应支持对实时视音频数据进行存储。 应支持根据时间段或名称对历史视音频文件进行检索。

应支持对历史视音频数据进行远程调阅，调阅过程应支持正常播放、快速播放、慢速播放、画面暂停、随机拖放等媒体回放控制。

媒体存储、检索与调阅的信令交互流程见9.4节。

7.5 业务控制

应支持会议组管理、会议呼叫、结束会议、添加/强退成员、指定/取消发言、申请发言、会场转发、会议讨论/取消会议讨论、暂停会议/取消暂停会议等会议业务控制操作。 应支持ZH组管理、ZH呼叫、结束ZH、添加/强退成员、ZH转发/取消ZH转发、暂停ZH/取消暂停ZH等ZH业务控制操作，宜支持专向ZH/取消专向ZH、协同ZH/取消协同ZH、越级ZH/取消越级ZH、授权ZH/取消授权ZH、接替ZH/取消接替ZH等ZH业务控制操作。

业务控制的信令交互流程见9.5节。

7.6 信息同步

应支持组织机构、用户、编解码设备、服务节点和服务设备等静态信息的管理，并能够与资源目录

服务单元进行同步，静态信息同步采用LDAP协议v3版本，组织机构、用户、编解码设备、服务节点和服 务设备等静态信息在资源目录服务单元上的数据组织方式见附录B。

应支持设备、用户和路由等状态信息的跨服务节点同步；应支持漫游信息的跨服务节点同步。 状态信息同步和漫游信息同步的信令交互流程见9.6节。

#### 8 信道适应要求

8.1 会话保链

应支持在会话过程中采用UPDATE方法进行会话的更新。当在一个会话更新有效时间内未收到会话更新消息或响应时，会话失效一次，若会话连续失效次数未达到设定值（应可配置，建议默认值为3次） 并在下一个会话更新有效时间内收到会话更新消息或响应，则会话应能够正常接续，同时会话连续失效次数清零，若会话连续失效测试达到设定值，则会话完全失效，此时应结束会话。

会话更新有效时间采用Session-Expires和Min-SE两个头域字段进行协商。

Session-Expires字段用于指定会话更新的有效时间，该字段只能用于INVITE消息头域和INVITE消 息的2xx响应头域中。Session-Expires字段无缺省值，当INVITE消息头域或INVITE消息的2xx响应头域中不包含Session-Expires字段时，则表示本次会话长期有效，无需进行会话更新。

Min-SE字段表示会话更新有效时间的最小值，该字段只能用于INVITE消息头域和422响应的头域中。 当Min-SE字段被用于INVITE请求中时，表示希望的会话更新有效时间的最小值。当Min-SE字段被用于422响应中时，表示服务器原意接受的会话更新有效时间的最小值。Min-SE字段取值不能小于90秒，缺省值 为90秒。

会话更新有效时间的协商方法和要求应符合IETF RFC 4028的规定。

8.2 编码调整

会话参与方应支持在会话进行过程中采用INVITE方法，即Re-INVITE过程发起对编码速率、视频分辨率和视频帧率等参数的调整。会话建立时所协商的编码能力是此次会话的最高能力基线，编码调整不 能高于此能力。若调整请求被拒绝，则会话应保持不变。

会话参与方应支持在会话进行过程中采用Re-INVITE过程调整媒体传输方向、开关媒体传输通道，通过调整会话描述的媒体传输方向属性（a=sendrecv\sendonly\recvonly\inactive）实现。当网络条件变差时，编码调整应优先保证音频通信。对于关闭下行视频传输的Re-INVITE请求，解码单元在收到 该请求后应作出相应提示，并在接收到开启下行视频传输的Re-INVITE请求之前持续显示。

8.3 I 帧申请

应支持采用SIP INFO方法沿着信令传输路径申请更新 I帧，INFO请求的Content-Type字段采用application/media\_control+xml标识，消息体的内容采用XML格式，具体见附录A.3.6。

8.4 控制及信息同步命令分包传输及丢包重传

应支持对控制及信息同步命令进行分包传输和丢包重传。 控制及信息同步命令分包传输机制见附录A.1，丢包重传请求及响应方法见附录A.2.2。

8.5 媒体丢包重传

应支持通过RTCP扩展申请重传丢失的媒体数据包，具体方法应符合IETF RFC 4585的规定。媒体丢包重传机制是否启用应通过SDP的rtcp-fb参数进行协商，具体见附录C。 媒体数据包重传应采用与原数据包一致的媒体收发端口，重传的数据包内容应与原数据包完全相同，若媒体传输采用端口复用模式，端口复用标识也应相同。

8.6 定时器配置

应支持通过配置界面修改T1、T2、T4定时器参数。

9 信令流程要求

9.1 注册和注销

9.1.1 注册和注销基本要求

用户代理应采用4.1节规定的REGISTER请求和响应方法进行注册和注销。如果用户代理身份合法，信令控制服务单元应发送注册成功响应200 OK，并认为该用户代理的状态为在线。用户代理在注册过期 时间到来之前，应向信令控制服务单元进行注册刷新，应遵循IETF RFC 3261对注册刷新的规定，注册过期时间应可配置，建议取值范围为60~3600s，若用户代理注册过期，则信令控制服务单元认为该用户 代理的状态为离线。

如果注册请求中未包含Authorization头域字段，信令控制服务单元应发送401响应，并给出合适的 认证机制和参数。如果用户密码错误，则信令控制服务单元应发送403响应。如果用户不存在，则信令控制服务单元应发送404响应。如果注册失败，应延迟一定的时间间隔后重新注册，延迟的时间间隔应 可配置，不应小于60s。

用户代理每次启动新的注册流程时，REGISTER请求的CSeq头域序号值应从1开始累加。 信令网关服务单元应支持代理特定设备或用户向信令控制服务单元进行第三方注册和注销。第三方注册和注销时，采用To头域字段标识将要创建、删除或更新注册状态的AOR，采用From头域字段标识实 际负责注册或注销的第三方代理的AOR。

信令网关服务单元宜支持代理特定设备或用户的集合向信令控制服务单元进行第三方批量注册和 注销。第三方批量注册和注销时，To头域字段的AOR不生效，From头域字段标识实际负责注册或注销的第三方代理的AOR，待注册或注销的资源以列表形式在Contact头域字段表示，且Contact头字段应唯一。 列表中每个资源以资源编号为名称，以资源的URI为网络位置，用“R”标识资源序号，“R”取值为16位正整数。

示例：Contact:"101xxxx"<[sip:101xxxx@xx.xx.xx.xx](mailto:101xxxx@xx.xx.xx.xx)>;R=1;expires=3600,"102xxxx"<sip [:102xxxx@xx.xx.xx.xx](mailto:102xxxx@xx.xx.xx.xx)>;expires= 3600;R=2,"103xxxx"<sip:103xxxx@ xx.xx.xx.xx>;expires=3600;R=3

#### 9.1.2信令流程

* + - 1. 注册流程

用户代理注册流程见图1。

用户代理

信令控制 服务单元

**1:REGISTER**

**2:401 Unauthorized**

**3:REGISTER**

**4:200 OK**

注册信令流程描述如下：

图 1 注册流程

1:用户代理向信令控制服务单元发送REGISTER请求，REGISTER请求的Expires字段值为60~3600，以 秒为单位，请求中未包含Authorization字段。

2:信令控制服务单元向用户代理发送401响应（未授权），并在响应的消息头WWW\_Authenticate字 段给出合适的认证机制和参数。

3:用户 代理 应重 新向 信令 控制 服务 单元 发送 REGISTER 请求 ，REGISTER请求 的 Expires字段 值为

60~3600，以秒为单位，并在请求的Authorization字段包含认证信息。

4:信令控制服务单元对请求进行认证。如果用户代理身份合法，则向用户代理发送注册成功响应200

OK。如果用户代理身份不合法，应发送403响应。如果用户身份不存在，应发送404响应。

* + - 1. 注销流程

用户代理注销流程见图2。

用户代理

信令控制 服务单元

**1:REGISTER(Expires=0)**

**2:401 Unauthorized**

**3:REGISTER(Expires=0)**

**4:200 OK**

注销信令流程描述如下：

图 2 注销流程

1:用户代理向信令控制服务单元发送REGISTER请求，REGISTER请求的Expires字段值为0，请求中未 包含Authorization字段。

2:信令控制服务单元向用户代理发送401响应（未授权），并在响应的消息头WWW\_Authenticate字 段给出合适的认证机制和参数。

3:用户代理重新向信令控制服务单元发送REGISTER请求，REGISTER请求的Expires字段值为0，请求 的Authorization字段包含认证信息。

4:信令控制服务单元对请求进行认证，如果用户代理身份合法，则向用户代理发送注销成功响应200

OK。如果用户代理身份不合法，应发送403响应。如果用户身份不存在，应发送404响应。

#### 9.2 点播和呼叫

9.2.1 点播和呼叫基本要求

点播和呼叫应采用4.2节规定的INVITE方法建立会话，媒体协商应符合6.3节的规定，信令传输应符合6.1节的规定，媒体传输应符合6.2节的规定，信令路由应符合6.5节的规定，媒体路由应符合6.6节的 规定。点播和呼叫可由应用控制单元绑定的编/解码单元发起，也可由应用服务单元代理应用控制单元发起。

* + 1. 信令流程

##### 9.2.2.1 解码单元发起的点播流程

解码单元发起的点播流程见图3。

应用控制单元 解码单元 应用服务单元 信令控制服务单元 媒体处理服务单元 编码单元

用户 发起点播业务 同步业务状态、权限信息

开始点播

**1:INVITE(with SDP)**

**6:180 Ringing**

**10:DEVICESTATUS**

**13:200 OK (with SDP)**

**14:ACK**

**16:DEVICESTATUS**

媒体数据

**2:INVITE**

**3:200 OK (with SDP)**

**8:ACK(with SDP)**

**11:INVITE(with SDP)**

**12:200 OK (with SDP)**

**15:ACK**

**17:SWITCHSET**

**18:OPTSTATUS**

**4:INVITE(with SDP)**

**5:180 Ringing**

**7:200 OK (with SDP)**

**9:ACK**

媒体数据

结束点播

**20:UPDATE**

**21:200 OK**

**24:BYE**

**25:200 OK**

**28:DEVICESTATUS**

**19:CHANSTATUS**

**26:BYE**

**27:200 OK**

**22:UPDATE**

**23:200 OK**

**29:BYE**

**30:200 OK**

**31:CHANSTATUS**

**34:DEVICESTATUS**

**32:BYE**

**33:200 OK**

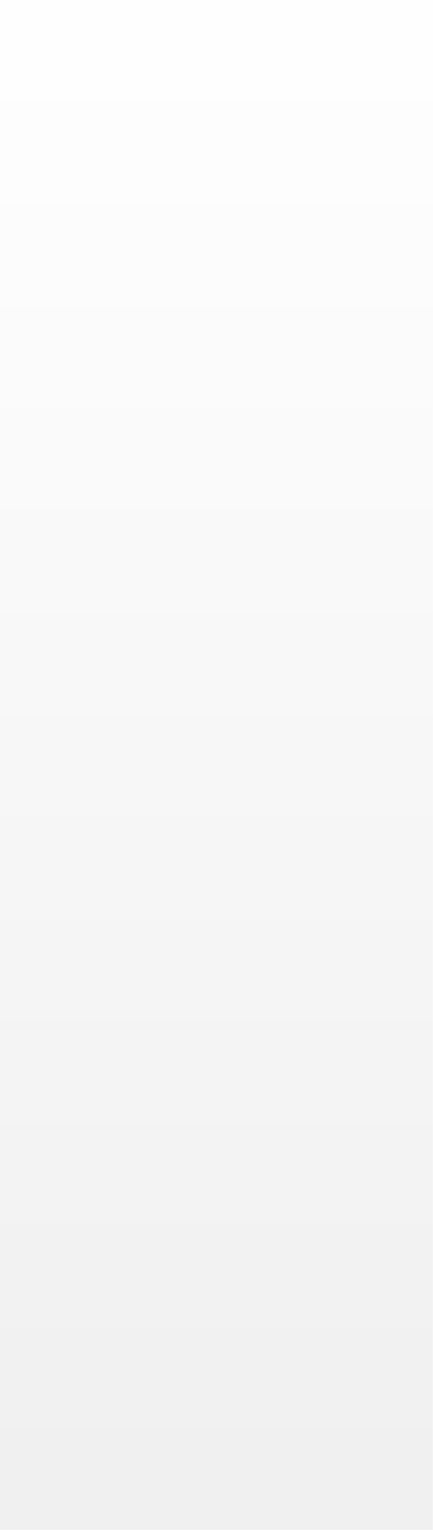


图 3 用户解码单元发起的点播流程

其中，1、11、12、13、14、15为信令控制服务单元接收到解码单元的呼叫请求后通过B2BUA方式建

立解码单元与媒体处理服务单元之间媒体连接的信令过程，2至9为通过信令控制服务单元建立编码单元

与媒体处理服务单元之间媒体连接的信令过程，24至27为解码单元断开与媒体处理服务单元之间媒体连 接的信令过程，28至32为信令控制服务单元断开媒体处理服务单元与编码单元之间媒体连接的信令过程。

信令流程描述如下： 1:解码单元向信令控制服务单元发送INVITE请求，请求中包含目标地址，并携带SDP消息体，具备

单向媒体接收属性（recvonly），消息头域应携带Subject字段表明会话的初始发起方，SDP消息体中s 字段为“Play”代表实时点播。

2:信令控制服务单元收到INVITE请求后，向媒体处理服务单元发送INVITE请求，建立媒体处理服务 单元和编码单元之间的媒体连接，此消息不携带SDP消息体。

3:媒体处理服务单元收到信令控制服务单元的INVITE请求后，回复200 OK响应，携带SDP消息体， 消息体描述了媒体处理服务单元接收媒体流的IP、端口、媒体格式等内容，具备双向媒体收发属性

（sendrecv），媒体属性应包含rtpid属性。

4:信令控制服务单元收到媒体处理服务单元返回的200 OK响应后，向编码单元发送INVITE请求，请 求应携带Subject字段表明会话的初始发起方，请求应携带消息3中媒体处理服务单元回复的200 OK响应

消息体，其中，s字段应为“Play”代表实时点播，媒体属性可不包含rtpid属性。

5:编码单元收到信令控制服务单元的INVITE请求后，编码单元返回180 Ringing响应（可选）。

6:信令控制服务单元收到编码单元的180 Ringing响应后，向解码单元返回180 Ringing响应（可 选）。

7:编码单元收到信令控制服务单元的INVITE请求后，回复200 OK响应，携带SDP消息体，消息体描

述了编码单元发送媒体流的IP、端口、媒体格式等内容，具备单向媒体发送属性（sendonly）。 8:信令控制服务单元收到编码单元返回的200 OK响应后，向媒体处理服务单元发送ACK请求，请求

中携带消息7中编码单元回复的200 OK响应消息体，完成与媒体处理服务单元的INVITE会话建立过程。 9:信令控制服务单元收到编码单元返回的200 OK响应后，向编码单元发送ACK请求，请求中不携带

消息体，完成与编码单元的INVITE会话建立过程。

10:编码单元与媒体处理服务单元建立媒体连接后，信令控制服务单元向应用服务单元发送设备状

态通知（设备正会话中、有媒体连接）。

11:编码单元与媒体处理服务单元建立媒体连接后，信令控制服务单元向媒体处理服务单元发送

INVITE请求，请求中携带消息1中的消息体。

12:媒体处理服务单元收到INVITE请求后，回复200 OK响应，携带SDP消息体，消息体描述了媒体处 理服务单元接收媒体流的IP、端口、媒体格式等内容，具备单向媒体发送属性（sendonly），媒体属性

应包含rtpid属性。

13:信令控制服务单元收到媒体处理服务单元回复的200 OK响应后，将消息12转发给解码单元。

14:解码单元向信令控制服务单元发送ACK请求（不携带SDP消息体），完成与信令控制服务单元的 INVITE会话建立过程。

15:信令控制服务单元将消息14转发给媒体处理服务单元，完成与媒体处理服务单元的INVITE会话 建立过程。

16:解码单元与媒体处理服务单元建立媒体连接后，信令控制服务单元向应用服务单元发送解码设 备状态（设备正会话中、有媒体连接）。

17:信令控制服务单元向媒体处理服务单元发送交换命令（媒体转发），实现点播监看视频。

18:媒体处理服务单元收到媒体转发命令后，应对转发指令进行处理，根据处理结果，媒体处理服 务单元向信令控制服务单元发送操作响应。

19:点播建立成功，双方建立经媒体处理服务单元转发的单向媒体流，媒体处理服务单元向信令控

制服务单元发送媒体传输已建立的通道状态通知。

20:解码单元向信令控制服务单元发送UPDATE请求更新会话时长。

21:信令控制服务单元收到UPDATE请求后，应更新会话时长，并回复200 OK响应。

22:编码单元向信令控制服务单元发送UPDATE请求更新会话时长。

23:信令控制服务单元收到UPDATE请求后，应更新会话时长，并回复200 OK响应。

24:用户结束点播，解码单元向信令控制服务单元发送BYE消息，断开由消息1、13、14建立会话。

25:信令控制服务单元收到BYE消息后回复200 OK响应，会话断开。

26:信令控制服务单元向媒体处理服务单元发送BYE消息，断开由消息11、12、15建立的会话。

27:媒体处理服务单元收到BYE消息后回复200 OK响应，会话断开。

28:解码单元与媒体处理服务单元断开媒体连接后，信令控制服务单元向应用服务单元发送设备状 态通知（设备在线）。

29:信令控制服务单元向媒体处理服务单元发送BYE消息，断开由消息2、3、8建立的会话。

30:媒体处理服务单元收到BYE消息后回复200 OK响应，会话断开。

31:媒体处理服务单元向信令控制服务单元发送媒体传输终止的通道状态通知。

32:信令控制服务单元向编码单元发送BYE消息，断开由消息4、7、9建立的会话。

33:编码单元收到BYE消息后回复200 OK响应，会话断开。

34:编码单元与媒体处理服务单元断开媒体连接后，信令控制服务单元向应用服务单元发送设备状 态通知（设备在线）。

##### 9.2.2.2 应用服务单元发起的点播流程

应用服务单元发起的点播流程见图4。

应用控制单元 解码单元 应用服务单元 信令控制服务单元 媒体处理服务单元 编码单元

用户 发起点播业务 同步业务状态、权限信息

开始点播

**1:MEDIAPUSH**

**2:OPTSTATUS**

**10:DEVICESTATUS**

**3:INVITE**

**4:200 OK (with SDP)**

**8:ACK(with SDP)**

**5:INVITE(with SDP)**

**6:180 Ringing 7:200 OK (with SDP)**

**9:ACK**

媒体数据

**11:INVITE**

**12:180 Ringing 13:200 OK (with SDP)**

**16:ACK(with SDP)**

**18:DEVICESTATUS**

媒体数据

**14:INVITE(with SDP)**

**15:200 OK (with SDP)**

**17:ACK**

**19:SWITCHSET**

**20:OPTSTATUS**

**21:CHANSTATUS**

结束点播

**22:UPDATE**

**23:200 OK**

**26:MEDIAPUSH**

**27:OPTSTATUS**

**24:UPDATE**

**25:200 OK**

**28:BYE**

**29:200 OK**

**32:DEVICESTATUS**

**30:BYE**

**31:200 OK**

**33:BYE**

**34:200 OK**

**35:CHANSTATUS**

**38:DEVICESTATUS**

**36:BYE**

**37:200 OK**

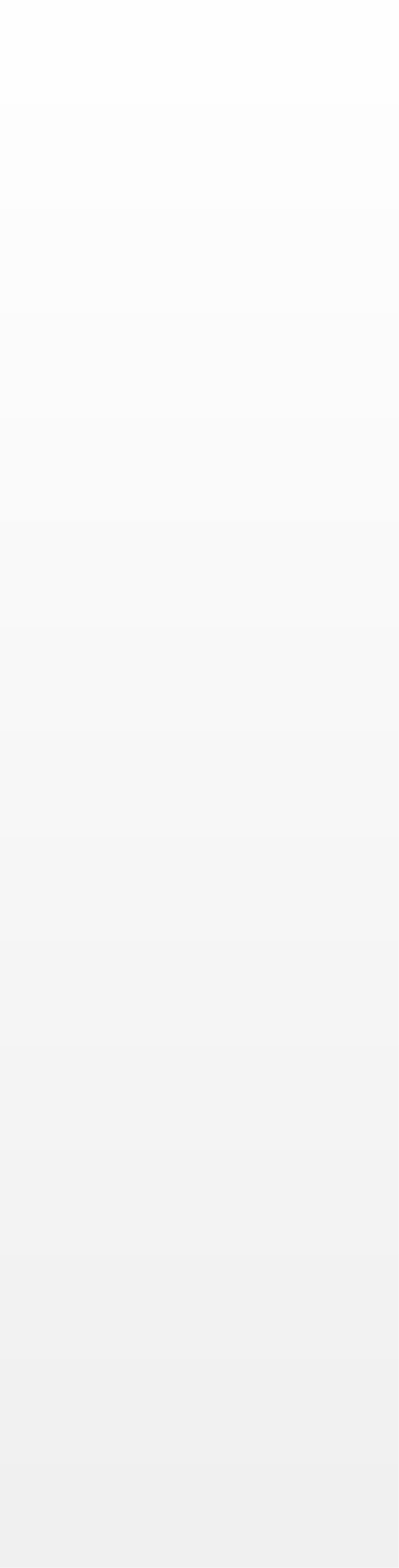


图 4 应用服务单元发起的点播流程

其中，3至9为信令控制服务单元建立媒体处理服务单元与编码单元之间媒体连接的信令过程，11至 17为信令控制服务单元建立媒体处理服务单元与解码单元之间媒体连接的信令过程，28至31为断开解码 单元与媒体处理服务单元之间媒体连接的信令过程，33至37为断开编码单元与媒体处理服务单元之间媒 体连接的信令过程。

信令流程描述如下： 1:用户开始点播操作后，应用服务单元向信令控制服务单元发送媒体推送开始命令（MEDIAPUSH）。

2:信令控制服务单元收到MEDIAPUSH命令后，回复操作反馈指令（OPTSTATUS），返回操作结果。

3:信令控制服务单元收到MEDIAPUSH后，向媒体处理服务单元发送INVITE请求，建立媒体处理服务 单元和编码单元之间的媒体连接，此消息不携带SDP消息体。

4:媒体处理服务单元收到信令控制服务单元的INVITE请求后，回复200 OK响应，携带SDP消息体，

消息体中描述了媒体处理服务单元接收媒体流的IP、端口、媒体格式等内容，具备双向媒体收发属性

（sendrecv），媒体属性应包含rtpid属性。

5:信令控制服务单元收到媒体处理服务单元返回的200 OK响应后，向编码单元发送INVITE请求，请 求应携带Subject字段表明会话的初始发起方，请求应携带消息3中媒体处理服务单元回复的200 OK响应

消息体，其中，s字段应为“Play”代表实时点播。 6:编码单元收到信令控制服务单元的INVITE请求后，回复180 Ringing响应（可选）。

7:编码单元回复200 OK响应，携带SDP消息体，消息体描述了编码单元发送媒体流的IP、端口、媒 体格式等内容，具备单向媒体发送属性（sendonly）。

8:信令控制服务单元收到编码单元返回的200 OK响应后，向媒体处理服务单元发送ACK请求，请求 中携带消息7中编码单元回复的200 OK响应消息体，完成与媒体处理服务单元的INVITE会话建立过程。

9:信令控制服务单元收到编码单元返回的200 OK响应后，向编码单元发送ACK请求，请求中不携带 消息体，完成与编码单元的INVITE会话建立过程。

10:编码单元与媒体处理服务单元建立媒体连接后，信令控制服务单元向应用服务单元发送编码设 备状态通知（设备正会话中、有媒体连接）。

11:编码单元与媒体处理服务单元建立媒体连接后，信令控制服务单元向解码单元发送INVITE请求， 此消息不携带SDP消息体。

12:解码单元收到信令控制服务单元的INVITE请求后，回复180 Ringing响应（可选）。

13:解码单元回复200 OK响应，携带SDP消息体，消息体中描述了解码单元接收媒体流的IP、端口、 媒体格式等内容，具备单向媒体接收属性（recvonly）。

14:信令控制服务单元收到解码单元的200OK响应后，向媒体处理服务单元发送INVITE请求，请求

中携带消息13中解码单元回复的200 OK响应消息体，其中，s字段应为“Play”代表实时点播。 15:媒体处理服务单元收到信令控制服务单元的INVITE请求后，回复200 OK响应，携带SDP消息体，

消息体中描述了媒体处理服务单元发送媒体流的 IP、端口、媒体格式等内容，具备单向发送属 性

（sendonly），媒体属性应包含rtpid属性。

16:信令控制服务单元收到媒体处理服务单元返回的200 OK响应后，向解码单元发送ACK请求，请求 中携带消息15中媒体处理服务单元回复的200 OK响应消息体，完成与解码单元的INVITE会话建立过程。

17:信令控制服务单元收到媒体处理服务单元返回的200 OK响应后，向媒体处理服务单元发送ACK请

求，请求中不携带消息体，完成与媒体处理服务单元的INVITE会话建立过程。

18: 解码单元与媒体处理服务单元建立媒体连接后，信令控制服务单元向应用服务单元发送解码设 备状态（设备正会话中、有媒体连接）。

19:信令控制服务单元向媒体处理服务单元发送交换命令（媒体转发），实现点播监看视频。

20:媒体处理服务单元收到媒体转发命令后，应对转发指令进行处理，根据处理结果，媒体处理服 务单元向信令控制服务单元发送操作响应。

21:点播建立成功，双方建立经媒体处理服务单元转发的单向媒体流，媒体处理服务单元向信令控 制服务单元发送媒体传输已建立的通道状态通知。

22:解码单元向信令控制服务单元发送UPDATE请求更新会话时长。

23:信令控制服务单元收到UPDATE请求后，应更新会话时长，并回复200 OK响应。

24:编码单元向信令控制服务单元发送UPDATE请求更新会话时长。

25:信令控制服务单元收到UPDATE请求后，应更新会话时长，并回复200 OK响应。

26:用户结束点播操作，应用服务单元向信令控制服务单元发送媒体推送结束命令（MEDIAPUSH）。

27:信令控制服务单元收到MEDIAPUSH命令后，回复操作反馈指令（OPTSTATUS），返回操作结果。

28:信令控制服务单元向解码单元发送BYE消息，断开由消息11、13、16建立的会话。

29:解码单元收到BYE消息后回复200 OK响应，会话断开。

30:信令控制服务单元向媒体处理服务单元发送BYE消息，断开由消息14、15、17建立的会话。

31:媒体处理服务单元收到BYE消息后回复200 OK响应，会话断开。

32:解码单元与媒体处理服务单元断开媒体连接后，信令控制服务单元向应用服务单元发送设备状 态通知（设备在线）。

33:信令控制服务单元向媒体处理服务单元发送BYE消息，断开由消息2、4、8建立的会话。

34:媒体处理服务单元收到BYE消息后回复200 OK响应，会话断开。

35:媒体处理服务单元向信令控制服务单元发送媒体传输终止的通道状态通知。

36:信令控制服务单元向编码单元发送BYE消息，断开由消息5、7、9建立的会话。

37:编码单元收到BYE消息后回复200 OK响应，会话断开。

38:编码单元与媒体处理服务单元断开媒体连接后，信令控制服务单元向应用服务单元发送设备状 态通知（设备在线）。

##### 9.2.2.3 编/解码单元发起的呼叫流程

编/解码单元发起的呼叫流程见图5。



编**/**解码单元 用户

**1:INVITE (with SDP)**

信令控制 服务单元

媒体处理

服务单元 编**/**解码单元

**6:180 Ringing**

**12:200 OK (with SDP)**

**2:INVITE**

**3:200 OK(with SDP)**

**8:ACK (with SDP)**

**10:INVITE (with SDP)**

**11:200 OK (with SDP)**

**4:INVITE (with SDP)**

**5:180 Ringing**

**7:200 OK (with SDP)**

**9:ACK**

媒体传输

**13:ACK**

**14:ACK**

媒体传输

**15:SWITCHSET**

媒体传输

**16:OPTSTATUS**

媒体传输

**18:UPDATE**

**19:200 OK**

用户 **22:BYE**

**23:200 OK**

**17:CHANSTATUS:ESTABLISH**

**24:BYE**

**25:200 OK**

**26:BYE**

**27:200 OK**

**28:CHANSTATUS:CLEANUP**

**20:UPDATE**

**21:200 OK**

**29:BYE**

**30:200 OK**

图 5 点对点呼叫流程

其中，1、10、11、12、13、14为信令控制服务单元接收到源编/解码单元的呼叫请求后通过B2BUA方

式建立源编/解码单元与媒体处理服务单元之间媒体连接的信令过程，2至9为通过信令控制服务单元建 立目标编/解码单元与媒体处理服务单元之间媒体连接的信令过程，22至25为源编/解码单元断开与媒体

处理服务单元之间媒体连接的信令过程，26至30为信令控制服务单元断开媒体处理服务单元与目标编/ 解码单元之间媒体连接的信令过程。

信令流程描述如下： 1:源编/解码单元向信令控制服务单元发送INVITE请求，请求中包含目标地址，并携带SDP消息体，

具备双向媒体传输属性（sendrecv），消息头域应携带Subject字段表明会话的初始发起方，SDP消息体 中s字段为“Call”代表实时呼叫。

2:信令控制服务单元收到INVITE请求后，向媒体处理服务单元发送INVITE请求，建立媒体处理服务 单元和源编/解码单元之间的媒体连接，此消息不携带SDP消息体。

3:媒体处理服务单元收到信令控制服务单元的INVITE请求后，回复200 OK响应，携带SDP消息体， 消息体描述了媒体处理服务单元接收媒体流的IP、端口、媒体格式等内容，具备双向媒体收发属性

（sendrecv），媒体属性应包含rtpid属性。

4:信令控制服务单元收到媒体处理服务单元返回的200 OK响应后，向目标编/解码单元发送INVITE 请求，请求应携带Subject字段表明会话的初始发起方，请求应携带消息3中媒体处理服务单元回复的200

OK响应消息体，其中，s字段应为“Call”代表实时呼叫，媒体属性可不包含rtpid属性。

5:目标编/解码单元收到信令控制服务单元的INVITE请求后，回复180 Ringing响应（可选）。

6:信令控制 服务单元 收到 目标编 /解码 单元的 180 Ringing 响应后， 向源编 /解码 单元返 回180 Ringing响应（可选）。

7:目标编/解码单元收到信令控制服务单元的INVITE请求后，回复200 OK响应，携带SDP消息体，消

息 体 描 述 了 目 标 编 /解 码 单 元 接 收 媒 体 流 的 IP、 端 口 、 媒 体 格 式 等 内 容 ， 具 备 双 向 媒 体 传 输 属 性

（sendrecv）。

8:信令控制服务单元收到目标编/解码单元返回的200 OK响应后，向媒体处理服务单元发送ACK请求， 请求中携带消息7中目标编/解码单元回复的200 OK响应消息体，完成与媒体处理服务单元的INVITE会话

建立过程。

9:信令控制服务单元收到编码单元返回的200 OK响应后，向目标编/解码单元发送ACK请求，请求中 不携带消息体，完成与目标编/解码单元的INVITE会话建立过程。

10:目标编/解码单元与媒体处理服务单元建立媒体连接后，信令控制服务单元向媒体处理服务单元

发送INVITE请求，请求中携带消息1中的消息体。

11:媒体处理服务单元收到INVITE请求后，回复200 OK响应，携带SDP消息体，消息体描述了媒体处 理服务单元接收媒体流的IP、端口、媒体格式等内容，具备双向媒体传输属性（sendrecv），媒体属性

应包含rtpid属性。

12:信令控制服务单元收到媒体处理服务单元回复的200 OK响应后，将消息11转发给源编/解码单元。

13:源编/解码单元向信令控制服务单元发送ACK请求（不携带SDP消息体），完成与信令控制服务单 元的INVITE会话建立过程。

14:信令控制服务单元将消息13转发给媒体处理服务单元，完成与媒体处理服务单元的INVITE会话

建立过程。

15:信令控制服务单元向媒体处理服务单元发送交换命令（媒体转发），实现呼叫。

16:媒体处理服务单元收到媒体转发命令后，应对转发指令进行处理，根据处理结果，媒体处理服 务单元向信令控制服务单元发送操作响应。

17:呼叫建立成功，双方建立经媒体处理服务单元转发的双向媒体流，媒体处理服务单元向信令控 制服务单元发送媒体传输已建立的通道状态通知。

18:源编/解码单元向信令控制服务单元发送UPDATE请求更新会话时长。

19:信令控制服务单元收到UPDATE请求后，应更新会话时长，并回复200 OK响应。

20:目标编/解码单元向信令控制服务单元发送UPDATE请求更新会话时长。

21:信令控制服务单元收到UPDATE请求后，应更新会话时长，并回复200 OK响应。

22:用户结束呼叫，源编/解码单元向信令控制服务单元发送BYE消息，断开由消息1、12、13建立的 会话。

23:信令控制服务单元收到BYE消息后回复200 OK响应，会话断开。

24:信令控制服务单元向媒体处理服务单元发送BYE消息，断开由消息10、11、14建立的会话。

25:媒体处理服务单元收到BYE消息后回复200 OK响应，会话断开。

26:信令控制服务单元向媒体处理服务单元发送BYE消息，断开由消息2、3、8建立的会话。

27:媒体处理服务单元收到BYE消息后回复200 OK响应，会话断开。

28:媒体处理服务单元向信令控制服务单元发送媒体传输终止的通道状态通知。

29:信令控制服务单元向目标编/解码单元发送BYE消息，断开由消息4、7、9建立的会话。

30:编码单元收到BYE消息后回复200 OK响应，会话断开。

##### 9.2.2.4 应用服务单元发起的呼叫流程

应用服务单元发起的呼叫流程见图6。

应用服务单元 应用控制端

**1:MEDIACALL**

编**/**解码单元

信令控制 服务单元

媒体处理

服务单元 编**/**解码单元

**2:OPTSTATUS**

**3:INVITE**

**4:200 OK(with SDP)**

**5:INVITE (with SDP)**

**6:180 Ringing**

**10:DEVICESTATUS**

**18:DEVICESTATUS**

**11:INVITE**

**12:180 Ringing 13:200 OK (with SDP)**

**16:ACK (with SDP)**

**8:ACK (with SDP)**

**14:INVITE (with SDP)**

**15:200 OK (with SDP)**

**17:ACK**

**7:200 OK (with SDP)**

**9:ACK**

媒体传输

媒体传输

**19:SWITCHSET**

**20:OPTSTATUS**

应用控制端

**26:MEDIACALL**

**27:OPTSTATUS**

**32:DEVICESTATUS**

**22:UPDATE**

**23:200 OK**

**28:BYE**

**29:200 OK**

**21:CHANSTATUS:ESTABLISH**

**24:UPDATE**

**25:200 OK**

**30:BYE**

**31:200 OK**

**33:BYE**

**34:200 OK**

**35:CHANSTATUS:CLEANUP**

**38:DEVICESTATUS**

**36:BYE**

**37:200 OK**

图 6 应用控制单元发起呼叫流程



其中，3至9为信令控制服务单元建立媒体处理服务单元与目标编/解码设备之间媒体连接的信令过

程，11至17为信令控制服务单元建立媒体处理服务单元和源编/解码单元之间媒体连接的信令过程，28 至31为断开源编/解码单元与媒体处理服务单元之间媒体连接的信令过程，33至37为断开目标编/解码单

元与媒体处理服务单元之间媒体连接的信令过程。 信令流程描述如下：

1:用户开始呼叫操作后，应用服务单元向信令控制服务单元发送建立呼叫开始命令（MEDIACALL）。

2:信令控制服务单元收到MEDIACALL命令后，回复操作反馈指令（OPTSTATUS），返回操作结果。

3:信令控制服务单元收到MEDIACALL后，向媒体处理服务单元发送INVITE请求，建立媒体处理服务 单元和目标编/解码单元之间的媒体连接，此消息不携带SDP消息体。

4:媒体处理服务单元收到信令控制服务单元的INVITE请求后，回复200 OK响应，携带SDP消息体， 消息体中描述了媒体处理服务单元接收媒体流的IP、端口、媒体格式等内容，具备双向媒体收发属性

（sendrecv），媒体属性应包含rtpid属性。

5:信令控制服务单元收到媒体处理服务单元返回的200 OK响应后，向目标编/解码单元发送INVITE 请求，请求应携带Subject字段表明会话的初始发起方，及消息3中媒体处理服务单元回复的200 OK响应

消息体，其中，s字段应为“Call”代表呼叫。

6:编码单元收到信令控制服务单元的INVITE请求后，回复180 Ringing响应（可选）。

7:目标编/解码单元收到信息控制服务单元的INVITE请求后，回复200 OK响应，携带SDP消息体，消 息 体 描 述 了 目 标 编 /解 码 单 元 接 收 媒 体 流 的 IP、 端 口 、 媒 体 格 式 等 内 容 ， 具 备 双 向 媒 体 传 输 属 性

（sendrecv）。

8:信令控制服务单元收到目标编/解码单元返回的200 OK响应后，向媒体处理服务单元发送ACK请求， 请求中携带消息7中目标编/解码单元回复的200 OK响应消息体，完成与媒体处理服务单元的INVITE会话

建立过程。

9:信令控制服务单元收到目标编/解码单元返回的200 OK响应后，向目标编/解码单元发送ACK请求， 请求中不携带消息体，完成与目标编/解码单元的INVITE会话建立过程。

10:目标编/解码单元与媒体处理服务单元建立媒体连接后，信令控制服务单元向应用服务单元发送

目标编/解码单元的状态通知（设备正会话中、有媒体连接）（可选，若目标编/解码单元为编解码设备 应发送此状态通知）。

11:目标编/解码单元码单元与媒体处理服务单元建立媒体连接后，信令控制服务单元向源编/解码 单元发送INVITE请求，此消息不携带SDP消息体。

12:源编/解码单元收到信令控制服务单元的INVITE请求后，回复180 Ringing响应（可选）。

13:源编/解码单元回复200 OK响应，携带SDP消息体，消息体中描述了源编/解码单元接收媒体流的 IP、端口、媒体格式等内容，具备双向媒体传输属性（sendrecv）。

14:信令控制服务单元收到源编/解码单元的200 OK响应后，向媒体处理服务单元发送INVITE请求，

请求中携带消息13中解码单元回复的200 OK响应消息体，其中，s字段应为“Call”代表实时呼叫。 15:媒体处理服务单元收到信令控制服务单元的INVITE请求后，回复200 OK响应，携带SDP消息体，

消息体中描述了媒体处理服务单元接收媒体流的 IP、端口、媒体格式等内容，具备双向传输属 性

（sendrecv），媒体属性应包含rtpid属性。

16:信令控制服务单元收到媒体处理服务单元返回的200 OK响应后，向源编/解码单元发送ACK请求， 请求中携带消息16中媒体处理服务单元回复的200 OK响应消息体，完成与源编/解码单元的INVITE会话 建立过程。

17:信令控制服务单元收到媒体处理服务单元返回的200 OK响应后，向媒体处理服务单元发送ACK请 求，请求中不携带消息体，完成与媒体处理服务单元的INVITE会话建立过程。

18:源编/解码单元与媒体处理服务单元建立媒体连接后，信令控制服务单元向应用服务单元发送源 编/解码设备状态（设备正会话中、有媒体连接）(可选)。

19:信令控制服务单元向媒体处理服务单元发送交换命令（媒体转发），实现实时呼叫。

20:媒体处理服务单元收到媒体转发命令后，应对转发指令进行处理，根据处理结果，媒体处理服 务单元向信令控制服务单元发送操作响应。

21:呼叫建立成功，双方建立经媒体处理服务单元转发的双向媒体流，媒体处理服务单元向信令控

制服务单元发送媒体传输已建立的通道状态通知。

22:源编/解码单元向信令控制服务单元发送UPDATE请求更新会话时长。

23:信令控制服务单元收到UPDATE请求后，应更新会话时长，并回复200 OK响应。

24:目标编/解码单元向信令控制服务单元发送UPDATE请求更新会话时长。

25:信令控制服务单元收到UPDATE请求后，应更新会话时长，并回复200 OK响应。

26:用户结束呼叫操作，应用服务单元向信令控制服务单元发送建立呼叫结束命令（MEDIACALLL）。

27:信令控制服务单元收到MEDIACALL命令后，回复操作反馈指令（OPTSTATUS），返回操作结果。

28:信令控制服务单元向源编/解码单元发送BYE消息，断开由消息11、13、16建立的会话。

29:源编/解码单元收到BYE消息后回复200 OK响应，会话断开。

30:信令控制服务单元向媒体处理服务单元发送BYE消息，断开由消息14、15、17建立的会话。

31:媒体处理服务单元收到BYE消息后回复200 OK响应，会话断开。

32:源编/解码单元与媒体处理服务单元断开媒体连接后，信令控制服务单元向应用服务单元发送设 备状态通知（设备在线）（可选）。

33:信令控制服务单元向媒体处理服务单元发送BYE消息，断开由消息3、4、8建立的会话。

34:媒体处理服务单元收到BYE消息后回复200 OK响应，会话断开。

35:媒体处理服务单元向信令控制服务单元发送媒体传输终止的通道状态通知。

36:信令控制服务单元向编码单元发送BYE消息，断开由消息5、7、9建立的会话。

37:目标编/解码单元收到BYE消息后回复200 OK响应，会话断开。

38:目标编/解码单元与媒体处理服务单元断开媒体连接后，信令控制服务单元向应用服务单元发送 设备状态通知（设备在线）。

* + - 1. 点播转呼叫流程

##### 点播转呼叫流程见图7。

应用控制端 编/解码单元 应用服务单元 信令控制服务单元 媒体处理服务单元 编/解码单元

用户 发起监控业务

同步业务状态、权限信息 点播监控视频

**1:INVITE(with SDP)**

**6:180 Ringing**

**10:DEVICESTATUS**

**2:INVITE**

**3:200 OK (with SDP)**

**8:ACK(with SDP)**

**4:INVITE(with SDP)**

**5:180 Ringing 7:200 OK (with SDP)**

**9:ACK**

媒体数据

**13:200 OK (with SDP)**

**14:ACK**

**11:INVITE(with SDP)**

**12:200 OK (with SDP)**

**15:ACK**

**16:DEVICESTATUS**

媒体数据

**17:SWITCHSET**

**18:OPTSTATUS**

**19:CHANSTATUS**

用户 呼叫监控视频

**20:Re-INVITE(with SDP)**

**27:DEVICESTATUS**

**21:INVITE**

**22:200 OK (with SDP)**

**25:ACK(with SDP)**

**23:INVITE(with SDP)**

**24:200 OK (with SDP)**

**26:ACK**

媒体数据

**30:200 OK (with SDP)**

**31:ACK**

**28:INVITE(with SDP)**

**29:200 OK (with SDP)**

**32:ACK**

**33:DEVICESTATUS**

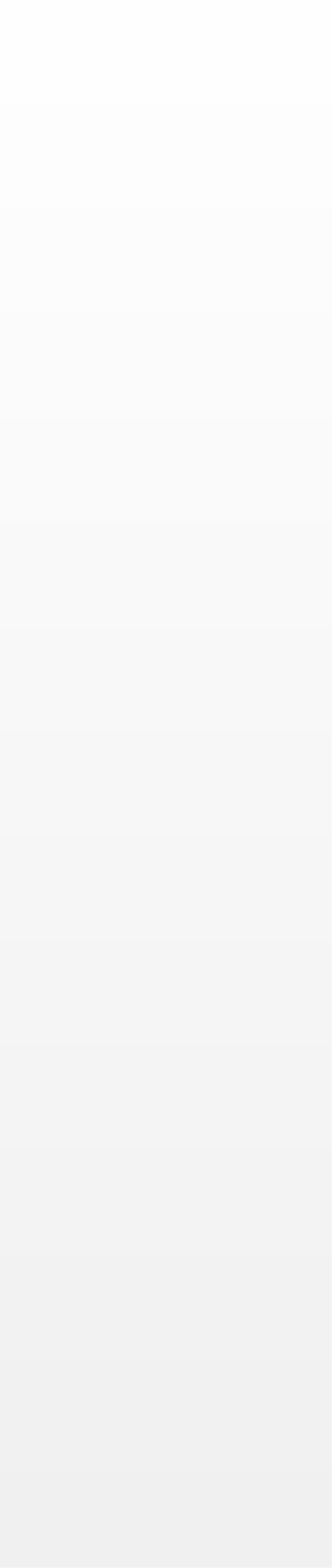
媒体数据

**34:SWITCHSET**

**35:OPTSTATUS**

**36:CHANSTATUS**

图 7 视频点播转视频呼叫流程图



应用控制单元可通过所绑定的编/解码单元向信令控制服务单元发送呼叫信令点播编/解码单元视 频，也可通过应用服务单元向信令控制服务单元发送控制命令点播编/解码单元视频。

在对编/解码单元的视频点播建立后，用户采用 Re-INVITE 发起对编/解码单元的呼叫，携带双向媒

体传输属性以及视音频参数，建立双向视音频传输。

其中，1、11、12、13、14、15为信令控制服务单元接收到编/解码单元的呼叫请求后通过B2BUA方式 建立解码单元与媒体处理服务单元之间的媒体流信令过程，2至9为通过信令控制服务单元建立目标编/ 解码单元与媒体处理服务单元之间的媒体流信令过程。

命令流程描述如下： 1:源编/解码单元向信令控制服务单元发送INVITE请求，请求中包含目标地址，并携带SDP消息体，

具备单向媒体接收属性（recvonly），消息头域应携带Subject字段表明会话的初始发起方，SDP消息体

中s字段为“Play”代表实时点播。 2:信令控制服务单元收到INVITE请求后，应建立媒体处理服务单元和源编/解码单元之间的媒体连

接。向媒体处理服务单元发送INVITE请求，此消息不携带SDP消息体。 3:媒体处理服务单元收到信令控制服务单元的INVITE请求后，回复200 OK响应，携带SDP消息体，

消息体描述了媒体处理服务单元接收媒体流的IP、端口、媒体格式等内容，具备双向媒体收发属性

（sendrecv），媒体属性应包含rtpid属性。

4:信令控制服务单元收到媒体处理服务单元返回的200 OK响应后，向目标编/解码单元发送INVITE 请求，请求应携带Subject字段表明会话的初始发起方，请求应携带消息3中媒体处理服务单元回复的200

OK响应消息体，其中，s字段应为“Play”代表实时点播，媒体属性可不包含rtpid属性。 5:目标编/解码单元收到信令控制服务单元的INVITE请求后，目标编/解码单元返回180 Ringing响

应（可选）。

6:信令控制 服务单元 收到 目标编 /解码 单元的 180 Ringing 响应后， 向源编 /解码 单元返 回180 Ringing响应（可选）。

7:目标编/解码单元收到信息控制服务单元的INVITE请求后，回复200 OK响应，携带SDP消息体，消

息 体 描 述 了 目 标 编 /解 码 单 元 发 送 媒 体 流 的 IP、 端 口 、 媒 体 格 式 等 内 容 ， 具 备 单 向 媒 体 发 送 属 性

（sendonly）。

8:信令控制服务单元收到目标编/解码单元返回的200 OK响应后，向媒体处理服务单元发送ACK请求， 请求中携带消息7中目标编/解码单元回复的200 OK响应消息体，完成与媒体处理服务单元的INVITE会话

建立过程。

9:信令控制服务单元收到目标编/解码单元返回的200 OK响应后，向目标编/解码单元发送ACK请求， 请求中不携带消息体，完成与目标编/解码单元的INVITE会话建立过程。

10:目标编/解码单元与媒体处理服务单元建立媒体连接后，信令控制服务单元向应用服务单元发送

目标编/解码单元的状态通知（设备正会话中、有媒体连接）（可选，若目标编/解码单元为编解码设备 应发送此状态通知）。

11:目标编/解码单元与媒体处理服务单元建立媒体连接后，信令控制服务单元向媒体处理服务单元 发送INVITE请求，请求中携带消息1中的消息体。

12:媒体处理服务单元收到INVITE请求后，回复200 OK响应，携带SDP消息体，消息体描述了媒体处

理服务单元接收媒体流的IP、端口、媒体格式等内容，具备单向媒体发送属性（sendonly），媒体属性 应包含rtpid属性。

13:信令控制服务单元收到媒体处理服务单元回复的200 OK响应后，将消息12转发给源编/解码单元。

14:源编/解码单元向信令控制服务单元发送ACK请求（不携带SDP消息体）完成与信令控制服务单元 的INVITE会话建立过程。

15:信令控制服务单元将消息14转发给媒体处理服务单元，完成与媒体处理服务单元的INVITE会话

建立过程。

16:源编/解码单元与媒体处理服务单元建立媒体连接后，信令控制服务单元向应用服务单元发送源 编/解码单元的状态通知（设备正会话中、有媒体连接）（可选，若源编/解码单元为编解码设备应发送 此状态通知）。

17:信令控制服务单元向媒体处理服务单元发送交换命令（媒体转发），实现点播。

18:媒体处理服务单元收到媒体转发命令后，应对转发指令进行处理，根据处理结果，媒体处理服 务单元向信令控制服务单元发送操作响应。

19:点播建立成功，双方建立经媒体处理服务单元转发的单向媒体流，媒体处理服务单元向信令控 制服务单元发送媒体传输已建立的通道状态通知。

20:源编/解码单元向信令控制服务单元发送ReINVITE请求，请求中包含目标地址，并携带SDP消息 体，具备双向媒体传输属性（sendrecv），消息头域应携带Subject字段表明会话的初始发起方，SDP消

息体中s字段为“Call”代表实时呼叫。 21:信令控制服务单元收到ReINVITE请求后，应建立媒体处理服务单元和源编/解码单元之间的媒体

连接。向媒体处理服务单元发送ReINVITE请求，此消息不携带SDP消息体。 22:媒体处理服务单元收到信令控制服务单元的ReINVITE请求后，回复200 OK响应，携带SDP消息体，

消息体描述了媒体处理服务单元接收媒体流的IP、端口、媒体格式等内容，具备双向媒体收发属性

（sendrecv），媒体属性应包含rtpid属性。 23:信令控制服务单元收到媒体处理服务单元返回的 200 OK响应后，向目标编/解码单元发送

ReINVITE请求，请求应携带Subject字段表明会话的初始发起方，请求应携带消息22中媒体处理服务单

元回复的200 OK响应消息体，其中，s字段应为“Call”代表实时呼叫，媒体属性可不包含rtpid属性。 24:目标编/解码单元收到信息控制服务单元的ReINVITE请求后，回复200 OK响应，携带SDP消息体，

消息体描述了目标编/解码单元发送媒体流的IP、端口、媒体格式等内容，具备双向媒体传输属性

（sendrecv）。

25:信令控制服务单元收到目标编/解码单元返回的200 OK响应后，向媒体处理服务单元发送ACK请 求，请求中携带消息24中目标编/解码单元回复的200 OK响应消息体，完成与媒体处理服务单元的

ReINVITE会话修改过程。

26:信令控制服务单元收到目标编/解码单元返回的200 OK响应后，向目标编/解码单元发送ACK请求， 请求中不携带消息体，完成与目标编/解码单元的ReINVITE会话修改过程。

27:目标编/解码单元与媒体处理服务单元建立媒体连接后，信令控制服务单元向应用服务单元发送

目标编/解码单元的状态通知（设备正会话中、有媒体连接）（可选，若目标编/解码单元为编解码设备 应发送此状态通知）。

28:目标编/解码单元与媒体处理服务单元建立媒体连接后，信令控制服务单元向媒体处理服务单元 发送ReINVITE请求，请求中携带消息20中的消息体。

29:媒体处理服务单元收到ReINVITE请求后，回复200 OK响应，携带SDP消息体，消息体描述了媒体

处理服务单元接收媒体流的IP、端口、媒体格式等内容，具备双向媒体传输属性（sendrecv），媒体属 性应包含rtpid属性。

30:信令控制服务单元收到媒体处理服务单元回复的200 OK响应后，将消息29转发给源编/解码单元。

31:源编/解码单元向信令控制服务单元发送ACK请求（不携带SDP消息体）完成与信令控制服务单元 的ReINVITE会话修改过程。

32:信令控制服务单元将消息31转发给媒体处理服务单元，完成与媒体处理服务单元的REINVITE会

话修改过程。

33:源编/解码单元与媒体处理服务单元建立媒体连接后，信令控制服务单元向应用服务单元发送源 编/解码单元的状态通知（设备正会话中、有媒体连接）（可选，若源编/解码单元为编解码设备应发送 此状态通知）。

34:信令控制服务单元向媒体处理服务单元发送交换命令（媒体转发），实现呼叫。

35:媒体处理服务单元收到媒体转发命令后，应对转发指令进行处理，根据处理结果，媒体处理服 务单元向信令控制服务单元发送操作响应。

36:呼叫建立成功，双方建立经媒体处理服务单元转发的双向媒体流，媒体处理服务单元向信令控 制服务单元发送媒体传输已建立的通道状态通知。

* + - 1. 呼叫级联流程

呼叫级联流程见图8。

本节点 外部节点



应用服务单元

编/解码单元

信令控制 服务单元

媒体处理 服务单元

信令控制 服务单元

媒体处理 服务单元

编/解码单元

应用服务单元

**28:DEVICESTATUS**

**1:INVITE (with SDP)**

**10:180 Ringing**

**25:200 OK (with SDP)**

**26:ACK**

媒体传输

**2:INVITE**

**3:200 OK(with SDP)**

**18:ACK (with SDP)**

**23:INVITE(with SDP)**

**24:200 OK(with SDP)**

**27:ACK**

**29:SWITCHSET**

**30:OPTSTATUS**

**4:INVITE (with SDP)**

**9:180 Ringing**

**17:200 OK(with SDP)**

**19:ACK**

媒体传输

**5:INVITE**

**6:200 OK(with SDP)**

**12:ACK (with SDP)**

**15:INVITE (with SDP)**

**16:200 OK (with SDP)**

**20:ACK**

**21:SWITCHSET**

**22:OPTSTATUS**

**7:INVITE (with SDP)**

**8:180 Ringing**

**11:200 OK(with SDP)**

**13:ACK**

媒体传输

**14:DEVICESTATUS**

**41:DEVICESTATUS**

**33:UPDATE 34:200 OK**

**39:BYE**

**40:200 OK**

**31:CHANSTATUS**

**42:BYE**

**43:200 OK**

**44:BYE**

**45:200 OK**

**46:CHANSTATUS**

**37:UPDATE**

**38:200 OK**

**47:BYE**

**48:200 OK**

**32:CHANSTATUS**

**49:BYE**

**50:200 OK**

**51:BYE**

**52:200 OK**

**53:CHANSTATUS**

**35:UPDATE**

**36:200 OK**

**54:BYE**

**55:200 OK**

**56:DEVICESTATUS**

图 8 呼叫级联流程 多级呼叫级联与两级呼叫级联的流程一致。 其中，1、23、24、25、26、27为本服务节点信令控制服务单元接收源编/解码单元的呼叫请求后通

过B2BUA方式建立源编/解码单元与本服务节点媒体处理服务单元之间的媒体流信令过程，2、3、4、15、 16、17、18、19、20为通过信令控制服务单元建立本服务节点媒体处理服务单元与外部服务节点媒体处理服务单元之间的媒体流信令过程，5、6、7、11、12、13为通过外部服务节点信令控制服务单元建立目 标编/解码单元与外部服务节点媒体处理服务单元之间的媒体流信令过程，39、40、42、43为源编/解码 单元断开与本服务节点媒体处理服务单元之间的媒体流信令过程，44、45、47、48、49、50为本服务节 点媒体处理服务单元断开与外部服务节点媒体处理服务单元之间的媒体流信令过程，51、52、54、55为 外部服务节点信令控制单元断开外部服务节点媒体处理单元与目标编/解码单元之间的媒体流信令过程。

命令流程描述如下： 1:源编/解码单元向本服务节点信令控制服务单元发送INVITE请求，请求中包含目标地址，并携带

SDP消息体，具备双向媒体传输属性（sendrecv），消息头域应携带Subject字段表明会话的初始发起方，

SDP消息体中s字段为“Call”代表实时呼叫。 2:本服务节点信令控制服务单元收到INVITE请求后，应建立本媒体处理服务单元和源编/解码单元

之间的媒体连接。向媒体处理服务单元发送INVITE请求，此消息不携带SDP消息体。 3:本服务节点媒体处理服务单元收到本服务节点信令控制服务单元的INVITE请求后，回复200 OK响

应，携带SDP消息体，消息体描述了媒体处理服务单元接收媒体流的IP、端口、媒体格式等内容，具备 双向媒体收发属性（sendrecv），媒体属性应包含rtpid属性，会话属性应包含端口复用标识。

4:本服务节点信令控制服务单元收到媒体处理服务单元返回的200 OK响应后，向外部服务节点信令 控制服务单元发送INVITE请求，请求应携带Subject字段表明会话的初始发起方，请求应携带消息3中本

服务节点媒体处理服务单元回复的200 OK响应消息体，其中，s字段应为“Call”代表实时呼叫，媒体 属性可不包含rtpid属性，会话属性应包含端口复用标识。

5:外部服务节点信令控制服务单元收到本服务节点信令控制服务单元的INVITE请求后，向外部服务 节点媒体处理服务单元发送INVITE请求，此消息不携带SDP消息体。

6:外部服务节点媒体处理服务单元收到信令控制服务单元的INVITE请求后，回复200 OK响应，携带 SDP消息体，消息体描述了外部媒体处理服务单元接收媒体流的IP、端口、媒体格式等内容，具备双向

媒体收发属性（sendrecv），媒体属性应包含rtpid属性。 7:外部服务节点信令控制服务单元收到外部服务节点媒体处理服务单元返回的200 OK响应后，向目

标编/解码单元发送INVITE请求，请求应携带Subject字段表明会话的初始发起方，请求应携带消息6中 外部服务节点媒体处理服务单元回复的200 OK响应消息体，其中，s字段应为“Call”代表实时呼叫，

媒体属性可不包含rtpid属性。 8:目标编/解码单元收到外部服务节点信令控制服务单元的INVITE请求后，目标编/解码单元返回

180 Ringing响应（可选）。

9:外部服务节点信令控制服务单元收到目标编/解码单元的180 Ringing响应后，向本服务节点信令 控制服务单元返回180 Ringing响应（可选）。

10:本服务节点信令控制服务单元收到外部服务节点信令控制服务单元的180 Ringing响应后，向源

编/解码单元返回180 Ringing响应（可选）。 11:目标编/解码单元收到外部服务节点信令控制服务单元的INVITE请求后，回复200 OK响应，携带

SDP消息体，消息体描述了目标编/解码单元接收媒体流的IP、端口、媒体格式等内容，具备双向媒体传

输属性（sendrecv）。

12:外部服务节点信令控制服务单元收到目标编/解码单元返回的200 OK响应后，向外部服务节点媒 体处理服务单元发送ACK请求，请求中携带消息12中编码单元回复的200 OK响应消息体，完成与外部服

务节点媒体处理服务单元的INVITE会话建立过程。

13:外部服务节点信令控制服务单元收到目标编/解码单元返回的200 OK响应后，向目标编/解码单 元发送ACK请求，请求中不携带消息体，完成与目标编/解码单元的INVITE会话建立过程。

14:目标编/解码单元与外部服务节点媒体处理服务单元建立媒体连接后，外部服务节点信令控制服 务单元向外部服务节点应用服务单元发送设备状态通知（设备正会话中、有媒体连接）。

15:目标编/解码单元与外部服务节点媒体处理服务单元建立媒体连接后，外部服务节点信令控制服 务单元向外部服务节点媒体处理服务单元发送INVITE请求，请求中携带消息4中的消息体（会话属性应

包含端口复用标识）。

16:外部服务节点媒体处理服务单元收到INVITE请求后，回复200 OK响应，携带SDP消息体，消息体 描述了外部服务节点媒体处理服务单元接收媒体流的IP、端口、媒体格式等内容，具备双向媒体传输属

性（sendrecv），媒体属性应包含rtpid属性，会话属性应包含端口复用标识。

17:外部服务节点信令控制服务单元收到外部服务节点媒体处理服务单元回复的200 OK响应后，将 消息16转发给本服务节点信令控制服务单元。

18:本服务节点信令控制服务单元收到外部服务节点信令控制服务单元返回的200 OK响应后，向本 服务节点媒体处理服务单元发送ACK请求，请求中携带消息17中外部服务节点媒体处理服务单元回复的

200 OK响应消息体，完成与本服务节点媒体处理服务单元的INVITE会话建立过程。

19:本服务节点信令控制服务单元收到外部服务节点信令控制服务单元返回的200 OK响应后，向外 部服务节点信令控制服务单元发送ACK请求（不携带SDP消息体），完成与外部服务节点信令控制服务单元的INVITE会话建立过程。

20:外部服务节点信令控制服务单元将消息19转发给外部服务节点媒体处理服务单元，完成与外部 服务节点媒体处理服务单元的INVITE会话建立过程。

21:外部服务节点信令控制服务单元向外部服务节点媒体处理服务单元发送交换命令（媒体转发）。

22:外部服务节点媒体处理服务单元收到外部服务节点媒体转发命令后，应对转发指令进行处理， 根据处理结果，外部服务节点媒体处理服务单元向外部服务节点信令控制服务单元发送操作响应。

23:本服务节点媒体处理服务单元与外部服务节点媒体处理服务单元建立媒体连接后，本服务节点

信令控制服务单元向本服务节点媒体处理服务单元发送INVITE请求，请求中携带消息1中的消息体。 24:本服务节点媒体处理服务单元收到INVITE请求后，回复200 OK响应，携带SDP消息体，消息体描

述了本服务节点媒体处理服务单元接收媒体流的IP、端口、媒体格式等内容，具备双向媒体传输属性

（sendrecv），媒体属性应包含rtpid属性。 25:本服务节点信令控制服务单元收到本服务节点媒体处理服务单元回复的200 OK响应后，将消息

24转发给源编/解码单元。

26:源编/解码单元向本服务节点信令控制服务单元发送ACK请求（不携带SDP消息体），完成与本服 务节点信令控制服务单元的INVITE会话建立过程。

27:本服务节点信令控制服务单元将消息26转发给本服务节点媒体处理服务单元，完成与本服务节 点媒体处理服务单元的INVITE会话建立过程。

28:源编/解码单元与本服务节点媒体处理服务单元建立媒体连接后，本服务节点信令控制服务单元

向本服务节点应用服务单元发送设备状态通知（设备正会话中、有媒体连接）。 29:本服务节点信令控制服务单元向本服务节点媒体处理服务单元发送交换命令（媒体转发）。

30:本服务节点媒体处理服务单元收到本服务节点媒体转发命令后，应对转发指令进行处理，根据 处理结果，本服务节点媒体处理服务单元向本服务节点信令控制服务单元发送操作响应。

31:呼叫建立成功，双方建立经媒体处理服务单元转发的双向媒体流，本服务节点媒体处理服务单 元向本服务节点信令控制服务单元发送媒体传输已建立的通道状态通知。

32:呼叫建立成功，双方建立经媒体处理服务单元转发的双向媒体流，外部服务节点媒体处理服务 单元向外部服务节点信令控制服务单元发送媒体传输已建立的通道状态通知。

33:源编/解码单元向本服务节点信令控制服务单元发送UPDATE请求更新会话时长。

34:本服务节点信令控制服务单元收到UPDATE请求后，应更新会话时长，并回复200 OK响应。

35:目标编/解码单元向外部服务节点信令控制服务单元发送UPDATE请求更新会话时长。

36:外部信令控制服务单元收到UPDATE请求后，应更新会话时长，并回复200 OK响应。

37:本服务节点信令控制单元向外部服务节点信令控制服务单元发送UPDATE请求更新会话时长。

38:外部信令控制服务单元收到UPDATE请求后，应更新会话时长，并回复200 OK响应。

39:源编/解码单元向信令控制服务单元发送BYE消息，断开消息1、25、26建立的同源编/解码单元 的INVITE会话。

40:本服务节点信令控制服务单元收到BYE消息后回复200 OK响应，会话断开。

41:本服务节点信令控制服务单元向本服务节点应用服务单元发送设备状态通知。

42:源编/解码单元媒体资源不再被使用，本服务节点信令控制服务单元向本服务节点媒体处理服务 单元发送BYE消息，断开23、24、27建立的同媒体处理服务单元的INVITE会话。

43:本服务节点媒体处理服务单元收到BYE消息后回复200 OK响应，会话断开。

44:本服务节点信令控制服务单元向本服务节点媒体处理服务单元发送BYE请求，断开消息2、3、18 建立的会话。

45:本服务节点媒体处理服务单元收到回复200 OK响应，会话断开。

46:本服务节点媒体处理服务单元向本服务节点信令控制服务单元发送媒体传输终止的状态通知。

47:本服务节点信令控制服务单元向外部服务节点信令控制服务单元发送BYE请求，断开消息4、9、

17、19建立的会话。

48:外部服务节点信令控制服务单元在收到BYE请求后，回复200 OK响应，会话断开。

49:外部服务节点信令控制服务单元向外部服务节点媒体处理服务单元发送BYE请求，断开消息15、

16、20建立的会话。

50:外部服务节点媒体处理服务单元在收到BYE请求后，回复200 OK响应，会话断开。

51:外部服务节点信令控制服务单元向外部服务节点媒体处理服务单元发送BYE请求，断开消息5、

6、12建立的会话。

52:外部服务节点媒体处理服务单元对BYE请求回复200 OK响应，会话断开。

53:外部服务节点媒体处理服务单元向外部服务节点信令控制服务单元发送媒体传输终止的状态通

知。

54:外部服务节点信令控制服务单元向目标编/解码单元发送BYE请求，断开消息7、8、11、13建立的会话。

55:目标编/解码单元收到BYE请求后回复200 OK响应，会话断开。

56:目标编/解码单元与外部服务节点媒体处理服务单元断开媒体连接后，外部服务节点信令控制服 务单元向外部服务节点应用服务单元发送设备状态通知（设备在线）。

9.3 设备控制

9.3.1 设备控制基本要求

设备控制命令包括云镜控制、预置位操作、字幕设置和I帧间隔配置等，命令类型为附录A.1定义的Control类型，采用Message方法承载，消息的Content-Type头域应符合附录A.1规定的格式。云镜控制 命令见附录A.3.2，预置位控制命令见附录A.3.3，I帧间隔配置命令见附录A.3.5，字幕设置命令附录见A.3.4。

#### 9.3.2 信令流程

##### 设备控制信令流程见图9。

应用服务单元

应用控制端 **1:Message(**设备控制命令**)**

信令控制

服务单元 目标设备

**2:Message(**设备控制命令**)**

**3:200 OK**

**4:200 OK**

**6:Message(**设备控制响应命令**) 7:200 OK**

**5:Message(**设备控制响应命令**)**

**8:200 OK**

命令流程描述如下：

图 9 有应答设备控制命令流程图

1:应用服务单元向信令控制服务单元发送设备控制命令，设备控制命令采用MESSAGE方法携带，应 包含Subject头域。

2:信令控制服务单元向目标设备（或目标用户）转发设备控制命令，设备控制命令采用MESSAGE方 法携带，应包含Subject头域。

3:目标设备（或目标用户）收到命令后返回200 OK。

4:信令控制服务单元返回200 OK。

5:目标设备向信令控制服务单元发送设备控制响应命令，设备控制响应命令采用MESSAGE方法携带， 应包含Subject头域。

6:信令控制服务单元向应用服务单元转发设备控制命令响应命令，设备控制响应命令采用MESSAGE 方法携带，应包含Subject头域。

7:应用服务单元收到命令后返回200 OK。

8:信令控制服务单元返回200 OK。

9.4 媒体存储、检索与调阅

9.4.1 媒体存储、检索和调阅基本要求

媒体存储业务开始时，由应用控制单元向应用服务单元发送媒体存储发起命令，开启媒体存储业务，再用MESSAGE方法在应用服务单元、信令控制服务单元和媒体存储服务单元之间发送媒体存储操控指令。 媒体检索应支持名称和时间为条件进行查询，用MESSAGE消息发送检索请求和返回查询结果。

媒体调阅分为应用控制单元通过编/解码单元主动获取历史媒体存储资源，以及应用服务单元从媒体存储服务单元向编/解码单元推送历史媒体存储资源两种方式。 媒体存储、检索和调阅相关请求命令采用附录A.4规定的XML格式。MESSAGE消息Content-Type头域应符合附录A.1规定的格式。

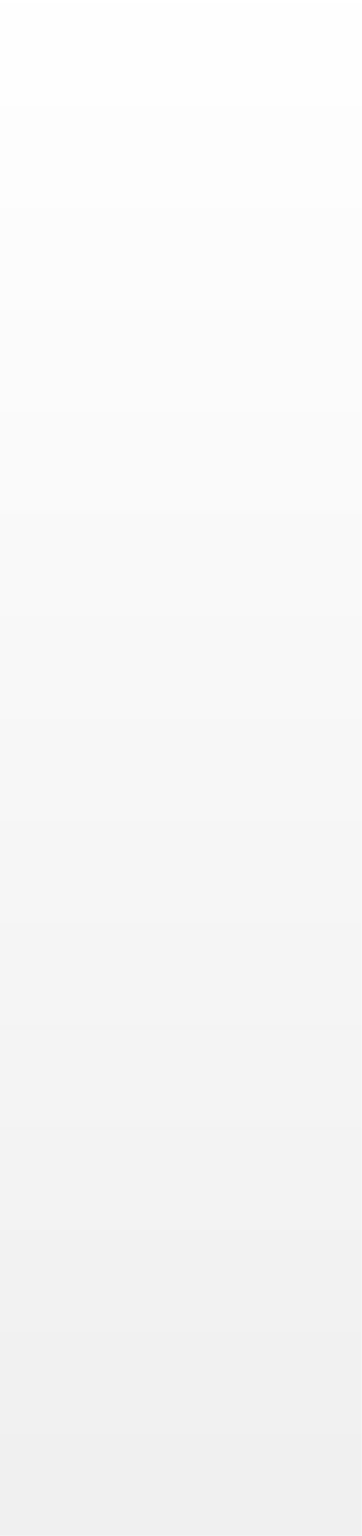
媒体调阅的媒体回放控制命令应采用INFO方法的消息体携带，实现视音频播放、停止、进/退和停 止等视音频播放控制过程。携带媒体调阅回放控制命令的INFO消息头Content-Type字段应为：Content- Type:application/play\_control，回放控制命令应使用附录E规定的格式。

* + 1. 信令流程

##### 媒体存储流程

* + - 1. 媒体存储流程

媒体存储流程见图10。



应用控制单元 编/解码单元 应用服务单元 信令控制服务单元 媒体处理服务单元 媒体存储服务单元

用户

发起存储业务

同步业务状态、权限信息

**1:STORESET**

**2:STORESET**

**4:OPTSTATUS**

**3:OPTSTATUS**

**5:INVITE**

**6:200 OK (with SDP)**

**7:INVITE(with SDP)**

**11:DEVICESTATUS**

**12:INVITE**

**13:180 Ringing**

**9:ACK(with SDP)**

**8:200 OK (with SDP)**

**10:ACK**

**14:200 OK (with SDP)**

**17:ACK (with SDP)**

**19:DEVICESTATUS**

**15:INVITE(with SDP)**

**16:200 OK (with SDP)**

**18:ACK**

待存储的媒体数据

**20:SWITCHSET**

**21:OPTSTATUS**

待存储的媒体数据

用户 终结存储业务

**24:STORENOTIFY**

**25:STORESET**

**26:BYE**

**27:200 OK**

**22:CHANSTATUS**

**23:STORENOTIFY**

**28:DEVICESTATUS**

**29:BYE**

**30:200 OK**

**31:BYE**

**32:200 OK**

**36:DEVICESTATUS**

**33:CHANSTATUS**

**34:BYE**

**35:200 OK**

图 10 媒体存储流程

当媒体存储业务开始时，应用控制单元向应用服务单元发送媒体存储发起命令；业务开始后，应用

服务单元与应用控制单元进行业务状态、权限等的同步。 命令流程描述如下：

1:应用服务单元向信令控制服务单元发送存储设置命令。

2:信令控制服务单元向媒体存储服务单元转发存储设置命令。

3:媒体存储服务单元发送操作响应。

4:信令控制服务单元转发操作响应。

5:信令控制服务单元向媒体处理服务单元发送INVITE请求（不带SDP消息体）。

6:媒体处理服务单元返回200 OK响应，携带SDP消息体，具备双向媒体收发属性。

7:信令控制服务单元向媒体存储服务单元发起INVITE请求，携带SDP消息体，具备双向媒体收发属 性。

8:媒体存储服务单元返回200 OK响应，携带SDP消息体，具备单向媒体接收属性。

9:信令控制服务单元向媒体处理服务单元发送ACK请求，携带SDP，具备单向媒体接收属性。

10:信令控制服务单元向媒体存储服务单元发送ACK请求（不携带SDP消息体）。

11:信令控制服务单元向应用服务单元发送设备状态通知。

12:信令控制服务单元向编/解码单元发起INVITE请求（不携带SDP消息体）。

13:编/解码单元返回180 Ringing响应（可选）。

14:编/解码单元返回200 OK响应，携带SDP消息体，具备双向媒体收发属性。

15:信令控制服务单元向媒体处理服务单元发送INVITE请求，携带SDP消息体，具备双向媒体收发属 性。

16:媒体处理服务单元返回200 OK响应，携带SDP消息体，具备单向媒体接收属性。

17:信令控制服务单元向编/解码单元发送ACK请求，携带SDP消息体，具备单向媒体接收属性。

18:信令控制服务单元向媒体处理服务单元发送ACK请求（不携带SDP消息体）。

19:信令控制服务单元向应用服务单元发送设备状态通知。

20:信令控制服务单元向媒体处理服务单元发送交换命令。

21:媒体处理服务单元向信令控制服务单元发送操作响应。

22:媒体处理服务单元向信令控制服务单元发送状态通知。

23:当磁盘即将存满时，媒体存储服务单元应向信令控制服务单元发送存储通知；

24:信令控制服务单元向应用服务单元转发状态通知。

25:终结存储业务时，应用服务单元向信令控制服务单元发送存储设置命令。

26:信令控制服务单元向编/解码单元发送BYE请求，断开消息12、13、14、17建立的会话。

27:信令控制服务单元收到BYE请求后回复200 OK响应，会话断开。

28:信令控制服务单元向应用服务单元发送设备状态通知。

29:信令控制服务单元向媒体处理服务单元发送BYE请求，断开消息15、16、18建立的会话。

30:媒体处理服务单元收到BYE请求后回复200 OK响应，会话断开。

31:信令控制服务单元向媒体处理服务单元发送BYE请求，断开消息5、6、9建立的会话。

32:媒体处理服务单元收到BYE请求后回复200 OK响应，会话断开。

33:媒体处理服务单元向信令控制服务单元发送状态通知。

34:信令控制服务单元向目标编/解码单元发送BYE请求，断开消息7、8、10建立的会话。

35:目标编/解码单元收到BYE请求后回复200 OK响应，会话结束。

36:信令控制服务单元向应用服务单元发送设备状态通知。

##### 媒体检索流程

* + - 1. 媒体检索流程

媒体检索流程见图11。

应用控制单元 应用服务单元 信令控制服务单元 媒体存储服务单元

用户

发起检索业务

同步业务状态、权限信息

媒体检索命令

**1:FILEQUERY**

**2:FILEQUERY**

**4:QUERYLIST**

**3:QUERYLIST**

图 11 媒体检索流程



当媒体检索业务开始时，发起业务的应用控制单元向应用服务单元发送媒体检索命令，命令流程描

述如下：

1:应用服务单元向信令控制服务单元发送媒体检索FILEQUERY消息，请求消息体包含视音频文件检 索条件。信令控制单元收到MESSAGE请求应立即返回200 OK，200 OK中不携带消息体，该过程包含在

FILEQUERY消息流程步骤1内。 2:信令控制服务单元收到媒体检索请求信息后，向媒体存储服务单元转发消息1。

3:媒体存储服务单元处理收到媒体检索请求后处理，返回QUERYLIST响应（MESSAGE方法），响应中 含文件目录，当一条MESSAGE消息无法传送完所有查询结果时，应分包采用多条消息传送。

4:信令控制服务单元向应用服务单元转发消息3。

##### 媒体调阅流程

* + - 1. 媒体调阅流程

媒体调阅包括应用控制单元通过编/解码单元主动获取媒体存储资源，以及应用服务单元从媒体存

储服务单元向编/解码单元推送媒体存储资源两种方式。 a) 主动获取媒体调阅流程见图12。

应用控制单元 编/解码单元 应用服务单元 信令控制服务单元 媒体处理服务单元 媒体存储服务单元

用户

发起调阅业务

同步业务状态、权限信息

开始调阅

**1:INVITE(with SDP)**

**6:180 Ringing**

**10:DEVICESTATUS**

**13:200 OK (with SDP)**

**14:ACK**

**16:DEVICESTATUS**

历史媒体数据

**2:INVITE**

**3:200 OK (with SDP)**

**8:ACK(with SDP)**

**11:INVITE(with SDP)**

**12:200 OK (with SDP)**

**15:ACK**

**17:SWITCHSET**

**18:OPTSTATUS**

**4:INVITE(with SDP)**

**5:180 Ringing 7:200 OK (with SDP)**

**9:ACK**

历史媒体数据

**19:CHANSTATUS**

**21:FILETOEND**

用户

**20:FILETOEND**

播放控制

**22:INFO 24:INFO**

**23:200OK**

**25:200OK**

用户 终结调阅业务

**26:BYE**

**27:200 OK**

**28:DEVICESTATUS**

**36:DEVICESTATUS**

**29:BYE**

**30:200 OK**

**31:BYE**

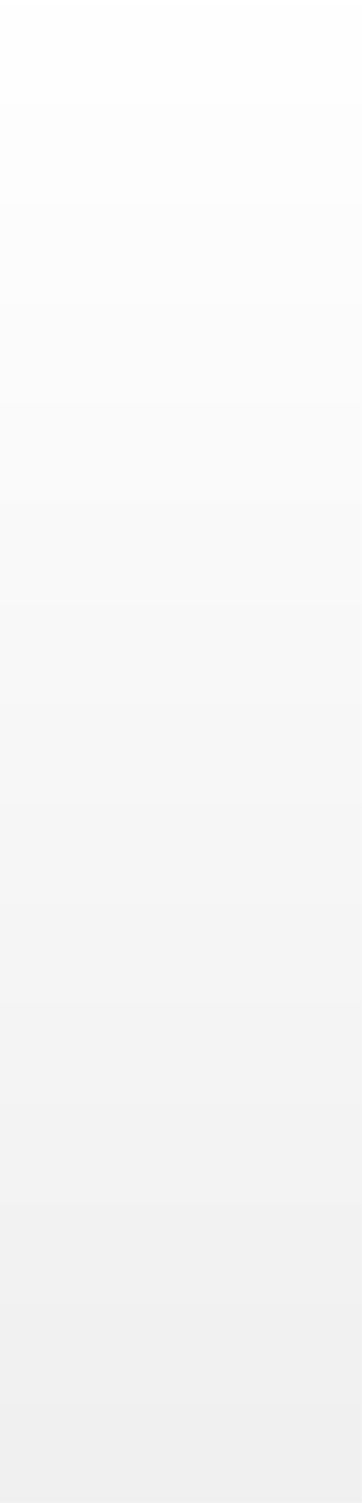
**32:200 OK**

**33:CHANSTATUS**

**34:BYE**

**35:200 OK**

图 12 主动获取媒体调阅流程



当媒体调阅业务开始时，发起业务的应用控制单元向应用服务单元发送媒体调阅发起命令；业务开

始后，应用服务单元与应用控制单元进行业务状态、权限等的同步；当开始调阅时，应用控制单元可通 过所绑定的编/解码单元向信令控制服务单元发送呼叫信令建立视频通信，或通过应用服务单元向信令

控制服务单元发送控制命令建立视频通信。 命令流程描述如下：

1:编/解码单元向信令控制服务单元发送INVITE请求，请求中包含目标地址，并携带SDP消息体，SDP 消息体中s字段为“Playback”代表历史回放，u字段代表播放文件的URL，t字段代表回放时间段。

2:信令控制服务单元向媒体处理服务单元发送INVITE请求（不带SDP消息体）。

3:媒体处理服务单元返回200 OK响应，携带SDP消息体，SDP消息体中描述媒体处理服务单元接收媒 体流的IP、端口、媒体格式等内容。

4:信令控制服务单元向媒体存储服务单元发起INVITE请求，请求中携带消息3中媒体处理服务单元 回复的200 OK响应消息体，并修改s字段为“Playback”代表历史回放，u字段代表播放文件的URL，t字

段代表回放时间段。

5:媒体存储服务单元返回180 Ringing响应（可选）。

6:信令控制服务单元返回180 Ringing响应（可选）。

7:媒体存储服务单元返回200 OK响应，携带SDP消息体，具备单向媒体发送属性。

8:信令控制服务单元向媒体处理服务单元发送ACK请求，携带SDP消息体，具备单向媒体发送属性。

9:信令控制服务单元向媒体存储服务单元发送ACK请求（不携带SDP消息体）。

10:信令控制服务单元向应用服务单元发送设备状态通知。

11:信令控制服务单元向媒体处理服务单元发起INVITE请求，携带SDP消息体，具备单向媒体接收属性。

12:媒体处理服务单元向信令控制服务单元返回200 OK响应，携带SDP消息体，具备单向媒体发送属 性。

13:信令控制服务单元向源编/解码单元发送200 OK响应，携带SDP消息体，具备单向媒体发送属性。

14:源编/解码单元向信令控制服务单元发送ACK请求（不携带SDP消息体）。

15:信令控制服务单元向媒体处理服务单元发送ACK请求（不携带SDP消息体）。

16:信令控制服务单元向应用服务单元发送设备状态通知。

17:信令控制服务单元向媒体处理服务单元发送交换命令。

18:媒体处理服务单元向信令控制服务单元发送操作响应。

19:呼叫建立，双方建立经媒体处理服务单元转发的单向媒体流，媒体处理服务单元向信令控制服

务单元发送状态通知。 20:当文件播放至结尾时，媒体存储服务单元向信令控制服务单元发送文件结尾通知。

21:信令控制服务单元向应用服务单元转发文件结束通知。

22:在回放过程中，媒体流接收的编/解码单元通过向信令控制服务单元发送会话内INFO消息进行回 放控制，包括视频的暂停、播放、快放、慢放、随机拖放播放等操作，INFO消息体见附录E。

23:信令控制服务单元向媒体流接收的编/解码单元回复200 OK响应。

24:信令控制服务单元收到消息22后转发给媒体存储服务单元。

25:媒体存储服务单元向信令控制服务单元回复200 OK响应。

26:源编/解码单元向信令控制服务单元发送BYE请求，断开消息1、6、13、14建立的会话。

27:信令控制服务单元收到BYE请求后回复200 OK响应，会话断开；

28:信令控制服务单元向应用服务单元发送设备状态通知。

29:源编/解码单元媒体资源不再被使用，信令控制服务单元向媒体处理服务单元发送BYE请求，断 开消息11、12、15建立的会话。

30:媒体处理服务单元收到BYE请求后回复200 OK响应，会话断开。

31:目标编/解码单元媒体资源不再被使用，信令控制服务单元向媒体处理服务单元发送BYE请求， 断开消息2、3、8建立的会话。

32:媒体处理服务单元收到BYE请求后回复200 OK响应，会话断开。

33:媒体处理服务单元向信令控制服务单元发送状态通知。

34:信令控制服务单元向目标编/解码单元发送BYE请求，断开消息4、5、7、9建立的会话。

35:目标编/解码单元收到BYE请求后回复200 OK响应，会话结束。

36:信令控制服务单元向应用服务单元发送设备状态通知。

b) 媒体资源推送流程见图13。



应用服务单元 应用控制端

**1:MEDIAPUSH**

编/解码单元 信令控制

服务单元

媒体处理 服务单元

媒体存储 服务单元

**2:OPTSTATUS**

**10:DEVICESTATUS**

**17:DEVICESTATUS**

**11:INVITE**

**12:200 OK (with SDP)**

**15:ACK (with SDP)**

**3:INVITE**

**4:200 OK(with SDP)**

**8:ACK (with SDP)**

**13:INVITE (with SDP)**

**14:200 OK (with SDP)**

**16:ACK**

**5:INVITE (with SDP)**

**6:180 Ringing 7:200 OK (with SDP)**

**9:ACK**

媒体传输

**18:SWITCHSET**

**19:OPTSTATUS**

媒体传输

**20:CHANSTATUS:ESTABLISH**

应用控制端

**21:MEDIAPUSH**

**22:OPTSTATUS**

**27:DEVICESTATUS**

**23:BYE**

**24:200 OK**

**25:BYE**

**26:200 OK**

**28:BYE**

**29:200 OK**

**30:CHANSTATUS:CLEANUP**

**33:DEVICESTATUS**

**31:BYE**

**32:200 OK**

命令流程描述如下：

图 13 资源推送媒体调阅流程

1:应用服务单元向信令控制服务单元发送媒体推送开始命令。

2:应用服务单元从信令控制服务单元获取操作结果。

3:信令控制服务单元向媒体处理服务单元发送INVITE请求（不带SDP消息体）。

4:媒体处理服务单元返回200 OK响应（携带SDP消息体）。

5:信令控制服务单元向媒体存储服务单元发起INVITE请求（携带SDP消息体）。

6:媒体存储服务单元返回180 Ringing响应（可选）。

7:媒体存储服务单元返回200 OK响应（携带SDP消息体）。

8:信令控制服务单元向媒体处理服务单元发送ACK请求（携带SDP消息体）。

9:信令控制服务单元向媒体存储服务单元发送ACK请求（不携带SDP消息体）。

10:信令控制服务单元向应用服务单元发送设备状态通知。

11:信令控制服务单元向编/解码单元发起INVITE请求（不携带SDP消息体）。

12:编/解码单元返回200 OK响应（携带SDP消息体）。

13:信令控制服务单元向媒体处理服务单元发送INVITE请求（带SDP消息体）。

14:媒体处理服务单元返回200 OK响应（携带SDP消息体）。

15:信令控制服务单元向编/解码单元发送ACK请求（携带SDP消息体）。

16:信令控制服务单元向媒体处理服务单元发送ACK请求（不携带SDP消息体）。

17:信令控制服务单元向应用服务单元发送设备状态通知。

18:信令控制服务单元向媒体处理服务单元发送交换命令。

19:媒体处理服务单元向信令控制服务单元发送操作响应。

20:呼叫建立，双方建立经媒体处理服务单元转发传输的媒体流，媒体处理服务单元向信令控制服 务单元发送状态通知。

21:应用服务单元向信令控制服务单元发送媒体推送结束命令。

22:应用服务单元从信令控制服务单元获取操作结果。

23:信令控制服务单元向编/解码单元发送BYE请求，断开消息11、12、15建立的会话。

24:编/解码单元收到BYE请求后回复200 OK响应，会话断开。

25:信令控制服务单元向媒体处理服务单元发送BYE请求，断开消息13、14、16建立的会话。

26:媒体处理服务单元收到BYE请求后回复200 OK响应，会话断开。

27:信令控制服务单元向应用服务单元发送设备状态通知。

28:信令控制服务单元向媒体处理服务单元发送BYE请求，断开消息3、4、8建立的会话。

29:媒体处理服务单元收到BYE请求后回复200 OK响应，会话断开。

30:媒体处理服务单元向信令控制服务单元发送状态通知。

31:信令控制服务单元向媒体存储服务单元发送BYE请求，断开消息5、6、7、9建立的会话。

32:媒体存储服务单元收到BYE请求后回复200 OK响应，会话结束。

33:信令控制服务单元向应用服务单元发送设备状态通知。

#### 9.5 业务控制

9.5.1 业务控制基本要求

业务控制包括ZH业务和会议业务。进行ZH或会议业务时，应能选定业务成员，并指定成员角色。

成员包括编码单元、编/解码单元或用户。 业务进行中，业务成员应能根据角色和具体业务，主动发起呼叫“拉”视音频建立会话，或通过应用服务单元发送信令控制服务单元发送媒体推送命令建立视音频会话，保证视音频通信逻辑的正确实现。

单服务节点内，业务成员至应用服务单元之间接口为应用控制单元与应用服务单元之间的接口，为 私有接口不在技术规范讨论范围内。多服务节点参与业务时，服务节点间应通过业务信息同步机制使用

附录A.6规定的业务控制命令完成ZH和会议业务的交互。 业务信息同步机制采用关联发送的同步方式，当服务节点的业务信息发生变化时，该服务节点应将信息发送至与该信息有关的所有其他服务节点。 对于ZH和会议业务中的开启、暂停、恢复、停止、成员强退、成员状态更新、成员加入（增量、

全量）、讨论模式、指定发言、取消发言、专向ZH、协同ZH、越级ZH、授权ZH、接替ZH等业 务变化时，服务节点应用服务单元需将业务信息推送至所有其他业务相关服务节点的应用服务单元，业务相关服务节点中任一服务节点业务信息发生变化时，也应将本服务节点更新的业务信息推送至所有其 他业务相关服务节点的应用服务单元。具体同步流程见[图15](#_bookmark0)。

对于授权信息变化和组信息管理时，服务节点应用服务单元需将授权变化信息推送至所有其他相关 服务节点的应用服务单元。具体同步流程见[图16](#_bookmark1)。

对于ZH和会议业务中的开启转发授权、停止转发授权等业务变化仅需将发生变化的业务信息推送 至与该业务相关服务节点的应用服务单元。具体同步流程见[图17](#_bookmark2)。

对于ZH和会议业务中的成员退出请求、成员退出响应、申请发言请求、申请发言响应等业务，发 生业务更新的服务节点应向具备业务管理能力的服务节点提出申请，申请得到响应后，具备业务管理能 力的服务节点再将更新的业务信息推送至所有其他业务相关服务节点的应用服务单元。具体同步流程见 [图18](#_bookmark3)。

业务信息同步采用MESSAGE方法携带规定的XML实现。MESSAGE方法应包含头域Subject用于携带操作 发起方信息。

业务信息同步机制适用的命令详细描述见附录A.6.2静态组信息同步、A.6.3分组业务基础控制、 A.6.4会议业务控制、A.6.5ZH业务控制和A.6.6授权信息通知。

9.5.2 信令流程

##### 9.5.2.1 业务会话建立流程

业务成员根据具体业务要求主动发起呼叫信令建立视音频通信流程见图14。

业务成员 应用服务单元 信令控制服务单元 媒体处理服务单元 业务成员

开始分组业务 同步业务状态、权限信息

开始分组业务通知

**1:INVITE(with SDP)**

**6:180 Ringing**

**2:INVITE**

**3:200 OK (with SDP)**

**4:INVITE(with SDP)**

**5:180 Ringing 7:200 OK (with SDP)**

**13:200 OK (with SDP)**

**10:DEVICESTATUS**

**8:ACK(with SDP)**

**11:INVITE(with SDP)**

**12:200 OK (with SDP)**

**9:ACK**

媒体数据

**14:ACK**

**15:ACK**

**16:DEVICESTATUS**

**17:SWITCHSET**

媒体数据

**18:OPTSTATUS**

**19:CHANSTATUS**

分组业务操作 结束分组业务

会话建立通知 分组业务操作通知 分组业务结束通知

**20:BYE**

**21:200 OK**

**24:DEVICESTATUS**

**22:BYE**

**23:200 OK**

**25:BYE**

**26:200 OK**

**30:DEVICESTATUS**

**27:CHANSTATUS**

会话结束通知

**28:BYE**

**29:200 OK**

图 14 分组业务流程



其中，1、11、12、13、14、15为信令控制服务单元接收到发起成员的呼叫请求后通过B2BUA方式建

立发起成员与媒体处理服务单元之间的媒体流信令过程，2至9为通过信令控制服务单元建立目标成员与 媒体处理服务单元之间的媒体流信令过程，24至27为发起成员断开与媒体处理服务单元之间的媒体流信

令过程，28至32为信令控制服务单元断开媒体处理服务单元与目标成员之间的媒体流信令过程。 命令流程描述如下：

1:发起成员向信令控制服务单元发送INVITE请求，请求中包含目标地址，并携带SDP消息体，具备 单向媒体接收属性（recvonly），消息头域应携带Subject字段表明会话的初始发起方，SDP消息体中s 字段为“Play”代表实时点播。

2:信令控制服务单元收到INVITE请求后，应建立媒体处理服务单元和目标成员之间的媒体连接。向 媒体处理服务单元发送INVITE请求，此消息不携带SDP消息体。

3:媒体处理服务单元收到信令控制服务单元的INVITE请求后，回复200 OK响应，携带SDP消息体，

消息体描述了媒体处理服务单元接收媒体流的IP、端口、媒体格式等内容，具备双向媒体收发属性

（sendrecv），媒体属性应包含rtpid属性。

4:信令控制服务单元收到媒体处理服务单元返回的200 OK响应后，向目标成员发送INVITE请求，请 求应携带Subject字段表明会话的初始发起方，请求应携带消息3中媒体处理服务单元回复的200 OK响应

消息体，其中，s字段应为“Play”代表实时点播。 5:目标成员收到信令控制服务单元的INVITE请求后，目标成员返回180 Ringing响应（可选）。

6:信令控制服务单元收到目标成员的180 Ringing响应后，向发起成员返回180 Ringing响应（可 选）。

7:目标成员收到信令控制服务单元的INVITE请求后，回复200 OK响应，携带SDP消息体，消息体描 述了编码单元发送媒体流的IP、端口、媒体格式等内容，具备单向媒体发送属性（sendonly）。

8:信令控制服务单元收到目标成员返回的200 OK响应后，向媒体处理服务单元发送ACK请求，请求 中携带消息7中目标成员回复的200 OK响应消息体，完成与媒体处理服务单元的INVITE会话建立过程。

9:信令控制服务单元收到目标成员返回的200 OK响应后，向目标成员发送ACK请求，请求中不携带 消息体，完成与目标成员的INVITE会话建立过程。

10:目标成员与媒体处理服务单元建立媒体连接后，信令控制服务单元向应用服务单元发送设备状 态通知（设备正会话中、有媒体连接）。

11:目标成员与媒体处理服务单元建立媒体连接后，信令控制服务单元向媒体处理服务单元发送 INVITE请求，请求中携带消息1中的消息体。

12:媒体处理服务单元收到INVITE请求后，回复200 OK响应，携带SDP消息体，消息体描述了媒体处 理服务单元接收媒体流的IP、端口、媒体格式等内容，具备单向媒体发送属性（sendonly），媒体属性应包含rtpid属性。

13:信令控制服务单元收到媒体处理服务单元回复的200 OK响应后，将消息12转发给发起成员。

14:发起成员向信令控制服务单元发送ACK请求（不携带SDP消息体），完成与信令控制服务单元的 INVITE会话建立过程。

15:信令控制服务单元将消息14转发给媒体处理服务单元，完成与媒体处理服务单元的INVITE会话 建立过程。

16:发起成员与媒体处理服务单元建立媒体连接后，信令控制服务单元向应用服务单元发送设备状

态（设备正会话中、有媒体连接）。 17:信令控制服务单元向媒体处理服务单元发送交换命令（媒体转发），实现视音频通信。

18:媒体处理服务单元收到媒体转发命令后，应对转发指令进行处理，根据处理结果，媒体处理服 务单元向信令控制服务单元发送操作响应。

19:媒体通信建立成功，双方建立经媒体处理服务单元转发的单向媒体流，媒体处理服务单元向信 令控制服务单元发送媒体传输已建立的通道状态通知。

20:发起成员结束分组业务，发起成员向信令控制服务单元发送BYE消息，断开消息1、13、14建立 的同解码单元的INVITE会话。

21:信令控制服务单元收到BYE消息后回复200 OK响应，会话断开。

22:信令控制服务单元向媒体处理服务单元发送BYE消息，断开11、12、15建立的同媒体处理服务单 元的INVITE会话。

23:媒体处理服务单元收到BYE消息后回复200 OK响应，会话断开。

24:发起成员与媒体处理服务单元断开媒体连接后，信令控制服务单元向应用服务单元发送设备状 态通知（设备在线）。

25:信令控制服务单元向媒体处理服务单元发送BYE消息，断开2、3、8建立的同媒体处理服务单元

的INVITE会话。

26:媒体处理服务单元收到BYE消息后回复200 OK响应，会话断开。

27:媒体处理服务单元向信令控制服务单元发送媒体传输终止的通道状态通知。

28:信令控制服务单元向目标成员发送BYE消息，断开4、7、9建立的同目标成员的INVITE会话。

29:目标成员收到BYE消息后回复200 OK响应，会话断开。

30:目标成员与媒体处理服务单元断开媒体连接后，信令控制服务单元向应用服务单元发送状态通 知（设备在线）。

9.5.2.2 业务信息同步流程

##### a) 业务信息同步消息流程1。

该同步流程的同步模式适用于ZH和会议业务的开始、暂停、恢复、停止、成员退出、成员状态更 新、成员加入（增量、全量）、讨论模式、指定发言、取消发言、专向ZH、协同ZH、越级ZH、授权ZH、接替ZH等控制操作，具体流程见图15。

业务发起节点

业务相关节点**1** 业务相关节**2**



业务相关节点**3**

应用服务单元

应用服务单元

应用服务单元

应用服务单元

**1:MESSAGE:GROUP/BIZCNF/BIZCMD**

**2:200 OK**

**3:MESSAGE:GROUP/BIZCNF/BIZCMD**

**4:200 OK**

**5:MESSAGE:GROUP/BIZCNF/BIZCMD**

**6:200 OK**

**7:MESSAGE:GROUP/BIZCNF/BIZCMD**

**8:200 OK**

**9:MESSAGE:GROUP/BIZCNF/BIZCMD**

**10:200 OK**

**11:MESSAGE:GROUP/BIZCNF/BIZCMD**

**12:200 OK**

图 15 业务信息同步消息流程 1 业务信息同步的级联同步流程1描述如下： 1:业务发起服务节点的应用服务单元向业务相关服务节点1的应用服务单元推送所有的业务信息。

2:业务相关服务节点1的应用服务单元向业务发起服务节点的应用服务单元的MESSAGE方法消息发 送200 OK响应。

3:业务发起服务节点的应用服务单元向业务相关服务节点2的应用服务单元推送所有的业务信息。

5:业务发起服务节点的应用服务单元向业务相关服务节点3的应用服务单元推送所有的业务信息。

6:业务相关服务节点3的应用服务单元向业务发起服务节点的应用服务单元的MESSAGE方法消息发 送200 OK响应。

7:业务相关服务节点2更新业务信息时，其应用服务单元向业务发起服务节点的应用服务单元同步

该服务节点更新的业务信息。 8:业务发起服务节点的应用服务单元向业务相关服务节点2的应用服务单元的MESSAGE方法消息发

送200 OK响应。 9:业务相关服务节点2更新业务信息时，其应用服务单元向业务相关服务节点1的应用服务单元同步

该服务节点更新的业务信息。 10:业务相关服务节点1的应用服务单元向业务相关服务节点2的应用服务单元的MESSAGE方法消息

发送200 OK响应。 11:业务相关服务节点2更新业务信息时，其应用服务单元向业务相关服务节点3的应用服务单元同

步该服务节点更新的业务信息。 12:业务相关服务节点3的应用服务单元向业务相关服务节点2的应用服务单元的MESSAGE方法消息

发送200 OK响应。

##### b) 业务信息同步消息流程2。

该同步流程的同步模式适用于授权信息通知和静态组信息同步，具体流程见图16。

在用户漫游时，与该用户有关的授权信息更新和组信息更新应同时同步至用户登录服务节点应用服

务单元和原归属服务节点应用服务单元。

业务发起节点

业务相关节点**1** 业务相关节点**2**

应用服务单元

应用服务单元

应用服务单元

**1:MESSAGE:GROUP/AUTHNOTIFY**

**2:200 OK**

**3:MESSAGE:OPTSTATUS**

**4:200 OK**

**5:MESSAGE:GROUP/AUTHNOTIFY**

**6:200 OK**

**7:MESSAGE:OPTSTATUS**

**8:200 OK**

图 16 业务信息同步消息流程 2 授权信息通知和静态组信息同步的级联同步流程2描述如下： 1:业务发起服务节点的应用服务单元向业务相关服务节点1的应用服务单元推送业务信息，MESSAGE



方法携带Control消息。

3:业务相关服务节点1更新信息时，其应用服务单元向业务发起服务节点的应用服务单元发送操作 反馈。

4:业务发起服务节点的应用服务单元向业务相关服务节点1的应用服务单元的MESSAGE方法操作反 馈消息发送200 OK响应。

5:业务发起服务节点的应用服务单元向业务相关服务节点2的应用服务单元推送业务信息，MESSAGE 方法携带Control消息，消息内容与1一致。

6:业务相关服务节点2的应用服务单元向业务发起服务节点的应用服务单元的MESSAGE方法消息发 送200 OK响应。

7:业务相关服务节点2更新信息时，其应用服务单元向业务发起服务节点的应用服务单元发送操作 反馈。

8:业务发起服务节点的应用服务单元向业务相关服务节点2的应用服务单元的MESSAGE方法操作反 馈消息发送200 OK响应。

##### c) 业务信息同步消息流程3。

该同步流程的同步模式适用于开启转发授权、停止转发授权，具体流程见图17。

业务相关节点**1**

业务相关节点**2**

应用服务单元

应用服务单元



**1:MESSAGE:BIZCNF/BIZCMD**

**2:200 OK**

图 17 业务信息同步消息流程 3 业务信息同步的级联同步流程3描述如下： 1:业务相关服务节点1需通知业务相关服务节点2主动发生业务变化时，业务相关服务节点1的应用

服务单元向业务相关服务节点2的应用服务单元推送需进行业务更改的信息。 2:业务相关服务节点2的应用服务单元向业务发起服务节点的应用服务单元的MESSAGE方法消息发

送200 OK响应。

##### d) 业务信息同步消息流程4。

该同步流程的同步模式适用于成员退出请求、成员退出响应、申请发言请求、申请发言响应，具体

流程见图18。

业务发起节点

业务相关节点**1** 业务相关节点**2**

应用服务单元

应用服务单元

应用服务单元



**1:MESSAGE:BIZCNF/BIZCMD**

**2:200 OK**

**3:MESSAGE:BIZCNF/BIZCMD**

**4:200 OK**

**5:MESSAGE:BIZCNF/BIZCMD**

**6:200 OK**

**7:MESSAGE:BIZCNF/BIZCMD**

**8:200 OK**

图 18 业务信息同步消息流程 4

业务信息的同步流程4描述如下：

1:业务相关服务节点1需向业务发起服务节点请求发生业务变化时，业务相关服务节点1的应用服务 单元向业务发起服务节点的应用服务单元推动业务更改请求，如成员退出请求或申请发言请求。

2:业务发起服务节点的应用服务单元向业务相关服务节点1的应用服务单元的MESSAGE方法消息发

送200 OK响应。 3:业务发起服务节点收到业务相关服务节点1的业务变化请求后，业务发起服务节点的应用服务单

元向业务相关服务节点1的应用服务单元发送业务更改请求的响应，如成员退出响应或申请发言响应。 4:业务相关服务节点1的应用服务单元向业务发起服务节点的应用服务单元的MESSAGE方法消息发

送200 OK响应。 5:业务发起服务节点同意业务相关服务节点1的业务变化请求后，向所有业务相关服务节点推送业

务更新信息。业务发起服务节点的应用服务单元向业务相关服务节点1的应用服务单元推送更新的业务 信息，如成员主动退出通知或成员主动申请发言通知。

6:业务相关服务节点1的应用服务单元向业务发起服务节点的应用服务单元的MESSAGE方法消息发 送200 OK响应。

7:业务发起服务节点的应用服务单元向业务相关服务节点2的应用服务单元推送更新的业务信息， 消息体与5中消息体一致，如成员主动退出通知或成员主动申请发言通知。

8:业务相关服务节点2的应用服务单元向业务发起服务节点的应用服务单元的MESSAGE方法消息发

送200 OK响应。

9.6 信息同步

9.6.1 信息同步基本要求

信息同步是多服务节点参与的点播、呼叫、设备控制、ZH和会议等业务或操作正常进行的前提。

作为应用服务单元之间的同步消息，应经服务节点间的信令控制服务单元路由透传完成。 多服务节点间的信息同步，包括状态信息同步、路由信息同步和漫游信息同步。

a) 状态信息同步机制和路由信息同步机制

状态信息同步和路由信息同步均采用基于订阅-发布模式的同步机制。 若存在多个上级服务节点，应选择一个可用的上级服务节点建立订阅-发布关系，用于状态信息和

路由信息的同步。当正在使用的上级服务节点发生异常时，应重新选择一个可用的上级服务节点建立订 阅-发布关系。切换正在使用的上级服务节点应支持自动和手动两种切换方式。

对于订阅-发布模式的同步机制，下级服务节点与上级服务单元互相注册成功后，选用一个上级服 务节点，下级服务节点应用服务单元向该上级服务节点应用服务单元发送订阅请求（SUBSCRIBE），上

级服务节点应用服务单元接收后，将已知的、正确的的全部状态信息和全部路由信息发送（NOTIFY）给 下级服务节点应用服务单元，并在订阅列表中新增该下级服务节点；下级服务节点应用服务单元向上级

服务节点应用服务单元上传（PUBLISH）已知的、正确的全部状态信息和全部路由信息；上级服务节点 应用服务单元接收后，应将信息下发（NOTIFY）至订阅列表中其它下级服务节点，并将信息逐级上传

（PUBLISH）至根服务节点，根服务节点再逐级下发至上述上级节点与其订阅列表外的其他服务节点。 下级服务节点应用服务单元第一次向上级服务节点应用服务单元上传（PUBLISH）本节点已知的、

正确的状态或路由信息应为全量信息，之后在状态或路由信息发生变化时上传增量变化信息。上级服务 节点应用服务单元第一次向下级服务节点应用服务单元发送（NOTIFY）本服务节点已知的、正确的状态

或路由信息应为全量信息，之后在状态或路由信息发生变化时发送增量变化信息。 当状态信息同步数据较大时，应将信息分批传输，每批可采用分包方式传输，分包方式应符合附录

A.1规定的分包格式。

基于订阅-发布模式的同步机制采用SUBSCRIBE、NOTIFY和PUBLISH方法实现。SUBSCRIBE、NOTIFY和 PUBLISH为不同SIP事务，在状态和路由信息同步中，应先完成发布订阅后，下级服务节点再使用PUBLISH

方法上传信息。

SUBSCRIBE请求中应包含Event和Expires头域，Event头域用于表明订阅事件的类型，Expires头域 用于表明订阅者期望的有效订阅时长。在有效期内应周期性发送SUBSCRIBE请求延长订阅时间。在不需

使用状态信息时，应取消订阅。应用服务单元每次启动新的订阅流程时，SUBSCRIBE请求的CSeq头域序 号值应从1开始累加。

SUBSCRIBE、NOTIFY和PUBLISH方法消息头Event头域为Event:presence。 NOTIFY和PUBLISH方法的Content-Type头域应符合附录A.1规定的格式。

NOTIFY请求中应包含扩展的Subscription-State头域，用于指示创建的订阅状态。 SUBSCRIBE、NOTIFY和PUBLISH方法应包含头域Subject用于携带请求发起应用服务单元，例如：

示例：Subject:<sip:应用服务单元编号@IP 地址:端口>。 SUBSCRIBE、NOTIFY和PUBLISH方法的使用应分别符合4.10、4.12和4.11的要求。 状态信息同步、路由信息同步请求命令采用附录A.5.4和A.5.5规定的XML格式。

b) 漫游时信息同步机制

当设备或用户发生漫游时，应采用漫游信息同步，保证设备或用户在漫游状态下能正确使用系统提 供的服务。

当设备或用户发生漫游时，其登录服务节点应用服务单元应向其归属服务节点应用服务单元发送请 求信息，归属服务节点应用服务单元将响应请求并将信息发送至登录服务节点的应用服务单元。

漫游时信息同步机制适用于权限信息请求与响应、分组信息请求与响应，预置位请求与响应。 权限信息请求与响应、分组信息请求与响应和预置位请求与响应采用MESSAGE方法携带规定的XML实

现。MESSAGE方法应包含头域Subject用于携带操作发起方信息。 漫游时信息同步机制适用的命令包括A.5.7预置位请求、A.5.8预置位响应、A.5.9权限信息请求、

A.5.10权限信息响应、A.5.11业务组信息请求和A.5.12业务组信息响应。

#### 9.6.2 信令流程

##### 9.6.2.1 状态与动态路由同步流程

状态与路由信息同步流程见图19。

子节点**1**



父节点 子节点**2**

应用服务单元

应用服务单元

应用服务单元

**1:SUBSCRIBE**

**2:200OK**

**3:NOTIFY:SYNCINFO/SYNCROUTELINK**

**4:200OK**

**5:PUBLISH: SYNCINFO/SYNCROUTELINK**

**6:200OK**

**7:NOTIFY:SYNCINFO/SYNCROUTELINK**

**9:PUBLISH: SYNCINFO/SYNCROUTELINK**

**8:200OK**

**10:200OK**

**11:NOTIFY:SYNCINFO/SYNCROUTELINK**

**12:200OK**

**13:SUBSCRIBE**

**14:200OK**

**15:SUBSCRIBE**

**16:200OK**

图 19 状态信息同步消息流程

基于订阅-发布级联同步的信息同步流程描述如下：

1:子服务节点1信令控制服务单元与父服务节点信令控制服务单元互相注册成功后，子服务节点1的 应用服务单元向其父服务节点应用服务单元发送订阅请求。

2:父服务节点应用服务单元对子服务节点1应用服务单元的订阅请求发送200 OK响应，并在订阅列 表中新增该子服务节点。

3:父服务节点应用服务单元在发出200 OK命令响应后立刻向发出向订阅请求的子服务节点1应用服 务单元发送初始同步的NOTIFY命令，通过该命令将父服务节点已知的全部信息（状态或路由）下发至子

服务节点1。

4:子服务节点1应用服务单元对父服务节点应用服务单元的NOTIFY命令发送200 OK响应。

5:子服务节点1应用服务单元向其父服务节点应用服务单元发送PUBILSH命令，通过该命令将该服务 节点已知的全部信息（状态或路由）上传至父服务节点应用服务单元。

6:父服务节点应用服务单元对子服务节点1应用服务单元的PUBILSH命令发送200 OK响应。

7:若父服务节点应用服务单元收到信息，则父服务节点应用服务单元应先校验信息是否存在更新， 再向订阅列表中其它子服务节点应用服务单元发送NOTIFY命令，通过该命令将子服务节点1的信息下发

至其它子服务节点应用服务单元。若该父服务节点也存在父服务节点，也应向其父服务节点应用服务单 元发送PUBLISH命令，通过该命令将增量信息发送至其父服务节点应用服务单元。

8:其它子服务节点应用服务单元对父服务节点应用服务单元的NOTIFY命令发送200 OK响应。

9:当子服务节点1发生信息变化时，子服务节点1应用服务单元向父服务节点发送PUBILSH命令，通 过该命令将该服务节点的增量信息变化上传至父服务节点应用服务单元。

10:父服务节点应用服务单元对子服务节点1应用服务单元的PUBILSH命令发送200 OK响应。

11:若父服务节点应用服务单元收到信息，则父服务节点应用服务单元向订阅列表中其它子服务节 点应用服务单元发送NOTIFY命令，通过该命令将子服务节点1的增量信息下发至其它子服务节点应用服

务单元。

12:其它子服务节点应用服务单元对父服务节点应用服务单元的NOTIFY命令发送200 OK响应。

13:子服务节点1的应用服务单元向其父服务节点应用服务单元发送刷新订阅请求。

14:父服务节点应用服务单元对子服务节点1应用服务单元的订阅请求发送200 OK响应，并在订阅列 表中更新子服务节点1的订阅时长。

15:在需要取消订阅时，子服务节点1的应用服务单元向其父服务节点应用服务单元发送取消订阅请

求。

16:父服务节点应用服务单元对子服务节点1应用服务单元的取消订阅请求发送200 OK响应，并在订

阅列表中删除子服务节点1的订阅。 子服务节点1也可定时周期性发送该服务节点的全部信息，定时器可设置，具体流程同9~12。

##### 9.6.2.2 漫游信息同步流程

应用服务单元之间的消息经由服务节点间的信令控制服务单元路由透传完成交互，其流程见图20。

拜访服务节点



归属服务节点

应用服务单元

|  |  |  |  |
| --- | --- | --- | --- |
|  | 应用服务单元 | | |
| **1:AUTHREQ/GROUPREQ**  **2:AUTHRES/GROUPRES** | |  |  |
|  | |

用户

用户

图 20 漫游信息同步消息流程

漫游信息级联同步流程描述如下：

1:用户于非归属服务节点的应用控制单元登录，该服务节点的应用服务单元将向用户归属服务节点 发送权限获取请求或分组信息请求命令。

2:用户归属服务节点响应权限请求或分组信息请求命令，并将权限信息或分组信息发送至用户拜访

服务节点。

#### 附 录 A (规范性附录) 控制及信息同步命令集

##### A.1命令格式

* 1. 命令格式

控制及信息同步命令集采用UTF-8编码，不区分大小写。控制及信息同步命令分为有应答命令与无 应答命令，命令分类如表A.1所示。有应答命令要求源功能实体发出命令后，目标功能实体应回复命令 进行应答，Control类型命令采用附录A.2.3定义的操作反馈命令进行应答。无应答命令无需目标功能实 体回复应答命令。可采用有应答命令或无应答命令对控制及信息同步命令集进行扩展。

控制及信息同步命令应支持分包方式传输，若消息体未进行分包，请求和响应命令消息头Content- Type值应为application/command+xml，若消息体存在分包，请求和响应命令消息头Content-Type值应 为application/command+xml+packet\_seq-len-cur-max，其中，seq表示消息内容序列号，len表示消息 内容总长度，cur表示当前包号，max表示分包总数。

控制及信息同步命令应支持zip方式压缩，若消息采用zip压缩，请求和响应命令消息头Content-

Encoding值应为zip。当信道类型为无线信道时，所传输的控制及信息同步命令应进行zip压缩。

表 A.1 命令分类

|  |  |  |
| --- | --- | --- |
| 分类 | 类型标识 | 分类说明 |
| 有应答命令 | control | 表示一个控制动作 |
| query | 表示一个查询动作 |
| request | 表示一个请求动作 |
| response | 表示一个请求动作的应答 |
| 无应答命令 | notify | 表示一个通知动作 |

命令类型的结构头定义如下。

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" targetNamespace=["http:](http://www.w3.org/namespace/)/[/www.w3.org/namespace/](http://www.w3.org/namespace/)"

xmlns:tg=["htt](http://www.w3.org/namespace/)p[://www.w3.org/namespace/"](http://www.w3.org/namespace/)>

<!--请求命令结构头文件定义-->

<xsd:choice maxOceurs="unbounded">

<xsd:element ref="tg:control"/>

<xsd:element ref="tg:request"/>

<xsd:element ref="tg:query"/>

<xsd:element ref="tg:notify"/>

</xsd:choice>

<!--应答命令结构头文件定义-->

<xsd:element ref="tg:Response"/>

本部分声明的全局数据类型如表A.2所示，全局类型是指XML Schema类型或是Schema元素的直接子 项，可以在不同位置被多次引用。

表 A.2 全局类型名称及其说明

|  |  |
| --- | --- |
| 名称 | 说明 |
| deviceIDType | 设备编码类型，11 位数字组成 |
| userIDType | 用户编码类型，11 位数字组成 |
| uuidType | UUID 类型，仅包含数字、大小写字母的字符串，最大长度 32 位 |

全局类型规定如下。

<!--设备编码类型-->

<xsd:simpleType name="deviceIDType">

<xsd:restriction base="ID">

<xsd:pattern value="\d{11}"/>

</xsd:restriction>

</xsd:simpleType>

<!--用户编码类型-->

<xsd:simpleType name="userIDType">

<xsd:restriction base="ID">

<xsd:pattern value="\d{11}"/>

</xsd:restriction>

</xsd:simpleType>

<!--uuid类型-->

<xsd:simpleType name="uuidType">

<xsd:restriction base="ID">

<xsd:maxLength value="32"/>

</xsd:restriction>

</xsd:simpleType>

##### A.2 基础命令

* 1. 基础命令
     1. 命令集

基础命令集如表A.3所示。

表 A.3 基础命令集

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 消息名称 | 消息标识 | 消息类型 | 承载方法类型 |
| 1 | 丢包重传请求 | resendreq | request | MESSAGE |
| 2 | 丢包重传响应 | resendres | response | MESSAGE |
| 3 | 操作反馈 | optstatus | response | MESSAGE |

* + 1. 丢包重传

###### 丢包重传

* + - 1. 丢包重传请求

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="xsd:string"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<!-- ts时间戳，格林威治时间自1970年1月1日(00:00:00 GMT)至当前时间的总毫秒数 -->

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- syncidentifer需重传消息序列号-->

<xsd:element name="syncidentifer" type="uuidType"/>

<!-- packet需要同步的包，数据格式为"a/b"，a表示需要重传的包号，b表示总包数-->

<xsd:element name="packet" type="xsd:string" minOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

示例：

<request>

<commandname>resendreq</commandname>

<!-- 当前消息序列号 -->

<seq>8106b5c18e1111e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<!-- 需重传消息序列号 -->

<syncidentifer>8bad5ce08e1111e8b445060400ef5315</syncidentifer>

<packet>1/5</packet>

</request>

###### 丢包重传响应 丢包重传响应用于对丢包重传请求进行应答

* + - 1. 丢包重传响应 丢包重传响应用于对丢包重传请求进行应答。 XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:element name="response">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="xsd:string"/>

<!-- seq命令序号，等于请求消息中的序号值 -->

<xsd:element name="seq" type="uuid"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- syncidentifer重传消息序列号-->

<xsd:element name="syncidentifer" type="xsd:string"/>

<!-- packet需要同步的包，数据格式为"a/b"，a表示同步的包号，b表示总包数-->

<xsd:element name="packet" type="xsd:string" minOccurs="1"/>

<!-- content需要重传内容，此处为CDATA区段-->

<xsd:element name="content" type="xsd:string" minOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

示例：

<response>

<commandname>resendres</commandname>

<seq>8106b5c18e1111e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<syncidentifer>8bad5ce08e1111e8b445060400ef5315</syncidentifer>

<packet>1/5</packet>

<content>

<![CDATA[

<!-- 此处填需重传的内容-->

]]>

</content>

</response>

###### 操作反馈 操作反馈用于对control类型命令进行应答

* + 1. 操作反馈 操作反馈用于对control类型命令进行应答。 XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:element name="response">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string"/>

<!-- 被响应消息的序列号 -->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="response" type="xsd:string"/>

<!-- 异常操作反馈错误描述 -->

<xsd:element name="error" type="xsd:string" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

示例1：正常操作反馈

<response>

<commandname>optstatus</commandname>

<response>200</response>

<seq>64fc6c0a8e1411e8b445060400ef5315</seq>

<ts>1533630770000</ts>

</response>

示例2：异常操作反馈

<response>

<commandname>optstatus</commandname>

<response>401</response>

<seq>64fc6c0a8e1411e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<error>describe</error>

</response>

##### A.3 设备控制命令

* + 1. 命令集

设备控制命令集如表A.4所示。

表 A.4 设备控制命令集

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 消息名称 | 消息标识 | 消息类型 | 承载方法类型 |
| 1 | 云镜控制 | ptzctrl | control | MESSAGE |
| 2 | 预置位操作 | pointindexoperation | control | MESSAGE |
| 3 | 字幕设置 | setosd | control | MESSAGE |
| setosds |
| customosd |
| 4 | I 帧间隔配置 | iframeconfreq | control | MESSAGE |
| 5 | I 帧申请 | - | - | INFO |

###### 云镜控制

* + 1. 云镜控制

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 镜头变倍 -->

<xsd:simpleType name="zoomctrl\_type">

<xsd:restriction base="xsd:string">

<!-- 镜头放大 -->

<xsd:enumeration value="in"/>

<!-- 镜头缩小 -->

<xsd:enumeration value="out"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 聚焦 -->

<xsd:simpleType name="focusctrl\_type">

<xsd:restriction base="xsd:string">

<!-- 近 -->

<xsd:enumeration value="near"/>

<!-- 远 -->

<xsd:enumeration value="far"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 光圈 -->

<xsd:simpleType name="aperturectrl\_type">

<xsd:restriction base="xsd:string">

<!-- 放大 -->

<xsd:enumeration value="+"/>

<!-- 缩小 -->

<xsd:enumeration value="-"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 云台垂直方向运动 -->

<xsd:simpleType name="tilt\_direction\_type">

<xsd:restriction base="xsd:string">

<!-- 上 -->

<xsd:enumeration value="u"/>

<!-- 下 -->

<xsd:enumeration value="d"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 云台水平方向运动 -->

<xsd:simpleType name="pan\_direction\_type">

<xsd:restriction base="xsd:string">

<!-- 左 -->

<xsd:enumeration value="l"/>

<!-- 右 -->

<xsd:enumeration value="r"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 取值范围0-255的整数，可用于控制云台运动速度相对值、聚焦速度、光圈速度 -->

<!-- 相对速度最大值255，若设备实际可控速度为[a,b]，设置速度值为x时，则设备实际速度值为 a+(b-a)/255\*x -->

<xsd:simpleType name="Integer0\_255">

<xsd:restriction base="xsd:integer">

<xsd:maxInclusive value="255"/>

<xsd:minInclusive value="0"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 取值范围0-127的整数，可用于控制镜头变倍速度相对值 -->

<!-- 相对速度最大值127，若设备实际可控速度为[a,b]，设置速度值为x时，则设备实际速度值为 a+(b-a)/127\*x -->

<xsd:simpleType name="Integer0\_127">

<xsd:restriction base="xsd:integer">

<xsd:maxInclusive value="127"/>

<xsd:minInclusive value="0"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 操作类型 -->

<xsd:simpleType name="operation\_type">

<xsd:restriction base="xsd:string">

<!-- 开始 -->

<xsd:enumeration value="start"/>

<!-- 停止 -->

<xsd:enumeration value="stop"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 云台垂直运动控制（包括方向和速度相对值） -->

<xsd:complexType name="tilt\_servo\_type">

<xsd:sequence>

<xsd:element name="direction" type="tilt\_direction\_type" minOccurs="1"/>

<xsd:element name="speed" type="Integer0\_255" minOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

<!-- 云台水平运动控制（包括方向和速度） -->

<xsd:complexType name="pan\_servo\_type">

<xsd:sequence>

<xsd:element name="direction" type="pan\_direction\_type" minOccurs="1"/>

<xsd:element name="speed" type="Integer0\_255" minOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

<!-- 镜头变倍控制（放大或缩小及速度相对值） -->

<xsd:complexType name="zoom\_type">

<xsd:sequence>

<xsd:element name="zoomctrl" type="zoomctrl\_type" minOccurs="1"/>

<xsd:element name="speed" type="Integer0\_127" minOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

<!--聚焦控制（远或近及速度） -->

<xsd:complexType name="focus\_type">

<xsd:sequence>

<xsd:element name="focusctrl" type="focusctrl\_type" minOccurs="1"/>

<xsd:element name="speed" type="Integer0\_255" minOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

<!--光圈控制（放大或缩小及速度） -->

<xsd:complexType name="aperture\_type">

<xsd:sequence>

<xsd:element name="aperturectrl" type="aperturectrl\_type" minOccurs="1"/>

<xsd:element name="speed" type="Integer0\_255" minOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

<!-- 设备控制 -->

<xsd:element name="control">

<xsd:complexType>

<xsd:sequence>

<!-- 命令名称（必选）-->

<xsd:element name="commandname" type="xsd:string"/>

<!-- 操作类型（必选）-->

<xsd:element name="operation" type="operation\_type"/>

<!-- 操作序列号值（必选） -->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 设备名称（必选） -->

<xsd:element name="devname" type="devnameType"/>

<!-- 设备ID（必选） -->

<xsd:element name="devid" type="deviceIDType"/>

<!-- 云台垂直运动控制（可选） -->

<xsd:element name="tiltservo" type="tilt\_servo\_type" minOccurs="0" maxOccurs="1" />

<!-- 云台水平运动控制（可选） -->

<xsd:element name="panservo" type="pan\_servo\_type" minOccurs="0" maxOccurs="1" />

<!-- 镜头变倍控制（可选） -->

<xsd:element name="zoom" type="zoom\_type" minOccurs="0" maxOccurs="1"/>

<!-- 聚焦远近控制（可选） -->

<xsd:element name="focus" type="focus\_type" minOccurs="0" maxOccurs="1"/>

<!-- 光圈放大缩小控制（可选） -->

<xsd:element name="aperture" type="aperture\_type" minOccurs="0" maxOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

示例1：云台以127的相对速度值向上方向运动

<control>

<commandname>ptzctrl</commandname>

<operation>start</operation>

<seq>66edf2e88e1611e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000003</devid>

<tiltservo>

<direction>u</direction>

<speed>127</speed>

</tiltservo>

</control>

示例2：镜头以64的相对速度值变倍放大

<control>

<commandname>ptzctrl</commandname>

<operation>start</operation>

<seq>66edf2e88e1611e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000003</devid>

<zoom>

<zoomctrl>in</zoomctrl>

<speed>64</speed>

</zoom>

</control>

示例3：镜头以128的速度值焦距近

<control>

<commandname>ptzctrl</commandname>

<operation>start</operation>

<seq>66edf2e88e1611e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000003</devid>

<focus>

<focusctrl>near</focusctrl>

<speed>128</speed>

</focus>

</control>

示例4：镜头以128的速度值缩小光圈

<control>

<commandname>ptzctrl</commandname>

<operation>start</operation>

<seq>66edf2e88e1611e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000003</devid>

<aperture>

<aperturectrl>-</aperturectrl>

<speed>128</speed>

</aperture>

</control>

示例5：组合实例，云台以127的速度值向上方向运动，同时以50的速度值向右方向运动，实际是斜向右上方向运行；与此同时，镜头以64的速度值变倍缩小

<control>

<commandname>ptzctrl</commandname>

<operation>start</operation>

<seq>66edf2e88e1611e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000003</devid>

<tiltservo>

<direction>u</direction>

<speed>127</speed>

<tiltservo>

<panservo>

<direction>r</direction>

<speed>50</speed>

</panservo>

<zoom>

<zoomctrl>out</zoomctrl>

<speed>64</speed>

</zoom>

</control>

示例6：PTZ控制止

<control>

<commandname>ptzctrl</commandname>

<operation>stop</operation>

<seq>66edf2e88e1611e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000003</devid>

</control>

###### 预置位操作

* + 1. 预置位操作

若需跨服务节点增加或删除设备预置位信息，相关操作应同时同步至设备归属服务节点的应用服务

单元。

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:simpleType name="operation\_type">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="add"/>

<xsd:enumeration value="remove"/>

<xsd:enumeration value="invoke"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="Integer0\_255">

<xsd:restriction base="xsd:integer">

<xsd:maxInclusive value="255"/>

<xsd:minInclusive value="0"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="control">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string" fixed="pointindexoperation"/>

<xsd:element name="operation" type="operation\_type"/>

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="devname" type="xsd:string"/>

<xsd:element name="devid" type="deviceIDType"/>

<!-- 预置位名称 -->

<xsd:element name="pointindexname" type="xsd:string" minOccurs="0" />

<!-- 预置位索引值 -->

<xsd:element name="pointindexvalue" type="Integer0\_255"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

示例1：增加/设置预置位

<control>

<commandname>pointindexoperation</commandname>

<operation>add</operation>

<seq>c8591d048e1711e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000003</devid>

<pointindexname>领导1</pointindexname>

<pointindexvalue>128</pointindexvalue >

</control>

示例2：调用预置位

<control>

<commandname>pointindexoperation</commandname>

<operation>invoke</operation>

<seq>c8591d048e1711e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000003</devid>

<pointindexvalue>128</pointindexvalue >

</control>

示例3：删除预置位

<control>

<commandname>pointindexoperation</commandname>

<operation>remove</operation>

<seq>c8591d048e1711e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000003</devid>

<pointindexvalue>255</pointindexvalue>

</control>

###### 字幕设置

* + 1. 字幕设置

单行字幕设置

* + - 1. 单行字幕设置

叠加字符宏定义为：OSD\_DATE是叠加系统当前日期，格式为YYYY-MM-DD；OSD\_TIME是叠加系统当前

时间，格式为HH:mm:SS；OSD\_DATETIME是叠加系统当前日期及时间，格式为YYYY-MM-DD HH:mm:SS； OSD\_NAME是叠加当前终端/设备名称。

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:simpleType name="Integer0\_15">

<xsd:restriction base="xsd:integer">

<xsd:maxInclusive value="15"/>

<xsd:minInclusive value="0"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="Integer0\_7">

<xsd:restriction base="xsd:integer">

<xsd:maxInclusive value="7"/>

<xsd:minInclusive value="0"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="Integer1\_9">

<xsd:restriction base="xsd:integer">

<xsd:maxInclusive value="9"/>

<xsd:minInclusive value="1"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="Integer0\_99">

<xsd:restriction base="xsd:integer">

<xsd:maxInclusive value="99"/>

<xsd:minInclusive value="0"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="FontName\_type">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="宋体"/>

<xsd:enumeration value="黑体"/>

<xsd:enumeration value="楷体"/>

<xsd:enumeration value="仿宋"/>

<xsd:enumeration value="微软雅黑"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="Color\_type">

<xsd:restriction base="xsd:string">

<xsd:length value="11"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 字幕内容，字幕最大长度为32个全角字符 -->

<xsd:simpleType name="Text\_type">

<xsd:restriction base="xsd:string">

<xsd:maxLength value="32"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="control">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string"/>

<xsd:element name="seq" type="seqType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="devname" type="devnameType"/>

<!-- 设备编码id类型 -->

<xsd:element name="devid" type="deviceIDType"/>

<!-- OSD行号（可选，取值0-7），缺省为0 -->

<xsd:element name="index" type="Integer0\_7"/>

<!-- 是否显示，0-不显示 1-显示（可选），缺省为1 -->

<xsd:element name="show" type="xsd:boolean"/>

<!-- OSD居左占比（与屏幕宽度的相对占比，取值0-99，0表示最靠左）（可选），缺省为20 -->

<xsd:element name="left" type="Integer0\_99"/>

<!-- OSD居上占比（与屏幕高度的相对占比，取值0-99，0表示最靠上）（可选），当行号为0时缺 省为20，当行号为1时缺省为30，当行号为2时缺省为40，当行号为3时缺省为50，当行号为4时缺省为60， 当行号为5时缺省为70，当行号为6时缺省为80，当行号为7时缺省为90 -->

<xsd:element name="top" type="Integer0\_99"/>

<!-- OSD字体名称（可选），缺省为"宋体" -->

<xsd:element name="fontname" type="FontName\_type"/>

<!-- OSD字体大小（可选，1-9表示字体大小相对值，代表字体大小支持9档调节，缺省为5，字体 大小相对值应根据解码设备实际单位完成映射 -->

<xsd:element name="fontsize" type="Integer1\_9"/>

<!-- OSD字体颜色，字符格式（R,G,B）（R=0-255,G=0-255,B=0-255）（可选），缺省为"255-255- 255" -->

<xsd:element name="color" type="Color\_type" default="255-255-255"/>

<!-- OSD字体内容（必选），UTF-8编码，空表示清空字幕 -->

<xsd:element name="text" type="Text\_type"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

示例1：添加字幕，可选项均取缺省值情况（字幕行号值0、显示字幕、字幕位置（20%,20%）、宋体、字体相对大小5、 字体颜色255-255-255），实际体现为在左上角位置显示白色字幕“a test osd”

<control>

<commandname>setosd</commandname>

<seq>69d66d4e8e1d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000001</devid>

<text>a test osd</text>

</control>

示例2：添加字幕，部分可选项为缺省值情况（字幕行号值7、显示字幕、字幕位置（20%,80%）、黑体、字体相对大小

9、字体颜色255-000-000），实际体现为在左下角位置显示黑体、字体相对大小为9、红色字幕“a test osd”

<control>

<commandname>setosd</commandname>

<seq>69d66d4e8e1d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000001</devid>

<index>7</index>

<fontname>黑体</fontname>

<fontsize>9</fontsize>

<color>255-000-000</color>

<text>a test osd</text>

</control>

示例3：添加字幕，均为自定义值情况（字幕行号值3、显示字幕、字幕位置（50%,50%）、楷体、字体相对大小1、字

体颜色000-255-000），实际体现为在较中间位置显示楷体、字体相对大小为1、绿色字幕“a test osd”

<control>

<commandname>setosd</commandname>

<seq>69d66d4e8e1d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000001</devid>

<index>3</index>

<show>1</show>

<left>50</left>

<top>50</top>

<fontname>楷体</fontname>

<fontsize>1</fontsize>

<color>000-255-000</color>

<text>a test osd</text>

</control>

示例4：添加系统当前日期及时间，实际体现为在左上角位置显示宋体、字体相对大小为5、白色的系统当前日期及时 间

<control>

<commandname>setosd</commandname>

<seq>69d66d4e8e1d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000001</devid>

<index>3</index>

<text>osd\_datetime</text>

</control>

示例5：擦除行号值为0的字幕

<control>

<commandname>setosd</commandname>

<seq>69d66d4e8e1d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000001</devid>

<show>0</show>

<text></text>

</control>

示例6：擦除行号值为3的字幕

<control>

<commandname>setosd</commandname>

<seq>69d66d4e8e1d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000001</devid>

<index>3</index>

<show>0</show>

<text></text>

</control>

多行字幕设置

* + - 1. 多行字幕设置

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 字幕单项信息，每个字段含义、是否为必选与单行字幕设置完全一致 -->

<xsd:complexType name="osditem\_type">

<xsd:sequence>

<xsd:element name="index" type="Integer0\_7"/>

<xsd:element name="show" type="xsd:boolean"/>

<xsd:element name="left" type="Integer0\_99"/>

<xsd:element name="top" type="Integer0\_99"/>

<xsd:element name="fontname" type="FontName\_type"/>

<xsd:element name="fontsize" type="Integer1\_9"/>

<xsd:element name="color" type="Color\_type" default="255-255-255"/>

<xsd:element name="text" type="Text\_type"/>

</xsd:sequence>

</xsd:complexType>

<!-- 字幕列表 -->

<xsd:complexType name="osdlist\_type">

<xsd:sequence>

<xsd:element name="osditem" type="osditem\_type" minOccurs="1" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="control">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string"/>

<xsd:element name="seq" type="seqType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="devname" type="devnameType"/>

<xsd:element name="devid" type="deviceIDType"/>

<xsd:element name="osdlist" type="osdlist\_type"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

示例1：同时添加多行字幕

<control>

<commandname>setosds</commandname>

<seq>69d66d4e8e1d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000001</devid>

<osdlist>

<osditem>

<index>0</index>

<show>1</show>

<left>50</left>

<top>50</top>

<fontname>楷体</fontname>

<fontsize>1</fontsize>

<color>000-255-000</color>

<text>a test osd</text>

</osditem>

<osditem>

<index>2</index>

<text>osd\_datetime</text>

</osditem>

</osdlist>

</control>

示例2：同时擦除多行字幕

<control>

<commandname>setosds</commandname>

<seq>69d66d4e8e1d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000001</devid>

<osdlist>

<osditem>

<index>0</index>

<show>0</show>

<text></text>

</osditem>

<osditem>

<index>2</index>

<show>0</show>

<text></text>

</osditem>

</osdlist>

</control>

自定义字幕设置

* + - 1. 自定义字幕设置

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:element name="control">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string"/>

<xsd:element name="seq" type="seqType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="devname" type="devnameType"/>

<!-- 设备编码id类型 -->

<xsd:element name="devid" type="deviceIDType"/>

<!-- 自定义字幕类型标识 -->

<xsd:element name="customtype" type="xsd:string"/>

<!-- 自定义字幕详细信息 -->

<xsd:element name="customdata" type="xsd:string"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

示例1：自定义字幕

<control>

<commandname>customosd</commandname>

<seq>69d66d4e8e1d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000001</devid>

<customtype>1</customtype>

<customdata>

<!-- 此处填自定义字幕的内容-->

</customdata>

</control>

I帧间隔配置

* + 1. I帧间隔配置

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:element name="control">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string"/>

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="devname" type="xsd:string"/>

<xsd:element name="devid" type="deviceIDType"/>

<!-- Iframedelta是配置的I帧间隔，以毫秒为单位-->

<xsd:element name="iframedelta" type="xsd:integer"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

示例：

<control>

<commandname>iframeconfreq</commandname>

<seq>44b4ba2c8e1b11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000001</devid>

<iframedelta>3000</iframedelta>

</control>

帧申请 在会话进行过程中，沿着信令控制路径更新I帧，通过INFO请求发送。

* + 1. I帧申请 在会话进行过程中，沿着信令控制路径更新I帧，通过INFO请求发送。 XML结构描述如下：

<?xml version="1.0" encoding="utf-8" ?>

<xs:schema id="TightMediaControl" elementFormDefault="qualified" xmlns:xs="<http://www.w3.org/2001/XMLSchema>">

<xs:element name="media\_control">

<xs:complexType>

<xs:sequence>

<xs:element name="vc\_primitive" type="vc\_primitive" minOccurs="0" maxOccurs="unbounded" />

<xs:element name="general\_error" type="xs:string" minOccurs="0" maxOccurs="unbounded"

/>

</xs:sequence>

</xs:complexType>

</xs:element>

<!-- Video control primitive. -->

<xs:complexType name="vc\_primitive">

<xs:sequence>

<xs:element name="to\_encoder" type="to\_encoder" />

<xs:element name="stream\_id" type="xs:string" minOccurs="0" maxOccurs="unbounded" />

</xs:sequence>

</xs:complexType>

<!-- Encoder Command: Picture Fast Update -->

<xs:complexType name="to\_encoder">

<xs:choice>

<xs:element name="picture\_fast\_update"/>

</xs:choice>

</xs:complexType>

</xs:schema> 应用举例如下：

示例：

<?xml version='1.0' encoding='utf-8'

?>

<media\_control>

<vc\_primitive>

<to\_encoder>

<picture\_fast\_update>

</picture\_fast\_update>

</to\_encoder>

</vc\_primitive>

</media\_control>

##### A.4媒体控制命令

A.4 媒体控制命令

* + 1. 命令集

媒体处理服务单元在与信令控制服务单元进行会话协商时，所提供SDP的视音频媒体类型描述（m=）

部分应包含方向属性，并对每个媒体类型增加RTP流ID属性。 RTP流ID属性描述如下：

a=rtpid:<rtp stream id value>[<rtp stream id value>] rtpid取值为32位随机正整数。rtpid参数仅在从媒体处理服务单元到信令控制服务单元的SDP消息

中有效，若在信令控制服务单元到媒体处理服务单元的SDP中含有rtpid参数，则该参数无效。 如果媒体处理服务单元作为SDP的请求方，则应采用a=sendrecv方向属性；如果媒体处理服务单元

作为SDP的应答方，则可以采用a=sendrecv、recvonly、sendonly、inactive方向属性。

媒体处理服务单元的SDP，若采用a=sendrecv方向属性，a=rtpid:<recv rtp id><send rtp id>， 分别表示接收和发送的媒体流id；若采用a=recvonly方向属性，a=rtpid:<recv rtp id>，表示接收的 媒体流id；若采用a=sendonly方向属性，a=rtpid:<send rtp id>，表示发送的媒体流id。

采用呼叫号码、RTP流ID唯一标识一个媒体链路，该媒体链路可以作为媒体调度的信源或信宿。媒 体流方向标识以媒体处理服务单元为中心，即具备sendonly标识，媒体处理服务单元可以对外作为信源；

具备recvonly标识，媒体处理服务单元可以对外作为信宿，具备sendrecv标识，媒体处理服务单元可以

对外作为信源或信宿。 媒体控制命令集如表A.5所示。

表 A.5 媒体控制命令集

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 消息名称 | 消息标识 | 消息类型 | 承载方法类型 |
| 1 | 媒体推送 | mediapush | control | MESSAGE |
| 2 | 建立呼叫 | mediacall | control | MESSAGE |
| 3 | 存储设置 | storeset | control | MESSAGE |
| 4 | 存储通知 | storenotify | notify | MESSAGE |
| 5 | 检索请求 | filequery | query | MESSAGE |
| 6 | 检索响应 | querylist | response | MESSAGE |
| 7 | 转发设置 | switchset | control | MESSAGE |
| 8 | 传输通知 | chanstatus | notify | MESSAGE |
| 9 | 媒体下载 | mediadownload | control | MESSAGE |
| 10 | 文件结尾通知 | filetoend | notify | MESSAGE |

###### 媒体推送 媒体推送命令主要用于应用服务单元控制信令控制单元为编解码单元建立单向点播关系。

* + 1. 媒体推送 媒体推送命令主要用于应用服务单元控制信令控制单元为编解码单元建立单向点播关系。 XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:include schemaLocation="common.xsd"/>

<!-- 媒体推送操作，start表示开始、stop表示停止 -->

<xsd:simpleType name="operation\_type">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="start"/>

<xsd:enumeration value="stop"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="audio\_type">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="g.711"/>

<xsd:enumeration value="aac-lc"/>

<xsd:enumeration value="g.729a"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="video\_type">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="h.264"/>

<xsd:enumeration value="h.265"/>

<xsd:enumeration value="javs"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="ring\_type">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="online"/>

<xsd:enumeration value="offline"/>

<xsd:enumeration value="ringing"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:complexType name="src\_type">

<xsd:sequence>

<!-- userid可以为用户ID或设备ID -->

<xsd:element name="userid" type="userIDType" minOccurs="1"/>

<xsd:element name="postfix" type="xsd:string" minOccurs="0"/>

<xsd:element name="ring" type="ring\_type"/>

<!-- 若包含url值，表示点播用户或设备的媒体文件；若不包含url值，表示直接点播用户或设备-

->

<xsd:element name="url" type="xsd:string" minOccurs="0"/>

<xsd:element name="video" type="video\_type" default="h.264" minOccurs="0"/>

<xsd:element name="audio" type="audio\_type" default="g.711" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="control">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string"/>

<xsd:element name="operation" type="operation\_type"/>

<xsd:element name="seq" type="xsd:string"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="src" type="src\_type"/>

<xsd:element name="dst" type="src\_type"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

示例1：点播建立

<control>

<commandname>mediapush</commandname>

<operation>start</operation>

<seq>660bf8d0946411e8b446060400ef5315</seq>

<ts>1533630770000</ts>

<src>

<userid>00000000001</userid>

<ring>online</ring>

<url><http://mediaserver/file1></url>

<video>h.264</video>

<audio>g.711</audio>

</src>

<dst>

<userid>00000000002</userid>

<video>h.264</video>

<audio>g.711</audio>

</dst>

</control>

示例2：点播停止

<control>

<commandname>mediapush</commandname>

<operation>stop</operation>

<seq>660bf8d0946411e8b446060400ef5315</seq>

<ts>1533630770000</ts>

<src>

<userid>00000000001</userid>

<ring>online</ring>

<url><http://mediaserver/file1></url>

<video>h.264</video>

<audio>g.711</audio>

</src>

<dst>

<userid>00000000002</userid>

<video>h.264</video>

<audio>g.711</audio>

</dst>

</control>

###### 建立呼叫 建立呼叫送命令主要用于应用服务单元控制信令控制单元为编解码单元建立双向呼叫关系。

* + 1. 建立呼叫 建立呼叫送命令主要用于应用服务单元控制信令控制单元为编解码单元建立双向呼叫关系。 XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:include schemaLocation="common.xsd"/>

<!-- 呼叫操作，start表示开始、stop表示停止 -->

<xsd:simpleType name="operation\_type">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="start"/>

<xsd:enumeration value="stop"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:complexType name="src\_type">

<xsd:sequence>

<xsd:element name="userid" type="userIDType" minOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="control">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string"/>

<xsd:element name="operation" type="operation\_type"/>

<xsd:element name="seq" type="xsd:string"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="src" type="src\_type"/>

<xsd:element name="dst" type="src\_type"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

示例1：呼叫建立

<control>

<commandname>mediacall</commandname>

<operation>start</operation>

<seq>660bf8d0946411e8b446060400ef5315</seq>

<ts>1533630770000</ts>

<src>

<userid>00000000001</userid>

</src>

<dst>

<userid>00000000002</userid>

</dst>

</control>

示例2：呼叫停止

<control>

<commandname>mediacall</commandname>

<operation>stop</operation>

<seq>660bf8d0946411e8b446060400ef5315</seq>

<ts>1533630770000</ts>

<src>

<userid>00000000001</userid>

</src>

<dst>

<userid>00000000002</userid>

</dst>

</control>

##### A.4.4媒体存储

* + 1. 媒体存储

###### 存储设置

* + - 1. 存储设置

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:simpleType name="fileformat\_type">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="mp4"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="operation\_type">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="start"/>

<xsd:enumeration value="stop"/>

<xsd:enumeration value="pause"/>

<xsd:enumeration value="resume"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="control">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string"/>

<xsd:element name="operation" type="operation\_type"/>

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 需存储的资源ID，资源可以是编码设备、用户等 -->

<xsd:element name="resid" type="deviceIDType"/>

<xsd:element name="filename" type="xsd:string"/>

<!-- 当未携带fileformat字段时，不限制存储文件的格式 -->

<xsd:element name="fileformat" type="fileformat\_type" minOccurs="0"/>

<!-- 存储时间以秒为单位，若未携带timeduration字段时，应由结束存储指令控制存储时长 -->

<xsd:element name="timeduration" type="xsd:unsignedinteger" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

示例1：开始存储

<control>

<commandname>storeset</commandname>

<operation>start</operation>

<seq>660bf8d4946411e8b446060400ef5315</seq>

<ts>1533630770000</ts>

<resid>00000000001</resid>

<filename>helloworld</filename>

<fileformat>mp4</fileformat>

<timeduration>10</timeduration>

</control>

示例2：结束存储

<control>

<commandname>storeset</commandname>

<operation>stop</operation>

<seq>660bf8d4946411e8b446060400ef5315</seq>

<ts>1533630770000</ts>

<resid>00000000001</resid>

<filename>helloworld</filename>

</control>

示例3：暂停存储

<control>

<commandname>storeset</commandname>

<operation>pause</operation>

<seq>660bf8d4946411e8b446060400ef5315</seq>

<ts>1533630770000</ts>

<resid>00000000001</resid>

<filename>helloworld</filename>

</control> 示例4：接续存储

<control>

<commandname>storeset</commandname>

<operation>resume</operation>

<seq>660bf8d4946411e8b446060400ef5315</seq>

<ts>1533630770000</ts>

<resid>00000000001</resid>

<filename>helloworld</filename>

</control>

###### 存储通知

* + - 1. 存储通知

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!--设备编码类型-->

<xsd:simpleType name="deviceIDType">

<xsd:restriction base="ID">

<xsd:pattern value="\d{11}"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string"/>

<xsd:element name="seq" type="xsd:string"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 需存储的资源ID，资源可以是编码设备、用户等 -->

<xsd:element name="resid" type="deviceIDType"/>

<xsd:element name="filename" type="xsd:string"/>

<xsd:element name="error" type="xsd:string"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema> 应用举例如下：

示例：

<notify>

<commandname>storenotify</commandname>

<seq>660bf8d1946411e8b446060400ef5315</seq>

<ts>1533630770000</ts>

<resid>00000000001</resid>

<filename>helloworld</filename>

<error>nospace</error>

</notify>

##### 媒体检索

* + 1. 媒体检索

###### 检索请求

* + - 1. 检索请求

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:include schemaLocation="common.xsd"/>

<!-- 文件检索模式 -->

<xsd:simpleType name="mode\_type">

<xsd:restriction base="xsd:string">

<!-- 名称检索 -->

<xsd:enumeration value="file"/>

<!-- 时间检索 -->

<xsd:enumeration value="time"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 文件检索 -->

<xsd:element name="query">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string"/>

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="mode" type="mode\_type"/>

<!-- 按名称检索的文件名称 -->

<xsd:element name="filename" type="xsd:string" minOccurs="0"/>

<!-- 按时间检索的开始时间 -->

<xsd:element name="recordstarttime" type="xsd:string" minOccurs="0" />

<!-- 按时间检索的结束时间 -->

<xsd:element name="recordstoptime" type="xsd:string" minOccurs="0"/>

<!-- 检索文件的开始索引号 -->

<xsd:element name="beginindex" type="xsd:integer" minOccurs="1"/>

<!-- 检索结果包含的最大文件数量 -->

<xsd:element name="maxpagelen" type="xsd:integer" minOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

示例1：按名称检索，beginindex的取值为1

<query>

<commandname>filequery</commandname>

<seq>3c7982da93db11e8b446060400ef5315</seq>

<ts>1533630770000</ts>

<mode>file</mode>

<filename>hello</filename>

<beginindex>1</beginindex>

<maxpagelen>10</maxpagelen>

</query>

示例2：按时间检索，beginindex的取值为4

<query>

<commandname>filequery</commandname>

<seq>3c7982da93db11e8b446060400ef5315</seq>

<ts>1533630770000</ts>

<mode>time</mode>

<recordstarttime>2014/05/04/12:00:00</recordstarttime>

<recordstoptime>2014/05/05/12:00:00</recordstoptime>

<beginindex>4</beginindex>

<maxpagelen>10</maxpagelen>

</query>

###### 检索响应

* + - 1. 检索响应 检索请求用于对媒体检索请求进行应答。 XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:include schemaLocation="common.xsd"/>

<!-- 单文件描述 -->

<xsd:complexType name="fileproperty\_type">

<xsd:sequence>

<xsd:element name="url" type="xsd:string"/>

<xsd:element name="filename" type="xsd:string"/>

<xsd:element name="recordtime" type="xsd:string"/>

<!-- filesize以字节为单位计算文件大小 -->

<xsd:element name="filesize" type="xsd:integer"/>

<!-- fileduration以秒为单位 -->

<xsd:element name="fileduration" type="xsd:integer"/>

<xsd:element name="filedescript" type="xsd:string"/>

</xsd:sequence>

</xsd:complexType>

<!-- 文件检索模式 -->

<xsd:simpleType name="mode\_type">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="file"/>

<xsd:enumeration value="time"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 文件列表 -->

<xsd:complexType name="filelist\_type">

<xsd:sequence>

<xsd:element name="fileproperty" type="fileproperty\_type" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<!-- 文件检索响应 -->

<xsd:element name="response">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string"/>

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="mode" type="mode\_type"/>

<!-- 检索的文件结果总数 -->

<xsd:element name="totalnum" type="xsd:integer" minOccurs="0"/>

<!-- 检索的文件结果的开始索引 -->

<xsd:element name="beginindex" type="xsd:integer" minOccurs="0"/>

<!-- 检索的文件结果的结束索引 -->

<xsd:element name="endindex" type="xsd:integer" minOccurs="0"/>

<!-- 检索结果文件列表 -->

<xsd:element name="filelist" type="filelist\_type" minOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

示例1：按名称检索的响应

<response>

<commandname>querylist</commandname>

<seq>3c7982da93db11e8b446060400ef53151</seq>

<ts>1533630770000</ts>

<mode>file</mode>

<totalnum>2</totalnum>

<beginindex>1</beginindex>

<endindex>2</endindex>

<filelist>

<fileproperty>

<url><http://mediaserver/helloworld></url>

<filename>helloworld</filename>

<recordtime>2014/05/05/12:00:00</recordtime>

<filensize>32154</filensize>

<fileduration>01:00:00</fileduration>

<filedescript>gate safegurd</filedescript>

</fileproperty>

<fileproperty>

<url><http://mediaserver/helloabc></url>

<filename>helloabc</filename>

<recordtime>2014/05/05/12:00:00</recordtime>

<filensize>32154</filensize>

<fileduration>01:00:00</fileduration>

<filedescript>window safegurd</filedescript>

</fileproperty>

</filelist>

</response>

示例2：按名称检索的响应，检索结果为空

<response>

<commandname>querylist</commandname>

<seq>3c7982da93db11e8b446060400ef53151</seq>

<ts>1533630770000</ts>

<mode>file</mode>

<totalnum>0</totalnum>

<filelist>

</filelist>

</response>

示例3：按时间检索响应，比较totalnum与endindex值可以判定录像文件响应是否发送完毕

<response>

<commandname>querylist</commandname>

<seq>3c7982da93db11e8b446060400ef53151</seq>

<ts>1533630770000</ts>

<mode>time</mode>

<totalnum>45</totalnum>

<beginindex>1</beginindex>

<endindex>3</endindex>

<filelist>

<fileproperty>

<url><http://mediaserver/hello></url>

<filename>hello</filename>

<recordtime>2014/05/04/12:00:00</recordtime>

<filensize>2154</filensize>

<fileduration>01:00:00</fileduration>

<filedescript>window safegurd</filedescript>

</fileproperty>

<fileproperty>

<url><http://mediaserver/helloworld></url>

<filename>helloworld</filename>

<recordtime>2014/05/04/12:00:00</recordtime>

<filensize>32154</filensize>

<fileduration>01:00:00</fileduration>

<filedescript>gate safegurd</filedescript>

</fileproperty>

<fileproperty>

<url><http://mediaserver/helloabc></url>

<filename>helloabc</filename>

<recordtime>2014/05/04/12:00:00</recordtime>

<filensize>32154</filensize>

<fileduration>01:00:00</fileduration>

<filedescript>exit safegurd</filedescript>

</fileproperty>

</filelist>

</response>

示例4：按时间检索的响应，beginindex=4

<response>

<commandname>querylist</commandname>

<seq>3c7982da93db11e8b446060400ef53151</seq>

<ts>1533630770000</ts>

<mode>time</mode>

<totalnum>45</totalnum>

<beginindex>4</beginindex>

<endindex>6</endindex>

<filelist>

<fileproperty>

<url><http://mediaserver/hello4></url>

<filename>hello4</filename>

<recordtime>2014/05/04/12:00:00</recordtime>

<filensize>32154</filensize>

<fileduration>01:00:00</fileduration>

<filedescript>window safegurd</filedescript>

</fileproperty>

<fileproperty>

<url><http://mediaserver/hello5></url>

<filename>hello5</filename>

<recordtime>2014/05/04/12:00:00</recordtime>

<filensize>32154</filensize>

<fileduration>01:00:00</fileduration>

<filedescript>gate safegurd</filedescript>

</fileproperty>

<fileproperty>

<url><http://mediaserver/hello6></url>

<filename>hello6</filename>

<recordtime>2014/05/04/12:00:00</recordtime>

<filensize>32154</filensize>

<fileduration>01:00:00</fileduration>

<filedescript>exit safegurd</filedescript>

</fileproperty>

</filelist>

</response> 示例5：按时间检索的响应，检索结果为空

<response>

<commandname>querylist</commandname>

<seq>3c7982da93db11e8b446060400ef53151</seq>

<ts>1533630770000</ts>

<mode>time</mode>

<totalnum>0</totalnum>

<filelist>

</filelist>

</response>

##### 媒体传输

* + 1. 媒体传输

###### 转发设置

* + - 1. 转发设置

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 转发设置管理操作，包括开始转发、停止转发-->

<xsd:simpleType name="operationType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="start"/>

<xsd:enumeration value="stop"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:complexType name="src\_type">

<xsd:sequence>

<xsd:element name="rtpid" type="xsd:string"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="dst\_type">

<xsd:sequence>

<xsd:element name="rtpid" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="control">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string"/>

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="operation" type="operationType"/>

<xsd:element name="max-dst" type="xsd:integer"/>

<xsd:element name="src" type="src\_type"/>

<xsd:element name="dst" type="dst\_type"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

示例1：开始转发

<control>

<commandname>switchset</commandname>

<seq>3c7982db93db11e8b446060400ef5315</seq>

<ts>1533630770000</ts>

<operation>start</operation>

<max-dst>3</max-dst>

<src>

<rtpid>1</rtpid>

</src>

<dst>

<rtpid>2</rtpid>

<rtpid>3</rtpid>

<rtpid>4</rtpid>

</dst>

</control>

**示例2：停止转发**

<control>

<commandname>switchset</commandname>

<seq>3c7982db93db11e8b446060400ef5315</seq>

<ts>1533630770000</ts>

<operation>stop</operation>

<max-dst>3</max-dst>

<src>

<rtpid>1</rtpid>

</src>

<dst>

<rtpid>2</rtpid>

<rtpid>3</rtpid>

<rtpid>4</rtpid>

</dst>

</control>

###### 传输通知

* + - 1. 传输通知

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!--媒体传输通知，establish表示媒体传输建立通知、cleanup表示媒体传输终止通知-->

<xsd:simpleType name="channelstatus\_type">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="establish"/>

<xsd:enumeration value="cleanup"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string"/>

<xsd:element name="seq" type="xsd:string"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="channelstatus" type="channelstatus\_type"/>

<xsd:element name="srcrtp" type="xsd:string"/>

<xsd:element name="dstrtp" type="xsd:string"/>

<xsd:element name="chanid" type="xsd:string"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

示例1：媒体传输建立通知

<notify>

<commandname>chanstatus</commandname>

<seq>660bf8d5946411e8b446060400ef5315</seq>

<ts>1533630770000</ts>

<channelstatus>establish</channelstatus>

<srcrtp>1</srcrtp>

<dstrtp>2</dstrtp>

<chanid>1</chanid>

</notify>

示例2：媒体传输终止通知

<notify>

<commandname>chanstatus</commandname>

<seq>660bf8d5946411e8b446060400ef53151</seq>

<ts>1533630770000</ts>

<channelstatus>cleanup</channelstatus>

<srcrtp>1</srcrtp>

<dstrtp>2</dstrtp>

<chanid>1</chanid>

</notify>

##### 媒体下载

* + 1. 媒体下载 媒体媒体下载命令属于应用服务单元至信令控制服务单元命令。 XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:include schemaLocation="common.xsd"/>

<!-- 媒体下载操作，start表示开始、stop表示停止 -->

<xsd:simpleType name="operation\_type">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="start"/>

<xsd:enumeration value="stop"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="audio\_type">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="g.711"/>

<xsd:enumeration value="aac-lc"/>

<xsd:enumeration value="g.729a"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="video\_type">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="h.264"/>

<xsd:enumeration value="h.265"/>

<xsd:enumeration value="javs"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:simpleType name="ring\_type">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="online"/>

<xsd:enumeration value="offline"/>

<xsd:enumeration value="ringing"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:complexType name="src\_type">

<xsd:sequence>

<xsd:element name="url" type="xsd:string" minOccurs="0"/>

<xsd:element name="userid" type="userIDType" minOccurs="0"/>

<xsd:element name="video" type="video\_type" default="h.264"/>

<xsd:element name="audio" type="audio\_type" default="g.711"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="control">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string"/>

<xsd:element name="operation" type="operation\_type"/>

<xsd:element name="seq" type="xsd:string"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="src" type="src\_type"/>

<xsd:element name="dst" type="src\_type"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例1：下载开始

<control>

<commandname>mediadownload</commandname>

<operation>start</operation>

<seq>660bf8d0946411e8b446060400ef5315</seq>

<ts>1533630770000</ts>

<src>

<url><http://mediaserver/file1></url>

<video>h.264</video>

<audio>g.711</audio>

</src>

<dst>

<userid>00000000002</userid>

<video>h.264</video>

<audio>g.711</audio>

</dst>

</control>

###### 示例2：下载停止

<control>

<commandname>mediadownload</commandname>

<operation>stop</operation>

<seq>660bf8d0946411e8b446060400ef5315</seq>

<ts>1533630770000</ts>

<src>

<url><http://mediaserver/file1></url>

<video>h.264</video>

<audio>g.711</audio>

</src>

<dst>

<userid>00000000002</userid>

<video>h.264</video>

<audio>g.711</audio>

</dst>

</control>

##### 文件尾通知

* + 1. 文件结尾通知

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string"/>

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="devid" type="deviceIDType"/>

<xsd:element name="filename" type="xsd:string"/>

<xsd:element name="error" type="xsd:string" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例：

<notify>

<commandname>filetoend</commandname>

<seq>3c7982db93db11e8b446060400ef5315</seq>

<ts>1533630770000</ts>

<filename>hello</filename>

<devid>00000000001</devid>

</notify>

#### A.5信息同步命令

A.5 信息同步命令

* + 1. 命令集

信息同步命令集如表A.6所示。

表 A.6 信息同步命令集

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 消息名称 | 消息标识 | 消息类型 | 承载方法类型 |
| 1 | 设备状态通知 | devicestatus | notify | MESSAGE |
| 2 | 链路状态通知 | linkstatus | notify | MESSAGE |
| 3 | 状态信息通知 | syncinfo | notify | NOTIFY PUBLISH |
| 4 | 路由信息通知 | syncroutelink | notify | NOTIFY PUBLISH |
| 5 | 授权信息通知 | authnotify | control | MESSAGE |
| 6 | 预置位请求 | pointindexoperation | request | MESSAGE |
| 7 | 预置位响应 | pointindexoperation | response | MESSAGE |
| 8 | 权限信息请求 | authreq | request | MESSAGE |
| 9 | 权限信息通知 | authres | response | MESSAGE |
| 10 | 业务组信息请求 | groupreq | request | MESSAGE |
| 11 | 业务组信息通知 | groupres | response | MESSAGE |

##### 设备状态通知

* + 1. 设备状态通知 设备状态通知用于信令控制服务单元向应用服务单元报告当前设备状态。 XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 设备状态，0-不在线，1-在线,2-会话中有媒体连接（仅编解码设备使用状态2） -->

<xsd:simpleType name="status\_type">

<xsd:restriction base="xsd:integer">

<xsd:enumeration value="0"/>

<xsd:enumeration value="1"/>

<xsd:enumeration value="2"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 设备类型（包括编解码设备和服务设备），2表示编码设备，3表示解码设备，6表示媒体处理 服务单元，11表示编解码一体设备-->

<!-- 信令控制服务单元和应用服务单元的状态在应用服务单元中已知，无需采用当前命令额外通 知-->

<xsd:simpleType name="devType">

<xsd:restriction base="xsd:integer">

<xsd:enumeration value="2"/>

<xsd:enumeration value="3"/>

<xsd:enumeration value="6"/>

<xsd:enumeration value="11"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string"/>

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="devid" type="deviceIDType" minOccurs="1"/>

<xsd:element name="devtype" type="devType" minOccurs="1"/>

<xsd:element name="status" type="status\_type" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例1：设备不在线

<notify>

<commandname>devicestatus</commandname>

<seq>44b4ba2c8e1b11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devid>00000000001</devid>

<devtype>2</devtype>

<status>0</status>

</notify>

###### 示例2：设备上线

<notify>

<commandname>devicestatus</commandname>

<seq>44b4ba2c8e1b11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devid>00000000001</devid>

<devtype>2</devtype>

<status>1</status>

</notify>

###### 示例3：设备处于会话中，有媒体连接

<notify>

<commandname>devicestatus</commandname>

<seq>44b4ba2c8e1b11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devid>00000000001</devid>

<devtype>2</devtype>

<status>2</status>

</notify>

###### 示例4：媒体处理服务单元设备上线

<notify>

<commandname>devicestatus</commandname>

<seq>44b4ba2c8e1b11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devid>00000000006</devid>

<devtype>6</devtype>

<status>1</status>

</notify>

##### 链路状态通知

* + 1. 链路状态通知

链路状态通知用于信令控制服务单元向应用服务单元报告当前信令控制服务单元与其他信令控制

服务单元之间链路状态的通知。 若A服务节点信令控制服务单元向B服务节点信令控制服务单元注册成功，表示A至B链路可达；若B

服务节点信令控制服务单元向A服务节点信令控制服务单元注册成功，表示B至A链路可达。当A至B可达， 且B至A可达时，认为A服务节点信令控制服务单元和B服务节点信令控制服务单元之间路由互通。

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 链路状态，0-不可达，1-可达-->

<xsd:simpleType name="status\_type">

<xsd:restriction base="xsd:integer">

<xsd:enumeration value="0"/>

<xsd:enumeration value="1"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string"/>

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!--若本节信令控制服务单元往其他服务节点信令控制服务单元注册成功，fromscid为本服务节点 信令控制服务单元编号，toscid为其他服务节点信令控制服务单元编号；若其他服务节点信令控制服务 单元往本服务节点信令控制服务单元注册成功，fromscid为其他服务节点信令控制服务单元，toscid为 本服务节点信令控制服务单元。 -->

<xsd:element name="fromscid" type="deviceIDType"/>

<xsd:element name="toscid" type="deviceIDType"/>

<xsd:element name="status" type="status\_type" minOccurs="0"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例1：链路可达

<notify>

<commandname>linkstatus</commandname>

<seq>44b4ba2c8e1b11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<fromscid>10000000001</fromscid>

<toscid>10000000002</toscid>

<status>1<status>

</notify>

###### 示例2：链路失效

<notify>

<commandname>linkstatus</commandname>

<seq>44b4ba2c8e1b11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<fromscid>10000000002</fromscid>

<toscid>10000000001</toscid>

<status>0<status>

</notify>

##### 状态信息同步

* + 1. 状态信息同步

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 状态，0表示不在线，1表示在线-->

<xsd:simpleType name="statusType">

<xsd:restriction base="xsd:integer">

<xsd:enumeration value="0"/>

<xsd:enumeration value="1"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 设备类型（包括编解码设备和服务设备），2表示编码设备，3表示解码设备，4表示应用服务 单元，6表示媒体处理服务单元，11表示编解码一体设备-->

<xsd:simpleType name="devType">

<xsd:restriction base="xsd:integer">

<xsd:enumeration value="2"/>

<xsd:enumeration value="3"/>

<xsd:enumeration value="4"/>

<xsd:enumeration value="6"/>

<xsd:enumeration value="11"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 用户信息类型，包含用户ID、状态（1--在线、0--不在线）、登录服务节点ID、绑定的设备 ID，用户可以不存在绑定的设备ID -->

<xsd:complexType name="useritemType">

<xsd:sequence>

<xsd:element name="userid" type="userIDType"/>

<xsd:element name="status" type="statusType"/>

<xsd:element name="visitednodeid" type="uuidType"/>

<xsd:element name="binddevid" type="deviceIDType" minOccurs="0" maxOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

<!-- 设备信息类型，包含设备ID、设备状态（1--在线、0--不在线）、设备类型（包括编解码设 备和服务设备）、登录服务节点ID -->

<xsd:complexType name="devitemType">

<xsd:sequence>

<xsd:element name="devid" type="deviceIDType"/>

<xsd:element name="status" type="statusType"/>

<xsd:element name="devtype" type="devType"/>

<xsd:element name="visitednodeid" type="uuidType"/>

</xsd:sequence>

</xsd:complexType>

<!-- 用户表类型，包括若干个用户信息-->

<xsd:complexType name="userlistType">

<xsd:sequence>

<xsd:element name="useritem" type="useritemType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<!-- 设备表类型，包括若干个设备信息-->

<xsd:complexType name="devlistType">

<xsd:sequence>

<xsd:element name="devitem" type="devitemType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="xsd:string" fixed="syncinfo"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!--userlist为用户列表，包括若干用户的状态信息-->

<xsd:element name="userlist" type="userlistType"/>

<!--devicelist为设备列表，包括若干设备的状态信息-->

<xsd:element name="devlist" type="devlistType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例：

<notify>

<commandname>syncinfo</commandname>

<seq>8cd41a008e4311e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<userlist>

<useritem>

<userid>00000000001</userid>

<status>1</status>

<visitednodeid>9e4dee468e4311e8b445060400ef5315</visitednodeid>

<binddevid>10000000001</binddevid>

</useritem>

<useritem>

<userid>00000000002</userid>

<status>1</status>

<visitednodeid>9e4dee468e4311e8b445060400ef5315</visitednodeid>

</useritem>

</userlist>

<devlist>

<devitem>

<devid>10000000001</devid>

<status>1</status>

<devtype>2</devtype>

<visitednodeid>9e4dee468e4311e8b445060400ef5315</visitednodeid>

</devitem>

<devitem>

<devid>10000000002</devid>

<status>1</status>

<devtype>3</devtype>

<visitednodeid>9e4dee468e4311e8b445060400ef5315</visitednodeid>

</devitem>

<devitem>

<devid>10000000006</devid>

<status>1</status>

<devtype>4</devtype>

<visitednodeid>9e4dee468e4311e8b445060400ef5315</visitednodeid>

</devitem>

<devitem>

<devid>10000000008</devid>

<status>1</status>

<devtype>6</devtype>

<visitednodeid>9e4dee468e4311e8b445060400ef5315</visitednodeid>

</devitem>

<devitem>

<devid>10000000009</devid>

<status>1</status>

<devtype>11</devtype>

<visitednodeid>9e4dee468e4311e8b445060400ef5315</visitednodeid>

</devitem>

</devlist>

</notify>

##### 路由信息同步

* + 1. 路由信息同步

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 全局连通矩阵从左到右对应的服务节点 -->

<xsd:complexType name="nodelistType">

<xsd:sequence>

<xsd:element name="nid" type="uuidType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<!-- 矩阵表述类型，raw表示原始格式矩阵表述，tight表示经过游程编码的矩阵表述 -->

<!-- 连通矩阵内容，矩阵中(i,j)元素若为1，则表明i,j两个服务节点可直接连通，若为0则不可 直接连通；-->

<!-- 若连通矩阵宽度较小，则建议按原始格式传输矩阵，方法为逐行扫描，跨行以“,”分隔；若 连通矩阵宽度较大，则建议按紧凑格式传输矩阵，方法为以除矩阵主对角线的上三角元素进行逐行扫描， 并以0的个数为起始，进行游程长度编码，将每个游程长度以“,”分隔

例： 有5个服务节点A~E，连通矩阵如下， A B C D E

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | 1 | 1 | 1 | 1 | 1 |
| B | 1 | 1 | 0 | 0 | 1 |
| C | 1 | 0 | 1 | 1 | 1 |
| D | 1 | 0 | 1 | 1 | 0 |
| E | 1 | 1 | 1 | 0 | 1 |

原始格式矩阵"11111,11001,10111,10110,11101"

紧凑格式矩阵，逐行扫描1111001110，游程编码后"0,4,2,3,1" -->

<xsd:simpleType name="matrixtype">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="raw"/>

<xsd:enumeration value="tight"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="xsd:string" fixed="syncroutelink"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="matsize" type="xsd:integer"/>

<xsd:element name="mattype" type="matrixType"/>

<xsd:element name="matcontent" type="xsd:string"/>

<xsd:element name="nlist" type="nodelistType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例1：原始矩阵表示的增量信息

<notify>

<commandname>syncroutelink</commandname>

<seq>8cd41a008e4311e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<matsize>4</matsize>

<mattype>raw</mattype>

<matcontent>11,11</matcontent>

<nlist>

<nid>15558693f2345f03bccf64ad29b2890d</nid>

<nid>bbf7b4a0533659cba4190f59352dd050</nid>

<nlist>

</notify>

###### 示例2：紧凑格式表示的全量信息

<notify>

<commandname>syncroutelink</commandname>

<seq>8cd41a008e4311e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<matsize>25</matsize>

<mattype>tight</mattype>

<matcontent>0,4,2,3,1</matcontent>

<nlist>

<nid>15558693f2345f03bccf64ad29b2890d</nid>

<nid>bbf7b4a0533659cba4190f59352dd050</nid>

<nid>5e73d2d1a2c6572cb99b3203f79e786e</nid>

<nid>3b516e7a32255712ba9d3b4105395638</nid>

<nid>08282ec6faca5ce29c8cc7c21b0cc7f9</nid>

<nlist>

</notify>

###### 示例3：原始格式表示的全量信息

<notify>

<commandname>syncroutelink</commandname>

<seq>8cd41a008e4311e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<matsize>25</matsize>

<mattype>raw</mattype>

<matcontent>11111,11001,10111,10110,11101</matcontent>

<nlist>

<nid>15558693f2345f03bccf64ad29b2890d</nid>

<nid>bbf7b4a0533659cba4190f59352dd050</nid>

<nid>5e73d2d1a2c6572cb99b3203f79e786e</nid>

<nid>3b516e7a32255712ba9d3b4105395638</nid>

<nid>08282ec6faca5ce29c8cc7c21b0cc7f9</nid>

<nlist>

</notify>

##### 授权信息通知

* + 1. 授权信息通知 授权信息同步为增量同步，授权操作成功后后应将授权信息同步至相关所有服务节点。 XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 授权操作类型，add表示新增、edit表示编辑、remove表示删除全部权限-->

<xsd:simpleType name="operation\_type">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="add"/>

<xsd:enumeration value="edit"/>

<xsd:enumeration value="remove"/>

</xsd:restriction>

</xsd:simpleType>

<!--功能、视频资源权限类型-->

<xsd:simpleType name="authType">

<xsd:restriction base="ID">

<xsd:pattern value="\d{16}"/>

</xsd:restriction>

</xsd:simpleType>

<!--用户可使用权限的其他用户信息-->

<xsd:complexType name="userauthitemType">

<xsd:sequence>

<!--用户可使用权限的用户id-->

<xsd:element name="userid" type="userIDType"/>

<!--用户包含点播、呼叫、ZH、会议、录制、回放、下载、控制等权限 -->

<!--用户权限16位，后8位为保留位，前八位从左到右分别代表点播、呼叫、ZH、会议、录制、回 放、下载、控制，每一位中1代表具备该权限，0代表不具备该权限-->

<xsd:element name="auth" type="authType"/>

</xsd:sequence>

</xsd:complexType>

<!--用户可使用权限的设备信息-->

<xsd:complexType name="devauthitemType">

<xsd:sequence>

<!--用户可使用权限的设备id-->

<xsd:element name="devid" type="deviceIDType"/>

<!--设备包含点播、录制、回放、下载、控制、画面调节等权限 -->

<!-- 设备权限16位，后10位为保留位，前六位从左到右分别代表点播、录制、回放、下载、控制、 画面调节，每一位中1代表具备该权限，0代表不具备该权限 -->

<xsd:element name="auth" type="authType"/>

</xsd:sequence>

</xsd:complexType>

<!--用户可使用权限的其他用户列表-->

<xsd:complexType name="userauthlistType">

<xsd:sequence>

<xsd:element name="useritem" type="userauthitemType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<!--用户可使用权限的设备列表-->

<xsd:complexType name="devauthlistType">

<xsd:sequence>

<xsd:element name="devitem" type="devauthitemType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="control">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="xsd:string" fixed="authnotify"/>

<!-- 操作类型-->

<xsd:element name="operation" type="operation\_type"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 授权操作服务节点-->

<xsd:element name="authnodeid" type="uuidType"/>

<!-- 授权操作用户ID-->

<xsd:element name="authuserid" type="userIDType"/>

<!-- 接受授权用户ID-->

<xsd:element name="userid" type="userIDType"/>

<!-- 用户操作权限列表-->

<xsd:element name="userlist" type="userauthlistType"/>

<!-- 设备操作权限列表-->

<xsd:element name="devlist" type="devauthlistType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例1：增加/编辑授权信息同步

<control>

<commandname>authnotify</commandname>

<operation>add<operation>

<seq>1fec5ca28e4011e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<authnodeid>2bfcc9328e4011e8b445060400ef5315</authnodeid>

<authuserid>00000000003</authuserid>

<userid>00000000001</userid>

<userlist>

<useritem>

<userid>00000000002</userid>

<auth>1111111100000000</auth>

</useritem>

</userlist>

<devlist>

<devitem>

<devid>12000000001</devid>

<auth>1111110000000000</auth>

</devitem>

</devlist>

</control>

###### 示例2：删除全部权限，删除权限时用户或设备列表不携带auth权限信息

<control>

<commandname>authnotify</commandname>

<operation>remove<operation>

<seq>1fec5ca28e4011e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<authnodeid>2bfcc9328e4011e8b445060400ef5315</authnodeid>

<authuserid>00000000003</authuserid>

<userid>00000000001</userid>

<userlist>

<useritem>

<userid>00000000002</userid>

</useritem>

</userlist>

<devlist>

<devitem>

<devid>12000000001</devid>

</devitem>

</devlist>

</control>

##### 预置位请求

* + 1. 预置位请求

若需跨服务节点调用设备预置位，应向设备的归属服务节点发送预置位请求，获取设备的预置位信

息。

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>"

elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:element name="request">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string" fixed="pointindexoperation"/>

<xsd:element name="operation" type="xsd:string" fiexed="query"/>

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="devname" type="xsd:string"/>

<xsd:element name="devid" type="deviceIDType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例：

<request>

<commandname>pointindexoperation</commandname>

<operation>query</operation>

<seq>31d0f0908e1811e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000003</devid>

</request>

##### 预置位响应

* + 1. 预置位响应 预置位响应用于对预置位请求进行应答。 XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!--预置位单项信息-->

<xsd:complexType name="pointindexitem\_type">

<xsd:sequence>

<!-- 预置位名称 -->

<xsd:element name="pointindexname" type="xsd:string"/>

<!-- 预置位索引值 -->

<xsd:element name="pointindexvalue" type="xsd:integer"/>

</xsd:sequence>

</xsd:complexType>

<!--预置位信息表-->

<xsd:complexType name="pointindexlist\_type">

<xsd:sequence>

<xsd:element name="pointindexitem" type="pointindexitem\_type" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="response">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string" fixed="pointindexoperation"/>

<xsd:element name="operation" type="xsd:string" fiexed="response"/>

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="devname" type="xsd:string"/>

<xsd:element name="devid" type="deviceIDType"/>

<xsd:element name="pointindexlist" type="pointindexlis\_type"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例：

<response>

<commandname>pointindexoperation</commandname>

<operation>response</operation>

<seq>e27738e08e1911e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<devname>member1</devname>

<devid>00000000003</devid>

<pointindexlist>

<pointindexitem>

<pointindexname>领导1</pointindexname>

<pointindexvalue>128</pointindexvalue >

</pointindexitem>

</pointindexlist>

</response>

##### 权限信息请求

* + 1. 权限信息请求 当用户处于漫游状态时，应采用权限信息请求从其归属服务节点获取权限信息。 XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:element name="request">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string" fixed="authreq"/>

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="userid" type="userIDType"/>

<!--用户归属服务节点id-->

<xsd:element name="homenodeid" type="uuidType"/>

<!--用户登录服务节点id-->

<xsd:element name="visitednodeid" type="uuidType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例：

<request>

<commandname>authreq</commandname>

<seq>47cbf6908e3d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<userid>00000000001</userid>

<homenodeid>5c4dcec28e3d11e8b445060400ef5315</homenodeid>

<visitednodeid>62dcd3468e3d11e8b445060400ef5315</visitednodeid>

</request>

##### 权限信息响应

* + 1. 权限信息响应 授权信息响应用于对授权信息请求进行应答。 XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!--功能、视频资源权限类型-->

<xsd:simpleType name="authType">

<xsd:restriction base="ID">

<xsd:pattern value="\d{16}"/>

</xsd:restriction>

</xsd:simpleType>

<!--用户可使用权限的其他用户信息-->

<xsd:complexType name="userauthitemType">

<xsd:sequence>

<!--用户可使用权限的用户id-->

<xsd:element name="userid" type="userIDType"/>

<!--用户包含点播、呼叫、ZH、会议、录制、回放、下载、控制等权限 -->

<!--用户权限16位，后8位为保留位，前八位从左到右分别代表点播、呼叫、ZH、会议、录制、回 放、下载、控制，每一位中1代表具备该权限，0代表不具备该权限-->

<xsd:element name="auth" type="authType"/>

</xsd:sequence>

</xsd:complexType>

<!--用户可使用权限的设备信息-->

<xsd:complexType name="devauthitemType">

<xsd:sequence>

<!--用户可使用权限的设备id-->

<xsd:element name="devid" type="deviceIDType"/>

<!--设备包含点播、录制、回放、下载、控制、画面调节等权限 -->

<!-- 设备权限16位，后10位为保留位，前六位从左到右分别代表点播、录制、回放、下载、控制、 画面调节，每一位中1代表具备该权限，0代表不具备该权限 -->

<xsd:element name="auth" type="authType"/>

</xsd:sequence>

</xsd:complexType>

<!--用户可使用权限的其他用户列表-->

<xsd:complexType name="userauthlistType">

<xsd:sequence>

<xsd:element name="useritem" type="userauthitemType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<!--用户可使用权限的设备列表-->

<xsd:complexType name="devauthlistType">

<xsd:sequence>

<xsd:element name="devitem" type="devauthitemType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="response">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string" fixed="authres"/>

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<xsd:element name="userid" type="userIDType"/>

<!--归属服务节点id-->

<xsd:element name="homenodeid" type="uuidType"/>

<!--登录服务节点id -->

<xsd:element name="visitednodeid" type="uuidType"/>

<!-- 用户操作权限列表-->

<xsd:element name="userlist" type="userauthlistType"/>

<!-- 设备操作权限列表-->

<xsd:element name="devlist" type="devauthlistType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例：

<response>

<commandname>authres</commandname>

<seq>47cbf6908e3d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<userid>000000000001</userid>

<homenodeid>5c4dcec28e3d11e8b445060400ef5315</homenodeid>

<visitednodeid>62dcd3468e3d11e8b445060400ef5315</visitednodeid>

<userlist>

<useritem>

<userid>00000000002</userid>

<auth>1111111100000000</auth>

</useritem>

</userlist>

<devlist>

<devitem>

<devid>10000000001</devid>

<auth>1111110000000000</auth>

</devitem>

</devlist>

</response>

##### 业务组信息请求

* + 1. 业务组信息请求 当用户处于漫游状态时，应采用组信息请求从其归属服务节点获取该用户的业务组信息。 XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:element name="request">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string" fixed="groupreq"/>

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 成员ID -->

<xsd:element name="mid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例：

<request>

<commandname>groupreq</commandname>

<seq>fd303e7c8e2611e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<mid>00000000002</mid>

</request>

##### 业务组信息响应

* + 1. 业务组信息响应 业务组信息响应用于对业务组信息请求进行应答。 XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 组业务类型，cnf表示会议组，cmd表示ZH组-->

<xsd:simpleType name="bizType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="cnf"/>

<xsd:enumeration value="cmd"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 成员类型，en表示编码设备，ende表示编解码设备,usr表示用户-->

<xsd:simpleType name="mType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="en"/>

<xsd:enumeration value="ende"/>

<xsd:enumeration value="usr"/>

</xsd:restriction>

</xsd:simpleType>

<!--成员信息-->

<xsd:complexType name="mitemType">

<xsd:sequence>

<xsd:element name="mid" type="userIDType"/>

<xsd:element name="mname" type="xsd:string"/>

<!-- 成员类型 -->

<xsd:element name="mtype" type="mType"/>

<!-- 上级成员ID -->

<xsd:element name="pid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

<!-- 成员列表-->

<xsd:complexType name="mlistType">

<xsd:sequence>

<xsd:element name="minfo" type="mitemType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<!-- 组信息 -->

<xsd:complexType name="bizinfoType">

<xsd:sequence>

<!-- 组业务类型 -->

<xsd:element name="biztype" type="bizType"/>

<!-- 组名称 -->

<xsd:element name="bizname" type="xsd:string"/>

<!-- 创建者ID -->

<xsd:element name="creatorid" type="userIDType"/>

<!-- 最上级成员ID -->

<xsd:element name="topid" type="userIDType"/>

<!-- 成员列表 -->

<xsd:element name="mlist" type="mlistType"/>

</xsd:sequence>

</xsd:complexType>

<!--组单项信息-->

<xsd:complexType name="groupitem\_type">

<xsd:sequence>

<!-- 组ID -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 主题 -->

<xsd:element name="subject" type="xsd:string"/>

<xsd:element name="bizinfo" type="bizinfoType"/>

</xsd:sequence>

</xsd:complexType>

<!--组信息表-->

<xsd:complexType name="grouplist\_type">

<xsd:sequence>

<xsd:element name="gitem" type="groupitem\_type" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="response">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string" fixed="groupres"/>

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 成员ID，可能为编码设备、编解码一体设备或用户 -->

<xsd:element name="mid" type="userIDType"/>

<!-- 该成员所在的业务组列表 -->

<xsd:element name="glist" type="grouplist\_type"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例：

<response>

<commandname>groupres</commandname>

<seq>fd303e7c8e2611e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<mid>00000000002</mid>

<glist>

<gitem>

<gid>123</gid>

<subject>XX项目第三次研讨会</subject>

<bizinfo>

<biztype>cnf</biztype>

<bizname>研发一组</bizname>

<creatorid>00000000001<creatorid>

<topid>00000000001</topid>

<mlist>

<minfo>

<mid>00000000002<mid>

<mname>张三</mname>

<mtype>usr</mtype>

<pid>00000000001</pid>

</minfo>

</mlist>

</bizinfo>

</gitem>

</glist>

</response>

#### A.6 业务控制命令

* + 1. 命令集

业务控制命令集如表A.7所示。

表 A.7 业务控制命令集

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 消息名称 | 消息标识 | 消息类型 | 承载方法类型 |
| 1 | 静态组信息同步 | group | control | MESSAGE |
| 2 | 启动分组业务 | bizcnf/bizcmd | notify | MESSAGE |
| 3 | 暂停分组业务 | notify | MESSAGE |
| 4 | 恢复分组业务 | notify | MESSAGE |
| 5 | 结束分组业务 | notify | MESSAGE |
| 6 | 成员加入通知 | notify | MESSAGE |
| 7 | 成员退出通知 | notify | MESSAGE |
| 8 | 成员退出请求 | request | MESSAGE |
| 9 | 成员退出响应 | response | MESSAGE |
| 10 | 成员状态更新 | notify | MESSAGE |
| 11 | 全量信息同步 | notify | MESSAGE |
| 12 | 媒体转发授权 | notify | MESSAGE |
| 13 | 媒体转发通知 | notify | MESSAGE |
| 14 | 讨论模式 | bizcnf | notify | MESSAGE |
| 15 | 发言通知 | notify | MESSAGE |
| 16 | 取消发言 | notify | MESSAGE |
| 17 | 申请发言请求 | request | MESSAGE |
| 18 | 申请发言响应 | response | MESSAGE |
| 19 | 专向ZH | bizcmd | notify | MESSAGE |
| 20 | 协同ZH | notify | MESSAGE |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 消息名称 | 消息标识 | 消息类型 | 承载方法类型 |
| 21 | 越级ZH |  | notify | MESSAGE |
| 22 | 授权ZH | notify | MESSAGE |
| 23 | 接替ZH | notify | MESSAGE |

##### 静态组信息同步

* + 1. 静态组信息同步

静态组信息同步命令用于在不同服务节点之间实现静态组的管理，包括创建、更新和删除。静态组 成员归属于不同服务节点时，应通过静态组信息同步命令将静态业务组信息同步至所有相关服务节点的 应用服务单元。当用户处于漫游状态时，对静态组的管理操作应同步至其归属服务节点应用服务单元。 在分组业务进行过程中，业务成员的加入与退出不应触发静态组的更新。

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 静态组管理操作，包括创建组、更新组、销毁组-->

<xsd:simpleType name="operationType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="create"/>

<xsd:enumeration value="update"/>

<xsd:enumeration value="destroy"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 组业务类型，cnf表示会议组，cmd表示ZH组-->

<xsd:simpleType name="bizType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="cnf"/>

<xsd:enumeration value="cmd"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 成员类型，en表示编码设备，ende表示编解码设备,usr表示用户-->

<xsd:simpleType name="mType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="en"/>

<xsd:enumeration value="ende"/>

<xsd:enumeration value="usr"/>

</xsd:restriction>

</xsd:simpleType>

<!--成员信息-->

<xsd:complexType name="mitemType">

<xsd:sequence>

<xsd:element name="mid" type="userIDType"/>

<xsd:element name="mname" type="xsd:string"/>

<!-- 成员类型 -->

<xsd:element name="mtype" type="mType"/>

<!-- 上级成员ID；会议组成员的上级成员ID为主席；ZH组成员的上级成员为其直接上级；主席 和最高级ZH员无上级不应显示此标签 -->

<xsd:element name="pid" type="userIDType" minOccurs="0" maxOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

<!-- 成员列表，应包含当前组所有成员 -->

<xsd:complexType name="mlistType">

<xsd:sequence>

<xsd:element name="minfo" type="mitemType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<!-- 组信息 -->

<xsd:complexType name="bizinfoType">

<xsd:sequence>

<!-- 组业务类型 -->

<xsd:element name="biztype" type="bizType"/>

<!-- 组名称 -->

<xsd:element name="bizname" type="xsd:string"/>

<!-- 创建者ID -->

<xsd:element name="creatorid" type="userIDType"/>

<!-- 最上级成员ID -->

<xsd:element name="topid" type="userIDType"/>

<!-- 成员列表 -->

<xsd:element name="mlist" type="mlistType"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="control">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="commandname" type="xsd:string" fixed="group"/>

<xsd:element name="operation" type="operationType"/>

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 组ID，组ID不允许重复，应唯一 -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 主题，创建和更新组时携带，销毁组时不携带 -->

<xsd:element name="subject" type="xsd:string" minOccurs="0" maxOccurs="1"/>

<!-- 组信息，创建和更新组时携带，销毁组时不携带 -->

<xsd:element name="bizinfo" type="bizinfoType" maxOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例1：创建业务组

<control>

<commandname>group</commandname>

<operation>create</operation>

<seq>1882caf88e2411e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<subject>XX项目第三次研讨会</subject>

<bizinfo>

<!-- cnf或cmd-->

<biztype>cnf</biztype>

<bizname>研发一组</bizname>

<creatorid>00000000001<creatorid>

<topid>00000000001</topid>

<mlist>

<!-- 主席-->

<mname>li</mname>

<mtype>usr</mtype>

</minfo>

<minfo>

<mid>00000000002<mid>

<mname>张三</mname>

<!-- en或ende或usr-->

<mtype>usr</mtype>

<pid>00000000001</pid>

</minfo>

</mlist>

</bizinfo>

</control>

###### 示例2：更新业务组

<control>

<commandname>group</commandname>

<operation>update</operation>

<seq>1882caf88e2411e8b445060400ef1253</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<subject>XX项目第三次研讨会</subject>

<bizinfo>

<!-- cnf或cmd-->

<biztype>cnf</biztype>

<bizname>研发一组</bizname>

<creatorid>00000000001<creatorid>

<topid>00000000001</topid>

<mlist>

<!-- 主席-->

<minfo>

<mid>00000000001<mid>

<mname>li</mname>

<mtype>usr</mtype>

</minfo>

<mname>张三</mname>

<!-- en或ende或usr-->

<mtype>usr</mtype>

<pid>00000000001</pid>

</minfo>

<minfo>

<mid>00000000003<mid>

<mname>李四</mname>

<!-- en或ende或usr-->

<mtype>usr</mtype>

<pid>00000000001</pid>

</minfo>

</mlist>

</bizinfo>

</control>

###### 示例3：销毁业务组

<control>

<commandname>group</commandname>

<operation>destroy</operation>

<seq>1882caf88e2411e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

</control>

* + 1. 分组业务基础控制

##### 启动分组业务

* + - 1. 启动分组业务

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 命令名称类型，bizcnf表示会议组业务命令，bizcmd表示ZH组业务命令 -->

<xsd:simpleType name="commandnameType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="bizcnf"/>

<xsd:enumeration value="bizcmd"/>

</xsd:restriction>

<!-- 组业务操作类型，包括启动、暂停、恢复、结束、成员加入通知、成员退出通知、成员退出请 求、成员退出响应、成员状态更新、全量信息同步、媒体转发授权/取消、媒体转发通知/取消、开启/停 止讨论模式、发言通知、取消发言、申请发言请求、申请发言响应、开始/结束专向ZH、开始/结束协 同ZH、开始/结束越级ZH、开始/结束授权ZH、开始/结束接替ZH-->

<xsd:simpleType name="operationType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="start"/>

<xsd:enumeration value="pause"/>

<xsd:enumeration value="resume"/>

<xsd:enumeration value="stop"/>

<xsd:enumeration value="maddinc"/>

<xsd:enumeration value="mquit"/>

<xsd:enumeration value="mquitreq"/>

<xsd:enumeration value="mquitres"/>

<xsd:enumeration value="mstatus"/>

<xsd:enumeration value="maddfull"/>

<xsd:enumeration value="fauthon"/>

<xsd:enumeration value="fauthoff"/>

<xsd:enumerationvalue="pullmediastart"/>

<xsd:enumeration value="pullmediastop"/>

<xsd:enumeration value="discstart"/>

<xsd:enumeration value="discstop"/>

<xsd:enumeration value="spkset"/>

<xsd:enumeration value="spkcal"/>

<xsd:enumeration value="spkreq"/>

<xsd:enumeration value="spkres"/>

<xsd:enumeration value="secretstart"/>

<xsd:enumeration value="secretstop"/>

<xsd:enumeration value="corpstart"/>

<xsd:enumeration value="corpstop"/>

<xsd:enumeration value="crossstart"/>

<xsd:enumeration value="crossstop"/>

<xsd:enumeration value="authstart"/>

<xsd:enumeration value="authstop"/>

<xsd:enumeration value="replacestart"/>

<xsd:enumeration value="replacestop"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 组业务类型，cnf表示会议组，cmd表示ZH组-->

<xsd:simpleType name="bizType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="cnf"/>

<xsd:enumeration value="cmd"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 业务的组类型，temp表示临时组，preset表示静态组-->

<xsd:simpleType name="groupType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="temp"/>

<xsd:enumeration value="preset"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 成员类型，en表示编码设备，ende表示编解码设备，usr表示用户-->

<xsd:simpleType name="mType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="en"/>

<xsd:enumeration value="ende"/>

<xsd:enumeration value="usr"/>

</xsd:restriction>

</xsd:simpleType>

<!--成员信息-->

<xsd:complexType name="mitemType">

<xsd:sequence>

<xsd:element name="mid" type="userIDType"/>

<xsd:element name="mname" type="xsd:string"/>

<!-- 成员类型 -->

<xsd:element name="mtype" type="mType"/>

<!-- 上级成员ID -->

<xsd:element name="pid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

<!-- 成员列表-->

<xsd:complexType name="mlistType">

<xsd:sequence>

<xsd:element name="minfo" type="mitemType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<!-- 组信息 -->

<xsd:complexType name="bizinfoType">

<xsd:sequence>

<!-- 组业务类型 -->

<xsd:element name="biztype" type="bizType"/>

<!-- 组名称 -->

<xsd:element name="bizname" type="xsd:string"/>

<!-- 创建者ID -->

<xsd:element name="creatorid" type="userIDType"/>

<!-- 最上级成员ID -->

<xsd:element name="topid" type="userIDType"/>

<!-- 成员列表 -->

<xsd:element name="mlist" type="mlistType"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="commandnameType"/>

<!-- 操作名称，启动分组业务取值为start -->

<xsd:element name="operation" type="operationType"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 业务组ID，若业务组属于静态组（业务组类型取值为preset，静态组操作见组信息同步）， 业务组ID等于静态组ID；若业务组为临时业务组（业务组类型取值为temp），组ID需动态生成且不能与 静态组ID重复 -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 本操作发起成员ID-->

<xsd:element name="op" type="userIDType"/>

<!-- 业务组类型，temp表示临时业务，preset表示基于静态组的业务-->

<xsd:element name="grouptype" type="groupType"/>

<!-- 业务主题 -->

<xsd:element name="subject" type="xsd:string"/>

<!-- 业务开始时间，格式为YYYY-MM-DD HH:mm:SS -->

<xsd:element name="stime" type="xsd:string"/>

<!-- bizinfo为业务信息 -->

<!-- 若业务组属于静态组，bizinfo信息可以不携带-->

<xsd:element name="bizinfo" type="bizinfoType" minOccurs="0" maxOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例1：基于静态组开始会议业务

<notify>

<commandname>bizcnf</commandname>

<operation>start</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<grouptype>preset</grouptype>

<subject>XX项目第三次研讨会</subject>

<stime>2018-08-10 08:00:00<stime>

</notify>

###### 示例2：基于临时组开始会议业务

<notify>

<commandname>bizcnf</commandname>

<operation>start</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>1234567</gid>

<op>00000000001</op>

<grouptype>temp</grouptype>

<subject>XX项目第三次研讨会</subject>

<stime>2018-08-10 08:00:00<stime>

<bizinfo>

<biztype>cnf</biztype>

<bizname>研发一组</bizname>

<creatorid>00000000001<creatorid>

<topid>00000000001</topid>

<mlist>

<minfo>

<mid>00000000002<mid>

<mname>张三</mname>

<mtype>usr</mtype>

<pid>00000000001</pid>

</minfo>

</mlist>

</bizinfo>

</notify>

###### 示例3：基于静态组开始ZH业务

<notify>

<commandname>bizcmd</commandname>

<operation>start</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123123</gid>

<op>00000000001</op>

<grouptype>preset</grouptype>

<subject>XX专项XX</subject>

<stime>2018-08-10 08:00:00<stime>

</notify>

##### 暂停业务

* + - 1. 暂停分组业务

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 命令名称类型，bizcnf表示会议组业务命令，bizcmd表示ZH组业务命令-->

<xsd:simpleType name="commandnameType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="bizcnf"/>

<xsd:enumeration value="bizcmd"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="commandnameType"/>

<!-- 操作名称，类型定义见A.6.3.1启动分组业务，暂停分组业务取值为pause -->

<xsd:element name="operation" type="operationType"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 业务组ID -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 本操作发起成员ID-->

<xsd:element name="op" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例：业务暂停

<notify>

<commandname>bizcmd</commandname>

<operation>pause</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123123</gid>

<op>00000000001</op>

</notify>

##### 恢复分组业务

* + - 1. 恢复分组业务

XML结构描述与[A.6.3.2 暂停分组业务](#_bookmark4)相同，消息体中operation标签取值为resume。

示例：业务恢复至上一个状态

<notify>

<commandname>bizcmd</commandname>

<operation>resume</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123123</gid>

<op>00000000001</op>

</notify>

##### 结束分组业务

* + - 1. 结束分组业务

XML结构描述与[A.6.3.2 暂停分组业务](#_bookmark4)相同，消息体中operation标签取值为stop。

示例：停止业务

<notify>

<commandname>bizcmd</commandname>

<operation>stop</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123123</gid>

<op>00000000001</op>

</notify>

##### 成员加入通知

* + - 1. 成员加入通知 成员加入通知用于通知相关服务节点有成员加入业务。 XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 命令名称类型，bizcnf表示会议组业务命令，bizcmd表示ZH组业务命令-->

<xsd:simpleType name="commandnameType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="bizcnf"/>

<xsd:enumeration value="bizcmd"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 成员类型，en表示编码设备，ende表示编解码设备,usr表示用户-->

<xsd:simpleType name="mType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="en"/>

<xsd:enumeration value="ende"/>

<xsd:enumeration value="usr"/>

</xsd:restriction>

</xsd:simpleType>

<!--成员信息-->

<xsd:complexType name="mitemType">

<xsd:sequence>

<xsd:element name="mid" type="userIDType"/>

<xsd:element name="mname" type="xsd:string"/>

<!-- 成员类型 -->

<xsd:element name="mtype" type="mType"/>

<!-- 上级成员ID -->

<xsd:element name="pid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

<!-- 成员列表-->

<xsd:complexType name="mlistType">

<xsd:sequence>

<xsd:element name="minfo" type="mitemType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="commandnameType"/>

<!-- 操作名称，类型定义见A.6.3.1启动分组业务，成员加入通知取值为maddinc -->

<xsd:element name="operation" type="operationType"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 业务组ID -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 本操作发起成员ID-->

<xsd:element name="op" type="userIDType"/>

<!-- 成员列表-->

<xsd:element name="mlist" type="mlistType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例：成员加入通知

<notify>

<commandname>bizcnf</commandname>

<operation>maddinc</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<mlist>

<minfo>

<mid>00000000003<mid>

<mname>李四</mname>

<mtype>usr</mtype>

<pid>00000000001</pid>

</minfo>

</mlist>

</notify >

##### 成员退出通知

* + - 1. 成员退出通知

成员退出通知用于通知相关服务节点有成员退出业务，可由会议主席或一级使用人员强退成员或由成

员主动退出。

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 命令名称类型，bizcnf表示会议组业务命令，bizcmd表示ZH组业务命令-->

<xsd:simpleType name="commandnameType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="bizcnf"/>

<xsd:enumeration value="bizcmd"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 成员退出类型，p表示成员被强制退出，r表示成员主动退出-->

<xsd:simpleType name="quitType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="p"/>

<xsd:enumeration value="r"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="commandnameType"/>

<!-- 操作名称，类型定义见A.6.3.1启动分组业务，成员退出通知取值为mquit -->

<xsd:element name="operation" type="operationType"/>

<!-- 成员退出类型-->

<xsd:element name="quittype" type="quitType"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 业务组ID -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 本操作发起成员ID-->

<xsd:element name="op" type="userIDType"/>

<!-- 本操作相关成员ID，退出成员ID-->

<xsd:element name="mid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例1：成员被强制退出

<notify>

<commandname>bizcnf</commandname>

<operation>mquit</operation>

<quittype>p</quittype>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<mid>00000000002</mid>

</notify>

###### 示例2：成员主动退出

<notify>

<commandname>bizcmd</commandname>

<operation>mquit</operation>

<quittype>r</quittype>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<mid>00000000002</mid>

</notify>

##### 成员退出请求

* + - 1. 成员退出请求 成员退出请求用于业务成员向会议主席或一级使用人员申请退出业务。 XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 命令名称类型，bizcnf表示会议组业务命令，bizcmd表示ZH组业务命令-->

<xsd:simpleType name="commandnameType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="bizcnf"/>

<xsd:enumeration value="bizcmd"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="request">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="commandnameType"/>

<!-- 操作名称，类型定义见A.6.3.1启动分组业务，成员退出请求取值为mquitreq -->

<xsd:element name="operation" type="operationType"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 业务组ID -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 本操作发起成员ID-->

<xsd:element name="op" type="userIDType"/>

<!-- 本操作相关成员ID-->

<xsd:element name="mid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例：成员退出请求

<request>

<commandname>bizcnf </commandname>

<operation>mquitreq</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<!-- 退出成员ID -->

<op>00000000002</op>

<!-- 可处理成员退出请求的成员ID -->

<mid>00000000001</mid>

</request>

##### 成员退出响应

* + - 1. 成员退出响应

成员退出响应用于对成员退出请求进行应答。当接收到成员退出请求后，应发送成员退出响应消息，

同时，若同意成员退出业务，会议主席或一级使用人员所在服务节点应向本业务所有相关的服务节点发送 成员退出通知，成员退出通知见A.6.3.7，成员退出类型为主动退出。

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 命令名称类型，bizcnf表示会议组业务命令，bizcmd表示ZH组业务命令-->

<xsd:simpleType name="commandnameType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="bizcnf"/>

<xsd:enumeration value="bizcmd"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 响应，0表示不同意、1表示同意-->

<xsd:simpleType name="codeType">

<xsd:restriction base="xsd:integer">

<xsd:enumeration value="0"/>

<xsd:enumeration value="1"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="response">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="commandnameType"/>

<!-- 操作名称，类型定义见A.6.3.1启动分组业务，成员退出响应取值为mquitres -->

<xsd:element name="operation" type="operationType"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 业务组ID -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 本操作发起成员ID-->

<xsd:element name="op" type="userIDType"/>

<!-- 本操作相关成员ID-->

<xsd:element name="mid" type="userIDType"/>

<xsd:element name="code" type="codeType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例：成员退出响应

<response>

<commandname>bizcnf</commandname>

<operation>mquitres</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<!-- 处理成员退出请求处理的成员ID -->

<op>00000000001</op>

<!-- 退出成员ID -->

<mid>00000000002</mid>

<code>1</code>

</response>

##### 成员状态更新

* + - 1. 成员状态更新

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:simpleType name="commandnameType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="bizcnf"/>

<xsd:enumeration value="bizcmd"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 状态，1表示正在业务、2表示暂时离开-->

<xsd:simpleType name="statusType">

<xsd:restriction base="xsd:integer">

<xsd:enumeration value="1"/>

<xsd:enumeration value="2"/>

</xsd:restriction>

</xsd:simpleType>

<!--成员信息-->

<xsd:complexType name="mitemType">

<xsd:sequence>

<xsd:element name="mid" type="userIDType"/>

<xsd:element name="mstatus" type="statusType"/>

</xsd:sequence>

</xsd:complexType>

<!-- 成员列表-->

<xsd:complexType name="mlistType">

<xsd:sequence>

<xsd:element name="minfo" type="mitemType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="commandnameType"/>

<!-- 操作名称，类型定义见A.6.3.1启动分组业务，成员状态更新取值为mstatus -->

<xsd:element name="operation" type="operationType"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 业务组ID -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 本操作发起成员ID-->

<xsd:element name="op" type="userIDType"/>

<!-- 本操作相关成员列表-->

<xsd:element name="mlist" type="mlistType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例：成员状态更新

<notify>

<commandname>bizcmd</commandname>

<operation>mstatus</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<mlist>

<minfo>

<mid>00000000002</mid>

<mstatus>2</mstatus>

</minfo>

</mlist>

</notify >

##### 全量信息同步

* + - 1. 全量信息同步

全量信息同步命令用于向参与业务的服务节点同步当前业务的全部状态。当有新成员加入且该成员

所在的服务节点未参与业务，或参与业务的服务节点发生异常并恢复时，则会议主席或一级使用人员所在 的服务节点应采用全量信息同步命令向新加入的服务节点同步业务的当前状态。

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 命令名称类型，bizcnf表示会议组业务命令，bizcmd表示ZH组业务命令-->

<xsd:simpleType name="commandnameType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="bizcnf"/>

<xsd:enumeration value="bizcmd"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 组业务类型，cnf表示会议组，cmd表示ZH组-->

<xsd:simpleType name="bizType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="cnf"/>

<xsd:enumeration value="cmd"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 业务的组类型，temp表示临时组，preset表示静态组-->

<xsd:simpleType name="groupType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="temp"/>

<xsd:enumeration value="preset"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 成员类型，en表示编码设备，ende表示编解码设备,usr表示用户-->

<xsd:simpleType name="mType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="en"/>

<xsd:enumeration value="ende"/>

<xsd:enumeration value="usr"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 成员状态，1表示正在业务、2表示暂时离开、3表示已退出-->

<xsd:simpleType name="statusType">

<xsd:restriction base="xsd:integer">

<xsd:enumeration value="1"/>

<xsd:enumeration value="2"/>

<xsd:enumeration value="3"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 会议业务模式状态，0表示主席模式、1表示讨论模式-->

<xsd:simpleType name="cnfstatuscodeType">

<xsd:restriction base="xsd:integer">

<xsd:enumeration value="0"/>

<xsd:enumeration value="1"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 业务状态，0表示正常业务、1表示暂停-->

<xsd:simpleType name="statuscodeType">

<xsd:restriction base="xsd:integer">

<xsd:enumeration value="0"/>

<xsd:enumeration value="1"/>

</xsd:restriction>

</xsd:simpleType>

<!--成员信息-->

<xsd:complexType name="mitemType">

<xsd:sequence>

<xsd:element name="mid" type="userIDType"/>

<xsd:element name="mname" type="xsd:string"/>

<!-- 成员类型 -->

<xsd:element name="mtype" type="mType"/>

<!-- 成员状态 -->

<xsd:element name="mstatus" type="statusType"/>

<!-- 上级成员ID -->

<xsd:element name="pid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

<!-- 成员列表-->

<xsd:complexType name="mlistType">

<xsd:sequence>

<xsd:element name="minfo" type="mitemType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<!-- 组信息 -->

<xsd:complexType name="bizinfoType">

<xsd:sequence>

<!-- 组类型 -->

<xsd:element name="biztype" type="biztype"/>

<!-- 组名称 -->

<xsd:element name="bizname" type="xsd:string"/>

<!-- 创建者ID -->

<xsd:element name="creatorid" type="userIDType"/>

<!-- 最上级成员ID -->

<xsd:element name="topid" type="userIDType"/>

<!-- 成员列表，应包含所有成员（包括新成员）信息，若是ZH业务，成员关系应是授权或接替 后的新ZH上下级关系-->

<xsd:element name="mlist" type="mlistType"/>

</xsd:sequence>

</xsd:complexType>

<!-- 会议业务状态信息 -->

<xsd:complexType name="cnfstatusType">

<xsd:sequence>

<!-- 会议模式，0表示主席模式、1表示讨论模式 -->

<xsd:element name="mode" type="cnfstatuscodeType"/>

<!-- 会议状态，0表示正常业务、1表示暂停 -->

<xsd:element name="status" type="statuscodeType"/>

<!-- 若有成员正在发言，携带正在发言者ID -->

<xsd:element name="spkid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

<!-- 授权ZH状态信息 -->

<xsd:complexType name="authitemType">

<xsd:sequence>

<xsd:element name="op" type="userIDType"/>

<xsd:element name="accepauthid" type="userIDType"/>

<xsd:element name="cmdedid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

<!-- 接替ZH状态信息 -->

<xsd:complexType name="replaceitemType">

<xsd:sequence>

<xsd:element name="op" type="userIDType"/>

<xsd:element name="targid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

<!-- 专向ZH列表 -->

<xsd:complexType name="secretlistType">

<xsd:sequence>

<xsd:element name="secretitem" type="secretitemType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="secretitemType">

<xsd:sequence>

<xsd:element name="upid" type="userIDType"/>

<xsd:element name="downid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

<!-- 协同ZH列表 -->

<xsd:complexType name="croplistType">

<xsd:sequence>

<xsd:element name="cropitem" type="cropitemType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="cropitemType">

<xsd:sequence>

<xsd:element name="mid" type="userIDType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<!-- 越级ZH列表 -->

<xsd:complexType name="crosslistType">

<xsd:sequence>

<xsd:element name="crossitem" type="crossitemType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name="crossitemType">

<xsd:sequence>

<xsd:element name="upid" type="userIDType"/>

<xsd:element name="downid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

<!-- ZH业务状态信息 -->

<xsd:complexType name="cmdstatusType">

<xsd:sequence>

<!-- ZH状态，0表示正常业务、1表示暂停业务 -->

<xsd:element name="status" type="statuscodeType"/>

<!-- 授权ZH状态项，与接替ZH状态项互斥-->

<xsd:element name="authitem" type="authitemType"/>

<!-- 接替ZH状态项，与授权ZH状态项互斥-->

<xsd:element name="replaceitem" type="replaceitemType"/>

<!-- 专向ZH列表-->

<xsd:element name="secretlist" type="secretlistType"/>

<!-- 协同ZH列表-->

<xsd:element name="croplist" type="croplistType"/>

<!-- 越级ZH列表-->

<xsd:element name="croslist" type="crosslistType"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="commandnameType"/>

<!-- 操作名称，类型定义见A.6.3.1启动分组业务，全量信息同步取值为maddfull -->

<xsd:element name="operation" type="operationType"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 业务组ID -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 本操作发起成员ID-->

<xsd:element name="op" type="userIDType"/>

<!-- 业务组类型，temp表示临时业务，preset表示基于静态组的业务-->

<xsd:element name="grouptype" type="groupType"/>

<!--业务主题 -->

<xsd:element name="subject" type="xsd:string"/>

<!--业务开始时间，格式为YYYY-MM-DD HH:mm:SS -->

<xsd:element name="stime" type="xsd:string"/>

<!-- bizinfo为业务信息 -->

<!-- 应携带bizinfo信息，应包含所有成员（包括新成员）信息，若是ZH业务，成员关系应是授 权或接替后的新ZH上下级关系-->

<xsd:element name="bizinfo" type="bizinfoType" minOccurs="1" maxOccurs="1"/>

<!-- 若是会议业务，应携带会议状态信息 -->

<xsd:element name="cnfstatus" type="cnfstatusType" minOccurs="0" maxOccurs="1"/>

<!-- 若是ZH业务，应携带ZH状态信息 -->

<xsd:element name="cmdstatus" type="cmdstatusType" minOccurs="0" maxOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例1：会议业务成员加入

<notify>

<commandname>bizcnf</commandname>

<operation>maddfull</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<grouptype>preset</grouptype>

<subject>XX项目第三次研讨会</subject>

<stime>2018-08-10 08:00:00<stime>

<bizinfo>

<biztype>cnf</biztype>

<bizname>研发一组</bizname>

<creatorid>00000000001<creatorid>

<topid>00000000001</topid>

<mlist>

<minfo>

<mid>00000000001<mid>

<mname>li</mname>

<mtype>usr</mtype>

<mstatus>1</mstatus>

</minfo>

<minfo>

<mid>00000000002<mid>

<mname>张三</mname>

<mtype>usr</mtype>

<mstatus>1</mstatus>

<pid>00000000001</pid>

</minfo>

<minfo>

<mid>00000000003<mid>

<mname>李四</mname>

<mtype>usr</mtype>

<mstatus>1</mstatus>

<pid>00000000001</pid>

</minfo>

</mlist>

</bizinfo>

<cnfstatus>

<mode>0</mode>

<status>0</status>

<spkid>00000000002</spkid>

</cnfstatus>

</notify>

###### 示例2：临时会议业务成员加入

<notify>

<commandname>bizcnf</commandname>

<operation>maddfull</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>1234567</gid>

<op>00000000001</op>

<grouptype>temp</grouptype>

<subject>XX项目第三次研讨会</subject>

<stime>2018-08-10 08:00:00<stime>

<bizinfo>

<biztype>cnf</biztype>

<bizname>研发一组</bizname>

<creatorid>00000000001<creatorid>

<topid>00000000001</topid>

<mlist>

<minfo>

<mid>00000000001<mid>

<mname>li</mname>

<mtype>usr</mtype>

<mstatus>1</mstatus>

</minfo>

<minfo>

<mid>00000000002<mid>

<mname>张三</mname>

<mtype>usr</mtype>

<mstatus>1</mstatus>

<pid>00000000001</pid>

</minfo>

<minfo>

<mid>00000000003<mid>

<mname>李四</mname>

<mtype>usr</mtype>

<mstatus>1</mstatus>

<pid>00000000001</pid>

</minfo>

</mlist>

</bizinfo>

<cnfstatus>

<mode>0</mode>

<status>1</status>

</cnfstatus>

</notify>

###### 示例3：ZH业务成员加入

<notify>

<commandname>bizcmd</commandname>

<operation>maddfull</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>124</gid>

<op>00000000001</op>

<grouptype>preset</grouptype>

<subject>XXXX</subject>

<stime>2018-08-10 08:00:00<stime>

<bizinfo>

<biztype>cmd</biztype>

<bizname>某单位</bizname>

<creatorid>00000000001<creatorid>

<topid>00000000001</topid>

<mlist>

<minfo>

<mid>00000000001<mid>

<mname>li</mname>

<mtype>usr</mtype>

<mstatus>1</mstatus>

</minfo>

<minfo>

<mid>00000000002<mid>

<mname>张三</mname>

<mtype>usr</mtype>

<mstatus>1</mstatus>

<pid>00000000001</pid>

</minfo>

<minfo>

<mid>00000000003<mid>

<mname>李四</mname>

<mtype>usr</mtype>

<mstatus>1</mstatus>

<pid>00000000002</pid>

</minfo>

<minfo>

<mid>00000000004<mid>

<mname>王五</mname>

<mtype>usr</mtype>

<mstatus>1</mstatus>

<pid>00000000002</pid>

</minfo>

</mlist>

</bizinfo>

<cmdstatus>

<status>0</status>

<crosslist>

<crossitem>

<upid>00000000001</upid>

<downid>00000000003</downid>

</crossitem>

</crosslist>

</cmdstatus>

</notify>

##### 媒体转发授权

* + - 1. 媒体转发授权

媒体转发授权用于向业务成员授予或取消授予转发视音频的权限，被授权成员应向目标成员发送媒

体转发通知，让目标成员主动“拉”取转发的视音频，媒体转发通知见A.6.3.12。 XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 命令名称类型，bizcnf表示会议组业务命令，bizcmd表示ZH组业务命令-->

<xsd:simpleType name="commandnameType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="bizcnf"/>

<xsd:enumeration value="bizcmd"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="commandnameType"/>

<!-- 操作名称，类型定义见A.6.3.1启动分组业务，媒体转发授权取值为fauthon，取消媒体转发 授权取值为fauthoff -->

<xsd:element name="operation" type="operationType"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 业务组ID -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 本操作发起成员ID-->

<xsd:element name="op" type="userIDType"/>

<!-- 本操作相关成员ID-->

<xsd:element name="mid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例1：转发授权

<notify>

<commandname>bizcnf</commandname>

<operation>fauthon</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<mid>00000000002</mid>

</notify>

###### 示例2：取消转发授权

<notify>

<commandname>bizcnf</commandname>

<operation>fauthoff</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<mid>00000000002</mid>

</notify>

##### 媒体转发通知

* + - 1. 媒体转发通知

媒体转发通知用于向业务成员转发或取消转发视音频，业务成员在接收到媒体转发通知后应主动

“拉”取被转发的视音频。

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 命令名称类型，bizcnf表示会议组业务命令，bizcmd表示ZH组业务命令-->

<xsd:simpleType name="commandnameType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="bizcnf"/>

<xsd:enumeration value="bizcmd"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 成员列表 -->

<xsd:complexType name="mlistType">

<xsd:sequence>

<xsd:element name="mid" type="userIDType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<!-- 媒体源（用户或设备编号）列表 -->

<xsd:complexType name="medialistType">

<xsd:sequence>

<xsd:element name="mediasrcid" type="userIDType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="commandnameType"/>

<!-- 操作名称，类型定义见A.6.3.1启动分组业务，开启媒体转发通知取值为pullmediastart，停 止媒体转发通知取值为pullmediastop -->

<xsd:element name="operation" type="operationType"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 业务组ID -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 本操作发起成员ID-->

<xsd:element name="op" type="userIDType"/>

<!-- 本操作相关成员ID，媒体接收成员ID列表-->

<xsd:element name="mlist" type="mlistType"/>

<!-- 媒体源（用户或设备编号）列表-->

<xsd:element name="medialist" type="medialistType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例1：开启媒体转发通知

<notify>

<commandname>bizcnf</commandname>

<operation>pullmediastart</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<mlist>

<mid>00000000002</mid>

<mid>00000000003</mid>

</mlist>

<medialist>

<mediasrcid>00000000004</mediasrcid>

<mediasrcid>00000000005</mediasrcid>

</medialist>

</notify>

###### 示例2：停止媒体转发通知

<notify>

<commandname>bizcnf</commandname>

<operation>pullmediastop</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<mlist>

<mid>00000000002</mid>

<mid>00000000003</mid>

</mlist>

<medialist>

<mediasrcid>00000000004</mediasrcid>

<mediasrcid>00000000005</mediasrcid>

</medialist>

</notify>

* + 1. 会议业务控制

##### 讨论模式

* + - 1. 讨论模式

XML结构描述与[A.6.3.2 暂停分组业务](#_bookmark4)相同，消息体中operation标签取值为discstart（开启讨论

模式）或discstop（结束讨论模式）。

###### 示例1：开启讨论模式

<notify>

<commandname>bizcnf</commandname>

<operation>discstart</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

</notify>

###### 示例2：结束讨论模式

<notify>

<commandname>bizcnf</commandname>

<operation>discstop</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

</notify>

##### 发言通知

* + - 1. 发言通知

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 命令名称类型，bizcnf表示会议组业务命令，bizcmd表示ZH组业务命令-->

<xsd:simpleType name="commandnameType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="bizcnf"/>

<xsd:enumeration value="bizcmd"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 成员发言类型，p表示成员被指定发言，r表示成员主动申请发言-->

<xsd:simpleType name="spkType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="p"/>

<xsd:enumeration value="r"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="commandnameType"/>

<!-- 操作名称，类型定义见A.6.3.1启动分组业务，发言通知取值为spkset -->

<xsd:element name="operation" type="operationType"/>

<!-- 成员退出类型-->

<xsd:element name="spktype" type="spkType"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 业务组ID -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 本操作发起成员ID-->

<xsd:element name="op" type="userIDType"/>

<!-- 本操作相关成员ID，发言成员ID-->

<xsd:element name="mid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例1：指定发言，成员被指定发言

<notify>

<commandname>bizcnf</commandname>

<operation>spkset</operation>

<spktype>p</spktype>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<mid>00000000002</mid>

</notify>

###### 示例2：成员主动申请发言通知

<notify>

<commandname>bizcnf</commandname>

<operation>spkset</operation>

<spktype>r</spktype>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<mid>00000000002</mid>

</notify>

##### 取消发言

* + - 1. 取消发言

XML结构描述与[A.6.3.11 媒体转发授权](#_bookmark7)相同，消息体中operation标签取值为spkcal。

示例：取消发言

<notify>

<commandname>bizcnf</commandname>

<operation>spkcal</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<mid>00000000002</mid>

</notify>

##### 申请发言请求

* + - 1. 申请发言请求

XML结构描述与[A.6.3.7 成员退出请求](#_bookmark5)相同，消息体中operation标签取值为spkreq。

###### 示例：申请发言

<request>

<commandname>bizcnf</commandname>

<operation>spkreq</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<!-- 申请发言成员ID -->

<op>00000000002</op>

<!-- 可处理申请发言请求的成员ID -->

<mid>00000000001</mid>

</request>

##### 申请发言响应

* + - 1. 申请发言响应

申请发言响应用于对申请发言请求进行应答。当接收到申请发言请求后，应发送申请发言响应消息，

同时，若同意成员发言，会议主席所在服务节点应向本业务所有相关的服务节点发送成员主动申请发言 消息，成员主动申请发言通知消息见A.6.4.2发言通知。

XML结构描述与[A.6.3.8 成员退出响应](#_bookmark6)相同，消息体中operation标签取值为spkres。

示例：申请发言响应

<response>

<commandname>bizcnf</commandname>

<operation>spkres</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<!-- 申请发言请求处理的成员ID -->

<op>00000000001</op>

<!-- 申请发言成员ID -->

<mid>00000000002</mid>

<code>1</code>

</response>

* + 1. ZH业务控制

##### 专向ZH

* + - 1. 专向ZH

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 命令名称类型，bizcnf表示会议组业务命令，bizcmd表示ZH组业务命令，专向ZH只能取 值为bizcmd -->

<xsd:simpleType name="commandnameType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="bizcnf"/>

<xsd:enumeration value="bizcmd"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="commandnameType"/>

<!-- 操作名称，类型定义见A.6.3.1启动分组业务，开始专向ZH取值为secretstart，结束专向 ZH取值为secretstop -->

<xsd:element name="operation" type="operationType"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 业务组ID -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 本操作发起成员ID-->

<xsd:element name="op" type="userIDType"/>

<!-- 专向上级成员ID-->

<xsd:element name="upid" type="userIDType"/>

<!-- 专向下级成员ID-->

<xsd:element name="downid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例1：开始专向

<notify>

<commandname>bizcmd</commandname>

<operation>secretstart</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<upid>00000000001</upid>

<downid>00000000002</downid>

</notify>

###### 示例2：结束专向

<notify>

<commandname>bizcmd</commandname>

<operation>secretstop</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<upid>00000000001</upid>

<downid>00000000002</downid>

</notify>

##### 协同ZH

* + - 1. 协同ZH

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 命令名称类型，bizcnf表示会议组业务命令，bizcmd表示ZH组业务命令，协同ZH只能取 值为bizcmd -->

<xsd:simpleType name="commandnameType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="bizcnf"/>

<xsd:enumeration value="bizcmd"/>

</xsd:restriction>

</xsd:simpleType>

<!-- 协同ZH成员列表 -->

<xsd:complexType name="mlistType">

<xsd:sequence>

<xsd:element name="mid" type="userIDType" minOccurs="0" maxOccurs="unbounded"/>

</xsd:sequence>

</xsd:complexType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="commandnameType"/>

<!-- 操作名称，类型定义见A.6.3.1启动分组业务，开始协同ZH取值为corpstart，结束协同ZH取值为corpstop -->

<xsd:element name="operation" type="operationType"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 业务组ID -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 本操作发起成员ID-->

<xsd:element name="op" type="userIDType"/>

<!-- 协同成员列表-->

<xsd:element name="mlist" type="mlistType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例1：开始协同

<notify>

<commandname>bizcmd</commandname>

<operation>corpstart</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<mlist>

<mid>00000000002</mid>

<mid>00000000003</mid>

</mlist>

</notify>

###### 示例2：结束协同

<notify>

<commandname>bizcmd</commandname>

<operation>corpstop</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<mlist>

<mid>00000000002</mid>

<mid>00000000003</mid>

</mlist>

</notify>

##### 越级ZH

* + - 1. 越级ZH

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 命令名称类型，bizcnf表示会议组业务命令，bizcmd表示ZH组业务命令，越级ZH只能取 值为bizcmd -->

<xsd:simpleType name="commandnameType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="bizcnf"/>

<xsd:enumeration value="bizcmd"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="commandnameType"/>

<!-- 操作名称，类型定义见A.6.3.1启动分组业务，开始越级ZH取值为crossstart，结束越级ZH取值为crossstop -->

<xsd:element name="operation" type="operationType"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 业务组ID -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 本操作发起成员ID-->

<xsd:element name="op" type="userIDType"/>

<!-- 越级ZH上级成员ID-->

<xsd:element name="upid" type="userIDType"/>

<!-- 越级ZH下级成员ID-->

<xsd:element name="downid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例1：开始越级

<notify>

<commandname>bizcmd</commandname>

<operation>crossstart</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<upid>00000000001</upid>

<downid>00000000004</downid>

</notify>

###### 示例2：结束越级

<notify>

<commandname>bizcmd</commandname>

<operation>crossstop</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<upid>00000000001</upid>

<downid>00000000004</downid>

</notify>

##### 授权ZH

* + - 1. 授权ZH

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 命令名称类型，bizcnf表示会议组业务命令，bizcmd表示ZH组业务命令，授权ZH只能取 值为bizcmd -->

<xsd:simpleType name="commandnameType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="bizcnf"/>

<xsd:enumeration value="bizcmd"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="commandnameType"/>

<!-- 操作名称，类型定义见A.6.3.1启动分组业务，开始授权ZH取值为authstart，结束授权ZH取值为authstop -->

<xsd:element name="operation" type="operationType"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 业务组ID -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 本操作发起成员ID-->

<xsd:element name="op" type="userIDType"/>

<!-- 授权接收成员ID-->

<xsd:element name="accepauthid" type="userIDType"/>

<!-- 被ZH成员ID-->

<xsd:element name="cmdedid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例1：开始授权ZH

<notify>

<commandname>bizcmd</commandname>

<operation>authstart</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<accepauthid>00000000002</accepauthid>

<cmdedid>00000000004</cmdedid>

</notify>

###### 示例2：结束授权ZH

<notify>

<commandname>bizcmd</commandname>

<operation>authstop</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<accepauthid>00000000002</accepauthid>

<cmdedid>00000000004</cmdedid>

</notify>

##### 接替ZH

* + - 1. 接替ZH

XML结构描述如下：

<?xml version="1.0" encoding="utf-8"?>

<xsd:schema xmlns:xsd="<http://www.w3.org/2001/XMLSchema>" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!-- 命令名称类型，bizcnf表示会议组业务命令，bizcmd表示ZH组业务命令，接替ZH只能取 值为bizcmd -->

<xsd:simpleType name="commandnameType">

<xsd:restriction base="xsd:string">

<xsd:enumeration value="bizcnf"/>

<xsd:enumeration value="bizcmd"/>

</xsd:restriction>

</xsd:simpleType>

<xsd:element name="notify">

<xsd:complexType>

<xsd:sequence>

<!-- commandname命令名称-->

<xsd:element name="commandname" type="commandnameType"/>

<!-- 操作名称，类型定义见A.6.3.1启动分组业务，开始接替ZH取值为replacestart，结束接替 ZH取值为replacestop -->

<xsd:element name="operation" type="operationType"/>

<!-- seq命令序号-->

<xsd:element name="seq" type="uuidType"/>

<xsd:element name="ts" type="xsd:unsignedLong"/>

<!-- 业务组ID -->

<xsd:element name="gid" type="xsd:string"/>

<!-- 本操作发起成员ID-->

<xsd:element name="op" type="userIDType"/>

<!-- 目标成员ID-->

<xsd:element name="targid" type="userIDType"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

###### 示例1：开始接替ZH

<notify>

<commandname>bizcmd</commandname>

<operation>replacestart</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<targid>00000000002</targid>

</notify>

###### 示例2：结束接替ZH

<notify>

<commandname>bizcmd</commandname>

<operation>replacestop</operation>

<seq>3fa306388e4d11e8b445060400ef5315</seq>

<ts>1533630770000</ts>

<gid>123</gid>

<op>00000000001</op>

<targid>00000000002</targid>

</notify>

#### 附 录 B (规范性附录) 静态资源信息管理

* 1. 协议要求

静态资源信息存储在资源目录服务单元中，外部端点通过LDAP V3协议访问资源目录服务单元，获 取存储在资源目录服务单元中的组织机构、用户、编解码设备、服务节点、服务设备等信息。

* 1. 目录结构

资源目录服务单元存储静态资源信息的编目结构如图B.1所示：

dc 图例：



ou 组织机构



设备资源



|  |  |
| --- | --- |
|  |  |
|  |
|  |
|  |  |

用户资源



ou



节点资源

服务设备资源



dc



ou



ou

ou

ou

图 B.1 资源目录服务单元编目结构示意图

其中，dc取值“resources”，组织机构信息的ou取值“departInfo”,用户信息的ou取值“userInfo”，

编解码设备信息的ou取值“equipInfo”，服务节点信息的ou取值“nodeInfo”，服务设备信息的ou取值 “serInfo”。

* 1. 数据描述

资源目录服务存储的数据主要包括：

a) 组织机构信息，包括组织 ID、名称、行政隶属关系、ZH隶属关系、厂商信息等；

b) 用户信息，包括：用户 ID、用户编号、名称、账号、密码、优先级、类型、组织归属、服务节 点归属、厂商信息等；

c) 编解码设备信息，包括：设备 ID、设备编号、名称、IP 地址、端口、密码、设备类型、组织归 属、服务节点归属、厂商信息等。

d) 服务节点信息，包括：服务节点 ID、服务节点名称、上级服务节点、关联服务节点、厂商信息 等。

e) 服务设备信息，包括：设备 ID、设备编号、名称、IP 地址、端口、密码、设备类型、协议类型、 服务节点归属、厂商信息等。

* + 1. 组织机构信息
       1. 对象类定义

组织机构信息LDAP对象类定义如下所示：

objectclass ( 1.1.1.1 NAME departInfo STRUCTURAL

MUST (orgUuid $orgName $ orgRalation $ orgCmdRalation $ orgFactInfo )

)

* + - 1. 对象类字段描述

组织机构信息字段描述见表B.1。

##### 表 B.1 组织机构信息对象类字段描述

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 字段描述 | 字段名称 | 字段类型 | 字段长度 | 备注 |
| 1 | 组织 ID | orgUuid | directoryString | 0-32 | UUID，不包含除数字、大小写字 母外的其他特殊字符 |
| 2 | 组织名称 | orgName | directoryString | 0-256 |  |
| 3 | 行政隶属关系 | orgRelation | directoryString | 0-32 | 上级组织 ID 若无，填“NULL” |
| 4 | ZH隶属关系 | orgCmdRelation | directoryString | 0-32 | ZH上级 ID 默认值为上级组织 ID |
| 5 | 厂商信息 | orgFactInfo | directoryString | 0-512 |  |

* + 1. 用户信息
       1. 对象类定义

用户信息LDAP对象类定义如下所示：

objectclass ( 1.1.1.2 NAME userInfo STRUCTURAL

MUST ( userUuid $ userNo $ userName $ userAccount $ userPwd $ userLevel $ userType

$ userOrg $ userNode $ userFactInfo)

)

* + - 1. 对象类字段描述

用户信息字段描述见表B.2。

##### 表 B.2 用户信息对象类字段描述

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 字段描述 | 字段名称 | 字段类型 | 字段长度 | 备注 |
| 1 | 用户 ID | userUuid | directoryString | 32 | UUID，不包含除数字、大小写字  母外的其他特殊字符 |
| 2 | 用户编号 | userNo | numericString | 11 | SIP 账号 |
| 3 | 用户名称 | userName | directoryString | 0-256 |  |
| 4 | 用户账号 | userAccount | directoryString | 0-128 |  |
| 5 | 用户密码 | userPwd | directoryString | 0-64 | Base64 密文 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 字段描述 | 字段名称 | 字段类型 | 字段长度 | 备注 |
| 6 | 用户级别 | userLevel | integer | 1-2 | 0~99 |
| 7 | 用户类型 | userType | integer | 1-2 | 1-普通用户  2-管理员用户 可扩展 |
| 8 | 组织归属 | userOrg | directoryString | 0-32 |  |
| 9 | 服务节点归属 | userNode | directoryString | 0-32 |  |
| 10 | 厂商信息 | userFactInfo | directoryString | 0-512 |  |

* + 1. 编解码设备信息
       1. 对象类定义

编解码设备信息LDAP对象类定义如下所示：

objectclass ( 1.1.1.3 NAME equipInfo STRUCTURAL

MUST ( equipUuid $ equipNo $ equipName $ equipAddr $ equipPort $ equipType $ equipPwd

$ equipOrg $ equipNode $ equipFactInfo )

)

* + - 1. 对象类字段描述

编解码设备信息字段描述见表B.3。

##### 表 B.3 编解码设备信息对象类字段描述

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 字段描述 | 字段名称 | 字段类型 | 字段长度 | 备注 |
| 1 | 设备 ID | equipUuid | directoryString | 32 | UUID，不包含除数字、大小写字母 外的其他特殊字符 |
| 2 | 设备编号 | equipNo | numericString | 11 | SIP 账号 |
| 3 | 设备名称 | equipName | directoryString | 0-256 |  |
| 4 | 设备 IP | equipAddr | directoryString | 0-256 |  |
| 5 | 设备端口 | equipPort | integer | 1-10 |  |
| 6 | 设备类型 | equipType | integer | 1-2 | 1-业务控制单元  2-编码设备  3-解码设备  4-应用服务单元  5-信令控制服务单元  6-媒体处理服务单元  7-媒体存储服务单元  8-信令网关服务单元  9-媒体网关服务单元  10-导播控制服务单元  11-编解码一体设备 可扩展  编解码设备类型只能是 2、3 或 11 |
| 7 | 密码 | equipPwd | directoryString | 0-64 | Base64 密文，SIP 注册密码 |
| 8 | 组织归属 | equipOrg | directoryString | 0-32 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 9 | 服务节点归属 | equipNode | directoryString | 0-32 |  |
| 10 | 厂商信息 | equipFactInfo | directoryString | 0-512 |  |

* + 1. 服务节点信息
       1. 对象类定义

服务节点信息LDAP对象类定义如下所示：

objectclass ( 1.1.1.4 NAME nodeInfo STRUCTURAL

MUST ( nodeUuid $ nodeName $ nodeFather $ nodeRelations $ nodeFactInfo )

)

* + - 1. 对象类字段描述

服务节点信息字段描述见表B.4。

##### 表 B.4 服务节点信息对象类字段描述

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 字段描述 | 字段名称 | 字段类型 | 字段长度 | 备注 |
| 1 | 服务节点 ID | nodeUuid | directoryString | 0-32 | UUID，不包含除数字、大小写字母 外的其他特殊字符 |
| 2 | 服务节点名称 | nodeName | directoryString | 0-256 |  |
| 3 | 上级服务节点 ID | nodeFather | directoryString | 0-512 | 上级服务节点 ID 集合，采用英文半 角逗号符号分割，应与关联服务节 点互斥  若无，填“NULL” |
| 4 | 关联服务节点 ID | nodeRelations | directoryString | 0-512 | 关联服务节点 ID 集合，采用英文半 角逗号符号分割，应与上级服务节 点互斥  若无，填“NULL” |
| 5 | 厂商信息 | nodeFactInfo | directoryString | 0-512 |  |

* + 1. 服务设备信息
       1. 对象类定义

服务设备信息LDAP对象类定义如下所示：

objectclass ( 1.1.1.5 NAME serInfo STRUCTURAL

MUST ( serUuid $ serName $ serAddr $ serPort $ serPwd $ serType $ serPro $ serNode

$ serFactInfo )

)

* + - 1. 对象类字段描述

服务设备信息字段描述见表B.5。

##### 表 B.5 服务设备信息对象类字段描述

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 字段描述 | 字段名称 | 字段类型 | 字段长度 | 备注 |
| 1 | 设备 ID | serUuid | directoryString | 32 | UUID，不包含除数字、大小写字母 外的其他特殊字符 |
| 2 | 设备编号 | serNo | numericString | 11 | SIP 账号 |
| 3 | 设备名称 | serName | directoryString | 0-256 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | IP 地址 | serAddr | directoryString | 0-256 |  |
| 5 | 端口 | serPort | In teger | 1-10 |  |
| 6 | 密码 | serPwd | directoryString | 0-64 | Base64 密文，SIP 注册密码 |
| 7 | 设备类型 | serType | integer | 1-2 | 1-业务控制单元  2-编码设备  3-解码设备  4-应用服务单元  5-信令控制服务单元  6-媒体处理服务单元  7-媒体存储服务单元  8-信令网关服务单元  9-媒体网关服务单元  10-导播控制服务单元  11-编解码一体设备 可扩展  服务设备类型不包括 2、3 或 11 |
| 8 | 协议类型 | serPro | integer | 1-2 | 设备支持的协议类型 1-TCP  2-UDP  3-TCP 和 UDP |
| 9 | 服务节点归属 | serNode | directoryString | 0-32 |  |
| 10 | 厂商信息 | serFactInfo | directoryString | 0-512 |  |

#### 附 录 C (规范性附录) 扩展 SDP 定义

系统SIP消息体携带的SDP应符合IETF RFC 2327的相关要求，具有如下字段： 会话描述

v=（协议版本） o=（发起者和会话标识符）

s=（会话名字） i=\*（会话信息）

u=\*（URI标识符） e=\*（邮件地址）

p=\*（电话号码） c=\*（连接信息）

b=\*（带宽信息） 时间描述

t=\*（时间信息） r=\*（重复时间）

媒体描述 a=\*（零或多个全局媒体属性）

m=\*（媒体名称和传输地址） b=\*（带宽信息）

c=\*（连接信息） a=\*（零或多个媒体属性）

说明： s字段：用于描述请求媒体流的操作类型。s字段格式如下：

s=<trans app>

其中，<trans app>取值包括：“Call”代表双向视音频实时呼叫，支持建立双向媒体传输；“Play” 代表单向视音频实时点播；“Playback”代表历史回放；“Store”代表媒体流存储。

c字段：用于描述会话的连接信息，应在会话描述中包含一个c字段或在每个媒体描述都包含c字段，

若会话描述和媒体描述中均包含c字段，则应采用媒体描述中的c字段。c字段格式如下： c=<网络类型><地址类型><链接地址>

其中，<网络类型>取值为文本字符串“IN”，表示“Internet”；<地址类型>取值包括：“IP4”和 “IP6”；<连接地址>标识地址信息，当<地址类型>为IP4和IP6时，定义如下：

a) 当会话是单播时，地址为单播地址；当会话是组播时，地址为组播组地址；

b) 当地址类型为 IP4 时，需给出 TTL 值，表示数据包的生存时间，取值范围是 0~255。当地址类型 为 IP6 时，无需给出 TTL 值；

c) 对于有多个组播地址的情况，可用如下方法表示：

<组播地址基地址>[/]/<地址数量>

若未给出地址数量，则默认数量为1，例如：

c=IN IP4 224.2.1.1/127/3

语法上等价于：

c=IN IP4 224.2.1.1/127

c=IN IP4 224.2.1.2/127 c=IN IP4 224.2.1.3/127

此种多地址的表示方法只能用于媒体描述，而不能用于会话描述，且不能用于表示多个单播地址。

b字段：用于描述会话的带宽信息，应在会话描述中包含一个b字段或在每个媒体描述都包含b字段， 若会话描述和媒体描述中均包含b字段，则应采用媒体描述中的b字段。b字段格式如下：

b=<带宽类型>:<带宽值>

其中，带宽类型取值包括：“AS”和“TIAS”，会话描述b字段应采用“AS”方式描述带宽，表示所 占用的最大带宽，“AS”带宽计算方法统计IP头、UDP头、RTP头和RTP净载，媒体描述b字段应采用“TIAS”

方式描述带宽，应对每种媒体类型（主要是视频）进行独立描述，采用RTP传输时，“TIAS”带宽计算方 法只统计RTP净载，不统计IP头、UDP头、RTP头等内容；带宽值单位是kbps。

t字段：用于描述会话的开始时间和结束时间。如果会话在不规则的多个时间段内有效，则应采用 多个t字段。如果时间段规则，则应使用r字段。t字段格式如下：

t=<开始时间><结束时间> 其中，开始时间和结束时间的取值采用以1900年开始计算的NTP时间的十进制表示，以秒为单位。

如果开始时间为0，则表示会话为永久会话；如果结束时间为0，则表示会话时间上界不限定。 当媒体调阅时，应采用t字段值标识待回放媒体文件的开始时间和结束时间，开始时间和结束时间

均为待回放媒体文件录制时间段中的某个时刻。 r字段：用于描述会话的重复时间。缺省情况下以秒（s）为单位，也可以使用分钟（m）、小时（h）

或天（d）为单位。r字段格式如下： r=<重复的间隔><持续时长><相对开始时间的偏移>

a字段：用于描述媒体属性，会话描述和媒体描述中均可包含a字段，会话描述中的a字段表示全局 的媒体属性，媒体描述中的a字段表示具体该类型媒体的属性。a字段格式如下：

a=<属性>或a=<属性>:<值> 若媒体传输支持RTP端口复用，则应采用RTP端口复用属性和RTP流复用标识两个全局媒体属性描述，

具体如下：

* + - * 1. a=rtpport-mux RTP端口复用属性，用于表示支持RTP端口复用机制。若使用端口复用机制，请求方应在会话层携带

RTP端口复用属性以及RTP流复用标识，响应方应在响应消息中确认是否采用该机制，若响应方在会话层 也带有RTP端口复用属性和RTP流复用标识，则表示本次会话媒体流采用端口复用机制进行传输，否则媒 体协商失败，应终止会话。

* + - * 1. a=muxid:value RTP流复用标识，muxid属性代表RTP流的复用ID标识，value值为32位随机无符号整数，取值范围是

（0~4294967295）。

m字段：用于描述媒体的类型、端口、传输层协议和净载类型等信息。媒体描述以m字段开始，到下 一个m字段结束或者到整个SDP结束，SDP可包含多条媒体描述。m字段格式如下：

m=<媒体类型><端口><协议><净载类型>

其中，<媒体类型>取值包括“video”和“audio”，采用“video”表示传输视频内容，采用“audio” 表示传输音频内容；<端口>标识媒体的传输端口；<协议>标识媒体的传输协议，取值应为“RTP/AVP”，

表示传输层协议为RTP over UDP；净载类型标识媒体的RTP净载类型,H.264宜采用98，H.265宜采用100，

AVS宜采用102。 媒体描述可以包含多种属性，具体如下：

1. a=ptime:<packet time>

通常只用于描述音频编码/封包时长，单位为毫秒。

1. a=maxptime:<maximum packet time> 用于描述媒体封包的最大时长，通常只用于音频数据，若基于帧编码，则通常是帧长的整倍数，单

位为毫秒。

1. a=rtpmap:<payload type><encoding name>/<clock rate>[/<encoding parameters>] 用于匹配RTP净载类型与媒体编码格式。<payload type>用于匹配m字段的净载类型；<encoding

name>用于描述媒体编码格式的名称，<clock rate>用于描述时钟频率，<encoding parameters>用于描 述编码参数，若编码参数与编解码协议相关，应该在其它的属性中（如“a=fmtp”）定义。

1. a=recvonly 用于描述媒体传输方向为只接收。
2. a=sendonly 用于描述媒体传输方向为只发送。
3. a=sendrecv 用于描述媒体传输方向为双向，既接收又发送。
4. a=inactive 非激活模式 用于描述媒体传输为非激活模式，既不接收也不发送。
5. a=fmtp:<format><format specific parameters> 用于描述具体的媒体编解码参数，<format>用于匹配媒体编码格式，应是m字段中标识的某种净载

类型；<format specific parameters>用于描述编解码参数，可以采用任意参数，不依赖于字符集，每 个媒体描述最多只能有一个该属性，若媒体类型为H.264，<format specific parameters>应遵循IETF RFC 3984的要求,若媒体类型为H.265，<format specific parameters>应遵循IETF RFC 7798的要求。

1. a=rtpid:<rtp stream id value>[<rtp stream id value>] rtpid属性的详细描述见附录A.4.1。
2. a=crypto:<tag><crypto-suite><key-params>[<session-params>] 用于为单播会话描述加密方法、密钥参数和会话参数，只能出现在媒体级，详见IETF RFC 4568。
3. a=connection:<value> 用于描述连接是否已存在，属性值包括new和existing。其中，new表示为新建立连接，existing表

示目前连接已存在。

1. a=rtcp-fb:<payload type> nack

用于表示媒体支持的RTCP Feedback扩展，应采用a=rtcp-fb:<payload type> nack表示支持媒体层

直接进行丢包反馈和丢包重传申请。如果请求方携带“rtcp-fb”参数，并且应答方也包含“rtcp-fb” 参数，则代表呼叫双方能够支持该RTCP扩展，否则不支持，详细见IETF RFC 4585。

#### 附 录 D (规范性附录) 方法响应命令集

D.1 常用响应命令集

信令方法常用响应命令集见表D.1。

表 D.1 方法响应命令集

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 方法 | 响应 |
| 1 | 注册 | REGISTER | 100、200、400、401、403、404、423 |
| 2 | 呼叫 | INVITE | 100、180、181、182、183、200、300、301、302、305、380、399、400、  404、407、415、417、423、481、482、483、486、488、503 |
| 3 | 呼叫确认 | ACK |  |
| 4 | 取消 | CANCEL | 200、481、487 |
| 5 | 能力查询 | OPTIONS | 100、200、486、488 |
| 6 | 挂断 | BYE | 100、200、408、481 |
| 7 | 控制消息 | MESSAGE | 200、202、400、417、513、451、452、453、454、455、456、457、488 |
| 8 | 通知消息 | NOTIFY | 200、404、405、410、416、451、452、453、454、455、457、480-485、  488、489、501、604 |
| 9 | 更新消息 | UPDATE | 200、423、488、491、500 |
| 10 | 信号消息 | INFO | 200、481、488 |
| 11 | 订阅消息 | SUBSCRIBE | 200、202、204、403、404、405、410、416、480-485、488、489、501、604 |
| 12 | 发布消息 | PUBLISH | 200、400、412、415、423、451、452、453、454、455、457、488、489、  503、513 |

其中，100 Trying、180 Ringing、181 Call Is Being Forwarded、182 Queued和183 Session Progress等1xx临时响应见RFC 3261。

200 OK最终响应见RFC 3261，202 Accepted最终响应见RFC 6665，204 No Notification最终响应 见RFC 5839。

300 Multiple Choices、301 Moved Permanently、302 Moved Temporarily、305 User Proxy、380 Alternative Service等3xx最终响应见RFC 3261。

400 Bad Request、401 Unauthorized、403 Forbidden、404 Not Found、405 Method Not Allowed、

407 Proxy Authentication Required、408 Request Timeout、410 Gone、415 Unsupported Media Type、416 Unsupported URI Scheme、423 Interval Too Brief、480 Temporarily Unavailable、481

Call/Transaction Does Not Exist、482 Loop Detected、483 Too Many Hops、484 Address Incomplete、

485 Ambiguous、486 Busy Here、487 Request Terminated、491 Request Pending等4xx最终响应见RFC 3261，412 Conditional Request Failed最终响应见RFC 3903，417 Unsupported Resource Priority

最终响应见RFC 4412，489 Bad Event最终响应见RFC 6665

500 Server Internal Error、501 Not Implemented、503 Service Unavailable、513 Message Too Large等5xx最终响应见RFC 3261。

604 Does Not Exist Anywhere最终响应见RFC 3261。

451 XML Error、452 Not Accepted Label、453 Error Value、454 Unknow Label、455 Incomplete Label、 456 Control Authorization Required、457 Bad Content Type、488 Not Acceptable Here

等最终响应作为自定义用途，详细见附录D.2节。

* 1. 自定义响应
     1. 488 Not Acceptable Here

该响应在本部分中表示请求为跨服务节点消息，但服务节点间未互相注册，请求被拒绝。

* + 1. 451 XML Error

该响应表示XML错误，用户代理服务器无法解析该消息。该响应适用于MESSAGE、PUBLISH、NOTIFY

请求。

* + 1. 452 Not Accpted Label

该响应表示请求消息的消息体完整，有字段无法解析，但可以被处理。该响应适用于MESSAGE、

PUBLISH、NOTIFY请求。

* + 1. 453 Error Value

该响应表示请求消息的消息体完整，存在字段的值错误，处理失败。该响应适用于MESSAGE、PUBLISH、

NOTIFY请求。

* + 1. 454 Unknown Label

该响应表示请求消息的消息体不完整，存在无法解析字段，处理失败。该响应适用于MESSAGE、

PUBLISH、NOTIFY请求。

* + 1. 455 Incomplete Label

该响应表示请求消息的消息体不完整，缺字段，处理失败。该响应适用于MESSAGE、PUBLISH、NOTIFY

请求。

* + 1. 456 Control Authorization Required

该响应表示无操作权限，应先获得授权。该响应适用于MESSAGE请求。

* + 1. 457 Bad Content Type

该响应表示请求消息的Content-Type头域无法解析。

#### 附 录 E (规范性附录) 媒体调阅控制命令

E.1 命令名称和说明

媒体调阅控制命令由客户端到服务器的请求消息和由服务器到客户端的应答消息完成，请求和应答 引用RTSP（IETF RFC 2326）协议中的部分请求和应答消息格式。

消息包括一个起始行、一个或多个消息头（message header）、一个表示标题头结束的空行（即CRLF 前没有内容的行）和一个消息体（可选）。

message=start-line message header CRLF

[messagebody]

消息有请求和应答两种，在每对请求-应答消息中，应包含相同的CSeq头域，具体描述如下：

* + - 1. 请求

请求消息的起始行格式为Method SP RTSP-Version CRLF。

其中Method：请求命令；SP：空白符；RTSP-Version：协议版本号；CR：回车；LF：换行。 请求命令包括：PLAY，PAUSE，TEARDOWN。

* + - 1. 应答

应答消息的起始行格式为Status-Line=RTSP-Version SP Status-Code SP Reason-Phrase CRLF。

其中RTSP-Version：协议版本号；Status-Code：3位状态码，用于回应请求时表示主机状态；Reason- Phrase：是与状态码对应的文本解释。

* 1. 命令定义
     1. 媒体播放命令

客户端发送PLAY请求消息，请求服务器发送媒体。应支持Range头，在Range头中给出播放时间范围，

播放指定时间段的媒体，见IETF RFC 2326的12.29节；时间范围应支持npt，smpte相对时间戳范围。服 务器的响应消息中给出RTP-Info头信息，见IETF RFC 2326的12.33节。

* + 1. 暂停播放命令

客户端发送PAUSE请求消息，请求服务器暂停发送媒体，但不释放资源。见IETF RFC 2326的10.6节。

* + 1. 快进／慢进命令

在客户端发送的PLAY请求消息中，应使用Scale头控制播放的快慢，见IETF RFC 2326的12.34节。

Scale等于1时，为正常播放；Scale不等于1时，为正常播放速率的倍数；Scale等于负数时，为倒放。应 支持Range头，在Range头中给出npt格式的播放时间范围。

* + 1. 随机拖放命令

在客户端发送的PLAY请求消息中，应支持Range头域，使用smpte相对时间戳范围，实现随机拖放播

放。

* + 1. 停止命令

客户端发送TEARDOWN请求消息，停止发送指定流，结束会话，并释放相关资源。

* + 1. 应答命令

客户端、服务器端应支持应答命令的状态码200、4xx、5xx。

#### 附 录 F (规范性附录) 优先级控制方法

应 支 持 根 据 用 户 优 先 级 分 配 使 用 系 统 资 源 。 优 先 级 控 制 应 采 用 Resource-Priority 和 Accept- Resource-Priority头域。

Resource-Priority头域用于标识请求方期望获得的资源访问优先级，采用Require头域告知被叫方 对Resource-Priority的强制要求。当请求包含Resource-Priority头域时，信令控制服务单元、信令网

关服务单元、媒体处理服务单元、编解码单元等相关功能实体应做出判决，保证当资源使用发生冲突时 高级别用户优先获得资源使用权限。在优先级排序时，具备Resource-Priority头域的请求的优先级应

高于不具备该头域的请求。

请求的2xx响应采用Accept-Resource-Priority头域标识所采用的优先级，还应包含Supported头域 明确扩展能力，或在417 Unknown Resource-Priority响应中采用Accept-Resource-Priority头域标识

所能接受的优先级。

优先级值的格式为“命名空间.优先级”，不区分大下写。命名空间定义为“CMD”，优先级分为100 级，应遵循高优先级抢占低优先级、相同优先级互不抢占的资源控制策略。优先级值从低到高为：

cmd.0 (最低) cmd.1

cmd.2 cmd.3

…

cmd.96 cmd.97 cmd.98

cmd.99 (最高)

优先级控制方法应符合RFC 4412的具体规定。

#### 附 录 G (建议性附录) 端口使用建议表

系统端口使用建议如表G.1所示。

表 G.1 端口使用建议表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能 | 传输层协议 | 端口建议值 | 备注 |
| 1 | 信令传输 | TCP、UDP | 5060 |  |
| 2 | 媒体传输 | UDP | RTP：19000、19002、19004、19006  RTCP：19001、19003、19005、19007 |  |
| 3 | LDAP | TCP、UDP | LDAP：389  LDAP over SSL：636 | 若资源目录由其它系统提供，  则端口使用应遵循对端系统要 求 |
| 4 | SNMP | TCP、UDP | GET\SET：161 TRAP：162 | 若网管功能由其它系统提供，  则端口使用应遵循对端系统要 求 |
| 5 | NTP | UDP | 123 | 若时统服务由其它系统提供，  则端口使用应遵循对端系统要 求 |
| 6 | HTTP/HTTPS | TCP | HTTP：8080  HTTPS：443 |  |
| 7 | 预留 | TCP、UDP | 18100~18115 | 包括数据库端口及系统其它自  定义端口 |

#### 附 录 H (建议性附录) 媒体协商方法

##### H.1 H.264 媒体协商

H.264视频编解码参数描述应遵循IETF RFC 3984的要求，为保证媒体协商的一致性，典型分辨率宜 遵循表H.1给出的参考值进行媒体协商。

表 H.1 H.264 典型分辨率媒体协商参数参考值

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 分辨率 | 参考帧率 | level | max-mbps | max-fs |
| 3840\*2160 | 30 | 5.1 | 972000 | 32400 |
| 1920\*1200 | 60 | 5 | 540000 | 9000 |
| 1920\*1200 | 30 | 5 | 270000 | 9000 |
| 1920\*1080 | 60 | 4.2 | 486000 | 8100 |
| 1920\*1080 | 30 | 4 | 243000 | 8100 |
| 1280\*720 | 60 | 3.2 | 216000 | 3600 |
| 1280\*720 | 30 | 3.1 | 108000 | 3600 |
| 720\*576 | 60 | 3.1 | 97200 | 1620 |
| 720\*576 | 30 | 3.1 | 48600 | 1620 |
| 704\*576 | 60 | 3.1 | 95040 | 1584 |
| 704\*576 | 30 | 3.1 | 47520 | 1584 |
| 704\*576 | 25 | 3 | 39600 | 1584 |
| 352\*288 | 30 | 2 | 11880 | 396 |
| 176\*144 | 30 | 1.1 | 2970 | 99 |

##### H.2 H.265 媒体协商

H.265视频编解码参数描述应遵循IETF RFC 7798的要求，为保证媒体协商的一致性，典型分辨率宜 遵循表H.2给出的参考值进行媒体协商。

表 H.2 H.265 典型分辨率媒体协商参数参考值

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 分辨率 | 参考帧率 | level | Max luma sample rate  (samples/s) | Max luma picture size  (samples) |
| 3840\*2160 | 60 | 5.1 | 497664000 | 8294400 |
| 3840\*2160 | 30 | 5 | 248832000 | 8294400 |
| 1920\*1200 | 60 | 5 | 138240000 | 2304000 |
| 1920\*1200 | 30 | 5 | 69120000 | 2304000 |
| 1920\*1080 | 60 | 4.1 | 124416000 | 2073600 |
| 1920\*1080 | 30 | 4 | 62208000 | 2073600 |
| 1280\*720 | 60 | 4 | 55296000 | 921600 |
| 1280\*720 | 30 | 3.1 | 27648000 | 921600 |
| 720\*576 | 60 | 3.1 | 24883200 | 414720 |
| 720\*576 | 30 | 3 | 12441600 | 414720 |
| 704\*576 | 60 | 3.1 | 24330240 | 405504 |
| 704\*576 | 30 | 3 | 12165120 | 405504 |
| 704\*576 | 25 | 3 | 10137600 | 405504 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 352\*288 | 30 | 2 | 3041280 | 101376 |
| 176\*144 | 30 | 1 | 760320 | 25344 |

##### AVS媒体协商

* 1. AVS媒体协商
     1. SDP参数

媒体类型video/AVS字符串到SDP域的匹配方法为：

a) SDP “m=” 行的媒体名为 video（媒体类型名）；

1. SDP “a=rtpmap” 行的编码名为 AVS（媒体子类型名）；
2. SDP “a=rtpmap” 行的时钟率（clock rate）为 90000；
3. 如果存在可选参数 profile-level-id、max-mbps、max-fs、max-dpb、max-br、sprop-parameter- sets、parameter-add、packetization-mode、sprop-interleaving-depth、sprop-deint-buf- req、deint-buf-cap、sprop-init-buf-time、sprop-max-don-diff 和 max-rcmd-nalu-size，应

包括在 SDP “a=fmtp”行。这些参数的表示形式为一系列由半角分号“；”分开的“参数=值” 对。

示例：

SDP媒体描述（基准档次级别6.0）： m=video 49170 RTP/AVP 102 a=rtpmap:102 JAVS/90000

a=fmtp:102 profile-level-id=2040;sprop-parameter-sets=[SH#0] 其中[SH#0]表示一个序列头的base64表示。

其中，可选媒体类型参数的语义如下所示：

profile-level-id

《XX音视频编解码算法》（报批稿）第一部分规定的序列头中以下两个字节的base16表示： profile\_id和level\_id。

如果profile-level-id用于表明《XX音视频编解码算法》（报批稿）第1部分中视频位流的特性，

表明为了解码时符合《XX音视频编解码算法》（报批稿）第1部分的一个解码器应支持的档次和级别。 如果profile-level-id用于能力交换或会话建立过程，表明该编解码器支持的档次和支持的对应该

档次的最高级别。

如果profile-level-id不存在，默认档次和级别为基准档次4.0级别。 max-mbps, max-fs, max-dpb 和 max-br

这些参数可以用来表明接收端的能力，而不能用作任何其他用途。包含这些参数中任何一个的能力描述应同时包含profile-level-id参数。接收端应完全支持profile-level-id表明的级别。如下所述， 这四个参数可以用来表明接收端级别规定之外的扩展能力。

当这四个参数中多于一个参数存在时，接收端应同时支持所有表明的能力。例如，如果max-mbps和 max-br同时存在，则接收端支持由profile-level-id指定的相应级别加上扩展的帧率和比特率。也就是说，接收端能够解码这样的视频位流，其最大CTU处理速度（即每秒最大CTU个数）为max-mbps，最大比 特率为max-br，编码图像缓冲区大小按照下面max-br的语义规定的方法导出，而其他的特性符合相应的级别。

接收端给出的这四个参数不能同时符合一个比由profile-level-id指定的相应级别更高的级别的要求。

max-mbps

max-mbps的值是一个整数，指定每秒最大CTU个数，单位为CTU每秒。max-mbps表明接收端能够以高 于profile-level-id指定的级别要求的速率进行解码。如果max-mbps存在，接收端应能够解码这样的视 频位流：除了用max-mbps替代《XX音视频编解码算法》（报批稿）第一部分附录B中表B.5至B.15中的 每秒最大CTU个数之外，该位流符合相应的级别。max-mbps的值应大于或等于《XX音视频编解码算法》

（报批稿）第1部分附录B中表B.5至B.15中给出的每秒最大CTU个数的值。发送端可能根据这个信息发送 比相应级别指定的更高的帧率。

max-fs

max-fs的值是一个整数，指定最大帧尺寸，单位为CTU。max-fs表明接收端能够解码大于profile- level-id指定的级别要求的图像尺寸。如果max-fs存在，接收端应能够解码这样的视频位流：除了用max- fs替代《XX音视频编解码算法》（报批稿）第一部分附录B中表B.5至B.15中的每帧最大CTU个数之外， 该位流符合相应的级别。max-fs的值应大于或等于《XX音视频编解码算法》（报批稿）第1部分附录B 中表B.5至B.15给出的每帧最大CTU个数的值。发送端可能根据这个信息发送比相应级别指定的尺寸更大 的图像，此时帧率可能减小。

max-dpb

max-dpb的值是一个整数，指定最大解码图像缓冲区大小，单位为1000 bit。max-dpb表明接收端具 有比profile-level-id指定的级别要求的最少解码图像缓冲区内存更多的内存。如果max-dpb存在，接 收端应能够解码这样的视频位流：除了用1000×(max-dbp)替代《XX音视频编解码算法》（报批稿）第 一部分附录B中表B.5至B.15的BBV缓冲区大小（单位为位）等于之外，该位流符合相应的级别。1000× (max-dbp)的值应大于或等于《XX音视频编解码算法》（报批稿）第1部分附录B中表B.5至B.15中给出 的BBV缓冲区大小（单位为位）的值。发送端器可能根据这个信息生成比相应级别指定的BBV缓冲区大小 能够达到的更高的压缩效率。

max-br

max-br的值是一个整数，指定最大视频比特率，单位为1000 bit。max-br表明接收端的视频解码器 能够解码比profile-level-id指定的级别要求的更高的比特率。如果max-br存在，接收端的视频解码器 应能够解码这样的视频位流：除了用1000×(max-br)替代《XX音视频编解码算法》（报批稿）第一部 分附录B中表B.5至B.15的最大比特率之外，该位流符合相应的级别。1000×(max-br)的值应大于或等于

《XX音视频编解码算法》（报批稿）第1部分附录B中表B.5至B.15给出的最大比特率的值。发送端可 能根据这个信息发送比相应级别指定的更大的最大比特率，从而提高视频质量。

sprop-parameter-sets 这个参数可以用来传输视频序列头。该参数不能用在任何能力交换过程中表明编解码器的能力。该

参数的值为序列头码流的base64表示。这些序列头按解码顺序排列，每两个序列头中间插入一个半角逗 号“,”。

parameter-add

这个参数可以用来表明是否允许这个参数的接收者在它的信令响应中使用sprop-parameter-sets 参数添加序列头，值为0或1。0表示不允许添加序列头；1表示允许添加序列头。如果该参数不存在，其 默认值为1。

packetization-mode 这个参数可以用来表明一个RTP净载类型的特性或接收端的能力。由于只能指出一个配置点，当表

明具有支持多种打包模式的能力时，应使用多个配置点。

值为0或不存在时，应使用《XX音视频编解码算法》（报批稿）第三部分4.1.2.2.2介绍的单NALU 模式；值为1时应使用《XX音视频编解码算法》（报批稿）第三部分4.1.2.2.2介绍的非交织模式；值

为2时应使用《XX音视频编解码算法》（报批稿）第三部分4.1.2.2.2介绍的交织模式。packetization- mode的取值范围为0～2。

sprop-interleaving-depth

当packetization-mode不存在或packetization-mode值为0或1时，sprop-interleaving-depth不存 在。当packetization-mode值为2时，sprop-interleaving-depth应存在。

sprop-interleaving-depth表明NALU流的特性。它指定相对于任一NALU传输顺序在前而解码顺序在

后的NALU数目的最大值。因此，可以保证接收端能够采用大于或等于sprop-interleaving-depth+1个 NALU的解码顺序恢复缓冲区来重建NALU解码顺序。

sprop-interleaving-depth的取值范围为0～32767。 sprop-deint-buf-req

当packetization-mode不存在或packetization-mode值为0或1时，sprop-deint-buf-req不存在。 当packetization-mode值为2时，sprop-deint-buf-req应存在。

sprop-deint-buf-req表明NALU流需要的解交织缓冲区尺寸，单位为字节。该参数的值应大于或等 于解交织缓冲区需要的最大缓冲区充满度（单位为字节）。

sprop-deint-buf-req的取值范围为0～4294967295。 deint-buf-cap

该参数表明接收端的能力，指定接收端具有的用来重建NALU解码顺序的解交织缓冲区空间的大小

（单位为字节）,并表明接收端能够处理任何sprop-deint-buf-req值小于或等于deint-buf-cap的流。 如果参数不存在，则deint-buf-cap的默认值为0。deint-buf-cap的取值范围为0～4294967295。

sprop-init-buf-time

该参数可以用来表明NALU流的特性。如果packetization-mode值为0或1，sprop-init-buf-time不 存在。

该参数表明接收端开始解码前从传输顺序恢复NALU解码顺序所经过的初始缓冲时间。假设传输是可 靠的和瞬时的，传输和解码的时间线相同且第一个包到达时解码开始，则sprop-init-buf-time等于一

个NALU的传输时间和解码时间之差的最大值。

示例： NALU流的传输顺序为从左到右，数值表示解码时间：

0 2 1 3 5 4 6 8 7 ...

假设NALU传输速率恒定，则传输时间为:

0 1 2 3 4 5 6 7 8 ...

让传输时间和解码时间按列相减，得到：

0 -1 1 0 -1 1 0 -1 1 ...

该示例中sprop-init-buf-time的值为1，单位为NALU传输时间间隔。

sprop-init-buf-time为非负整数，单位为1/90000s，取值范围为0～4294967295。如果该参数不存 在，则初始缓冲时间未定义。

除了sprop-init-buf-time，接收端应考虑各种网络单元产生的传输延时抖动。

sprop-max-don-diff 该参数可以用来表明NALU流的特性，但不能用来表明发送端、接收端或编解码器的能力。如果

packetization-mode值为0或1，sprop-init-buf-time不存在。sprop-max-don-diff的取值范围为0～ 32767。该参数不存在表示参数值未规定。sprop-max-don-diff的计算方法见式（1）：

*sprop* max*don* *diff*

max*AbsDON**i**AbsDon**j*,i, j > i

............. (1)

式中：

AbsDON ——NALU的绝对解码顺序号，即其值到达65535后继续增加而不重置为0； i，j ——NALU的传输顺序索引。

AbsDON的计算方法如下：

令m和n为传输顺序连续的两个NALU。对于传输顺序上的第一个NALU（其索引为0），AbsDON(0) =

DON(0)。对于其他的NALU：

a) 如果 DON(m)等于 DON(n), 则 AbsDON 的计算方法见式（2）：

式中：

*AbsDON**n**AbsDON**m*

............................... (2)

m，n ——NALU的传输顺序索引。 如果DON(m)小于DON(n)且DON(n)减DON(m)小于32768,则AbsDON的计算方法见式（3）：

式中：

*AbsDON**n**AbsDON**m**DON**n**DON**m*

..................... (3)

m，n ——NALU的传输顺序索引。

如果DON(m)大于DON(n)且DON(m)减DON(n)大于或等于32768, 则AbsDON的计算方法见式（4）：

*AbsDON**n**AbsDON**m*65536 - *DON**m**DON**n*.................. (4)

式中：

m，n ——NALU的传输顺序索引。 如果DON(m)小于DON(n)且DON(n)减DON(m)大于或等于32768，则AbsDON的计算方法见式（5）：

*AbsDON**n**AbsDON**m**DON**m*65536 *DON**n*................. (5)

式中：

m，n ——NALU的传输顺序索引。

如果DON(m)大于DON(n)且DON(m)减DON(n)小于32768, 则AbsDON的计算方法见式（6）：

*AbsDON**n**AbsDON**m**DON**m**DON**n*..................... (6)

式中：

m，n ——NALU的传输顺序索引； DON(i)——传输顺序索引为i的NALU的解码顺序号，见《XX音视频编解码算法》（报批稿）第三

部分4.1.2.2.3。

max-rcmd-nalu-size 该参数可以用来表明接收端的能力，而不能用作任何其他目的。该参数值指定接收端能够有效处理

的最大NALU尺寸，单位为字节。该参数值是个推荐值而并不是一个严格的上限。发送端可以产生更大的

NALU但是其处理的代价可能更高。

max-rcmd-nalu-size为整数，取值范围为0～4294967295。如果该参数不存在，则其值为未知。发送 端仍然应考虑发送端和接收端之间的MTU尺寸。为此，发送端应运行MTU检测手段。

* + 1. 协商过程

###### 单播模式

* + - 1. 单播模式
         1. 用于识别视频媒体格式的媒体类型参数为 profile-level-id、packetization-mode 和 sprop- deint-buf-req（如果 packetization-mode 等于 2）。这三个参数应对称地使用：即回答者应要 么保留所有的配置参数，要么在无法支持一个或多个参数值的情况下完全去除该媒体格式（净 载类型）。为了简化这些配置的处理和匹配，提供中使用的 RTP 净载类型值应同样在回答中使 用。除非由上述三个参数决定的配置相同，回答中不能包含提供中用到的净载类型值；
         2. 媒体类型参数 sprop-parameter-sets、sprop-deint-buf-req、sprop-interleaving-depth、 sprop-max-don-diff 和 sprop-init-buf-time 用于指示提供者或回答者发送的具有某一配置的 NAL 单元流的特性。这有别于提供/回答参数的一般用途——指示提供者或回答者能够接收的流 的特性。对于视频，提供者假定回答者能够接收采用所提供配置编码的媒体；
         3. 能力参数 max-mbps、max-fs、max-dpb、max-br 和 max-rcmd-nalu-size 可以用来表明其他能力。 对它们的解释取决于方向属性。如果方向属性为 sendonly，则这些参数描述发送端能够产生的 RTP 包和 NALU 流的限制。如果方向属性为 sendrecv 或 recvonly，则这些参数描述接收端能够 接受的限制条件；

d) 如前所述，提供者应在提供中包含交织的 AVS 规定的视频流的解交织缓冲区的尺寸。为了使提 供者和回答者都能够告诉对方自己在解交织缓冲方面的能力，推荐二者都包含 deint-buf-cap。 这个信息可以在第二轮提供和回答过程中选择了 sprop-deint-buf-req 的值之后使用。对于交 织流，如果接收端的能力未知，推荐考虑提供多个缓冲需求不同的净载类型；

e) 参数 sprop-parameter-sets 的使用如前所述。另外，回答者应在它的回答中保留所有从提供中 收到的序列头。如果 parameter-add 为 0，回答中不能增加任何序列头；如果 parameter-add 为 1，回答者可以在它的回答的 sprop-parameter-sets 中增加序列头。无论 parameter-add 的值 为 0 或 1，回答者需答应接收采用包含在自己的回答中的 sprop-parameter-sets 的视频流。

###### 组播模式

* + - 1. 组播模式

对于通过组播方式发布的流，应额外遵守以下规则：

* + - * 1. 回 答 者 不 能 改 变 流 特 性 参 数 sprop-parameter-sets 、 sprop-deint-buf-req 、 sprop- interleaving-depth、sprop-max-don-diff 和 sprop-init-buf-time。因此，一个提供的净载类 型要么被原封不动地接受，要么被去除；
        2. 回答者应对所有申明为 sendrecv 或 recvonly 的流支持接收端能力参数 max-mbps、max-fs、max- dpb、max-br 和 max-rcmd-nalu-size。否则，回答者应去除该媒体格式或拒绝会话。

下面完整地列出各参数在不同情形下的意义：

a) 在“a=sendrecv”或者“a=recvonly” 或者未使用方向属性的提供和回答中各参数的意义如下： 1) 申明要接收的配置或特性：

—— profile-level-id

—— packetization-mode

2) 申明要发送的流的特性（仅当“a=sendrecv”或者未使用方向属性时适用）：

—— sprop-deint-buf-req

—— sprop-interleaving-depth

—— sprop-parameter-sets

—— sprop-max-don-diff

—— sprop-init-buf-time

3) 申明接收端实现的能力：

—— max-mbps

—— max-fs

—— max-dpb

—— max-br

—— deint-buf-cap

—— max-rcmd-nalu-size

4) 申明如何进行提供/回答协商：

—— parameter-add

b) 在“a=sendonly”的提供和回答中各参数的意义如下： 1) 申明提议要发送的流的配置和属性：

—— profile-level-id

—— packetization-mode

—— sprop-deint-buf-reg

—— sprop-max-don-diff

—— sprop-init-buf-time

—— sprop-parameter-sets

—— sprop-interleaving-depth

2) 申明发送端接收流时的能力：

—— max-mbps

—— max-fs

—— max-dpb

—— max-br

—— deint-buf-cap

—— max-rcmd-nalu-size

3) 申明如何进行提供/回答协商：

—— parameter-add

另外应考虑下面的因素：

a) 申明接收端能力的参数通常是可以降低等级的，即它们表达发送端可能的行为的上限。因此发 送端可以选择设定它的编码器只使用这些参数的更小（低）或相等的值。sprop-parameter-sets 不能用来申明发送端的能力；

b) 除了 profile-level-id 的级别（level）部分以外，申明配置点的参数不可以降低等级。这表明 了接收端期望使用的值，发送端应原封不动地使用这些值；

c) 如果申明了发送端的能力，并且申明中使用了不可降低等级的参数，则这些参数表明了一个可 接受的配置。为了达到更好的互用性，可以提供多个可替换的配置，例如不同的打包模式。由于 不可能采用同一个净载类型提供多个配置，如果提供多个配置，每个配置需要自己的 RTP 净载 格式；

d) 接收端应当理解一个净载格式中所有的媒体类型参数，即使它只支持该净载格式的一部分功能。

这保证了接收端能够理解一个提供的流可以被降级到自己能够支持的程度；

e) 回答者可以扩展额外的媒体格式配置到提供中。然而，为了能够使用这些额外的配置，大多数情 形下需要提供者的第二个提供，以给出媒体发送端要使用的流特性参数。这个过程还有另外一 个效果，就是提供者应能够接收这个媒体格式配置，而不仅仅是发送；

f) 如果提供者希望发送和接收的能力不对称，则它应提供不同的 RTP 会话，即分别申明为 recvonly 和 sendonly 的不同 SDP“m=”行。