

国际电信联盟

ITU-T

国际电信联盟
电信标准化部门

H.248.1

(09/2005)

H系列：视听及多媒体系统
视听业务的基础设施 — 通信规程

网关控制协议：第3版

ITU-T H.248.1建议书



ITU-T H系列建议书
视听及多媒体系统

可视电话系统的性质	H.100-H.199
视听业务的基础设施	
概述	H.200-H.219
传输多路复用和同步	H.220-H.229
系统概况	H.230-H.239
通信规程	H.240-H.259
活动图像编码	H.260-H.279
相关系统概况	H.280-H.299
视听业务的系统和终端设备	H.300-H.349
视听和多媒体业务的号码簿业务体系结构	H.350-H.359
视听和多媒体业务的服务质量体系结构	H.360-H.369
多媒体的补充业务	H.450-H.499
移动性和协作程序	
移动性和协作、定义、协议和程序概述	H.500-H.509
H系列多媒体系统和业务的移动性	H.510-H.519
移动多媒体协作应用和业务	H.520-H.529
移动多媒体应用和业务的安全性	H.530-H.539
移动多媒体协作应用和业务的安全性	H.540-H.549
移动性互通程序	H.550-H.559
移动多媒体协作互通程序	H.560-H.569
宽带和三网合一多媒体业务	
在VDSL上传送宽带多媒体业务	H.610-H.619

欲了解更详细信息，请查阅ITU-T建议书目录。

摘 要

为了获得更好的可扩展性，本建议书将ITU-T H.246建议书定义的H.323网关功能分解为多个子功能实体，并且制定协议用于子功能实体之间进行通信。从而分解后的H.323网关功能可获得更好的可扩展性，使其可以与广泛部署的电路交换网络（SCN）能力互通（例如SS7交换机处理能力）。同时H.323网关功能也可以由隶属于多个设备制造商分布在多个物理平台的功能实体组合完成。制定本建议书的目标是在现有H.323系统所支持的功能上扩展新的功能，以及在现有H.323系统所支持的操作模式的基础上提供一种新的模式。

本建议书与ITU-T H.248.1建议书第2版相比，主要包含了以下增强功能：

- 通过包来定义关联属性的能力；
- IEPS关联属性；
- “标志位”用于在注册阶段指示媒体网关中包含“退出服务”的终结点；
- 新定义的消息分段包和用于非分段传送的流程；
- 重新规定包定义的要求和新定义的包模版；
- 重新规定profile的定义要求和新的profile模版；
- 基于流的新增统计；
- 新增的信号请求标识符用于区分信号列表（SignalList）中的相似信号；
- 新增的基本信号属性用于指示播放信号的方向；
- 新增的两个新的拓扑类型；
- 新增的适用于信号列表中信号之间的时延计数器；
- 新增的用于命令响应的关联标识符列表（ContextIDList）；
- 新增的适用于命令和响应的终结点标识列表（TerminationIDList）结构；
- 重新规定业务改变流程；
- 新增的用于媒体网关控制器来调节所接收到的通知消息的速率；
- 新增的通过过滤条件来进行审计请求的能力。

来 源

ITU-T第16研究组（2005-2008年）按照ITU-T A.8建议书规定的程序，于2005年9月13日批准了ITU-T H.248.1建议书。

前 言

国际电信联盟（ITU）是从事电信领域工作的联合国专门机构。ITU-T（国际电信联盟电信标准化部门）是国际电信联盟的常设机构，负责研究技术、操作和资费问题，并且为在世界范围内实现电信标准化，发表有关上述研究项目的建议书。

每四年一届的世界电信标准化全会（WTSA）确定 ITU-T 各研究组的研究课题，再由各研究组制定有关这些课题的建议书。

WTSA 第 1 号决议规定了批准 ITU-T 建议书须遵循的程序。

属 ITU-T 研究范围的某些信息技术领域的必要标准，是与国际标准化组织（ISO）和国际电工技术委员会（IEC）合作制定的。

注

本建议书为简明扼要起见而使用的“主管部门”一词，既指电信主管部门，又指经认可的运营机构。

遵守本建议书的规定是以自愿为基础的，但建议书可能包含某些强制性条款（以确保例如互操作性或适用性等），只有满足所有强制性条款的规定，才能达到遵守建议书的目的。“应该”或“务必”等其他一些强制性用语及其否定形式被用于表达特定要求。使用此类用语不表示要求任何一方遵守本建议书。

知识产权

国际电联提请注意：本建议书的应用或实施可能涉及使用已申报的知识产权。国际电联对无论是其成员还是建议书制定程序之外的其他机构提出的有关已申报的知识产权的证据、有效性或适用性不表示意见。

至本建议书批准之日止，国际电联已经收到实施本建议书可能需要的受专利保护的知识产权的通知。但需要提醒实施者注意的是，这可能并非最新信息，因此特大力提倡他们通过下列网址查询电信标准化局（TSB）的专利数据库：<http://www.itu.int/ITU-T/ipr/>。

© 国际电联 2006

版权所有。未经国际电联事先书面许可，不得以任何手段复制本出版物的任何部分。

目 录

	页码
1 范围	1
2 参考文献	1
2.1 规范性参考文献	1
2.2 资料性参考文献	3
3 定义	4
4 缩写	5
5 惯例	6
6 连接模型	6
6.1 关联	7
6.2 终结点	8
6.3 通配原则	14
7 命令	16
7.1 描述符	16
7.2 命令应用编程接口	37
8 事务	52
8.1 命令参数	54
8.2 事务应用编程接口	54
8.3 消息	56
9 传送	57
9.1 命令顺序	57
9.2 预防重启动雪崩	58
9.3 预防Notify雪崩	59
10 安全考虑	59
10.1 协议连接的保护	59
10.2 过渡性AH机制	59
10.3 媒体连接的保护	60
11 MG-MGC控制接口	60
11.1 多个虚拟MG	60
11.2 冷启动	61
11.3 协议版本协商	61
11.4 MG故障	62
11.5 MGC故障	62
11.6 MGC-MG控制联系监控	63
12 包定义	63
12.1 定义包的指导原则	64
12.2 事件和信号参数定义的指导原则	67
12.3 标识符	68
12.4 包注册	68
13 协议子集	68
14 IANA注意事项	68

14.1	包	68
14.2	差错代码	69
14.3	业务改变原因值	69
14.4	协议子集	69
附件A	— 协议的二进制编码	70
A.1	通配符编码	70
A.2	ASN.1句法规范	71
A.3	DigitMap和路径名称	88
附件B	— 协议的文本编码	89
B.1	通配符编码	89
B.2	ABNF规范	89
B.3	十六进制字节编码	104
B.4	十六进制字节序列	104
附件C	— 媒体流属性标签	105
C.1	通用媒体属性	105
C.2	Mux属性	106
C.3	通用承载属性	106
C.4	通用ATM属性	107
C.5	帧中继属性	109
C.6	IP	109
C.7	ATM AAL 2	110
C.8	ATM AAL 1	111
C.9	承载能力	112
C.10	AAL 5属性	120
C.11	等效的SDP	120
C.12	H.245	121
附件D	— 采用IP的传送	121
D.1	使用应用层成帧（ALF）的采用IP/UDP的传送	121
D.2	采用TCP	125
附件E	— 基本包	126
E.1	通用包	126
E.2	基根包	128
E.3	信号音发生器包	131
E.4	信号音检测包	132
E.5	基本DTMF发生器包	135
E.6	DTMF检测包	137
E.7	呼叫进展信号音发生器包	139
E.8	呼叫进展信号音检测包	140
E.9	模拟线路监测包	141
E.10	基本连续性包	144
E.11	网络包	146

	页码
E.12 RTP包	149
E.13 TDM电路包	151
E.14 分段包	152
E.15 通知行为	155
附件F — 业务改变程序	159
F.1 引言	159
F.2 控制联系定义	161
F.3 导致ServiceChange程序的事件	161
F.4 ServiceChange单元描述	165
F.5 ServiceChange参数的使用	168
F.6 ServiceChange与TerminationState	170
附录一 — 呼叫流程示例	171
I.1 驻地网关之间的呼叫程序	171
附录二 — H.248包模版	180
附录三 — H.248协议子集定义模版	183

ITU-T H.248.1建议书

网关控制协议：第3版

1 范围

本建议书制定的协议可适用于由媒体网关物理分离的功能实体之间。从系统角度来说，分布在一个或者多个物理实体的被分离的网关与由ITU-T H.246建议书制定的单一网关之间，功能上并无区别。本建议书不对网关、多点控制单元（MCU）和交互式语音应答（IVR）的工作模式进行制定。相反，本建议书仅定义了一种通用的框架来实现以上应用。

分组网络接口可以是IP、ATM或其他任何可能接口。电路交换网接口能支持多种电路交换网络信令系统，如信号音信令、ISDN、ISUP、QSIG和GSM。并且部署在不同国家的以上电路交换网信令系统也将被支持。

宣称合乎ITU-T H.248.1建议书第1版的产品必须符合于2000年6月批准并于2002年3月重新发布的ITU-T H.248.1建议书的所有强制性要求。

宣称合乎ITU-T H.248.1建议书第2版的产品必须符合于2002年5月批准并于2004年3月更新的ITU-T H.248.1建议书的强制性要求。

宣称合乎本建议书的产品必须符合于2005年9月发布的ITU-T H.248.1建议书的所有强制性要求。

所有产品必须通过业务改变版本（ServiceCHangeVerison）参数来表明所用协议的版本，其中“1”指的是2002年3月发布的H.248.1建议书，“2”指的是2004年3月发布的H.248.1建议书勘误1，“3”指的是ITU-T H.248.1建议书（09/2005）。

2 参考文献

下列ITU-T建议书和其他参考文献的条款，在本建议书中的引用而构成本建议书的条款。在出版时，所指出的版本是有效的。所有的建议书和其他参考文献均会得到修订，本建议书的使用者应查证是否有可能使用下列建议书或其他参考文献的最新版本。当前有效的ITU-T建议书清单定期出版。本建议书引用的文件自成一体时不具备建议书的地位。

2.1 规范性参考文献

- ITU-T Recommendation E.106 (2003), *International Emergency Preference Scheme (IEPS) for disaster relief operations.*
- ITU-T Recommendation H.225.0 (2003), *Call signalling protocols and media stream packetization for packet-based multimedia communication systems.*
- ITU-T Recommendation H.235.0 (2005), *H.323 Security: Framework for security in H-series (H.323 and other H.245-based) multimedia systems.*
- ITU-T Recommendation H.245 (2005), *Control protocol for multimedia communication.*
- ITU-T Recommendation H.246 (1998), *Interworking of H-series multimedia terminals with H-series multimedia terminals and voice/voiceband terminals on GSTN and ISDN.*
- ITU-T Recommendation H.248.4 (2000), *Gateway control protocol: Transport over Stream Control Transmission Protocol (SCTP), plus Corrigendum 1 (2004).*

- ITU-T Recommendation H.248.5 (2000), *Gateway control protocol: Transport over ATM.*
- ITU-T Recommendation H.248.8 (2005), *Gateway control protocol: Error code and service change reason description.*
- ITU-T Recommendation H.248.14 (2002), *Gateway control protocol: Inactivity timer package.*
- ITU-T Recommendation H.323 (2003), *Packet-based multimedia communications systems.*
- ITU-T Recommendation I.363.1 (1996), *B-ISDN ATM Adaptation Layer specification: Type 1 AAL.*
- ITU-T Recommendation I.363.2 (2000), *B-ISDN ATM Adaptation Layer specification: Type 2 AAL.*
- ITU-T Recommendation I.363.5 (1996), *B-ISDN ATM Adaptation Layer specification: Type 5 AAL.*
- ITU-T Recommendation I.366.1 (1998), *Segmentation and Reassembly Service Specific Convergence Sublayer for the AAL type 2.*
- ITU-T Recommendation I.366.2 (2000), *AAL type 2 service specific convergence sublayer for narrow-band services*, plus Corrigendum 1 (2002).
- ITU-T Recommendation I.371 (2004), *Traffic control and congestion control in B-ISDN.*
- ITU-T Recommendation Q.763 (1999), *Signalling System No. 7 – ISDN user part formats and codes*, plus Amendment 3 (2004).
- ITU-T Recommendation Q.765.5 (2004), *Signalling System No. 7 – Application transport mechanism: Bearer Independent Call Control (BICC).*
- ITU-T Recommendation Q.931 (1998), *ISDN user-network interface layer 3 specification for basic call control*, plus Amendment 1 (2002): *Extensions for the support of digital multiplexing equipment.*
- ITU-T Recommendation Q.2630.1 (1999), *AAL type 2 signalling protocol – Capability Set 1.*
- ITU-T Recommendation Q.2931 (1995), *Digital subscriber signalling system No. 2 – User-Network Interface (UNI) layer 3 specification for basic call/connection control*, plus Amendment 4 (1999).
- ITU-T Recommendation Q.2941.1 (1997), *Digital subscriber signalling system No. 2 – Generic identifier transport.*
- ITU-T Recommendation Q.2961.1 (1995), *Digital subscriber signalling system No. 2 – Additional traffic parameters: Additional signalling capabilities to support traffic parameters for the tagging option and the sustainable call rate parameter set.*
- ITU-T Recommendation Q.2961.2 (1997), *Digital subscriber signalling system No. 2 – Additional traffic parameters: Support of ATM transfer capability in the broadband bearer capability information element*, plus Corrigendum 1 (1999).
- ITU-T Recommendation Q.2965.1 (1999), *Digital subscriber signalling system No. 2 – Support of Quality of Service classes*, plus Amendment 1 (2000).
- ITU-T Recommendation Q.2965.2 (1999), *Digital subscriber signalling system No. 2 – Signalling of individual Quality of Service parameters.*
- ITU-T Recommendation V.76 (1996), *Generic multiplexer using V.42 LAPM-based procedures*, plus Corrigendum 1 (2005).

- ITU-T Recommendation X.213 (2001) | ISO/IEC 8348:2002, *Information technology – Open Systems Interconnection – Network service definition.*
- ITU-T Recommendation X.680 (2002) | ISO/IEC 8824-1:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*, plus Amendment 2 (2004): *Alignment with changes made to Rec. X.660 | ISO/IEC 9834-1 for identifiers in object identifier value notation.*
- ITU-T Recommendation X.690 (2002) | ISO/IEC 8825-1:2002, *Information Technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*, plus Amendment 1 (2003): *Support for EXTENDED-XER.*
- ISO/IEC 10646 (2003), *Information technology – Universal Multiple-Octet Coded character Set (UCS).*
- ATM Forum (1996), *ATM User-Network Interface (UNI) Signalling Specification – Version 4.0.*
- IETF RFC 1006 (1987), *ISO Transport Service on top of the TCP, Version 3.*
- IETF RFC 2234 (1997), *Augmented BNF for Syntax Specifications: ABNF.*
- IETF RFC 2327 (1998), *SDP: Session Description Protocol.*
- IETF RFC 2402 (1998), *IP Authentication Header.*
- IETF RFC 2406 (1998), *IP Encapsulating Security Payload (ESP).*

2.2 资料性参考文献

- ITU-T Recommendation E.180/Q.35 (1998), *Technical characteristics of tones for the telephone service.*
- ITU-T Recommendation G.711 (1988), *Pulse code modulation (PCM) of voice frequencies.*
- ITU-T Recommendation H.221 (2004), *Frame structure for a 64 to 1920 kbit/s channel in audiovisual teleservices.*
- ITU-T Recommendation H.223 (2001), *Multiplexing protocol for low bit rate multimedia communication.*
- ITU-T Recommendation H.226 (1998), *Channel aggregation protocol for multilink operation on circuit-switched networks.*
- ITU-T Recommendation Q.724 (1998), *Telephone user part signalling procedures*, plus Amendment 1 (1993).
- ITU-T Recommendation Q.764 (1999), *Signalling System No. 7 – ISDN user part signalling procedures*, plus Amendment 3 (2004).
- ITU-T Recommendation Q.1902.4 (2001), *Bearer Independent Call Control protocol – (Capability Set 2): Basic call procedures*, plus Amendment 2 (2004).
- IETF RFC 768 (1980), *User Datagram Protocol.*
- IETF RFC 791 (1981), *Internet protocol.*
- IETF RFC 793 (1981), *Transmission control protocol.*
- IETF RFC 1661 (1994), *The Point-to-Point Protocol (PPP).*

- IETF RFC 2401 (1998), *Security Architecture for the Internet Protocol*.
- IETF RFC 2460 (1998), *Internet Protocol, Version 6 (IPv6) Specification*.
- IETF RFC 2805 (2000), *Media Gateway Control Protocol Architecture and Requirements*.
- IETF RFC 3261 (2002), *SIP: Session Initiation Protocol*.
- IETF RFC 3550 (2003), *RTP: A Transport Protocol for Real-Time Applications*.
- IETF RFC 3551 (2003), *RTP Profile for Audio and Video Conferences with Minimal Control*.

3 定义

本建议书规定下列术语：

- 3.1 access gateway 接入网关：** 提供用户网络接口（如ISDN等）的一种类型网关。
- 3.2 descriptor 描述符：** 协议中用于囊括相关属性的。例如：通过在一个命令中添加恰当的描述符，媒体控制器可以用于设置媒体网关中的媒体流属性。
- 3.3 Media Gateway (MG) 媒体网关（MG）：** 媒体网关可以将一种网络类型的媒体转换成另一种网络类型所要求的媒体格式，例如：媒体网关可以终结电路交换网络中的承载通道（如DS0 64K电路）和分组网络中的媒体流（如IP 网中的RTP媒体流）。媒体网关具备单独处理音频、视频或者T.120和以上三种媒体之间的任意组合的能力，媒体网关还具备全双工的媒体转换能力。媒体网关还支持视频/音频播放功能以及其他IVR功能，或支持媒体会议功能。
- 3.4 Media Gateway Controller (MGC) 媒体网关控制器（MGC）：** 对媒体网关中与媒体通道的连接控制相关的呼叫状态进行控制。
- 3.5 Multipoint Control Unit (MCU) 多点控制单元（MCU）：** 用于控制包含音频、视频和数据的多个用户会议的建立和和协调的功能实体。
- 3.6 residential gateway 驻地网关：** 用于将模拟线路连接到分组网络中的网关。一个驻地网关通常包含一个或多个模拟线路接口，并且物理上位于用户网络内。
- 3.7 SCN FAS signalling gateway 电路交换网络FAS信令网关：** 包含电路交换网络的信令接口用于终结7号信令、ISDN或其他呼叫控制和媒体通道共存的信令链路。
- 3.8 SCN NFAS signalling gateway 电路交换网络NFAS信令网关：** 包含电路交换网络的信令接口用于终结7号信令或其他呼叫控制和媒体通道分离的信令链路。
- 3.9 stream 流：** 在一个呼叫或会议中由媒体网关接收或发送的双向媒体流或控制流。
- 3.10 trunk 中继：** 位于两个电路交换系统之间的媒体通道，如T1或E1接口上的一条DS0电路。
- 3.11 trunking gateway 中继网关：** 位于电路交换网络和分组网络之间用来终结大量数字电路的网关。

4 缩写

本建议书采用下列缩写：

AAD	平均确认时延
AAL	ATM 适配层
ADEV	平均偏移
ALF	应用层成帧协议
ATM	异步传输模式
C	关联
CAS	随路信令系统
DNS	域名系统
DTMF	双音多频
FAS	设施有关信令系统
GSM	全球移动通信系统
GW	网关
IANA	互联网号码分配机构（被ICANN取代）
ICANN	互联网域名与地址管理机构
IEPS	国际紧急优选方案
IP	网际协议
IS	服务状态
ISUP	ISDN用户部分
IVR	交互式语音应答
MG	媒体网关
MGC	媒体网关控制器
MWD	最大等待时延
NFAS	设施无关信令系统
OoS	退出服务
PRI	一次群接口
PSTN	公众交换电路网
QoS	服务质量
RTP	实时传送协议
SC	电路交换网络
SCN	电路交换网络
SG	信令网关
SS7	7号信令系统
T, Term	终结点

5 惯例

本建议书中，“须（Shall）”指强制性要求，而“应（Should）”指推荐采取的非强制性的特性或程序。“可（May）”一词指非强制性措施，但并未推荐采取这种措施。

6 连接模型

连接模型用于描述媒体网关中可被媒体网关控制器所控制的逻辑单元或实体。终结点（Termination）和关联（Context）是连接模型所使用的两个主要抽象对象。

一个终结点用于发起或接收一个或多个媒体流。在多媒体会议中，一个终结点可以支持多媒体，用于发送或者接收多个媒体流。终结点封装了媒体流参数和媒体通道参数。

一个关联是一组终结点之间的联系。空关联是一类特殊的关联，用于包含所有未处于任何关联之内且未与任何其他终结点发生联系的终结点。例如，在分离后的接入网关中，所有空闲的模拟线将被放置于空关联之中。

终结点和关联的关系如图1所示。图1列举的几个示例无法概括所有类型的关联。每个关联中“*”符号用于表示终结点之间的逻辑联系。

媒体网关

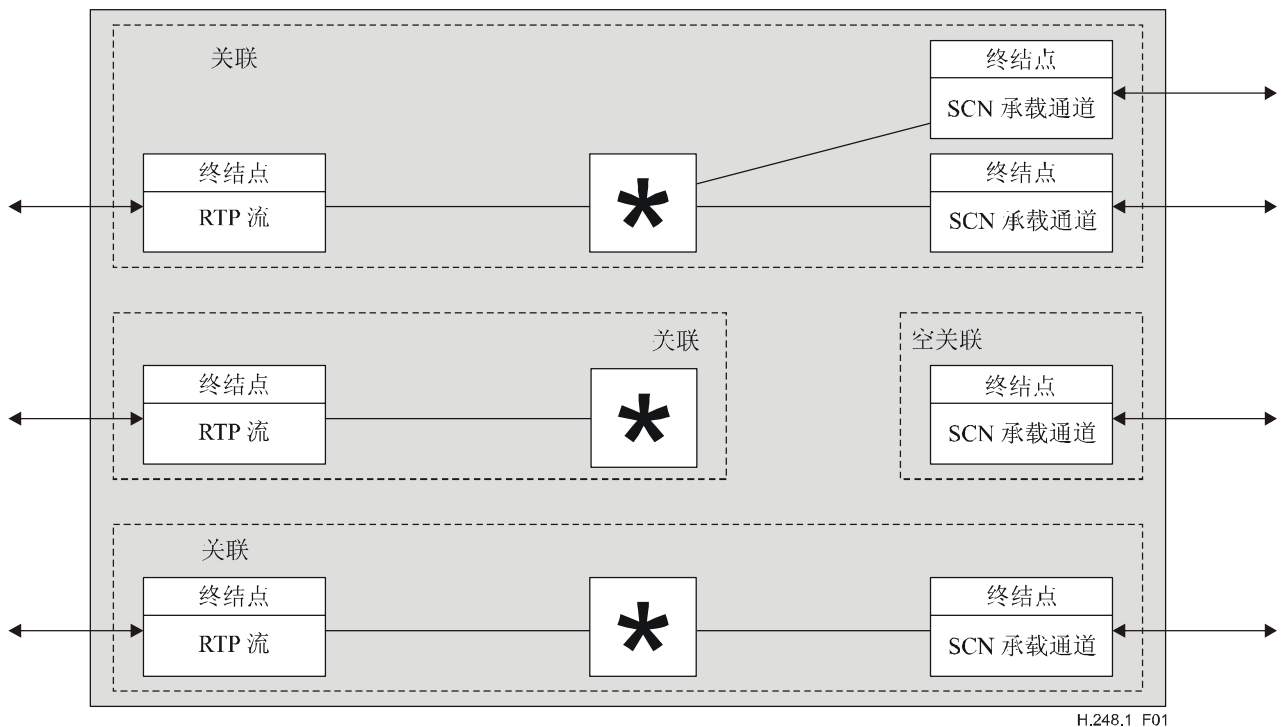


图 1/H.248.1 – H.248.1 连接模型示例

图2为分离的接入网关实现呼叫等待业务的连接模型示意图，该示意图描述了终结点在关联中被重新分配的过程。在一次双向语音呼叫中，T1和T2终结点被包含在关联C1之中。T1终结点正等待来自T3终结点的第二个呼叫。此时，T3独自处于关联C2之中。当T1接收来自T3的呼叫时，T2终结点将处于呼叫保持状态。从而，T1终结点将转移至关联C2之中，如图3所示。

媒体网关

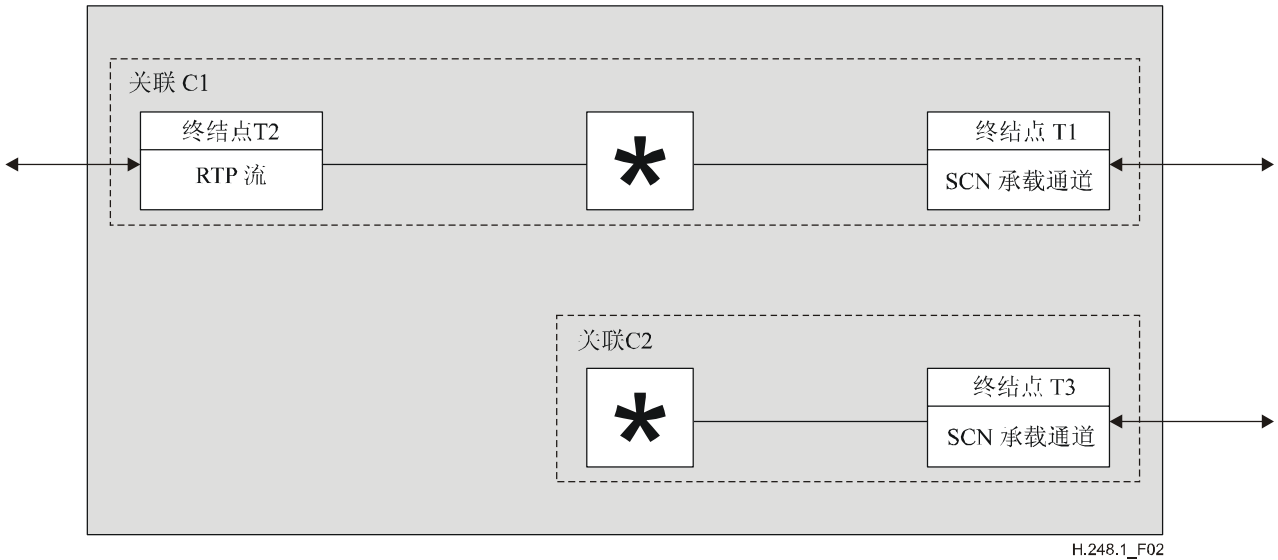


图 2/H.248.1 –呼叫等待模型/T1终结点振铃

媒体网关

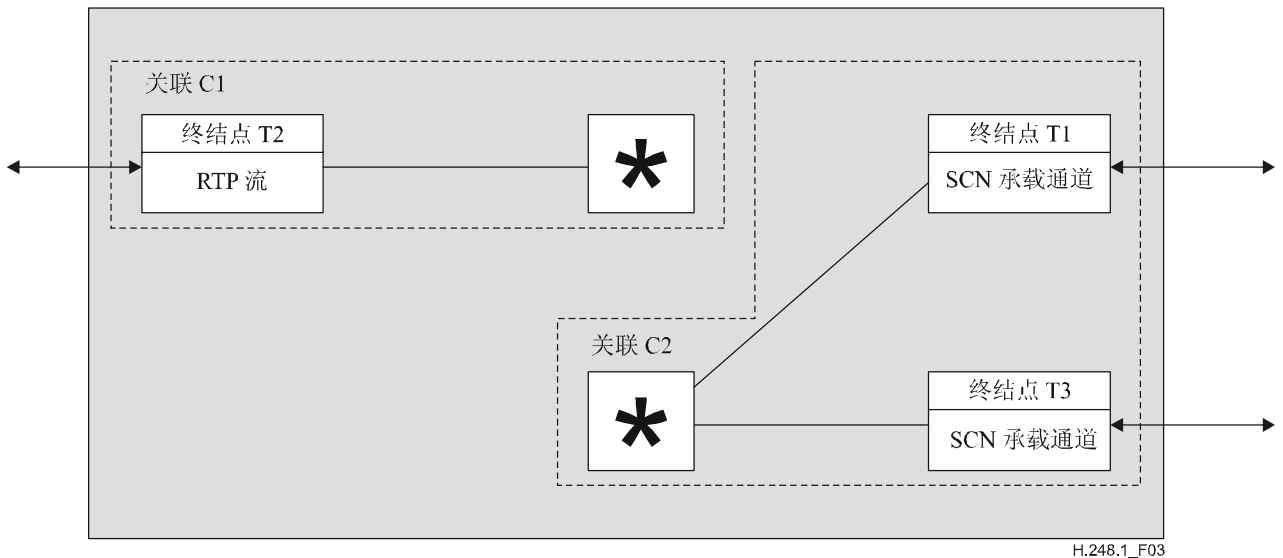


图3 /H.248.1 –呼叫等待模型/ T1终结点应答

6.1 关联

一个关联用于描述一组终结点之间的联系。如果一个关联包含两个以上的终结时点时，则关联用于描述终结点之间的拓扑（即某个终结点接收或发送媒体）和媒体混合或交换参数。

空关联是一类特殊的关联，用于包含所有未被包含于任何关联之内且未与任何其他终结点发生联系的终结点。处于空关联中的终结点参数可以被审计或修改，此外，空关联中终结点上发生的事件还可以被检测。

通常，Add命令用于向关联添加终结点。如果媒体网关控制器未指定一个已存在的关联用于添加终结点，则媒体网关将创建一个新的关联。Subtract命令可以将一个终结点从一个关联中删除，Move命令可以将一个终结点从一个关联转移至另一个关联中。一个终结点必须只能存在于一个关联内。

一个关联中所允许支持的终结点最大数目由媒体网关的属性决定。对于仅支持点对点连接的媒体网关，每个关联最多支持两个终结点。对于支持多点会议的媒体网关，每个关联可以支持三个或三个以上的终结点。

6.1.1 关联特征和描述符

关联特征可以由如下参数进行描述：

- 关联标识符（ContextID）。
- 拓扑（Topology）描述符（即某个终结点接收或发送媒体）。

关联的拓扑用于描述关联中终结点之间的媒体流方向。相比而言，终结点的模式属性（“仅发送”（SendOnly）或“仅接受”（ReceiveOnly）等）用于描述媒体网关的入口和出口处的媒体流流向。

- 优先级用于向媒体网关指示应使用某个特定的流程对某个关联进行处理。在某种情形下（如重新启动时），当媒体网关必须同时处理大量关联时，媒体网关控制器可以利用关联的优先级实现对不同优先级的业务流量进行平滑处理。优先级“0”为最低优先级，优先级“15”为最高优先级。
- 紧急呼叫标识符用于向媒体网关提供一种优先处理机制。
- IEPS 呼叫标识符用于实现E.106建议书中规定的业务和技术细节。
- 关联特征(ContextAttribute)描述符通过7.1.19所定义的包扩展机制来扩展关联特性。

6.1.2 创建、删除和修改关联

本协议可以用来创建和修改已存在的关联参数。本协议定义了若干命令用于向关联添加、从关联中删除终结点和在关联之间转移终结点。当关联中最后一个终结点被删除或被转移时，隐含地，该关联也就被删除了。

6.2 终结点

终结点是媒体网关中的一个逻辑实体，它用于发起和/或接收媒体流和/或控制流。终结点可用一组特性进行描述，特性可以隶属于一系列作为命令的输入和输出参数的描述符。当媒体网关创建终结点时，将赋予终结点一个唯一的终结点标识符（TerminationID）。

代表物理实体的终结点具有半永久性特性。例如，一个标识TDM通路的终结点将永远存在，只要媒体网关中存在该通路。代表临时媒体流的终结点（如RTP流），通常只能生存一段时间。

临时终结点可由Add命令创建，以及由 Subtract命令进行删除。相比而言，当使用Add命令向一个关联添加一个物理终结点时，该物理终结点将来自空关联中，当使用Subtract命令从一个关联中删除物理终结点时，该物理终结点将返回至空关联中。

终结点可以用于播放信号音（参见7.1.11）。且终结点还可以用于事件检测，一旦检测到事件发生，媒体网关可以将通知消息或由其执行的相应动作上报给媒体网关控制器。终结点上的统计参数可以进行累计。当接受来自媒体网关控制器的请求消息时（例如AuditValue命令，参见7.2.5），以及当终结点不复存在或当Subtract命令将终结点从一个关联返回至空关联时，统计参数可以通过请求响应上报给媒体网关控制器。

多媒体网关可以处理复用的媒体流。例如，ITU-T H.221建议书描述了一种帧结构用于在一组64kbit/s数字通路上进行多个媒体流的复用。连接模型中处理复用媒体流的方式如下所描述。对于承载部分复用流的每个数字通路，将有一个物理或临时的终结点与之对应。所有这些用于发起或接收数字承载通道的终结点将连接到一个被称为“复用终结点”的终结点。复用终结点为一类临时的性终结点，用于描述面向帧格式的会话。复用终结点可使用复用描述符来描述所采用的复用类型（如H.320会话采用H.221复用方式）以及所有被复用的数字通路组合成帧的顺序。

复用终结点可采用多级复用类型（例如，为了支持H.324会话，可使用H.226复用数字同事类型适配成H.223复用类型）。

复用终结点可用流描述符（Stream）来描述会话中所承载的每个媒体流。这些媒体流与关联中由终结点发起或接收的媒体流有关，而与进行复用的承载终结点无关。每一个承载终结点仅支持单个数据流。这些数据流在复用终结点上是不可见，而是隐含在关联中。

图4、图5和图6是复用终结点和复用描述符的典型应用示例。

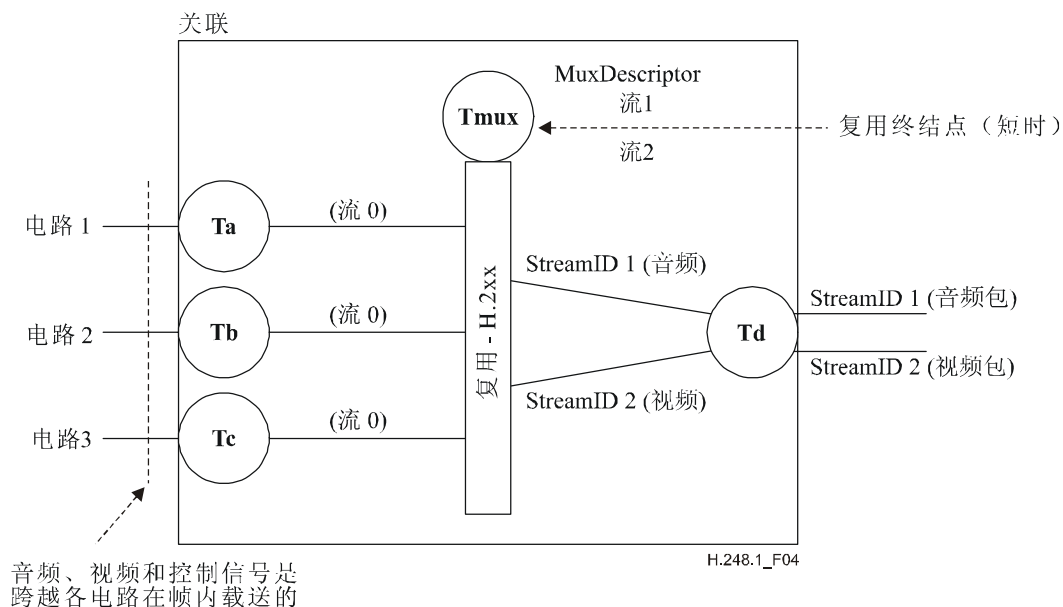


图 4/H.248.1 – 复用终结点（电路侧一分组侧）

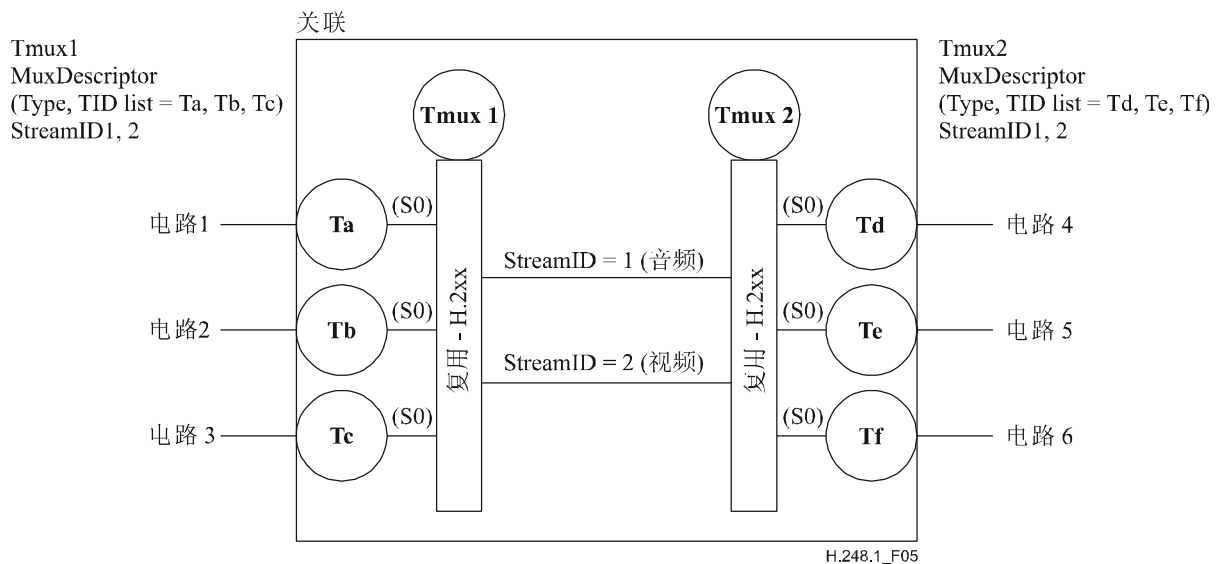


图 5/H.248.1 – 复用终结点（电路—电路终结点）

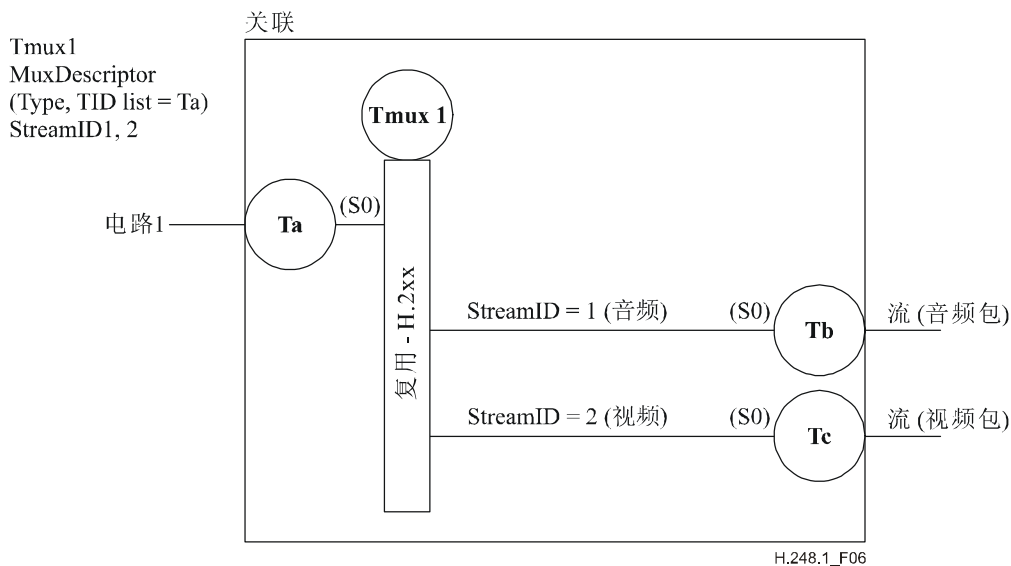


图 6/H.248.1 – 复用终结点（单一多终结点）

与前一段落所描述的复用终结点不同的是，如果被复用的承载终结点是诸如ATM AAL2类型的复用承载通道，则被复用的终结点将不能承载媒体流。此类复用的承载终结点将严格地用于创建和删除真实的承载类型。当一个新的复用承载被创建时，则相应地在关联中创建一个临时性终结点。当该终结点被删除时，则被复用的承载类型也应被删除。

6.2.1 动态的终结点

本协议可以用于创建新的终结点或修改已存在的终结点的属性。修改的方式可以是增加或删除事件或信号。终结点属性、事件和信号将在本建议书的后续各小节进行描述。媒体网关控制器可以仅仅释放或修改处于空关联中或通过Add命令刚被创建的终结点的相关资源。

6.2.2 TerminationID

终结点可用终结点标识符（TerminationID）进行标识，其中终结点描述符可由媒体网关选用任意机制来进行命名。

媒体网关对物理终结点的终结点描述符进行分配。终结点描述符可选用数据结构进行定义。例如，终结点描述符可以由中继群组和中继来进行标识。

终结点标识符可使用两种类型的通配符，即“ALL”和“CHOOSE”。“ALL”用于一次选择多个终结点，而“CHOOSE”则用来指示媒体网关必须选择一个符合条件的终结点。例如，媒体网关控制器可以指示媒体网关选择一个中继群组中的一条中继电路。

终结点标识符可使用列表来进行描述。终结点标识符列表（TerminationIDList）被推荐适用于当不希望对每个终结点发送单个命令和垂直终结点标识符不被允许时。

6.2.3 包

不同类型的网关可以处理特性相差甚远的不同类型的终结点。通过允许终结点被媒体网关配置可选的属性、事件、信号和统计参数，可以实现终结点的多样性。

为了实现媒体网关和媒体网关控制器之间的互操作，这些可选参数被组合成包(Packages)，通常一个终结点可以实现一个包集合。包的详细定义参见本建议书第12节。媒体网关控制器可以通过审计命令（Audit）来查询终结点支持的包类型。

由包所定义的特性、事件、信号和统计及其参数可以分别用标识符进行标识。标识符具备特定的有效范围。对每个包而言，属性标识符、事件标识符、信号标识符、统计标识符和参数标识符都有独立的命名空间，所有标识符可使用命名相同的标识符。由不同类型包所定义的两个属性标识符也使用相同的标识符。

为了实现某种类型的包，媒体网关必须支持此包中定义的所有属性、事件、信号、统计。同时，媒体网关必须支持所有信号和事件所定义的参数。如果媒体网关无法支持某个属性、事件、信号、统计所对应的功能，则媒体网关应返回“句法差错”或“无法识别的标识符”差错代码。

对一个特定的属性或参数而言，媒体网关可以仅支持某一个包中所定义的参数值集的一个子集。如果媒体网关控制器指定了一个媒体网关所不支持的参数值，则媒体网关应返回501差错代码“无法实施”。

当包被扩展时，基本包中定义的属性、事件、信号和统计可以通过扩展包名称或基本包名称进行引用。例如，A包定义了事件e1，B包是A包的扩展包，那么B/e1是适用于支持B包的终结点上的一个事件。根据规定，实现扩展包的媒体网关必须支持基本包，然而是否公开基本包接口为可选。如果媒体网关选择公开A包，当媒体网关接收到AuditValue请求中携带包描述符（Package）时，媒体网关应将A包和B包同时上报给媒体网关控制器，同理，A/e1也是终结点上的一个有效的事件。如果媒体网关选择不公开A包，则仅有B/e1是有效的事件。如果通过AuditValue消息进行公布，则A/e1和B/e1为相同事件。

为了保证互通和后向兼容性，除已被注明“仅用于扩展”的那些基本包之外，媒体网关可以公开其所包含的终结点所支持的所有基本包和扩展包。但是有一个例外，就是不能公开那些仅仅定义未“仅用于扩展”的基本包。

6.2.4 终结点属性和描述符

终结点可以具备属性。每个属性采用一个唯一的属性标识符进行标识。大多数属性参数是有默认值的，其默认值可由本建议书或包（参见第12节）进行规定，或者通过配置进行设置。对于无默认值的属性，当一个终结点被第一次创建和返回至空关联时，除了终结点状态（TerminationState）描述符和本地控制（LocalControl）描述符之外，其他所有描述符中的属性值应设置为“空”或“无参数值”。当一个终结点被第一次创建和返回至空关联时，终结点可以表示为处于空闲状态的模拟线，中继或其他物理实体。“终结点状态（TerminationState）”描述符和“本地控制（LocalControl）”描述符定义参见7.1.5和7.1.7。

对媒体网关中的属性值进行配置可以替换本建议书或包中所定义的属性默认值。因而，如果媒体网关控制器期望完全控制终结点的属性值，则媒体网关控制器向一个关联添加一个终结点时，应该明确规定所添加的终结点属性值。媒体网关通过对空关联中的终结点进行审计，可以获知终结点的任意一个配置后的属性值。

终结点具有一些公共属性，以及与媒体流相关的属性。公共属性也称为终结点状态(TerminationState)属性。对于每一个媒体流，其包括本地（Local）属性和接收和/或发送流属性。

本建议书中未包含的属性可由包进行定义。属性可以通过包名称和属性标识符进行标识。大多数属性具备默认值，且由包进行定义。属性值可以分为“只读”和“可读/可写”两类。属性的所有可能值以及当前值可以通过审计获知。对于那些具备“可读可写”的属性，媒体网关控制器可以设置其参数值。如果实现某一个包的所有终结点共享一个属性参数值，则该属性被称为“全局参数”。为了方便起见，某些相关的属性可以被组合成描述符。

当使用Add命令将一个终结点添加到一个关联时，可添加适当的描述符作为Add命令的输入参数来设置具备“可读可写”属性的参数值。同理，某个关联中终结点的属性值可以通过Modify命令进行修改。当Move命令将一个终结点从一个关联转移至另一个关联时，可以导致终结点的属性值发生改变。当以上情况发生时，命令的输出参数通常包含相应的描述符。

如果对同一个关联中的不同终结点设置不同的属性值，则隐含的表示媒体网关应执行某种特定的功能。例如，终结点A和终结点B 分别设置为支持G.711和G.729语音编码，则一旦两个终结点之间有媒体流进行传输，媒体网关应激活语音编码转换功能（例如，每一个终结点上应设置模式（mode）属性为不同于“非活动（inactive）”状态的其他状态。

注 — 为了避免媒体网关资源的不必要激活，媒体网关控制器应对某一指定的终结点和媒体流的模式属性设置为“非活动”，直至该媒体流将被使用时。

通常，如果某个描述符在命令被省略，则执行该命令的终结点中描述符的属性值将保持不变。另一方面，如果某个可读可写的属性被命令所省略（例如，描述符仅被部分指定），如果包未定义其他操作方式，则执行该命令的终结点的属性值将被恢复为默认值。处理单个描述符的细节参见7.1。

以上规定同样适用于信号和时间以及其相关的参数。配置事件描述符时应包含一个请求标识符（RequestID）和事件所包含的任意参数。为了避免来自媒体网关的错误的Notify命令响应，媒体网关控制器应对任何与配置媒体网关中事件描述符的请求描述符进行复制。

下列表格列举了本建议书所定义的所有描述符和其用法。以下表格中并非所有描述符可作为同一个命令的输入和输出参数。

描述符名称	说 明
调制解调 (Modem)	标识所使用的调制解调器类型和属性 (注)
复用 (Mux)	描述多媒体终结点的复用类型(如H.221、H.223和H.225.0)和组成复用终结点的终结点
媒体 (Media)	媒体流的列表 (参见7.1.4)
终结点状态 (TerminationState)	包中所定义的与特定媒体流无关的终结点属性
流 (Stream)	单个媒体流的远端/本地/本地控制描述符的列表
本地 (Local)	包含媒体网关从远端实体接收到的媒体流属性
远端 (Remote)	包含媒体网关发送至远端实体的媒体流属性
本地控制 (LocalControl)	包中所定义的包含与媒体网关和媒体网关控制器有关的一些属性。
事件 (Events)	描述可以被媒体网关检测的事件, 以及当事件被检测发生时的处理机制。
事件缓存 (EventBuffer)	描述当事件缓存处于激活状态时, 可以被媒体网关所检测的事件
信号 (Signals)	描述适用于终结点的信号 (参见7.1.11)
审计 (Audit)	适用于Audit命令, 描述需要审计的信息
包 (Packages)	适用于AuditValue命令, 返回由终结点实现的包的列表
数字映射 (DigitMap)	定义一组特定的事件被匹配的模式, 使得事件可以按组而不是单个上报
业务改变 (ServiceChange)	适用于业务改变 (ServiceChange) 命令, 描述何种业务发生改变以及业务发生改变的原因
被观察事件 (ObservedEvents)	适用于Notify 或者 AuditValue命令, 上报已检测到的事件
统计值 (Statistics)	适用于Subtract、Audit命令, 上报与终结点有关的统计数据
拓扑 (Topology)	描述关联中终结点之间的媒体流流向
关联特征 (ContextAttribute)	包中所定义可以影响整个关联的属性
差错 (Error)	包含差错代码和可选的差错描述文本, 适用于Notify请求或所有命令响应
注 — ITU-T H.248.1建议书第2版 (05/2002) 不再推荐使用 Modem描述符。	

6.2.5 根终结点

命令有时将需要作用于整个媒体网关, 而不是仅仅作用于其中的某个终结点。为了实现上述目标, 本建议书定义了一类特殊的终结点标识符即根终结点 (Root)。根终结点可以指定所支持的包类型。根终结点也可以有属性、事件、信号和统计参数。相应地, 根终结点可适用于以下命令中:

- **Modify命令:** 用来修改属性、发送信号或设置需要检测的事件。
- **Notify命令:** 用于上报已检测到的事件。
- **AuditValue命令响应:** 用于审计根终结点上实现的属性值和统计参数。
- **AuditCapabilities命令:** 用于审计根终结点已经具备的属性。
- **ServiceChange命令:** 用于声明网关处于“服务”状态或“退出服务”状态。

除上述命令外，其他命令不允许使用根终结点标识符。当其他命令使用根终结点标识符时，必须返回差错代码410（非法的标识符非法）。

6.3 通配原则

本节内容所定义将对关联标识符和终结点标识符通配原则将适用于本建议书所规定的所有命令。命令必须支持以下两种类型的通配方式：

- 1) 关联通配；
- 2) 终结点通配。

为了实现通配，包含多个终结点标识符（TerminationID）的终结点标识符列表（TerminationIDList）被看作为通配的终结点标识符。当执行一个事务中包含通配的关联和可选通配的终结点时，某一个特定关联标识符中所关联的所有命令将在下一个关联标识符之前被顺序执行。在某种情况下，一个事务可能包含有多个命令，仅仅当符合通配条件的关联中的第一个命令中有一个终结点标识符（TerminationID可以是通配或确定）满足终结点通配条件时，该事务中的后续命令才可以被处理。如果该关联中后续的命令中无终结点标识符（通配或确定）满足终结点通配条件时，应返回差错代码431（无终结点标识符满足通配条件），并且如果产生差错代码的命令没有标记为可选，则后续与通配的关联标识符相关的处理将被停止。

特定通配组合的执行将在下文介绍。

6.3.1 明确的ContextID和通配的TerminationID

当关联标识符是明确的，而终结点标识符为通配符“ALL”时，等同于向满足此通配条件的所有终结点执行该命令。通配符“ALL”并不等同于根终结点。由于每一条命令消息与响应消息相对应，因此所有响应的数目将变得非常庞大。因此，如果关联中有多个终结点满足通配条件时，应尝试所有可能的终结点，并且每个终结点将返回一条响应消息。如果指定的关联标识符未包含有满足通配条件的终结点时，则应返回差错代码431（无终结点标识符满足通配条件）。未包含于指定关联标识符中的终结点不需要返回相应的差错代码。

例如：假定一个网关包含有4个终结点t1/1, t1/2, t2/1和 t2/2。假定关联1 包含终结点t1/1 和t2/1，关联2 包含终结点t1/2 和t2/2。

命令：

```
Context=1{Command=t1/*{Descriptor/s}}
```

响应：

```
Context=1{Command=t1/1{Descriptor/s}}
```

6.3.2 通配关联标识符“ALL”和明确指定的终结点标识符

当关联标识符采用通配符时（例如，关联标识符等于ALL），且终结点标识符为完全确定时，等同于向包含该确定终结点标识符的所有非空的关联执行命令。此时，务必查询所有满足通配条件的关联，且同一时刻只允许执行一个命令。未包含确定终结点的关联不需要返回差错响应。如果某个终结点未包含在任意一个非空的关联中，则返回431差错代码（无终结点标识符满足通配条件）。如果除了空关联之外，无任何其他的关联，则返回差错代码411（事务引用无法识别的关联标识符）。本建议书不推荐使用这类通配的关联标识符，然而，在解决诸如媒体网关和媒体网关控制器之间的冲突时，通配的关联标识符是非常有用的。

例如：网关配置参见6.3.1，命令：

```
Context={Command=t1/1{Descriptor/s}}
```

响应：

```
Context=1{Command=t1/1{Descriptor/s}}
```

6.3.3 关联标识符（ALL）和通配的终结点标识符

当关联标识符采用通配符时（例如，关联标识符等于“ALL”）且终结点标识符也采用通配符时，等同于向满足通配条件的每一个终结点标识符所在的所有非空关联执行该命令。因此，如果在满足通配条件的某个关联内有多个满足通配条件的终结点时，则应尝试所有的终结点，并且每一个终结点将返回响应消息。如果某个关联满足通配条件但关联中未包含满足通配条件的单个终结点时，不应上报差错。如果同时无法满足通配关联和通配终结点时，则返回差错代码431（无终结点标识符满足通配条件）。

例如：网关配置参见6.3.1，命令：

```
Context={Command=t1/*{Descriptor/s}}
```

响应：

```
Context=1{Command=t1/1{Descriptor/s}}
```

```
Context=2 {Command=t1/2{Descriptor/s}}
```

当同时有多个命令包含通配的终结点标识符和/或通配的关联标识符时，此时如果第一个命令无法与第一个终结点标识符和关联标识符相匹配时，则所有后续的命令不应被继续执行。

6.3.4 通配响应

当不要求使用单个响应时，可允许使用通配响应。通配响应是包含所有单个响应且省略每个响应中冗余的参数值的响应集合，如果不采用通配响应，则相应地应发送每一个单个响应。假定终结点Ta具有属性P₁等于“a”和P₂等于“b”以及终结点Ta具有属性P₂等于“c”和P₃等于“d”，则响应集合将包含一个通配的终结点标识符和一个属性序集合（P₁ = a, P₂ = b, c, P₃ = d）。通配响应对于Audit命令的作用尤其明显。如果一个通配响应与通配的关联标识符同时使用，则应返回一个通配响应，且该通配响应应当包含一个所有满足通配条件的终结点集合。通配响应可以包含等于“ALL”的关联标识符，通配的终结点标识符和一序列的属性值。

如果在执行通配请求过程中发生差错，且该请求指定了使用通配响应机制，则应采用某种特殊处理机制来提供差错信息，且仍然将响应数目保持在一个正常范围内。当使用上面描述的通配响应机制时，如果在执行命令的过程中即使发生一个或多个差错，该命令必须被继续执行，然而该事务中后续的命令将不允许被执行（除非该发生差错的命令被标志为可选项）。发生差错的命令应当返回多个命令响应。其中，第一个命令响应必须是普通的通配响应，且该通配响应包含所有被成功执行的命令响应集合。如果无命令被成功执行，则响应集合必须为空。且必须为每个执行失败的事务标识符返回包含正确的差错描述符的附加命令响应。

例如，命令：

```
Context={Command=t1/*{Descriptor/s}}
```

差错响应:

Context={Command=t1/*{Union response descriptors},

Command=t1/3{Error=errorcode}}

通配符编码规则参见本建议书附件A和附件B。

7 命令

本建议书定义了命令用于对连接模型中的逻辑实体（包括关联和终结点）进行操作。命令可以达到对关联和终结点实现完全控制的目的。例如，命令可用于向一个关联添加、修改和删除终结点，以及审计关联或终结点的属性。命令可以实现对关联或终结点属性的完全控制。这些控制包括指定终结点需要上报的事件、需要向终结点播放的信号音和执行的的操作，以及确定关联的拓扑关系（终结点之间的流向）。

大多数命令都是由命令网关控制器发起，而媒体网关作为命令响应方。然而，通知（Notify）和业务改变（ServiceChange）命令是两个例外：通知命令由媒体网关发送至媒体网关控制器，而业务改变命令可以由媒体网关控制器或者媒体网关任意一方发起。本建议书定义的命令概要如下所示，有关命令的详细内容参见7.2。

- 1) **添加（Add）**：Add 命令用于向一个关联添加一个终结点。当使用 Add 命令向一个关联添加第一个终结点时，等同于创建了一个新的关联。
- 2) **修改（Modify）**：Modify 命令用于修改终结点的属性、事件和信号。
- 3) **删除（Subtract）**：Subtract 命令用于从一个关联中删除一个终结点且返回关联中终结点的当前统计值。当 Subtract 命令将关联中的最后一个终结点删除时，等同于删除了该关联。
- 4) **转移（Move）**：Move 命令用于自动地将一个终结点从一个关联转移至另一个关联。
- 5) **审计值（AuditValue）**：AuditValue 命令用于获知终结点的当前属性、事件、信号和统计信息。
- 6) **审计能力（AuditCapabilities）**：Auditcapabilities 命令用于获知媒体网关所允许的终结点的属性、事件和信号的所有可能值。
- 7) **通知（Notify）**：媒体网关使用 Notify 命令将已检测到的事件上报给媒体网关控制器。
- 8) **业务改变（ServiceChange）**：媒体网关使用 ServiceChange 命令向媒体网关控制器上报一个终结点或者一组终结点即将退出服务或返回服务。媒体网关也可以使用 ServiceChange 命令向媒体网关控制器进行注册通报其可用性，以及向媒体网关控制器报告媒体网关即将开始或已完成重新启动流程。当媒体网关控制器发生切换时，媒体网关控制器可以通过 ServiceChange 命令告知媒体网关。媒体网关控制器也可以使用 ServiceChange 命令指示媒体网关将一个终结点或者一组终结点进入服务或退出服务。

命令的详细说明和使用规则参见本建议书的7.2.1至7.2.8。

7.1 描述符

描述符可作为命令的输入和输出参数。描述符由描述符名称和一系列参数组成。某些参数允许设置数值。许多命令使用相同的描述符。本节将罗列本建议书所定义的描述符。描述符可作为命令的返回参数。在所有返回的描述符中，任何空描述符将只使用描述符名称进行描述且不携带任何参数。对于某个特定命令所包含的参数及其用法将在描述该命令的小节中进行介绍。

7.1.1 参数相关规定

命令参数可以由一组描述符组成。通常，描述符的文本格式如下所描述：
DescriptorName=<someID>{parm=value, parm=value...}。

参数值类型可以是“完全指定”、“部分指定”和“多重指定”：

1) “完全指定 (Fully specified)”的参数具有唯一、确定的值，命令发起方指示命令响应方使用完全指定的该参数值。

2) “部分指定 (Underspecified)”的参数使用通配值“CHOOSE”，允许命令响应方选择任意一个所支持的参数值。

3) “多重指定 (Overspecified)”的参数具有多个可能的参数值列表。该列表中参数值的顺序指示了命令发起方对这些参数值选择的优选级。命令响应方可以从该列表中选择一个参数值返回给命令发起方。

除审计 (Audit) 描述符之外，如果命令中省略了某个其他类型的描述符 (例如，该描述符默认)，则终结点中该描述符的参数值将保持不变。除 Subtract 命令之外，如果任何其他命令未包含审计描述符则等同于该审计描述符视为空。如果某个描述符中省略了某个参数值，则媒体万贯于对于省略参数值的处理方式因描述符而异。如果某个参数为“部分指定”或“多重指定”类型时，命令响应方所选择的参数值应作为命令响应的输出参数包含在相关的描述符中。

每一个命令都要指定其所操作的终结点标识符。终结点标识符可使用通配符。当命令中所包含的终结点采用通配符时，必须对所有满足通配条件的终结点重复执行该命令。

7.1.2 调制解调描述符 (Modem)

Modem 描述符用于定义调制解调器的类型和参数，例如，在 H.324 协议和文本会话中所使用 Modem 类型和参数。Modem 描述符可以包含以下调制解调方式：V.18、V.22、V.22bis、V.32、V.32bis、V.34、V.90、V.91、同步 ISDN，以及其他相应的扩展。默认情况下，终结点将不使用 Modem 描述符。

2002年5月颁布的ITU-T H.248.1建议书及其后续版本不再推荐使用Modem描述符。因此，如果命令接收方接收到Modem描述符，必须认为 Modem描述符不是传送内容的一部分而将其忽略，或者必须将该描述符作为命令的可选项。包含数据流属性的Modem类型将在本地描述符和远端描述符进行描述。

7.1.3 Multiplex描述符

在多媒体呼叫中，媒体流在一组承载通路 (承载类型可以不同) 上进行传送。复用描述符将媒体流和承载类型进行关联。复用描述符可以包含以下复用类型：

- H.221;
- H.223;
- H.226;
- V.76;
- N × 64K;
- 其他扩展复用类型，

以及一组用于表示被复用承载通路的终结点标识符。例如：

$$\text{Mux} = \text{H.221}\{\text{MyT3/1/2}, \text{MyT3/2/13}, \text{MyT3/3/6}, \text{MyT3/21/22}\}$$

$N \times 64\text{K}$ 复用类型用于实现 $N \times 64\text{K}$ 业务（例如，Q.931建议书所定义的信息传输速率或Q.763建议书所规定的媒体传输要求）。从关联来看， $N \times 64\text{K}$ 复用终结点可以支持一个宽带数据流。当一个承载终结点被添加到一个关联中，则创建了一个 $N \times 64\text{K}$ 复用终结点，除承载终结点带宽必须为64 kbit/s之外，与承载终结点相关的流描述符中的参数值必须等同于复用终结点所定义的流描述符中的参数值。

7.1.4 媒体描述符 (Media)

Media描述符用于定义描述所有媒体流的参数值。这些参数值可以分为两类描述符：终结点状态描述符(TerminationState)和一个或若干个流描述符(Stream)，其中，终结点状态描述符与特定媒体流无关，每个流描述符用于描述一个媒体流。

媒体流可使用流标识符 (StreamID) 进行标识。流标识符必须是1至65 535之间的任意整数。流标识符用于连接关联中彼此相互连接的媒体流。终结点上的复用媒体流必须彼此同步。流描述符最多可以包含本地控制描述符、本地描述符、远端描述符和统计值描述符等四个子描述符，这几个描述符之间关系如下所示：

媒体描述符

TerminationState Descriptor

Stream

Descriptor

LocalControl Descriptor

Local Descriptor

Remote Descriptor

Statistics Descriptor

为简便起见，本地控制、本地、远端或统计值描述符可以由未包含流描述符的媒体描述符进行定义。当此情况发生时，流标识符假定为1。

7.1.5 终结点状态描述符 (TerminationState)

终结点状态描述符包括服务状态(ServiceStates)属性、事件缓存控制(EventBufferControl)属性以及由包所定义的与特定流无关的终结点属性。

服务状态属性可以用来描述与流无关的终结点全局状态。终结点状态可以分为以下三种：“检测(test)”，“退出服务(Out of Service)”和“服务(in Service)”。检测状态用于指示一个终结点正在处于被检测的状态。退出服务状态用于指示一个终结点无法再进行业务处理。服务状态用于指示一个终结点处于正常的业务处理状态。终结点状态描述符的默认值为“服务”。

这些属性的参数值可以为相对简单的数据类型（例如整型、字符型或枚举类型），也可以是“未指定”的参数值，此时，媒体网关可以参照以下规则从多个参数值中选取合适的参数值：

- 可选择的参数值：多个参数值列表，务必从其中选择一个参数值；
- 范围：最小和最大值，可以选择包括边界在内的最大值和最小值之间的任何数值；
- 大于或小于：所选择的数值必须大于或小于指定的参数值；
- CHOOSE 通配符：媒体网关必须选择满足通配条件的参数值。

事件缓存控制属性用于定义对事件描述符所检测到的事件应进行缓存，或立即处理。具体内容参见7.1.9。

7.1.5.1 终结点状态属性

7.1.5.1.1 服务状态 (ServiceStates)

属性名称: ServiceStates

描述: 该属性值用于指示终结点的当前所处的服务状态。

类型: 枚举

可能的取值:

InService: 终结点处于正常的服务状态。

OutOfService: 终结点处于退出服务状态，此时无法再处理业务。

Test: 终结点正处于被检测的状态。

默认值: InService

定义于: TerminationState

属性: 读/写

7.1.5.1.2 事件缓存控制 (EventBufferControl)

属性名称: EventBufferControl

描述: 用于定义对事件描述符所检测到的事件应进行缓存，或立即处理。参见7.1.9。

类型: 枚举

可能的取值:

LockStep: 事件应缓存处理，参见7.1.9。

OFF: 事件应立即处理。

默认值: OFF

定义于: TerminationState

属性: 读/写

7.1.6 流描述符 (Stream)

流描述符用于指定一个双向流的相关属性。这些属性可以被组合成三种描述符：一类是与流相关的终结点属性、以及本地和远端。流描述符可以使用流标识符 (StreamID) 来标识其所描述的媒体流。通过在一个关联中所包含的一个终结点上指定一个新的流标识符，可以创建一个新的流。当某个流所对应的本地和远端描述符被设置为空且先前支持该流的所有终结点上的本地控制描述符所包含 ReserveGroup和 ReserveValue参数设置为“false”时，则可以删除一个已存在的流。

流标识符由媒体网关控制器进行分配的，仅在MGC和MG之间有效的局部参数。在一个关联内，流标识符用于指示相互连接的媒体流。具有相同流标识符的流彼此相互连接。

当Move命令可以将一个终结点从一个关联移至另一个关联时，对于被移入终结点的关联，其效果等同于添加了一个新的终结点，且新加入终结点上的流标识符等同于移入之前的终结点上的流标识符。

7.1.7 本地控制描述符 (alControl Descriptor)

本地控制描述符由模式 (Mode)、预留组 (ReserveGroup)、预留值 (ReserveValue)、以及包所定义的和特定流相关的终结点属性组成。属性参数值可以参照7.1.1进行规定。

模式属性可有以下几种取值：“只发送（Sendonly）”，“只接收（Receiveonly）”，“可发可收”，“休止（Inactive）”和“环回（Loopback）”。“只发送”、“只接收”和“环回”是相对于关联的外部而言，因此，当流的模式设置为“只发送”时，不应将接收到的媒体传送至关联内。当流的模式设置为“环回”时，本地所接受到的媒体流应环回至媒体的发送方（远端），关联内的终结点之间无媒体流传送。被环回的媒体流应依据远端描述符来进行发送。模式属性的默认值为“Inactive”。信号和事件均不受模式属性的影响。本地控制描述符中的模式属性将优先于本地和远端描述符中的模式属性被处理。

预置值（ReserveValue）和预留组（ReserveGroup）是两类布尔类型的预留属性，它们用于明确当接收到本地和/或远端描述符之后，媒体网关所采取的处理规则。

如果预留属性值为“True”，则当资源充足时，媒体网关必须为本地或远端描述符中指定的所有可选属性预留资源。媒体网关还必须通过命令响应返回其已预留的资源。如果媒体网关无法支持可选属性的任意一种，则媒体网关必须在返回给媒体网关控制器的命令响应中将本地和/或远端描述符设置为空。当媒体网关已经为一个以上的属性预留了资源时，则媒体流可以在任何一个预留资源上进行发送或接收，以及被处理，尽管同时可能只允许有一个预留资源处于激活状态。

如果预留属性值为“False”，则媒体网关必须选择由本地或远端描述符所指定一个属性。如果媒体网关尚未给所选择的属性预留资源，则媒体网关必须预留资源。相反，如果媒体网关已经为指定的属性预留了资源（之前ReserveValue或ReserveGroup属性值为“True”时所进行的消息交互），则媒体网关必须释放先前预留的任何冗余资源。处理完毕后，媒体网关必须向媒体网关控制器返回响应消息，且响应中需包含本地和/或远端描述符所选择的属性。当由于资源不足，导致媒体网关无法支持所指定可选属性的任意一种时，媒体网关必须返回差错代码510（资源不足）。

ReserveValue和 ReserveGroup参数的默认值为“FALSE”。有关ReserveValue和 ReserveGroup参数的详细描述参见7.1.8。

通过新设置一个本地控制描述符可以完全替换媒体网关中先前设置的本地控制描述符。因此，如果期望本地控制描述符保持先前设置的参数值，则媒体网关控制器务必在新的本地控制描述符包含先前设置的参数值。如果媒体网关控制器期望从已存在的本地控制描述符中删除某些信息，则只需在Modify命令中再次发送本地控制描述符，且将不需要的信息从中删除。

注一 模式属性已知为“流模式”，其编码方式参见附件A和B。在本建议书中，模式和流模式是可以相互替换的术语。

7.1.7.1 本地控制属性（LocalControl）

7.1.7.1.1 模式

属性名称： StreamMode

描述： 该属性值用于指示终结点的当前服务状态。

类型： 枚举。

可能的取值:

Inactive:	终结点不允许处理任何媒体流。
SendOnly:	终结点可以将媒体流从关联内部传送至外部。
RecvOnly:	终结点可以将媒体流从关联外部传送至内部。
SendRecv:	终结点可以将媒体流在关联内部和外部之间进行双向传送。
LoopBack:	终结点将接受到的媒体流环回至发送方。

默认值: Inactive

定义于: LocalControl

属性: 读/写

7.1.7.1.2 预留组 (ReserveGroup)

属性名称: ReserveGroup

描述: 指示媒体网关应该预留资源来支持单个媒体群或尽可能的媒体群, 这些媒体群由本地和远端描述符进行定义。参见7.1.8。

注 — “媒体群”可以是采用ASN.1编码后的所定义的PropertyGroup (见附件A), 或者是采用文本编码后的单个SDP会话。

类型: 布尔型

可能的取值:

True:	媒体网关将为本地和远端描述符预留所有可能的媒体群。
False:	媒体网关将参照7.1.8所定义的规则为本地和远端描述符预留所有一个媒体群。

默认值: False

定义于: LocalControl

属性: 读/写

7.1.7.1.3 预留值 (ReserveValue)

属性名称: ReserveValue

描述: 指示媒体网关应该预留资源来支持单个属性集或尽可能的属性集, 这些属性集由本地和远端描述符进行定义。参见7.1.8。

类型: 布尔型

可能的取值:

True:	媒体网关将尽可能地预留资源来支持所选择的媒体群所定义属性组 (此时, 如果ReserveGroup等于“False”), 或者本地和远端描述符所定义的每一个媒体群 (此时, 如果ReserveGroup等于“False”)。
False:	媒体网关将预留一个属性集来支持所选择的媒体群所定义属性组 (此时, 如果ReserveGroup等于“False”), 或者由本地和远端描述符所定义的每一个媒体群 (此时, 如果ReserveGroup等于“False”)。

默认值: False
定义于: LocalControl
属性: 读/写

7.1.8 本地和远端描述符 (Local和Remote)

通过本地和远端描述符, 媒体网关控制器可以指示媒体网关为终结点预留媒体流编解码资源。通过在响应中包含本地或远端描述符, 媒体网关可以返回其实际预留的资源。如果某些必选参数未包含在由媒体网关控制器发送的消息中, 则媒体网关应在响应消息中附加这些被省略的属性或参数值。(例如, 当媒体网关控制器仅指定了负载类型时, 媒体网关应返回视频的具体编码类型和参数)。

当请求媒体网关预留资源的命令消息中使用了“未指定”的通配符参数时(例如, CHOOSE), 为了避免所引起的混淆, 媒体网关控制器应提供尽可能多的信息, 从而媒体网关可以作出正确的选择。例如, 当CHOOSE未指定应用类型(例如, SDP编码中的媒体名称), 则需要提供更多的有用信息(例如, SDP编码中的属性行)。

本地描述符用于描述由媒体网关接收到的媒体, 远端描述符用于描述由媒体网关发送的媒体流。

当采用文本方式进行编码, 本地和远端描述符应使用RFC 2327所规定的SDP会话描述来进行构造。如果相关的SDP会话描述内容是由媒体网关控制器发送至媒体网关, 则其句法规定与RFC 2327的句法规定在某些方面会有所不同:

- “s=”行、“t=”行和“o=”行为可选项;
- 单个参数值允许使用通配值“CHOOSE”; 以及
- 单个参数值允许使用多个可选参数。

流描述符用于对一个双向媒体流进行描述, 因此SDP会话描述部分不允许包含多个媒体描述行(“m=”)。流描述符可以包含附加的多个会话描述部分。终结点上的每一个媒体流必须明确出现在流描述符中。如果一个描述符包含多个SDP会话描述部分, 则会话描述部分应以“v=”行作为分隔行, 否则这多个SDP会话描述部分将作为可选项发送至媒体网关。会话描述部分必须遵循RFC 2327的相关规定。当采用二进制方式进行编码, 本地和远端描述符由本建议书附件C所定义的属性集组成。每一个属性集包含一个会话描述的相关参数。

以下部分将对本地和远端描述符进行详细说明。详细说明包含两个部分。第一部分对描述符基本内容进行说明。第二部分对媒体网关接收到本地和远端描述符后的操作方式进行规定。媒体网关所采取的具体操作方式与本地控制描述符中规定的ReserveValue 和ReserveGroup参数值有关。

本地或远端描述符可使用以下类型参数值:

- 未指定“unspecified”(例如: 描述符省略);
- 空;
- “部分指定(underspecified)” : 属性参数数值使用CHOOSE进行部分指定;
- “完全指定(fully specified)” ; 或

- “多重指定 (overspecified)”：描述符具有多组属性或属性具有多个参数值。

当本地或远端描述符由媒体网关控制器发送至媒体网关时，媒体网关应必须参照本建议书的7.1.1和以下规则对描述符进行注释和澄清：

- a) 如果本地或远端描述符的参数值为“未指定”类型，则应视为必选参数丢失。媒体网关控制器要求媒体网关使用该描述符的先前值。如果之前未对此描述符进行规定，则随后处进行命令处理时应忽略该描述符。
- b) 如果媒体网关控制器发送给媒体网关的本地或远端描述符为“空”，则表明媒体网关应释放其为接收（或发送）媒体流所预留的全部资源。
- c) 如果本地或远端描述符包含多组属性，或一组属性包含多组参数值，则属性值的选择优先级应依次递减。
- d) 如果媒体网关控制器发送给媒体网关的本地或远端描述符包含“部分指定”或“多重指定”的属性，则表明媒体网关可以为每个属性选择一个或多个可能参数值。当采用“多重指定”的属性时，其属性参数列表的选择优先级依次递减。

在遵循以上规则的前提下，以下操作方式将与本地控制描述符中的ReserveValue和ReserveGroup的属性值有关。

如果ReserveGroup参数值为“True”，则表明媒体网关应为所要求的且当前其所支持的属性组预留尽可能多的资源。如果ReserveValue参数值为“True”，则表明媒体网关应为所要求的且当前其所支持的属性参数预留尽可能多的资源。

注一 当本地或远端描述符包含多组属性，且ReserveGroup参数值为“True”时，则要求媒体网关应预留任何其所能支持的媒体流编解码资源。例如，如果本地描述符包含两组属性，一组是G.711A-律音频包，另一组是G.723.1音频包，则媒体网关所预留的资源可以支持G.711A-律或G.723.1音频包的编解码。然而，媒体网关不必同时预留可以处理两种音频包编解码所需的资源。ReserveValue参数值的处理方式与ReserveGroup参数值处理方式类似。

如果ReserveGroup参数值为“True”或ReserveValue参数值为“True”，应按如下规则进行处理：

- 如果媒体网关无法提供足够的资源来支持媒体网关所要求的本地和远端描述符所定义的所有资源，则媒体完预留的资源应该至少能支持本地和/或远端描述符中所指定的一种资源。
- 如果媒体网关无法提供足够的资源来支持媒体网关控制器所要求的本地或远端描述符的任何一种资源，则媒体网关应返回本地或远端描述符为空的命令响应。
- 如果媒体网关的请求中包含本地和远端描述符，则媒体网关所返回的响应必须包含所有其所预留的资源属性组及其属性值。如果媒体网关无法支持本地(或远端)描述符的任何一种资源请求，则返回的本地（或远端）描述符必须为“空”。
- 如果本地控制描述符中的模式属性值为“只发送”、“可发可收”或“环回”，则媒体网关必须准备开始接收在对媒体网关响应中所定义的任何编码方式的媒体流。

如果ReserveGroup参数值为“False”且ReserveValue参数值为“False”，媒体网关应参照如下规则为本地和远端描述符中分别选择一个参数值：

- 媒体网关应当选择本地描述符的第一个选项以支持远端描述符。

- 如果媒体网关无法支持本地和远端描述符所指定的任何参数值，则返回差错代码 510（资源不足）。
- 媒体网关应在其所返回的响应种包含其所选择的本地和远端描述符的可选参数值。

新设置一个本地或远端描述符将完全替换之前由媒体网关所设置的相关描述符。因此，如果期望保留先前本地和远端描述符的信息，则必须在新设置的本地和远端描述符中保留先前的描述符相关参数。如果期望从本地和远端描述符中删除某些内容，则媒体网关控制器可通过向媒体网关发送Modify命令，且命令中将不包含被删除的描述符相关内容。

7.1.9 事件描述符 (Events)

事件描述符参数由请求标识符 (RequestID) 和媒体网关被要求检测的一组事件列表构成。请求标识符用于将事件请求和通知命令进行关联。被请求检测的事件包括传真信号音、导通检测结果、摘机/挂机和插簧拍插等。当事件描述符为空 (例如，没有事件被要求检测)，则请求标识符应被忽略。

描述符中所定义事件由事件名称，流标识符，“KeepActive”标志位，“NotifyBehaviour”标志位和其他一些可选参数构成，其中流标识符、“KeepActive”和“NotifyBehaviour”标记位为可选参数。事件名称可使用包名称 (该事件所在的包) 和事件标识符 (EventID) 两部分来标识。事件标识符可使用通配符“ALL”，即检测包中所定义的所有事件。流标识符的默认值等于“0”，表明所需要检测的事件与特定的媒体流无关。事件可能包含附加的事件参数。这些附加参数使得无需创建大量的事件就可以实现事件的多样性。有关事件参数的详细定义参见包定义。

如果事件描述符包含有“数字映射完成”事件，则事件描述符中的EventDM参数用来携带与数字映射相关的字符和数值。具体内容参见本建议书的7.1.14。

当事件描述符处于激活状态时，且检测到其中所包含的事件发生时，默认地，媒体网关应向媒体网关控制器发送通知 (Notify) 命令进行事件上报。如果检测到发生的事件是一个激活的数字映射所采集的当前拨号字符串，则 Notify消息可以延迟发送 (具体要求参见7.1.14)。是否发送 Notify消息还依赖于“NotifyBehaviour”标志位。当处于激活事件描述符中的事件被检测时，可能导致当前的信号音被停止播放，或导致当前的事件和/或信号描述符被替换。如果事件描述符没有被另一个事件描述符所替换，则某个事件被检测到之后，原事件描述符的状态仍处于激活状态。

如果事件缓存控制 (EventBufferControl) 的属性值为“LockStep”，则检测到事件发生之后，正常的事件处理将被暂停。此后，当检测到事件缓存描述符中所描述的任何一个事件之后，应将被检测到事件以及事件发生的时间添加到事件缓存队列 (先进先出 (FIFO) 队列) 的尾部。之后，媒体网关必须等待新的Events描述符被加载。如果接收到的命令中包含新的事件描述符或被一个内嵌的事件描述符激活，将导致一个新的事件描述符被加载。

如果“EventBufferControl”参数值为“Off”，媒体网关应继续按照当前激活的事件描述符进行处理。

如果内嵌的事件描述符被激活，媒体网关应继续按照最近被激活的事件描述符进行处理。

注1— 为了实现事件缓存处理，内嵌事件描述符激活时的处理方式与接收到新的事件描述符处理方式相同。

如果媒体网关接收到的命令中包含新的事件描述符，媒体网关中的事件缓存队列可能已经缓存了一个或多个事件。“EventBufferControl”的属性值将决定媒体网关处理这些缓存事件的方式：

情况1：

如果EventBufferControl参数值等于“LockStep”且媒体网关接收到新的Events描述符，则媒体网关将首先检查事件缓存队列并采用以下操作：

- 1) 如果事件缓存队列为空，媒体网关将根据新的 Event 描述符检测事件。
- 2) 如果事件缓存队列为非空，则媒体网关从该 FIFO 队列的第一个事件开始处理：
 - a) 如果新的 Event 描述符中包含事件缓存队列中某个的事件，则默认地，媒体网关将向媒体网关控制器发送 Notify 命令，并将该事件从事件缓存队列中删除。Notify 命令中所包含的时间戳参数必须为检测到事件发生的实际时间。此后 MG 应等待新的事件描述符到达。在等待新的事件描述符过程中，任何与事件缓存描述符所匹配的事件将放置在事件缓存队列中，事件处理步骤 1) 开始重复；
 - b) 如果新的事件描述符中不包含事件缓存队列中的事件，媒体必须删除该事件，事件处理步骤 1) 开始重复。

情况2

如果EventBufferControl的参数值等于“OFF”且媒体网关接收到新的事件描述符，则媒体网关应根据新的事件描述符进行新一轮的事件检测。

如果媒体网关接收到EventBufferControl参数值等于“OFF”，则必须删除事件缓存队列中的所有事件。

只要不引起由上报事件所带来的不必要的时延，媒体网关可以采用一个事务上报多个事件。

与Notify命令的有关流程参见本建议书相关附件或 H.248.x系列建议书。

EventBufferControl参数的默认值为“OFF”。

注2 — 由于EventBufferControl参数值被包含在终结点状态描述符中，媒体网关接收到的命令中可以用来改变EventBufferControl的参数值而命令本身却不包含任何事件描述符。

通常，被检测的事件可能导致激活信号被停止播放。如果被检测的事件包含KeepActive参数，则如果终结点上检测到事件发生，其上的激活信号不应被媒体网关停止播放。

“NotifyBehaviour”标记位用于指示Notify命令按如下方式操作：

- “NotifyImmediate”：立即发送（默认值）；
- “NeverNotify”：从不发送；
- 或根据媒体网关控制器的负载调节发送（发送或压缩发送）。

本建议书的E.15对“NotifyBehaviour”的使用方法做了进一步描述。当“NotifyBehaviour”与DigitMap一起使用时，活动“DigitMap”的结束会导致Notify消息的发送。调整通知（Regulated Notify）消息可以关联有一个替代的内嵌事件描述符或信号描述符。如果一个通知消息是调整过的，则与之关联的替代内嵌描述符应被触发。否则，最初的内嵌描述符应被触发。如果“NotifyImmediate”或“NeverNotify”被置位，则当检测到事件发生时，所有初始的内嵌描述符应被触发。

一个事件对“ResetEventDescriptor”标记位的处理方式取决于包含该事件的事件描述符是否是内嵌的。对于一个内嵌的事件描述符，“ResetEventsDescriptor”标记位的设置将使得终结点上的事件描述符被重置为内嵌的事件描述符被触发之前的状态（也就是，由 Modify 指令明确设置的最后一个事件描述符。如果终结点最后一次重置为空关联（NULL Context）之后没有过 Modify 指令，则该描述符被重置为“MG-provisioned”事件描述符）。对于一个非内嵌类型的描述符，“ResetEventsDescriptor”标记位的设置将使得该终结点的活动事件描述符被重置。此时，所有匹配的而且是去激活的 DigitMap 完成事件将被重新激活。

“ResetEventsDescriptor”标记位只能在物理终结点上设置。“ResetEventsDescriptor”标记位应当在终结点处于空关联状态时才能被使用。如果终结点已经脱离空关联而又发现“ResetEventsDescriptor”标记位的话，该标记位将不起任何作用。

事件可内嵌信号描述符和/或事件描述符，当内嵌存在时且检测到该事件发生时，内嵌的信号描述符和事件描述符可替代现有的信号或事件描述符。例如，当检测到摘机事件时，可以指定播放拨号音信号，或者当检测到拨号事件时，停止播放拨号音信号。媒体网关控制器发送的事件描述符中不允许同时包含KeepActive标记和内嵌信号描述符。

内嵌描述符最多允许一级。一个内嵌的事件描述符不允许再嵌入其他事件描述符，然而却给继续内嵌一个信号描述符。

MG接收到新的事件描述符将替换原有的Events描述符。此时，必须结束正在执行的事件上报，在包含新事件描述符中的命令被处理之后，应根据新接收到描述符进行事件检测。

当事件描述符为空时，将停止事件检测和事件上报。当事件缓存描述符为空时，将清空事件缓存队列中的所有事件，且在LockStep状态下累计的所有事件将失效：只有事件描述符处于激活状态，才允许执行事件上报。如果一个空的事件描述符被激活且终结点处于“LockStep”状态，则事件缓存队列应被立即清空。

7.1.10 事件缓存描述符（EventBuffer）

事件缓存描述符由事件列表和与事件相关的参数构成。当事件缓存控制（EventBufferControl）参数值等于“LockStep”时，媒体网关应进行事件检测且缓存被检测到的事件（参见7.1.9）。

7.1.11 信号描述符（Signals）

信号是由媒体网关产生的诸如信号音、通知音或其他线路相关信号（如插簧闪断）等媒体。复杂的信号可以包含一个简单信号序列，以及媒体或线路信号接收和分析的先决条件。例如，在导通检测包中对接收的数据进行反射。信号描述符也可以请求为下一个信号准备媒体资源。

信号描述符包含媒体网关用于向终结点播放的信号集。信号描述符可以包含多个信号或信号序列。信号描述符可以包含“空”信号和信号序列。其中，支持信号序列为可选要求。

信号通过包进行定义。信号必须由包标识符和信号标识符（SignalID）一起进行标识。信号标识符不允许使用通配符。信号还可以包括以下可选参数：流标识符（默认值为0，指示信号与特定的媒体流无关）、信号类型、持续时间和与信号相关的其他附加参数。这些可选参数使得无需定义大量的单个信号可以实现信号的多样性。

媒体网关控制器通过“通知结束(NotifyCompletion)”参数来指示其期望在信号播放结束之后才上报通知消息。导致信号结束的原因可能是：信号超时(或信号自动结束)、信号被事件中中断、信号描述符被替换、信号停止或由于某种原因导致信号无法重新启动。如果信号描述符中不包含“通知结束”参数，则仅仅当由于其他原因导致信号结束时，媒体网关才需要向媒体网关控制器发送Notify消息。为了允许向媒体网关控制器上报信号播放完成，当前激活的事件描述符必须允许包含“信号结束(Signal Completion)”事件(参见E.1.2)。当同一个信号标识符包含多个信号时，可选参数“请求标识符”用于将与某个特定的信号与信号标识符进行关联。从而媒体网关控制器可以区分与某一个信号有关的不同“信号完成”事件。当NotifyCompletion参数未包含在命令中，请求标识符参数也将不允许出现在命令中。

持续时间参数类型为整数，单位为百分之一秒。

信号类型可以分为以下三类：

- 开/关(On/Off)：信号间一直持续到被切断才会结束；
- 超时(Timeout)：信号将持续规定的一段时间后自行结束；
- 简约(Brief)：该信号的持续时间很短，会自动结束，或许由于新接受到信号导致该信号结束。该信号类型不需包含持续时间参数值。

当默认类型为非超时类型的信号被重新设置为超时信号类型时，则信号描述符必须包含持续时间参数值。

如果信号描述符明确规定了某个信号的类型，则该信号类型将覆盖默认的信号类型(参见12.1.4)。如果“开关”信号类型指定了持续时间参数，则该参数应被忽略。

信号序列由信号列表标识符和一序列依据顺序进行播放的信号构成。除了信号列表中最后一个信号可以是“开关”信号类型之外，该列表中其他信号不允许是“开关”信号类型。信号列表的持续时间为列表内所包含的所有信号的持续时间的总和，该总和还包含每两个信号时延总和。

如果“信号间播放时延”参数被指定于不处于信号序列列表中的信号或信号序列列表的最后一个信号，则该参数应当被忽略且该信号应当被视为在信号播放时延参数超时之前，信号的播放已经完成。处于信号列表中一个信号的播放时长与信号间播放时延一起构成了两个信号播放之间的时延。

如果同一个信号描述符包含多个信号和信号序列，则媒体网关必须同时播放所有信号。

如果未明确规定信号的方向，则信号应朝着关联的外部终结点进行播放。如果明确规定了信号的方向，则它将覆盖默认的信号方向。对于具有方向属性的信号，本建议书规定该信号的方向属性将具有比包定义的方向属性更高的优先级。如果媒体网关无法按照某个由媒体网关控制器指定的方向进行播放，则媒体网关将返回差错代码501(无法实施)。当同一信号被作用于同一个事务中的多个终结点时，媒体网关应该考虑使用相同资源来产生信号。

新接收的信号描述符或在终结点上检测到事件发生将导致终结点上信号被停止播放(参见7.1.9)。

新接收的信号描述符可以替换任何现有的信号描述符。正在播放的信号如果未包含在新接收的信号描述符中，则必须停止正在播放的信号。然而，以下情形应当除外：如果被替换的信号描述符中的信号包含KeepActive参数且该信号尚未完成播放，则该信号必须继续被播放。如果被替换的信号描述符中的信号。

包含KeepActive参数且信号尚未开始播放，则该信号必须被忽略。如果被替换的信号描述符包含信号列表且与现有的信号描述符具有相同的标识符，则按照以下方式处理：

- 忽略被替换后的信号描述符中信号列表中的信号类型和信号序列；且
- 包含在当前信号描述符的信号列表中且正在被播放的信号，将不允许被中断。

7.1.12 审计描述符 (Audit)

审计描述符用于指定需要被审计的终结点信息。审计描述符由需要返回的描述符列表和/或单个属性构成。审计描述符可作为任意一个命令的输入参数用于强迫媒体网关返回属性、事件、信号和统计的当前参数值，即使命令中未包含某个描述符或该描述符为“部分指定”时，也必须如此。审计描述符可以包含如下内容：

调制解调 (不再推荐使用, 参见 7.1.2)	
复用	
事件	
媒体	
信号	
被观察事件	
数字映射	
统计	
包	
事件缓存	
单个统计项:	
媒体属性	数字映射
事件	统计
事件缓存	包
信号和信号列表	关联属性

审计描述符可以为空，表明此时不需要返回描述符相关信息。尤其是当使用通配符时，此方式可以用于Subtract命令来禁止返回统计信息。

7.1.13 业务改变描述符 (ServiceChange)

业务改变描述符可以包含以下参数：

- 业务改变方法 (ServiceChangeMethod) ；
- 业务改变原因 (ServiceChangeReason) ；
- 业务改变地址 (ServiceChangeAddress) ；
- 业务改变时延 (ServiceChangeDelay) ；
- 业务改变协议子集 (ServiceChangeProfile) ；
- 业务改变版本 (ServiceChangeVersion) ；
- 业务改变媒体网关控制器标识符 (ServiceChangeMGCIID) ；
- 时戳 (TimeStamp) ；
- 其他扩展 (Extension) ；
- 业务改变信息 (ServiceChangeInfo) ；
- 业务改变未完成标志位 (ServiceChangeIncompleteFlag) 。

参见7.2.8。

7.1.14 数字映射描述符 (DigitMap)

7.1.14.1 数字映射的定义、创建、修改和删除

数字映射是驻留在媒体网关内部、用于检测和上报终结点拨号事件的拨号计划。数字映射描述符由数字映射名称和相应的数字映射构成。数字映射描述符可以事件描述符被预先加载于媒体网关之中,且通过数字映射名称被引用;数字映射描述符也可以动态定义,随后通过所定义的数字映射名称进行引用;或者可以通过事件描述符指定实际的数字映射。与数字映射描述符的传输顺序无关,事件描述符中所包含的“数字映射结束”事件可以通过“数字映射名称”来引用同一命令中由数字映射描述符定义的数字映射参数。

数字映射描述符允许被包含在本建议书所规定的任何命令中。一旦数字映射被定义,则可以适用于命令中由终结点标识符(可能为通配符)所指定的所有终结点。定义在根终结点上中的数字映射具有全局性,可以适用于媒体网关中的所有终结点,除非某个终结点上已经定义了一个相同名称的数字映射。数字映射描述符可按以下方式动态定义数字映射参数:

- 可以通过定义一个未使用的数字映射名称来创建一个新的数字映射,且必须设置该新创建的数字映射的参数值。
- 通过为数字映射更新一个新的参数值,数字映射参数值可以被更新。数字映射更新后,当前正使用该数字映射的所有终结点必须继续使用更新前的数字映射。数字映射更新后,如果事件描述符引用了更新后的数字映射名称,则这些事件描述符都必须使用更新后的数字映射。
- 通过设置一个已被定义的数字映射名称为空值,可以删除一个数字映射。当前使用被删除的数字映射的所有终结点应继续使用该数字映射。

7.1.14.2 数字映射定时器

用来保护根据数字映射来采集号码的定时器分为三类:起始定时器(T),短定时器(S)和长定时器(L)。

- 1) 起始定时器T作用于任何与数字映射相关的号码采集之前。如果起始定时器被重新设置为0 ($T = 0$),则起始定时器将被禁止。此时,媒体网关立刻等待拨号事件。
- 2) 如果媒体网关确认当前号码字符串至少还需要一位号码来匹配数字映射中的任意一种方案,则位间定时器值应该设置为长定时器L(例如16 s)。
- 3) 如果当前号码字符串能够匹配数字映射中的某一方案,但同时有可能接收到多位号码而导致与不同的方案发生匹配时,此时,不应立即上报匹配情况,媒体网关必须使用短定时器S来等待接收更多的号码。

数字映射中的定时器参数是可配置的。定时器的默认值由媒体网关进行设置,然而其可通过数字映射描述符进行重新设置。

7.1.14.3 数字映射句法

数字映射的正式句法是由协议的正式句法描述中的数字映射规则来说明的(参见附件A和附件B)。根据句法规则,数字映射可以由字符串和字符串列表构成。字符串列表中的每一个字符串是一组事件序列,它可以是一组数字映射符号,也可以是数字映射符号的标准表达形式。数字映射字符中的数字是从“0”到“9”之间的任意数字,字母是从字符“A”到信令系统所允许的字母最大值,其中最大值不允许超过字符K,这些数字和字母将与该数字映射所适用的终结点上的事件描述符所描述的事件相对应。(在随路信令系统(CAS)中,数字映射字符与拨号事件之间的映射关系,在与DTMF, MF, R2相关的包中进行了规定。“0”至“9”的数字字符必须与信令系统中的拨号事件相关联。数字映射字母也应该按一定的逻辑结构进行分配,以便采用“范围”来进行识别)。

字母“X”为通配值，其用于代表“0”至“9”数字范围内的任何号码。字符串可包含固定的范围，以代表任意一个满足该范围的拨号事件。符号“.”代表在“.”之前的0次或多次重复事件（事件、事件范围、可选事件集合或通配符）。根据前面所描述的三种定时器规则，满足“.”符号的事件之间将默认采用短定时器S。

除了以上事件符号之外，字符串还可以包含“S”和“L”定时器指示符以及“Z”指示符。“S”与“L”分别用来指定媒体网关对后续事件应该采用短定时器S或长定时器L来替换先前适用的定时器规则。如果一个被明确指定的定时器指示符在数字映射字符序列中生效，且任何其他数字映射字符序列未规定其他的定时器指示符，则必须使用该定时器指示符所规定的定时器。如果所有被明确指定的定时器控制的序列从可选序列中被删除时，则定时器将被恢复至相应的默认值。当S和L定时器指示符出现在由范围描述的字符序列中，则S和L定时器必须被忽略。如果当不同可选字符序列中定时器指示符发生冲突时，则必须使用长定时器L。

“Z”定时器标识符用于定义一个长持续时间的事件：当“Z”被放置于满足条件的字符之前，表示只有当事件的持续时间超过此时间门限时，该事件才将完全匹配。该定时器的门限值由媒体网关进行设置，该参数值与T、L和S定时器一样，可以通过数字映射被重新设置。如果Z指示符后不跟随0~9和A~K范围内的合法字符，则媒体网关将拒绝该数字映射。如果Z标识符被包含于范围定义之内，则该标识符仅仅适用于下一个被接收的号码。如果Z标识符被包含在范围定义之前，则该标识符适用于所有满足该范围条件的号码（此时，需要一段较长的时间来完成范围内所有号码的匹配）。

7.1.14.4 数字映射结束事件

当激活数字映射的事件描述符处于激活状态且该事件描述符未结束时，则该数字映射也处于激活状态。当以下情况发生时，数字映射将视为结束：

- 定时器超时；或
- 已经与某一个可选的字符序列发生匹配，通过检测其他事件，该数字映射中无任何其他字符序列可以被继续完全匹配（或模糊匹配）；或
- 检测到一个事件之后，之后无论接收到任何事件都将无法与该数字映射中一个完整字符序列相匹配。

一旦数字映射结束，必须产生“数字映射结束”事件来指示接收的事件与数字映射相匹配。此时，数字映射处于去激活状态，后续接收到的事件应按以下方式进行处理：

- 如果EventBufferControl参数值为“ON”，则后续的号码事件将按照与其他事件相同的处理机制进行处理；
- 如果EventBufferControl参数值为“OFF”以及激活的事件描述符未发生改变，且该描述符中不允许出现单个号码事件，则必须启动号码缓存机制。缓存将一直持续到前一个数字映射结束事件所指定的缓存时间超时或当前激活的事件描述符被替换；

号码缓存可采用数字映射所定义的字符序列的逻辑结构，字符之前可包含“Z”定时器标识符。用于识别长事件的信号音持续时间的门限值必须等同于最近结束的数字映射所规定的门限值。

如果数字映射结束事件未明确该缓存的时间，则号码缓存时间的默认值为0。由于缓存定时器超时导致缓存终止，则被缓存的内容也相应的被丢弃。

如果新接收的事件描述符导致号码缓存发生终止，且该事件描述符包含一个“数字映射结束事件”参数，且该参数与之前的“数字映射结束事件”源于相同的包，则被缓存的号码应按以下方式进行处理。此时，应按照数字映射结束后继续检测到号码事件的实现方式，对无法被新接收的事件描述符所匹配的缓存号码进行处理。

相反，如果新接收的事件描述符允许上报单个号码事件，则整个被缓存的号码必须立即被处理和上报，然后将缓存的号码清空。

最后，如果新接收的事件描述符既不允许“数字映射结束”事件也不允许单个号码进行上报，则被缓存的号码被删除并停止缓存。

7.1.14.5 数字映射流程

对于即将结束的数字映射，后续的字符事件必须根据如下规则进行处理：

- 1) “当前号码字符串”是一个内部变量，其起始值为空。候选的字符序列集包括数字映射所规定所有可能字符事件。
- 2) 在每一步骤中，如果缓存队列中包含被缓存的号码，则最早进入缓存队列的字符（可能伴随长定时器标识符Z）将从缓存中被删除，且就像检测到一个字符事件，将前进至下一步骤进行处理。在每一步中，等待下一个字符事件的定时器将采用默认的参数值，或者采用一个或多个字符序列中明确规定的定时器。当定时器超时时，如果当前号码字符串可以与候选的字符序列集中所包含的一个字符序列完全匹配，则报告“定时器超时，完全匹配”。当定时器超时，如果当前号码字符串不能与候选的字符序列集完全匹配，或没有任何候选的字符序列可以匹配，则报告“定时器超时，部分匹配”。无论是“定时器超时，完全匹配”还是“定时器超时，部分匹配”，如果数字映射结束事件允许详细的超时报告，则上报的号码字符串应使用符合条件的“L”、“S”或“T”作为其结束字符。
- 3) 如果在定时器超时之前检测到字符事件，应将字符事件映射成号码字符串，并将其添加到当前号码字符串的后面。当且仅当字符事件的持续时间与当前位置有关时，不论该字符事件所持续时间的长短，都应记录该事件的持续时间（此时至少有一个候选的字符序列中在此位置有一个“Z”指示符）。
- 4) 当前号码字符串应与候选的字符序列进行比较。当且仅当在该位置上具有长持续时间的字符事件序列与之相匹配时（例如，字符事件是长持续事件并满足该位置的要求），则任何该位置上未规定长持续时间的候选字符序列都将被丢弃，且在最近接收到的号码字符前插入“Z”以修改当前的号码字符串。如果在该位置可能发生的长持续时间字符事件无法与接收到的字符相匹配，则该长持续时间字符事件将从候选的字符序列中丢弃。如果候选的字符序列在给定位置未规定长持续时间字符事件，并且应用上述规则之后仍保留在候选的字符序列中，则进行匹配评估时，被检测的字符事件的持续时间将视为无关。

- 5) 如果仅剩余一个候选字符序列且完全匹配时，则产生一个完全匹配的“数字映射结束”事件。如果无候选的字符序列相匹配，则最近接受到的字符事件将会从当前号码字符串中被删除；如果在检测到最近一个字符事件之前，前一步中号码字符串可以与一个候选字符序列完全匹配或部分匹配时，将产生一个匹配完成事件。从当前号码字符串删除的字符将作为一个独立事件按照上一节定义的规则进行上报、缓存或丢弃处理。详细规则如下：
- a) “数字映射结束”事件可以指定被删除的字符事件作为该事件的一个返回参数。这与后续的号码事件处理无关；
 - b) “数字映射结束”事件可以指定其他被删除的字符事件。此时，这些字符事件应立即被丢弃。任何缓存或其他处理机制仅适用于后续的字符事件。
- 6) 经过第5)步之后，如果未上报“数字映射结束”事件，则返回到第2)步进行处理。

7.1.14.6 数字映射的激活

当新接收的事件描述符作用于终结点或嵌套的事件描述符被激活时，且事件描述符包含“数字映射结束”事件，则数字映射将被激活。“数字映射结束”事件中包含一个EventDM参数值。每个被激活的新数字映射应从遵循以上一节所规定流程的第1)步开始，且当前号码字符串为空。在当前号码字符串之前，由前一次数字映射激活所获得任何字符事件将被丢弃。

如果“数字映射结束”事件未包含EventDM参数，则该“数字映射结束”事件是非法的。如果媒体网关接收到的事件描述符中包含这种非法的“数字映射结束”事件，媒体网关必须向媒体网关控制器上报差错代码457（信号或事件发生参数丢失）。

7.1.14.7 数字映射和事件的交互处理

当数字映射被激活时，应检测包含“数字映射结束”事件的包中定义的所有事件。除下列情况外，正常的事件处理方式对检测到的事件继续适用（例如，如果“数字映射结束”事件的KeepActive标志未被标记，则将停止信号）：

- 对于包含“数字映射结束”事件的包，除结束事件之外的其他事件不允许进行单独上报的；且
- 对于导致部分匹配结束的事件不进行识别，因此直到随后检测到“数字映射结束”事件之前，这些事件将不会发生作用。同理，缓存的字符事件将也不会被识别，直到被缓存的字符事件被处理之前，被缓存的字符事件将不起作用。

7.1.14.8 通配符

当包中定义了“数字映射结束”事件时，如果某个事件由该包的名称和一个通配的事件标识符进行定义，则数字映射将被激活；根据7.1.14.6的规定，该事件必须包括一个EventDM参数值。如果该包本身包含字符事件，则当单个字符事件被检测时，这种通配事件将导致媒体网关将被检测的单个字符事件上报至媒体网关控制器。

7.1.14.9 示例

例如，拨号计划如下所示：

0	本地呼叫
00	长途呼叫
xxxx	本地扩展号码(首位为“1至7”之间的任意数字)
8xxxxxxx	本地号码
#xxxxxxx	离线扩展
*xx	补充业务
91xxxxxxxxxx	长途号码
9011 + 最多15位	国际号码

如果参照E.6定义的DTMF检测包来采集被叫号码，则以上拨号计划对应的数字映射方式如下所示：

(0 | 00 | [1-7]xxx | 8xxxxxxx | Fxxxxxxx | Exx | 91xxxxxxxxxx | 9011x。)

7.1.15 统计值描述符 (Statistics)

统计值描述符用于对临时终结点在其生存期之内或者脱离空关联的物理终结点的状态和使用情况进行描述。统计信息可以是基于终结点之上，或者基于一个特定的流描述符进行统计。默认的，统计是基于终结点之上进行的。如果一个流无法支持统计功能，则应返回差错代码460“无法对流进行统计”。通常，每个终结点都支持一组标准的统计参数(例如，发送或接收的字节数)。默认的，一个终结点可以采集的统计参数由终结点实现的包所决定。通过描述符也可以指定终结点所采集的统计参数。对一个统计值描述符进行重新设置可以覆盖其之前的统计值描述符参数值。因此，为了保护现有的统计参数，这些统计参数应当被包含在新的统计值描述符之中，且统计参数值不允许被清空。已经从统计值描述符中被删除的那些统计参数应当保持其统计参数值，直到该终结点被删除。然而，如果某个统计参数由于一个后续的统计值描述符的出现导致重新激活，则其参数值应当被清空。媒体网关控制器通过一个包含包标识符(PackageID)和统计标识符(StatisticID)为通配符“ALL”的统计值描述符，可以对终结点或流之上的所有统计参数进行重新激活。为了对某个包所包含的所有统计参数在终结点或流上进行重新激活，统计值描述符应该当一个统计统计项，且该统计项由一个特定的包标识符和一个等于通配符“ALL”的统计标识符构成。如果某个终结点接收到一个空的统计值描述符，则表明该终结点不需要采集任何统计参数。

默认的，如果在未接收到空的统计值描述符用于指示不需要对采集任何统计参数之前，由于Subtract命令导致当终结点删除或被返回至空关联之中，基于终结点或流的统计参数应该上报给媒体网关控制器。以上默认模式也可以通过在Subtract命令中包含一个空的审计描述符(Audit)而被取代。根据7.1.6的规定，当删除一个流时，默认情况下，基于流的统计参数不会上报给媒体网关控制器。因此，为了对流的统计参数实现采集功能，在流被删除之前，必须向流实施一个包含统计值描述符的审计描述符。统计参数可以通过实施AuditValue命令，或者任何包含了审计描述符的Add/Move/Modify命令而被返回。

统计参数值时可以累加的，统计参数值的上报不会导致参数值被清空。基于终结点的统计参数值通常是对该终结点之上的所有媒体流进行“求和”或者“平均”算法之后的参数值。具体采用哪一种算法鱼具体的统计类型有关。如果未指定具体的统计算法，默认的算法是对参数值进行“求和”操作。由于Subtract命令导致当终结点删除或被返回至空关联之中，这些统计参数将被清空。

7.1.16 包描述符 (Packages)

包描述符仅适用于AuditValue命令，它将用来返回终结点实现的包列表。

7.1.17 被观察事件描述符 (ObservedEvents)

被观察事件描述符适用于Notify命令，用于向媒体网关控制器通知检测到的事件。当Auditvalue命令中使用了该描述符时，则该命令的返回响应将返回事件缓存中尚未上报的事件。被观察事件描述符包含触发Notify命令的事件描述符的请求标识符 (RequestID)，被检测到的事件和被检测到的事件发生的时间，其中被检测的事件发生的时间为可选项。被检测事件发生的时间可以精确到百分之几秒。

7.1.18 拓扑描述符 (Topology)

拓扑描述符用于描述关联中终结点之间的媒体流流向。拓扑描述符仅适用于关联，而不适用于终结点。拓扑的默认值是任意一个终结点可以接收来自关联中任何其他终结点的媒体流。在命令中，拓扑描述符为可选项。当媒体网关不支持拓扑描述符，而媒体网关接收命令中包含拓扑描述符，应返回差错代码444（不支持或未知的描述符），可选地，在差错描述符中的差错文本描述中包含一个不支持的描述符类型“拓扑”。

拓扑描述符应置于一个动作中所包含的所有命令之前。如果接收动作的关联已经存在，则可以在一个动作仅包含一个拓扑描述符。

拓扑描述符的格式为(*T1*, *T2*, *association*[,*StreamID*])。T1和T2用于指定关联中的终结点，可使用通配值“ALL”或“CHOOSE”。如果使用可选参数流标识符，则“相关性”参数仅用于指定两个终结点之间的流标识符所标识的媒体流流向。如果拓扑描述符未包含流标识符参数，则该拓扑描述符适用于终结点上的所有媒体流。“相关性”规定的终结点之间的媒体流流向：

- (*T1*, *T2*, *isolate*)：指示终结点T2和T1之间彼此不允许从对端接收媒体流。
- (*T1*, *T2*, *oneway*)：指示终结点T2可以从终结点T1单向接收媒体流，而不能反向接收。T1或T2之间的任意一个允许使用通配符ALL，但是不允许T1和T2同时使用通配符ALL。
- (*T1*, *T2*, *onewayexternal*)：指示终结点T2接收由T1终结点向外部发送的媒体流，反之不允许。此时，T1不允许使用通配符ALL。
- (*T1*, *T2*, *onewayboth*)：指示终结点T2接收由T1终结点向外部发送和接收的媒体流，反之不允许。此时，T1和T2不允许使用通配符ALL。
- (*T1*, *T2*, *bothway*)：指示终结点T2和T1终结点之间彼此可以双向接收媒体流。此T1和T2允许使用通配值ALL。此时，如果一个终结点同时匹配两者，将不会导致环回。

T1和T2可使用通配符“CHOOSE”，然而应受以下限制：

- 拓扑描述符所在的动作（参见第8节）中包含一个Add命令，且Add命令包含通配符“CHOOSE”；
- 如果终结点T1或T2已经使用了CHOOSE通配符，则不允许再使用部分通配。

当媒体网关为第一个使用了通配符CHOOSE的Add命令分配了终结点标识符时，则同一动作中的拓扑描述符中CHOOSE通配符将于Add命令分配的终结点标识符匹配。在已经包含一个终结点的关联中，与T1或T2匹配的终结点将按照拓扑描述符的要求与新添加的终结点建立连接。如果某个终结点未包含在拓扑描述符中，则该终结点和任何其他终结点之间的拓扑关系保持不变。如果拓扑描述符未指定终结点之间的拓扑关系，新增加至关联中的终结点与其他终结点的“相关性”采用默认值“Bothway”。（例如，Add命令将T3添加到一个已包含T1和T2终结点的关联中，且该命令中拓扑描述符指定为(T3, T1, oneway)，则T3与T2之间的拓扑相关性将使用默认值“bothway”）。

如果拓扑描述符指定了一个特定的流，例如(T1, T2, association[,StreamID])，则终结点之间其他流的拓扑不应受该拓扑描述符的影响。

为了避免操作混淆，无论拓扑描述符是否包含可选的流标识符参数，同一个拓扑描述符不允许指定两个终结点之间的多个“相关性”。例如，(T1, T2, bothway) 和(T1, T2, isolate, S1)必须不允许被包含在同一个拓扑描述符中。当接收到此类拓扑描述符时，媒体网关应返回差错代码421（“未知动作或违法的动作组合”）。

实现单向连接时务必使关联中的其他终结点不感知拓扑发生的变化。

图7、图8和下表是动作中包含拓扑描述符的示例，且假定拓扑描述符是按照图所示的顺序依次执行。图9和图10用于描述相关性为“onewayexternal”和“onewayboth”的操作模式。

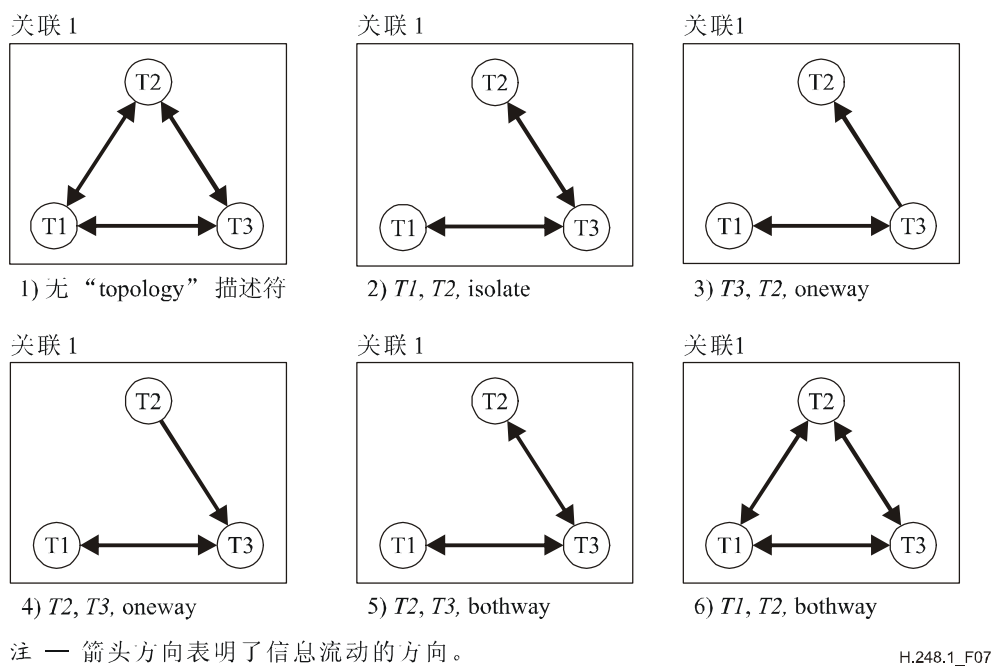


图7/H.248.1 – 拓扑描述符示例

拓 扑	描 述
1	无拓扑描述符 当未包含拓扑描述符时，所有终结点间都具有双向连接。
2	<i>T1, T2 Isolate</i> 删除 <i>T1</i> 与 <i>T2</i> 之间的连接。 <i>T3</i> 与 <i>T1</i> ，以及 <i>T3</i> 和 <i>T2</i> 之间都具有双向连接。
3	<i>T3, T2 oneway</i> <i>T3</i> 到 <i>T2</i> 之间具有的单向连接(即 <i>T2</i> 从 <i>T3</i> 接收媒体流)。 <i>T1</i> 与 <i>T3</i> 之间具有双向连接。
4	<i>T2, T3 oneway</i> <i>T2</i> 与 <i>T3</i> 之间具有单向连接。 <i>T1</i> 与 <i>T3</i> 保持双向连接。
5	<i>T2, T3 bothway</i> <i>T2</i> 被双向连接到 <i>T3</i> 。结果同拓扑2。
6	<i>T1, T2 bothway</i> (<i>T2, T3 bothway</i> 和 <i>T1, T3 bothway</i> 具有隐含或明确的双向连接)。所有终结点之间彼此具有双向连接。

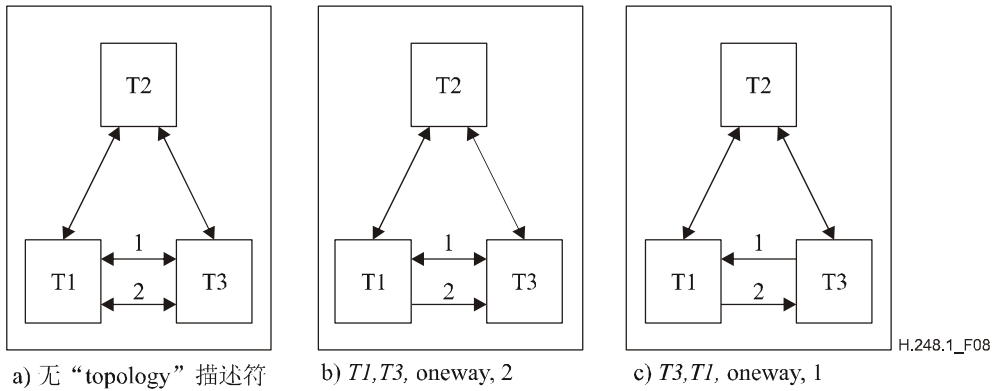


图 8/H.248.1 – 与流相关的拓扑描述符示例

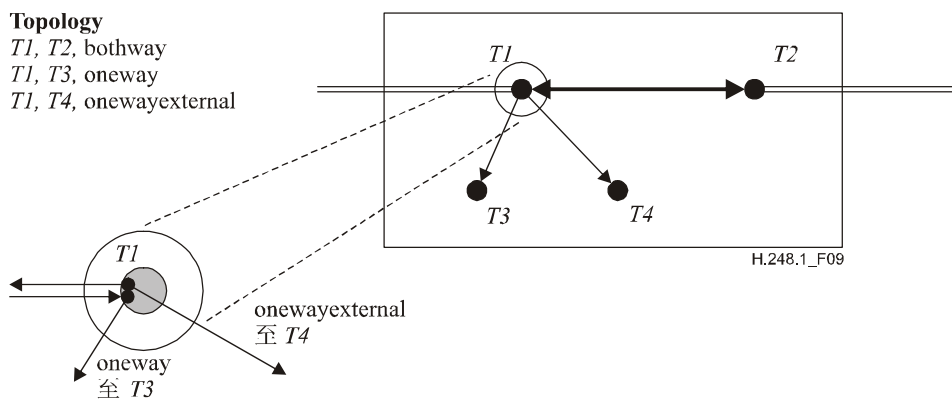


图9/H.248.1 – Onewayexternal与oneway拓扑的对比

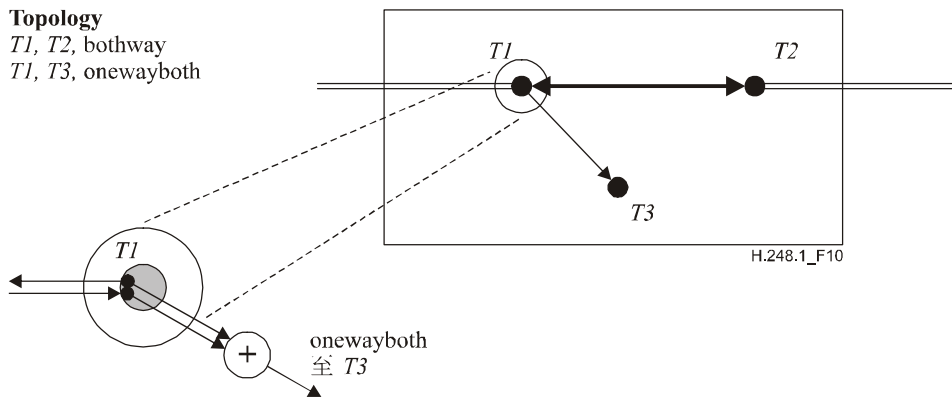


图 10/H.248.1 – onewayboth拓扑的操作

7.1.19 关联属性描述符（ContextAttribute）

ContextAttribute描述符用于向一个关联指定仅仅适用于整个关联而不是某一个终结点的属性（定义在包中）。终结点属性不允许被包含在ContextAttribute描述符之中。在一个动作（Action）之中ContextAttribute描述符应当出现在命令之前。对于一个已经存在的关联，一个动作可以仅包含一个ContextAttribute描述符。关联属性的参数值是“部分指定”（underspecified）的，参见7.1.1。

一个新的ContextAttribute描述符可以完全替换媒体网关所包含的前一个描述符。因此，为了保护的前一个配置信息，媒体网关控制器应当在新的描述符中同时包含前一个配置信息。如果媒体网关控制器期望从现有的描述符中删除某些配置信息，则只要重新发送一个删除了不期望信息的描述符即可。如果ContextAudit或ContextAttribute同时被包含在一个仅包含AuditValue 或 AuditCapabilities命令的动作之中，不会导致如上所述的配置改变。

ContextIDList参数用于提供一组集成的关联标识符。为了实现在响应消息中使用ContextIDList参数的功能，ContextIDList也必须应当被包含在相应的请求消息之中。

7.1.20 差错描述符（Error）

在事务请求处理过程中，如果响应方发生差错，则响应方务必返回包含差错描述符的事务响应。Notify请求也可以包含差错描述符。

差错描述符由已在IANA注册的差错代码组成，可选地包含差错文本描述。本建议书定义的合法的差错代码和差错文本描述参见ITU-T H.248.8建议书。

差错描述符在语义上必须尽可能地与所描述的差错相吻合，应尽可能地说明发生差错请求的差错原因。差错描述符可以按照语义进行解释而并不遵循差错代码的具体内容。例如，差错描述符422（动作句法差错）可以适用于命令，尽管它原本是针对动作而言，而不是具体针对某个命令。

7.2 命令应用编程接口

本节将对本建议书定义的命令和应用编程接口进行描述。应用编程接口将仅对命令和其输入和输出参数进行了说明，而未规定命令的具体操作方式（例如，闭塞呼叫功能）。命令名之后括号内的参数为命令的输入参数，命令名之前为命令的返回参数。其中，输入和输出参数仅做描述性介绍，有关命令的句法定义和编码规则将在本建议书其他小节进行规定。命令参数在命令中出现的顺序是不固定的。命令接收方必须按照描述符出现的顺序执行。

任何命令的响应可以包含差错描述符，此处的应用编程接口未明确出现该差错描述符。

所有被包含在 [...] 中的参数是可选参数。

7.2.1 Add

Add命令用来向一个关联添加一个终结点。

TerminationIDList (终结点标识列表)

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

Add(TerminationIDList

 [, MediaDescriptor]

 [, ModemDescriptor] (*)

 [, MuxDescriptor]

 [, EventsDescriptor]

 [, EventBufferDescriptor]

 [, SignalsDescriptor]

 [, DigitMapDescriptor]

 [, AuditDescriptor]

 [, StatisticsDescriptor]

)

(*) H.248.1 (第2版) (05/2002) 不推荐使用Modem描述符。

终结点标识描述符列表用于指定向关联中添加一个或多个终结点。这些终结点可以是新创建的，也可以是来自从空关联的终结点。如果终结点标识符使用通配符“CHOOSE”，则Add的命令相应将返回被选择终结点的终结点标识符。Add命令可以使用通配符“ALL”，然而这种用法不普遍。如果某个通配值与多个终结点标识符相匹配，那么该命令将尝试所有可能匹配的终结点，同时应返回对每个终结点操作的结果。媒体网关可以按照任意的先后顺序尝试多个终结点标识符。

媒体描述符为可选项，用于描述所有的媒体流。

如果可选项复用描述符出现在命令中，则该描述符用于描述一个复用方式。为简便起见，如果Add命令中包含由复用描述符，且列举了所有目前未处于关联中的终结点，则与Add命令添加单个终结点一样，这些被列举的所有终结点将被添加至该关联中。当处理这种隐含方式的Add命令发生差错时，应返回差错代码471（复用描述符中隐含Add命令差错），进一步命令处理必须被终止。

事件描述符为可选项。如果命令中包含事件描述符，则用于指定终结点上应该检测的事件列表。

事件缓存描述符为可选项。如果命令中包含事件缓存描述符，且事件缓存控制（EventBufferControl）参数值等于“LockStep”时，则该描述符指定媒体网关应当检测和缓存的事件列表。

信号描述符为可选项。如果命令中包含信号描述符，则该描述符用于指定终结点上应该产生的信号列表。

数字映射描述符为可选项。如果命令中包含数字映射描述符，则该描述符用于指定在事件描述符中可被使用的数字映射。

审计描述符为可选项。如果命令中包含审计描述符，则命令响应将返回审计描述符中所包含的任意描述符、属性、信号、事件和统计参数值。

统计值描述符为可选项。如果命令中包含统计值描述符，则该描述符用于指定媒体网关控制器期望收集来自终结点或媒体流的统计信息。统计值描述符每被设置一次，则与该统计值描述符相关的参数值将被重置。

如果可以被修改描述符的参数是“部分指定”或“多重指定”的，则媒体网关可以在命令响应中返回这些描述符。当审计描述符请求对已检测事件、统计、包、事件缓存描述符进行审计时，该描述符可以作为命令的输出参数被返回。

Add命令必须不允许作用于处于“退出服务”（OutOfService）状态的终结点。

7.2.2 Modify

Modify命令用于修改终结点的属性。

TerminationIDList

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

Modify(TerminationIDList

 [, MediaDescriptor]

 [, ModemDescriptor] (*)

 [, MuxDescriptor]

 [, EventsDescriptor]

 [, EventBufferDescriptor]

 [, SignalsDescriptor]

 [, DigitMapDescriptor]

 [, AuditDescriptor]

 [, StatisticsDescriptor]

)

(*) H.248.1 (第2版) (05/2002) 不推荐使用Modem描述符。

终结点标识描述符列表用于指定需要被修改的终结点。此时，终结点标识符终结点应当是确定的。

在某些情况下，终结点标识符允许使用通配符。如果有多个终结点标识符值与通配符匹配，则该命令将修改所有匹配的终结点，同时返回对每个终结点的操作结果。媒体网关可以按照任意的先后顺序尝试多个终结点标识符。**Modify**命令不允许使用通配符“**CHOOSE**”，因为**Modify**命令仅适用于已被使用的终结点。

为简单起见，如果**Modify**命令中包含复用描述符，则按以下方式进行处理：

- 如果复用描述符包含的终结点未包含在当前关联中，则就像**Add**命令添加单个终结点一样，这些终结点将被添加至此关联中；
- 如果先前被包含在复用描述符中的终结点，未包含在新接收到的复用描述符中，则与**Subtract**命令删除终结点一样，这些终结点将从关联中被删除。

Modify命令的其他参数与**Add**中参数定义用法相同，且所有参数的返回值也与**Add**命令的参数的返回值用法相同。

7.2.3 Subtract

Subtract命令用来将一个终结点从所处关联中删除，同时返回与被删除的终结点的统计参数。

TerminationIDList

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

Subtract(TerminationIDList

[, AuditDescriptor]

)

(*) H.248.1 (第2版) (05/2002) 不推荐使用**Modem**描述符。

终结点标识描述符列表用于指定需要被删除的终结点。终结点标识符可以是确定值或通配符，通配符用来指示**Subtract**命令将删除关联中的所有终结点（或一组终结点）。如果多个终结点与通配符匹配，则将删除所有匹配的终结点，同时报告对每个终结点的操作结果。删除多个终结点的顺序可以是任意的。媒体网关可以按照任意的先后顺序尝试多个终结点标识符。

由于**Subtract**命令只允许作用于已存在的终结点，因此终结点标识符不允许使用通配符“**CHOOSE**”。

关联标识符和终结点标识符允许使用通配符“**ALL**”，此时**Subtract**命令将删除所有关联和所有临时的终结点以及将所有的物理终结点返回至空关联中。**Subtract**命令不允许作用于处于空关联中的终结点。

为简便起见，如果**Subtract**命令中包含有复用终结点，则与该命令作用于单个终结点一样，复用描述符所包含的所有终结点将从关联中被删除。

默认地，**Subtract**命令的返回值将返回与终结点和媒体流相关的统计信息的统计值描述符。这些统计信息与关联中被删除的终结点有关。

审计描述符为可选项。如果命令中包含审计描述符，将返回审计描述符中所指定的描述符，审计描述符可能为空。如果命令中未包含审计描述符，默认地应返回统计值描述符。

Subtract命令的允许返回参数与**Add**命令的返回参数相同。

当一个被配置终结点从一个关联中被删除时，其描述符参数值应按照以下方式处理：

- 如果指定的参数值不能恢复为预置的参数值，则属性值恢复为默认值；
- 否则，恢复为预置值。

7.2.4 Move

Move命令用于将一个终结点从它当前所在的关联转移至另一个关联中。**Move**命令是唯一命令被作用于与命令所在关联无关的终结点。使用**Move**命令将终结点从空关联之中移走或将终结点转移至空（NULL）关联之中是不允许的。

TerminationIDList

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

Move(TerminationIDList

 [, MediaDescriptor]

 [, ModemDescriptor] (*)

 [, MuxDescriptor]

 [, EventsDescriptor]

 [, EventBufferDescriptor]

 [, SignalsDescriptor]

 [, DigitMapDescriptor]

 [, AuditDescriptor]

 [, StatisticsDescriptor]

)

(*) H.248.1 (第2版) (05/2002) 不推荐使用**Modem**描述符。

终结点标识描述符列表用于指定需要被转移的终结点。终结点标识符允许使用除“**CHOOSE**”之外的通配符。如果有多个终结点与通配符匹配，那么将尝试转移所有匹配的终结点，同时报告对每个终结点的操作结果。媒体网关可以按照任意的先后顺序尝试多个终结点标识符。“动作”中的目标关联标识符决定了终结点将被转移到哪一个关联。如果**Move**命令将最后一个终结点从一个关联转移，则相应地该关联被删除。

除了Move命令中明确指定了需要修改的描述符属性之外，Move命令将不会改变终结点的其他属性。例如，包含统计参数选项的审计描述符将仅仅返回终结点在执行Move命令之前统计数据。Move命令的允许返回参数与Add命令的返回参数相同。

为简单起见，如果Move命令作用于一个复用终结点时，则复用描述符中包含的所有终结点将被从关联中转移，就像对关联中的每个终结点执行了Move命令一样。

Move命令不允许被作用于“退出服务”（OutOfService）的终结点。

7.2.5 AuditValue

AuditValue命令用于获取终结点的属性、事件、信号和统计的当前值。AuditValue命令也可以用于请求获取描述符中的某一个属性、事件、信号和统计的参数值。AuditValue命令对于保持媒体网关控制器和媒体网关之间状态的同步非常有用。AuditValue命令可以设置过滤条件来要求返回某些特定的参数值。

TerminationIDList

[,MediaDescriptor]

[,ModemDescriptor] (*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,DigitMapDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

[,PackagesDescriptor]

```
AuditValue(TerminationIDList,  
           AuditDescriptor  
           )
```

(*) H.248.1 (第2版) (05/2002) 不推荐使用Modem描述符。

终结点标识符列表可以采用确定值或通配符。如果通配符与多个的终结点匹配时，则该命令将尝试审计所有满足匹配条件的终结点，同时报告对每个终结点的操作结果。媒体网关可以按照任意的先后顺序尝试多个终结点标识符。当采用通配响应时，只需返回一个包含了所有满足匹配的终结点集合的命令响应。采用通配响应可以减少审计多个终结点时产生的消息交互。终结点标识符不允许使用通配符“CHOOSE”

被审计的对象可以是描述符、单个的属性、信号、事件和统计信息。返回的参数可以设置条件进行过滤。

- 如果被审计的对象是描述符，则AuditValue命令所包含的审计描述符应指定需要审计的描述符。
- 如果被审计的对象是媒体描述符中的某个属性，则输入参数中还应包括流标识符，组标识符（GroupID）和属性标识符，其中，组标识符为可选参数。返回的参数是属性的当前值。当本地控制描述符包含“ReserveGroup”标志位时，输入参数应包含组标识符。组标识符等于的“1”表示被媒体网关所预留的第一组资源，组标识符等于的“2”表示被媒体网关所预留的第二组资源，依次类推。

- 如果被审计的对象是信号时，则输入参数应包括信号标识符和信号标识符列表。此时，流标识符或请求标识符为可选参数。仅仅当信号处于激活状态时，信号才可以被该命令被审计，返回的信号参数包括KeepActive标识位、信号类型、持续时间参数、“信号结束”标志位和包定义的相关属性。
- 如果被审计的对象是事件或处于事件缓存队列中的事件，则输入参数应包括事件标识符、流标识符（可选）和请求标识符（可选）。返回的参数可以包括：事件和包定义的相关参数。
- 如果被审计的对象与终结点相关，则输入参数可以包含统计标识符。如果被审计的对象与流有关，则媒体描述符中可以包含流标识符（可选）、组标识符（可选）和统计标识符。返回的参数为参数的当前值。此时，与统计值描述符相关的参数值不会被重置。
- 如果被审计的对象是包，则输入参数应包括包标识符和包的版本信息。所有由该包中定义的属性、事件、信号和统计的当前值将被返回。
- 为了对返回的参数进行过滤，AuditValue命令给包含过滤条件。且该命令可以包含多个过滤条件。当命令中包含多个过滤条件时，命令中还允许出现“AND”或“OR”等逻辑操作。如果命令中不包含逻辑操作，则其默认值为“AND”。

AuditValue命令可以包含多个被审计对象。

如果AuditValue命令中包含审计描述符，则命令返回的审计描述符应包含与终结点相关的当前参数值。如果一个描述符中的某个参数有多个参数值，则应认为该参数值可以支持多个可选值，且描述符中的每个参数相互独立。

被观察事件描述符将返回事件缓存队列中的被缓存的事件。如果被审计的对象是被观察事件描述符，当数字映射处于激活状态时，则返回的被观察事件描述符应包含“数字映射完成”事件用于指示被返回的事情是当前的号码字符串而非终结点的拨号方案。

事件缓存描述符用于返回该描述符允许的事件列表和相关参数。包描述符用于返回终结点实现的包。数字映射描述符用于返回终结点当前的数字映射名称与值。如果AuditValue命令的输入参数包含数字映射且终结点标识符为通配值“ALL”，则该命令应返回媒体网关支持的所有数字映射。统计值描述符用于返回终结点包含的统计参数当前值。如果AuditValue命令中统计值描述符为空，则将只返回被审计的终结点的终结点标识符。通过这种方式，可以获得与通配符匹配的终结点标识符列表。本建议书附件A和B定义了一种特殊的句法规则用于实现上述方式，因此AuditValue命令不必对每一个终结点进行重复操作。

AuditValue命令的返回参数与关联和终结点的值有关，例如“确定”、“空”或“通配符”（要注意，关联标识符“ALL”并不包括空关联）。终结点标识符可以是“确定”或通配符。

关联和终结点都使用通配符时的通配响应如下所示。

假设媒体网关包含4个终结点：t1/1，t1/2，t2/1和t2/2。假设终结点t1/*实现了包aaa和bbb，t2/*实现了包ccc和ddd。假设关联1中包含终结点t1/1和t2/1，关联2包含终结点t1/2和t2/2。

命令：

```
Context=1{AuditValue=t1/1{Audit{Packages}}}
```

响应：

```
Context=1{AuditValue=t1/1{Packages{aaa,bbb}}}
```

命令：

```
Context={AuditValue=t2/*{Audit{Packages}}}
```

响应：

```
Context=1{AuditValue=t2/1{Packages{ccc,ddd}}},
```

```
Context=2{AuditValue=t2/2{Packages{ccc,ddd}}}
```

命令：

```
Context={W-AuditValue=t1/*{Audit{Packages}}}
```

响应：

```
Context={AuditValue=t1/*{Packages{aaa,bbb}}}
```

注一 通配响应也可用于其他命令，如Subtract。

当命令中包含有通配符“ROOT”时，媒体网关控制器指示其期望获得压缩后的关联标识符，而不是单个的关联。

假设媒体网关包含4个关联，分别为1、2、3和4，且每个关联包含两个终结点（t1-t8）。媒体网关处理AuditValue命令的方式将如下所示。

命令：

```
Context={AuditValue=Root{Audit{}}}
```

响应：

```
Context=1{AuditValue=Context{*}}, Context=2{AuditValue=Context{*}},
```

```
Context=3{AuditValue=Context{*}}, Context=4{AuditValue=Context{*}}
```

或

```
Context=1{AuditValue=t1{ },AuditValue=t2{ }},Context=2{AuditValue=t3{ },AuditValue=t4{ }},Context=3{AuditValue=t5{ },AuditValue=t6{ }},Context=4{AuditValue=t7{ },AuditValue=t8{ }}
```

命令：

```
Context={ContextAttr{ContextList={*}},AuditValue=Root{Audit{}}}
```

响应：

```
Context={ContextAttr{ContextList={1,2,3,4}},AuditValue=Root{ }}
```

通过AuditValue命令可对下表所列的内容进行审计：

关联标识符	终结点标识符	审计信息
明确	通配符号	对关联中匹配的终结点进行审计
明确	明确	对关联中的单个终结点进行审计
NULL	根	对媒体网关的状态和事件进行审计
NULL	通配符号	对空关联中所有匹配的终结点进行审计
NULL	明确	对所有关联之外的单个终结点进行审计
ALL	通配符号	对处于非空关联的所有满足通配条件的终结点进行审计
ALL	根	所有关联标识符列表（关联标识符列表由多个动作响应返回，每一个动作响应包含该列表中的一个关联标识符）。返回方式由命令中所包含的关联标识符列表决定。
ALL	明确	该终结点当前所处的非空关联标识符

7.2.6 AuditCapabilities

AuditCapabilities 命令用于获取终结点的属性、事件、信号和统计参数的所有可能值。AuditCapabilities 命令可以用来请求获取描述符中的某一个属性、事件、信号和统计的参数值。

TerminationIDList

[,MediaDescriptor]

[,ModemDescriptor](*)

[,MuxDescriptor]

[,EventsDescriptor]

[,SignalsDescriptor]

[,ObservedEventsDescriptor]

[,EventBufferDescriptor]

[,StatisticsDescriptor]

AuditCapabilities(TerminationIDList,
AuditDescriptor)

(*) H.248.1 (第2版) (05/2002) 不推荐使用Modem描述符。

被审计的对象可以是描述符、单个属性、信号、事件和统计。

- 如果被审计的对象是描述符，则命令输入参数中的审计描述符中应明确指定需要审计的描述符。
- 如果被审计的对象是单个属性，则输入参数应包括流标识符（可选）和属性标识符PropertyID。审计结果将返回所有可能的属性值列表。
- 如果被审计的对象是信号，则输入参数可以包括流标识符（可选）和/或请求标识符（可选）。审计将返回一组与每个信号参数有关的可能值。包含返回参数将不包含KeepActive标志位、信号类型、持续时间参数、信号结束标志位。
- 如果被审计的对象是事件，则输入参数可以包含流标识符（可选）、事件标识符和请求标识符（可选）。审计将返回一组与每个事件参数有关的可能值（包括事件和包定义的相关属性）。

- 如果被审计的对象与终结点相关，则输入参数可以包含统计标识符。如果被审计的对象与流有关，则媒体描述符中可以包含流标识符（可选）、组标识符（可选）和统计标识符。返回的参数为参数的可能值。此时，与统计值描述符相关的参数值不会被重置。

如果输入参数包含一个审计描述符，则AuditCapabilities命令将返回与被审计的终结点描述符及其所有可能值。当描述符存在多个可能值时，该命令中的描述符可以被重复。

如果希望采用通配响应，则媒体网关将仅产生命令响应，且该响应包含了所有与通配符匹配的终结点的参数值。采用通配响应可以减少审计多个终结点时产生的消息交互。

如果被审计的对象是属性、信号、事件或统计参数，则AuditCapabilities命令响应将返回与终结点能力相关的所有属性、信号、事件或统计参数。

当关联标识符和终结点标识符使用类型不同的参数时，被审计的对象要求与AuditValue命令中定义的类型类似。

如果属性和参数值类型为字符串、字符或8字节字符串时，媒体网关返回的响应中参数值必须为空。当采用文本编码时，字符串和字符将返回一个空的“quoted”字符，8字节字符串将返回NULL（0x00）。该处理方式可通过包进行重新定义。

事件描述符可以返回与终结点相关的可能事件列表以及事件描述符参数的所有可能参数值。事件缓存描述符返回的参数与事件描述符类似。信号描述符可以返回适用于该终结点的所有信号及信号参数所包含的所有可能参数值。统计值描述符也可以返回终结点所保存的统计参数。被观察事件描述符用于返回在终结点中处于激活状态的事件。Auditcapabilities命令中不允许使用数字映射与包描述符。

通过AuditCapabilities命令获得的信息如下表所示：

关联标识符	终结点标识符	获得的信息
明确	通配符	对关联中匹配的终结点进行审计
明确	明确	对关联中的单个终结点进行审计
NULL	根	对媒体网关的状态和事件进行审计
NULL	通配符	对空关联中所有匹配的终结点进行审计
NULL	明确	对所有关联之外的单个终结点进行审计
ALL	通配符	对处于非空关联的所有满足通配条件的终结点进行审计
ALL	根	与AuditValue一样
ALL	明确	与AuditValue一样

7.2.7 Notify

媒体网关通过Notify命令向媒体网关控制器报告上报媒体网关内发生的事件。

TerminationID

```
Notify(TerminationID,  
       ObservedEventsDescriptor,  
       [ErrorDescriptor])
```

终结点标识符指示用于指示发起Notify命令的终结点。该标识符必须是一个完全有效的名称。

被观察事件描述符包含请求标识符和事件列表，事件列表中包含的事件按照媒体网关检测到事件发生的先后顺序进行排列。每一个事件还可以包含与该事件相关的参数以及包含事件被检测的时刻等可选参数。请求标识符等于0的流程有待进一步研究。

只有当终结点检测到事件描述符中指定的事件后且事件描述符处于激活状态时，Notify命令中的请求标识符才允许不等于0。

Notify命令中包含的请求标识符应与触发Notify命令的事件描述符中的请求标识符相同。通过这种关联，事件上报和事件请求可以匹配。如果请求标识符不等于0，则事件列表中的事件务必由事件描述符或内嵌的事件描述符来要求进行请求。

当发生518差错时（事件缓存满）时，Notify命令中的输入参数可以包含差错描述符。

7.2.8 ServiceChange

媒体网关通过ServiceChange命令向媒体网关控制器报告一个终结点或者一组终结点将要退出服务或刚返回服务。媒体网关控制器也可以通过ServiceChange命令指示媒体网关应将一个终结点或者一组终结点退出服务或重新返回服务。媒体网关可以通过ServiceChange命令向媒体网关控制器上报其终结点能力发生改变。媒体网关控制器可以告知媒体网关应切换到另一个媒体网关控制器。

TerminationIDList,

```
[ServiceChangeDescriptor]  
ServiceChange(TerminationIDList,  
              ServiceChangeDescriptor  
              )
```

终结点标识符列表用于指定退出服务或返回服务的终结点。除了通配符“CHOOSE”之外，终结点标识符可以任何其他的通配符。当终结点标识符使用“Root”时，ServiceChange命令将作用于整个媒体网关。

注一 在起初的H.248.1建议书第1版中，ServiceChange命令不允许使用终结点标识符列表。

7.2.8.1 业务改变描述符（ServiceChange）基本内容

业务改变描述符可以包含以下参数：

- 业务改变方法（ServiceChangeMethod）；
- 业务改变原因（ServiceChangeReason）；
- 业务改变时延（ServiceChangeDelay）；

- 业务改变地址（ServiceChangeAddress）；
- 业务改变协议子集（ServiceChangeProfile）；
- 业务改变版本（ServiceChangeVersion）；
- 业务改变媒体网关控制器标识符(ServiceChangeMgcId)；
- 时戳（TimeStamp）；
- 扩展参数（ExtensionParameter）；
- 业务改变信息（ServiceChangeInfo）；
- 业务改变未完成标签（ServiceChangeIncompleteFlag）。

7.2.8.1.1 业务改变方法（ServiceChangeMethod）

ServiceChangeMethod参数定义了以下几种业务改变方式：

- 1) 适度的（Graceful）：指示终结点将在ServiceChangeDelay参数设置的时间到达之后退出业务；且已建立的连接将不受影响，然而媒体网关控制器应该避免建立新的连接，且应该试图逐渐拆除命令中指定终结点的已有连接。媒体网关应将ServiceChangeDelay超时之后的终结点或从激活的关联（任意第一个关联）中删除的终结点的业务状态设置为“退出服务”。
- 2) 强制（Forced）：命令指定的终结点将立刻中断服务，且任何与该终结点关联的连接可能会中断。对于一个非Root的终结点，媒体网关控制器将负责删除与该失效终结点相关的任何关联。至少媒体网关控制器必须从关联中删除该终结点。此时，终结点的业务状态应设置为“退出服务”。对于根终结点而言，媒体网关控制器可以假定媒体网关上的所有连接将被中断以及所有终结点将被删除。
- 3) 重启动（Restart）：当ServiceChangeDelay指定的时间到达时，命令中指定的终结点应执行业务恢复。一旦ServiceChangeDelay超时，终结点业务状态应设置为“服务”状态。
- 4) 断开连接（Disconnected）：该方式适用于Root终结点，用来指示媒体网关与媒体网关控制器的通信连接发生中断，但随后连接被重新恢复到同一个媒体网关控制器（媒体网关可能已尝试连接预置列表中的其他媒体网关控制器）。由于媒体网关的状态可能发生改变，因此媒体网关控制器可以使用Audit命令来实现媒体网关与媒体网关控制器之间重新同步。
- 5) 切换（Handoff）：该参数由媒体网关控制器发送给媒体网关，用于指示媒体网关控制器将退出服务，媒体网关应尝试与一个新的媒体网关控制器建立新的连接。当该参数从媒体网关发送给媒体网关控制器时，指示媒体网关正按照之前连接的媒体网关控制器发送来的Handoff参数尝试与一个新的媒体网关控制器建立新的连接。
- 6) 失败（Failover）：该参数由媒体网关发送给媒体网关控制器，指示主控媒体网关将退出服务，备用的媒体网关正开始启动服务。当媒体网关检测到媒体网关控制器发生差错时，该参数可以由媒体网关发送至媒体网关控制器。
- 7) 其他参数值：媒体网关和媒体网关之间相互理解的其他参数值。

7.2.8.1.2 业务改变原因（ServiceChangeReason）

该参数指定已经发生或将要发生的业务改变的原因。该参数包括一个已经在IANA注册的文字和数字符号，以及可选的解释性文字描述。

7.2.8.1.3 业务改变地址 (ServiceChangeAddress) 和业务改变媒体网关控制器标识符 (ServiceChangeMgcID)

业务改变地址参数为可选项，用于指定后续通信的地址(例如IP网络端口号)。该参数可包含在输入或输出描述符中。业务改变地址与业务改变媒体网关控制器标识符参数不允许同时出现在业务改变描述符或其返回值中。业务改变地址参数用于指定一个与当前关联进行协商的媒体网关控制器地址，然而业务改变媒体网关控制器标识符参数用于提供一个媒体网关可以尝试连接的备用地址。本建议书不推荐使用业务改变地址参数。媒体网关和媒体网关控制器务必可以处理包含IP地址或仅包含端口号（例如TCP端口号）的业务改变地址参数。

7.2.8.1.4 业务改变时延 (ServiceChangeDelay)

业务改变时延参数为可选项，单位为秒。如果命令中未包含该参数或该参数为0时，则该参数值应该认为是空。当业务改变方法参数为Graceful时，如果业务改变时延参数为“NULL”，指示媒体网关控制器应等待现有连接自然拆除，且不应建立新的连接。当业务改变方法参数为graceful，业务改变时延参数等于“NULL”意味着媒体网关不允许将终结点业务状态设置为“退出服务”，直到终结点处于空关联中才允许。

7.2.8.1.5 业务改变协议子集 (ServiceChangeProfile)

业务改变协议子集参数为可选项，用于指定协议所支持的任何协议子集。该参数应包含支持的协议子集版本。当该参数空缺时，该参数默认为“无协议子集”。

7.2.8.1.6 ServiceChangeVersion

业务改变版本参数为可选项，包含所支持的协议版本，当发生协议版本协商时，才需要使用该参数（参见11.3）。

7.2.8.1.7 时戳 (TimeStamp)

时戳参数为可选项，用于指定发送方当前的真实时间。该时间并不一定是由本地时区所确定的绝对时间；该参数可以仅仅记录一个会话过程中发送方后续将要发送时戳的任意起始时间。接收方可根据此参数来判断接收方所使用的时间含义与发送方是否不同。时戳参数的精度为百分之几秒。

7.2.8.1.8 扩展参数 (ExtensionParameter)

扩展参数为可选项，可以包含媒体网关和媒体网关控制器可以相互理解的任意参数值。“X-SC”是保留的参数值，适用于H.248.1建议书第3版（及后续版本）所增加的业务改变参数。当业务改变版本大于或等于3时，“X-SC”参数仅仅适用于从媒体网关发送至媒体网关控制器的由H.248.1建议书第1版所编码的第一个业务改变 (ServiceChange) 命令之中。“X-SC”参数值的ABNF结构如下所示：

$$\text{X-SC} = 1 * (\text{NAME EQUAL paramValue [COMMA]})$$

当增加一个新的业务改变参数时，本建议书推荐业务改变参数的名称和参数值都应该在一个被发送的业务改变命令之中予以确定。例如：

Name (名称)： 不大于64个字符长度的字符串，且不允许包含任何空格符，首字符为字母，后续字符可以是字母或数字，以及特殊字符下划线“_”。

类型： 参见12.1.2的“属性 (Properties)”

可能的取值： 参见12.1.2的“属性 (Properties)”

7.2.8.1.9 业务改变信息 (ServiceChangeInfo)

业务改变信息参数为可选项，可以包括引起业务发生改变原因的包、属性、信号、事件或统计。

7.2.8.1.10 业务改变未完成标记 (ServiceChangeIncompleteFlag)

业务改变未完成标记为可选项，用于指示媒体网关后续将向媒体网关控制器发送包含终结点状态的业务改变命令。该参数仅仅适用于在媒体网关注册或重新启动过程中（业务改变命令中终结点为“ROOT”且业务改变方法参数为“Restart”），媒体网关需要上报整个媒体网关以及终结点的状态给媒体网关控制器时。当媒体网关控制器接收到此参数，媒体网关控制器应禁止向媒体网关发送终结点不是“ROOT”的任何命令。如果媒体网关在初始的注册和重新启动过程中发送了该参数，则所有后续的业务改变命令应继续包含此参数，直到媒体网关认为已经将网关状态和终结点状态上报至媒体网关控制器。当网关和终结点状态已经上报至媒体网关控制器之后，该参数不应在此后的业务改变命令中被继续使用。一旦媒体网关控制器接收到不包含该参数的业务改变命令，媒体网关控制器可以继续向媒体网关发送其他命令。如果H.248.1建议书第1版需要使用该参数，业务改变命令可使用扩展参数“X-SC”：

名称： SIC

类型： 布尔型

可能的取值： ON： 标记被包含

注一 由于后续命令将在H.248.1建议书第3版中被编码，所以不需要使用参数值“OFF”。

7.2.8.1.11 ServiceChange命令和响应

当业务改变命令中终结点标识符为“Root”且业务改变方法参数为“Restart”时，指示媒体网关向媒体网关控制器发送注册请求命令。当媒体网关检测到媒体网关控制器发生故障时，媒体网关可以向媒体网关控制器发送终结点标识符为“Root”且业务改变方法参数为“Failover”的业务改变命令。当一个消息中包含的业务改变命令中终结点标识符为“ROOT”且业务改变方法参数为“Restart”或“Failover”时，则该消息不允许再包含任何其他命令，因为这些命令需要使用新的业务改变地址参数所定义的媒体网关控制器地址和协商后的协议版本。

媒体网关应配置一个主控媒体网关控制器和其他几个备用的媒体网关控制器。

当媒体网关接收到业务改变命令的响应消息时，则注册过程相应被完成，除非媒体网关控制器返回一个可选的业务改变媒体网关控制器标识符参数（如下所示）。

媒体网关可以指定业务改变地址参数，该参数被媒体网关控制器所使用而包含在业务改变描述符中的媒体网关可以指定媒体网关控制器所使用的业务改变地址，用于将消息发送至业务改变地址参数所指定的地址。媒体网关可以在业务改变命令请求中指定业务改变地址参数，媒体网关控制器也可以在业务改变响应消息中指定业务改变地址参数。无论采用何种方式，接收方必须使用指定的地址作为所有后续事务处理请求的目的地址。同时，如第9章所示，事务响应和临时响应也必须发送至发起事务请求的地址。即使请求和响应不能被组合在一起发送，也必须遵循此规则。

时戳参数必须被包含在注册请求和响应消息中被发送。

在媒体网关注册阶段，如果媒体网关判定网关已经处于“服务”状态，但是个别终结点尚处于“退出服务”状态，媒体网关可以向媒体网关发送包含“ServiceChangeIncomplete”标记位的业务改变消息。通过使用“ServiceChangeIncomplete”标记位，媒体网关控制器可以获知正处于退出服务状态的终结点。后续的业务改变命令将用于上报这些处于退出服务状态的终结点。

媒体网关控制器可以返回业务改变媒体网关控制器标识符“ServiceChangeMgcID”参数，用于指示媒体网关应该优先选择的媒体网关控制器。此时，媒体网关必须向新的媒体网关控制器重新发送业务改变请求。如果媒体网关控制器指定了业务改变媒体网关控制器标识符媒体网关控制器。当媒体网关控制器向媒体网关发送包含Handoff参数的消息时，业务改变媒体网关控制器标识符参数用于指示将替换现有媒体网关控制器的新的媒体网关控制器。

除非命令中终结点标识符为“ROOT”，业务改变命令的返回参数应该为空。当命令中终结点标识符为“ROOT”时，返回参数可以包含如下参数：

- **ServiceChangeAddress**：如果媒体网关控制器期望为所有来自媒体网关的消息指定一个新的目的地，则应包含ServiceChangeAddress参数。
- **ServiceChangeMgcID**：如果媒体网关控制器不期望与媒体网关保持连接，则应包含ServiceChangeMgcID参数。
- **ServiceChangeProfile**：如果响应方期望与请求方协商协议所使用的协议子集，则应包含ServiceChangeProfile参数。只有当媒体网关控制器无法支持业务改变命令中所指定的协议子集时，ServiceChangeProfile参数才应被返回。返回的ServiceChangeProfile参数值必须指定媒体网关所支持的协议子集和版本，或者当无支持的协议子集时，应返回的参数值为“无协议子集（NoProfile）”。一旦媒体网关接收到协议子集参数之后，应与当前的媒体网关控制器继续保持连接或尝试与备用MGC建议新的连接。如果媒体网关控制器所返回的协议子集与媒体网关所发送的协议子集不同，则媒体网关应遵循如下规则：
 - a) 继续控制协商，向媒体网关控制器重新发送一个新的包含双方一致协议子集的业务改变命令；或者
 - b) 维持现有控制协商，媒体网关控制器将使用其所使用的业务改变响应命令中的协议子集；或者
 - c) 尝试向不同的媒体网关控制器发起一个包含起始协议子集的控制协商。
- **ServiceChangeVersion**：如果响应方期望与请求方协商协议版本，则应包含ServiceChangeVersion参数。

ServiceChangeReasons参数值在ITU-T H.248.8建议书中被定义。在IANA注册的ServiceChangeReasons参数值可以进行扩展，参见14.3。

7.2.9 关联属性的操作和审计

前面几小节讨论了适用于终结点的所有命令。本节将描述对关联进行操作和审计的方式。

一个动作（Action）可以包含其所作用的关联标识符（ContextID）和相关审计内容（参见第8节）。

一个被发送至媒体网关的动作可以包含一个请求用于对关联属性进行审计。审计的类型可以分为两类：

“值审计（Value Audit）”：MGC可以通过审计一个指定的关联以确定该关联当前的属性参数值。MGC可以通过在Audit请求中将ContextID设置为“ALL”来审计所有现有非空关联的当前特性值。如果关联属性被与Audit请求同处一个Action中的其他Add或Modify命令所添加或修改，则返回的参数值应当为Action操作之后的结果。关联属性可以用作选择标准来过滤Audit返回参数值。允许包含多个选择标准。当包含有多个选择标准时，可以用过“或(OR)”或者“与(AND)”逻辑来决定如何处理多个选择标准。如果未指定逻辑操作，则默认的逻辑操作为“AND”。

“能力审计（Capabilities Audit）”：媒体网关控制器可以对某个关联的所有允许值进行统计。通过在Audit命令中包含关联标识符（ContextID）为“ALL”的通配符，媒体网关控制器可以获知整个媒体网关的所有允许值。

通过审计关联，可以获得以下信息：

关联标识符 (ContextID)	终结点标识符 (TerminationID)	AuditValue	AuditCapabilities
指定	不适用	由指定关联所包含的属性当前值	由指定关联所包含的属性允许值
NULL	不适用	不允许	不允许
ALL	不适用	所有现有非空关联的当前值。 Audit请求中包含ContextID为ALL，其响应包含针对每个关联的动作响应	针对ContextID为ALL的Audit请求响应，包含整个网关的所有允许值

一个动作可以包含请求消息来修改关联的属性。

一个动作的响应可包含关联属性，用于将相关信息返回给MGC。这些信息可以是对关联属性进行审计所得到的结果，或者是对关联进行操作后的详细结果。

如果MG接收到一个动作请求，既要对其属性进行审计，也要对这些属性进行处理，则该动作的响应所包含的被审计的属性值必须为该动作被处理之后的属性值。

7.2.10 通用命令句法

H.248.1协议可使用二进制或文本进行编码。媒体网关控制器应该支持两种类型的编码方式，媒体网关可以支持两种类型中的任意一种或全部。

使用二进制编码的协议句法参见本建议书附件A。本地和远端描述符使用二进制编码方式参见本建议书附件C。

使用ABNF的文本编码方式参见本建议书附件B，有关ABNF的内容参见RFC 2234。本地和远端描述符使用二进制编码方式参见本建议书的7.1.8。

8 事务(Transactions)

媒体网关和媒体网关控制器之间的命令被组合成事务，每一个事务可由一个事务标识符(TransactionID)进行标识。事务可以由一个或者多个动作(Action)组成。一个动作由作用于一个关联的一系列非空的命令组成，关联属性的修改和审计都局限于在一个关联内。因此，通常每一个动作使用一个关联标识符。然而，在两种情形中，可以使用不明确的关联标识符。第一种情形是修改关联之外的终结点。另外一种情形是媒体网关控制器要求媒体网关创建一个新的关联。事务、动作和命令之间的关系如图11所示。

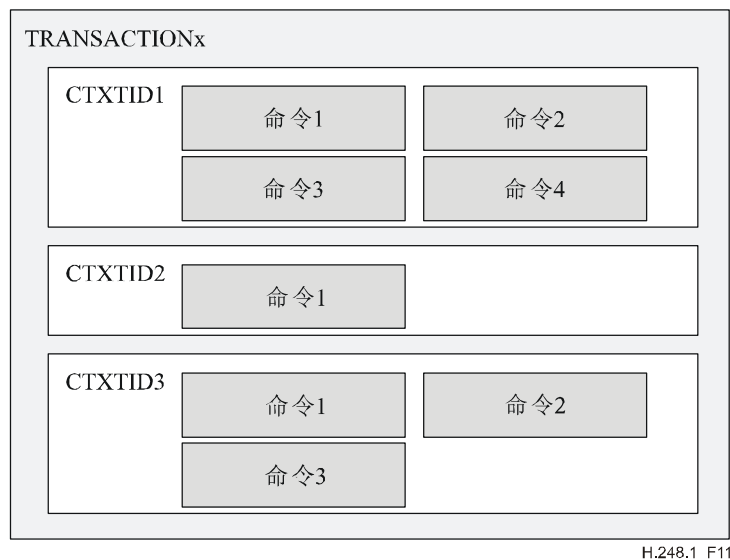


图 11/H.248.1 – 事务、动作和命令

事务由事务请求（TransactionRequest）发起。相应的事务请求响应使用一个独立的响应，在收到响应之前，可以先返回一些事务正在被处理（TransactionPending）消息（参见8.2.3）。

事务保证对命令的有序处理。即一个事务中的命令应按顺序执行。不同事务之间则不保证顺序执行，每个事务可按任意顺序或同时执行；然而，如果消息中同时包含事务请求和事务响应，则要求事务响应于事务请求之前被执行。

如果在事务执行过程之中，有一个命令执行失败，则该事务中所有后续命令都将停止执行。如果命令包含的终结点标识符为通配符，则应对满足匹配条件的每个终结点执行此命令。且应对每一个满足匹配条件的终结点返回事务响应，即使，某个或某些终结点在执行命令时发生差错。如果所有满足匹配条件的终结点都发生了差错，则在此被通配的命令之后的所有命令将不再被执行。

当命令标记为“可选”时，则该命令可以被忽略——例如如果一个标记为“可选”的命令执行失败，则事务中在该命令之后的命令仍可继续执行。如果某个命令失败，则媒体网关 必须在继续处理命令之前尽可能将状态恢复至命令执行失败之前所处的状态。

事务响应包含对事务请求中所有命令的执行结果。事务响应包含执行成功的返回值，以及所有执行失败的命令名和差错描述符。TransactionPending消息可用于周期性地向事务请求发送方报告该事务正在被处理，且事务处理尚未结束。

每个事务应该设置一个应用层定时器。当定时器超时，应该对请求消息进行重传。当接收到响应后，应该停止定时器。当接收到TransactionPending消息时，应该重新启动定时器。

8.1 命令参数

8.1.1 事务标识符

事务由事务标识符进行标识，事务标识符是由事务发起方进行分配且在发送方范围内具有唯一值。如果事务请求中未包含事务标识符，则事务响应返回一个差错描述符用于指示事务请求未包含标识符，且响应方相应所使用的事务响应中事务标识符必须设置为0。

8.1.2 关联标识符

关联由关联标识符进行标识，关联标识符可以由媒体网关分配且应在MG范围具备唯一值。在所有后续与该关联相关的事务中，媒体网关控制器必须使用由媒体网关提供的这个关联标识符。媒体网关控制器可以使用一个特殊的“空（NULL）”关联标识符来标识一个不在任何关联中的终结点。

当关联标识符为通配符“CHOOSE”时，媒体网关被请求创建一个新的关联。

媒体网关控制器可以使用通配符“ALL”来标识所有非空关联的关联。当使用ALL通配符时，将不包含空关联。

媒体网关控制器不允许使用包含“CHOOSE或“ALL”通配符的部分指定的关联标识符。

8.2 事务应用编程接口

事务的应用编程接口如下所示。该应用编程接口仅用于说明事务及其参数，而不是用于指定具体的执行方式（例如，使用闭塞呼叫功能）。该接口将从较高角度来描述可适用于不同事务协议的输入参数和输出参数。事务句法和编码在后续各小节进行定义。

8.2.1 事务请求（TransactionRequest）

事务请求由发送方发起。每一个请求与一个事务相对应。一个事务请求包含一个或者多个动作，每个动作都指定了它的目标关联。

```
TransactionRequest(TransactionID {  
    ContextID {Command ... Command},  
    ...  
    ContextID {Command ... Command}})
```

事务标识符（TransactionId）参数务必为一个确定值，以便将事务请求与接收方发出的事务响应或事务正在被处理（TransactionPending）关联起来。

关联标识符（ContextID）务必为属于同一个关联的所有命令指定一个值，直到指定下一个关联标识符或事务请求结束。

命令（Command）参数可以是本建议书的7.2规定的任意命令（“命令应用编程接口”）。

8.2.2 事务响应（TransactionReply）

事务响应（TransactionReply）由事务请求接收方发起。每一个事情请求与一个事务响应相对应。一个事务响应可以包含一个或者多个动作，其中每个动作都务必指定动作的目标关联，同时事务响应可以包含与目标关联相对应的一个或者多个响应消息。一个响应可以被封装成多个消息。事务响应由接收到事务请求的响应方进行触发。

如果事务请求已按以下方式进行处理，则相应地事务请求被执行完毕：

- 事务请求中所有命令已执行完毕；或
- 除发生差错的命令为可选命令之外，处理事务请求时发生一个差错。

如果消息中所包含的事务请求超过了接收方的处理能力，则接收方将返回差错描述符413(“消息所包含的事务数目超过最大值”)。

当命令中的所有描述符执行完毕时，则命令执行完毕。

对于信号描述符来说，如果该描述符句法描述正确，请求接收方支持信号描述符所指定的信号类型，且指定的信号正准备生效，则信号描述符执行完毕。

对于事件或事件缓存描述符来说，如果该描述符句法描述正确，请求接收方能够检测这些描述符中所指定的事件或内嵌的事件类型，以及能产生内嵌的信号，且请求接收方正处于检测事件状态，则事件或事件缓存描述符执行完毕。

```
TransactionReply(TransactionID {  
    ContextID {Response ... Response},  
    ...  
    ContextID {Response ... Response}})
```

事务响应中的事务标识符 (TransactionID) 参数必须与其相关的事务请求中事务标识符相同。

关联标识符必须为属于同一个动作的所有响应指定一个值。关联标识符可以为“NULL”、“Specific”或“ALL”。

“Response”参数是本建议书的7.2定义的命令返回值。如果某个命令执行失败，将返回差错描述符。同时，该命令所在关联后的其他命令将停止执行，未执行的命令将不产生命令响应。

如果某个命令为可选，当该命令执行失败时，则该事务中该命令所在关联后的所有命令将继续执行，并在发送差错之后继续返回相应的命令响应。

本建议书的7.1.20定义了差错描述符的生成方式。以下是讨论一些差错描述符的示例。

如果请求接收方能够识别接收的关联标识符，但是执行关联标识符时发生差错，则返回的关联响应中将包含执行失败的关联标识符和一个差错描述符，其中差错代码为422(动作发生句法差错)。如果请求接收方无法识别所接收的所有关联标识符，则事务响应将包含事务标识符和一个差错描述符，其中差错代码为422(动作句法差错)。

如果请求接收方在执行一个事务时无法确认某个动作是否合法，则返回的事务响应中将包含事务标识符和一个差错描述符，差错代码为422(动作句法差错)。如果请求接收方无法确认某个动作的最后部分是否合法，但是动作可以分解执行，则请求接收方在执行动作中发生差错之前的部分之后，返回差错代码422(动作句法差错)作为该事务的最后响应。如果请求接收者无法确认所接收的事务是否合法，则返回一个事务标识符为空的事务响应和一个差错描述符，差错代码为403(事务请求句法差错)。

如果请求接收方无法确认某个事务的最后部分是否合法，但是可以识别一个或多个动作，则接收方在处理可以识别的动作之后，返回一个差错代码403(事务请求句法差错)作为该事务的最后一个动作响应。如果接收方无法识别任何一个动作，则仅返回差错代码403(事务请求句法差错)。

如果请求接收者无法确认所接收事务中的某个终结点标识符是否合法，则返回的关联响应将包含一个差错描述符，差错代码为442（命令发生句法差错）。

如果请求接收方无法确认某个命令的最后部分是否合法，则返回一个包含差错描述符的命令响应作为该动作的最后响应，差错代码为442（命令发生句法差错）。

8.2.3 事务正处理（TransactionPending）

TransactionPending 消息由接收方发送。该消息用于表示一个事务正在被处理，然而事务处理还未最后完成。发送该消息的目的是用于当一个事务正在被处理且还需一些时间才能执行完成时，以避免发送方认为事务请求已经丢失。

TransactionPending(TransactionID { })

事务标识符（TransactionID）务必与事务请求中的事务标识符相同。当根终结点的“NormalMGExecution”属性由媒体网关控制器设置时，该参数用于指示在指定的时间间隔内媒体网关控制器应该期望从媒体网关接收到任何一个事务的事务响应。另外一个根终结点的“NormalMGCEExecution”属性由媒体网关控制器设置时，该参数用于指示在指定的时间间隔内（网路时延除外）媒体网关期望从媒体网关控制器接收到任何一个事务的事务响应。“MGProvisionalResponseTimer”参数用于指示在一段时间间隔内，如果事务请求处理尚未完成，则媒体网关控制器期望从媒体网关接收到事务正处理消息（该参数值可以设置为“NormalMGExecution”参数值和网络时延的总和或者稍小于其总和）。“MGProvisionalResponseTimer”参数对于媒体网关的意义类似。事务发送方可能会接收到一个事务的多个事务正处理（TransactionPending）消息。当接收方处于“事务正处理状态”时，接收到一个重复的事务请求时，则接收方应立即发送一个重复的事务正处理消息，或者等待定时器超时后发送另外一个事务正处理消息。

根终结点的“MGOriginatedPendingLimit”属性由媒体网关控制器设置，用于指示媒体网关控制器可以接收的来自媒体网关的事务正处理消息的数目。当超出该参数值时，媒体网关必须停止事务处理，并返回一个事务响应，否则，媒体网关控制器将判断事务处理发生故障。

根终结点的“MGCOrganatedPendingLimit”属性由媒体网关控制器设置，用于指示媒体网关可以接收的来自媒体网关控制器的事务正处理消息的数目。当超出该参数值时，媒体网关控制器必须停止事务处理，并返回一个事务响应，否则，媒体网关将判断事务处理发生故障。

由于命令处理时间过长或差错发生（如命令被环回），可能导致MGOriginatedPendingLimit 和MGCOrganatedPendingLimit属性发生超出门限。此时，事务请求的接收方应返回一个事务响应，且包含差错描述符和相应的消息响应参数“命令过长”或“命令导致差错”。该事务的后续命令必须被停止处理。此时，必须返回差错代码 506（事务正处理消息数目超出）。

注 — 为了防止由于差错而导致MGOriginatedPendingLimit 和MGCOrganatedPendingLimit参数值超出门限值，以及为了阻止事务请求的接收方持续发送事务正处理消息，事务请求的接收方应该启动管理保护机制来启动相应的防御操作。事务请求的发送方应保存接收到的事务正处理消息的数目并启动校正操作。

8.3 消息

消息可以由多个事务组成。每个消息都有一个消息头，用于包含消息发送方的身份标识。消息标识符(MID)可以设置为消息传送实体的指定名称（例如域名地址/域名/设备名）。默认值将使用域名。在媒体网关控制器和媒体网关控制联系期间，一个H.248.1实体（媒体网关或媒体网关控制器）务必将所有其发起的消息使用相同的消息标识符。

每一个消息应包括一个版本字段用于标识消息所参照的协议版本。版本字段为1位或2位数字，起始版本为1。本建议书所定义的协议版本为3。

消息中所包含的事务相互独立。本建议书对事务执行顺序不进行具体规定；本建议书未规定消息的协议证实机制。消息实质上只是一种传输机制。例如，消息X包括事务请求A、B和C，则消息X的响应可以是Y和Z，Y包含对A和C的应答，Z包含对B的应答。同理，消息L包括事务请求D，消息M包括事务请求E，其响应可以是消息N，且N同时包含对D和E的事务响应。

9 传送

本建议书所规定的协议传送机制应该支持在媒体网关和媒体网关控制器之间所有事务的可靠传送。传送机制必须与正在被传送的具体命令无关，且必须适用于所有应用状态。协议可以采用几种方式进行传送，分别定义在本建议书的附件或H.248系列建议书（如H.248.4和H.248.5建议书）。其他可选的传送机制可以定义在其他H.248系列建议书中。如果协议采用IP进行传送，则媒体网关控制器必须支持实现TCP和UDP/ALF，媒体网关必须支持实现TCP或UDP/ALF，或者同时支持两者。

媒体网关应为主控MGC或备用MGC（参见7.2.8）配置一个名字或者地址（例如域名或IP地址），该地址是媒体网关用于向媒体网关控制器发送消息的对端媒体网关控制器的地址。如果传送层协议采用的是TCP或UDP，当消息发送方无法通过其他方式获知初始ServiceChange请求应发送到哪个端口，则消息发送方应该将该请求发送至默认的协议端口。如果传送层协议采用的是TCP或UDP，对于文本编码的消息，默认的协议端口为2944，对于二进制编码的消息，默认的协议端口为2945。媒体网关控制器接收到来自媒体网关的ServiceChange请求消息之后，应当能够从中解析出媒体网关的地址。如本建议书的7.2.8所示，媒体网关和媒体控制器都可以在ServiceChangeAddress参数中提供一个地址，以便后续的事务请求都务必发送至该地址，然而，所有请求的响应（包括对初始的ServiceChange请求的响应）都务必发送至请求的源地址。例如，在IP网络中，IP头消息包含源地址且TCP/UDP/SCTP消息包含源端口。

9.1 命令顺序

本建议书不强制底层协议确保事务的顺序发送。该方式有利于实现事务中命令的及时处理，却具有如下缺陷：

- 当一个包含已改变的事务描述符的命令被传输后，可能导致Notify命令被延迟发送和延迟到达MGC；
- 当在命令未确认之前发送一个新的命令，不能确保先前的命令在新命令之前被正确执行。

为了确保MGC与MG的一致操作，建议书采用以下规则。以下规定仅与处于不同事务中的命令有关。同一个事务中的命令应被顺序执行（参见第8节）。

- 1) 当MG需要对多个终结点进行处理时，与不同终结点相关的命令可以同时发送，例如，每一个终结点或一组终结点都分别由一个与之相对应的进程或线程进行处理。
- 2) 除非这些主要命令不包含在同一个事务中，通常一个终结点上应该最多只有一个主要命令（如Add、Modify或Move）。然而，Subtract命令可以在任何时刻发送给终结点。因此，当MG接收到一个Modify命令作用于先前已被删除的终结点时，MG应该忽略该Modify命令并返回一个包含差错代码的响应。

- 3) 对于不保证顺序传输消息的传送协议（如UDP），则任意时刻上在指定的终结点上应该至少有一个重要的Notify命令。
- 4) 有时，当发送Add命令之前，可能有一个包含显式或隐式通配符的Subtract命令作用于一组终结点时，则MG应该在Add命令执行之前删除subtract命令中包含的所有终结点。同时，在包含通配符的Subtract命令未被确认之前，不应发送包含通配符（或通配符隐含在MuxDescriptor中）的新的Add命令。
- 5) AuditValue和AuditCapabilites命令可以不按顺序任意发送。
- 6) 参照MG重启动流程，MG进行重启动时必须首先向MGC发送ServiceChange命令。任何其他的命令或响应务必在ServiceChange命令之后发送。

以上规则不适用于命令响应，每一个命令应该返回命令响应。

9.2 预防重启动雪崩

大量媒体网关同时加电重启动时，将同时引发大量的ServiceChange注册流程，从而在业务重启动期间导致媒体网关控制器的消息处理流程发生崩溃，消息丢失和网络拥塞。为避免重启动雪崩，本建议书采用以下规则进行处理：

- 1) 当媒体网关加电重启动时，应该启动一个重启动定时器，并将该定时器值初始化为0和最大等待时延(MWD)之间的一个随机值。当多个媒体网关使用相同的重启动定时器值生成算法时，应该避免媒体网关之间所生成的重启动定时器随机值出现同步。
- 2) 应该等待重启动定时器超时或者检测到某一事件发生，例如媒体网关检测到用户摘机事件，媒体网关才发起重启动流程。
- 3) 当重启动定时器超时或者检测到某一事件发生，媒体网关应该发起重启动流程。

重启动过程仅要求媒体网关控制器从媒体网关接收的第一个消息是通知媒体网关控制器有关重启动的ServiceChange消息。

注 — MWD参数是MG中与网关类型相关的配置参数。驻地媒体网关（RG）的MWD可以参照以下因素进行计算。

通常媒体网关控制器的规模应满足处理每小时话务负荷峰值的需求。在负荷峰值期间，平均有10%的模拟电话线用户处于忙状态，且平均呼叫时长为3分钟。通常媒体网关控制器与媒体网关之间完成一次呼叫需要5-6个事务交互。简单估算，平均每三十分钟媒体网关控制器需要为每个终结点处理5-6个事务，或者平均每5-6分钟为每个终结点处理一个事务。因此，建议RG的MWD值为10-12分钟。当媒体网关未明确定义MWD值时，MWD应该选择默认值600 s（10分钟）。

对于中继网关（TG）或商务网关（Business Gateway）而言，MWD值应该更小，因为这些网关需要处理的终结点数量更多，并且在忙时峰值期间，终结点的利用率也要大于10%，通常为60%。因此，在忙时，媒体网关控制器的处理能力将按每分钟为每个终结点处理一个事务。因此，中继网关中每条中继线的MWD值应比驻地网关中模拟电话线的MWD值小6倍，并且与正在重启动的终结点数量成反比。例如，处理T1中继线的中继网关的MWD值应该设置为2.5秒，则处理T3中继线的MWD值则被设置为60毫秒。

9.3 预防Notify雪崩

由于事务处理发生故障或者媒体网关在非常短的时间内检测到大量的事件发生，导致媒体网关需要同时发送的大量的Notify消息，媒体网关应按照一种严格的方式来发送Notify消息直到这种雪崩效应已经缓解。

10 安全考虑

本节对在IP网络中传送本建议书所规定协议的安全性进行讨论。

10.1 协议连接的保护

为了防止未经授权的实体利用本协议建立非法呼叫，或者干预合法呼叫，显然需要一种安全机制。当在IP网络上传输本协议时，可使用IPsec安全机制（RFC 2401至RFC 2411）。

认证头（AH）协议（参见RFC 2402）为媒体网关和媒体网关控制器之间传送的消息提供数据源认证，无连接完整性保护和可选的抗重放保护。封装安全净荷（ESP）协议则对媒体网关和媒体网关控制器之间传送的消息提供加密，ESP协议内容参见RFC 2406。例如，当SDP用于携带会话密钥时，应该使用ESP加密机制。

采用ESP协议实现本协议必须遵循RFC 2406第5节，其中定义了完整性检查和加密的算法最小集合。采用AH协议实现本协议必须遵循RFC 2402第5节，其中定义了采用手工密钥实现完整性检查的算法最小集合。

本协议规定应该使用密钥交换（IKE）协议来提供更有效的加密手段。实现IKE协议应该支持RSA签字和RSA公开密钥加密。

10.2 过渡性AH机制

实现IPSec时要求在IP头后面紧跟AH头或ESP头。然而，在实际应用中难以实现该方式。因此，当底层协议不支持IPsec时，这给协议实现带来的困难。

作为一种过渡性方案，本建议书定义了一个可选的AH头字段。这个AH头中的各个域的定义与在RFC 2402中定义的SPI、SEQUENCE NUMBER和DATA域的定义完全一致。除完整性检验值（ICV）的计算有所不同之外，AH头的句法与RFC 2402中定义的传输模式的语义完全相同。在IPsec中，应该对包含IP头在内的整个IP包进行ICV计算。该计算方式可以有效防止地址欺骗。为了实现相同的功能，本建议书应该根据消息中所有事务以及包含32比特的源IP地址，32比特的目的IP地址的IP头和16比特的UDP目的端口号来完成ICV计算，以上信息将编码为20个16进制的数。当过渡型AH机制在TCP上实现时，仅仅是计算时将UDP端口变换为TCP端口，其他规则与UDP完全一样。

当底层协议支持IPsec时，本H.248.1协议必须支持在Ipsec上传送。当底层协议使用IPv4时，必须支持过渡的AH机制。然而，当底层协议支持IPsec时，则不允许使用过渡性AH机制。例如，Ipv6支持IPsec时，不允许采用过渡性AH机制。

过渡性AH机制的实现必须遵循RFC 2402第5节的相关内容，其中定义了采用手工密钥实现完整性检查的算法最小集。

由于这种过渡AH机制无法防止窃听，因此应该禁止第三方监视连接到指定的终结点之上。这种过渡AH机制无法提供抗重放攻击，该过渡AH方案不必保护由于媒体网关或媒体网关控制器伪装而进行的DoS攻击。然而，此过渡AH机制可以提供一种伪装实体的识别机制，从而通过维护程序取消对非法实体的授权。

10.3 保护媒体连接

媒体网关控制器可以向媒体网关提供会话密钥，用来对音频消息进行加密来防止窃听。

分组网络的一个缺陷是非受控侵入。非法实体可以通过向一个指定的IP地址和UDP端口发送IP包来发起攻击。如果无相应的保护措施，则这些入侵的分组包将被解码从而导致在线路端产生“干扰信号”。

针对这种攻击的一种保护方法是仅接受来自获得授权实体的分组，检查其源IP地址和UDP端口号是否和远端描述符指定的IP地址和UDP端口号一致。然而这种保护方式可能会导致连接建立的延迟和接收方受到地址欺骗的影响：

- 为了允许基于地址的保护，媒体网关务必从输出端的媒体网关获得远端描述符，并将其传送至输入端的媒体网关。因而，这将导致出现网络往返时延，使处理变得困难；或者允许无需等待来自对端的远端描述符就进行呼叫，例如，“删除”远端描述符通知流程，或者等待来自对端的远端描述符之后，开始慢速呼叫建立流程。
- 如果攻击方可以获得合法的源地址和端口，则源地址欺骗才有效，例如，攻击方可通过监听流量获得源地址和端口。为了避免源地址欺骗，应对所有网络接入点进行控制，然而这种方式在实际运用中非常难以操作。

另一种源地址检查方法是在呼叫建立过程中传送密钥，该密钥用于分组进行加密和认证。采用这种方法不会导致连接建立的延迟且可以有效地防止地址欺骗。

11 MG-MGC控制接口

当媒体网关冷启动时，应对媒体网关与媒体网关控制器之间的控制连接进行初始化，此时，ServiceChange消息可作为初始化请求，然而，媒体网关与媒体网关控制器之间的控制连接可能随着后面发生的事件改变，诸如MG/MGC故障或人为操作。

注一 尽管协议未明确规定相关机制来支持多个媒体网关控制器控制一个物理媒体网关，然而协议可实现一个物理媒体网关内支持多个虚拟媒体网关，且多个虚拟媒体网关可与不同的媒体网关控制器建立控制连接。

11.1 多个虚拟MG

一个物理媒体网关可以分割成一个或多个虚拟媒体网关。一个虚拟媒体网关由一个静态分割的物理终结点集合和/或临时终结点集合构成。一个物理终结点可由一个媒体网关控制器控制。在这种模型下，除了终结点之外，其他资源不能被静态分配。为虚拟媒体网关分配终结点是一种管理手段，不在本建议书范围内进行定义。每个虚拟媒体网关对媒体网关控制器表现为一个完整的媒体网关。

一个物理媒体网关可以只有一个网络接口，则多个虚拟媒体网关必须共享这一个物理接口。此时，分组网络侧的终结点必须是共享的。应该指出的是，这种共享接口要求对每个流都能创建一个临时的终结点，从而直接共享终结点。这种机制有时会变得相对复杂，例如当网络接口有事件发生时，媒体网关务必知道需要将发生的事件通知给哪一个控制它的媒体网关控制器。

一般地，虚拟媒体网关受控于媒体网关控制器来创建网络流（如果MG为发起端）或等待流请求（如果MG为终结端），从而避免混淆。然而，当未期望的事件发生时，虚拟媒体网关必须知道如何对它所控制的物理资源进行操作。

当事件恢复后需要改变物理接口的状态时，应该只允许一个媒体网关控制器进行操作。这些问题可以通过下列方式来解决，即允许任何一个媒体网关控制器创建事件描述符，其中包含仅能被通知的事件，然而仅有一个媒体网关控制器具备对物理接口属性进行读/写操作的权限，而所有其他的媒体网关控制器则只具备读操作权限。此时，应通过管理机制指定哪一个媒体网关控制器具备读/写操作权限，且指定具备读/写操作权限的媒体网关控制器为主控MGC。

每个虚拟的媒体网关都具有一个根终结点。大多数情况下，根终结点的特征值可以由每个媒体网关控制器独立设置。当参数只能有一个取值时，除了主控MGC可以对该参数值进行读写操作外，其他MGC都只能进行读操作。

ServiceChange命令允许仅应用于一个虚拟媒体网关内或由虚拟媒体网关所创建的一个终结点或一组终结点（临时终结点）。

11.2 冷启动

一个媒体网关采用管理机制来配置一个主控MGC和其他备用MGC不在本建议书讨论范围之内。当媒体网关冷启动时，该媒体网关将向主控MGC发送终结点标识符等于Root，ServicechangeMethod参数等于“Restart”的ServiceChange消息。如果媒体网关控制器接受该媒体网关，则它将返回一个不包含ServiceChangeMgcID参数的事务响应。如果媒体网关控制器不接受媒体网关的注册，它将返回一个事务响应，其中包含ServiceChangeMgcId参数提供下一个可供联系的MGC地址。

如果媒体网关接收到一个含有ServiceChangeMgcID参数的事务响应，则将发送ServiceChange至该参数指定的MGC。媒体网关继续该过程直到获得某个MGC接受其注册或无法获得事务响应。当媒体网关无法从主控MGC或任何指定的后续MGC获得事务响应时，媒体网关将对其预先提供的备用MGC进行尝试。如果媒体网关无法与任何一个MGC建立控制连接，则媒体网关必须在等待一个随机时间后(参见9.2)，然后开始重新连接其主控MGC，如果有必要，可以再次连接其备用MGC。

ServicechangeMethod参数等于“Restart”的ServiceChange消息响应可能发生丢失，且媒体网关可能在接收到ServiceChange响应之前接收到其他命令，此时，媒体网关必须返回包含Error描述符的响应，差错代码为505（在接收到重启动响应之前接受到命令）。

11.3 协议版本协商

媒体网关发送的第一个ServiceChange命令，必须包含ServiceChangeVersion参数用于指定该媒体网关所支持协议的版本号。不管ServiceChangeVersion参数设置的版本值是多少，消息的命令必须被编码为版本1的消息。当接收到版本为1的消息时，如果媒体网关控制器仅能支持低版本协议，则媒体网关控制器必须在ServiceChange响应中返回其所支持的协议版本，且MGC和MG之间传送的所有消息必须遵循较低版本的要求。如果媒体网关无法遵循且MG已经建立和MGC之间的网络层连接，则媒体网关应关闭其连接。媒体网关应该拒绝随后来自媒体网关控制器的所有请求，并返回一个差错响应406（版本不支持）。

如果媒体网关控制器支持的版本只能高于媒体网关支持的版本，则媒体网关控制器必须拒绝所有请求，且返回一个406差错响应（版本不支持）。

如果媒体网关控制器支持媒体网关指定的版本，则所有后续消息必须遵循该版本。可选地，媒体网关控制器可以在ServiceChange响应中返回其所支持的版本信息。

当ServiceChangeMethod等于“handoff”或“failover”的，也可能发生协议版本协商。

当使用新版本将旧协议进行升级时，应该遵循下列规则：

- 1) 现有的协议单元（例如程序、参数、描述符、特性和值）不应该被改变，除非需要校正协议差错或为了支持新协议需要改变业务的操作。
- 2) 不应该改变命令、参数、描述符、属性和值的语义。
- 3) 不应该修改消息和参数的格式和编码规则。
- 4) 当协议中的信息单元过时，可将这些信息单位标识为不可用。然而，信息单元标识符应标识为保留值，且新版本中不冷使用这些被保留的标识符。

11.4 MG故障

当MG发生故障时，如果MG还可以向MGC发送消息，则媒体网关将发送ServiceChange消息，其中包含ServiceChangeMethod参数（“Graceful”或“Forced”），且终结点标识符等于root。当MG故障排除时，MG将发送ServiceChangeMethod等于“Restart”的ServiceChange消息。

本建议书允许MGC能够向一个发生故障而进行冗余切换的一对MG重复发送消息。此时，必须只允许正常工作的MG能够接收或拒绝来自MGC的事务消息。当主控MG发生故障时，它将发送ServiceChangeMethod参数等于“Failover”的ServiceChange命令，差错原因为“MG临近故障”。随后，MGC会使用备用MG作为一个激活的MG。当故障排除后，工作正常的MG将发送一个ServiceChangeMethod等于“Restart”的”ServiceChange”消息。

注 — 由于本建议书未规定任何机制使得MGC可以向使用ALF的备用MG发送证实消息，因此冗余备份的MG需要使用可靠的传送机制。

11.5 MGC故障

当MG检测到控制它的MGC发生故障时，MG就会试图联系其预先分配表上的下一个MGC。MG从分配表首部（即主控MGC）开始尝试，除非该MGC出现故障，MG才尝试与第一个备用MGC进行连接。此时，MG应向备用MGC发送ServiceChangeMethod参数等于“Failover”的ServiceChange消息，原因为“MGC临近故障”。如果MG不能与任何一个MGC建立控制连接，则MG必须在等待一段随机时间后（参见9.2），开始重新尝试其主控MGC，如果有必要，尝试其备用MGC。当MG与原控制MGC建立连接时，MG应向MGC发送ServiceChangeMethod参数等于“Disconnected”的ServiceChange消息。

当MGC发生部分故障时或者由于人工维护的原因，MGC会指示其所控的MG去联系另外一个MGC。此时，MGC必须向MG发送ServiceChangeMethod参数等于“Handoff”的ServiceChange消息，且ServiceChangeMgcID参数应指定可替换的MGC。如果MG支持“Handoff”，MG应向指定的MGC发送ServiceChangeMethod参数等于“Handoff”的ServiceChange消息，且Reasons为“MGC指定改变”。如果MG未能从指定MGC获得响应，MG必须认为MGC发生故障，开始尝试备用MGC。若MG不能与MGC建立控制连接，MG必须在等待一段随机时间后（参见8.2），尝试连接主控MGC，如果有必要，尝试连接备用MGC。

本协议对MGC如何进入“Handoff”管理状态未作规定，该内容不在本协议的讨论范围内。当切换发生时，MGC和MG可以使用以下规则。当MGC发生切换时，MGC切换操作对于MG来说应该是透明的。事务可以按任何顺序来执行，且执行ServiceChange时事务也可以同时执行。因此，发生MGC切换时，MG中正在执行的命令应继续执行，所有对来自切换前MGC的命令的响应务必返回至切换前的MGC。如果MG和先前的MGC之间的连接已经被中止，则不应该再发送命令响应。MG可能从新的MGC接受到事务响应。在控制连接建立之前，必须不允许MG将新的消息发送至新的MGC。重复发送的事务请求必须直接发送至新的MGC。MG必须保持所有终结点和关联的状态。

当一个发生故障的MGC由一个正常工作的MGC取代时，新的MGC的标识可能与发生故障的MGC相同。此时，ServiceChangeMgcID可以指定为先前的MGC标识符，且MG必须按照如前所述MGC值已经发生改变的原则，发送ServiceChange消息。

对于可以进行故障冗余切换的MGC，可通知其受控的MGC按照如前所述的规则可以进行切换。

11.6 MGC-MG控制联系监控

监控媒体网关和媒体网关控制器之间的连接状态对于一个高可靠性的网络来说是非常必要。通过对媒体网关和媒体网关控制器之间的链路状态进行连续检测可以达到此目标。许多底层传送机制已经提供了相应的检测功能，因此，从协议层面进行的检测是不必要的。

对于不提供链路检测的底层传送网络，链路检测功能通过 H.248建议书所规定的现有消息就可以简单实现。当媒体网关和媒体网关控制器之间的链路上没有任何H.248消息时，媒体网关控制器可以通过向媒体网关发送基于“ROOT”的包含一个空的 Audit描述符的AuditValue 命令来检测链路是否发生中断。媒体网关通过ITU-T H.248.14建议书中所制定的“休眠定时器(Inactivity Timer)”包就可以实现链路中断检测功能。

12 包定义

本建议书所规定协议的主要扩展机制是通过包来实现。包对终结点上相关的附加属性、事件、信号和统计进行了定义。

IETF扩展的包类型可由独立的RFC文稿进行定义。

ITU-T扩展的包类型可在其他相关建立文档（如H.248系列建议书）中进行定义。

- 1) 应规定一个公开文档或标准论坛文档，用于按照上述知道原则描述包。
- 2) 文稿必须指定其所描述的包的版本。
- 3) 文档应该可以通过公开Web地址获得，且应具备一个固定的URL。该地址应该可以提供一种可以提交意见且对意见进行反馈的机制。

12.1 定义包的指导原则

包是用于对属性、事件、信号和统计进行定义。

包可以根据14.2制定的规则定义一个新的差错代码。为了方便文件管理，包文档提交给IANA的同时应将差错代码进行注册。如果包内容进行了修改，如果差错代码描述本身未进行修改，则不需向IANA提交一个包含差错代码的新的参考文档。

包必须由包标识符PackageID（用于唯一标识一个包类型）和其他参数标识符（包中定义的唯一参数标识符）进行标识。当采用文本编码时，引用包中的参数，必须使用“/”字符将PackageID和参数标识符进行隔离。例如：Togen/playtone是采用文本编码，表示信号是信号音生成包（Tone Generation）中的Playtone播放信号音信号

包定义包含以下部分和关键字（标记为“黑体”文字）。定义包的模版参见本建议书附录二。

12.1.1 包（Package）

包的基本描述，包括：

包的名称：简单说明

PackageID：是一个标识符

描述：对包做出描述

版本：

一个新版本的包只能添加可选的属性、事件、信号和统计以及在前一个版本中定义的现有参数的新参数值。包不允许修改和删除属性、事件、信号和统计以及参数。版本号用1至99之间的整数标识。

是否仅用于扩展（可选项）：是

该部分表示此包不能被直接引用，仅可被扩展成其他包。例如，此包可能不包含任何功能或完全无意义。因此，当MG需要上报其所支持的包类型时，MG不应该包含此包标识符。

扩展：现有的包标识符和版本

包可以在现有包基础上进行扩展。扩展时务必指定被扩展的包的版本。当一个包在另一个包基础上进行扩展时，只允许扩展新的属性，事件，信号和统计，或对现有包定义的属性，事件，信号和统计扩展新的参数值，或对其中的已有参数扩展新的允许值。不允许扩展包对扩展之前的包或其他扩展包（或多级扩展包）的参数标识符进行重定义。例如，当版本为1的包B是对版本为1的包A的扩展，如果版本为2的包A已经引用了版本为1的包B中的某个参数标识，则版本为2的包B不能对版本为2的包A进行扩展。如果包不是任何其他包的扩展包，该部分应该指定为“None”（无）。

12.1.2 属性（Properties）

由包定义的属性参数包含以下内容：

属性名称：简单说明

PropertyID：是一个标识符

描述：对属性的功能做出描述

类型： 包括以下类型：

布尔型

字串：UTF-8字串

八位组字串：若干八位组。其编码见附件A和B.3。

整数：4字节有符号整数

双精度：8字节有符号整数

字符：单个字母的Unicode UTF-8字符编码。可多于一个字节。

枚举：唯一允许值列表中的一个值。

子列表，包含列表中若干参数值的列表，且必须说明子列表的类型。除子列表类型之外，子列表的类型必须是本节定义的的数据类型的一种。例如，类型为枚举的子列表。子列表的编码见附件A和B.2。

可能的取值：

包务必明确指定属性的一组取值以及确定取值的方式。当某个描述符中省略了属性参数值时，包务必指定一个默认值或默认行为。例如，当描述符中省略属性参数值时，与该属性相关的流程应暂停处理。

默认值：

一个默认值（不包括默认程序）可以指定为“可临时选择的”。

定义于：

对属性做出规定的H.248.1描述符。LocalControl属性为与特定媒体流相关的属性，TerminationState是与特定媒体流无关的属性。ContextAttribute属性是反应整个关联的属性，即混和属性。这是最常见的情形，但属性也可以在其他描述符中做出规定。关联属性必须在ContextAttribute描述符中做出规定。

属性： 只读或读/写，以及（可选项）全局

表明属性为只读还是读/写，以及是否为全局的。如果宣布某个属性是全局的，则该属性的值由实现该包的所有终结点所共享。如果宣布某个关联属性是全局的，则该属性由实现该包的所有关联所共享。

12.1.3 事件（Events）

由包定义的属性参数包含以下内容：

事件名称： 简单说明

EventID： 是一个标识符

描述： 对事件的功能做出描述

EventsDescriptor（事件描述符）参数：

该参数包含在事件描述符中（参见12.2），被MGC用于配置检测的事件。如果事件描述符中未包含任何参数，则应视为“none”（无）。

ObservedEventsDescriptor（被观察事件描述符）参数：

这些参数包含在被观察事件描述符中（参见12.2），用于MG向MGC发送的Notify命令，或者当MGC发送的审计请求中包含被观察事件描述符时，这些参数应包含在对MGC发送的审计请求响应中。如果被观察事件描述符未包含任何参数，则应视为“none”（无）。

12.1.4 信号 (Signals)

由包定义的信号包含以下内容:

信号名称: 简单说明

SignalID: 是一个标识符。SignalID用于信号描述符中。

描述: 对信号的功能做出描述

Signal类型: 包括以下类型:

OO (开/关)

TO (超时)

BR (简约)

注 — 信号类型可以与一个或多个参数值无关。包必须定义默认的信号类型。如果默认信号类型为TO, 则包必须指定信号的默认持续时间。默认持续时间对于信号类型为BR的信号来说无任何意义。

时长: 单位为百分之一秒

附加参数: 见12.2

12.1.5 统计值 (Statistics)

由包定义的统计值包含以下内容:

统计值名称: 简单说明

StatisticID: 是一个标识符

StatisticID用于统计值描述符中。

描述: 对统计值做出的描述

类型: 包括以下类型:

布尔型

字串: UTF-8字串

八位组字串: 若干八位组。其编码见附件A和B.3

整数: 4字节有符号整数

双精度: 8字节有符号整数

字符: 单个字母的Unicode UTF-8字符编码。可多于一个字节。

枚举: 唯一允许值列表中的一个值。

子列表, 包含列表中若干参数值的列表, 且必须说明子列表的类型。除子列表类型之外, 子列表的类型应该是本节定义的的数据类型的一种。例如, 类型为枚举的子列表。子列表的编码参见附件A和B.2。

可能的取值:

在该单元或上述类型单元部分中, 包必须指定度量单位, 例如毫秒、包数目及其相关的范围限制。

等级: 指定该统计值可以维持在终结点层面、流层面或者两者之一。

12.1.6 差错代码 (Error Codes)

如果包未定义任何差错代码，则该部分可以省略。反之，该部分用来描述由包所指定的差错代码：

差错代码 #： 差错代码的编号。

名称： 差错名称

定义： 对差错代码做出的描述

差错描述符中的差错文本：

对差错描述符中将返回什么样的差错文本做出描述。

注释： 有关差错代码使用的任何进一步说明。

12.1.7 程序 (Procedures)

关于包的用途的附加说明。

12.2 事件和信号参数定义的指导原则

参数名称： 简单说明

ParameterID： 参数标识符，标识一个参数。当采用文本编码时，不允许事件和信号参数的参数标识符以“EPA”和“SPA”字符开始，且不允许参数标识符为“ST”，“Stream”，“SY”，“SignalType”，“DR”，“Duration”，“NC”，“NotifyCompletion”，“KA”，“KeepActive”，“EB”，“Embed”，“DM”，“DigitMap”，“SPADI”，“SPADirection”，“SPARQ”或“SPAResponseID”。

描述： 对参数的功能做出描述

类型： 包括以下类型：

布尔型

字符串：UTF-8八位组串

八位组字符串：若干八位组。其编码见附件A和B.3

整数：4字节有符号整数

双精度：8字节有符号整数

字符：单个字母的Unicode UTF-8字符编码。可多于一个字节。

枚举：唯一允许值列表中的一个值。

子列表：包含列表中若干参数值的列表，且必须说明子列表的类型。除子列表类型之外，子列表的类型必须是本节定义的数据类型的一种。例如，类型为枚举的子列表。子列表的编码见附件A和B.2。

可选项： 是/否

该部分指示该参数是否可以在信号或事件参数中被省略。

可能的取值：

包必须明确指定属性的一组取值或如何确定取值的方式。当某个描述符中省略了属性参数值时，包必须指定一个默认值或默认行为。例如，当描述符中省略属性参数值时，与该属性相关的程序应暂停处理。

默认值：

一个默认值（不包括默认程序）可以指定为“可临时选择的”。

12.3 标识符

当采用文本编码时，标识符最大长度必须不允许超过64个字符，且不允许包含空格，除了标识符的第一位必须是字母之外，标识符可以包含字母和/或数字，以及下划线字符“_”。

当采用二进制编码时，标识符长度为2个字节。

必须为包含在枚举类型中的标识符在内的每个标识符分别指定文本和二进制的命名。

12.4 包注册

为了实现互操作，包必须向IANA注册。具体内容参见第14节。

13 协议子集

协议子集是对本建议书的进一步规定，用于规定H.248.1协议的用法和MG支持的功能。协议子集本身用于规定哪一些与H.248.1协议相关联的可选项已经被使用。例如，传输方式和支持某个应用的包。

协议子集使用名称和版本进行标识，且该名称已通过IANA进行注册。协议子集的名称不分大小写字母且最大长度必须为64字节。版本的取值范围必须为1至99。

协议子集本身是一个文档，用于规定适用于某个特殊应用的可选项。文档无固定的格式。唯一强制要求的是协议子集文档内应该包含一个用于描述协议子集的名称和版本以及摘要的部分。

以下两个选项是协议子集的必选项：

- 协议子集标识（Profile Identification）：定义Servicechange命令中包含的协议子集名称和版本。
- 摘要：用于描述协议子集。

本建议书附录三包含定义“协议子集”的模版。该模版可用作定义“协议子集”的基础。

14 IANA注意事项

14.1 包

当需要向IANA注册包时，必须考虑以下注意事项：

- 1) 应为每一个包注册唯一的字符名、序列号和版本号。字符名使用文本编码。序列号必须使用二进制编码。对于序列号编码来说，0x8000至0xFFFF之间为预留给值。“0”也是预留给值。
- 2) 必须指定联系人的名称、Email和邮编。必要时，必须通过定义组织机构以获得联系人最新信息。
- 3) 用于描述包的参考文档应该是公开的。该文档必须指定文档中所描述的包的版本号。如果文档是公开的，该文档应该放置在一个公开的Web地址上，且具备一个固定的URL。该地址应该提供一种提交意见和对意见进行反馈的机制。
- 4) 由其他组织注册的包，其包名长度必须不少于8个字符。
- 5) 当所有其他条件满足时，所有包名采用先到先得方式进行分配。

14.2 差错代码

当需要向IANA注册差错代码时，必须考虑以下注意事项：

- 1) 应为每一个差错代码注册一个差错代码和一行（不超过80个字符）的字符串。
- 2) 对检测到差错发生的条件的完整描述应包含在一个公开文档中。且必须清楚地表述该差错代码与其他所有现有差错代码之间的差别。

- 3) 该文档应该放置在一个公开的Web地址上，且具备一个固定的URL。
- 4) 由其他组织注册的差错代码，其差错代码长度必须具备3或4个字符。
- 5) 有所有其他组织或个人注册的差错代码长度必须是4个字符。
- 6) 除注册该差错代码的组织或个人外，必须不允许其他任何人对差错代码进行重新定义或修改。

14.3 业务改变原因值

当需要向IANA注册业务改变原因值时，必须考虑以下注意事项：

- 1) 应为每一个业务改变原因值注册一个最大长度为80个字符且具备唯一的原因值的短语。
- 2) 必须在一个公开文档中对检测原因发生的条件进行完整描述。且必须清楚地表述该原因与其他所有的现有原因之间的差别。
- 3) 该文档应该放置在一个公开的Web地址上，且具备一个固定的URL。

14.4 协议子集

当需要向IANA注册协议子集时，必须考虑以下注意事项：

- 1) 应为每一个协议子集注册一个唯一的字符串名和版本号（当协议子集名称包含通配符时，可以省略版本号）。
- 2) 必须指定联系人的名称、Email和邮编。必要时，可以通过定义组织机构来获得联系人的最新信息。
- 3) 由其他组织注册的协议子集，其协议子集名长度必须不少于6个字符。
- 4) 如果协议子集名的前6个字符是完全指定的，则协议子集名尾部字符使用通配符“*”是运行的。注册协议子集名的组织对与通配符相关的命令空间进行管理。IANA不允许颁布或接受与“name*”所匹配的任何其他协议子集。

如果其他所有条件均满足时，所有协议子集名采用先到先得方式进行分配。

附 件 A

协议的二进制编码

本附件规定了使用ASN.1所定义的抽象句法记法进行编码的方式，其中ASN.1的内容参见ITU-T X.680建议书《信息技术 — 抽象句法记法1 (ASN.1)：基本记法规范》。当采用ASN.1描述协议句法时，消息的编码规则必须遵循ITU-T X.690建议书《信息技术 — ASN.1编码规则：基本编码规则 (BER)、规范编码规则 (CER) 和特异编码规则 (DER) 的规范》的基本编码规则。

A.1 通配符编码

本协议允许使用通配符“ALL”和“CHOOSE”。通过这些通配符，媒体网关控制器可以部分指定终结点标识符 (TerminationID)，指示媒体网关从满足匹配条件的终结点中选择一个终结点。终结点标识符可采用分级方式进行编码。分级命名方式可以预先设定。例如，终结点标识符可以由中继群号、中继号和电路号组成。每一级命名都允许使用通配符。以下各段将详细描述通配符的应用。

当使用ASN.1进行编码时，终结点标识符的最大长度为8个字节。即最多不超过64个比特。一个完全指定的终结点标识符可使用一个通配字段来作为其前缀。通配字段的长度为1个字节。其中，通配字段的第7比特（最高位）用于标识使用的通配类型：当该比特值等于“1”时表示通配值“ALL”，该比特值等于“0”时表示通配值“CHOOSE”。通配字段的第6比特用于标识通配适用于终结点标识符分级命名中的某一级：当该比特等于“0”时，通配只作用于通配字段所标识的终结点标识符中命名的那一级，当该比特等于“1”时，通配将作用于通配字段所标识的终结点标识符中命名的级以及下一级。通配字段的第5比特至第0比特定义了终结点标识符中通配开始的起始位置。

通配字段使用方法如下所示。其中，表示字符串的最高位将位于左侧。

假定终结点标识符为3个字节长度，每一个字节标识一级命名。该终结点标识符如下所示：

00000001 00011110 01010101。

假定需要指定前缀为00000001 00011110的所有终结点，则通配字段应描述如下：

通配字段：10000111

终结点标识符：00000001 00011110 xxxxxxxx。

终结点标识符中标为“x”的比特是无关的，接收方必须忽略该部分内容。

假定接收方必须选择通配符为“CHOOSE”的终结点满足第二个字节为0001110，则通配描述如下：

通配字段：00010111 之后为00000111

终结点标识符：xxxxxxx 00011110 xxxxxxxx。

第一个通配字段表示该“CHOOSE”通配是从最高位（第23比特）开始的单级通配。第二个通配字段表示该“CHOOSE”通配是从最低位（第7比特）开始的单级通配。

最后，假定通配符为“CHOOSE”的第一字节为00000001，则通配描述如下：

通配字段：01001111

终结点标识符：0000001 xxxxxxxx xxxxxxxx。

第6比特为“1”之表明该通配将作用于通配字段所标识的那一级以及以下各级。

关联标识符也可以使用通配符。然而，关联标识符不允许使用局部通配。关联标识符“0x0”表示空关联；关联标识符为“0xFFFFFFFFE”表示通配值为“CHOOSE”的关联；关联标识符为“0xFFFFFFFF”于表示通配值为“ALL”的关联。

终结点标识符等于“0xFFFFFFFFFFFFFFFF”，用于表示根终结点。

A.2 ASN.1句法规范

A.2 ASN.1 句法定义

本节定义了H.248.1建议书协议句法的ASN.1规范。

注1 — 当采用支持ALF的传送机制时，以下事务的定义将有所不同。具体内容参见本建议书附件或H.248.x 分支系列建议书中定义传送机制的其他建议书。

注2 — 以下句法规范并不强制支持所有对信息单元和参数值的限制条件。某些附加的限制条件可以包含在后面的注释中，或者其他限制条件可出现在本建议书中。即使本建议书未强制规定必须实现附加的限制条件，然而附加的限制条件仍然是本协议的一个组成部分。

注3 — ASN.1句法使用8比特长字节对特性参数、信号参数、事件参数和统计进行编码。参数值的实际数值类型可以不同，不同特性值的编码可参见本建议书附件C和其他相关的包定义。

作为首选，参数值应根据参数类型和下表进行BER编码。BER编码的结果将被编码为一个ASN.1字符串，且参数值被双重封装。参照下表，对特性参数进行BER编码参见本建议书附件C或其他包定义：

包定义的类型	ASN.1 BER 类型
字串	IA5String或UTF8String (注 4)
整数 (4字节)	INTEGER
双精度 (8字节有符号整数)	INTEGER (注 3)
字符 (UTF-8) (注1)	IA5String
枚举	ENUMERATED
布尔型	BOOLEAN
无符号整数 (注2)	INTEGER (注3)
八位组(字串)	OCTET STRING
注1 — 长度可大于1字节。 注2 — 无符号整数参见本建议书附件C。 注3 — INTEGER的BER编码不意味着使用4字节。 注4 — 当内容全部为ASCII字符时，字串应该被编码成IA5String，当内容不全部为ASCII字符时，应编码成UTF-8字串。	

1字节字串值的编码见8.7/X.690。

```

MEDIA-GATEWAY-CONTROL {itu-t(0) recommendation(0) h(8) h248(248) modules(0)
media-gateway-control(0) version3(3)}
DEFINITIONS AUTOMATIC TAGS ::=
BEGIN
MegacoMessage      ::= SEQUENCE
{
    authHeader      AuthenticationHeader OPTIONAL,
    mess            Message

```

```

}

AuthenticationHeader ::= SEQUENCE
{
    secParmIndex      SecurityParmIndex,
    seqNum            SequenceNum,
    ad                AuthData
}

SecurityParmIndex ::= OCTET STRING(SIZE(4))
SequenceNum       ::= OCTET STRING(SIZE(4))
AuthData          ::= OCTET STRING(SIZE(12..32))
Message           ::= SEQUENCE
{
    version           INTEGER(0..99),
    -- 此处规定协议版本等于3。
    mId              Mid, -- 消息发起者的名称/地址
    messageBody      CHOICE
    {
        messageError      ErrorDescriptor,
        transactions      SEQUENCE OF Transaction
    },
    ...
}

Mid ::= CHOICE
{
    ip4Address        IP4Address,
    ip6Address        IP6Address,
    domainName        DomainName,
    deviceName        PathName,
    mtpAddress        OCTET STRING(SIZE(2..4)),
    -- mtpAddress 地址结构如下:
    --      25 - 15      0
    --      | PC        | NI |
    --      24 - 14 比特  2 比特
    -- 注 — 14 比特为国际规范使用。
    -- 有两个国家的点码是 16 或 24 比特。
    -- 与 mtpAddress 并列的 MSB 应编码必须为 0。
    ...
}

DomainName ::= SEQUENCE
{
    name              IA5String,
    -- 名称始以字母开始的字符序列
    -- 包含字母数字、连字符(-)、句号(.)。
    -- 然而, 不允许包含两个连续的句号。
    portNumber        INTEGER(0..65535) OPTIONAL
}

IP4Address ::= SEQUENCE
{
    address           OCTET STRING(SIZE(4)),
    portNumber        INTEGER(0..65535) OPTIONAL
}

IP6Address ::= SEQUENCE
{
    address           OCTET STRING(SIZE(16)),
    portNumber        INTEGER(0..65535) OPTIONAL
}

```



```

}

PathName          ::= IA5String(SIZE (1..64))
-- 见 A.3

Transaction ::= CHOICE
{
    transactionRequest      TransactionRequest,
    transactionPending      TransactionPending,
    transactionReply        TransactionReply,
    transactionResponseAck  TransactionResponseAck,
    -- 响应证实与低层传送协议有关
    ...,
    segmentReply            SegmentReply
}

TransactionId     ::= INTEGER(0..4294967295) -- 32 比特无符号整数

TransactionRequest ::= SEQUENCE
{
    transactionId          TransactionId,
    actions                SEQUENCE OF ActionRequest,
    ...
}

TransactionPending ::= SEQUENCE
{
    transactionId          TransactionId,
    ...
}

TransactionReply   ::= SEQUENCE
{
    transactionId          TransactionId,
    immAckRequired        NULL OPTIONAL,
    transactionResult      CHOICE
    {
        transactionError    ErrorDescriptor,
        actionReplies        SEQUENCE OF ActionReply
    },
    ...,
    segmentNumber          SegmentNumber OPTIONAL,
    segmentationComplete  NULL OPTIONAL
}

SegmentReply       ::= SEQUENCE
{
    transactionId          TransactionId,
    segmentNumber          SegmentNumber,
    segmentationComplete  NULL OPTIONAL,
    ...
}

SegmentNumber     ::= INTEGER(0..65535)

TransactionResponseAck ::= SEQUENCE OF TransactionAck
TransactionAck      ::= SEQUENCE
{
    firstAck              TransactionId,
    lastAck               TransactionId OPTIONAL
}

ErrorDescriptor    ::= SEQUENCE

```

```

{
    errorCode          ErrorCode,
    errorText         ErrorText OPTIONAL
}

ErrorCode            ::= INTEGER(0..65535)
-- 参见第 14 节中与差错代码相关地 IANA 注意事项

ErrorText            ::= IA5String

ContextID             ::= INTEGER(0..4294967295)
-- 关联 “NULL” 的标识符: 0
-- 关联 “CHOOSE” 的标识符: 4294967294 (0xFFFFFFFFE)
-- 关联 “ALL” 的标识符: 4294967295 (0xFFFFFFFF)

ActionRequest         ::= SEQUENCE
{
    contextId          ContextID,
    contextRequest     ContextRequest OPTIONAL,
    contextAttrAuditReq ContextAttrAuditRequest OPTIONAL,
    commandRequests   SEQUENCE OF CommandRequest
}

ActionReply           ::= SEQUENCE
{
    contextId          ContextID,
    errorDescriptor    ErrorDescriptor OPTIONAL,
    contextReply       ContextRequest OPTIONAL,
    commandReply       SEQUENCE OF CommandReply
}

ContextRequest        ::= SEQUENCE
{
    priority           INTEGER(0..15) OPTIONAL,
    emergency          BOOLEAN OPTIONAL,
    topologyReq       SEQUENCE OF TopologyRequest OPTIONAL,
    ...,
    iepscallind       BOOLEAN OPTIONAL,
    contextProp        SEQUENCE OF PropertyParm OPTIONAL,
    contextList       SEQUENCE OF ContextID OPTIONAL
}
-- 当返回 “contextList” 参数时, ActionReply 数据结构中的 contextId 将返回所关联的
-- ActionRequest 中的 contextId。

ContextAttrAuditRequest ::= SEQUENCE
{
    topology          NULL OPTIONAL,
    emergency         NULL OPTIONAL,
    priority          NULL OPTIONAL,
    ...,
    iepscallind       NULL OPTIONAL,
    contextPropAud    SEQUENCE OF IndAudPropertyParm OPTIONAL,
    selectpriority    INTEGER(0..15) OPTIONAL,
    -- 用于选择指定的优先
    selectemergency   BOOLEAN OPTIONAL,
    -- 用于选择 “emergency” 参数是否设置 (T/F)
    selectiepscallind BOOLEAN OPTIONAL,
    -- 用于选择 “IEPS” 参数是否设置 (T/F)
    selectLogic       SelectLogic OPTIONAL -- 默认值为 “AND”
}

```

```

SelectLogic ::= CHOICE
{
    andAUDITSelect    NULL,    -- 所有过滤条件满足
    orAUDITSelect     NULL,    -- 至少一个过滤条件满足
    ...
}

CommandRequest ::= SEQUENCE
{
    command           Command,
    optional           NULL OPTIONAL,
    wildcardReturn    NULL OPTIONAL,
    ...
}

Command ::= CHOICE
{
    addReq            AmmRequest,
    moveReq           AmmRequest,
    modReq            AmmRequest,
    -- Add, Move, Modify 参数使用相同的输入参数
    subtractReq       SubtractRequest,
    auditCapRequest   AuditRequest,
    auditValueRequest AuditRequest,
    notifyReq         NotifyRequest,
    serviceChangeReq ServiceChangeRequest,
    ...
}

CommandReply ::= CHOICE
{
    addReply          AmmsReply,
    moveReply         AmmsReply,
    modReply          AmmsReply,
    subtractReply     AmmsReply,
    -- Add, Move, Modify, Subtract 参数使用相同的输入参数
    auditCapReply     AuditReply,
    auditValueReply   AuditReply,
    notifyReply       NotifyReply,
    serviceChangeReply ServiceChangeReply,
    ...
}

TopologyRequest ::= SEQUENCE
{
    terminationFrom   TerminationID,
    terminationTo     TerminationID,
    topologyDirection ENUMERATED
    {
        bothway(0),
        isolate(1),
        oneway(2)
    },
    ...,
    streamID          StreamID OPTIONAL,
    topologyDirectionExtension ENUMERATED
    {
        onewayexternal(0),
        onewayboth(1),
        ...
    }
}

AmmRequest ::= SEQUENCE

```

```

{
    terminationID          TerminationIDList,
    descriptors            SEQUENCE OF AmmDescriptor,
    -- 每一个类型最多一个描述符(参见 AmmDescriptor)
    -- 允许以序列方式出现。
    ...
}

AmmDescriptor ::= CHOICE
{
    mediaDescriptor       MediaDescriptor,
    modemDescriptor       ModemDescriptor,
    muxDescriptor         MuxDescriptor,
    eventsDescriptor      EventsDescriptor,
    eventBufferDescriptor EventBufferDescriptor,
    signalsDescriptor     SignalsDescriptor,
    digitMapDescriptor    DigitMapDescriptor,
    auditDescriptor       AuditDescriptor,
    ...,
    statisticsDescriptor  StatisticsDescriptor
}

AmmsReply ::= SEQUENCE
{
    terminationID          TerminationIDList,
    terminationAudit       TerminationAudit OPTIONAL,
    ...
}

SubtractRequest ::= SEQUENCE
{
    terminationID          TerminationIDList,
    auditDescriptor       AuditDescriptor OPTIONAL,
    ...
}

AuditRequest ::= SEQUENCE
{
    terminationID          TerminationID,
    auditDescriptor       AuditDescriptor,
    ...,
    terminationIDList     TerminationIDList OPTIONAL
}
-- 当 AuditRequest 包含 terminationIDList 数据结构时, terminationID 应包含
-- 该列表中的第一个终结点

AuditReply ::= CHOICE
{
    contextAuditResult     TerminationIDList,
    error                  ErrorDescriptor,
    auditResult            AuditResult,
    ...,
    auditResultTermList    TermListAuditResult
}

AuditResult ::= SEQUENCE
{
    terminationID          TerminationID,
    terminationAuditResult TerminationAudit
}

TermListAuditResult ::= SEQUENCE
{
    terminationIDList      TerminationIDList,

```

```

    terminationAuditResult      TerminationAudit,
    ...
}

TerminationAudit      ::= SEQUENCE OF AuditReturnParameter

AuditReturnParameter  ::= CHOICE
{
    errorDescriptor      ErrorDescriptor,
    mediaDescriptor      MediaDescriptor,
    modemDescriptor      ModemDescriptor,
    muxDescriptor         MuxDescriptor,
    eventsDescriptor     EventsDescriptor,
    eventBufferDescriptor EventBufferDescriptor,
    signalsDescriptor     SignalsDescriptor,
    digitMapDescriptor   DigitMapDescriptor,
    observedEventsDescriptor ObservedEventsDescriptor,
    statisticsDescriptor StatisticsDescriptor,
    packagesDescriptor   PackagesDescriptor,
    emptyDescriptors     AuditDescriptor,
    ...
}

AuditDescriptor       ::= SEQUENCE
{
    auditToken           BIT STRING
    {
        muxToken(0),
        modemToken(1),
        mediaToken(2),
        eventsToken(3),
        signalsToken(4),
        digitMapToken(5),
        statsToken(6),
        observedEventsToken(7),
        packagesToken(8),
        eventBufferToken(9)
    } OPTIONAL,
    ...,
    auditPropertyToken  SEQUENCE OF IndAuditParameter OPTIONAL
}

IndAuditParameter    ::= CHOICE
{
    indaudmediaDescriptor IndAudMediaDescriptor,
    indaudeventsDescriptor IndAudEventsDescriptor,
    indaudeventBufferDescriptor IndAudEventBufferDescriptor,
    indaudsignalsDescriptor IndAudSignalsDescriptor,
    indauidigitMapDescriptor IndAudDigitMapDescriptor,
    indaudstatisticsDescriptor IndAudStatisticsDescriptor,
    indaudpackagesDescriptor IndAudPackagesDescriptor,
    ...
}

IndAudMediaDescriptor ::= SEQUENCE
{
    termStateDescr      IndAudTerminationStateDescriptor OPTIONAL,
    streams              CHOICE
    {
        oneStream        IndAudStreamParms,
        multiStream      SEQUENCE OF IndAudStreamDescriptor
    } OPTIONAL,
    ...
}

```

```

IndAudStreamDescriptor ::= SEQUENCE
{
    streamID                StreamID,
    streamParms             IndAudStreamParms
}

IndAudStreamParms ::= SEQUENCE
{
    localControlDescriptor  IndAudLocalControlDescriptor OPTIONAL,
    localDescriptor         IndAudLocalRemoteDescriptor OPTIONAL,
    remoteDescriptor        IndAudLocalRemoteDescriptor OPTIONAL,
    ...,
    statisticsDescriptor    IndAudStatisticsDescriptor OPTIONAL
}

IndAudLocalControlDescriptor ::= SEQUENCE
{
    streamMode              NULL OPTIONAL,
    reserveValue            NULL OPTIONAL,
    reserveGroup            NULL OPTIONAL,
    propertyParms           SEQUENCE OF IndAudPropertyParm OPTIONAL,
    ...,
    streamModeSel           StreamMode OPTIONAL
}
-- 不允许同时包含 streamMode 和 streamModeSel
-- 如果同时包含 streamMode 和 streamModeSel, 则只有 streamModeSel 被处理

IndAudPropertyParm ::= SEQUENCE
{
    name                    PkgdName,
    ...,
    propertyParms           PropertyParm OPTIONAL
}
-- 基于属性参数值进行选择
-- AND 或 OR 选择逻辑由关联进行指定

IndAudLocalRemoteDescriptor ::= SEQUENCE
{
    propGroupID             INTEGER(0..65535) OPTIONAL,
    propGrps                IndAudPropertyGroup,
    ...
}

IndAudPropertyGroup ::= SEQUENCE OF IndAudPropertyParm

IndAudTerminationStateDescriptor ::= SEQUENCE
{
    propertyParms           SEQUENCE OF IndAudPropertyParm,
    eventBufferControl      NULL OPTIONAL,
    serviceState            NULL OPTIONAL,
    ...,
    serviceStateSel        ServiceState OPTIONAL
}
-- 不允许同时包含 serviceState 和 serviceStateSel
-- 同时包含 serviceState 和 serviceStateSel, 则只有 serviceStateSel 被处理

IndAudEventsDescriptor ::= SEQUENCE
{
    requestID               RequestID OPTIONAL,
    pkgdName                PkgdName,
}

```

```

        streamID          StreamID OPTIONAL,
        ...
    }
IndAudEventBufferDescriptor ::= SEQUENCE
{
    eventName            PkgdName,
    streamID            StreamID OPTIONAL,
    ...
}
IndAudSignalsDescriptor ::= CHOICE
{
    signal              IndAudSignal,
    seqSigList          IndAudSeqSigList,
    ...
}
IndAudSeqSigList      ::= SEQUENCE
{
    id                  INTEGER(0..65535),
    signalList          IndAudSignal OPTIONAL
}
IndAudSignal          ::= SEQUENCE
{
    signalName          PkgdName,
    streamID            StreamID OPTIONAL,
    ...,
    signalRequestID     RequestID OPTIONAL
}
IndAudDigitMapDescriptor ::= SEQUENCE
{
    digitMapName        DigitMapName OPTIONAL
}
IndAudStatisticsDescriptor ::= SEQUENCE
{
    statName            PkgdName
}
IndAudPackagesDescriptor ::= SEQUENCE
{
    packageName         Name,
    packageVersion      INTEGER(0..99),
    ...
}
NotifyRequest         ::= SEQUENCE
{
    terminationID       TerminationIDList,
    observedEventsDescriptor ObservedEventsDescriptor,
    errorDescriptor     ErrorDescriptor OPTIONAL,
    ...
}
NotifyReply           ::= SEQUENCE
{
    terminationID       TerminationIDList,
    errorDescriptor     ErrorDescriptor OPTIONAL,
    ...
}
ObservedEventsDescriptor ::= SEQUENCE

```

```

{
    requestId          RequestID,
    observedEventList SEQUENCE OF ObservedEvent
}
ObservedEvent ::= SEQUENCE
{
    eventName          EventName,
    streamID           StreamID OPTIONAL,
    eventParList       SEQUENCE OF EventParameter,
    timeNotation       TimeNotation OPTIONAL,
    ...
}

EventName ::= PkgdName

EventParameter ::= SEQUENCE
{
    eventParameterName Name,
    value               Value,
    -- 关于 extraInfos 的使用, 见与 PropertyParm 有关的说明
    extraInfo          CHOICE
    {
        relation        Relation,
        range            BOOLEAN,
        sublist          BOOLEAN
    } OPTIONAL,
    ...
}

ServiceChangeRequest ::= SEQUENCE
{
    terminationID      TerminationIDList,
    serviceChangeParms ServiceChangeParm,
    ...
}

ServiceChangeReply ::= SEQUENCE
{
    terminationID      TerminationIDList,
    serviceChangeResult ServiceChangeResult,
    ...
}

-- ServiceChangeResult 参数无必选参数项。
-- 因此必须区分 ServiceChangeParm 和 ServiceChangeResParm
ServiceChangeResult ::= CHOICE
{
    errorDescriptor    ErrorDescriptor,
    serviceChangeResParms ServiceChangeResParm
}

WildcardField ::= OCTET STRING(SIZE(1))

TerminationID ::= SEQUENCE
{
    wildcard           SEQUENCE OF WildcardField,
    id                 OCTET STRING(SIZE(1..8)),
    ...
}

-- 参见A.1对于通配机制的描述。
-- TerminationID为“0xFFFFFFFFFFFFFFFF”表示ROOT终结点。

```



```

TerminationIDList      ::= SEQUENCE OF TerminationID

MediaDescriptor        ::= SEQUENCE
{
    termStateDescr      TerminationStateDescriptor OPTIONAL,
    streams              CHOICE
    {
        oneStream        StreamParms,
        multiStream      SEQUENCE OF StreamDescriptor
    } OPTIONAL,
    ...
}

StreamDescriptor       ::= SEQUENCE
{
    streamID             StreamID,
    streamParms          StreamParms
}

StreamParms            ::= SEQUENCE
{
    localControlDescriptor LocalControlDescriptor OPTIONAL,
    localDescriptor       LocalRemoteDescriptor OPTIONAL,
    remoteDescriptor      LocalRemoteDescriptor OPTIONAL,
    ...,
    statisticsDescriptor  StatisticsDescriptor OPTIONAL
}

LocalControlDescriptor ::= SEQUENCE
{
    streamMode           StreamMode OPTIONAL,
    reserveValue         BOOLEAN OPTIONAL,
    reserveGroup         BOOLEAN OPTIONAL,
    propertyParms        SEQUENCE OF PropertyParm,
    ...
}

StreamMode             ::= ENUMERATED
{
    sendOnly(0),
    recvOnly(1),
    sendRecv(2),
    inactive(3),
    loopBack(4),
    ...
}

```

-- *PropertyParm*参数值是8字节字符串序列。当该参数由MGC发送时，应采用以下方式进行解析：

- 空序列表示CHOOSE
- 一个序列指定参数值。
- 如果子序列字段未选择，则更长的序列表示应选择一个参数值（例如，参数值1或参数值2，或）
- 如果子序列字段被选择，则单元数目大于1的序列应对一个使用列表值的特性进行编码（例如参数值1和参数值2，和）
- 如果参数值序列的长度为1，则只有与该字段最近的字段可以被选择。MG必须选择为特性选择一个参数值。例如， $x > 3$ 指示MG必须为属性X选择大于3的参数值。
- 如果参数序列的长度为2，则只有范围字段可以被选择。指示MG必须选择一个包含在规定包含边界范围内

-- 的参数值。
 -- 当该参数由MG发送时，只有AuditCapability请求响应消息中可以包含多个参数值，范围或
 -- 一个关联字段。

```
PropertyParm ::= SEQUENCE
{
    name                PkgdName,
    value               SEQUENCE OF OCTET STRING,
    extraInfo           CHOICE
    {
        relation        Relation,
        range            BOOLEAN,
        sublist         BOOLEAN
    } OPTIONAL,
    ...
}
```

```
Name ::= OCTET STRING(SIZE(2))
```

```
PkgdName ::= OCTET STRING(SIZE(4))
```

-- 用于标识包名（2字节）、属性、事件、信号名和统计值标识符（2个字节）。
 -- 如果对包使用通配符，则前两个字符必须为0xFFFF，不允许使用通配符CHOOSE。
 -- 参考本建议书附件C定义的特性标签，前两个字符应为0x0000。
 -- 为了对特性、事件、信号和统计标识符使用通配符，后两个字节应为0xFFFF，不允许使用通配符
 -- CHOOSE。
 -- 只有当特性、事件、信号和统计标识符被通配时，包名才允许使用通配符。

```
Relation ::= ENUMERATED
{
    greaterThan(0),
    smallerThan(1),
    unequalTo(2),
    ...}

```

```
LocalRemoteDescriptor ::= SEQUENCE
{
    propGrps           SEQUENCE OF PropertyGroup,
    ...
}
```

```
PropertyGroup ::= SEQUENCE OF PropertyParm
```

```
TerminationStateDescriptor ::= SEQUENCE
{
    propertyParms      SEQUENCE OF PropertyParm,
    eventBufferControl EventBufferControl OPTIONAL,
    serviceState       ServiceState OPTIONAL,
    ...
}
```

```
EventBufferControl ::= ENUMERATED
{
    off(0),
    lockStep(1),
    ...
}
```

```
ServiceState ::= ENUMERATED
{
    test(0),
    outOfSvc(1),
    ...
}
```

```

        inSvc (2),
        ...
    }
MuxDescriptor ::= SEQUENCE
{
    muxType          MuxType,
    termList         SEQUENCE OF TerminationID,
    nonStandardData NonStandardData OPTIONAL,
    ...
}
MuxType ::= ENUMERATED
{
    h221 (0),
    h223 (1),
    h226 (2),
    v76 (3),
    ...,
    nx64k (4)
}

StreamID ::= INTEGER(0..65535) -- 16 比特无符号整数

EventsDescriptor ::= SEQUENCE
{
    requestID      RequestID OPTIONAL,
    -- 如果 eventList 为非空,
    -- 则包含必须 RequestID
    eventList      SEQUENCE OF RequestedEvent,
    ...
}

RequestedEvent ::= SEQUENCE
{
    pkgdName       PkgdName,
    streamID       StreamID OPTIONAL,
    eventAction    RequestedActions OPTIONAL,
    evParList      SEQUENCE OF EventParameter,
    ...
}

RegulatedEmbeddedDescriptor ::= SEQUENCE
{
    secondEvent    SecondEventsDescriptor OPTIONAL,
    signalsDescriptor SignalsDescriptor OPTIONAL,
    ...
}

NotifyBehaviour ::= CHOICE
{
    notifyImmediate NULL,
    notifyRegulated RegulatedEmbeddedDescriptor,
    neverNotify      NULL,
    ...
}

RequestedActions ::= SEQUENCE
{
    keepActive     BOOLEAN OPTIONAL,
    eventDM        EventDM OPTIONAL,
    secondEvent    SecondEventsDescriptor OPTIONAL,
    signalsDescriptor SignalsDescriptor OPTIONAL,
    ...
}

```

```

    notifyBehaviour      NotifyBehaviour OPTIONAL,
    resetEventsDescriptor NULL OPTIONAL
}

EventDM                ::= CHOICE
{
    digitMapName        DigitMapName,
    digitMapValue       DigitMapValue
}

SecondEventsDescriptor ::= SEQUENCE
{
    requestID           RequestID OPTIONAL,
    eventList           SEQUENCE OF SecondRequestedEvent,
    ...
}

SecondRequestedEvent   ::= SEQUENCE
{
    pkgdName            PkgdName,
    streamID            StreamID OPTIONAL,
    eventAction         SecondRequestedActions OPTIONAL,
    evParList           SEQUENCE OF EventParameter,
    ...
}

SecondRequestedActions ::= SEQUENCE
{
    keepActive          BOOLEAN OPTIONAL,
    eventDM             EventDM OPTIONAL,
    signalsDescriptor   SignalsDescriptor OPTIONAL,
    ...,
    notifyBehaviour     NotifyBehaviour OPTIONAL,
    resetEventsDescriptor NULL OPTIONAL
}

EventBufferDescriptor  ::= SEQUENCE OF EventSpec

EventSpec              ::= SEQUENCE
{
    eventName           EventName,
    streamID            StreamID OPTIONAL,
    eventParList        SEQUENCE OF EventParameter,
    ...
}

SignalsDescriptor      ::= SEQUENCE OF SignalRequest

SignalRequest          ::= CHOICE
{
    signal              Signal,
    seqSigList          SeqSigList,
    ...
}

SeqSigList             ::= SEQUENCE
{
    id                  INTEGER(0..65535),
    signalList          SEQUENCE OF Signal
}

Signal                 ::= SEQUENCE
{
    signalName          SignalName,

```

```

    streamID          StreamID OPTIONAL,
    sigType           SignalType OPTIONAL,
    duration          INTEGER (0..65535) OPTIONAL,
    notifyCompletion  NotifyCompletion OPTIONAL,
    keepActive        BOOLEAN OPTIONAL,
    sigParList        SEQUENCE OF SigParameter,
    ...,
    direction         SignalDirection OPTIONAL,
    requestID         RequestID OPTIONAL,
    intersigDelay     INTEGER (0..65535) OPTIONAL
}

SignalType ::= ENUMERATED
{
    brief(0),
    onOff(1),
    timeOut(2),
    ...
}

SignalDirection ::= ENUMERATED
{
    internal(0),
    external(1),
    both(2),
    ...
}

SignalName ::= PkgdName

NotifyCompletion ::= BIT STRING
{
    onTimeOut(0), onInterruptByEvent(1),
    onInterruptByNewSignalDescr(2), otherReason(3), onIteration(4)
}

SigParameter ::= SEQUENCE
{
    sigParameterName Name,
    value             Value,
    -- 关于extraInfo的使用, 见与PropertyParm有关的说明
    extraInfo         CHOICE
    {
        relation      Relation,
        range          BOOLEAN,
        sublist        BOOLEAN
    } OPTIONAL,
    ...
}

-- 如果AuditCapReply包含所有事件, 则RequestID必须为通配符ALL
-- ALL通配符用0xffffffff标识
RequestID ::= INTEGER(0..4294967295) -- 32位无符号整数

ModemDescriptor ::= SEQUENCE
{
    mtl              SEQUENCE OF ModemType,
    mpl              SEQUENCE OF PropertyParm,
    nonStandardData NonStandardData OPTIONAL
}

ModemType ::= ENUMERATED
{

```

```

    v18(0),
    v22(1),
    v22bis(2),
    v32(3),
    v32bis(4),
    v34(5),
    v90(6),
    v91(7),
    synchISDN(8),
    ...
}

DigitMapDescriptor ::= SEQUENCE
{
    digitMapName          DigitMapName OPTIONAL,
    digitMapValue         DigitMapValue OPTIONAL
}

DigitMapName          ::= Name

DigitMapValue         ::= SEQUENCE
{
    startTimer           INTEGER(0..99) OPTIONAL,
    shortTimer           INTEGER(0..99) OPTIONAL,
    longTimer            INTEGER(0..99) OPTIONAL,
    digitMapBody         IA5String,
    -- 起始定时器、长定时器和短定时器的单位为秒，定时器时长单位为毫秒。
    -- 因而，起始定时器、长定时器和短定时器的范围为1至99秒，
    -- 时长为100 ms至9.9 s。
    -- digitmap 句法定义参见本 A.3。
    ...,
    durationTimer        INTEGER (0..99) OPTIONAL
}

ServiceChangeParm    ::= SEQUENCE
{
    serviceChangeMethod  ServiceChangeMethod,
    serviceChangeAddress ServiceChangeAddress OPTIONAL,
    serviceChangeVersion INTEGER(0..99) OPTIONAL,
    serviceChangeProfile ServiceChangeProfile OPTIONAL,
    serviceChangeReason  Value,
    -- serviceChangeReason 参数包含一个数字原因值和可选的文本描述。
    -- serviceChangeReason 参数必须是一个由十进制原因码组成的字符，
    -- 可选地后面可包含一个空格字符和文本描述字符。
    -- 字符采用 BER 规则编码成 IA5String 字符
    -- BER 编码地结果随后被编码成
    -- ASN.1 字节字串类型，且进行“双重封装”。
    serviceChangeDelay   INTEGER(0..4294967295) OPTIONAL,
    -- 32 位无符号整数
    serviceChangeMgcId   Mid OPTIONAL,
    timeStamp            TimeNotation OPTIONAL,
    nonStandardData      NonStandardData OPTIONAL,
    ...,
    serviceChangeInfo    AuditDescriptor OPTIONAL,
    serviceChangeIncompleteFlag NULL OPTIONAL
}

ServiceChangeAddress ::= CHOICE
{
    portNumber           INTEGER(0..65535),    -- TCP/UDP 端口号

```

```

    ip4Address          IP4Address,
    ip6Address          IP6Address,
    domainName         DomainName,
    deviceName         PathName,
    mtpAddress         OCTET STRING(SIZE(2..4)),
    ...
}

ServiceChangeResParm ::= SEQUENCE
{
    serviceChangeMgcId      MId OPTIONAL,
    serviceChangeAddress    ServiceChangeAddress OPTIONAL,
    serviceChangeVersion    INTEGER(0..99) OPTIONAL,
    serviceChangeProfile    ServiceChangeProfile OPTIONAL,
    timestamp              TimeNotation OPTIONAL,
    ...
}

ServiceChangeMethod ::= ENUMERATED
{
    failover(0),
    forced(1),
    graceful(2),
    restart(3),
    disconnected(4),
    handOff(5),
    ...
}

ServiceChangeProfile ::= SEQUENCE
{
    profileName           IA5String(SIZE (1..67))
    -- 名称为 64 字符, 1 表示 "/", 2 表示与 ABNF 匹配的版本
}

PackagesDescriptor ::= SEQUENCE OF PackagesItem

PackagesItem ::= SEQUENCE
{
    packageName          Name,
    packageVersion       INTEGER(0..99),
    ...
}

StatisticsDescriptor ::= SEQUENCE OF StatisticsParameter

StatisticsParameter ::= SEQUENCE
{
    statName             PkgdName,
    statValue           Value OPTIONAL
}
-- 如果统计值由一个子列表组成, 则 statValue
-- 参数可以多于一个字符串。

NonStandardData ::= SEQUENCE
{
    nonStandardIdentifier NonStandardIdentifier,
    data                 OCTET STRING
}

```

```

NonStandardIdentifier ::= CHOICE
{
    object                OBJECT IDENTIFIER,
    h221NonStandard      H221NonStandard,
    experimental         IA5String(SIZE(8)),
    -- 前两个字符应该为 "X-" 或 "X+"
    ...
}

H221NonStandard ::= SEQUENCE
{
    t35CountryCode1      INTEGER(0..255),
    t35CountryCode2      INTEGER(0..255),      -- T.35 定义的国家码
    t35Extension         INTEGER(0..255),      -- 国家分配
    manufacturerCode     INTEGER(0..65535),    -- 国家分配
    ...
}

TimeNotation ::= SEQUENCE
{
    date                 IA5String(SIZE(8)), -- yyyyymmdd 格式
    time                 IA5String(SIZE(8))  -- hhmmsssss 格式
    -- 按照 ISO 8601:2004
}

Value ::= SEQUENCE OF OCTET STRING

END

```

A.3 DigitMap和路径名称

从句法角度来讲，数字映射是具备句法约束的一组字串。数字映射的合法句法参见RFC 2234。本节下面列举的部分数字映射仅是描述性的。它与本建议书附件B中定义的数字映射有一定的差异。

```

digitMap                = (digitString / LWSP "(" LWSP digitStringList LWSP
                          ")" LWSP)

digitStringList         = digitString *(LWSP "|" LWSP digitString)

digitString              = 1*(digitStringElement)

digitStringElement      = digitPosition [DOT]

digitPosition           = digitMapLetter / digitMapRange

digitMapRange           = ("x" / (LWSP "[" LWSP digitLetter LWSP "]" LWSP))

digitLetter             = *((DIGIT "-" DIGIT) / digitMapLetter)

digitMapLetter          = DIGIT                ;Basic event symbols
                        / %x41-4B / %x61-6B ; a-k, A-K
                        / "L" / "S" / "T"    ;Inter-event timers
                        ; (long, short, start)
                        / "Z"                ;Long duration modifier

DOT                     = %x2E                ; "."

SP                      = %x20                ; space

HTAB                    = %x09                ; horizontal tab

CR                      = %x0D                ; Carriage return

```



```

LF           = %x0A                ; linefeed
LWSP        = *(WSP / COMMENT / EOL)
EOL         = (CR [LF] / LF)
WSP         = SP / HTAB            ; white space
SafeChar    = DIGIT / ALPHA / "+" / "-" / "&" /
              "!" / "_" / "/" / "|" / "?" / "@" /
              "^" / "~" / "~" / "*" / "$" / "\" /
              "(" / ")" / "%" / "|" / "."
RestChar    = ";" / "[" / "]" / "{" / "}" / ":" / "," / "#" /
              "<" / ">" / "="
DIGIT       = %x30-39              ; 0-9
ALPHA       = %x41-5A / %x61-7A   ; A-Z / a-z

```

路径名称是具有句法格式要求的字符串。其ABNF定义方式参数本建议书附件B。

; 路径名称的总长度务必不允许超过64字符。

```

pathname    = ["*"] NAME *("/" / "*" / ALPHA / DIGIT / "_" / "$" )
              ["@" pathDomainName ]

```

; ABNF 允许使用两个或多个连续的“.”字符。

; 尽管此用法在路径名称中是毫无意义的。

```

pathDomainName = (ALPHA / DIGIT / "*" ) *63 (ALPHA / DIGIT / "-" / "*" / ".")
NAME           = ALPHA *63 (ALPHA / DIGIT / "_")

```

附 件 B

协议的文本编码

B.1 通配符编码

当协议使用文本方式进行编码时，当终结点是任意的且具有某种选择性的名字时，通配字符“*”将非常有效。当通配发生时，将尝试所有具有相同属性的所有终结点。例如，假定有三个终结点R13/3/1、R13/3/2和R13/3/2，则终结点R13/3/*将匹配所有三个终结点。所有终结点应遵循约定的命名规则。终结点标识符为“*”表示通配值“ALL”。终结点标识符为“\$”表示通配值“CHOOSE”，媒体网关控制器指示媒体网关创建一个临时终结点或选择一个空闲的物理终结点。

B.2 ABNF 规范

ABNF规范的具体要求参见RFC 2234。

注1— 以下句法规范并不强制信息单元和参数值必须遵循所有的限制条件。某些附加的限制条件包含在后面的注释中，或者其他限制条件包含在本建议书的正文部分。尽管本建议书未强制规定必须实现所有的附加限制条件，然而这些附加的限制条件仍然是本协议的一个组成部分。

注2 一句法定义与关联有关。例如，“Add”命令可以是一个Add符号或与关联相关的名称。

采用ABNF方式进行文本编码的任何内容都不分大小写，即对大小写字符不进行区分。这包括TerminationID、DigitmapID，等。然而，SDP单元将大小写字符进行区分，参见RFC 2327。

```
; 注 — 本节中所使用的 ABNF 句法采用 VALUE 结构或 VALUE 结构列表对包
; 以及其所定义的特性、信号等参数进行编码。参数值类型可以是不同的，
; 分别在各自相关的包中进行定义。
; 参数值类型的定义参见第 12.2 节。
;
; ABNF 规范中定义的 VALUE 结构允许使用 quotedString 格式或 SafeChars 集合。
; 包参数值采用 ABNF VALUE 结构进行编码的规则将在本节进行描述。
; 如果某种类型的参数值编码要求使用字符而非 SafeChars，
; 即使某个特定的参数值仅包含 SafeChars，
; 则该类型的所有参数值都务必使用 quotedString 格式。
;
; 字符串：字符串务必采用 quotedString 格式的 VALUE 结构进行编码。
; 且字符串可以包含 quotedString 格式允许的任何字符。
;
; 整数、双精度整数和无符号整数：
; 使用 0-9 数字的十进制数。十六进制数务必使用“0x”前缀进行标识，
; 且每一个十六进制数可以是 0-9、a-f 或 A-F 之间的任意数值。不允许使用八进制编码方式。
; 负整数务必以字符“-”作为其前缀，且必须是十进制的整数。
; 整数、双精度整数和无符号整数务必使用格式为 SafeChars 的 VALUE 结构进行编码。
;
; 字符：用单引号字符包围的单个 UTF-8 字符。
;
; 枚举：枚举类型务必使用 SafeChars 的 VALUE 结构，且可以包含 SafeChar 允许的任何字符。
;
; 布尔型：“ON”或“OFF”，该类型不分大小写。务必使用格式为 SafeChars 的 VALUE 结构。
;
; 其他类型：任何自定义的类型参数值务必满足
; ABNF 规范对 VALUE 结构定义的要求。
; 如果某种类型参数值的编码要求使用字符而非 SafeChars，
; 即使某个特定的参数值仅包含 SafeChars，
; 则该类型的所有参数值务必使用 quotedString 格式。
;
; 在参数值结构中，不允许使用双引号字符。
;
; SDP 不允许在每一行的开始使用空格字符，H.248.1 协议允许 Local 或 Remote 描述符中的
; SDP 单元的起始部分使用空格字符。
; 在跟随 Local 和 Remote 符号后的 LBPKT 与 SDP 起始部分之间，
; 应该允许使用行分隔。
```

```
megacoMessage          = LWSP [authenticationHeader SEP] message

authenticationHeader   = AuthToken EQUAL SecurityParmIndex COLON
                        SequenceNum COLON AuthData

SecurityParmIndex      = "0x" 8 (HEXDIG)
SequenceNum            = "0x" 8 (HEXDIG)
AuthData                = "0x" 24*64 (HEXDIG)

Message                = MegacopToken SLASH Version SEP mId SEP messageBody
; 此处规定协议版本等于 3。

messageBody            = (errorDescriptor / transactionList)
```

```

transactionList      = 1*(transactionRequest / transactionReply /
                        transactionPending / transactionResponseAck /
                        segmentReply)
; 应答响应与底层传输协议有关

transactionPending   = PendingToken EQUAL TransactionID LBRKT RBRKT

transactionResponseAck = ResponseAckToken LBRKT transactionAck
                        *(COMMA transactionAck) RBRKT
transactionAck       = TransactionID / (TransactionID "-" TransactionID)

transactionRequest   = TransToken EQUAL TransactionID LBRKT
                        actionRequest *(COMMA actionRequest) RBRKT

actionRequest        = CtxToken EQUAL ContextID LBRKT ((contextRequest
                        [COMMA commandRequestList]) /
                        commandRequestList) RBRKT

contextRequest       = ((contextProperties [COMMA contextAudit])
                        / contextAudit)

contextProperties     = contextProperty *(COMMA contextProperty)

; 最多一次
; EmergencyOff 仅仅在 H.248.1 第一版本和第二版本中被使用, 由 MG 传送至 MGC 方向。
; EmergencyToken 或 EmergencyOffToken, 但不能两者同时使用
contextProperty      = (topologyDescriptor / priority / EmergencyToken /
                        EmergencyOffToken / iepValue /
                        contextAttrDescriptor)

contextAudit         = ContextAuditToken LBRKT (contextAuditProperties
                        *(COMMA contextAuditProperties)) /
                        indAudcontextAttrDescriptor      RBRKT

; 除 contextAuditSelector 参数之外, 最多一次
contextAuditProperties = (TopologyToken / EmergencyToken /
                        PriorityToken / IEPSToken / pkgdName /
                        contextAuditSelector)

; 最多一次
contextAuditSelector = priority / emergencyValue / iepValue /
                        contextAttrDescriptor / auditSelectLogic

auditSelectLogic     = [ AndAUDITselectToken / OrAUDITselectToken ]
; 如果为空, 则假定对选择条件进行 "AND" 操作。

indAudcontextAttrDescriptor
                    = ContextAttrToken LBRKT contextAuditProperties
                        *(COMMA contextAuditProperties) RBRKT

; "O-" 指示为可选命令
; "W-" 只是为一个命令的通配响应
commandRequestList   = ["O-"] ["W-"] commandRequest *(COMMA ["O-"]
                        ["W-"] commandRequest)

commandRequest       = (ammRequest / subtractRequest / auditRequest /
                        notifyRequest / serviceChangeRequest)

transactionReply     = ReplyToken EQUAL TransactionID [SLASH segmentNumber
                        [SLASH SegmentationCompleteToken]] LBRKT
                        [ImmAckRequiredToken COMMA]

```

```

        (errorDescriptor / actionReplyList) RBRKT
segmentReply      = MessageSegmentToken EQUAL TransactionID SLASH
                  segmentNumber [SLASH SegmentationCompleteToken]
segmentNumber    = UINT16
actionReplyList  = actionReply *(COMMA actionReply)
actionReply      = CtxToken EQUAL ContextID [LBRKT (errorDescriptor /
                  commandReply /(commandReply COMMA
                  errorDescriptor)) RBRKT]
commandReply     = ((contextProperties [COMMA commandReplyList]) /
                  commandReplyList)
commandReplyList = commandReplies *(COMMA commandReplies)
commandReplies   = (serviceChangeReply / auditReply / ammsReply /
                  notifyReply)
; Add、Move 和 Modify 命令具有相同的请求参数
ammRequest       = (AddToken / MoveToken / ModifyToken) EQUAL
                  termIDList [LBRKT ammParameter *(COMMA
                  ammParameter) RBRKT]
;最多一次
ammParameter     = (mediaDescriptor / modemDescriptor / muxDescriptor /
                  eventsDescriptor / signalsDescriptor /
                  digitMapDescriptor / eventBufferDescriptor /
                  auditDescriptor / statisticsDescriptor)
ammsReply        = (AddToken / MoveToken / ModifyToken / SubtractToken)
                  EQUAL termIDList [LBRKT terminationAudit RBRKT]
subtractRequest  = SubtractToken EQUAL termIDList [LBRKT
                  auditDescriptor RBRKT]
auditRequest     = (AuditValueToken / AuditCapToken) EQUAL
                  termIDList LBRKT auditDescriptor RBRKT
auditReply       = (AuditValueToken / AuditCapToken)
                  (contextTerminationAudit / auditOther)
auditOther       = EQUAL termIDList [LBRKT terminationAudit RBRKT]
terminationAudit = auditReturnParameter *(COMMA auditReturnParameter)
contextTerminationAudit = EQUAL CtxToken (terminationIDList / LBRKT
                  errorDescriptor RBRKT)
auditReturnParameter = (mediaDescriptor / modemDescriptor / muxDescriptor /
                  eventsDescriptor / signalsDescriptor /
                  digitMapDescriptor / observedEventsDescriptor /
                  eventBufferDescriptor / statisticsDescriptor /
                  packagesDescriptor / errorDescriptor /
                  auditReturnItem)
auditReturnItem  = (MuxToken / ModemToken / MediaToken /
                  DigitMapToken / StatsToken / ObservedEventsToken /
                  PackagesToken)
auditDescriptor  = AuditToken LBRKT [auditItem *(COMMA auditItem)]

```

```

RBRKT

notifyRequest      = NotifyToken EQUAL termIDList LBRKT
                    (observedEventsDescriptor [COMMA errorDescriptor])
                    RBRKT

notifyReply        = NotifyToken EQUAL termIDList [LBRKT errorDescriptor
                    RBRKT]

serviceChangeRequest = ServiceChangeToken EQUAL termIDList LBRKT
                    serviceChangeDescriptor RBRKT

serviceChangeReply = ServiceChangeToken EQUAL termIDList [LBRKT
                    (errorDescriptor / serviceChangeReplyDescriptor)
                    RBRKT]

errorDescriptor    = ErrorToken EQUAL ErrorCode LBRKT [quotedString]
                    RBRKT

ErrorCode          = 1*4(DIGIT) ; could be extended

TransactionID      = UINT32

mId                = ((domainAddress / domainName) [":" portNumber]) /
                    mtpAddress / deviceName

; ABNF 句法规则允许使用两个连续的"." 字符, 尽管该字符对于域名是毫无意义的。
domainName         = "<" (ALPHA / DIGIT) *63(ALPHA / DIGIT / "-" / ".")
                    ">"

deviceName         = pathNAME

; 0x0, 0xFFFFFFFF 和 0xFFFFFFFF 为预留值。
; '-' 用于标识空关联, '*' 用于标识通配符 "ALL"。 '$' 用于标识 通配符 "CHOOSE"。
ContextID          = (UINT32 / "*" / "-" / "$")

domainAddress      = "[" (IPv4address / IPv6address) "]"

; IPv6 地址定义参见 RFC 2373。
IPv6address        = hexpart [":" IPv4address]

IPv4address        = V4hex DOT V4hex DOT V4hex DOT V4hex

V4hex              = 1*3(DIGIT) ; "0".."255"

; 该参数虽在 RFC 2373 中出现, 但没有引用
RFC 2373, is not referenced
; IPv6prefix        = hexpart SLASH 1*2DIGIT

hexpart            = hexseq ":@" [hexseq] / ":@" [hexseq] / hexseq

hexseq             = hex4 *(":" hex4)

hex4               = 1*4HEXDIG

portNumber         = UINT16

; mtpAddress 地址结构如下:
; 25 - 15          0
; | PC           | NI |
; 24 - 14 bits   2 bits
; 注 — 14 比特为国际用法。
; 当点码为 16 位或者 24 位时, 有两个 national 选项存在。为了使 mtpAddress 编码成整字符,

```

```

; MSB 应该用“0”进行填充。一个字节由两个 16 进制的字符表示。
mtpAddress          = MTPToken LBRKT 4*8 (HEXDIG) RBRKT

termIDList          = (TerminationID / LSBRKT TerminationID 1*(COMMA
TerminationID) RSRBKT)

terminationIDList   = LBRKT TerminationID *(COMMA TerminationID) RBRKT

; pathNAM 的长度不允许超过 64 字符的长度。
pathNAME            = ["*"] NAME *("/" / "*" / ALPHA / DIGIT / "_" / "$")
["@" pathDomainName]

; ABNF 句法规则允许使用两个连续的"." 字符，尽管该字符对于域名是毫无意义的。
; in a path domain name.
pathDomainName      = (ALPHA / DIGIT / "*") *63(ALPHA / DIGIT / "-" /
"*" / ".")

; '*' 表示通配符“ALL”， '$'表示通配符 CHOOSE.
TerminationID       = "ROOT" / pathNAME / "$" / "*"

mediaDescriptor     = MediaToken LBRKT mediaParm *(COMMA mediaParm) RBRKT

; terminationStateDescriptor 最多出现一次
; 以及 streamParm 或 streamDescriptor, 但不允许同时存在。
mediaParm           = (streamParm / streamDescriptor /
terminationStateDescriptor)

; 每个参数最多一次
streamParm          = (localDescriptor / remoteDescriptor /
localControlDescriptor / statisticsDescriptor)

streamDescriptor    = StreamToken EQUAL StreamID LBRKT streamParm *(COMMA
streamParm) RBRKT

localControlDescriptor = LocalControlToken LBRKT localParm *(COMMA localParm)
RBRKT

; 除 propertyParm 之外，其他每个参数最多一次
localParm           = (streamMode / propertyParm / reservedValueMode /
reservedGroupMode)

reservedValueMode   = ReservedValueToken EQUAL ("ON" / "OFF")

reservedGroupMode   = ReservedGroupToken EQUAL ("ON" / "OFF")

streamMode          = ModeToken EQUAL streamModes

streamModes         = (SendonlyToken / RecvonlyToken / SendrecvToken /
InactiveToken / LoopbackToken)

propertyParm        = pkgdName parmValue

; '$'字符为通配符 CHOOSE, '*'字符为通配符 ALL
;
parmValue           = (EQUAL alternativeValue / INEQUAL VALUE)

alternativeValue     = (VALUE
/ LSBRKT VALUE *(COMMA VALUE) RSRBKT
; 子列表 (即 A 和 B 和...)
/ LBRKT VALUE *(COMMA VALUE) RBRKT
; 备选 (即 A 或 B 或...)
/ LSBRKT VALUE COLON VALUE RSRBKT)

```

; 范围

INEQUAL = LWSP (">" / "<" / "#") LWSP ; '#' 表示“不等于”

LSBRKT = LWSP "[" LWSP

RSBRKT = LWSP "]" LWSP

; 注 — 字符“0”不允许存在于允许字符之内。该规则仅适用于 SDP 定义，

; 字符“0”是 SDP 中的非法字符。

localDescriptor = LocalToken LBRKT octetString RBRKT

remoteDescriptor = RemoteToken LBRKT octetString RBRKT

eventBufferDescriptor = EventBufferToken [LBRKT eventSpec*(COMMA eventSpec) RBRKT]

eventSpec = pkgdName [LBRKT eventSpecParameter *(COMMA eventSpecParameter) RBRKT]

eventSpecParameter = (eventStream / eventOther)

eventBufferControl = BufferToken EQUAL eventBufferControlValue

eventBufferControlValue = ("OFF" / LockStepToken)

terminationStateDescriptor

= TerminationStateToken LBRKT terminationStateParm *(COMMA terminationStateParm) RBRKT

; 除 propertyParm 之外，其他每个参数最多一次

terminationStateParm = (propertyParm / serviceStates / eventBufferControl)

serviceStates = ServiceStatesToken EQUAL serviceStatesValue

serviceStatesValue = (TestToken / OutOfSvcToken / InSvcToken)

muxDescriptor = MuxToken EQUAL MuxType terminationIDList

MuxType = (H221Token / H223Token / H226Token / V76Token / extensionParameter / Nx64kToken)

StreamID = UINT16

pkgdName = (PackageName SLASH ItemID) ; 特定参数
/ (PackageName SLASH "*") ; 包中的所有参数
/ ("*" SLASH "*") ; MG 所支持的所有参数

PackageName = NAME

ItemID = NAME

eventsDescriptor = EventsToken [EQUAL RequestID LBRKT requestedEvent *(COMMA requestedEvent) RBRKT]

requestedEvent = pkgdName [LBRKT eventParameter *(COMMA eventParameter) RBRKT]

notifyRegulated = NotifyRegulatedToken [LBRKT (embedWithSig/ embedNoSig) RBRKT]

notifyBehaviour = NotifyImmediateToken / notifyRegulated / NeverNotifyToken

```

; KeepActiveToken、notifyBehaviour、eventDM、ResetEventsDescriptor 和
; eventStream 最多一次
; embedWithSig 或 embedNoSig 最多一次，但不允许同时存在
; KeepActiveToken 和 embedWithSig 不允许同时存在
eventParameter          = (embedWithSig / embedNoSig / KeepActiveToken /
                           eventDM / eventStream / eventOther /
                           notifyBehaviour / ResetEventsDescriptorToken)

embedWithSig            = EmbedToken LBRKT signalsDescriptor [COMMA
                           embedFirst] RBRKT

embedNoSig              = EmbedToken LBRKT embedFirst RBRKT

; 每个参数最多一次
embedFirst              = EventsToken [EQUAL RequestID LBRKT
                           secondRequestedEvent *(COMMA secondRequestedEvent)
                           RBRKT]

secondRequestedEvent    = pkgdName [LBRKT secondEventParameter *(COMMA
                           secondEventParameter) RBRKT]

; embedSig、KeepActiveToken、notifyBehaviour、eventDM、
; ResetEventsDescriptor 或 eventStream 最多一次
; KeepActiveToken 和 embedSig 不允许同时存在
secondEventParameter    = (embedSig / KeepActiveToken / eventDM /
                           eventStream / eventOther / notifyBehaviour /
                           ResetEventsDescriptorToken)

embedSig                = EmbedToken LBRKT signalsDescriptor RBRKT

eventStream             = StreamToken EQUAL StreamID

eventOther              = eventParameterName parmValue

eventParameterName     = NAME

eventDM                = DigitMapToken EQUAL((digitMapName) / (LBRKT
                           digitMapValue RBRKT))

signalsDescriptor       = SignalsToken [LBRKT signalParm *(COMMA signalParm)
                           RBRKT]

signalParm              = signalList / signalRequest

signalRequest           = signalName [LBRKT sigParameter *(COMMA sigParameter)
                           RBRKT]

signalList              = SignalListToken EQUAL signalListId LBRKT
                           signalListParm *(COMMA signalListParm) RBRKT

signalListId           = UINT16

; signalType 正好一次，时长和每一个信号参数最多一次
signalListParm          = signalRequest

signalName              = pkgdName

; sigStream 最多一次，sigSignalType 最多一次，
; sigDuration 最多一次，sigDirection 最多一次，
; sigRequestID 最多一次，sigIntsigDelay 最多一次
; 每个 signalParameterName 最多一次
sigParameter           = sigStream / sigSignalType / sigDuration / sigOther /

```



```

notifyCompletion / KeepActiveToken /
sigDirection / sigRequestID / sigIntsigDelay

sigStream          = StreamToken EQUAL StreamID
sigOther           = sigParameterName parmValue
sigParameterName  = NAME
sigSignalType     = SignalTypeToken EQUAL signalType
signalType        = (OnOffToken / TimeOutToken / BriefToken)
sigDuration       = DurationToken EQUAL UINT16
sigDirection      = DirectionToken EQUAL direction
direction         = ExternalToken / InternalToken / BothToken
sigRequestID      = RequestIDToken EQUAL RequestID
sigIntsigDelay    = IntsigDelayToken EQUAL UINT16
notifyCompletion  = NotifyCompletionToken EQUAL (LBRKT
notificationReason *(COMMA notificationReason)
RBRKT)
notificationReason = TimeOutToken / InterruptByEventToken /
InterruptByNewSignalsDescrToken / OtherReasonToken /
IterationToken

observedEventsDescriptor
= ObservedEventsToken EQUAL RequestID LBRKT
observedEvent *(COMMA observedEvent) RBRKT

; 每个事件的时间, 因为事件可能被缓存
observedEvent      = [TimeStamp LWSP COLON] LWSP pkgdName [LBRKT
observedEventParameter *(COMMA
observedEventParameter) RBRKT]

; eventStream 最多一次, 每个 eventParameterName 最多一次
observedEventParameter = eventStream / eventOther

; 当 AuditCapReply 包含所有事件时, RequestID 参数应当为通配符 "ALL"。
RequestID          = UINT32 / "*"

modemDescriptor    = ModemToken ((EQUAL modemType) / (LSBRKT modemType
*(COMMA modemType) RSBKRT)) [LBRKT propertyParm
*(COMMA propertyParm) RBRKT]

; 除 extensionParameter 之外, 其他参数最多一次
modemType          = V32bisToken / V22bisToken / V18Token / V22Token /
V32Token / V34Token / V90Token / V91Token /
SynchISDNToken / extensionParameter)

digitMapDescriptor = DigitMapToken EQUAL ((LBRKT digitMapValue RBRKT) /
(digitMapName [LBRKT digitMapValue RBRKT]))

digitMapName       = NAME

digitMapValue      = ["T" COLON Timer COMMA] ["S" COLON Timer COMMA] ["L"
COLON Timer COMMA] ["Z" COLON Timer COMMA] digitMap

```

```

Timer = 1*2DIGIT
; T、S、和 L 定时器的单位为秒， Z 定时器的单位为 100 毫秒。
; 因此，T、S、和 L 的范围为 1 至 99 秒，Z 定时器的范围为 100 ms 至 9.9 s

digitMap = (digitString / LWSP "(" LWSP digitStringList LWSP
            )" LWSP)

digitStringList = digitString *(LWSP "|" LWSP digitString)

digitString = 1*(digitStringElement)

digitStringElement = digitPosition [DOT]

digitPosition = digitMapLetter / digitMapRange

digitMapRange = ("x" / (LWSP "[" LWSP digitLetter LWSP "]" LWSP))

digitLetter = *((DIGIT "-" DIGIT) / digitMapLetter)

digitMapLetter = DIGIT ;Basic event symbols
                  / %x41-4B / %x61-6B ; a-k, A-K
                  / "L" / "S" / "T" ;Inter-event timers
                  ; (long, short, start)
                  / "Z" ;Long duration modifier

; 最多一次，DigitMapToken 和 PackagesToken 参数不允许存在于 AuditCapabilities 命令中
auditItem = auditReturnItem / SignalsToken / EventBufferToken /
            EventsToken / indAudterminationAudit

indAudterminationAudit = indAudauditReturnParameter *(COMMA
                    indAudauditReturnParameter)

indAudauditReturnParameter
= indAudmediaDescriptor / indAudeventsDescriptor /
  indAudsignalsDescriptor /
  indAuddigitMapDescriptor /
  indAudeventBufferDescriptor /
  indAudstatisticsDescriptor /
  indAudpackagesDescriptor

indAudmediaDescriptor = MediaToken LBRKT indAudmediaParm *(COMMA
                    indAudmediaParm) RBRKT

; streamParm 或 streamDescriptor, 但不允许同时存在
indAudmediaParm = indAudstreamParm / indAudstreamDescriptor /
                    indAudterminationStateDescriptor

; 最多一次
indAudstreamParm = (indAudlocalControlDescriptor /
                    indAudstatisticsDescriptor /
                    indAudremoteDescriptor / indAudlocalDescriptor )

indAudremoteDescriptor = RemoteToken LBRKT octetString RBRKT
indAudlocalDescriptor = LocalToken LBRKT octetString RBRKT

indAudstreamDescriptor = StreamToken EQUAL StreamID LBRKT indAudstreamParm
                    RBRKT

indAudlocalControlDescriptor
= LocalControlToken LBRKT indAudlocalParm
  *(COMMA indAudlocalParm) RBRKT

; 每个参数最多一次

```

```

indAudlocalParm          = ModeToken [(EQUAL/INEQUAL) streamModes] / pkgdName /
                          propertyParm / ReservedValueToken /
                          ReservedGroupToken
; propertyParm 和 streamModes 仅仅在指定审计条件时被使用。AND 或者 OR 选择逻辑仅适用于关联。

indAudterminationStateDescriptor
    = TerminationStateToken LBRKT
      indAudterminationStateParm RBRKT

; 每个参数最多一次
indAudterminationStateParm
    = pkgdName / propertyParm / ServiceStatesToken
      [(EQUAL/INEQUAL) serviceStatesValue ] / BufferToken
; 当参数值被包含时，隐含为选择操作。AND 或者 OR 选择逻辑仅适用于关联。

indAudeventBufferDescriptor
    = EventBufferToken LBRKT indAudeventSpec RBRKT

indAudeventSpec          = pkgdName [LBRKT indAudeventSpecParameter RBRKT]

indAudeventSpecParameter= eventStream / eventParameterName

indAudeventsDescriptor  = EventsToken [EQUAL RequestID] LBRKT
                          indAudrequestedEvent RBRKT

indAudrequestedEvent     = pkgdName

indAudsignalsDescriptor = SignalsToken LBRKT [indAudsignalParm] RBRKT

indAudsignalParm         = indAudsignalList / indAudsignalRequest

indAudsignalRequest      = signalName [LBRKT indAudsignalRequestParm
    *(COMMA indAudsignalRequestParm) RBRKT]

indAudsignalRequestParm = sigStream / sigRequestID

indAudsignalList         = SignalListToken EQUAL signalListId [LBRKT
    indAudsignalListParm RBRKT]

indAudsignalListParm     = indAudsignalRequest

indAuddigitMapDescriptor= DigitMapToken EQUAL (digitMapName)

indAudstatisticsDescriptor
    = StatsToken LBRKT pkgdName RBRKT

indAudpackagesDescriptor= PackagesToken LBRKT packagesItem RBRKT

serviceChangeDescriptor = ServicesToken LBRKT serviceChangeParm *(COMMA
    serviceChangeParm) RBRKT

; 除 auditItem 之外，其他每个参数最多一次
; serviceChangeAddress 或 serviceChangeMgcId 参数最多一次，但不允许同时存在
; serviceChangeMethod 或 serviceChangeReason 为必选参数
serviceChangeParm        = serviceChangeMethod / serviceChangeReason /
    serviceChangeDelay / serviceChangeAddress /
    serviceChangeProfile / extension / TimeStamp /
    serviceChangeMgcId / serviceChangeVersion /
    ServiceChangeIncompleteToken / auditItem

serviceChangeReplyDescriptor
    = ServicesToken LBRKT servChgReplyParm *(COMMA
    servChgReplyParm) RBRKT

```

```

;最多一次. Version 参数在第一个 ServiceChange 响应中为必选参数
; serviceChangeAddress 或 serviceChangeMgcId 参数最多一次, 但不允许同时存在
servChgReplyParm      = serviceChangeAddress / serviceChangeMgcId /
                        serviceChangeProfile / serviceChangeVersion /
                        TimeStamp

serviceChangeMethod    = MethodToken EQUAL (FailoverToken / ForcedToken /
                        GracefulToken / RestartToken / DisconnectedToken /
                        HandOffToken / extensionParameter)

; serviceChangeReason 参数包含一个原因码和可选的文本描述。
; serviceChangeReason 参数必须采用 quotedString 类型的参数值进行编码。
; quotedString 必须包含一个十进制原因码, 可选的可以跟随一个空格和文本描述字符。
serviceChangeReason    = ReasonToken EQUAL VALUE

serviceChangeDelay     = DelayToken EQUAL UINT32

serviceChangeAddress   = ServiceChangeAddressToken EQUAL (mId / portNumber)

serviceChangeMgcId     = MgcIdToken EQUAL mId

serviceChangeProfile   = ProfileToken EQUAL NAME SLASH Version

serviceChangeVersion   = VersionToken EQUAL Version

extension              = extensionParameter parmValue

packagesDescriptor     = PackagesToken LBRKT packagesItem *(COMMA
                        packagesItem) RBRKT

Version                = 1*2 (DIGIT)

packagesItem           = NAME "-" UINT16

TimeStamp              = Date "T" Time ; 按照 ISO 8601:2004

; 日期 = yyyyymmdd
Date                   = 8 (DIGIT)

; 时间 = hhmmssss
Time                   = 8 (DIGIT)

statisticsDescriptor   = StatsToken LBRKT statisticsParameter *(COMMA
                        statisticsParameter) RBRKT

; 每个参数最多一次
statisticsParameter    = pkgdName [EQUAL VALUE /
                        (LSBRKT VALUE *(COMMA VALUE) RSBKRT)]

topologyDescriptor     = TopologyToken LBRKT topologyTriple *(COMMA
                        topologyTriple) RBRKT

topologyTriple         = terminationA COMMA terminationB COMMA
                        topologyDirection [COMMA eventStream]

terminationA           = TerminationID

terminationB           = TerminationID

topologyDirection     = BothwayToken / IsolateToken / OnewayToken /
                        OnewayExternalToken / OnewayBothToken

```

```

priority                = PriorityToken EQUAL UINT16

iepsValue               = IEPSToken EQUAL ("ON" / "OFF")

emergencyValue         = EmergencyValueToken EQUAL (EmergencyToken /
EmergencyOffToken)

contextAttrDescriptor  = ContextAttrToken LBRKT (contextIdList /
propertyParm *(COMMA propertyParm)) RBRKT

; 当使用 contextIdList 结构时, actionReply 中所包含的 ContextID 应当与相应的
; actionRequest 请求中的 ContextID 相同
contextIdList          = ContextListToken EQUAL LBRKT ContextID
                        *(COMMA ContextID) RBRKT

extensionParameter     = "X" ("- " / "+") 1*6(ALPHA / DIGIT)

; octetString 用于参照 RFC 2327 来定义 SDP。
; 如果 RFC 2327 中的 CRLF 被使用时, 应当尤其注意。
; 为了安全起见, 本句法规定使用"EOL"。
; 当"}" 出现在 SDP 中, 该字符由"\\"进行隔离, 例如,"\"}
octetString            = *(nonEscapeChar)

nonEscapeChar          = ("\\" / %x01-7C / %x7E-FF)

; 注 - 双引号字符不允许出现在 quotedString 字符串中。
quotedString           = DQUOTE *(SafeChar / EOL / %x80-FF / RestChar / WSP)
                        DQUOTE

UINT16                 = 1*5(DIGIT) ; %x0-FFFF

UINT32                 = 1*10(DIGIT) ; %x0-FFFFFFFF

NAME                   = ALPHA *63(ALPHA / DIGIT / "_")

VALUE                  = quotedString / 1*(SafeChar / %x80-FF)

SafeChar               = DIGIT / ALPHA / "+" / "-" / "&" /
                        "!" / " " / "/" / "!" / "?" / "@" /
                        "^" / "~" / "~" / "*" / "$" / "\" /
                        "(" / ")" / "%" / "|" / "."

EQUAL                  = LWSP %x3D LWSP ; "="

COLON                  = %x3A ; ":"

LBRKT                  = LWSP %x7B LWSP ; "{"

RBRKT                  = LWSP %x7D LWSP ; "}"

COMMA                  = LWSP %x2C LWSP ; ","

DOT                    = %x2E ; "."

SLASH                  = %x2F ; "/"

ALPHA                  = %x41-5A / %x61-7A ; A-Z / a-z

DIGIT                  = %x30-39 ; 0-9

DQUOTE                 = %x22 ; " (Double Quote)

```

```

HEXDIG      = DIGIT / "A" / "B" / "C" / "D" / "E" / "F"

SP          = %x20                ; space

HTAB       = %x09                ; horizontal tab

CR         = %x0D                ; Carriage return

LF         = %x0A                ; linefeed

LWSP       = *(WSP / COMMENT / EOL)

EOL        = (CR [LF] / LF)

WSP        = SP / HTAB          ; white space

SEP        = (WSP / EOL / COMMENT) LWSP

COMMENT    = ";" *(SafeChar/ RestChar / WSP / %x22) EOL

RestChar   = ";" / "[" / "]" / "{" / "}" / ":" / "," / "#" /
            "<" / ">" / "="

```

; 新增加的 sigParameter 参数必须采用"SPA*"格式, "*"可以是任意字符, 例如 SPAM
; 新增加的 eventParameter 参数必须采用"EPA*"格式, "*"可以是任意字符, 例如 EPAD

```

AddToken    = ("Add"                / "A")
AndAUDITSelectToken = ("ANDLgc")
AuditToken  = ("Audit"              / "AT")
AuditCapToken = ("AuditCapability"  / "AC")
AuditValueToken = ("AuditValue"     / "AV")
AuthToken   = ("Authentication"    / "AU")
BothToken   = ("Both"               / "B")
BothwayToken = ("Bothway"           / "BW")
BriefToken  = ("Brief"              / "BR")
BufferToken = ("Buffer"             / "BF")
CtxToken    = ("Context"            / "C")
ContextAuditToken = ("ContextAudit" / "CA")
ContextAttrToken = ("ContextAttr"   / "CT")
ContextListToken = ("ContextList"   / "CLT")
DigitMapToken = ("DigitMap"         / "DM")
DirectionToken = ("SPADirection"    / "SPADI")
DisconnectedToken = ("Disconnected" / "DC")
DelayToken  = ("Delay"              / "DL")
DurationToken = ("Duration"         / "DR")
EmbedToken  = ("Embed"              / "EM")
EmergencyToken = ("Emergency"       / "EG")
EmergencyOffToken = ("EmergencyOff" / "EGO")
EmergencyValueToken = ("EmergencyValue" / "EGV")
ErrorToken  = ("Error"              / "ER")
EventBufferToken = ("EventBuffer"   / "EB")
EventsToken = ("Events"             / "E")
ExternalToken = ("External"         / "EX")
FailoverToken = ("Failover"        / "FL")
ForcedToken  = ("Forced"            / "FO")
GracefulToken = ("Graceful"         / "GR")
H221Token   = ("H221")
H223Token   = ("H223")
H226Token   = ("H226")
HandOffToken = ("HandOff"           / "HO")
IEPSToken   = ("IEPSCall"          / "IEPS")
ImmAckRequiredToken = ("ImmAckRequired" / "IA")
InactiveToken = ("Inactive"         / "IN")

```

InternalToken	= ("Internal"	/ "IT")
IntsigDelayToken	= ("Intersignal"	/ "SPAIS")
IsolateToken	= ("Isolate"	/ "IS")
InSvcToken	= ("InService"	/ "IV")
InterruptByEventToken	= ("IntByEvent"	/ "IBE")
InterruptByNewSignalsDescrToken	= ("IntBySigDescr"	/ "IBS")
IterationToken	= ("Iteration"	/ "IR")
KeepActiveToken	= ("KeepActive"	/ "KA")
LocalToken	= ("Local"	/ "L")
LocalControlToken	= ("LocalControl"	/ "O")
LockStepToken	= ("LockStep"	/ "SP")
LoopbackToken	= ("Loopback"	/ "LB")
MediaToken	= ("Media"	/ "M")
MegacopToken	= ("MEGACO"	/ "I")
MessageSegmentToken	= ("Segment"	/ "SM")
MethodToken	= ("Method"	/ "MT")
MgcIdToken	= ("MgcIdToTry"	/ "MG")
ModeToken	= ("Mode"	/ "MO")
ModifyToken	= ("Modify"	/ "MF")
ModemToken	= ("Modem"	/ "MD")
MoveToken	= ("Move"	/ "MV")
MTPToken	= ("MTP")	
MuxToken	= ("Mux"	/ "MX")
NeverNotifyToken	= ("NeverNotify"	/ "NBNN")
NotifyToken	= ("Notify"	/ "N")
NotifyCompletionToken	= ("NotifyCompletion"	/ "NC")
NotifyImmediateToken	= ("ImmediateNotify"	/ "NBIN")
NotifyRegulatedToken	= ("RegulatedNotify"	/ "NBRN")
Nx64kToken	= ("Nx64Kservice"	/ "N64")
ObservedEventsToken	= ("ObservedEvents"	/ "OE")
OnewayToken	= ("Oneway"	/ "OW")
OnewayBothToken	= ("OnewayBoth"	/ "OWB")
OnewayExternalToken	= ("OnewayExternal"	/ "OWE")
OnOffToken	= ("OnOff"	/ "OO")
OrAUDITselectToken	= ("ORLgc")	
OtherReasonToken	= ("OtherReason"	/ "OR")
OutOfSvcToken	= ("OutOfService"	/ "OS")
PackagesToken	= ("Packages"	/ "PG")
PendingToken	= ("Pending"	/ "PN")
PriorityToken	= ("Priority"	/ "PR")
ProfileToken	= ("Profile"	/ "PF")
ReasonToken	= ("Reason"	/ "RE")
RecvonlyToken	= ("ReceiveOnly"	/ "RC")
ReplyToken	= ("Reply"	/ "P")
ResetEventsDescriptorToken	= ("ResetEventsDescriptor"	/ "RSE")
RestartToken	= ("Restart"	/ "RS")
RemoteToken	= ("Remote"	/ "R")
RequestIDToken	= ("SPARrequestID"	/ "SPARQ")
ReservedGroupToken	= ("ReservedGroup"	/ "RG")
ReservedValueToken	= ("ReservedValue"	/ "RV")
SegmentationCompleteToken	= ("END"	/ "&")
SendonlyToken	= ("SendOnly"	/ "SO")
SendrecvToken	= ("SendReceive"	/ "SR")
ServicesToken	= ("Services"	/ "SV")
ServiceStatesToken	= ("ServiceStates"	/ "SI")
ServiceChangeIncompleteToken	= ("ServiceChangeInc"	/ "SIC")
ServiceChangeToken	= ("ServiceChange"	/ "SC")
ServiceChangeAddressToken	= ("ServiceChangeAddress"	/ "AD")
SignalListToken	= ("SignalList"	/ "SL")

```

SignalsToken           = ("Signals"           / "SG")
SignalTypeToken        = ("SignalType"        / "SY")
StatsToken             = ("Statistics"       / "SA")
StreamToken            = ("Stream"           / "ST")
SubtractToken         = ("Subtract"         / "S")
SynchISDNToken        = ("SynchISDN"        / "SN")
TerminationStateToken = ("TerminationState"    / "TS")
TestToken              = ("Test"            / "TE")
TimeOutToken          = ("TimeOut"          / "TO")
TopologyToken          = ("Topology"         / "TP")
TransToken             = ("Transaction"     / "T")
ResponseAckToken      = ("TransactionResponseAck" / "K")
V18Token              = ("V18")
V22Token              = ("V22")
V22bisToken           = ("V22b")
V32Token              = ("V32")
V32bisToken           = ("V32b")
V34Token              = ("V34")
V76Token              = ("V76")
V90Token              = ("V90")
V91Token              = ("V91")
VersionToken          = ("Version"         / "V")

```

B.3 十六进制字节编码

十六进制编码是使用16进制数表示一个字节字串的方法，且一个字节由两个16进制数组成。当协议使用文本编码方式时，字节应该使用十六进制编码方式。

每个字节将8个比特编码成两个16进制的数。比特0应最先发送，比特7应最后发送。

比特7-4编码成第一个十六进制数，且比特7作为MSB，比特4作为LSB。比特3-0编码成第二个十六进制数，且比特3作为MSB，比特0作为LSB。

例如：

字节位模式	十六进制编码
00011011	D8
11100100	27
10000011 10100010 11001000 00001001	C1451390

B.4 十六进制字节序列

十六进制字节序列由偶数个十六进制数组成，且以<CR>字符结束。

附件 C

媒体流属性标签

当协议采用二进制编码方式时，Local、Remote和LocalControl描述符中包含的属性用一组标签值进行定义。本附件定义的属性包含4个部分，分别是属性标识（PropertyID）、属性标签（Property Tag）、属性类型（Type）和属性值。其中，如果参数值定义引用了参考性文件，则该参数值必须是标准的参数值。如果参数值定义未引用参考性文件，则该参数值仅作为参考值。

引用本建议书附件C中的属性将遵循PackageID/PropertyID的结构；然而，附件C本身并不是一个包类型。因此，当采用二进制编码方式是，附件C将由0x0000 标识为PackageID，当采用文本方式编码时，附件C将由anxc标识为PackageID。当H.248.1协议采用文本编码时，仅仅当某些属性既未在包中进行定义也不能由SDP进行描述时，附件C中的属性才允许被使用。附件C定义的属性不允许被相互封装。

标签由一个16进制数唯一标识。当MGC需要设置属性的参数值时，参照7.1.1的规定，MGC可对参数值进行部分指定。

本附件中列举的所有属性都为可选项，例如，可选择支持C.3定义的3个属性，或者仅支持C.8定义的5个属性。

枚举类型的属性值由括号中的参数值进行标识，例如，Send(0), Receive(1)。在协议编码时，附件C中属性类型包含有“N比特”或“M字节”时，其类型应该视为字节字符串。属性类型包含有“N比特整数”时，其类型必须被视为整数类型。“字符”类型的属性必须编码成IA5String格式。

当某种类型的属性字段少于一个字节时，必须从1个字节的最低比特位开始设置参数值。

C.1 通用媒体属性

属性标识符	属性标签	类型	参数值
Media	1001	枚举	Audio(0), Video(1), Data(2)
Transmission mode	1002	枚举	Send (0), Receive (1), Send&Receive (2)
Number of Channels	1003	无符号整数	0-255
Sampling rate	1004	无符号 整数	0-2 ³²
Bitrate	1005	整数	(0..4294967295) 注 — 单位为100 bit/s。
Acodec	1006	八位组串	音频编解码器类型： 参见：ITU-T Q.765.5 建议书 非ITU-T建议书定义的音频编解码器类型应使用组织标识码来识别。
Samplepp	1007	无符号整数	每一个包中的最大抽样或帧数： 0..65535
Silencesupp	1008	布尔型	静音压缩：真/假
Encrypttype	1009	八位组串	参见：ITU H.245建议书
Encryptkey	100A	八位组串大小(0..65535)	加密密钥 参见：H.235
Echocanc	100B		未使用。参见E.13中可能的回声控制属性示例
Gain	100C	无符号整数	未使用。参见E.13中的可用增益属性

Jitterbuff	100D	无符号整数	抖动缓存大小 (ms) : 0..65535
PropDelay	100E	无符号整数	传播时延: 0..65535 两个媒体网关之间承载连接的最大传播时延为毫秒级, 最大传播时延与承载技术有关。
RTPpayload	100F	整数	RTP协议子集为具有最小控制的音频和视频会议定义的Payload类型 参见: RFC 1890
Ptime	1010	整数	打包时长 该值指示媒体流在包中使用的时长标识, 单位毫秒。 参见: RFC 2327

C.2 Mux属性

属性标识符	属性标签	类型	参数值
H.221	2001	八位组串	H222LogicalChannelParameters 参见: H.245
H223	2002	八位组串	H223LogicalChannelParameters 参见: H.245
V76	2003	八位组串	V76LogicalChannelParameters 参见: H.245
H2250	2004	八位组串	H2250LogicalChannelParameters 参见: H.245

C.3 通用承载属性

属性标识符	属性标签	类型	参数值
Mediatx	3001	枚举	媒体传输类型 TDM Circuit(0), ATM(1), FR(2), Ipv4(3), Ipv6(4)等
BIR	3002	4 字节	参数值由传输技术决定
NSAP	3003	1-20字节	参见NSAP。 参见: X.213附件A

C.4 通用ATM属性

属性标识符	属性标签	类型	参数值
AESA	4001	20字节	ATM端系统地址
VPVC	4002	4字节: VPCI位于两个最低有效字节, VCI位于两个最高有效字节	VPCI/VCI 参见: ITU-T Q.2931建议书
SC	4003	枚举	业务类别: CBR(0), nrt-VBR1(1), nrt-VBR2(2), nrt-VBR3(3), rt-VBR1(4), rt-VBR2(5), rt-VBR3(6), UBR1(7), UBR2(8), ABR(9). 参见: ATM论坛UNI 4.0
BCOB	4004	5比特整数	宽带承载类别 参见: ITU-T Q.2961.2建议书
BBTC	4005	7比特整数	宽带传送能力 参见: ITU-T Q.2961.1建议书
ATC	4006	枚举	I.371 ATM业务量能力 DBR(0), SBR1(1), SBR2(2), SBR3(3), ABT/IT(4), ABT/DT(5), ABR(6) 参见: ITU-T I.371建议书
STC	4007	2比特	消波敏感度: 比特 <u>2 1</u> 0 0 对消波不敏感 0 1 对消波敏感 参见: ITU-T Q.2931建议书
UPCC	4008	2比特	用户平面连接配置: 比特 <u>2 1</u> 0 0 点对点 0 1 点对多点 参见: ITU-T Q.2931建议书
PCR0	4009	24比特整数	Peak信元速率 (对于CLP = 0) 参见: ITU-T Q.2931建议书
SCR0	400A	24比特整数	可维持信元速率 (对于CLP = 0) 参见: ITU-T Q.2961.1建议书
MBS0	400B	24比特整数	最大突发长度 (对于CLP = 0) 参见: ITU-T Q.2961.1建议书
PCR1	400C	24比特整数	峰值信元速率 (对于CLP = 0 + 1) 参见: ITU-T Q.2931建议书
SCR1	400D	24比特整数	可维持信元速率 (对于CLP = 0 + 1) 参见: ITU-T Q.2961.1建议书
MBS1	400E	24比特整数	最大突发长度 (对于CLP = 0 + 1) 参见: ITU-T Q.2961.1建议书

属性标识符	属性标签	类型	参数值	
BEI	400F	布尔型	尽力而为指示符 (BEI) 值1表示BEI将含在ATM信令内；值0表示BEI将不含在ATM信令内。 参见：ATM论坛UNI 4.0	
TI	4010	布尔型	加标签指示符 值0表示不允许加标签；值1表示要求加标签。 参见：ITU-T Q.2961.1建议书	
FD	4011	布尔型	帧丢弃 值0表示不允许丢帧；值1表示允许丢帧。 参见：ATM论坛UNI 4.0	
A2PCDV	4012	24比特整数	可接受2点CDV 参见：ITU-T Q.2965.2建议书	
C2PCDV	4013	24比特整数	累积2点CDV 参见：ITU-T Q.2965.2建议书	
APPCDV	4014	24比特整数	可接受点对点CDV 参见：ATM论坛UNI 4.0	
CPPCDV	4015	24比特整数	累积点对点CDV 参见：ATM论坛UNI 4.0	
ACLR	4016	8比特整数	可接受信元损失率 参见：ITU-T Q.2965.2建议书, ATM论坛 UNI 4.0	
MEETD	4017	16比特整数	最大端对端转换时延 参见：ITU-T Q.2965.2建议书, ATM论坛 UNI 4.0	
CEETD	4018	16比特整数	累积端对端转换时延 参见：ITU-T Q.2965.2建议书, ATM论坛 UNI 4.0	
QoSClass	4019	整数 0-5	QoS类别	
			QoS类别	含义
			0	ITU-T Q.2961.2建议书规定的与ATC有关的默认QoS
			1	严格
			2	容错
			3	双水平
			4	无界定
			5	严格双水平
			参见：ITU-T Q.2965.1建议书	

属性标识符	属性标签	类型	参数值
AALtype	401A	1字节	AAL类型 比特 <u>8 7 6 5 4 3 2 1</u> 00000000 AAL 用于语音 00000001 AAL 类型1 00000010 AAL 类型2 00000011 AAL 类型3/4 00000101 AAL 类型5 00010000 用户定义AAL 参见：ITU-T Q.2931建议书

C.5 帧中继属性

属性标识符	属性标签	类型	参数值
DLCI	5001	无符号整数	数据链路连接标识
CID	5002	无符号整数	子通道标识
SID/Noiselevel	5003	无符号整数	静音插入描述符
Primary Payload type	5004	无符号整数	主要的负荷类型 包括传真和编码

C.6 IP

属性标识符	属性标签	类型	参数值
IPv4	6001	32比特 Ipv4Address	IPv4地址 参见：IETF RFC 791
IPv6	6002	128比特	IPv6地址 参见：IETF RFC 2460
Port	6003	无符号整数	0..65535
Porttype	6004	枚举	TCP(0), UDP(1), SCTP(2)
RtcpbwRS	6005	整数	RS RTCP带宽修饰符表明指配给（由RTP规范规定的）有效数据发送器的RTCP带宽 参见：IETF RFC 3556
RtcpbwRR	6006	整数	RR RTCP带宽修饰符表明指配给其他接收者（即接收器）的RTCP带宽 参见：IETF RFC 3556

C.7 ATM AAL 2

属性标识符	属性标签	类型	参数值
AESA	7001	20字节	在引用的建议书中规定的AAL业务端点地址 ESEA NSEA 参见: ITU-T Q.2630.1建议书
BIR	见C.3	4字节	在引用的建议书中规定的用户自定义参考值 SUGR 参见: ITU-T Q.2630.1建议书
ALC	7002	12字节	在引用的建议书中规定的AAL 2链路特性 最大/平均CPS-SDU比特率; 最大/平均CPS-SDU大小 参见: ITU-T Q.2630.1建议书
SSCS	7003	I.366.2: 音频(8字节); 多速率(3字节), 或 I.366.1: 有保证SAR (14字节); 无保证SAR (7字节)。	在下列建议书中规定的业务特定融合子层信息: - ITU-T Q.2630.1建议书, 并用于: - ITU-T I.366.2建议书: 音频/多速率; - ITU-T I.366.1建议书: 有保证/无保证SAR。 参见: ITU-T Q.2630.1、I.366.1和I.366.2建议书
SUT	7004	1..254字节	在引用的建议书中规定的被服务用户传送参数 参见: ITU-T Q.2630.1建议书
TCI	7005	布尔型	在引用的建议书中规定的测试连接指示符 参见: ITU-T Q.2630.1建议书
Timer_CU	7006	32比特整数	定时器CU 发送前部分填充信元要维持的毫秒数
MaxCPSSDU	7007	8比特整数	公共部分子层最大SDU 参见: ITU-T Q.2630.1建议书
CID	7008	8比特	子通道标识: 0-255 参见: ITU-T I.363.2建议书

C.8 ATM AAL 1

属性标识符	属性标签	类型	参数值
BIR	见C.3中的表	4-29字节	GIT (通用标识符传送) 参见: ITU-T Q.2941.1建议书
AAL1ST	8001	1字节	AAL 1子层 比特 <u>8 7 6 5 4 3 2 1</u> 00000000 空 00000001 64 kbit/s上的话带信号传送 00000010 电路传送 00000100 高质量信号传送 00000101 视频信号传送 参见: ITU-T Q.2931建议书
CBRR	8002	1字节	CBR速率 比特 <u>8 7 6 5 4 3 2 1</u> 00000001 64 kbit/s 00000100 1544 kbit/s 00000101 6312 kbit/s 00000110 32 064 kbit/s 00000111 44 736 kbit/s 00001000 97 728 kbit/s 00010000 2048 kbit/s 00010001 8448 kbit/s 00010010 34 368 kbit/s 00010011 139 264 kbit/s 01000000 $n \times 64$ kbit/s 01000001 $n \times 8$ kbit/s 参见: ITU-T Q.2931建议书
MULT	见C.9中的表		复用器, 或 $n \times 64k/8k/300$ 参见: ITU-T Q.2931建议书
SCRI	8003	1字节	源时钟频率恢复方法 比特 <u>8 7 6 5 4 3 2 1</u> 00000000 空 00000001 SRTS 00000010 ACM 参见: ITU-T Q.2931建议书
ECM	8004	1字节	纠错方法 比特 <u>8 7 6 5 4 3 2 1</u> 00000000 空 00000001 FEC — 损失 00000010 FEC — 延迟 参见: ITU-T Q.2931建议书
SDTB	8005	16比特整数	结构数据传送块大小 SDT CBR业务块大小 参见: ITU-T I.363.1建议书

属性标识符	属性标签	类型	参数值
PFCI	8006	8比特整数	部分填充信元标识符 1-47 参见：ITU-T I.363.1建议书

C.9 承载能力

参考ITU-T Q.931建议书，以下表格对Q.931消息单元中的承载能力进行规定，而不是针对低层网络的承载能力。

属性标识符	属性标签	类型	参数值
TMR	9001	1字节	传输媒体要求 (ITU-T Q.763建议书) 比特 <u>8 7 6 5 4 3 2 1</u> 00000000 语音 00000001 备用 00000010 64 kbit/s不受限 00000011 3.1 kHz音频 00000100 保留用于交互语音 (业务2)/64 kbit/s不受限 (业务1) 00000101 保留用于交互64 kbit/s不受限 (业务1)/语音 (业务2) 00000110 64 kbit/s preferred 00000111 2 × 64 kbit/s不受限 00001000 384 kbit/s不受限 00001001 1536 kbit/s不受限 00001010 1920 kbit/s不受限 00001011 至 备用 00001111 00010000 3 × 64 kbit/s不受限 00010001 4 × 64 kbit/s不受限 00010010 5 × 64 kbit/s不受限 00010011 备用 00010100 7 × 64 kbit/s不受限 00010101 8 × 64 kbit/s不受限 00010110 9 × 64 kbit/s不受限 00010111 10 × 64 kbit/s不受限 00011000 11 × 64 kbit/s不受限 00011001 12 × 64 kbit/s不受限 00011010 13 × 64 kbit/s不受限 00011011 14 × 64 kbit/s不受限 00011100 15 × 64 kbit/s不受限 00011101 16 × 64 kbit/s不受限 00011110 17 × 64 kbit/s不受限 00011111 18 × 64 kbit/s不受限 00100000 19 × 64 kbit/s不受限 00100001 20 × 64 kbit/s不受限 00100010 21 × 64 kbit/s不受限 00100011 22 × 64 kbit/s不受限 00100100 23 × 64 kbit/s不受限 00100101 备用

属性标识符	属性标签	类型	参数值
			00100110 25×64 kbit/s不受限 00100111 26×64 kbit/s不受限 00101000 27×64 kbit/s不受限 00101001 28×64 kbit/s不受限 00101010 29×64 kbit/s不受限 00101011 至 备用 11111111 参见: ITU-T Q.763建议书
TMRSR	9002	1字节	传输媒体要求子速率 0 未指定 1 8 kbit/s 2 16 kbit/s 3 32 kbit/s
Contcheck	9003	布尔型	连续性检测 0 该电路不要求使用连续性检测 1 该电路要求使用连续性检测 参见: ITU-T Q.763建议书
ITC	9004	5比特	信息传送能力 比特 <u>54321</u> 00000 语音 01000 不受限数字信息 01001 受限数字信息 10000 3.1 kHz音频 10001 包含信号音和通知音的不受限数字信息 11000 视频 所有其他值留用 参见: ITU-T Q.763建议书
TransMode	9005	2比特	传送模式 比特 <u>21</u> 00 电路模式 10 分组模式 参见: ITU-T Q.931建议书
TransRate	9006	5比特	传送速率 比特 <u>54321</u> 00000 该代码必须用于分组模型信号 10000 64 kbit/s 10001 2×64 kbit/s 10011 384 kbit/s 10101 1536 kbit/s 10111 1920 kbit/s 11000 多速率 (64 kbit/s基本速率) 参见: ITU-T Q.931建议书

属性标识符	属性标签	类型	参数值
MULT	9007	7比特	速率倍增因子 2至n (B通道的最大数目) 的任何值 参见: ITU-T Q.931建议书
layer1prot	9008	5比特	第一层用户信息协议 比特 <u>5 4 3 2 1</u> 0 0 0 0 1 ITU-T标准化速率适配V.110 及X.30 0 0 0 1 0 ITU-T G.711建议书 μ 率 0 0 0 1 1 ITU-T G.711建议书A率 0 0 1 0 0 ITU-T G.726建议书32 kbit/s ADPCM 及ITU-T I.460建议书 0 0 1 0 1 ITU-T H.221和H.242建议书 0 0 1 1 0 ITU-T H.223和H.245建议书 0 0 1 1 1 非ITU-T标准化速率适配 0 1 0 0 0 ITU-T标准化速率适配V.120. 0 1 0 0 1 ITU-T标准化速率适配X.31 HDLC 标记填充 所有其他值留用 参见: ITU-T Q.931建议书
syncasync	9009	布尔型	同步/异步 0 同步数据 1 异步数据 参见: ITU-T Q.931建议书
negotiation	900A	布尔型	协商 0 允许带内协商 1 不允许带内协商 参见: ITU-T Q.931建议书

属性标识符	属性标签	类型	参数值
Userrate	900B	5比特	用户速率 比特 <u>5 4 3 2 1</u> 0 0 0 0 0 速率由ITU-T I.460建议书规定的E比特表明或带内协商 0 0 0 0 1 0.6 kbit/s ITU-T V.6和X.1建议书 0 0 0 1 0 1.2 kbit/s ITU-T V.6建议书 0 0 0 1 1 2.4 kbit/s ITU-T V.6和X.1建议书 0 0 1 0 0 3.6 kbit/s ITU-T V.6建议书 0 0 1 0 1 4.8 kbit/s ITU-T V.6和X.1建议书 0 0 1 1 0 7.2 kbit/s ITU-T V.6建议书 0 0 1 1 1 8 kbit/s ITU-T I.460建议书 0 1 0 0 0 9.6 kbit/s ITU-T V.6和X.1建议书 0 1 0 0 1 14.4 kbit/s ITU-T V.6建议书 0 1 0 1 0 16 kbit/s ITU-T I.460建议书 0 1 0 1 1 19.2 kbit/s ITU-T V.6建议书 0 1 1 0 0 32 kbit/s ITU-T I.460建议书 0 1 1 0 1 38.4 kbit/s ITU-T V.110建议书 0 1 1 1 0 48 kbit/s ITU-T V.6和X.1建议书 0 1 1 1 1 56 kbit/s ITU-T V.6建议书 1 0 0 1 0 57.6 kbit/s ITU-T V.14建议书扩展 1 0 0 1 1 28.8 kbit/s ITU-T V.110建议书 1 0 1 0 0 24 kbit/s ITU-T V.110建议书 1 0 1 0 1 0.1345 kbit/s ITU-T X.1建议书 1 0 1 1 0 0.100 kbit/s ITU-T X.1建议书 1 0 1 1 1 0.075/1.2 kbit/s ITU-T V.6和X.1建议书 1 1 0 0 0 1.2/0.075 kbit/s ITU-T V.6和X.1建议书 1 1 0 0 1 0.050 kbit/s ITU-T V.6和X.1建议书 1 1 0 1 0 0.075 kbit/s ITU-T V.6和X.1建议书 1 1 0 1 1 0.110 kbit/s ITU-T V.6和X.1建议书 1 1 1 0 0 0.150 kbit/s ITU-T V.6和X.1建议书 1 1 1 0 1 0.200 kbit/s ITU-T V.6和X.1建议书 1 1 1 1 0 0.300 kbit/s ITU-T V.6和X.1建议书 1 1 1 1 1 12 kbit/s ITU-T V.6建议书 所有其他值留用 参见: ITU-T Q.931建议书
INTRATE	900C	2比特	中间速率 比特 <u>2 1</u> 0 0 未用 0 1 8 kbit/s 1 0 16 kbit/s 1 1 32 kbit/s 参见: ITU-T Q.931建议书
nictx	900D	布尔型	发送方网络独立时钟 (NIC) 0 不需发带有网络独立时钟的送数据 1 需要发带有网络独立时钟的送数据 参见: ITU-T Q.931建议书

属性标识符	属性标签	类型	参数值
nicrx	900E	布尔型	接收方网络独立时钟 (NIC) 0 无法接受带有网络独立时钟的数据 (即发送器不支持这一选项程序) 1 可以接受带有网络独立时钟的数据 (即发送器支持这一选项程序) 参见: ITU-T Q.931建议书
flowconttx	900F	布尔型	发送 (Tx) 流量控制 0 无需发送带有流量控制机制的数据 1 需要发送带有流量控制机制的数据 参见: ITU-T Q.931建议书
flowcontrx	9010	布尔型	接收 (Rx) 流量控制 0 无法接受带有流量控制机制的数据 (即发送器不支持这一选项程序) 1 可以接受带有流量控制机制的数据 (即发送器支持这一选项程序) 参见: ITU-T Q.931建议书
rateadapthdr	9011	布尔型	速率适配头/无适配头 0 不使用速率适配头 1 使用速率适配头 参见: ITU-T Q.931建议书
multiframe	9012	布尔型	数据链路支持多帧建立 0 允许多帧建立。 且只允许UI 帧 1 支持多帧建立 参见: ITU-T Q.931建议书
OPMODE	9013	布尔型	操作模式 0 比特透明模式操作方式 1 协议敏感操作方式 参见: ITU-T Q.931建议书
llidnegot	9014	布尔型	逻辑链路标识符协商 0 默认值, 仅LLI = 256 1 完整协议协商 参见: ITU-T Q.931建议书
assign	9015	布尔型	被配方/分配方 0 消息协商符为 “默认被配方” 1 消息协商符为 “仅配方” 参见: ITU-T Q.931建议书
inbandneg	9016	布尔型	带内/带外协商 0 在临时信令连接上采用USER INFORMATION 消息完成协商 1 在逻辑通道0上完成带内协商 参见: ITU-T Q.931建议书

属性标识符	属性标签	类型	参数值
stopbits	9017	2比特	停止比特的数目 比特 <u>2 1</u> 0 0 未用 0 1 1 比特 1 0 1.5比特 1 1 2比特 参见：ITU-T Q.931建议书
databits	9018	2比特	数据比特的数目，不包括校验位，若存在该位的话 比特 <u>2 1</u> 0 0 未用 0 1 5比特 1 0 7比特 1 1 8比特 参见：ITU-T Q.931建议书
parity	9019	3比特	校验位信息 比特 <u>3 2 1</u> 0 0 0 奇校验位 0 1 0 偶校验位 0 1 1 无 1 0 0 强制为 0 1 0 1 强制为 1 所有其他值留用 参见：ITU-T Q.931建议书
duplexmode	901A	布尔型	双工模式 0 半双工 1 全双工 参见：ITU-T Q.931建议书

属性标识符	属性标签	类型	参数值
modem	901B	6比特	Modem 类型 比特 <u>6 5 4 3 2 1</u> 0 0 0 0 0 至 国内用途 0 0 0 1 0 1 0 1 0 0 0 ITU-T V.21建议书 0 1 0 0 1 ITU-T V.22建议书 0 1 0 0 1 ITU-T V.22建议书 <i>bis</i> 0 1 0 1 0 ITU-T V.23建议书 0 1 0 1 0 ITU-T V.26建议书 0 1 1 0 0 ITU-T V.26建议书 <i>bis</i> 0 1 0 1 1 ITU-T V.26建议书 <i>ter</i> 0 1 1 0 0 ITU-T V.27建议书 0 1 1 0 0 ITU-T V.27建议书 <i>bis</i> 0 1 1 0 1 ITU-T V.27建议书 <i>ter</i> 0 1 1 0 1 ITU-T V.29建议书 0 1 1 1 0 ITU-T V.32建议书 0 1 1 1 1 ITU-T V.34建议书 1 0 0 0 0 至 国内用途 1 0 1 1 1 1 1 0 0 0 至 用户特定 1 1 1 1 1 参见: ITU-T Q.931建议书
layer2prot	901C	5比特	第二层用户信息协议 比特 <u>5 4 3 2 1</u> 0 0 0 1 0 ITU-T Q.921/I.441建议书 0 0 1 1 0 ITU-T X.25建议书, 链路层 0 1 1 0 0 LAN逻辑链路控制 (ISO/IEC 8802-2) 所有其他值留用 参见: ITU-T Q.931建议书
layer3prot	901D	5比特	第三层用户信息协议 比特 <u>5 4 3 2 1</u> 0 0 0 1 0 ITU-T Q.931建议书 0 0 1 1 0 ITU-T X.25建议书, 分组层 0 1 0 1 1 ISO/IEC TR 9577 (网络层协议标识) 所有其他值留用 参见: ITU-T Q.931建议书

属性标识符	属性标签	类型	参数值
addlayer3prot	901E	字节	第三层附加用户信息协议 比特 比特 <u>4321</u> <u>4321</u> 11001100网际协议 (RFC 791) (ISO/IEC TR 9577) 11001111点对点协议 (RFC 1661) 参见: ITU-T Q.931建议书
DialledN	901F	30字节	被叫号码
DiallingN	9020	30字节	主叫号码
ECHOICI	9021		未用。见E.13中可能的回声抑制属性示例
NCI	9022	1字节	连接指示符的性质 比特 <u>21</u> 卫星电路指示符 00 连接中无卫星电路 01 连接中有一条卫星电路 10 连接中有两条卫星电路 11 备用 比特 <u>43</u> 连续性检测指示符 00 无需连续性检测 01 该电路需要连续性检测 10 对前一条电路进行连续性检测 11 备用 比特 <u>5</u> 回声抑制设备指示符 0 未包含输出回声抑制设备 1 包含输出回声抑制设备 比特 <u>876</u> 备用 参见: ITU-T Q.763建议书
USI	9023	八位组串	用户业务信息 参见: 3.57/Q.763

C.10 AAL 5属性

属性标识符	属性标签	类型	参数值
FMSDU	A001	32比特整数	前向最大CPCS-SDU长度： 从主叫方发送至被叫方的最大CPCS-SDU长度 参见：ITU-T Q.2931建议书
BMSDU	A002	32比特整数	后向最大CPCS-SDU长度： 从被叫方发送至主叫方的最大CPCS-SDU长度 参见：ITU-T Q.2931建议书
SSCS	见C.7中的表	见C.7中的表	见C.7中的表 附加值： VPI/VCI

C.11 等效的SDP

等效的SDP须遵守为采用文本编码时第7.1.8节中所罗列的例外条件。例如，无论采用二进制或者文本编码，CHOOSE通配符允许由MGC发送至MG。

属性标识符	属性标签	类型	参数值
SDP_V	B001	字符串	协议版本 参见：RFC 2327
SDP_O	B002	字符串	发起方和创立方以及会话标识符 参见：RFC 2327
SDP_S	B003	字符串	会话名称 参见：RFC 2327
SDP_I	B004	字符串	会话标识符 参见：RFC 2327
SDP_U	B005	字符串	描述符的URI 参见：RFC 2327
SDC_E	B006	字符串	email地址 参见：RFC 2327
SDP_P	B007	字符串	电话号码 参见：RFC 2327
SDP_C	B008	字符串	连接信息 参见：RFC 2327
SDP_B	B009	字符串	带宽信息 参见：RFC 2327
SDP_Z	B00A	字符串	时区调整 参见：RFC 2327
SDP_K	B00B	字符串	加密密钥 参见：RFC 2327

属性标识符	属性标签	类型	参数值
SDP_A	B00C	字符串	0个或多个会话属性 参见：RFC 2327
SDP_T	B00D	字符串	活动会话时间 参见：RFC 2327
SDP_R	B00E	字符串	0次或重复次数 参见：RFC 2327
SDP_M	B00F	字符串	媒体类型、端口、传送和格式 参见：RFC 2327

C.12 H.245

属性标识符	属性标签	类型	参数值
OLC	C001	八位组串	H.245 OpenLogicalChannel结构的参数值 参见：ITU-T H.245建议书
OLCack	C002	八位组串	H.245 OpenLogicalChannelAck结构的参数值 参见：ITU-T H.245建议书
OLCcnf	C003	八位组串	H.245 OpenLogicalChannelConfirm结构的参数值 参见：ITU-T H.245建议书
OLCrej	C004	八位组串	H.245 OpenLogicalChannelReject结构的参数值 参见：ITU-T H.245建议书
CLC	C005	八位组串	H.245 CloseLogicalChannel结构的参数值 参见：ITU-T H.245建议书
CLCack	C006	八位组串	H.245 CloseLogicalChannelAck结构的参数值 参见：ITU-T H.245建议书
LCN	C007	整数	H.245逻辑通道号0-65535 参见：ITU-T H.245建议书

附 件 D

采用IP的传送

D.1 使用应用层成帧（ALF）的采用IP/UDP的传送

本协议定义的协议消息可以采用UDP传输。如果对等实体未提供相应的通信端口（参见7.2.8），则协议消息应该被发送至到默认端口：2944（文本编码）和2945（二进制编码）。消息响应必须返回至消息发起方使用的端口。

ALF是一种应用层技术，可以通过应用层来影响消息发送至对端实体的方式。一种典型的ALF是，在应用层使用了消息缓存队列时，允许应用层改变消息发送的顺序。ALF无具体的规范要求。附件D.1定义了使用ALF机制的推荐流程。

当协议在使用了应用层成帧功能（ALF）的IP/UDP上进行传送时，应该考虑传送层对于协议最大传送单元（MTU）的限制。

D.1.1 提供“最多一次”功能

当协议消息在UDP上进行传输时，可能会发生消息丢失。如果发送的消息无法获得及时响应，则可能导致命令重复发送。命令类型不同，本协议所采用的执行方式也不同。例如Add命令可能会执行好几遍，从而导致MG的状态无法预知。因此，协议处理进程必须遵循“最多一次”机制。

协议实体应当在各自的内存中保留两个列表，一个用来记录执行完最近所接收到的TransactionRequest后返回的TransactionReply，另一个用来记录当前需要处理的事务。当协议实体接收到一个TransactionRequest消息时，它将接收到消息的TransactionID与最近为具有相同MID发出的TransactionReply响应的TransactionID相比较。两者经过比较后，如果发现TransactionID相同，则接收方不应执行接收到的事务，而只是重复发送TransactionReply响应消息。如果发现TransactionID不相同，则接收方将该接收到的TransactionRequest消息与需要处理的事务列表相比较。如果在列表中查找到TransactionID相同的消息，则表明此TransactionRequest消息为重复发送，接收方不应执行此TransactionRequest消息（参见D.1.4中TransactionPending发送机制）。

协议处理机制规定了一个长定时器值（LONG-TIMER），如下所示。其中，LONG-TIMER设定的时间值应该大于一个事务的最大持续时间，该时间还应该考虑事务重复最大次数、重复定时器的最大值以及分组在网络中的最大传输时延。本建议书规定该定时器的参考建议值为30秒。

当协议实体发出响应消息后，如果LONG-TIMER超时，或者已接收到来自对等实体包含“Response Acknowledgement Parameter”参数的响应证实消息，则协议实体重复发送的TransactionReply响应消息应被丢弃。为了检测到并忽略网络中产生的各种重复的TransactionRequest，当响应消息发出后，协议实体必须为响应消息的TransactionID保存一个备份，保存时间为LONG-TIMER设定的时间。

D.1.2 事务标识符和三次握手

D.1.2.1 事务标识符

TransactionID是一个32比特的整数。MGC可以为其控制范围内的每一个MG预留一个特定的命名空间，或者为其控制范围内同一组所有的MG预留同一个命名空间。MGC可以创建多个独立进程来实现对一个大型MG的操作管理。这些进程应共享一个TransactionID命名空间。共享方式有多种，例如可以采用事务TransactionID统一分配，或者采用不同的进程预先分配互不重叠的命名空间。具体实现时，应确保每一个逻辑MGC（消息的MID相同）产生的每一个事务必须使用唯一确定的TransactionID。通过这种方式，MG只需检测TransactionID和MID就可以判断接受到的事务是否为重复发送事务。

D.1.2.2 三次握手机制

所有消息都可以携带“响应证实参数（Response Acknowledgement Parameter）”参数。该参数携带了一组“已被确认的TransactionID范围”参数。如果协议实体检测到某个重复TransactionReply的TransactionID与“已被确认的TransactionID范围”参数中所包含的某个TransactionID相同，则应删除重复的TransactionReply。如果协议实体检测到某些消息的TransactionID包含在上述TransactionID范围内，协议实体也应该丢弃这些消息。

如果MG向MGC发送最后一个响应消息后，LONG-TIMER定时器超时，或者MG重新进入服务状态，则“已确认的TransactionID范围”参数必须失效。此后，MG应该继续接受TransactionRequest消息，而不再检测接收到的TransactionID。

包含“Response Acknowledgement Parameter”参数的消息可以以任何顺序进行传送。协议实体必须为接收到的“已确认的TransactionID范围”参数保留至LONG-TIMER定时器超时。

当协议采用二进制编码方式时，如果响应确认消息中包含第一个响应（参见A.2），则只有一个事务被确认。如果响应确认消息中包含第一个响应确认和最后一个响应确认，则从第一个响应确认到最后一个响应确认的所有Transaction都被确认。采用文本方式编码时，规定使用一个破折号用来指定某个范围内的TransactionID全部被确认（参见B.2）。

D.1.3 计算重传定时器

消息请求的发送方应为所有发送的事务指定一个重传定时器，当重传定时器超时，消息请求的发送方实体仍未接收到消息响应，则应当重复发送该事务。当重复多次发送的事务仍未获得响应时，且从第一次消息发送开始至定时器超时，请求的发送方应当采用其他机制和/或拆除现有或即将建立的连接。

本建议书未对重传定时器数值进行规定。通常重传定时器的数值与特定的网络状况有关。通常，重传定时器应该通过检测发出消息与接收消息响应之间的时间间隔来估计重传定时器数值。具体实现时，发生第一次消息重传后，重传定时器计算算法必须对此后发生的每一次重传的重传定时器数值按指数递增方式进行补偿。

注一 以下是一种适用TCP/IP传输的重传定时器计算算法，该算法使用以下两个变量：

- 平均确认时延AAD（Average Acknowledgement Delay），该参数是将时延采用指数平滑平均进行估算；
- 平均偏移ADEV（Average Deviation），该参数是将时延与当前平均值之间的绝对差值采用指数平滑平均进行估算。当采用TCP传输时，重传定时器数值被设置为AAD与N倍ADEV之和。重传定时器的最大值应该确保在LONG-TIMER超时后，MG不应再接收到重复的分组。重传定时器最大致的建议参考值为4秒。

当出现一次消息重传后，协议实体应该遵循以下规则：

- 应该将AAD的估算值加倍。
- 应该以相同概率随机选取0.5AAD到AAD之间的一个随机值。
- 应该将重传定时器取值设置为上述随机值与N倍ADEV之和。

采用以上规则计算重传定时器有以下两方面的作用：由于包含了一个指数递增单元，当事务处理进程发生拥塞时，将自动降低消息流的速率。此外，由于还包含了一个随机单元，它可以避免由同一外部事件触发的通知消息之间发生同步。

D.1.4 临时响应

在事务执行过程中，某些事务执行可能需要较长的时间。从而可能会导致事务执行与基于重传定时器的重传进程发生冲突。执行时间太长可能导致事务被重传多次，或者导致重传定时器数值过大而降低传输的效率。如果协议实体已经预见到某一事务需要较长的执行时间，则它可以先发送一个TransactionPending临时响应消息来指示该事务正在被处理。当协议实体接收到重复的TransactionRequest消息，如果该事务正在处理，也应该发送TransactionPending消息。

接收到TransactionPending消息的实体必须为TransactionPending消息对应的TransactionRequest消息的重传定时器设定一个不同数值。如果发起TransactionRequest的实体在接收到Transaction Pending消息之后，接收到最终的TransactionReply消息，则必须立即向对端实体发送一个确认消息，此后，必须重新使用通常的重传定时器。对于发送了TransactionPending临时响应消息的实体，必须在发送的最终响应中包含一个“immAckRequired（需要立即确认）”字段，通过这一字段来表明该实体希望能够获得一个立即的确认消息。对于同一个TransactionRequest，实协议体在接收到最终TransactionReply之后，必须忽略接收的TransactionPending消息。

D.1.5 请求、响应和证实重复发送

一系列的事务可以完成协议的交互流程，每一个事务都由一个TransactionRequest 和一个TransactionReply组成，TransactionReply消息通常也被称为响应消息。当使用UDP传送协议消息时，消息可能发生丢失。如果发送的消息未获得及时响应，事务可能会被重复执行。协议实体应当在各自的内存中保留两个列表，一个用来记录执行最近所接收的事务后返回的事务响应，例如，在最近一次LONG-TIMER超时内所发送的所有事务响应列表，另一个用来记录当前需要处理的事务。

使用重传机制，可以避免发生以下三种差错：

- 传输差错，例如，消息分组由于线路上的噪声或队列的拥塞而发生丢失；
- 协议实体单元故障，例如，消息到达某一协议实体时接口不可用；
- 协议实体故障，例如，整个实体不可用。

协议实体应该可对由于传输差错导致的丢包率进行估算。在合理配置的系统中，分组丢失率应该很低，通常小于1%。如果一个MGC或MG必须将某一消息重传多次，导致重传的原因可能是传输差错之外的其他原因。例如，如果丢包率为1%，则重传5次失败的概率为一千亿分之一。这种情况对于处理能力为1000事务/秒的MGC来说，应该每10天最多发生一次。（实际上，这种不正常的重传次数应该是当前丢包率的函数）。通常，可疑重传次数阈值Max1应该低于切断连接的重传次数阈值，后者取值应该更高。

一种典型的重传算法是简单地对连续重传次数进行统计。当某个事务重传次数超过发生一定数目N（通常 $7 < N < 11$ ）以上时，则判定该连接已断开。考虑到可能发生的无法检测或正在进行的“failover”因素的影响，应对这种算法加以改进，以确保当MG接收到一个表明发生“failover”的有效“ServiceChange”消息时，该MG可以向新的MGC发送命令消息。而消息响应应仍被送到消息发起的源地址。

为了能对网络负荷进行自适应的调整，定时器应按指数递增。如果定时器的初始值设置为200微秒，则第五次重传的消息发生丢失将在大约6s后被检测到。如果发生“failover”情况，6s是一个可以接受的等待时延。此后，丢失的重传消息应该在定时器超时后继续重传。采取这种方式可以克服MG与MGC之间瞬时连通性问题，另一方面也可以给执行“failover”的MGC更多的时间（等待时间为30秒也是一个可以接受的时间）。

然而，重传等待时间最大值T-MAX应当是固定的。消息在任何一次重传之前，应该确定第一次消息传送与重传的时间间隔不应超过T-MAX。如果该时间超过T-MAX，则MG可以判定MGC已经发生了故障且MGC开始执行故障恢复。此时，MG必须发送带有ServiceChangeMethod参数等于“Disconnected”的ServiceChange消息，以使新的MGC获知该MG丢失了一个或多个事务。T-MAX 与LONG-TIMER 有关，LONG-TIMER的数值应为T-MAX 数值与网络最大传输时延之和。

D.2 采用TCP

本协议中规定的各种消息可以采用TCP传送。如果对端实体未提供端口（参见7.2.8），各种消息应该被发送到默认端口上。TCP是一种基于流的协议，本协议规定可使用各种消息作为消息的传送单元。根据RFC 1006，必须使用TPKT来描述TCP流中的各种消息。

对于面向事务处理的协议，TransactionRequest与TransactionReply可能发生丢失。因此，在采用TCP进行传送时，各种协议实体应为请求和响应指定应用层的定时器，定时器的计算与在UDP/ALF传送类似。

D.2.1 提供“最多一次”功能

尽管协议在TCP上传输时不会发生传输丢失，但是在实际情况中TransactionRequest或TransactionReply仍可能发生丢失。如果TransactionRequest未获得及时响应，消息就会被重复传送。命令类型不同，本协议所采用的执行方式也不同，例如Add命令可能会执行好几遍，从而导致MG的状态无法预知。

为了防止出现消息丢失，建议协议实体使用D.1.1中描述的规则。

D.2.2 事务标识符和三次握手机制

尽管TCP提供了可靠的传输机制，但是TransactionReply消息仍然可能发生丢失。所以，当协议在TCP上传输时，建议各种协议实体应执行D.1.2.2描述的规则。

D.2.3 计算重传定时器

由于TCP提供了可靠的传输机制，TransactionRequest或TransactionReply发生丢失的概率应该很低。因此，当协议采用TCP进行传输时，只需要使用简单的定时机制。因此在TCP传输中采用的定时机制不应该采用UDP/ALF传输中使用的指数补偿算法，然而，某些实体可能已经实现了指数补偿算法，例如MGC中，MGC不仅务必实现TCP还务必实现UDP/ALF。

D.2.4 临时响应

与UDP上传输一样，在TCP传输时，某些事务执行可能需要较长的时间。如果协议实体可以预见某一事务需要较长的执行时间，则它可以先发送一个TransactionPending临时响应消息指示事务正在被处理。当协议实体接收到重复的TransactionRequest消息，如果该事务正在处理，也应该发送TransactionPending消息。

接收到TransactionPending消息的实体必须为对端实体指定一个更长的定时器。

在TransactionRequest和TransactionReply被确认以前，协议实体必须保留所有的TransactionRequest和TransactionReply消息。处理程序应该遵循附件D.1.4中的相关规定，唯一不同的是此时定时器应该只选择一个简单的值。当协议实体接收到最后的响应消息后，没有必要立即发送确认消息。

D.2.5 命令执行顺序

TCP可以支持事务的有序传输。因此在TCP上进行传输时，不需要使用特殊机制来保证命令顺序传输。应该指出的是，在ALF/UDP上传输时允许发送方在发生网络拥塞时修改其操作，此时该实体可以对事务进行重新排序。然而，TCP不能实现这种功能。

附件 E

基本包

本附件对ITU-T H.248.1建议书定义的基本包进行了规定。

E.1 通用包

包的名称: 通用包

PackageID: g (0x0001)

描述: 用于常见项目的通用包

版本: 2

扩展: 无

E.1.1 属性 (Properties)

无。

E.1.2 事件 (Events)

E.1.2.1 原因 (Cause)

事件名称: 原因

EventID: cause (0x0001)

描述: 通用差错事件

EventsDescriptor参数: 无

ObservedEvents描述符参数:

通用原因

参数名称: 通用原因

ParameterID: Generalcause (0x0001)

描述: 该参数将MGC可能发生的差错类型分为6类。

类型: 枚举

可选项: 否

可能的取值:	"NR"	(0x0001)	正常释放
	"UR"	(0x0002)	无可用资源
	"FT"	(0x0003)	临时故障
	"FP"	(0x0004)	永久故障
	"IW"	(0x0005)	互通差错
	"UN"	(0x0006)	不支持

默认值: 无

故障原因

参数名称: 故障原因

ParameterID: Failurecause (0x0002)

描述: 释放值由被释放设备产生, 例如, 网络连接释放。释放原因值由相关的承载控制协议定义。

类型: OCTET STRING
可选项: 是
可能的取值: OCTET STRING
默认值: 无

E.1.2.2 信号结束 (Signal Completion)

事件名称: 信号结束

EventID: sc (0x0002)

描述: 当notifycompletion参数被设置为允许上报事件结束时, 指示一个信号的终结。进一步描述, 参见7.1.1、7.1.11 和 7.2.7。

EventsDescriptor参数: 无

ObservedEvents描述符参数:

信号标识

参数名称: 信号标识

ParameterID: SigID (0x0001)

描述: 该参数用于指示被终结的信号标识符。当信号包含信号列表时, 应返回信号列表标识符参数。

类型: 二进制: 八位组 (串), 文本: 字串

可选项: 否

可能的取值: 被终结的信号标识。信号标识必须通过包名来识别, 且不允许使用通配符。

默认值: 无

结束方式

参数名称: 结束方式

ParameterID: Meth (0x0002)

描述: 该参数用于描述信号被终结的方法。

类型: 枚举

可选项: 否

可能的取值:

"TO" (0x0001) 信号超时或自行结束

"EV" (0x0002) 事件导致中断

"SD" (0x0003) 新的信号描述符导致中断

"NC" (0x0004) 未完成, 他原因

"PI" (0x0005) 第一至倒数第二次迭带。最后一次迭带使用TO。

默认值: 无

信号列表ID

参数名称: 信号列表ID

ParameterID: SLID (0x0003)

描述: 该参数用于指示信号属于某一个信号列表。当且仅当一个信号处于信号列表中, 该参数才能被返回。

类型: 整数

可选项: 是 (仅仅当使用了信号列表时, 才支持该参数)

可能的取值: 1-65535

默认值: 无

请求ID

参数名称: 请求ID

ParameterID: RID (0x0004)

描述: 用于指示与SignalID 相关联的NotifyCompletion 请求。

类型: 整数

可选项: 是 (仅仅当使用了信号列表时, 才支持该参数)

可能的取值: 1-4294967295

默认值: 无

E.1.3 信号

无。

E.1.4 统计值 (Statistics)

无。

E.2 基根包

包的名称: 基根包

PackageID: root (0x0002)

描述: 该包定义网关通用属性。

版本: 2

扩展: 无

E.2.1 属性

E.2.1.1 最大关联数

属性名称: MaxNrOfContexts

PropertyID: maxNumberOfContexts (0x0001)

描述: 该参数值用于指示网关中允许存在的最大关联数。且该参数值不包含空 (NULL) 关联。

类型: 双精度

可能的取值: 大于等于1

默认值: 预设

定义于： TerminationState

属性： 只读

E.2.1.2 每关联的最大终结点数目

属性名称： MaxTerminationsPerContext

PropertyID: maxTerminationsPerContext (0x0002)

描述: 该参数值用于定义一个关联中允许的最大终结点数目，见6.1。

类型: 整数

可能的取值: 任何正整数

默认值: 预设

定义于： TerminationState

属性： 只读

E.2.1.3 正常MG执行时间

属性名称： normalMGExecutionTime

PropertyID: normalMGExecutionTime (0x0003)

描述: 该参数由MGC设置，用于指示MGC发送事务请求后，MG应该返回事务响应的时延（时延包括网络时延）。

类型: 整数

可能的取值: 任何正整数；单位为毫秒

默认值: 预设

定义于： TerminationState

属性： 读/写

E.2.1.4 正常MGC执行时间

属性名称： normalMGCExecutionTime

PropertyID: normalMGCExecutionTime (0x0004)

描述: 该参数由MGC设置，用于指示MG发送事务请求后，MGC应该返回事务响应的时延（时延包括网络时延）。

类型: 整数

可能的取值: 任何正整数；单位为毫秒

默认值: 预设

定义于： TerminationState

属性： 读/写

E.2.1.5 MG临时响应定时器值

属性名称： MGProvisionalResponseTimerValue

PropertyID: MGProvisionalResponseTimerValue (0x0005)

描述: 该参数用于指示当MG不能完成事务处理时，MGC应该接收到事务临时响应的时延。起始值可以设置为normalMGExecutionTime与网络时延之和或少于其总和。

类型： 整数

可能的取值： 任何正整数；单位为毫秒

默认值： 预设

定义于： TerminationState

属性： 读/写

E.2.1.6 MGC临时响应定时器值

属性名称： MGCTemporaryResponseTimerValue

PropertyID： MGCTemporaryResponseTimerValue (0x0006)

描述： 该参数用于指示当MGC不能完成事务处理时，MG应该接收到事务临时响应的时延。起始值可以设置为normalMGCExecutionTime与网络时延之和或少于其总和

类型： 整数

可能的取值： 任何正整数；单位为毫秒

默认值： 预设

定义于： TerminationState

属性： 读/写

E.2.1.7 MGC发送临时响应门限值

属性名称： MGCOrientedPendingLimit

PropertyID： MGCOrientedPendingLimit (0x0007)

描述： 该参数用于指示MG可以从MGC接收到的TransactionPending响应的数目。一旦TransactionPending响应的数目超过该参数值，应该返回TransactionReply且包含差错代码506（TransactionPending数目超出），否则MG将认为事务处理发生差错。

类型： 整数

可能的取值： 任何正整数

默认值： 预设

定义于： TerminationState

属性： 读/写

E.2.1.8 MG发送临时响应门限值

属性名称： MGOrientedPendingLimit

PropertyID： MGOrientedPendingLimit (0x0008)

描述： 该参数用于指示可以MGC从MG接收到的TransactionPending响应的数目。一旦TransactionPending响应的数目超过该参数值，应该返回TransactionReply且包含差错代码506（TransactionPending数目超出），否则MGC将认为事务处理发生差错。

类型： 整数

可能的取值： 任何正整数

默认值： 预设

定义于： TerminationState

属性： 读/写

E.2.2 事件

无。

E.2.3 信号

无。

E.2.4 统计值

无。

E.2.5 程序

无。

E.3 信号音发生器包

包的名称： 信号音发生器包

PackageID: tonegen (0x0003)

描述： 该包用于定义产生信号音的信号。该包不能指定参数值。该包可以被扩展。信号音通常规定为单个信号，携带参数“ind”指示“interdigit”时延，信号音标识符用于指示待播放的信号音。对于同一个信号音，信号音标识符应该在任何信号音生成包中保持一致。相应信号音的特性应遵循MG所处国家的相关要求。

版本： 2

是否仅用于扩展： 是

扩展： 无

E.3.1 属性

无。

E.3.2 事件

无。

E.3.3 信号

E.3.3.1 播放信号音

信号名称： 播放信号音

SignalID: pt (0x0001)

描述： 用于指示在音频通道上播放信号音

Signal类型： 简洁

时长： 预设

附加参数：

信号音标识列表

参数名称： 信号音标识列表

ParameterID: tl (0x0001)

描述： 要按序播放的信号音列表

类型： 枚举子列表

可选项： 否

可能的取值：子列表必须包含一个或多个信号音标识。

默认值：无

信号间隔时长

参数名称：信号间隔时长

ParameterID： ind (0x0002)

描述：两个连续信号音之间的超时，单位为毫秒

类型：整数

可选项：是

可能的取值：任何大于或等于0的整数

默认值：预设

信号音方向

参数名称：信号音方向

ParameterID： btd (0x0003)

描述：应向终结点方向播放信号音。

类型：枚举

可选项：是

可能的取值：
"EXT" (0x0001) 外部
"INT" (0x0002) 内部
"BOTH" (0x0003) 外部和内部

默认值：外部

E.3.4 统计值

无。

E.3.5 程序

信号音生成包未定义信号音标识符。基于该包的扩展包中可以添加信号音标识符和信号音信号。

E.4 信号音检测包

包的名称：信号音检测包

PackageID： tonedet (0x0004)

描述：信号音检测包用于定义检测信号音事件。检测的信号音用信号标识符来进行指示。MG播放的信号音应遵循MG所处国家的相关规定。

信号音检测包未指定特定参数值。此包可以被扩展。

版本：1

是否仅用于扩展：是

扩展：无

E.4.1 属性

无。

E.4.2 事件

E.4.2.1 信号音开始检测

事件名称： 信号音开始检测

EventID： std (0x0001)

描述： 该事件参数用于指示检测起始信号音。信号音被检测的特性也具体实现有关。

EventsDescriptor参数：

信号音标识列表

参数名称： 信号音标识列表

ParameterID： tl (0x0001)

类型： 枚举子列表

可选项： 否

可能的取值： 该包所定义的唯一一个信号标识符为“wild card”，文本编码为“*”，二进制编码为0x0000。扩展包中可以增加信号音标识符。如果tl等于“wild card”，指示检测所有信号音。

默认值： 无

ObservedEventsDescriptor参数：

信号音ID

参数名称： 信号音ID

ParameterID： tid (0x0003)

描述： 被检测的信号音标识符。

类型： 枚举

可选项： 否

可能的取值： 此包所定义的唯一一个信号标识符为“wild card”。扩展包中可以增加信号标识符。

默认值： 无

E.4.2.2 信号音终止检测

事件名称： 信号音终止检测

EventID： etd (0x0002)

描述： 检测某个信号音的结束。

EventsDescriptor参数：

信号音标识列表

参数名称： 信号音标识列表

ParameterID： tl (0x0001)

描述： 被检测的信号音标识符列表。

类型： 枚举子列表

可选项：否

可能的取值：此包未定义允许值。扩展包允许增加信号音标识符的允许值。

默认值：无

ObservedEventsDescriptor参数:

信号音标识

参数名称：信号音标识

ParameterID: tid (0x0003)

描述：被检测的信号音标识符。

类型：枚举

可选项：否

可能的取值：此包所定义的唯一一个信号标识符为“wild card”。扩展包中可以增加信号标识符。

默认值：无

时长

参数名称：时长

ParameterID: dur (0x0002)

描述：该参数用于指示从检测开始到检测完成的之间的信号音播放的时间间隔。

类型：整数

可选项：是

可能的取值：任何正整数；单位为毫秒

默认值：无

E.4.2.3 长信号音检测

事件名称：长信号音检测

EventID: ltd (0x0003)

描述：该参数用于指示检测长信号音。

EventsDescriptor参数:

信号音标识列表

参数名称：信号音标识列表

ParameterID: tl (0x0001)

描述：被检测的信号音标识符列表。

类型：枚举子列表

可选项：否

可能的取值：此包未定义允许值扩展包允许增加信号音标识符的允许值。

默认值：无

时长

参数名称： 时长

ParameterID: dur (0x0002)

描述： 测试所需的时长。

类型： 整数

可选项： 是

可能的取值： 任何正整数；单位为毫秒

默认值： 预设

ObservedEventsDescriptor参数：

信号音标识

参数名称： 信号音标识

ParameterID: tid (0x0003)

描述： 被检测的信号音标识符。

类型： 枚举

可选项： 否

可能的取值： 该包未定义允许值。扩展包允许增加信号音标识符的允许值。

默认值： 无

E.4.3 信号

无。

E.4.4 统计值

无。

E.4.5 程序

无。

E.5 基本DTMF发生器包

包的名称： 基本DTMF发生器包

PackageID: dg (0x0005)

描述： 该包用于定义基本DTMF信号音，DTMF信号音是对tonegen包中的播放信号音信号中“tl”参数的扩展。

版本： 2

扩展： tonegen版本2

E.5.1 属性

无。

E.5.2 事件

无。

E.5.3 信号

E.5.3.1 DTMF字符0

信号名称: DTMF字符0

SignalID: d0 (0x0010)

描述: 产生DTMF “0” 信号音。DTMF “0” 信号音的物理属性由网关指定。

Signal 类型: 简洁

时长: 预设

附加参数:

信号音方向

参数名称: 信号音方向

ParameterID: btd (0x0001)

描述: 应向终结点方向播放信号音。

类型: 枚举

可选项: 是

可能的取值: "EXT" (0x0001) 外部
 "INT" (0x0002) 内部
 "BOTH" (0x0003) 外部和内部

默认值: 外部

附加值: d0 (0x0010)为适用于playtone的一个信号音标识。

其他DTMF字符信号定义与DTMF “0” 信号定义相同，如下表所示。每一个DTMF字符是在基本信号音生成包基础上扩展而来，由信号和信号音标识组成。DTMF信号音标识与数字映射（DitiMap）中使用的名称不同。

信号名称	SignalID/信号音 ID
DTMF 字符 0	d0 (0x0010)
DTMF 字符 1	d1 (0x0011)
DTMF 字符 2	d2 (0x0012)
DTMF 字符 3	d3 (0x0013)
DTMF 字符 4	d4 (0x0014)
DTMF 字符 5	d5 (0x0015)
DTMF 字符 6	d6 (0x0016)
DTMF 字符 7	d7 (0x0017)
DTMF 字符 8	d8 (0x0018)
DTMF 字符 9	d9 (0x0019)
DTMF 字符*	ds (0x0020)

信号名称	SignalID/信号音 ID
DTMF 字符#	do (0x0021)
DTMF 字符 A	da (0x001a)
DTMF 字符 B	db (0x001b)
DTMF 字符 C	dc (0x001c)
DTMF 字符 D	dd (0x001d)

E.5.4 统计值

无。

E.5.5 程序

无。

E.6 DTMF检测包

包的名称：DTMF检测包

PackageID: dd (0x0006)

描述: 该包规定检测基本的DTMF信号音。该包扩展了“信号音开始检测”、“信号音终止检测”和“长信号音检测”中“信号音ID”的可能值。

附加信号音ID值是由包dg所定义的所有信号音ID。

下表是7.1.14中定义的DTMF事件与DigitMap符号之间的映射关系表。

DTMF	事件符号
d0	"0"
d1	"1"
d2	"2"
d3	"3"
d4	"4"
d5	"5"
d6	"6"
d7	"7"
d8	"8"
d9	"9"
da	"A"或"a"
db	"B"或"b"
dc	"C"或"c"
dd	"D"或"d"
ds	"E"或"e"
do	"F"或"f"

版本： 1

扩展： tonedet版本1

E.6.1 属性

无。

E.6.2 事件

E.6.2.1 DTMF数字

事件名称: DTMF数字

EventID: 事件标识符与E.5.3中的表格中所定义的信号标识符相同。

描述: 在MG检测到一位数字时产生。

EventDescriptor参数: 无

ObservedEventsDescriptor参数: 无

E.6.2.2 DigitMap结束事件

事件名称: DigitMap结束事件

EventID: ce (0x0004)

描述: 在DigitMap如7.1.14所述完成时产生

EventsDescriptor参数: 无

ObservedEventsDescriptor参数:

DigitString

参数名称: DigitString

ParameterID: ds (0x0001)

描述: 如7.1.14所定义, 当前拨号字串与DigitMap所定义匹配规定发生部分或完全匹配。

类型: DigitMap符号(可能为空)串, 作为quotedString返回

可选项: 否

可能的取值: 字符“0”到“9”、“A”到“F”和长时指示符“Z”的字串组合。

默认值: 无

结束方法

参数名称: 结束方法

ParameterID: Meth (0x0003)

描述: 该参数用于指示Digitmap结束的原因。参见7.1.14中的程序。

类型: 枚举

可选项: 否

可能的取值:

"UM" (0x0001)	明确匹配
"PM" (0x0002)	由于定时器超时或未匹配事件发生, 导致部分匹配
"FM" (0x0003)	由于定时器超时或未匹配事件发生, 导致完全匹配

默认值: 无

E.6.3 信号

无。

E.6.4 统计值

无。

E.6.5 程序

仅仅当Event描述符中包含一个“DigitMap Completion”事件参数时（“DigitMap Completion”参数定义参见E.6.2），并且用于被请求的动作中的“DigitMap Completion”参数包含EventDM参数项（EventDM参数参见7.1.9），DigitMap的处理程序才会被激活。如果Event描述符中包含其他的参数（如KeepActive或内嵌信号描述符），则这些参数不会影响DigitMap处理程序的激活。

E.7 呼叫进展信号音发生器包

包的名称：呼叫进展信号音发生器包

PackageID：cg 0x0007

描述：该包是对信号音发生器包(tonegen)中playtone所包含的“t1”参数允许值进行扩展的扩展包。

版本：2

扩展：tonegen版本2

E.7.1 属性

无。

E.7.2 事件

无。

E.7.3 信号

E.7.3.1 拨号音

信号名称：拨号音

SignalID：dt (0x0030)

描述：生成拨号音。拨号音的物理特性由网关决定。

Signal 类型：TimeOut

时长：预设

附加参数：无

附加值：规定dt (0x0030)为playtone的一个信号音ID。

其他信号音定义采用与dt (0x0030)定义相同的方式。下表列举了“呼叫进展信号音发生器包”所定义的所有的信号名称和名号标识符。注意，每一个信号音是在基本信号音发生器包基础上扩展而来，由信号和信号音标识组成。

信号名称	SignalID/信号音 ID
拨号音	dt (0x0030)
振铃音	rt (0x0031)
忙音	bt (0x0032)
拥塞信号音	ct (0x0033)
特殊信息信号音	sit (0x0034)
(录音) 警告信号音	wt (0x0035)
付费电话识别信号音	prrt (0x0036)
呼叫等待信号音	cw (0x0037)
主叫等待信号音	cr (0x0038)

E.7.4 统计值

无。

E.7.5 程序

注一 各种信号音标示符的定义参见ITU-T E.180/Q.35建议书。信号音的含义参见ITU-T E.180/Q.35建议书。

E.8 呼叫进展信号音检测包

包的名称： 呼叫进展信号音检测包

PackageID: cd (0x0008)

描述： 该包用于对呼叫接续过程的信号音进行检测。该包是在“信号音开始检测”、“信号音终止检测”和“长信号音检测”中事件参数的允许值基础上进行扩展。

附加值

为“信号音开始检测”、“信号音终止检测”和“长信号音检测”定义的信号音标识与“呼叫进展信号音发生器包”中定义的信号音标识相同。

所需的信号音标识集合应符合ITU-T E.180/Q.35建议书。有关这些信号音的含义参见ITU-T E.180/Q.35建议书。

版本： 1

扩展： tonedet版本1

E.8.1 属性

无。

E.8.2 事件

呼叫进展信号音检测包中定义的事件与呼叫进展信号音发生器包中定义的信号标识符相同。具体参见E.7.3。

E.8.3 信号

无。

E.8.4 统计值

无。

E.8.5 程序

无。

E.9 模拟线路监测包

包的名称：模拟线路监测包

PackageID：al, 0x0009

描述：该包定义了适用于模拟线路的信号和各种事件检测机制。

版本：1

扩展：无

E.9.1 属性

无。

E.9.2 事件

E.9.2.1 挂机

事件名称：挂机

EventID：on (0x0004)

描述：检测挂机事件。当MG被要求检测某条模拟线的挂机事件时，且该线路已处于挂机状态，则MG必须根据“strict”参数进行操作。

EventsDescriptor参数：

严格转移

参数名称：严格转移

ParameterID：strict (0x0001)

描述：指示如何检测挂机事件。

类型：枚举

可选项：是

可能的取值："exact" (0x0000)：表示只有模拟线状态从实际叉簧状态转移至挂机状态时，该事件才发生。

"state" (0x0001)，表示模拟线状态从实际叉簧状态转移至挂机状态事件或者模拟线已经处于挂机状态，该事件发生。

默认值：Exact

ObservedEventsDescriptor参数：

初始状态

参数名称：初始状态

ParameterID：init (0x0002)

描述：挂机事件发生改变的原因。只有当“Strict Transition”参数设置为“state”是，该参数值才能被返回。

类型：布尔型

可能的取值：“True”（真）：表示当包括该事件的事件描述符被激活，模拟线已处于挂机状态，导致事件上报。

“False”（假）：表示模拟线从实际状态转移至挂机状态，导致事件上报。

默认值：无

E.9.2.2 摘机

事件名称：摘机

EventID： of (0x0005)

描述：检测摘机事件。当MG被要求检测某条模拟线的摘机事件时，且该线路已处于摘机状态，则MG必须根据“strict”参数进行操作。

EventsDescriptor参数：

严格转移

参数名称：严格转移

ParameterID： strict (0x0001)

描述：指示如何检测摘机事件。

类型：枚举

可能的取值：“exact” (0x0000)：表示只有模拟线状态从实际叉簧状态转移至摘机状态时，该事件才发生。

“state” (0x0001)，表示模拟线状态从实际叉簧状态转移至摘机状态事件或者模拟线已经处于摘机状态，该事件发生。

“failWrong” (0x0002)，表示模拟线已经处于摘机状态，当接收到该检测事件时命令失败，MG应上报差错响应。

默认值：Exact

ObservedEventsDescriptor参数：

初始状态

参数名称：初始状态

ParameterID： init (0x0002)

描述：摘机事件发生改变的原因。只有当“Strict Transition”参数设置为“state”是，该参数值才能被返回。

类型：布尔型

可选项：是

可能的取值：“True”（真）：表示当包括该事件的事件描述符被激活，模拟线已处于摘机状态，导致事件上报。

“False”（假）：表示模拟线从实际状态转移至摘机状态，导致事件上报。

默认值：无

E.9.2.3 闪断叉簧

事件名称：闪断叉簧

EventID： fl (0x0006)

描述：检测模拟线闪断事件。一个闪断事件是指挂机后接着摘机，且摘机的时长应处于最小时长到最大时长之间的时间间隔内。

EventsDescriptor参数:

最小时长

参数名称: 最小时长

ParameterID: mindur (0x0004)

描述: 挂机和摘机之间所发生的闪断事件的最小时间间隔。

类型: 整数

可选项: 是

可能的取值: 任何正整数; 单位为毫秒

默认值: 预设.

最大时长

参数名称: 最大时长

ParameterID: maxdur (0x0005)

描述: 挂机和摘机之间所发生的闪断事件的最大时间间隔。

类型: 整数

可选项: 是

可能的取值: 任何正整数; 单位为毫秒

默认值: 预设

ObservedEventsDescriptor参数: 无

E.9.3 信号

E.9.3.1 振铃

信号名称: 振铃

SignalID: ri (0x0002)

描述: 指示在线路上播放振铃音。

Signal 类型: TimeOut

时长: 预设

附加参数:

Cadence

参数名称: Cadence

ParameterID: cad (0x0006)

描述: 标识信号音on和off状态之间的时长, 信号音on和off可以构成一个完整振铃周期, 且周期的开始为信号音on状态。如果MG无法支持该功能, 可以忽略该参数。

类型: 整数的子列表

可选项: 是

可能的取值: 任何正整数; 单位为毫秒

默认值: 预设

频率

参数名称: 频率

ParameterID: freq (0x0007)

描述: 振铃信号音的频率。如果MG无法支持该功能，可以忽略该参数。

类型: 整数

可选项: 是

可能的取值: 任何正整数；单位为Hz

默认值: 预设

E.9.4 统计值

无。

E.9.5 差错代码

差错代码#: 540

名称: 意外的初始叉簧状态

定义: 当MGC尝试请求叉簧的转移状态时且“Strict”参数为“failWrong”，如果叉簧状态已经处于隐含的转移状态时，返回该差错代码。

差错描述符中的差错文本: -

注释: -

E.9.6 程序

如果MGC将包含在事件描述符的叉簧状态转移事件（如挂机或摘机）中所包含的“Strict”参数值设置为“failWrong”，叉簧已经处于其隐含的状态，则包含该事件描述符的命令执行失败。MGC必须返回一个差错响应，差错代码为540（意外的叉簧初始状态）。

E.10 基本连续性包

包的名称: 基本连续性包

PackageID: ct (0x000a)

描述: 该包用于定义进行通道连续性检测的事件和信号。连续性测试包括环回或收发器功能的测试。

版本: 1

扩展: 无

E.10.1 属性

无。

E.10.2 事件

E.10.2.1 连续性检测结束

事件名称: 完成

EventID: cmp (0x0005)

描述: 该事件用于检测连续性测试的结束。

EventsDescriptor参数: 无

ObservedEventsDescriptor参数:

连续性检测结果

参数名称: Result

ParameterID: res (0x0008)

描述: 该参数指示连续性检测的结果。

类型: 枚举

可选项: 否

可能的取值: "SUCCESS" (0x0001) 成功
"FAILURE" (0x0000) 失败

默认值: 无

E.10.3 信号

E.10.3.1 连续性检测

信号名称: 连续性检测

SignalID: ct (0x0003)

描述: MGC指示MG在终结点上开始发送连续性检测信号音。

Signal 类型: TimeOut

默认值: 预设

附加参数: 无

E.10.3.2 连续性检测响应

信号名称: 响应

SignalID: rsp (0x0004)

描述: 该信号用于对连续性检测发送响应。参见E.10.5。

Signal 类型: 开/关

默认值: 预设

附加参数: 无

E.10.4 统计值

无。

E.10.5 程序

当MGC期望对MG发起连续性检测时，MGC应向MG发送包含以下参数的命令：

- 包含连续性检测（ct）信号的信号描述符；
- 包含完成事件（cmp）的事件描述符。

当MG接收到包含以上参数的命令时，MG开始向命令所指定的终结点发送连续性检测信号音。如果在信号超时之前，MG可以检测到返回的信号音，且返回的信号音满足任何所要求的其他条件，则MG必须返回一个cmp参数值等于“Success”的完成事件。否则，必须返回一个cmp参数值等于“Failure”的完成事件。

当MGC期望MG产生连续性检测响应时，MGC必须向MG发送的命令中包含响应信号（rsp）的信号描述符。如果MG接收到包含此类信号的命令时，MG可以按发起环回检测或等待连续检测信号音两种情况进行处理。对于环回检测来说，MG应将接收到的任何信息进行环回。对于等待连续检测信号音，MG接收到任何正确的检测信号音后应该向MGC返回响应信号音。MGC负责何时终止rsp信号。

当期望对终结点进行连续性检测时，必须不允许终结点存在激活的回声设备和回波编码器。

语音通路检测作为连续性检测的一部分，其执行依赖于不同网络运营商之间制定的双边协议。

注一 信号音和连续性检测程序参见第7节和第8节/Q.724、2.1.8/Q.726，以及ITU-T Q.1902.4建议书。

E.11 网络包

包的名称：网络包

PackageID：nt (0x000b)

描述：该包定义了与网络承载类型无关的网络终结点的属性，网络承载类型不仅仅局限于TDM、IP或ATM网络。

版本：1

扩展：无

E.11.1 属性

E.11.1.1 最大抖动缓存

属性名称：最大抖动缓存

PropertyID：jit (0x0007)

描述：该属性用于设置MG中抖动缓存的最大值。

类型：整数

可能的取值：任何正整数；单位为毫秒

默认值：预设

定义于：LocalControl描述符

属性：读/写

E.11.2 事件

E.11.2.1 网络故障

事件名称：网络故障

EventID：netfail (0x0005)

描述：终结点在检测到由于内部或外部的网络原因导致的故障时报告该事件。

EventsDescriptor参数：无

ObservedEventsDescriptor参数：

原因

参数名称：原因

ParameterID：cs (0x0001)

描述：该参数可以与故障事件一起为故障原因提供诊断信息。

类型：String

可选项：是

可能的取值：任何文本字符串

默认值：无

E.11.2.2 质量告警

事件名称：质量告警

EventID： qualert (0x0006)

描述： 该事件参数允许MG报告网络连接的服务质量(QoS)发生降低。MG可以通过测量丢包率、到达间隔抖动、传播时延并指出QoS降低的百分比。

EventsDescriptor参数：

门限

参数名称： 门限

ParameterID： th (0x0001)

描述： 测量的服务质量降低百分比的门限基于一个规定的方法计算，并考虑例如丢包率、抖动和时延。当计算结果超过门限参数值时，该事件被触发。

类型： 整数

可选项：是

可能的取值：0至99

默认值：预设

ObservedEventsDescriptor参数：

门限

参数名称： 门限

ParameterID： th (0x0001)

描述： 测量的服务质量降低百分比的门限基于一个规定的方法计算，并考虑例如丢包率、抖动和时延。

类型： 整数

可选项：是

可能的取值：0至99

默认值：无

E.11.3 信号

无。

E.11.4 统计值

E.11.4.1 时长

统计值名称: 时长

StatisticsID: dur (0x0001)

描述: 该参数对终结点处于一个关联中或离开空关联的时间进行统计。

类型: 双精度

可能的取值: 任何正整数; 单位为毫秒

级别: 任意

E.11.4.2 发送字节数

统计值名称: 发送字节数

StatisticID: os (0x0002)

描述: 该参数提供了自从终端已经存在或离开NULL关联以来由终端或流所发送的字节数目。该字节数表示除掉所有传输开销以外的输出媒体流。在终端层面来看, 该字节数等于所有流上输出字节的总和。

对于模拟传输媒体, 字节计数应设置等于0。

类型: 双精度

可能的取值: 大于等于0的任何64比特整数

级别: 任意

E.11.4.3 接收字节数

统计值名称: 接收字节数

StatisticID: or (0x0003)

类型: 双精度

描述: 该参数提供了自从终端已经存在或离开NULL关联以来由终端或流所接收到的字节数。该字节数表示除掉所有传输开销以外的输入媒体流。在终端层面来看, 该字节数等于所有流上输入字节的总和。

对于模拟传输媒体, 字节计数应设置等于0。

可能的取值: 大于等于0的任何64比特整数

级别: 任意

E.11.5 程序

E.11.5.1 用于网络故障和质量告警事件的应用程序

由网络包所定义的两类事件的应用程序依靠:

- 特定的承载类型 (例如, TDM、模拟线、RTP、IP、AAL 2、AAL 1等);
- 底层传输层技术 (例如, 模拟线、PDH、SDH、SONET、xDSL、802.3、802.11等);
- 事件是否与MG的外部或内部资源相关联。

E.11.5.1.1 两类事件的共同点

任意一个事件的应用程序需要对MGC和MG之间的事情进行了解。如果MG和MGC对其间的事件缺乏一致的看法，则MGC将不能对这些事件进行有效的处理。

对此给出明确的建议已经超出了H.248.1核心协议的范围。这些事件的特殊用法应该在H.248 Profile规范中指定。

E.11.5.1.2 “网络故障”事件的特殊性

可以使用Cause（原因）参数来区别不同的故障模式。终端类型也有助于解释故障事件。

E.11.5.1.3 “质量告警”事件的特殊性

检测逻辑应该基于一个或多个与业务质量相关的性能度量，从中可获得与门限参数值相比拟的数值。

底层测量和评估过程（与QoS相关的观测变量、过滤器机制、更新周期长度等）的细节可能在某个H.248 Profile中指定。

E.11.5.1.4 与业务改变程序的关系

网络故障或质量告警事件可能触发所影响的终端上的业务改变程序。向OutOfService（脱离业务）状态的转移可以由MG或MGC来发起，这依据具体环境而定。

E.12 RTP包

包的名称： RTP包

PackageID： rtp (0x000c)

描述： 该包用于通过实时传送协议（RTP）支持基于分组的多媒体数据传输（RFC 3550）。

版本： 1

扩展： nt版本1

E.12.1 属性

无。

E.12.2 事件

E.12.2.1 有效载荷切换

事件名称： 有效载荷切换

EventID： pltrans (0x0001)

描述： 该事件用于检测RTP有效载荷发生切换。

EventsDescriptor参数： 无

ObservedEventsDescriptor参数：

RTP有效载荷类型

参数名称： rtppayload

ParameterID： rtppltype (0x01)

描述： 导致发生变化的有效载荷格式。

类型： 枚举子列表

可选项： 否

可能的取值：编码方式必须由一个或多个合法的编码名来指定，合法的编码名在已经在IANA注册的RTP AV约定中进行规定。

默认值：无

E.12.3 信号

无。

E.12.4 统计值

E.12.4.1 发包数

统计值名称：发包数

StatisticID： ps (0x0004)

描述：用于对自从开始存在或脱离空关联的终结点或流所发送的包数目进行统计。

类型： 双精度

可能的取值： 大于等于0的任何64比特整数

级别： 任意

E.12.4.2 收包数

统计值名称：收包数

StatisticID： pr (0x0005)

描述：用于对自从开始存在或脱离空关联的终结点或流所接收的包数目进行统计。

类型： 双精度

可能的取值： 大于等于0的任何64比特整数

级别： 任意

E.12.4.3 包丢失率

统计值名称：包丢失率

StatisticID： pl (0x0006)

描述：用于对当前RTP流中的丢包率进行统计，参见RFC 3550。丢包率以百分比来表示：在两次接收上报的时间间隔内所丢包的数目，除以这段时间内应接收到的包数。

类型： 双精度

可能的取值： 整数部分32位，小数部分32位

级别： 任意

E.12.4.4 Jitter

统计值名称： Jitter

StatisticID： jit (0x0007)

描述：用于对当前RTP流的到达间隔抖动进行统计，参见 RFC 3550。抖动参数可以衡量RTP数据包的到达间隔时间的变化。

类型： 双精度

可能的取值： 大于等于0的任何64比特整数

级别： 任意

E.12.4.5 时延

统计值名称: 时延

StatisticID: delay (0x0008)

描述: 时延计算应以RTP包中的时戳来进行计算。时延应按照平均时延计算方法进行计算。

类型: 双精度

可能的取值: 大于等于0的任何64比特整数

级别: 任意

E.12.5 程序

当RTCP与某个RTP流相关联时，RTCP应不受LocalControlDescriptor（本地控制描述符）中的H.248.1 Mode Property（模式属性）的影响。

当RTCP与某个RTP流相关联且MG从这个流接收到一个Empty Remote Descriptor（空远程描述符）时，MG应终止与相应RTP流相连的RTCP流。

E.13 TDM电路包

包的名称: TDM电路包

PackageID: tdmc (0x000d)

描述: TDM电路包可适用于支持增益和回声控制的任何类型的终结点。尽管该包通常只用于 TDM 终结点，然而其适用范围可以更广。

对该TDM电路包的扩展应该考虑非TDM类型的终结点。

版本: 1

扩展: 网络包版本1

E.13.1 属性

E.13.1.1 回声抑制

属性名称: 回声抑制

PropertyID: ec (0x0008)

类型: 布尔型

可能的取值: “True”（真）(当启动回声抑制时)
“False”（假）(当关闭回声抑制时)

默认值: 预设

定义于: LocalControl描述符

属性: 读/写

E.13.1.2 增益控制

属性名称: 增益控制

PropertyID: gain (0x000a)

描述: 增益控制，或者使用信号电平调节和信号电平降低来调节信号电平。然而，当进行Modem呼叫时，需要关闭此功能。当该参数值设置为“automatic”时，终结点采用自动增益调节，目标增益调节电平由MG设置，且调节方向朝着外部。

类型: 整数

可能的取值: 增益控制参数规定了增益，单位为分贝，最大值为214748647 (0x7fffffff)。214748647 (0x7fffffff) 预留给“automatic”。

默认值: 预设

定义于: LocalControl描述符

属性: 读/写

E.13.2 事件

无。

E.13.3 信号

无。

E.13.4 统计值

无。

E.13.5 程序

无。

E.14 分段包

包的名称: 分段包

PackageID: seg (0x00A3)

描述: 该包定义了属性用于在非分段传输时实现基于H.248的分段。

版本: 1

扩展: 根版本2

E.14.1 属性

E.14.1.1 MG分段定时器值

属性名称: MGSegmentationTimerValue

PropertyID: MGSegmentationTimerValue (0x0009)

描述: 指示一旦MGC接收到SegmentationCompleteToken后直至MGC期望接收到从MG发出的任何消息分段之间的时间。最初设置为MGProvisionalResponseTimerValue，但可能会有改变。

类型: 整数

可能的取值: 任何正整数；单位为毫秒

默认值: 预设

定义于: TerminationState

属性: 读/写

E.14.1.2 MGC分段定时器值

属性名称: MGCSegmentationTimerValue

PropertyID: MGCSegmentationTimerValue (0x000A)

描述: 指示一旦MG接收到SegmentationCompleteToken后直至MG期望接收到从MGC发出的任何消息分段之间的时间。最初设置为MGProvisionalResponseTimerValue, 但可能会有改变。

类型: 整数

可能的取值: 任何正整数; 单位为毫秒

默认值: 预设

定义于: TerminationState

属性: 读/写

E.14.1.3 MG最大PDU长度

属性名称: MGMaxPDUSize

PropertyID: MGMaxPDUSize (0x000B)

描述: 该值指示MG侧用于控制相关传送协议的输入协议数据单元的最大值。MGC应该避免构建超过该值大小的消息。

类型: 整数

可能的取值: 任何正整数; 单位为字节

默认值: 预设

定义于: TerminationState

属性: 只读

E.14.1.4 MGC最大PDU长度

属性名称: MGCMMaxPDUSize

PropertyID: MGCMMaxPDUSize (0x000C)

描述: 该值指示MGC侧用于控制相关传送协议的输入协议数据单元的最大值。MG应该避免构建超过该值大小的消息。

类型: 整数

可能的取值: 任何正整数; 单位为字节

默认值: 预设

定义于: TerminationState

属性: 读/写

E.14.2 事件

无。

E.14.3 信号

无。

E.14.4 统计值

无。

E.14.5 差错代码

差错代码#: 459

名称: 未接收到分段

定义: 该差错指示, 在等待所有被传输的分段消息时, 某个分段TransactionReply接收超时。

差错描述符中的差错文本: 包含未接收到的分段号码。

注释: -

E.14.6 程序

由于某些传输机制并不对超过最大传输单元的消息进行自动地分段, 所以这些消息将会被截短。如果正在处理中的消息包含多个TransactionReply, 则发送端应该在单独的消息中发送每个TransactionReply。如果存在某些超过MTU长度的消息, 则发送端可能对TransactionReply执行下列分段程序。TransactionRequest则不应该被分段。

当对某个TransactionReply分段时, 发送端应确保该消息包含完整的命令和/或动作。每个分段都由事务处理层的附加分段信息来指示。每个分段都应使用同样的transactionID, 且在随后的分段中不应重复命令或已完成的动作。每个分段应该从1开始连续编号且一直延续到最后一个分段, 最后一个分段用包含SegmentationCompleteToken来指示。在所有情况下, 分段的消息应依照句法来有效地构造。接收端使用分段响应顺序地响应每个分段消息。由于分段消息最后组成一个完整的消息, 接收实体在处理任何个别的分段之前无需等待更多的分段。

一旦接收到最后一个分段, 接收端应确定它是否已经接收到所有的分段消息。如果是, 则接收端应该以TransactionResponseAcknowledgement响应。

如果接收端未接收到所有的分段, 则它应启动一个定时器等待丢失的分段。在接收端, 定时器的长度应是预置的, 但不应与其他实体上的ProvisionalResponseTimerValue相同(参见E.2.1)。如果丢失的分段在定时器超时之前仍未到达, 则接收端应以差错代码459(“未接收到分段”)响应。

示例1:

```
Sender:      !/3 [12.34.56.78]:2944 P=1/1{C=1{AV=term1{...}, AV=term2{...}}}  
Receiver:    !/3 [12.34.56.79]:2944 SM=1/1  
Sender:      !/3 [12.34.56.78]:2944 P=1/2{C=1{AV=term3{...}}, C=2{AV=term4{...}}}  
Receiver:    !/3 [12.34.56.79]:2944 SM=1/2  
Sender:      !/3 [12.34.56.78]:2944 P=1/3/#{C=3{AV=term5{...}}}  
Receiver:    !/3 [12.34.56.79]:2944 SM=1/3/#  
Receiver:    !/3 [12.34.56.79]:2944 K=1
```

示例2:

```
Sender:      !/3 [12.34.56.78]:2944 P=1/1{C=1{AV=term1{...}, AV=term2{...}}}  
Receiver:    !/3 [12.34.56.79]:2944 SM=1/1  
Sender:      !/3 [12.34.56.78]:2944 P=1/4/#{C=3{AV=term5{...}}}  
Receiver:    !/3 [12.34.56.79]:2944 SM=1/4/#  
/* Segmentation Timer Expires */  
Receiver:    !/3 [12.34.56.79]:2944 ER=459{"2,3"}
```

E.15 通知行为

包的名称：通知行为包

PackageID：nb (0x009a)

描述：该包具有这样的功能，即向MGC发请求的MG能够调节所发送的Notify命令。该包与7.1.9中描述的NotifyBehaviour标记有交互作用。现在版本的包描述了基于百分比调节的调节行为。其他类型的调节行为有待进一步研究。

版本：1

扩展：无

E.15.1 属性

E.15.1.1 通知调节

属性名称：通知调节

PropertyID：notreg (0x0001)

描述：该包指示MG上应被抑制的通知的百分比。值0%指示不抑制。值100%指示所有的通知都应该被抑制。该属性应仅仅定义在Root终结点上。当在某个事件上发送“Regulated Notify”时，MG使用Notification Regulation Property（通知调节属性）来确定需要调节的通知的百分比（即抑制比）。

类型：整数

可能的取值：0-100

默认值：无

定义于：TerminationState

属性：读/写

E.15.2 事件

无。

E.15.3 信号

无。

E.15.4 统计值

无。

E.15.5 程序

E.15.5.1 NotifyBehaviour

NotifyBehaviour机制允许MGC去管理它应接收多少来自MG的Notify请求命令。

E.15.5.1.1 NeverNotify

这里有一些事例，即MGC可能对接收到Notify请求命令并不感兴趣。例如，当检测该事件的发生实质上是为了触发一个嵌套的信号或事件时，但MGC并不需要被告知该事件的发生。将某个事件上的NotifyBehaviour参数设置为“NeverNotify”可以确保检测到该事件后不通知MGC，当然，任何嵌套的信号或事件都将被执行。NeverNotify与Notification Regulation Property(nb/notreg)（通知调节属性）并没有任何关系，因此而被标记的事件则未被包含在调节计算中。

E.15.5.1.2 NotifyImmediate

MGC可能需要被告知一些重要事件，诸如‘Emergency Call（紧急呼叫）’事件，甚至某些其他事件正在被调节。这可以通过对该特定事件将参数NotifyBehaviour设置为“NotifyImmediate”来实现。当相应事件被检测到时，将事件告知MGC就成为当前的默认行为。NotifyImmediate与Notification Regulation Property(nb/notreg)（通知调节属性）并没有任何关系，因此而被标记的事件则未被包含在调节计算中。

E.15.5.1.3 RegulatedNotify

E.15.5.1.3.1 Percentage Based Regulated Notify

当某个特定事件上的NotifyBehaviour参数设置为“NotifyRegulated”时，一旦发生了任何该特定事件，决定是否向MGC发送通知由Root终结点上的Notification Regulation (nb/notreg) Property（通知调节属性）来管理。该属性的值由MGC确定和设置。例如，调节速率可以对应于MGC的拥塞程度。MG则负责确定整个MG内部所有标记为“NotifyRegulated”的事件中应该被抑制的Notify命令的百分比。

MG可以实现任何完成固定Notification Regulation百分比的算法。应该选择那些使得不同MG之间同步风险最小的算法。一种推荐的算法给出如下：

对于给定的调节百分比（例如，10%），MG对具有RegulatedNotify标记的事件做下列动作：

- 1) 抑制一定数目的通知（例如，抑制通知1）。
- 2) 接收后续的通知直到到达NotificationRegulation百分比（例如，接收后续9个通知）。
- 3) 重复步骤1和2直到修正确保NotificationRegulation百分比。
- 4) 为了避免同步问题，MG应在上述推荐的通知序列中选择一个随机位置来启动算法。

如果事件已经将NotifyBehaviour参数设置为“RegulatedNotify”且Notification Regulation Property（通知调节属性）设置为0，则这将与事件设置为“ImmediateNotify”具有同样的效果，而“ImmediateNotify”是事件通知的默认行为。

NotifyBehaviour机制应该小心使用以避免不正确的状态。例如，如果“off-hook（挂机）”事件的通知被MG抑制，则MGC将不能够提供正确的响应（例如，拨号音，忙音）。这种不正确的行为可以通过使用嵌套信号或嵌套事件机制来避免，并返回正确的响应而不需要MGC同时参与事件响应。当标记为“NotifyRegulated”的事件被检测到并被接受（即通知未被抑制）时，MGC可以设置嵌套的信号或事件描述符来进行触发；当事件被检测到并被调节（即通知被抑制）时，单独的调节的嵌套信号或事件描述符将被触发。

注 — 即便在上述情况中，MG仍然无法察觉到一个呼叫的完整组成。简单地执行预先程序化的任务远胜于执行呼叫控制功能。

E.15.5.2 示例场景

给定这样的场景，即MGC和MG以模拟线来终结，基于MGC是否处于过载状态或模拟线（ALN）是否发起一个紧急呼叫，MGC/MG可能显示出不同的行为。不同的行为如下面的表E.1所示：

表 E.1/H.248.1 – 潜在的MGC/MG过载行为

MG 行为		1) 呼叫/关联类型	
		紧急呼叫	非紧急呼叫
任务：处理模拟线呼叫信令（ALS）=POTS 终端（ALN，模拟线接口）的呼叫控制信令业务量			
2) MGC有效载荷状态 注 – 假设了两个状态模型。	非过载状态 (例如) 低, 中等, 高有效载荷	I) MG: 具有最高优先级的正常ALS处理	II) MG: 具有低优先级的正常ALS处理
	过载状态	III) MG: 正常ALS处理 ==> 按“普通H.248模式”向MGC转发呼叫控制信令	IV) MG: 已编程的本地ALS处理 ==> 不涉及MGC ==> 通过“忙音”来实现呼叫协议上的拒绝

E.15.5.2.1 区间 I 和 II 的行为

在MG上不需要专门的NotifyBehaviour，因为MGC并未过载且能够接收所有来自MG的消息（通知）。当然，一旦MGC检测到呼叫的类型，它可以使用适当的Context Attribute（关联属性）来设置MG上的优先级。

E.15.5.2.2 区间 III 和 IV 的行为

当MGC已经检测到它处于过载状态，则它可能向NULL关联中所有或某些被选中的终结点组群发送下列命令：

消息示例：

```
Transaction=1234 {
  Context = - {
    Modify = aln/* {           ; 选择所希望的终结点
      Events = 1234 {
        al/of {
          RegulatedNotify {
            Embed {
              Signals {cg/dt} ,
; 如果 off-hook 被调节, 则用拨号音。
          Events = 1235 {
            xdd/xce {
              DigitMap = PriorityDialPlan1,
              bc = 20,
              mp = enhanced,
              ImmediateNotify
            },
; 如果是紧急呼叫, 则产生发给 MGC 的通知
            xdd/xce {
              DigitMap = NonPriorityDialPlan1,
              bc = 20,
              mp = enhanced,
              NeverNotify,
              Embed{
                Signals {cg/ct}
```

```

    },
    },
; 如果是低优先级呼叫，则发送忙音和 NeverNotify
    al/on {
        NeverNotify,
        ResetEventsDescriptor
    }
; 如果检测到 on-hook 则重新设置 EventsDescriptor 和 Nevernotify。
    }
    }
    }
    }
    }
    }
    }
}

```

注 — 依据这种行为在MG上设置的次数和事件，关联的身份可能是特定的或NULL且terminationID可以是通配符或特定的。如果随着MGC进入过载状态而使过载处理行为被加载，则使用通配符响应将会限制响应的数目。

E.15.5.2.3 如果Notification Regulation Property设置为非零（过载状态下）

在检测到off-hook事件且该事件被调节的情况下，MG的行为应该是：

- 抑制发往MGC的off-hook事件通知；
- 采用拨号音；
- 接收用户拨号的数字；
- 如果确定用户所拨的号码是一个紧急号码，将总是产生与“优先级拨号方案”相匹配的事件通知；
- 如果确定用户所拨的号码是一个非优先级号码，将总是抑制与“非优先级拨号方案”相匹配的事件通知并使用忙音；
- 实现on-hook（带有NeverNotify）事件的检测，一旦接收到on-hook事件将引起终结点上的事件描述符重新设置为嵌套被启用之前的状态（或MG预置的事件描述符）。

E.15.5.2.4 如果Notification Regulation Property设置为零（非过载状态下）

在以上情况下，MG的行为应该是：

- 向MGC发送off-hook事件通知。

如果使用来自MGC的明确的Modify命令来设置过载行为事件描述符（而不是使用MG的预置值），而MGC同时处于过载状态下，则对每个终结点而言，作为清除通报给MGC的任何呼叫/呼叫尝试序列的一部分，过载行为事件描述符将被重新加载。

当MGC过载情况终止时，MGC将NULL关联中的终结点重新设置成默认的非过载行为。作为替代，MGC应将Notification Regulation property设置为0。

附件 F

业务改变程序

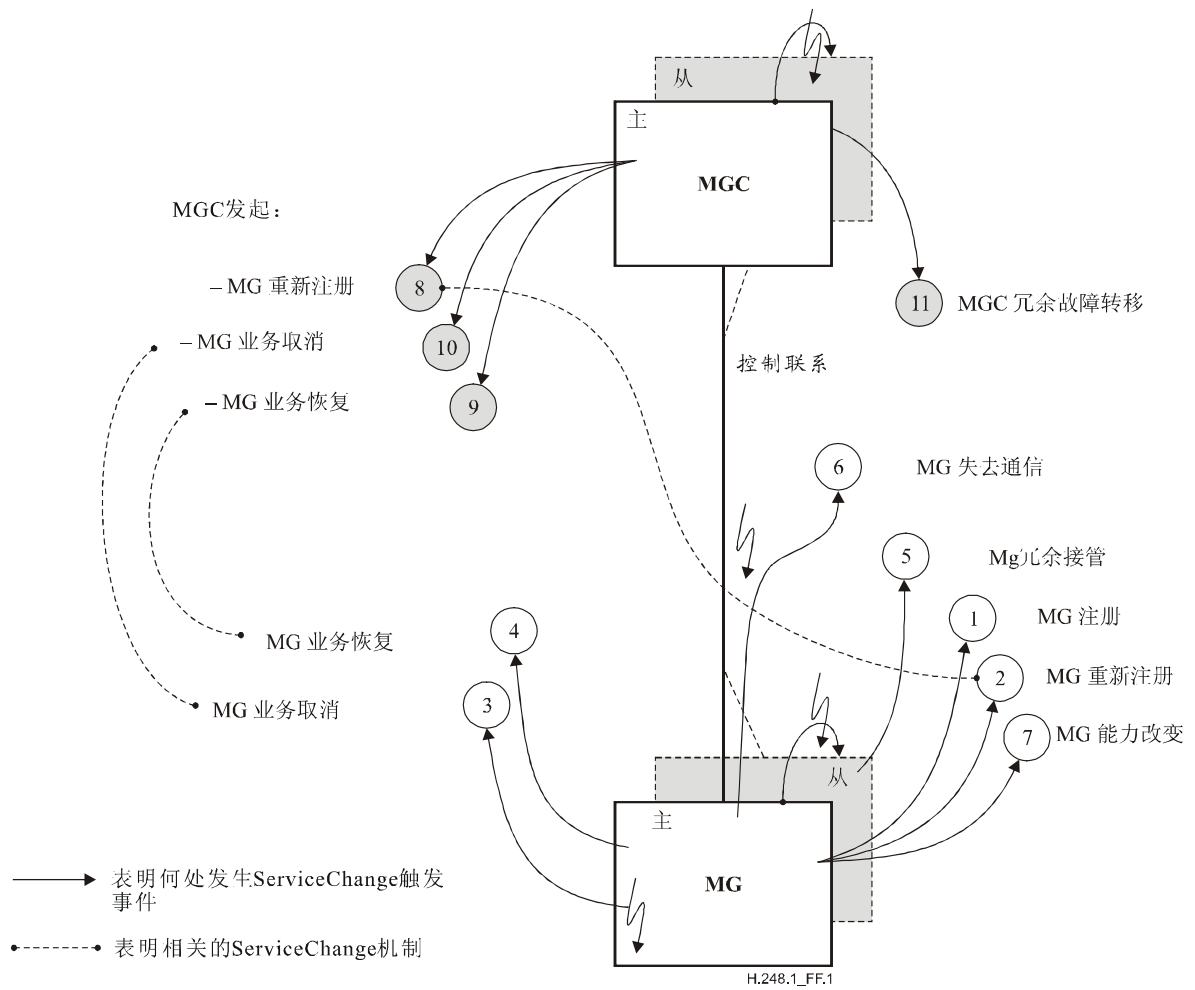
F.1 引言

本附件介绍了当某些事件发生在MG或MGC上时ServiceChange程序的使用。其目的是为了阐明7.2.8和第11节所描述的ServiceChange程序。如果本附件与本建议书存在差异，本建议书正文部分的程序比本附件所描述的那些程序具有高优先级。

导致ServiceChange命令发送的事件在本附件中描述如下：

- 1) MG注册 — MG向MGC注册并建立控制联系。
- 2) MG重新注册 — 当MG被MGC要求注册时，MG重新向MGC注册。
- 3) MG业务取消 — MG通知MGC，作为一个整体的终结点或MG将退出业务。
- 4) MG业务恢复 — MG从故障中恢复并通知MGC。
- 5) MG冗余切换 — 出现故障或对主控MG进行维护时，备用MG进行切换。
- 6) MG失去通信 — MG向MGC通告它先前与MGC失去通信但现在通信已经恢复了。
- 7) MG能力变化 — MG通知MGC，作为一个整体的MG或终结点的能力已经发生了变化。
- 8) MGC发起MG重新注册 — MGC通知MG，MG应该向同一MGC或其他MGC重新注册。
- 9) MGC发起业务恢复 — MGC从故障中恢复并命令MG将相关终结点置于初始状态。
- 10) MGC发起业务取消 — MGC通知MG，作为一个整体的MG或终结点将退出业务。
- 11) MGC故障冗余 — 主控MGC故障并指示MG向特定的备用MGC注册。

图F.1阐明了这些ServiceChange触发事件，展示了事件中的实体和相关的事件对。图F.2展示了MG和MG上的终结点应该遵循的状态模型。



图F.1/H.248.1 –ServiceChange的触发事件

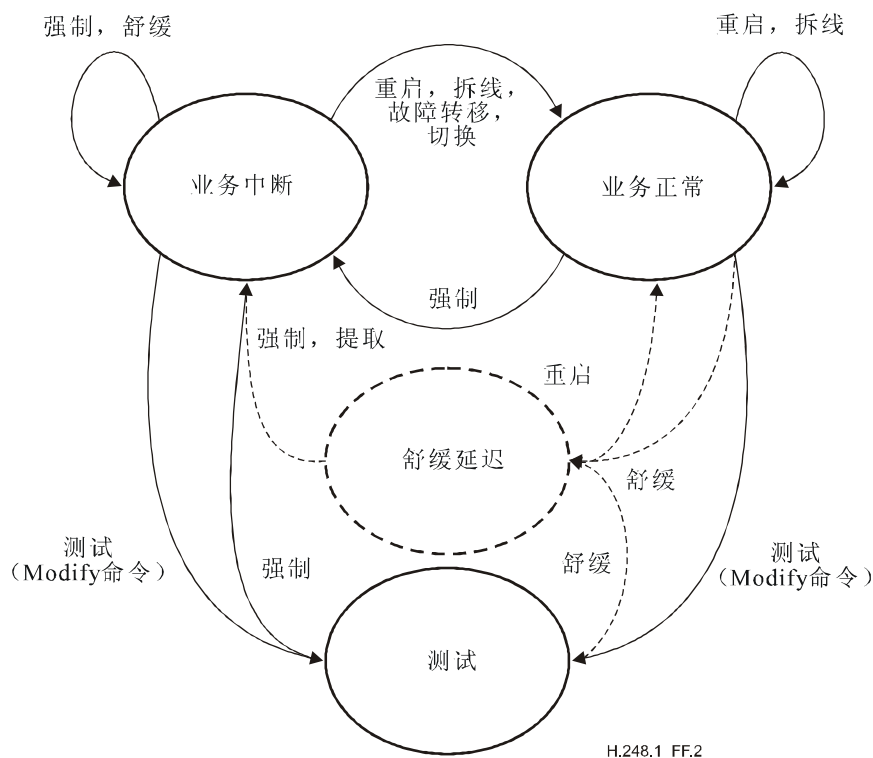


图 F.2/H.248.1 – 简单的ServiceChange状态模型

F.2 控制联系定义

控制联系是一种MGC凭借控制MG的通信关系。控制联系通过注册来实现。控制联系由于MG退出业务而终止，也可以成功地转移到替代MGC，或者成功地故障冗余至替代的MGC。在任何时候，MG应最多具有一个控制联系，除非一个物理的MG被分割为一个或多个虚拟MG。仅仅是MG能够建立控制联系。这里所定义的控制联系仅仅适用于媒体网关控制信令中的关联。该定义不应与传送协议关联（例如TCP、UDP）里的控制联系相混淆。

在Root终结点上使用ServiceChange命令仅仅影响Root终结点所属的虚拟MG。在同一物理MG上的虚拟MG并不受影响。

F.3 导致ServiceChange程序的事件

F.3.1 MG注册

MG可以用下列三种方式之一进行注册：

- 宣告其初始出现或“注册”，使用ServiceChangeReasons = 901 ("Cold Boot") 或 902 ("Warm Boot") 的ServiceChangeMethod "Restart"，
- 通过ServiceChangeMethod = "Failover"和ServiceChangeReason = 909("MGC Impending Failure") 或 908 ("MG Impending Failure")来指示MG正在改变其控制联系，
- 通过 ServiceChangeMethod = "Handoff" 和 ServiceChangeReason = 903 ("MGC Directed Change") (F.3.11)来指示MG被命令改变其控制联系。

当最初的ServiceChange命令由MG发往MGC时，注册开始；当MGC返回响应命令且该响应中没有替代的MGC地址或差错代码且ServiceChangeProfile在MGC和MG之间协商一致时，注册完成。一旦注册完成，则控制联系就建立了。注册总发生在Root终结点上。一旦初始化新的注册，则在MGC和MG之间的任何已有的控制联系将被终止。所有来自先前控制联系的命令都将被忽略。

注册消息使用ServiceChangeProfile，细节参见F.5.5。

注册消息使用ServiceChangeVersion，细节参见F.5.6。

F.3.2 MG重新注册

MG重新注册发生在两种情况下：

- 1) 当MGC要求MG重新注册（F.3.8）时，MG应发送ServiceChangeMethod = "Handoff"、ServiceChangeReason = 903 ("MGC Directed Change")和指定"ServiceChangeMgcID"的ServiceChange。
- 2) 当MGC发起业务恢复（F.3.9）时，MG应通过发送ServiceChangeMethod = "Restart"的ServiceChange来注册。

当MGC返回响应命令且该响应中没有替代的MGC地址或差错代码且ServiceChangeProfile在MGC和MG之间协商一致时，重新注册完成。如果未能从主控MGC接收到响应，则MG应遵循F.3.1的程序。建议采用ServiceChangeReasons = 903 ("MGC Directed Change")或909 ("MGC Impending Failure")的消息。

一旦完成注册，则控制联系就建立了。注册总是发生在Root终结点上。一旦初始化新的注册，则在MGC和MG之间的任何已有的控制联系将被终止。所有来自先前控制联系的命令都将被忽略。

注册消息使用ServiceChangeProfile，细节参见F.5.5。

注册消息使用ServiceChangeVersion，细节参见F.5.6。

F.3.3 MG业务取消

为了将MG置于"OutOfService"状态，MG在Root终结点上发送ServiceChangeMethod = "Forced"或"Graceful"的ServiceChange命令。细节参见F.4.1.1。

为了将一个终结点或一组终结点置于"OutOfService"状态，MG在所请求的终结点上发送ServiceChangeMethod = "Forced"或"Graceful"的ServiceChange命令。细节参见F.4.1.2和F.4.1.3。

ServiceChangeDelay指示业务取消将要发生之前的时间长度。细节参见F.5.3。

为了取消先前对整个MG发出的（和已经确认的）Graceful事件，MG在Root终结点上发送ServiceChangeMethod = "Restart"和ServiceChangeReason = 918("Cancel Graceful")的ServiceChange命令。MG将仍保持在InService状态，而且先前设置成OutOfService的所有终结点都返回业务态，除了那些被MG告知的终结点外。在Delay定时器超时之后接收到Cancel Graceful的事件中，则MG应必然进行重新注册，就如它发送Cold Boot事件一样。

为了取消先前发出的（和已经确认的）某个终结点或某组终结点上Graceful事件，MG在所请求的终结点上发送ServiceChangeMethod = "Restart"和ServiceChangeReason = "Cancel Graceful"的ServiceChange命令。终结点将仍保持在InService状态。在终结点已经转移至OutOfService状态的事件中，终结点应该返回至业务状态，就如它发送ServiceChange Restart事件一样。

F.3.4 MG业务恢复

当MG在故障或维护行为之后返回至业务时，将发生MG业务恢复事件。

当由MG发送消息时，MG向MGC发送ServiceChange命令来告知它已经重新启动或希望重新协商软件版本或H.248 Profile。在这种情况下使用ServiceChangeMethod="Restart" ServiceChangeReason=900 ("Service Restored")。ServiceChangeReason将指示MGC需要采取什么动作。

F.3.5 MG冗余切换

下列两种情况用于MG冗余切换：

1) 由主控MG发起的切换：

当MG即将退出业务并希望将处理转交给指定的备用MG，则MG向MGC发送ServiceChangeMethod="Failover"和ServiceChangeReason=908 ("MG Impending Failure")的ServiceChange命令。MGC将终止向主控MG发送消息并终止控制联系。备用MG将发送ServiceChangeMethod="Restart"和ServiceChangeReason=900 ("Service Restored")的ServiceChange命令。MGC将与备用MG建立新的控制联系。

2) 由备用MG发起的切换：

当可选的备用MG检测到主控MG进入维护期或故障期且主控MG不能向MGC通告故障时，备用MG向MGC发送ServiceChangeMethod="Failover"和ServiceChangeReason=919 ("Warm Failover")或920 ("Cold Failover")的ServiceChange命令。采用F.3.4程序。

MGC应忽略任何由未知MG发起的冗余切换尝试。通过预置或其他方式，MGC必须知道宣传冗余切换的备用MG是与故障的主控MG相关并授权认可对主控MG的切换。

如果备用MG是已知的且得到了对主控MG的切换授权，则MGC应尝试与主控MG通信以确定该MG是否仍然工作。这可以通过一个空的AuditValue命令或其他适当的方法（参见11.5）来完成。如果主控MG应答了，则MGC将拒绝故障切换尝试。如果主控MG无法应答，则MGC将假定主控MG已经发生故障了并接受warm或cold方式的故障切换。当MGC接收warm或cold方式的故障切换，则与主控MG之间的控制联系被消除并建议与备用MG之间的新控制联系。

F.3.6 MG失去通信

当MG检测到与MGC失去通信并随后重新建立与MGC的通信，MG在当前的控制联系中向MGC发送ServiceChangeMethod="Disconnected"的ServiceChange命令。如果MGC未能响应，MG则按顺序向其列表中的每个MGC发送ServiceChangeMethod="Failover"和ServiceChangeReason=909 ("MGC Impending Failure")的ServiceChange命令，直到成功地建立新的控制联系，否则MG将遍历其列表上的所有MGC。

如果最初控制联系中的MGC响应了ServiceChange命令，则控制联系继续存在而不会中断，而且所有的命令都正常处理就如同不存在通信丢失一样。否则，当MG向新的MGC发送ServiceChange命令时，最初的控制联系将被终止，而且所有来自先前控制联系的命令都将被忽略。一旦新MGC响应了ServiceChange并完成注册过程，则新的控制联系就建立了。

如果MG遍历了MGC列表而不能成功地建立控制联系，则MG等待一段随机长度的时间后又继续向其列表中的MGC尝试注册，从最初控制联系的MGC开始尝试。每次MG尝试与最初控制联系的MGC联系时，MG向该MGC发送ServiceChangeMethod="Disconnected"的ServiceChange。MG向所有其他MGC发送ServiceChangeMethod="Failover"的ServiceChange。

MGC接收到ServiceChangeMethod="Disconnected"的ServiceChange后应该审计MG以确定是否由于丢失消息而发送了状态不匹配的情况。

在这些场景中，推荐使用ServiceChangeReason=900 ("Service Restored")，尽管某些特殊的场景可能需要不同的ServiceChangeReason代码。

F.3.7 MG能力变化

为了通告MG能力上的变化，MG发送具有适当的ServiceChangeMethod、ServiceChangeReason=916 ("Packages Change")或917 ("Capabilities Change")的ServiceChange。MGC应审计MG以确定MG的新能力。对于处于业务态的MG且该MG发送了ServiceChangeMethod="Restart"的ServiceChange命令，则在能力改变过程中，控制联系应继续而不被中断，而且MG也不应被认为是经历了系统重启。

F.3.8 MGC发起MG重新注册

通过在Root终结点上发出ServiceChange命令，MGC可以请求MG重新注册，在该ServiceChange命令中，ServiceChangeMethod="Handoff"，ServiceChangeReason=903 ("MGC Directed Change")，并携带其自身的ServiceChangeMgcID (即当前MGC)。

MG所采取的动作参见F.3.2。

F.3.9 MGC发起业务恢复

为了请求MG重新启动，MGC向MG发送ServiceChange命令，其中ServiceChangeMethod="Restart"，ServiceChangeReason为适当的代码，诸如900("Service Restored")或901("Cold Boot")。按照F.3.2规定，MG必须使用明确的ServiceChangeReason来建立新的控制联系。

为了恢复某个终结点或某组终结点上的业务，MGC在所涉及的终结点上发送ServiceChangeMethod="Restart"的ServiceChange命令。MGC所采取的动作参见F.4.1.2和F.4.1.3。

ServiceChangeDelay指示业务恢复即将发生之前的时间长度。细节参见F.5.3。

F.3.10 MGC发起业务取消

F.3.10.1 Root 终结点

为了将MG置于"OutOfService"状态，MGC在Root终结点上发送ServiceChangeMethod="Forced"或"Graceful"的ServiceChange命令。适当的ServiceChangeReasons可以包括905("Termination taken out of service")。MGC所采取的动作参见F.4.1.1。

ServiceChangeDelay指示业务取消即将结束之前的时间长度。细节参见F.5.3。

为了取消先前对整个MG发出的（和已确认的）Graceful事件，MGC在Root终结点上发送ServiceChangeMethod="Restart"和ServiceChangeReason=918("Cancel Graceful")的ServiceChange命令。MG将仍保持在InService状态，而且先前设置成OutOfService的所有终结点都返回业务态，除了那些被MG告知的终结点外。在Delay定时器超时之后接收到Cancel Graceful的事件中，MG应向MGC报告差错代码502("Not Ready")。MG将被要求重新注册以返回业务态。

F.3.10.2 物理终结点

为了将某个终结点或某组终结点置于"OutOfService"状态，MGC在这些终结点上发送ServiceChangeMethod="Forced"或"Graceful"的ServiceChange命令。适当的ServiceChangeReasons可以包括904 ("Termination malfunctioning")、905 ("Termination taken out of service")、906 ("Loss of lower layer connectivity")或907 ("Transmission failure")。MGC所采取的动作参见F.4.2和F.4.3。

ServiceChangeDelay指示业务取消即将结束之前的时间长度。细节参见F.5.3。

为了取消先前对某个终结点或某组终结点发出的（和已确认的）Graceful事件，MG在这些终结点上发送ServiceChangeMethod="Restart"和ServiceChangeReason=918("Cancel Graceful")的ServiceChange命令。这些终结点将仍保持在InService状态。在终结点已经转移到OutOfService状态的事件中，终结点应该返回至业务状态就如同它使用ServiceChange Restart一样。

F.3.10.3 临时终结点

MGC应不使用ServiceChange来取消临时终结点的业务。从关联中减去终结点就足够删除终结点了。

F.3.11 MGC故障冗余

当控制联系中的MGC遭遇到维护或故障情况以至于MGC必须进入OutOfService状态，则MGC可以通过向MG发送ServiceChangeMethod="Handoff"、ServiceChangeReason=903 ("MGC Directed Change")、ServiceChangeMgcID参数设置为新MGC地址的ServiceChange命令来引导MG向特定的备用MGC注册。一旦接收到MG的响应，控制联系将被终止。MG向指定的MGC发送ServiceChangeMethod="Handoff"、ServiceChangeReason=903 ("MGC Directed Change")的ServiceChange命令。一旦接收到来自MGC的响应，则新的控制联系就建立了。

如果指定MGC拒绝了切换请求或MGC由于故障不能够发送带有ServiceChangeMethod Handoff的ServiceChange命令，则MG应按照F.3.6所述来执行失去通信的程序，并从其先前预置的主控MGC开始遍历。

F.4 ServiceChange单元描述

F.4.1 ServiceChangeMethod

本小节描述了不同终结点类型上的ServiceChangeMethod行为。本小节组织如下：

- Root终结点；
- 物理终结点；
- 临时终结点。

F.4.1.1 在Root终结点上的ServiceChange Method行为

依据不同的ServiceChangeMethod，在Root终结点上的ServiceChange命令有不同的效果。每个ServiceChangeMethod的结果描述如下：

- 1) Restart — 当由MG发送时，用于通告MG重新启动了并希望重新协商协议版本、Profile或通告能力的变化。ServiceChangeReason指示MGC所应该采取的动作。当由MGC发送该参数时，则MG应使用所指示的ServiceChangeReason重新启动。

- 2) **Forced** — 当由MG发送时，MG指示它将立刻进入OutOfService状态。MG一旦接收到来自MGC的命令响应，MG就终止控制联系。当由MGC发送时，MG应使其自身置于OutOfService状态，并在发送命令响应之后终止控制联系。注意：MGC不能够使MG重新返回业务态，尽管MGC可以请求MG进入OutOfService状态。MG向MGC发起注册以建立新的控制联系并指示业务恢复，就如同它完全其他的注册程序一样。ServiceChangeDelay对ServiceChangeMethod="Forced"没有任何影响。
- 3) **Graceful** — 当由MG发送时，指示MG将在ServiceChangeDelay期间的结束时进入OutOfService状态。控制联系将在ServiceChangeDelay期间的结束时被终止。当由MGC发送时，MG将其自身置于OutOfService状态并在ServiceChangeDelay期间的结束时终止控制联系。使用ServiceChangeDelay=0或不使用ServiceChangeDelay指示MG应进入OutOfService并终止控制联系，当最后一个关联通过删减其终结点而被删除时，并指示MGC不应增加新连接。MG应在ServiceChangeDelay期满后设置Root终结点的ServiceStates属性为“OutOfService”或从活动的关联中消除所有终结点（不管哪个是第一个）。为了取消先前发送的（和已确认的）ServiceChange命令且该命令的ServiceChangeMethod="Graceful"，发起ServiceChangeMethod=Graceful的实体发送ServiceChangeMethod=Restart和ServiceChangeReason=918("Cancel Graceful")的ServiceChange命令。
- 4) **Failover** — 当由MG向MGC发送且该MGC不在当前控制联系中时，MG指示其当前MGC已经发生故障了，它正通过接收响应命令来尝试注册。当由MG向MGC发送且该MGC在当前控制联系中时，主控MG指示备用MG已经切换了出现故障的主控MG。不管在哪种情况下，一旦发送了ServiceChange命令则先前的控制联系被终止，一旦接收到命令响应，则新的控制联系在MG与新MGC或MGC与新的MG之间建立。MGC不应发送ServiceChangeMethod="Failover"的ServiceChange命令。
- 5) **Handoff** — 当由MGC发送时，MGC指示MG将转移至新的MGC。一旦接收到命令响应，这将终止当前的控制联系。当由MG发送至MGC时，这指示MG正在按照在先前控制联系上接收到的MGC发送的切换命令来尝试建立新的控制联系。当由MG发送至MGC时，切换就是一个注册过程，一旦接收到命令响应就建立了新的控制联系。没有MGC命令MG这样做，MG不应使用Handoff。
- 6) **Disconnected** — 当由MG发送时，指示当前控制联系的通信已经丢失，但是现在已经重新建立了。当前控制联系被更新了。MGC不应发送ServiceChangeMethod="Disconnected"的ServiceChange命令。

F.4.1.2 在物理终结点上的ServiceChange Method行为

当MG处于活动的控制联系中时，依据不同的ServiceChangeMethod，在物理终结点上的ServiceChange命令有不同的效果。每个ServiceChangeMethod的结果描述如下：

- 1) **Restart** — 当由MG发送时，用于通告终结点已经重新启动了或通告能力的变化。ServiceChangeReason指示MGC所应该采取的动作。当由MGC发送该参数时，则MG应使用所指示的ServiceChangeReason重新启动终结点。

- 2) **Forced** — 当由MG发送时，指示终结点将立刻进入OutOfService状态。当由MGC发送时，MG应使终结点立即置于OutOfService状态。在以上两种情况下，ServiceStates参数应设置为"OutOfService"且MGC负责清除与该终结点相关的任何关联或资源。ServiceChangeDelay对ServiceChangeMethod="Forced"没有任何影响。
- 3) **Graceful** — 当由MG发送时，指示终结点将在ServiceChangeDelay期间的结束时进入OutOfService状态。当由MGC发送时，MG将在ServiceChangeDelay期间的结束时把该终结点置于OutOfService状态。一旦ServiceChangeDelay期满，或当终结点从活动的关联（不管哪个是第一个）中被删除且MGC负责清除与终结点相关的任何关联或资源时，ServiceStates属性就应设置为"OutOfService"。使用ServiceChangeDelay=0或不使用ServiceChangeDelay指示终结点应进入OutOfService，当通过删减命令将该终结点从关联中删除时。MGC不应为所指示的终结点建立连接直到Graceful被取消或终结点由后续的ServiceChange命令重新返回业务。为了取消先前发送的（和已确认的）ServiceChange命令且该命令的ServiceChangeMethod="Graceful"，发起ServiceChangeMethod=Graceful的实体发送ServiceChangeMethod=Restart和ServiceChangeReason=918 ("Cancel Graceful")的ServiceChange命令。
- 4) **Failover** — ServiceChangeMethod="Failover"不应用在非Root终结点上。
- 5) **Handoff** — ServiceChangeMethod="Handoff"不应用在非Root终结点上。
- 6) **Disconnected** — ServiceChangeMethod="Disconnected"不应用在非Root终结点上。

F.4.1.3 在临时终结点上的ServiceChange Method 行为

当MG处于活动的控制联系中时，依据不同的ServiceChangeMethod，在临时终结点上的ServiceChange命令有不同的效果。每个ServiceChangeMethod的结果描述如下：

- 1) **Restart** — 当由MG发送时，用于通告终结点已经重新启动了或通告能力的变化。ServiceChangeReason指示MGC所应该采取的动作。MGC不应向临时终结点发送ServiceChangeMethod="Restart"。
- 2) **Forced** — 当由MG发送时，指示终结点将立刻进入OutOfService状态。MGC负责删减终结点。MGC不应向临时终结点发送ServiceChangeMethod="Forced"。ServiceChangeDelay对ServiceChangeMethod="Forced"没有任何影响。
- 3) **Graceful** — 当由MG发送时，指示终结点将在ServiceChangeDelay期间的结束时进入OutOfService状态。MGC负责在ServiceChangeDelay期满时删减终结点。MGC不应向临时终结点发送ServiceChangeMethod="Graceful"。当终结点通过删除命令下关联中删除时，使用ServiceChangeDelay=0指示终结点将被消灭。MG将在ServiceChangeDelay期满或将该终结点从活动关联（不管哪个是第一个）中消除时把该终结点的ServiceStates属性置于"Out of Service"状态。为了取消先前发送的（和已确认的）ServiceChange命令且该命令的ServiceChangeMethod="Graceful"，发起ServiceChangeMethod=Graceful的实体发送ServiceChangeMethod=Restart和ServiceChangeReason=918 ("Cancel Graceful")的ServiceChange命令。

- 4) Failover — ServiceChangeMethod="Failover"不应用在非Root终结点上。
- 5) Handoff — ServiceChangeMethod="Handoff"不应用在非Root终结点上。
- 6) Disconnected — ServiceChangeMethod="Disconnected"不应用在非Root终结点上。

F.5 ServiceChange 参数的使用

F.5.1 ServiceChangeMethod

参见F.4中所描述的用法。

F.5.2 ServiceChangeReason

ServiceChangeReasons 允许接收端改变其行为以适应特殊的环境。例如，如果MG在其重新启动时向MGC发送ServiceChangeReason=901 ("Cold Boot")，则MGC可以假设MG脱离了其所有的状态，因此，MGC将不执行维护和审计任务以评估和整理MG至可用状态。

表F.1显示了特定ServiceChangeReasons发送时所携带的ServiceChangeMethod。

表 F.1/H.248.1 – ServiceChangeMethod和服务ChangeReason映射

SC 原因	SC方法						描述
	重启	强制	暂缓	拆线	故障 转移	切换	
900	X			仅用于根终结 点和MG			业务恢复
901	仅用于 根终结 点						冷启动
902	仅用于 根终结 点						热启动
903						仅用于 根终结 点	MGC重定向
904		X	X				终结点故障
905		X	X				终结点退出服务
906		X	X				底层连接丢失
907		X	X				传输故障
908		仅用于 根终结 点和 MG	仅用于 根终结 点和MG		仅用于 根终结 点和MG		MG临近故障
909					仅用于 根终结 点和MG		MGC临近故障
910	仅用于 MG	X	X				媒体能力故障
911	仅用于	X	X				Modem能力故障

表 F.1/H.248.1 – ServiceChangeMethod和服务ChangeReason映射

SC原因	SC方法						描述
	重启	强制	暂缓	拆线	故障转移	切换	
	MG						
912	仅用于MG	X	X				Mux能力故障
913	仅用于MG	X	X				信号能力故障
914	仅用于MG	X	X				事件能力故障
915		X	X				状态丢失
916	X			仅用于根终结点和MG	仅用于根终结点和MG		包类型改变
917	X			仅用于根终结点和MG	仅用于根终结点和MG		能力改变
918	X						取消暂缓
919					仅用于根终结点和MG		热故障切换
920					仅用于根终结点和MG		冷故障切换

F.5.3 ServiceChangeDelay

ServiceChangeDelay用于提供ServiceChange命令生效和MG或终结点上ServiceStates属性被改变之前的延时。ServiceChange命令被应答就如同接收到该消息时被执行一样，但是MG或终结点实际上并不改变状态直到ServiceChangeDelay期满。等于0的ServiceChangeDelay就等同于没有ServiceChangeDelay，但在与ServiceChangeMethod = "Graceful" 同时使用时例外。参见 F.4 以了解 ServiceChangeDelay 用不同ServiceChangeMethods上产生的效果。

F.5.4 ServiceChangeAddress

ServiceChangeAddress参数的使用在7.2.8描述。ServiceChangeAddress的使用是不推荐的。如果该阐述出现了，则该参数仅仅用于根终结点上的ServiceChange命令，而且任何新的事务将被发送至所指定的新地址和/或新端口号。应答应回送至相应的请求所发出的地址。

F.5.5 ServiceChangeProfile

ServiceChangeProfile参数允许MGC和MG去协商H.248 Profile以用于控制联系。MGC仍将审计MG以确定其他被支持的MG能力。ServiceChangeProfile应仅仅在注册和重新注册命令中被发送。

F.5.6 ServiceChangeVersion

ServiceChangeVersion用于协商H.248协议版本以用于MGC和MG之间。ServiceChangeVersion在初始注册中必选项，而且应该在任何其他注册命令中被发送，这些命令需要进行协议版本协商。ServiceChangeVersion不应在非注册命令中发送。参见F.3.1的注册程序。

F.5.7 ServiceChangeMgcID

MGC可能在去往Root终结点的ServiceChange命令中发送该参数。在注册尝试中一旦接收到到该参数，则MG应尝试向指定地址的MGC进行注册。当接收到来自MG的主控MGC发出的ServiceChange Handoff命令时，MG应使用11.5所描述的程序。

F.5.8 TimeStamp

可选TimeStamp参数的使用在7.2.8中描述。TimeStamp参数对ServiceChange命令的执行没有任何影响，但对于某些其他目的的ServiceChange命令接收者而言可能有用，诸如计费或定时协调。

F.6 ServiceChange 对 TerminationState

在TerminationState描述符中的ServiceStates属性保持有终结点状态的当前值。ServiceStates可能采用三个值："InService", "OutOfService", and "Test"。

MGC应仅仅使用Modify命令来设置ServiceStates属性成Test或从Test状态转移时设置。对于ServiceStates属性的所有其他修改都是对协议的损坏并应该接收到差错代码401("Protocol Error")。为了从状态"InService"转移到"OutOfService"或反过来，必须使用ServiceChange命令。

对于Test状态的终结点，MG应向MGC仅发送ServiceChangeMethod = "Forced" 或 "Graceful" 的ServiceChanges。随后的ServiceChanges命令将终结点的ServiceState属性设置为"InService"。

接收者应该进行每一次合理的尝试以接受ServiceChange命令，但在某些情况下，拒绝该命令是有正当理由的。例如，如果MG发送ServiceChange命令使终结点开始进入业务态，但MGC由于缺乏预置资源或其他原因没有足够的资源分配给该终结点，则MGC可能拒绝该命令。如果接收实体拒绝了ServiceChange命令，则终结点的状态应保持不变。发送端可以选择等待一定长度的事件并再次尝试，或等待接收者尝试改变终结点的状态。使终结点OutOfService的请求应该被处理。

表 F.2显示了什么命令可以用于影响终结点上的状态改变。

表F.2/H.248.1 – 状态转移命令及其影响

当前状态	新状态	命令	MGC允许?	MG允许?
InService	Test	ServiceChange	不可能	不可能
		Modify	是	不可能
InService	OutOfService	ServiceChange	是, 但仅用于物理和Root终结点	是
		Modify	否	不可能
OutOfService	InService	ServiceChange	是, 但不能用在Root终结点	是
		Modify	否	不可能
OutOfService	Test	ServiceChange	不可能	不可能
		Modify	是	不可能
Test	InService	ServiceChange	否	否
		Modify	是	不可能

表F.2/H.248.1 – 状态转移命令及其影响

Test	OutOfService	ServiceChange	否	是
		Modify	是	不可能

附 录 一

呼叫流程示例

所有H.248.1实施者在实施本建议书之前务必须认真阅读本建议书的规范性部分。任何人都不应将本附录提供的示例用做关于生成协议消息的单独解释。

本附录中使用SDP协议来定义本地和远端描述符，SDP协议具体内容参见RFC 2327。如果本附录提供的示范程序中与RFC 2327定义的SDP不同，则应该确认RFC文档可以修改。有关语音文件的定义参见RFC 1890和其他已在IANA注册的文档中。例如，在SDP协议中，G.711 A率称为PCMA，其类型被指定为0。G.723.1被成为G.723，其类型被指定为6。H.263被称为H263，其类型被指定为34。具体内容参见<http://www.iana.org/assignments/rtp-parameters>。

I.1 驻地网关之间的呼叫程序

本附录中的示范程序用于对IP网络中驻地网关之间的呼叫建立程序进行说明。为简单起见，假定两个驻地网关均受同一个MGC控制。

I.1.1 驻地网关中模拟线终结点空闲

本节以下内容所说明的MG和MGC之间的API接口用于确保该环境下的终结点处于空闲状态。主叫MG和被叫MG中的模拟线终结点处于空闲状态，且终结点被通过Modify命令进行配置，用于检测呼叫发起事件（如着摘机）。空关联用于指示关联中尚未包含终结点。根终结点用于指示整个MG，而不是MG内的某个终结点。

在该示范程序中，MG1的IP地址为124.124.124.222，MG2的IP地址为125.125.125.111，MGC的IP地址为123.123.123.4。MG1、MG2和MGC的协议默认端口为55555。

1) MG向MGC发送ServiceChange命令进行注册，命令如下：

MG1 至 MGC：

```
MEGACO/1 [124.124.124.222]
Transaction = 9998 {
  Context = - {
    ServiceChange = ROOT {Services {
      Method=Restart, Version=3,
      ServiceChangeAddress=55555, Profile=ResGW/1}
    }
  }
}
```

2) MGC 回送的响应如下:

```
MGC 至 MG1:
MEGACO/1 [123.123.123.4]:55555
Reply = 9998 {
  Context = - {ServiceChange = ROOT {
    Services {ServiceChangeAddress=55555, Profile=ResGW/1} } }
}
```

3) MGC将修改空关联中的一个终结点, TerminationId为A4444, streamId为1, 事件描述符中的requestId为2222。消息发送方的mID采用IP 地址和端口号的形式: [123.123.123.4]:55555。该媒体流的模式设置为“SendReceive”。“al”是模拟线检测包的缩写。本地和远端描述符为临时分配的。

```
MGC 至 MG1:
MEGACO/3 [123.123.123.4]:55555
Transaction = 9999 {
  Context = - {
    Modify = A4444 {
      Media { Stream = 1 {
        LocalControl {
          Mode = SendRecv,
          tdmc/gain=2, ; in dB,
          tdmc/ec=on
        },
      },
    },
    Events = 2222 {al/of {strict=state}}
  }
}
```

拨号计划可以已经预先记载在MG中。此时, 应等待摘机事件发生, 并播放拨号音和开始收集DTMF数字。然而, 在本示范程序中, 将在摘机事件发生后使用digitmap, 如步骤5) 所示。

如果可使用内嵌的事件描述符来集合步骤3) 和4), 步骤8) 和9), 从而步骤6)和7)可以省略。

4) MGC和 MG2之间的消息交互类似, 此时, 空闲终结点为A5555:

```
MG1 至 MGC:
MEGACO/3 [124.124.124.222]:55555
Reply = 9999 {
  Context = - {Modify = A4444}
}
```

5) MGC和 MG2之间的消息交互类似, 此时, 空闲终结点为A5555。

I.1.2 收集主叫方拨号事件和初始化终结点

以下步骤是在完成前面所述步骤的基础上进行的。从MGC发送至 MG1的事务用于获得主叫方终结点A4444收集的拨号事件后, 发起至MG2的连接。

6) MG1检测到用户1的摘机事件，则向MGC上报Notify命令。

```
MG1 至 MGC:
MEGACO/3 [124.124.124.222]:55555
Transaction = 10000 {
  Context = - {
    Notify = A4444 {ObservedEvents =2222 {
      19990729T22000000:al/of(init=OFF)}}
  }
}
```

7) Notify命令得到证实。

```
MGC 至 MG1:
MEGACO/3 [123.123.123.4]:55555
Reply = 10000 {
  Context = - {Notify = A4444}
}
```

8) MGC修改终结点，指示终结点播放拨号音，并按照Dialplan0的DigitMap收集数字，以及开始检测挂机事件。

```
MGC 至 MG1:
MEGACO/3 [123.123.123.4]:55555
Transaction = 10001 {
  Context = - {
    Modify = A4444 {
      Events = 2223 {
        al/on(strict=state), dd/ce {DigitMap=Dialplan0}
      },
      Signals {cg/dt},
      DigitMap= Dialplan0{
(0| 00|[1-7]xxx|8xxxxxxx|Fxxxxxxx|Exx|91xxxxxxxxxxx|9011x.)}
      }
    }
  }
}
```

9) Modify得到证实。

```
MG1 至 MGC:
MEGACO/3 [124.124.124.222]:55555
Reply = 10001 {
  Context = - {Modify = A4444}
}
```

10) 下一步，MG1收集了用户1所拨的被叫号码。检测到第一个被叫号码数字时，停止播放拨号音。当收集的被叫号码与A4444终结点上的拨号计划完全匹配时，向MGC发送 Notify命令。

```
MG1 至 MGC:
MEGACO/3 [124.124.124.222]:55555
Transaction = 10002 {
  Context = - {
    Notify = A4444 {ObservedEvents =2223 {
      19990729T22010001:dd/ce{ds="916135551212",Meth=UM}}}
  }
}
```

11) Notify命令得到证实。

```
MGC 至 MG1:
MEGACO/3 [123.123.123.4]:55555
Reply = 10002 {
    Context = - {Notify = A4444}
}
```

12) MGC分析被叫号码后，确定需要从MG1建立到MG2的连接。A4444终结点和RTP终结点被添加到MG1中的一个新的关联中。由于远端描述符未指定，此时，终结点模式设置为ReceiveOnly。MGC期望的编码按照优先级进行排列。

```
MGC 至 MG1:
MEGACO/3 [123.123.123.4]:55555
Transaction = 10003 {
    Context = $ {
        Add = A4444,
        Add = $ {
            Media {
                Stream = 1 {
                    LocalControl {
                        Mode = RecvOnly,

                        nt/jit=40 ; in ms
                    },
                    Local {
v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
a=ptime:30
v=0
c=IN IP4 $
m=audio $ RTP/AVP 0
                }
            }
        }
    }
}
```

注一 MGC可以本地描述符中的SDP部分包含其所期望的参数值。MG应在响应中返回本地描述符。

13) MG1返回响应，其中包含一个新的终结点，且填充本地描述符中IP地址和端口号。同时，它返回MG所期望的编码的选择结果。MG1将RTP端口号设置为2222。

```
MEGACO/3 [124.124.124.222]:55555
Reply = 10003 {
    Context = 2000 {
        Add = A4444,
        Add=A4445{
            Media {
                Stream = 1 {
                    Local {
v=0
o=- 2890844526 2890842807 IN IP4 124.124.124.222
s=-
t= 0 0
c=IN IP4 124.124.124.222
m=audio 2222 RTP/AVP 4
a=ptime:30
a=recvonly
                } ; RTP profile for G.723.1 is 4
            }
        }
    }
}
```

```

    }
  }
}

```

- 14) MGC将MG2上的终结点添加到一个新的关联，并建立一个RTP媒体流，例如，终结点标识符为A5556，模式设置为SendReceive。同时，MGC指示终结点A5555播放振铃信号音。

MGC 至 MG2:

```

MEGACO/3 [123.123.123.4]:55555
Transaction = 50003 {
  Context = $ {
    Add = A5555 { Media {
      Stream = 1 {
        LocalControl {Mode = SendRecv} }},
    Events=1234{al/of(strict=state)},
    Signals {al/ri}
  },
  Add = $ {Media {
    Stream = 1 {
      LocalControl {
        Mode = SendRecv,
        nt/jit=40 ; in ms
      },
      Local {
v=0
c=IN IP4 $
m=audio $ RTP/AVP 4
a=ptime:30
      },
      Remote {
v=0
c=IN IP4 124.124.124.222
m=audio 2222 RTP/AVP 4
a=ptime:30
    } ; G.723.1 的 RTPprofile 等于 4
      }
    }
  }
}
}
}

```

- 15) 该命令得到证实。媒体流的端口号应与控制流端口号不同，例如，端口号为1111（在SDP中）。

MG2 至 MGC:

```

MEGACO/3 [125.125.125.111]:55555
Reply = 50003 {
  Context = 5000 {
    Add = A5555,
    Add = A5556{
      Media {
        Stream = 1 {
          Local {
v=0

o=- 7736844526 7736842807 IN IP4 125.125.125.111
s=-
t= 0 0
c=IN IP4 125.125.125.111
m=audio 1111 RTP/AVP 4
      }
    } ; G.723.1 的 RTP profile 等于 4
  }
}
}

```

```

    }
}

```

16) 此时，将步骤15)中的MG2的IP地址和端口号告知MG1。

MGC 至 MG1:

```

MEGACO/3 [123.123.123.4]:55555
Transaction = 10005 {
  Context = 2000 {
    Modify = A4444 {
      Signals {cg/rt}
    },
    Modify = A4445 {
      Media {
        Stream = 1 {
          Remote {
v=0

o=- 7736844526 7736842807 IN IP4 125.125.125.111
s=-
t= 0 0
c=IN IP4 125.125.125.111
m=audio 1111 RTP/AVP 4
          }
        } ; G.723.1 的 RTP profile 等于 4
      }
    }
  }
}

```

MG1 至 MGC:

```

MEGACO/3 [124.124.124.222]:55555
Reply = 10005 {
  Context = 2000 {Modify = A4444, Modify = A4445}
}

```

17) 此时两个网关建立连接，用户1听到回铃音。MG2此时应等待用户2摘机，摘机后，建立双向连接。

从 MG2 至 MGC:

```

MEGACO/3 [125.125.125.111]:55555
Transaction = 50005 {
  Context = 5000 {
    Notify = A5555 {ObservedEvents =1234 {
      19990729T22020002:al/of(init=off)}}
  }
}

```

从 MGC 至 MG2:

```

MEGACO/3 [123.123.123.4]:55555
Reply = 50005 {
  Context = - {Notify = A5555}
}

```

从 MGC 至 MG2:

```

MEGACO/3 [123.123.123.4]:55555
Transaction = 50006 {
  Context = 5000 {
    Modify = A5555 {

```



```

        Events = 1235 {al/on(strict=state)},
        Signals ; to turn off ringing
    }
}
}

```

从 MG2 至 MGC:

```

MEGACO/3 [125.125.125.111]:55555
Reply = 50006 {
  Context = 5000 {Modify = A4445}
}

```

18) 将MG1的模式修改为SendReceive, 并停止播放回铃音。

MGC 至 MG1:

```

MEGACO/3 [123.123.123.4]:55555
Transaction = 10006 {
  Context = 2000 {
    Modify = A4445 {
      Media {
        Stream = 1 {
          LocalControl {
            Mode=SendRecv
          }
        }
      }
    },
    Modify = A4444 {
      Signals
    }
  }
}

```

从 MG1 至 MGC:

```

MEGACO/3 [124.124.124.222]:55555
Reply = 10006 {
  Context = 2000 {Modify = A4445, Modify = A4444}}

```

19) MGC指示对MG2上的 RTP 终结点进行审计。

```

MEGACO/3 [123.123.123.4]:55555
Transaction = 50007 {
  Context = 5000 {AuditValue = A5556{
    Audit{Media, DigitMap, Events, Signals, Packages, Statistics }}
  }
}

```

20) MGC返回命令响应。

```

MEGACO/3 [125.125.125.111]:55555
Reply = 50007 {
  Context = 5000 {
    AuditValue = A5556 {
      Media {
        TerminationState { ServiceStates = InService,
          Buffer = OFF },
        Stream = 1 {
          LocalControl { Mode = SendRecv,
            nt/jit=40 },
          Local {

```

v=0

o=- 7736844526 7736842807 IN IP4 125.125.125.111

```

s=-
t= 0 0
c=IN IP4 125.125.125.111
m=audio 1111 RTP/AVP 4
a=ptime:30
    },
    Remote {
v=0

o=- 2890844526 2890842807 IN IP4 124.124.124.222
s=-
t= 0 0
c=IN IP4 124.124.124.222
m=audio 2222 RTP/AVP 4
a=ptime:30
    } } },
    Events,
    Signals,
    DigitMap,
    Packages {nt-1, rtp-1},
    Statistics { rtp/ps=1200, ; 发送的包数目
                 nt/os=62300, ; 发送的字节数
                 rtp/pr=700, ; 接收的包数目
                 nt/or=45100, ; 接收的字节数
                 rtp/pl=0.2, ; 包丢失率%
                 rtp/jit=20,
                 rtp/delay=40 } ; 平时时延
    }
}
}

```

- 21) 当MGC接收到来自任何一个MG的挂机事件时，应拆除呼叫。本示例程序中，假定MG2上的用户先挂机。

从 MG2 至 MGC:

```

MEGACO/3 [125.125.125.111]:55555
Transaction = 50008 {
  Context = 5000 {
    Notify = A5555 {ObservedEvents =1235 {
      19990729T24020002:al/on(init=off)}
    }
  }
}

```

从 MGC 至 MG2:

```

MEGACO/3 [123.123.123.4]:55555
Reply = 50008 {
  Context = - {Notify = A5555}
}

```

- 22) MGC开始向主被叫MG发送Subtract命令来拆除呼叫连接，且每个终结点有响应的统计参数。MGC可以要求A5555和A5556返回其统计参数。A5555是物理终结点，A5556是RTP终结点。

从 MGC 至 MG2:

```

MEGACO/3 [123.123.123.4]:55555
Transaction = 50009 {
  Context = 5000 {
    Subtract = A5555 {Audit{Statistics}},

```

```

    Subtract = A5556 {Audit{Statistics}}
  }
}

```

从 MG2 至 MGC:

```

MEGACO/3 [125.125.125.111]:55555
Reply = 50009 {
  Context = 5000 {
    Subtract = A5555 {
      Statistics {
        nt/os=45123, ; 发送的字节数
        nt/or=45123, ; 接收的字节数
        nt/dur=40000 ; 单位为毫秒
      }
    },
    Subtract = A5556 {
      Statistics {
        rtp/ps=1245, ; 发送的包数目
        nt/os=62345, ; 发送的字节数
        rtp/pr=780, ; 接收的包数目
        nt/or=45123, ; 接收的字节数
        rtp/pl=10, ; 包丢失率%
        rtp/jit=27,
        rtp/delay=48, ; 平均时延
        nt/dur=38000 ; 单位为毫秒
      }
    }
  }
}

```

- 23) MGC开始指示MG1和MG2检测下一个摘机事件，如步骤1)所示。此时，如果处于空关联中的终结点处于默认状态，则不需从MGC发送消息至MG。一旦终结点返回至空关联中，该终结点应返回至默认参数值。

附录二

H.248 包模版

新的和已更新的H.248包应使用下列模版来定义。这是一个国际电联建议书的格式。来自非国际电联组织的编者至少应该使用模版的第6节中的结构。H.248包关键字以黑体字的形式来指示。需要被填充的部分用"<>"指示。如何创建新包的详细信息参见第12节。

注 — 包的特定差错代码需要向IANA注册。差错代码的分配应遵循4.1 /H.248.8 “差错代码的分配”中所描述的同样的规则。

ITU-T H.248.<xxx>建议书

网关控制协议：<xxx> 包

1 范围

<包的应用范围>

2 参考文献

下列 ITU-T 建议书和其他参考文献的条款，在本建议书中的引用而构成本建议书的条款。在出版时，所指出的版本是有效的。所有的建议书和其它参考文献均会得到修订，本建议书的使用者应查证是否有可能使用下列建议书或其它参考文献的最新版本。当前有效的 ITU-T 建议书清单定期出版。本建议书引用的文件自成一体时不具备建议书的地位。

- ITU-T H.248.1建议书 (<xx/xxxx>), 《网关控制协议：第<x>版》
- <其他参考文献>

3 术语和定义

本建议书采用下列术语和定义：

<术语和定义>

4 缩写

本建议书采用下列缩写：

<缩写>

5 惯例

<惯例>

6 <包的标题>

包的名称: <名称>
PackageID: <包的标识符: “文本”, 二进制<0x????>>
描述: <描述>
版本: <版本>
扩展: <扩展的包标识符和版本>

6.1 属性

6.1.1 <属性的标题>

属性名称: <名称>
PropertyID: <文本标识符, 二进制标识符 (0x????)>
描述: <描述>
类型: <类型>
可能的取值: <值>
默认值: <值>
定义于: <Local, Remote, LocalControl, TerminationState, ContextAttribute>
属性: <只读, 读/写>

6.2 事件

6.2.1 <事件的标题>

事件名称: <事件名称>
EventID: <(文本标识符), (二进制标识符 (0x????))>
描述: <描述>

6.2.1.1 EventsDescriptor参数

6.2.1.1.1 <参数的标题>

参数名称: <名称>
ParameterID: <(文本标识符), (二进制标识符 (0x????))>
描述: <描述>
类型: <类型>
可选项: <是/否>
可能的取值: <值>
默认值: <值>

6.2.1.2 ObservedEventsDescriptor参数

6.2.1.2.1 <参数的标题>

参数名称: <名称>
ParameterID: <(文本标识符), (二进制标识符 (0x????))>
描述: <描述>
类型: <类型>
可选项: <是/否>
可能的取值: <值>
默认值: <值>

6.3 信号

6.3.1 <信号的标题>

信号名称: <名称>
SignalID: <(文本标识符), (二进制标识符 (0x????))>
描述: <描述>
信号类型: <类型>
时长: <时长>

6.3.1.1 附加参数

6.3.1.1.1 <参数的标题>

参数名称: <名称>
ParameterID: <(文本标识符), (二进制标识符 (0x????))>
描述: <描述>
类型: <类型>
可选项: <是/否>
可能的取值: <值>
默认值: <值>

6.4 统计值

6.4.1 <统计值的标题>

统计值名称: <名称>
StatisticID: <(文本标识符), (二进制标识符 (0x????))>
描述: <描述>
类型: <类型>
可能的取值: <值>
级别: <Termination, Stream, Either>

6.5 差错代码

6.5.1 <差错代码的标题>

差错代码#: <编号>

名称: <名称>

定义: <定义>

差错描述符中的差错文本: <返回的差错文本>

注释: <注释>

6.6 程序

<与包有关的程序>

附 录 三

H.248协议子集定义模版

新的H.248协议子集应使用下列协议子集模版来定义。这是按照ITU建议书格式形成的。来自非ITU组织的编者至少应该使用模版的第6节中的结构。如果未使用该模版格式，则协议子集的编者应确保本附录中的标题和要点被其协议子集所涵盖。

下面结构中的标题代表了可以考虑作为可选项的条目。对于协议子集定义，某个H.248.1条目可以不被使用，尽管该条目在H.248.1中是必选项。这些条目也被包含在下面的结构中。未列出的条目被H.248.1协议认为是必选项。在下面的协议子集模版中，单元可以被定义为"optional（可选）"和"mandatory（必选）"。"Optional"意味着对于发送端或接收端都可以任选地在消息中包含该单元。如果接收实体接收到某个未实现的可选单元，按照6.2.3所述，它应该发送差错代码501("Not Implemented")。

如果编者要进一步阐明H.248.1的协议行为，则编者应该在下面的每个小节里提供书面描述。例如，如果Move命令（参见6.8.4）受限于某些终结点类型，这应该指明。

斜体字（楷体字）文本是要删除的。

括号里的<文本>是要填充的。

<协议子集的标题>

1 范围

<协议子集的应用范围>

2 参考文献

<参考文献>

3 术语和定义

本建议书采用下列术语和定义:

<术语和定义>

4 缩写

本建议书采用下列缩写:

<缩写>

5 惯例

<惯例>

6 协议子集描述

6.1 协议子集的标识

协议子集 名称:	<名称1-64字符>
版本:	<版本1-99>

协议子集的名称和版本在*ServiceChange*命令中发送。

6.2 摘要

<描述>

描述协议子集是怎样的。

6.3 网关控制协议版本

<版本号>

需要支持协议子集的最小*H.248*版本。这应基于基本句法支持而不是分配任意的版本。这与6.8.8中所述的*ServiceChangeVersion*相关。

6.4 连接模式

<用文本和图来描述>

描述在关联内所允许的终结点配置。

关联的最大数目:	<整数>
每个关联的终结点的最大数目:	<整数>
某个关联中允许的类型终结点组合:	<例如Context[a](IP,TDM), Context[b](TDM,AAL 2)等>

6.5 关联属性

关联属性	支持与否	支持的值
拓扑	<是/否>	见6.7.8
优先级指示符	<是/否>	<1-15>
紧急指示符	<是/否>	NA
IEPS指示符	<是/否>	NA
ContextAttribute描述符	<是/否>	如果“是”，则支持属性的细节见6.8.9。
ContextIDList 参数	<是/否>	NA

所支持的关联属性是AND/OR选择操作?

AND/OR 关联属性	<是/否>	<AND/OR/BOTH>
-------------	-------	---------------

6.6 终结点

6.6.1 终结点名称

<TerminationID的结构>

确定与物理、临时和复用终结点相关联的终结点标识。

6.6.2 复用终结点

支持复用终结点?	<是/否>
----------	-------

如果是，则：

支持的多种类型:	< H.221, H.223, H.226, V.76, N × 64K>
连至复用器的终结点的最大数目:	<整数>

使用复用的终结点? 如果是，进行描述。

6.7 描述符

6.7.1 Stream (流) 描述符

每类型终结点的stream的最大数目:	<TerminationType>	<整数>
---------------------	-------------------	------

如果多于1:

流的配置:	<说明允许的配置。是否允许一个以上的音频流? 等>
-------	---------------------------

6.7.1.1 LocalControl描述符

使用ReserveGroup和ReserveValue属性?

如果不是通用的, 则列出适当的终结点和流类型		终结点类型	流类型
使用ReserveGroup:	<是/否>	<类型>	<类型>
使用ReserveValue	<是/否>	<类型>	<类型>

使用哪个StreamMode值?

终结点类型	流类型	允许的StreamMode值
<类型>	<类型>	<SendOnly, RecvOnly, SendRecv, Loopback>

6.7.2 事件描述符

在本协议子集中所有包含的事件可以在任意终结点/流上设置[下列情况例外]。

注 — 在[]内的文本是可选的, 而且仅仅在存在这种例外情况下被包含。

事件 settable on 终结点类型 and 流类型:	<是/否>		
如果是	EventID	终结点类型	流类型
	<事件名称和标识, 例如通用差错事件 (g/cause, 0x0001/0x0001)>	<类型>	<流类型, 例如 Audio/Video/Data或 StreamID>

使用 EventBuffer Contro?

使用EventBuffer控制:	<是/否>
------------------	-------

使用事件KeepActive?

使用事件 KeepActive:	<是/否>
------------------	-------

在事件中使用嵌套?

在事件描述符中嵌套事件:	<是/否>
在事件描述符中嵌套信号:	<是/否>

支持 Regulated EmbeddedEvents?

触发调整嵌套事件的条件:	<None /指定单个事件>
--------------	----------------

使用ResetEventsDescriptor Flag?

和事件一起使用ResetEventsDescriptor:	<ALL / None / 指定单个事件>
-------------------------------	-----------------------

支持什么样的Notification Behaviour?

NotifyImmediate:	<ALL Events / None / 指定单个事件s>
NotifyRegulated:	<ALL事件 / None / 指定单个事件s>
NeverNotify:	<ALL事件 / None / 指定单个事件s>

6.7.3 EventBuffer描述符

支持该参数?

使用EventBuffer描述符:	<是/否>	
如果是	EventIDs	<事件名称和标识, 例如通用差错事件(g/cause, 0x0001/0x0001)或ALL>

6.7.4 信号描述符

在本协议子集中包含的所有信号可以在任意终结点/流上设置[下列情况例外]。

注 — 在[]内的文本是可选的, 而且仅仅在存在这种例外情况下被包含。

信号的设置依靠终结点类型和流类型:	<是/否> 注 — "No"意味着所有的信号可以在任意终结点或流上播放。如果“是”, 下面未列出的任何信号都可以在任意终结点或流上播放。		
如果是	SignalID	终结点类型	流类型 / ID
	<信号名称和标识, 例如 Playtone (tonegen/pt, 0x0003/0x0001)>	<类型>	<流类型 例如 Audio/Video/Data或 StreamID>

支持信号列表? 如果支持, 每个终结点/流类型上的每个列表所支持的最大数目的信号是多少?

支持的信号列表:	<是/否>	
如果是	终结点类型支持列表:	<类型/ALL>
	流类型支持列表:	<类型/ALL>
	的最大数目: 信号至信号列表:	<整数>
	支持的信号间延迟参数:	<是/否>

支持替换的信号类型和持续时间?

支持信号类型和时长:	<是/否>	
如果是	SignalID	替换的类型或时长
	<信号名称和标识, 例如 Playtone (tonegen/pt, 0x0003/0x0001)或"ALL">	<类型、时长、二者>

支持 *Signal* 方向?

支持信号方向:	<是/否>
---------	-------

支持 "notifyCompletion"? 支持什么类型? *RequestID* 与 "NotifyCompletion" 一起使用?

支持NotifyCompletion:	<是/否>	
如果是	SignalID	支持的完成类型
	<信号名称和标识, 例如Playtone (tonegen/pt, 0x0003/0x0001)或 ALL>	<ALL, TO, EV, ED, NC, PI>
支持的RequestID参数	<是/否>	

多个信号能够同时播放?

同时播放信号:	<是/否>	
如果是	可以同时播放的SignalID	<信号名称和标识, 例如Playtone (tonegen/pt, 0x0003/0x0001)或ALL>

"KeepActive" 支持用于信号?

在信号上使用KeepActive:	<是/否>
-------------------	-------

6.7.5 DigitMap描述符

支持 *DigitMaps*? 如果支持, 描述名称、结构和定时器。

支持DigitMaps:	<是/否>		
如果是	DigitMap名称	结构	定时器
	<名称>	<描述>	<定时器>

6.7.6 统计值描述符

支持终结点、流或二者的统计值?

支持的统计值:	<终结点/ 流/ 二者>
---------	--------------

统计值汇报否?

在Subtract中报告统计值:	<是/否>	
如果是	报告StatisticID:	<Statistics名称和标识, 例如Packets Sent (rtp/ps, 0x000c/0x0004)或ALL>

6.7.7 ObservedEvents描述符

支持检测时间?

支持检测时间:	<是/否>
---------	-------

6.7.8 Topology描述符

如果使用该参数，允许怎样的设置？

允许的三种方式：	<(T1、T2、单向) 等>
----------	----------------

6.7.9 Error描述符

支持H.248.8和包定义的哪些差错代码？

由MGC发送的差错代码：

支持的 H.248.8差错代码：	<所有H.248.8中差错代码, 列出当差错代码>
支持的在包中定义的差错代码：	对于差错代码列表, 参见6.14.x<参见下面6.14中适当的小节>

由MG发送的差错代码：

支持的 H.248.8差错代码：	<所有H.248.8中差错代码, 列出当差错代码>
支持的在包中定义的差错代码：	对于差错代码列表, 参见6.14.x<参见下面6.14中适当的小节>

6.8 API命令

注 — 假定在任何命令响应中返回Error描述符。

6.8.1 Add

哪些描述符可以使用在Add应答中？

Add应答所使用的描述符：	<Media, Mux, Events, EventBuffer, Signals, DigitMap, Audit, Statistics>
---------------	---

哪些描述符可以使用在Modify请求中？

Modify 请求所使用的描述符：	<Media, Mux, Events, EventBuffer, Signals, DigitMap, Audit>
-------------------	---

哪些描述符可以使用在Modify请求中？

Modify 应答所使用的描述符：	<Media, Mux, Events, EventBuffer, Signals, DigitMap, Audit>
-------------------	---

6.8.3 Subtract

Audi描述符可以使用在Subtract 请求中？

Subtract 请求所使用的描述符：	<Audit>
---------------------	---------

Statistics描述符可以使用在Subtract 应答中?

Subtract应答所使用的描述符:	<Statistics>
--------------------	--------------

6.8.4 Move

使用Move命令?某些关联配置可能不使用该命令。

使用Move命令:	<Yes/No>
-----------	----------

如果使用:

Move 请求所使用的描述符:	<Media, Mux, Events, EventBuffer, Signals, DigitMap, Audit, Statistics>
Move 应答所使用的描述符:	<Media, Mux, Events, EventBuffer, Signals, DigitMap, Audit, Statistics>

6.8.5 AuditValue

哪些描述符和/或单个属性、信号、事件或统计可以被审计?

审计的属性:	<Property 名称和标识, 例如的最大数目: contexts (Root/maxNumberOfContexts, 0x0002/0x0001), ALL或None>	<描述符: Local, Remote, LocalControl, TerminationState>
审计的统计值:	<Statistics 名称和标识, 例如发送的包数 (rtp/ps, 0x000c/0x0004), ALL或None>	
审计的信号:	<信号名称和标识, 例如Playtone (tonegen/pt, 0x0003/0x0001), ALL或None>	
审计的事件:	<事件名称和标识, 例如通用差错事件(g/cause, 0x0001/0x0001), ALL或None>	
有可能审计包:	包描述符可以被审计? <是/否>	

6.8.6 AuditCapabilities

哪些描述符和/或单个属性、信号、事件或统计可以被审计?

审计的属性:	<属性名称和标识, 例如的最大数目: contexts (Root/maxNumberOfContexts, 0x0002/0x0001), ALL或None>	<描述符: Local, Remote, LocalControl, TerminationState>
审计的统计值:	<Statistics 名称和标识, 例如Packets Sent (rtp/ps, 0x000c/0x0004), ALL或None>	
审计的信号:	<信号名称和标识, 例如Playtone (tonegen/pt, 0x0003/0x0001), ALL或None>	
审计的事件:	<事件名称和标识, 例如通用差错事件(g/cause, 0x0001/0x0001), ALL或None>	

有限范围内的审计是否可能?

用于有限范围的审计的属性/ ContextAttributes:	<None / ALL / 单个指定>
----------------------------------	---------------------

6.8.7 Notify

哪些描述符可以使用在Modify命令中?

Notify请求和应答所使用的描述符:	<ObservedEvents, Error>
---------------------	-------------------------

6.8.8 ServiceChange

支持哪些ServiceChangeMethods和Reasons?

由MGC发送的ServiceChangeMethods 和 ServiceChangeReasons:

所支持的ServiceChangeMethods:	所支持的ServiceChangeReasons:
<Graceful, Forced, Restart, Handoff, Failover, ALL, 其他?>	< 900-920 >

由MG发送的ServiceChangeMethods 和 ServiceChangeReasons:

所支持的ServiceChangeMethods:	所支持的ServiceChangeReasons:
<Graceful, Forced, Restart, Disconnected, Handoff, Failover, ALL, Other?>	< 900-920 >

使用ServiceChangeAddress否?

使用ServiceChangeAddress:	<是/否>
-------------------------	-------

使用 ServiceChangeDelay 否?

ServiceChangeDelay used:	<是/否>	
如果是	有效时间段:	<0-x> ms

使用ServiceChangeIncomplete Flag 否?

使用ServiceChange Incomplete Flag:	<是/否>
----------------------------------	-------

ServiceChangeVersion使用哪个版本的ITU-T H.248.1建议书? 这里的最低值是6.3中所定义的最小版本。

在ServiceChangeVersion中使用的版本:	<1, 2, 3>
------------------------------	-----------

按照ITU-T H.248.18建议书, 是否支持多个协议子集?

按照ITU-T H.248.18建议书进行协议子集协商:	<是/否>
------------------------------	-------

6.8.9 操作和审计关联属性

哪些关联属性可以被操作和/或审计?

可操作的关联属性:	<Topology, Emergency, Priority, IEPS指示符, ContextAttribute描述符(属性名称列表), ALL>
审计的关联属性:	<Topology, Emergency, Priority, IEPS指示符, ContextAttribute描述符(属性名称列表), ALL>

6.9 通用的命令句法和编码形式

指定协议子集所支持的编码形式事务处理。

所支持的编码形式:	<文本和二进制、二进制、文本>
-----------	-----------------

6.10 事务

每个消息中的`TransactionRequest/TransactionReply`的最大数目是多少?

每个消息中 <code>TransactionRequests / TransactionReply / TransResponseAcks / Segment Reply</code> 的最大数目:	<整数>
--	------

每个`TransactionRequest`中命令的最大数目是多少?

每个 <code>TransactionRequest</code> 中命令的最大数目:	<整数>
--	------

每个`TransactionReply`中命令的最大数目是多少?

每个 <code>TransactionReply</code> 中命令的最大数目:	<整数>
--	------

命令是否能够标记为“可选”? 请描述。

能够标记为“可选”的命令:	<Add, Modify, Move, Subtract, Auditvalue, Auditcapability, Servicechange, All, None>
---------------	--

指定事务定时器的值。

事务定时器:	值
<code>normalMGExecutionTime</code>	<整数或“预设”>
<code>normalMGCEExecutionTime</code>	<整数或“预设”>
<code>MGOriantedPendingLimit</code>	<整数或“预设”>
<code>MGCOriantedPendingLimit</code>	<整数或“预设”>
<code>MGProvisionalResponseTimerValue</code>	<整数或“预设”>
<code>MGCProvisionalResponseTimerValue</code>	<整数或“预设”>

6.11 消息

`MGC/MG`命名惯例: `MID`表示了`MGC/MG`名字的关联。

<描述>

指示每个消息中的事务的最大数目(如果不重要, 这可能被省略)。

每个消息中事务的最大数目:	<整数>
---------------	------

6.12 传输

规定由协议子集所支持的H.248的系列传输。

支持传送:	<UDP, TCP, SCTP, MTP3B, SSCOP/AAL 5, ALF/AAL 5>
-------	---

支持分段否? 如果支持, 使用什么方法?

支持分段:	<否, 传送中固有的, H.248分段>
-------	----------------------

是否使用控制联系监视 (参见11.6)? 如果使用, 采用什么方法?

支持控制联系监视:	<No, Inherent in Transport, Empty AuditValue on Root, H.248.14>
-----------	---

6.13 安全

指定所使用的安全机制。

支持安全:	<无, 临时AH方案, IPSec>
-------	--------------------

6.14 包

指定在本协议子集中所支持的包。

强制性要求: 指定在本协议子集中必须支持的包。

必选的包:		
包的名称	PackageID	版本
<名称>	<xxxx, (0x00xx)>	<1, 2, 3, ...>

可选项: 指定在本协议子集中可选支持的包。

可选的包			
包的名称	PackageID	版本	支持依据:
<名称>	<xxx, 0x00??>	<1, 2, 3, ...>	<描述>

包的使用信息

该表指定了以上的包将怎样被使用。例如:

- 该表列出了属性/信号事件/统计是可选或必选;
 - 如果属性/信号事件/统计的值是预置的, 则应被指定 (例如, H.248.7通告的名称和循环数目);
- 指定那些作为预置的属性值。

包的使用信息

6.14.x <包的名称>

属性	必选项/可选项	用于下列命令:	支持的参数值:	预设的参数值:
<名称和标识, 例如发送的包数 (rtp/ps, 0x000c/0x0004), ALL或None>	<M/O>	<ADD, MOD, MOVE, AUDITVALUE, AUDITCAP>	<值 / ALL >	<值 / 不适用>
信号	必选项/可选项	用于下列命令:		时长预设的参数值:
<名称和标识>	<M/O>	<ADD, MOD, MOVE, AUDITVALUE, AUDITCAP>		<值 / 不适用>
	Signal参数	必选项/可选项	支持的参数值:	时长预设的参数值:
	<名称和标识>	<M/O>	<值 / ALL>	<值 / 不适用>
事件	必选项/可选项	用于下列命令:		
<名称和标识>	<M/O>	<ADD, MOD, MOVE, NOTIFY, AUDITVALUE, AUDITCAP>		
	事件参数	必选项/可选项	支持的参数值:	预设的参数值:
	<名称和标识>	<M/O>	<值 / ALL>	<值 / 不适用>
	Observed事件参数	必选项/可选项	支持的参数值:	预设的参数值:
	<名称和标识>	<M/O>	<值 / ALL>	<值 / 不适用>
统计值	必选项/可选项	用于下列命令:	支持的参数值:	
<名称和标识>	<M/O>	<ADD, MOD, MOVE, SUBTRACT, AUDITVALUE, AUDITCAP>	<值 / ALL >	
差错代码	必选项/可选项			
<编号>	<M/O>			

另外的限制可以按照用户需要制表列出。

6.15 SDP和附件C信息单元的必选支持

规定哪些SDP属性和附件C信息单元应必须被支持。

所支持的附件C和SDP信息单元:		
信息单元	附件C支持	SDP支持
<名称>	<附加C属性>	<描述>

6.16 SDP和附件C信息单元的可选支持

规定哪些SDP属性和附件C信息单元可选被支持。

可选的附件C和SDP信息单元:			
信息单元	附件C支持	SDP支持	支持依据:
<名称>	<附件C属性>	<描述>	<描述>

6.17 程序

规定与协议子集相关的程序。

推荐程序采取下列格式:

6.17.1 <程序的名称>

当需要程序<名称>时, 应发起下列步骤。

从<MGC/MG>至<MG/MGC>发送一个<ADD.req, MOD.req, MOV.req, SUB.req, AuditValue.req, AuditCapability.req, ServiceChange.req, Notify.req>命令, 该命令带有下列信息。

<用适当的格式插入信息。例如, 描述符、属性、信号、事件、统计名称或根据信息单位提取实际参数。>

一旦接收到该命令, 则<MG/MGC>应:

- <表明所需的行动>

一旦完成处理命令(1), 从<MG/MGC>至<MGC/MG>发送一个<ADD.reply, MOD.reply, MOV.reply, SUB.reply, AuditValue.reply, AuditCapability.reply, ServiceChange.reply, Notify.reply>命令, 该命令带有下列信息。

<用适当的格式插入信息。例如, 描述符、属性、信号、事件、统计名称或根据信息单位提取实际参数。>

ITU-T 系列建议书

A系列	ITU-T工作的组织
D系列	一般资费原则
E系列	综合网络运行、电话业务、业务运行和人为因素
F系列	非话电信业务
G系列	传输系统和媒质、数字系统和网络
H系列	视听及多媒体系统
I系列	综合业务数字网
J系列	有线网络和电视、声音节目及其他多媒体信号的传输
K系列	干扰的防护
L系列	电缆和外部设备其他组件的结构、安装和保护
M系列	电信管理，包括TMN和网络维护
N系列	维护：国际声音节目和电视传输电路
O系列	测量设备的技术规范
P系列	电话传输质量、电话设施及本地线路网络
Q系列	交换和信令
R系列	电报传输
S系列	电报业务终端设备
T系列	远程信息处理业务的终端设备
U系列	电报交换
V系列	电话网上的数据通信
X系列	数据网、开放系统通信和安全性
Y系列	全球信息基础设施、互联网协议问题和下一代网络
Z系列	电信系统的使用的语言和一般性软件情况