

ICS 33.040

M 32



中华人民共和国通信行业标准

YD/T 2422-2012

基于 H.248 的媒体网关控制协议测试方法

Testing method for media gateway control protocol based on H.248

2012-12-28 发布

2013-03-01 实施

中华人民共和国工业和信息化部 发布

目 次

前 言..... II

1 范围..... 1

2 缩略语..... 1

3 H.248 协议结构..... 1

4 测试结构..... 1

5 H.248 MGC 测试项目..... 2

 5.1 ServiceChange..... 2

 5.2 各种包的定义和应用..... 22

 5.3 描述符的定义和应用..... 45

 5.4 通配符的应用..... 64

 5.5 定时器的应用..... 67

 5.6 错误码..... 72

 5.7 杂项..... 76

 5.8 安全..... 82

6 H.248 MG 测试项目..... 83

 6.1 ServiceChange..... 83

 6.2 各种包的定义和应用..... 115

 6.3 描述符的定义和应用..... 115

 6.4 通配符的应用..... 115

 6.5 定时器的应用..... 115

 6.6 错误码..... 116

 6.7 杂项..... 128

 6.8 安全..... 129

前 言

本标准是媒体网关控制协议系列标准之一。该系列标准结构及名称如下：

——YD/T 1292-2011 基于 H.248 的媒体网关控制协议技术要求；

——YD/T 2422-2012 基于 H.248 的媒体网关控制协议测试方法。

本标准按照 GB/T 1.1-2009 规则起草。

本部分由中国通信标准化协会提出并归口。

本部分起草单位：工业和信息化部电信研究院、华为技术有限公司。

本部分主要起草人：蒋晓琳、武 静、林杨波。

基于 H.248 的媒体网关控制协议测试方法

1 范围

本标准规定了媒体网关和媒体网关控制器之间、媒体网关和软交换之间进行通信时的协议测试要求。
本标准适用于媒体网关控制器（MGC）/软交换设备和媒体网关（MG）中的 MEGACO/H.248 协议测试。本标准不适用于性能测试。

2 缩略语

下列缩略语适用于本文件。

ATS	Abstract Test Suite	抽象测试套
CID	Context Id	关联ID
IUT	Implementation Under Test	待测实现
MG	Media Gateway	媒体网关
MGC	Media Gateway Controller	媒体网关控制器
PDU	Protocol Data Unit	协议数据单元
SUT	System under test	待测系统
TID	Termination ID	终结点ID

3 H.248 协议结构

测试环境应实现图 1 所示的 H.248 协议栈。

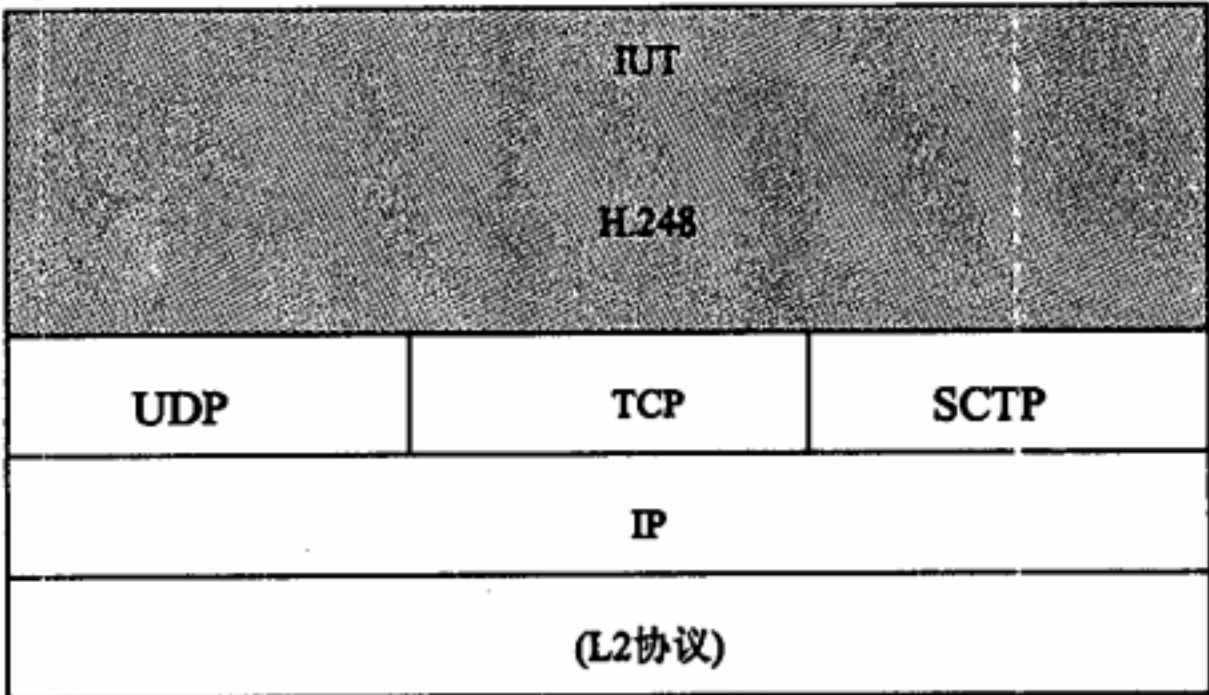
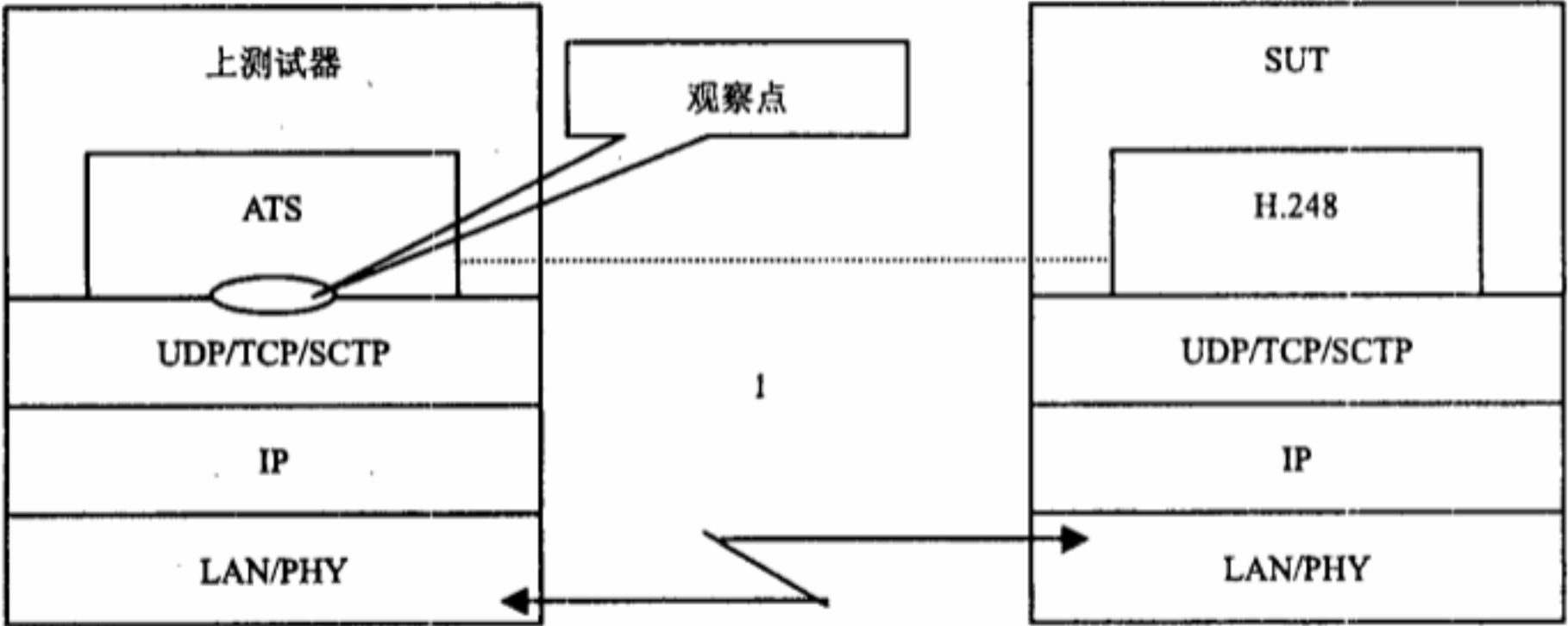


图 1 H.248 协议栈

4 测试结构

本标准测试内容分为 MG 网关(Media Gateway)测试和 MGC 网关控制器(Media Gateway Controller)测试两部分，均采用图 2 所示的测试结构。



注：根据测试需要，SUT 待测系统（System Under Test）在实际测试中可以由 MG 或者 MGC 来充当。

图 2 测试结构

在下列测试例中，Term 表示终结点（Termination），MGC 为媒体网关控制器，MG 为媒体网关。在某些特定测试例中，MG 分别适用于 TG 中继媒体网关类设备（Trunk Gateway）或者 RG 驻地媒体网关类设备（Residential Gateway），包括综合接入媒体网关和综合接入设备等。

5 H.248 MGC 测试项目

5.1 ServiceChange

测试编号：MGC_SC_1
测试项目：MGC 响应 MG 的注册请求
预置条件： MG 加电启动向 MGC 发送注册请求
测试过程： 步骤 1 MG 向 MGC 发送注册请求，ServiceChange 命令中的 method=Restart，Termination=Root。 步骤 2 MGC 接受注册，返回 Reply，其中包含 MGCID。 测试消息流程如下所示： <div><div>MG</div><div>MG</div><div> ----- ServiceChange ROOT Method=Restart -----> </div><div> <----- Reply ----- </div></div>
判决准则： MG 应能够在 MGC 上成功注册
测试说明： 无

测试编号：MGC_SC_2
测试项目：冷启动
预置条件： MG向主MGC注册时，主MGC不接受，返回备用MGC的地址，MG再根据所返回地址向备用MGC注册
测试过程： 步骤1 整机连接好电源，使媒体网关与MGC处于断开状态。 步骤2 将媒体网关与主媒体网关控制器相连接，媒体网关向主媒体网关控制器注册，主媒体网关控制器不接受媒体网关的注册，并在reply中返回备用媒体网关控制器的地址。 步骤3 媒体网关继续向备用媒体网关控制器进行注册，备用媒体网关控制器接受媒体网关的注册，并应返回正确的应答。 测试消息流程如下所示： <div><div>MG</div><div>主MGC</div><div> ----- ServiceChange ROOT Method= Restart -----> </div><div> <----- Reply (MGCId) ----- </div><div>MG</div><div>备用MGC</div><div> ----- ServiceChange ROOT Method= Restart -----> </div><div> <----- Reply ----- </div></div>
判决准则： MG应能够在备用MGC上成功注册
测试说明： 无

测试编号: MGC_SC_3
测试项目: MGC 响应 MG 的注销请求
预置条件: 在媒体网关和 MGC 上配置相应的数据
测试过程: 步骤 1 媒体网关发送的 ServiceChange 命令, ServiceChange 中的 ServiceChangeMethod 设置为 Graceful/Forced; 步骤 2 在正常情况下, 媒体网关控制器应返回正确的应答。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div> ----- ServiceChange ROOT Method=Graceful/Force -----> </div><div> <----- Reply ----- </div></div>
判决准则: MG 应能够在备用 MGC 上成功注销
测试说明: 无

测试编号: MGC_SC_4
测试项目: MGC 支持比 MG 低的软件版本
预置条件: MGC 仅支持 H.248 版本 2
测试过程: 步骤 1 MG 以 ServiceChange 命令向 MGC 注册, 其中带了所支持的版本信息; 步骤 2 MGC 发现 MG 的版本号比自己高, 返回自己所支持的版本号。 测试消息流程如下所示: <div><div>MG</div><div>主MGC</div><div> ----- ServiceChangeTermA (ServiceChangeVersion2) -----> </div><div> <----- Reply (ServiceChangeVersion 步骤 1) ----- </div></div>
判决准则: MGC 返回自己所支持的版本号, MGC 协商将使用的 H.248 版本
测试说明: 无

测试编号: MGC_SC_5
测试项目: MG 请求渐进退出服务
预置条件: MG 已经在 MGC 上注册成功且通过 TermA 建立一个呼叫
测试过程: 步骤 1 MG 发 ServiceChange 命令, 方法为 Graceful, 并设 delay, 分为 delay≠null 和 delay=null 两种情况; 步骤 2 在 delay 时间后, MGC 发出删除终结点 TermA 命令, TermA 退出服务; 步骤 3 当 delay=null, 在呼叫结束后, MGC 发出删除终结点 TermA 命令, TermA 退出服务。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>ServiceChange TermA Method=graceful Delay=x ServiceChangeReason=905</div><div>Reply</div><div>在 x 秒后, TermA 退出业务</div><div>Subtract TermA</div><div>Reply</div></div>
判决准则: delay 时间后在 MG 中的 TermA 处于空关联中
测试说明: 无

测试编号: MGC_SC_6	
测试项目: MG 请求强制退出服务	
预置条件: MG 已经在 MGC 上注册成功且通过 TermA 建立一个呼叫	
<p>测试过程:</p> <p>步骤 1 MG 产生 ServiceChange 命令, 方法为 Forced。分为指定关联和不指定关联 (contextID=null) 两种情况;</p> <p>步骤 2 MGC 立即发出删除终结点 TermA 命令, TermA 退出服务。</p> <p>测试消息流程如下所示:</p> <div></div>	
<p>判决准则:</p> <p>MGC 上看不到该 MG TermA 的信息</p>	
<p>测试说明:</p> <p>无</p>	

测试编号: MGC_SC_7
测试项目: MG 和 MGC 失去正常通信后又恢复通信
预置条件: MG 已经在 MGC 上注册成功, 在过段时间后断开 MG 与 MGC 之间的通信然后恢复
测试过程: 步骤 1 MG 失去和 MGC 的正常通信后又恢复; 步骤 2 MG 通过 ServiceChange 向 MGC 报告失去联系, method=Disconnected; 步骤 3 MGC 向 MG 发出 Audit 消息请求, 要求获得 MG 当前的资源状态信息。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>MG 失去与 MGC 的通信 ServiceChange ROOT Method=disconn</div><div>Reply</div><div>Audit</div><div>Reply</div></div>
判决准则: MGC 能够获得该 MG 的资源状态信息
测试说明: 无

测试编号: MGC_SC_8
测试项目: 启用备用 MG
预置条件: 两个备用 MG 已经在 MGC 上注册成功并配置为冗余备份
测试过程: 步骤 1 MGC 收到 MG 的 ServiceChange 消息, 其中 method=failover; 步骤 2 MGC 将备用 MG 作为 Active 的 MG. 测试消息流程如下所示: <div><div>MG步骤1</div><div>MGC</div><div> -----ServiceChange ROOT Method= failover-----> </div><div> <-----Reply----- </div></div>
判决准则: 该 MG 上的业务仍然能够被保持
测试说明: 无

测试编号: MGC_SC_9
测试项目: MGC 切换
预置条件: 配置两个备用 MGC
测试过程: 步骤 1 在运行期间, MGC 发生切换; 步骤 2 主 MGC 向 MG 发 ServiceChange 命令中 method=Handoff, 并携带要求 MG 重新注册的备用 MGC 的地址。 测试消息流程如下所示: <pre>sequenceDiagram participant MG participant MGC1 participant MGC2 MG->>MGC1: ServiceChange ROOT Method=Handoff MGC1-->>MG: Reply MG->>MGC2: ServiceChange ROOT Method=Handoff MGC2-->>MG: Reply</pre>
判决准则: 该 MG 上的业务仍然能够被保持
测试说明: 无

测试编号: MGC_SC_10
测试项目: MGC 强制 MG 的终结点退出服务
预置条件: MG 已经在 MGC 上注册成功
测试过程: 步骤 1 MGC 产生 ServiceChange 命令, 方法为 Forced, 终结点在 NULL 关联; 步骤 2 MG 应答, TermA 立即退出服务。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>ServiceChange TermA Method= Forced</div><div>Reply</div></div>
判决准则: 在 MG 上可以查询到 TermA 已经终止业务
测试说明: 无

测试编号: MGC_SC_11
测试项目: MGC 要求 MG 的终结点返回服务
预置条件: MG 已经脱离业务
测试过程: 步骤 1 MGC 产生 ServiceChange 命令, 方法为 Restart, 终结点在 NULL 关联; 步骤 2 MG 应答, TermA 立即返回服务。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>ServiceChange TermA Method=Restart</div><div>Reply</div></div>
判决准则: 在 MG 上可以查询到 TermA 已经进入业务运行态
测试说明: 无

测试编号: MGC_SC_12
测试项目: MG 终结点请求重新进入服务
预置条件: MG 终结点已经脱离业务
测试过程: 步骤 1 MG 产生 ServiceChange 命令, 方法为 Restart, 终结点在 NULL 关联; 步骤 2 MGC 应答, TermA 立即进入服务。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>ServiceChange TermA Method=Restart</div><div>Reply</div></div>
判决准则: 在 MG 上可以查询到 TermA 已经进入业务运行态
测试说明: 无

测试编号: MGC_SC_13
测试项目: 协议版本协商
预置条件: MGC 仅支持 H.248 版本 2, MG 支持 H.248 版本 3
测试过程: 步骤 1 MG 以 ServiceChange 命令向 MGC 注册, 版本信息为 V3; 步骤 2 MGC 发现 MG 的版本号比自己高, 返回自己所支持的版本号。 测试消息流程如下所示: <div><div>MG</div><div>主MGC</div><div> -----ServiceChangeROOT (Version3)-----> </div><div> <-----Reply (ServiceChangeVersion2)----- </div><div> -----ServiceChangeROOT (Version3)-----> </div><div> <-----Reply (ServiceChangeVersion1)----- </div></div>
判决准则: MGC 返回自己所支持的版本号, MG 根据协商结果使用的 H.248 版本
测试说明: 无

测试编号: MGC_SC_14
测试项目: MGC 响应 MG 的注册请求, 注册请求中携带 SIC 参数
预置条件: MG 加电启动向 MGC 发送注册请求, MG 上 A1 端点故障
测试过程: 步骤 1 MG 向 MGC 发送注册请求, ServiceChange 命令中的 method=Restart, Termination=Root, 且 SIC 参数值为 ON。 步骤 2 MGC 接受注册, 返回 Reply, 其中包含 MGCID。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>ServiceChange ROOT Method=Restart SIC=ON</div><div>Reply</div><div>ServiceChange TermTermA Method=Restart A1 故障</div><div>Reply</div></div>
判决准则: MG 发送 SC 消息, 携带 SIC 参数, 此时 MGC 等待故障信息上报, MG 再次发送 SC 消息, 携带故障端点信息, MGC 收到后完成注册
测试说明: 无

测试编号: MGC_SC_15
测试项目: MGC 响应 MG 的注册请求, 注册请求中携带 SIC 参数
预置条件: MG 加电启动向 MGC 发送注册请求, MG 上 A1 端点故障
测试过程: 步骤 1 MG 向 MGC 发送注册请求, ServiceChange 命令中的 method=Restart, Termination=Root, 且 SIC 参数值为 ON 步骤 2 MGC 接受注册, 返回 Reply, 其中包含 MGCID. 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div> -----ServiceChange ROOT Method=Restart SIC=ON-----> </div><div> <-----Reply----- </div><div> -----ServiceChange ROOT Method= Graceful -----> </div><div> <-----503 Service Unavailable----- </div></div>
判决准则: MGC 在故障端点收集状态时, 只处理 Forced 和 Restart 两种 method, 其余回错误码 503-Service Unavailable
测试说明: 无

测试编号: MGC_SC_16	
测试项目: MGC 发起的 MG 重注册	
预置条件: 配置备用 MGC	
测试过程: 步骤 1 在运行期间,主 MGC 向 MG 发 ServiceChange 命令中 method=Handoff,但是携带的注册 MGC 的地址仍然为主 MGCID, ServiceChangeReason =903 (“MGC Directed Change”). 步骤 2 MG 发起 ServiceChange 命令中 method=Handoff,但是携带的注册 MGC 的地址仍然为主 MGCID, ServiceChangeReason =903 (“MGC Directed Change”). 测试消息流程如下所示:	
<pre>sequenceDiagram participant MG participant MGC1 MG->>MGC1: ServiceChange ROOT Method=Handoff MGC1-->>MG: Reply MG->>MGC1: ServiceChange ROOT Method=Handoff MGC1-->>MG: Reply</pre>	
判决准则: MG 发起重注册,注册成功	
测试说明: 无	

测试编号: MGC_SC_17
测试项目: Service Restoration
预置条件: MG 已经成功注册在 MGC 上
测试过程: 步骤 1 在运行期间, 主 MGC 向 MG 发 ServiceChange 命令中 method= Restart, ServiceChangeReason=900 (“Service Restored”). 步骤 2 MG 收到 MGC 的 ServiceChange 命令, 也通过 ServiceChange 命令发起 Service Restored, ServiceChangeMethod= “Restart”. 测试消息流程如下所示: <div><div>MG</div><div>MGC1</div><div><div></div><div>ServiceChange ROOT Method= Restart</div><div>Reply</div><div>ServiceChange ROOT Method= Restart</div><div>Reply</div></div></div>
判决准则: MGC 发起 Service Restoration, 业务成功
测试说明: 无

测试编号: MGC_SC_18
测试项目: MGC Initiated Service Cancellation (Root Termination)
预置条件: MG 已经成功注册在 MGC 上
测试过程: 步骤 1 在运行期间, 主 MGC 向 MG 发 ServiceChange 命令中 method= Graceful, ServiceChangeReason =905 (“Termination taken out of service”). 步骤 2 在 ServiceChangeDelay 时间内, MGC 取消先前的 MG 注销, MGC 发送 ServiceChange 命令, 其中 ServiceChangeMethod = “Restart”, ServiceChangeReason =918 (“Cancel Graceful”). 测试消息流程如下所示: <div><div>MG</div><div>MGC1</div><div>ServiceChange ROOT Method= Graceful</div><div>Reply</div><div>ServiceChange ROOT Method= Restart</div><div>Reply</div></div>
判决准则: MGC 按照测试过程发起 SERVICE CHANGE, MG 保持业务状态为 IN SERVICE
测试说明: 无

测试编号: MGC_SC_19
测试项目: MGC Initiated Service Cancellation (Root 终结点)
预置条件: MG 已经成功注册在 MGC 上
测试过程: 步骤 1 在运行期间, 主 MGC 向 MG 发 ServiceChange 命令中 method= Graceful, ServiceChangeReason =905 (“Termination taken out of service”). 步骤 2 在 ServiceChangeDelay 时间已经失效后, MGC 取消先前的 MG 注销, MGC 发送 ServiceChange 命令, 其中 ServiceChangeMethod = “Restart”, ServiceChangeReason =918 (“Cancel Graceful”). 步骤 3 MG 回复错误码 502 (“Not Ready”). 测试消息流程如下所示: <div><div>MG</div><div>MGC1</div><div><div></div><div>ServiceChange ROOT Method= Graceful</div><div>Reply</div><div>ServiceChange ROOT Method= Restart</div><div>502 (Not Ready)</div></div></div>
判决准则: MGC 按照测试过程发起 SERVICE CHANGE, MG 退出服务, 回复错误原因 502
测试说明: 无

测试编号: MGC_SC_20	
测试项目: MGC Initiated Service Cancellation	
预置条件: MG 已经在 MGC 上注册成功且通过 TermA 建立一个呼叫	
测试过程: 步骤 1 MG 发 ServiceChange 命令, method= graceful, 并设 delay。 步骤 2 MG 发出 service change 命令, method= restart, ServiceChangeReason =918 (“Cancel Graceful”), 端点恢复服务。 测试消息流程如下所示:	
<div>MG<div>ServiceChange TermA Method=graceful Delay=x ServiceChangeReason=905</div><div>Reply</div><div>ServiceChange TermA Method=restart; ServiceChangeReason=918</div><div>Reply</div></div> <div>MGC</div>	
判决准则: MGC/MG 按照测试过程发起 SERVICE CHANGE, 在 delay 时间前, MG 发起取消注销, 此时端点保持 IN SERVICE 状态, 在 delay 时间前, MG 发起取消注销, 端点恢复 IN SERVICE 状态	
测试说明: 无	

5.2 各种包的定义和应用

5.2.1 Generic 包

测试编号: MGC_Generic-Package_1
测试项目: Generic 包正常释放
预置条件: MG 已经成功的建立呼叫
测试过程: 步骤1 MGC向MG发送AuditValue和AuditCapability消息。 步骤2 MG能够正确处理, 在Reply中返回Generalcause= “NR” , 进行正常的呼叫释放。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>Notify(g/cause {Generalcause="NR"})</div><div>Notify Resp</div></div>
判决准则: MGC 收到 MG 的响应之后, 能够释放资源

测试编号: MGC_Generic-Package_2
测试项目: Generic 包——没有可用资源
预置条件: MGC 向 MG 发起呼叫
测试过程: 步骤1 MGC向MG发送AuditVlue和AuditCapability消息。 步骤2 MG 因找不到可用资源, 在 Reply 中返回 Generalcause= “UR” 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>Notify(g/cause{Generalcause="UR"})</div><div>Reply</div></div>
判决准则: MGC 收到 MG 的响应之后, 不能建立呼叫
测试说明: 该测试项中测试议作为 TG

测试编号: MGC_Generic-Package_3
测试项目: Generic 包——暂时故障
预置条件: MGC 向 MG 发起呼叫
测试过程: 步骤 1 MG 出现暂时故障, MG 向 MGC 发 Notify 消息, Generalcause= “FT”。 步骤 2 MGC 能正确处理该错误事件。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>Notify(g/cause{Generalcause="FT"})</div><div>Reply</div></div>
判决准则: 呼叫未能建立
测试说明: 该测试项中测试议作为 TG

测试编号: MGC_Generic-Package_4
测试项目: Generic 包——永久故障
预置条件: MGC 向 MG 发起呼叫
测试过程: 步骤 1 MG 出现永久故障, MG 向 MGC 发 Notify 消息, Generalcause= “FP”。 步骤 2 MGC 能正确处理该错误事件的。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>Notify(g/cause {Generalcause="FP"})</div><div>Reply</div></div>
判决准则: 呼叫未能建立
测试说明: 该测试项中测试议作为 TG

测试编号: MGC_Generic-Package_5
测试项目: Generic 包——互联故障
预置条件: MGC 向 MG 发起呼叫
测试过程: 步骤 1 MG 出现互联故障, MG 向 MGC 发 Notify 消息, Generalcause= “IW”。 步骤 2 MGC 能正确处理该错误事件。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>Notify(g/cause {Generalcause="IW"})</div><div>Reply</div></div>
判决准则: 呼叫未能建立
测试说明: 该测试项中测试议作为 TG

测试编号: MGC_Generic-Package_6
测试项目: Generic 包——不支持
预置条件: MGC 向 MG 发起呼叫
测试过程: 步骤 1 MG 不支持 MGC 的命令请求, 向 MGC 发 Notify 消息, Generalcause= “UN”。 步骤 2 MGC 能正确处理该错误事件。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>Notify(g/cause{Generalcause="UN"})</div><div>Reply</div></div>
判决准则: 呼叫未能建立
测试说明: 该测试项中测试议作为 TG

5.2.3 信号音发生包

测试编号: MGC_Tone-Package_1
测试项目: 信号音发生包
预置条件: MG 已经成功注册到 MGC, MGC 设置与 MG 不同的信号音间隔
测试过程: 步骤 1 MG 向发起呼叫。 步骤 2 MGC 向 MG 发的 MODIFY 命令中指示 MG 设置各种信号音的 ToneID 以及音的间隔。 包括 dt/rt/bt/ct/sit/wt/prt/cw/cr。 测试消息流程如下所示: <div><div>MGC</div><div>MG</div><div>Modify</div><div>Reply</div></div>
判决准则: MG 根据 MGC 的命令修改各种信号音的 ToneID 和间隔
测试说明: 该测试项中测试仪作为 RG, 对每个信号音都要进行测试

5.2.4 信号音检测包

测试编号: MGC_Tone_Detect-Package_1
测试项目: 信号音检测包
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MG 发起呼叫。 步骤 2 MGC 向 MG 发的 MODIFY 命令中以 Event 描述符设置 MG 可检测的信号音。 测试消息流程如下所示: <div><div>MGC</div><div>MG</div><div>Modify</div><div>Reply</div></div>
判决准则: MG 根据 MGC 的命令修改可检测的信号音
测试说明: 该测试项中测试仪作为 RG

5.2.5 DTMF 相关包测试

测试编号: MGC_DTMF_Detect-Package_5
测试项目: DTMF 检测包
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MG 发起呼叫。 步骤 2 MGC 命令 MG 检测 0~9、*、#等基本 DTMF 信号。 测试消息流程如下所示: <div><div>MGC</div><div>MG</div><div>Notify</div><div>Reply</div></div>
判决准则: 在 Notify 命令中的 ObserventEventsDescriptor 中收集到的数字和所拨的数字是一样的
测试说明: 该测试项中测试仪作为 RG

测试编号: MGC_DTMF_Gen-Package_6
测试项目: DTMF 发生包
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MG 发起呼叫。 步骤 2 MGC 命令 MG 产生 0~9、*、#等基本 DTMF 信号。 测试消息流程如下所示: <div><div>MGC</div><div>MG</div><div><div>Modify (DigitMap)</div><div>Reply</div><div>Notify(number)</div><div>Reply</div></div></div>
判决准则: MG 将用户号码送到 MGC
测试说明: 该测试项中测试仪作为 RG

5.2.6 呼叫接续包测试

测试编号: MGC_Tone_Gen-Package_7
测试项目: 呼叫信号音生成包
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤1 MGC 向 MG 发的 MODIFY 命令中请求 MG 生成信号音,Signal 中设对应某种信号音的 ToneID (dt/rt/bt/ct/sit/wt/pt/cw/cr)。 测试消息流程如下所示: <div><div>MGC</div><div>MG</div><div>Modify (Signal)</div><div>Reply</div></div>
判决准则: MG 返回所支持的 ToneID 给 MGC
测试说明: 该测试项中测试仪作为 RG，对每个信号音都要进行测试

测试编号: MGC_Tone_Detect-Package_8
测试项目: 呼叫信号音检测包
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发的 MODIFY 命令中设置 Event 描述符请求 MG 检测某信号音 ToneID (dt/rt/bt/ct/sit/wt/pt/cw/cr)。 测试消息流程如下所示: <div><div>MGC</div><div>MG</div><div>Modify (Signal)</div><div>Reply</div></div>
判决准则: MG 返回应答给 MGC
测试说明: 该测试项中测试仪作为 RG，对每个信号音都要进行测试

5.2.7 基本导通测试包

测试编号: MGC_Basic-Package_1
测试项目: 基本导通测试包——发送导通测试音
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 媒体网关控制器向媒体网关发送 Modify 要求 MG 对特定中继中的特定线路进行连续性测试, Modify 消息中的终端状态设置为 ServiceState=test, 信号为 Signals={ct/ct}, 事件为 Events={ct/cmp}。 步骤 2 媒体网关回送应答消息, 并将信号“ct”应用于特定的终结点上。 步骤 3 媒体网关向交换电路网络侧发送连续性测试信号。 步骤 4 媒体网关接收到交换电路网络测试返回到连续性测试响应信号。 步骤 5 媒体网关检测到任何返回到音调/信号, 媒体网关向媒体网关控制器发送 Notify 消息报告, 在该消息的事件描述符中采用“cmp”用于通知在终端上观察到的活动, 响应参数值指示连续性测试的执行结果, 如果成功, 则 resp=success。 步骤 6 MGC 在接收到 Notify 消息后应答 MG。
测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div><div>Modify (1)</div><div>Modify Resp (2)</div><div>连续性测试信号(3)</div><div>测试响应信号(4)</div><div>Notify (5)</div><div>Notify Resp (6)</div></div></div>
判决准则: MG 能够将连续性测试信号向 MGC 报告
测试说明: 该测试项中测试仪做为 TG

测试编号: MGC_Basic-Package_2
测试项目: 基本导通测试包——响应导通测试
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 媒体网关控制器向媒体网关发送 Modify 要求 MG 响应 PSTN 交换机对特定中继中的特定线路进行连续性测试, Modify 消息中的终端状态设置为 ServiceState=test, 信号为 Signals={ct/rsp}, LocalControl 模式设置为 loopback。 步骤 2 PSTN 交换机向媒体网关发送连续性测试信号。 步骤 3 媒体网关向媒体网关控制器发送应答消息。 步骤 4 媒体网关向 PSTN 交换机返回到连续性测试响应信号。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>Modify (1)</div><div>Modify Resp (2)</div><div>连续性测试信号(3)</div><div>测试响应信号(4)</div></div>
判决准则: MG 能够将连续性测试信号向 MGC 报告
测试说明: 该测试项中的测试仪作为 TG

5.2.8 网络包测试

测试编号: MGC_Basic-Package_3
测试项目: 网络包
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MG 发起呼叫。 步骤 2 MGC 向 MG 发送的 ADD 消息, 采用描述符 LocalControl 设定 MG 的 nt/jit 值, 设置网络侧终端的最大抖动缓存。 步骤 3 在 MG 上通过网管查看网络侧终端的最大抖动缓存值。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div><div></div><div>Add (LocalControl)</div><div></div><div>Reply</div><div></div></div></div>
判决准则: MG 接受 MGC 所设置的最大抖动缓存值
测试说明: 无

5.2.9 RTP 包测试

测试编号: MGC_Basic-Package_4
测试项目: RTP 包
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MG 之间已经成功建立呼叫。 步骤 2 MGC 向 MG 发 AuditValue 命令, 针对 RTP Termination 用描述符 Statistics, 统计 ps、pr、pl、jit 和 delay (即发包数、收包数、丢包数、抖动、时延)。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>AuditValue</div><div>Reply</div></div>
判决准则: MG 向 MGC 返回统计值
测试说明: 无

5.2.10 TDM 电路包测试

测试编号：MGC_Basic-Package_5
测试项目：TDM 电路包
预置条件： MG 已经成功注册到 MGC
测试过程： 步骤 1 MGC 向 MG 发 Modify 命令，在 Properties 中设 TDM 终结点的回声抑制（ec=on）、增益控制等参数。 测试消息流程如下所示： <div><div>MG</div><div>MGC</div><div>Modify</div><div>Reply</div></div>
判决准则： MG 向 MGC 返回应答
测试说明： 该测试项中测试仪作为 TG

5.2.11 FSK 包测试

测试编号: MGC_Basic-Package_6
测试项目: FSK 包——在第一次振铃时发送 FSK 指示
预置条件: 用户 B 已经配置了主叫号码显示提供业务 CLIP 和呼叫等待等补充业务
测试结构: 用户 A—PSTN/PLMN 交换机—TG—MGC—AG—用户 B
测试过程: 步骤 1 用户 A 呼叫用户 B。 步骤 2 当通知被叫时, MGC 命令 AG 发送 ADS (Analog Display Signaling Megaco Package) 的 ‘display with alerting’ (=dwa) 信号。 ‘Caller ID’ 信息映射到 dwa 信号 ‘pattern’ 参数。 网关负责为被叫产生 FSK 信号, 在振铃期间被编码为 FSK 的 ‘Caller ID’ 信息在合适的时间被插入
判决准则: 用户 B 收到振铃且主叫数据显示在 B 的话机上。 从 MGC 到被叫 AG 的 Modify 命令在 dwa 信号中包括 Caller ID 数据
测试说明: 该测试项中测试仪作为 RG

测试编号: MGC_Basic-Package_7
测试项目: FSK 包——在通话过程中发送 FSK 指示
预置条件: 用户 B 已经配置了 CLIP 和呼叫等待等补充业务
测试结构: 用户 A—PSTN—TG—MGC— +AG—用户 B +AG—用户 C
测试过程: 步骤 1 用户 A 呼叫 B, 且正在通话中。 步骤 2 用户 C 呼叫用户 B。 步骤 3 当等待的呼叫 (C 到 B) 显示给用户 B 时, 呼叫等待音播放给 B。 用户 C 的 ‘Caller ID’ 信息也发送给 B 以区别主叫。 MGC 会命令 AG 发送 ADS (Analog Display Signaling Megaco Package) 的 ‘display with alerting’ (=dwa) 信号。 ‘Caller ID’ 信息映射到 dwa 信号 ‘pattern’ 参数。 网关负责为被叫产生 FSK 信号, 在呼叫等待期间被编码为 FSK 的 ‘Caller ID’ 信息在合适的时间被插入
判决准则: 用户 B 收到呼叫等待音且主叫数据显示在 B 的话机上。 从 MGC 到被叫 AG 的 Modify 命令在 dwa 信号中包括 Caller ID 数据
测试说明: 该测试项中测试仪作为 RG

测试编号：MGC_Basic-Package_8
测试项目：FSK 包——在挂机时发送 FSK 指示
预置条件： 用户 A 已经配置了 CLIP 和呼叫等待等补充业务
测试结构：用户 B—PSTN/PLMN 交换机—TG—MGC—AG—用户 A
测试过程： 步骤 1 用户 A 呼叫 B，且正在通话中。 步骤 2 为向用户 A 发送消息，MGC 会命令 AG 发送 ADS（Analog Display Signaling Megaco Package）的 ‘generic data signaling’（=data）信号。‘Caller ID’ 信息映射到 dwa 信号 ‘Display Data Block’ 参数内，发信号的终端类型映射到 TAS 参数。网关负责为被叫产生 FSK 信号，在呼叫等待期间被编码为 FSK 的 ‘Caller ID’ 信息在 TAS 信号后被插入
判决准则： 用户 A 收到消息等待提示且主叫数据显示在 B 的话机上。 从 MGC 到被叫 AG 的 Modify 命令在 TAS 类型中包含数据信号
测试说明： 该测试项中测试仪作为 RG

5.2.4 分片包

测试编号：MGC_Segmentation-package_1
测试项目：Segmentation package 属性参数设定
预置条件： MG 已经成功注册到 MGC。该分片机制只适用于 H248/UDP/IP 这种协议栈
测试过程： 步骤 1 MGC 向 MG 发送 AuditValue 命令，审计其中“MG Segmentation Timer Value”“MGC Segmentation Timer Value” “MG Maximum PDU Size” “MGC Maximum PDU Size” 的值。 <div><div>MGC</div><div>MG</div><div>AuditValue</div><div>Reply</div></div>
判决准则： MG 返回“MG Segmentation Timer Value” “MGC Segmentation Timer Value” “MG Maximum PDU Size” “MGC Maximum PDU Size” 4 个属性参数的审计值
测试说明： 无

测试编号: MGC_Segmentation-package_2	
测试项目: Segmentation package 应用	
预置条件: MG 已经成功注册到 MGC, 完成 Segmentation package 中的 4 个属性参数设定。	
测试过程: 步骤 1 MGC 向 MG 发 Audit Value 命令, 审计对象为所有 Context 中的所有 Termination; 步骤 2 MG 返回审计结果, 消息中包含多个 TransactionReplies, 每个 TransactionReply 用一个独立的消息发送; 步骤 3 每收到一个分片消息, MGC 回复确认消息, 当 MGC 收到最后一个分片消息, 检查所有分片完整后, 回复 Ack 消息。	
<div><div>MGC</div><div>MG</div><div><div>AuditValue (C1、C2、C3)</div><div>Reply1 (Seg1)</div><div>Reply2 (Seg2)</div><div>Reply3 (Seg3&)</div><div>Confirm Seg1</div><div>Confirm Seg2</div><div>Confirm Seg3&</div></div></div>	
判决准则: MGC 能够按照流程图收到的消息进行正确的回复	
测试说明: 无	

测试编号：MGC_Local-Descriptor_3
测试项目：本端控制描述符——sendreceive 模式
预置条件： MG 已经成功注册到 MGC
测试过程： 步骤 1 MG 发起呼叫。 步骤 2 呼叫过程中，MGC 用 Modify 命令将终结点的 LocalControl Descriptor 改为 “sendreceive”。
<div><div>MG</div><div>MGC</div><div><div>Modify(LocalControl)</div><div>Reply</div></div></div>
判决准则： MG 的终结点根据命令设置成相应的属性
测试说明： 无

测试编号：MGC_Local-Descriptor_4
测试项目：本端控制描述符——inactive 模式
预置条件： MG 已经成功注册到 MGC
测试过程： 步骤 1 MG 发起呼叫。 步骤 2 呼叫过程中，MGC 用 Modify 命令将终结点的 LocalControl Descriptor 改为 “inactive”。
<div><div>MG</div><div>MGC</div><div><div>Modify(LocalControl)</div><div>Reply</div></div></div>
判决准则： MG 的终结点根据命令设置成相应的属性
测试说明： 无

测试编号: MGC_Local-Descriptor_5
测试项目: 本端控制描述符——loopback 模式
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MG 发起呼叫。 步骤 2 呼叫过程中, MGC 用 Modify 命令将终结点的 LocalControl Descriptor 改为 “loopback”。
<div><div>MG</div><div>MGC</div><div>Modify(LocalControl)</div><div>Reply</div></div>
判决准则: MG 的终结点根据命令设置成相应的属性
测试说明: 无

测试编号: MGC_Local-Descriptor_6
测试项目: 本端控制描述符——inactive 模式
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MG 发起呼叫。 步骤 2 呼叫过程中, MGC 用 Modify 命令将终结点的 LocalControl Descriptor 改为 “inactive”。
<div><div>MG</div><div>MGC</div><div>Modify(LocalControl)</div><div>Reply</div></div>
判决准则: MG 的终结点根据命令设置成相应的属性
测试说明: 无

测试编号: MGC_Local-Descriptor_7
测试项目: Modem 描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MG 发起呼叫。 步骤 2 MGC 用 Modify 命令将 MG 的 Modem Descriptor 设为 V.18/V.22/V.22bis/V.32/V.32bis/V.34/V.90/V.92。
<div><div>MG</div><div>MGC</div><div>Modify(Modem)</div><div>Reply</div></div>
判决准则: MG 的终结点根据命令设置成相应的属性
测试说明: 无

5.3.2 事件描述符

测试编号: MGC_Event-Descriptor_1
测试项目: 事件缓存控制 BF=off
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发的消息中设终端状态中的参数 BF=off。
<div><div>MG</div><div>MGC</div><div>Modify(BF=off)</div><div>Reply</div></div>
判决准则: MGC 能够发送相应的消息
测试说明: 无

测试编号: MGC_Event-Descriptor_2
测试项目: 事件缓存控制=lockstep
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发的消息中设 BF=lockstep。 <div><div>MG</div><div>MGC</div><div>Modify(BF=lockstep)</div><div>Reply</div></div>
判决准则: MGC 能够发送相应的消息
测试说明: 无

测试编号: MGC_Event-Descriptor_3
测试项目: 事件描述符——挂机事件
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MG 成功建立呼叫。 步骤 2 用户挂机。 步骤 3 MG 用 Notify 向 MGC 报告挂机事件, Event Descriptor=on-hook。 步骤 4 MGC 向另一端用户所在 MG 发 Modify 命令指示其向用户送忙音,同时向挂机方 MG 发 Subtract 命令要求释放终结点。 步骤 5 MGC 向 MG 发的消息中设 eventbuffer control=lockstep
判决准则: MGC 能够发送相应的消息
测试说明: 该测试项测试仪作为 RG

测试编号: MGC_Event-Descriptor_4
测试项目: 事件描述符——摘机事件
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 用户摘机。 步骤 2 MG 用 Notify 向 MGC 报告摘机事件, Event Descriptor=off-hook。 步骤 3 MGC 向 MG 发 Modify 命令指示其向用户送拨号音和下发号码表 (*1), 接着向 MG 发 Add 命令创建一个关联并将用户侧终结点和网络侧终结点加入关联
判决准则: MGC 应该能够响应 MG 的上报消息
测试说明: 该测试项中测试仪作为 RG

测试编号: MGC_Event-Descriptor_5
测试项目: 事件描述符——传真信号
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 用户通过 MG 发传真。 步骤 2 MG 用 Notify 向 MGC 报告传真事件, Event Descriptor=fax tone。 步骤 3 MGC 向主被叫 MG 发 Modify 命令, 将终端状态改为传真
判决准则: MG 报告事件并接受 MGC 指令
测试说明: 该测试项中测试仪作为 RG

测试编号: MGC_Event-Descriptor_6
测试项目: 事件描述符——导通测试结果
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 用户侧发起连续性测试信号。 步骤 2 MG 用 Notify 向 MGC 报告, Observed Event Descriptor= continuity test results。 步骤 3 MGC 向 MG 回 NOTIFY Response 消息
判决准则: MG 报告事件并接受 MGC 指令
测试说明: 无

5.3.3 Modify 命令所支持的描述符

测试编号:	MGC_Modify-Descriptor_1
测试项目:	信号描述符
预置条件:	MG 已经成功注册到 MGC
测试过程:	步骤 1 MGC 向 MG 发 Modify 命令, 将其中 SignalsDescriptor 设为某个值
判决准则:	MG 命令设置相应的值
测试说明:	无

测试编号:	MGC_Modify-Descriptor_2
测试项目:	号码表描述符
预置条件:	MG 已经成功注册到 MGC
测试过程:	步骤 1 MGC 向 MG 发 Modify 命令, 将其中 DigitMapDescriptor 设为某个值
判决准则:	MG 命令设置相应的值
测试说明:	该测试项中测试仪作为 RG

测试编号:	MGC_Modify-Descriptor_3
测试项目:	观察事件描述符
预置条件:	MG 已经成功注册到 MGC
测试过程:	步骤 1 MGC 向 MG 发 Modify 命令, 将其中 Observed Event Descriptor 设为某个值
判决准则:	MG 命令设置相应的值
测试说明:	无

测试编号: MGC_Modify-Descriptor_4
测试项目: 统计描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 Modify 命令, 将其中 StatisticsDescriptor 设为某个值
判决准则: MG 命令设置相应的值
测试说明: 无

测试编号: MGC_Modify-Descriptor_5
测试项目: 包描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 Modify 命令, 将其中 packageDescriptor 设为某个值
判决准则: MG 命令设置相应的值
测试说明: 无

测试编号: MGC_Modify-Descriptor_6
测试项目: Mux 描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发送 Add 命令, 包含两个媒体流, 分别支持视频和音频。 步骤 2 MGC 向 MG 发送 Modify 命令, 修改 Mux 描述符
判决准则: MG 命令设置相应的值
测试说明: 无

测试编号: MGC_Modify-Descriptor_7
测试项目: 信号描述符 (RequestID)
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 Modify 命令, 其中 SignalsDescriptor 包含多个 Signal, 且每个 Signal 中都包含 RequestID 和 NotifyCompletion 参数。 步骤 2 Notify 消息的上报的 Generic 包中 Signal Completion 事件包含对应的 RequestID
<div><div>MGC</div><div>MG</div><div><div>Modify</div><div>Reply</div><div>Notify</div><div>Reply</div></div></div>
判决准则: MGC 能够按照测试过程中的描述, 在 Modify/Notify 命令中携带相关的 RequestID 参数
测试说明: 无

测试编号: MGC_Modify-Descriptor_8
测试项目: 指示信号播放方向—Internal/external/both
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发送的 MODIFY 命令中设置 Signal 描述符中的 direction 参数为 Internal/external/both。 测试消息流程如下所示:
<div><div>MGC</div><div>MG</div><div><div>Modify (Signal/direction=internal)</div><div>Reply</div></div></div>
判决准则: MG 返回应答给 MGC
测试说明: 三个 direction 参数值 Internal/external/both 均需要测试

测试编号: MGC_Modify-Descriptor_9
测试项目: 信号时延
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发送的 MODIFY 命令中设置 Signal 描述符中的 intersigDelay 参数为一个非 0 值。 测试消息流程如下所示: <div><div>MGC</div><div>MG</div><div>Modify (Signal/intersigDelay=10)</div><div>Reply</div></div>
判决准则: MG 返回应答给 MGC
测试说明: 无

5.3.4 Add 命令所支持的描述符

测试编号: MGC_Add-Descriptor_1
测试项目: 媒体描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 Add 命令, 将其中 MediaDescriptor 设为某个值
判决准则: MG 命令设置相应的值
测试说明: 无

测试编号: MGC_Add-Descriptor_2
测试项目: Modem 描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 Add 命令, 将其中 ModemDescriptor 设为某个值
判决准则: MG 命令设置相应的值
测试说明: 无

测试编号: MGC_Add-Descriptor_3
测试项目: 事件描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 Add 命令, 将其中 EventDescriptor 设为某个值
判决准则: MG 命令设置相应的值
测试说明: 无

测试编号: MGC_Add-Descriptor_4
测试项目: 信号描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 Add 命令, 将其中 SignalsDescriptor 设为某个值
判决准则: MG 命令设置相应的值
测试说明: 无

测试编号: MGC_Add-Descriptor_5
测试项目: 号码表描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 Add 命令, 将其中 DigitMapDescriptor 设为某个值
判决准则: MG 命令设置相应的值
测试说明: 该测试项中测试仪作为 RG

测试编号: MGC_Add-Descriptor_6
测试项目: ContextAttribute 支持 IEPS Indicator
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 Add 命令, 在 context 中携带 ContextAttribute, 携带 IEPS 参数。 测试消息流程如下所示: <div><div>MG</div><div>主 MGC</div><div> <-----ADD REQUEST(IEPS Indicator)----- </div><div> -----ADD Reply-----> </div></div>
判决准则: ContextAttribute 中携带 IEPS Indicator
测试说明: 无

5.3.5 Move 命令所支持的描述符

测试编号: MGC_Move-Descriptor_1
测试项目: 媒体描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 Move 命令, 将其中 MediaDescriptor 设为某个值
判决准则: MG 命令设置相应的值
测试说明: 无

测试编号: MGC_Move-Descriptor_2
测试项目: Modem 描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 Move 命令, 将其中 ModemDescriptor 设为某个值
判决准则: MG 命令设置相应的值
测试说明: 无

测试编号: MGC_Move-Descriptor_3
测试项目: 事件描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 Move 命令, 将其中 EventDescriptor 设为某个值
判决准则: MG 命令设置相应的值
测试说明: 无

测试编号: MGC_Move-Descriptor_4
测试项目: 信号描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 Move 命令, 将其中 SignalsDescriptor 设为某个值
判决准则: MG 命令设置相应的值
测试说明: 无

测试编号: MGC_Move-Descriptor_5
测试项目: 号码表描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 Move 命令, 将其中 DigitMapDescriptor 设为某个值
判决准则: MG 命令设置相应的值
测试说明: 该测试项中测试仪作为 RG

测试编号: MGC_Move-Descriptor_6
测试项目: 观察事件描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 Move 命令, 将其中 ObservedEventsDescriptor 设为某个值
判决准则: MG 命令设置相应的值
测试说明: 无

测试编号: MGC_Move-Descriptor_7
测试项目: 统计描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 Move 命令, 将其中 StatisticsDescriptor 设为某个值
判决准则: MG 命令设置相应的值
测试说明: 无

测试编号: MGC_Move-Descriptor_8
测试项目: 包描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤1 MGC 向 MG 发 Move 命令, 将其中 packageDescriptor 设为某个值
判决准则: MG 命令设置相应的值
测试说明: 无

5.3.6 Subtract 命令所支持的描述符

测试编号: MGC_Subtract-Descriptor_7
测试项目: 统计描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤1 MG 通话成功, MG 主被叫任意一方挂断; 步骤2 MGC 向 MG 发 Subtract 命令, 将其中 StatisticsDescriptor 设为某个值
判决准则: MG 回应的消息中应包含相应的统计值
测试说明: 无

5.3.7 AuditValue 命令所支持的描述符

测试编号: MGC_AuditValue-Descriptor_1
测试项目: 媒体描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤1 MGC 向 MG 对某个终结点发 AuditValue 命令, 其中包含 MediaDescriptor
判决准则: MG 回应的消息中应含有正在使用中的媒体信息
测试说明: 无

测试编号: MGC_AuditValue-Descriptor_2
测试项目: Modem 描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 AuditValue 命令, 审计 ModemDescriptor 值
判决准则: MG 上报相应的值
测试说明: 无

测试编号: MGC_AuditValue-Descriptor_3
测试项目: 事件描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 AuditValue 命令, 审计 EventDescriptor 值
判决准则: MG 上报相应的值
测试说明: 无

测试编号: MGC_AuditValue-Descriptor_4
测试项目: 信号描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 AuditValue 命令, 审计 SignalsDescriptor 值
判决准则: MG 上报相应的值
测试说明: 无

测试编号: MGC_AuditValue-Descriptor_5
测试项目: 号码表描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 AuditValue 命令, 审计 DigitMapDescriptor 值
判决准则: MG 回应的消息中应含有当前已经设置的 DigitMap
测试说明: 无

测试编号: MGC_AuditValue-Descriptor_6
测试项目: 观察事件描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 AuditValue 命令, 审计 ObservedEventsDescriptor 值
判决准则: MG 回应的消息中应返回当前观察到的事件
测试说明: 无

测试编号: MGC_AuditValue-Descriptor_7
测试项目: 统计描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 对某个终结点发 AuditValue 命令, 审计 StatisticsDescriptor 值
判决准则: MG 回应的消息中应含有当前终结点的统计信息
测试说明: 无

测试编号: MGC_AuditValue-Descriptor_8
测试项目: 包描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤1 MGC 向 MG 对某个终结点发 AuditValue 命令, 审计 packageDescriptor 值
判决准则: MG 回应的消息中应含有当前可以使用的包
测试说明: 无

测试编号: MGC_AuditValue-Descriptor_9 (查证一下是否为 v3 项目)
测试项目: Stream descriptor 中的 Statistics Descriptor 应用
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤1 MGC 向 MG 发 AuditValue 命令, Stream Descriptor 中包含 Statistics Descriptor
<div><div>MGC</div><div>MG</div><div><div></div><div>AuditValue</div><div></div></div><div><div></div><div>Reply</div><div></div></div></div>
判决准则: MGC 通过 AuditValue 消息, 在 stream descriptor 的 stream Para 中包含 Statistics Descriptor
测试说明: 无

5.3.8 AuditCapability 命令所支持的描述符

测试编号: MGC_AuditCapability-Descriptor_1
测试项目: 媒体描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤1 MGC 向 MG 发 AuditCapability 命令, 审计 MediaDescriptor 值
判决准则: MG 上报相应的值
测试说明: 无

测试编号: MGC_AuditCapability-Descriptor_2
测试项目: Modem 描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 AuditCapability 命令, 审计 ModemDescriptor 值
判决准则: MG 上报相应的值
测试说明: 无

测试编号: MGC_AuditCapability-Descriptor_3
测试项目: 事件描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 AuditCapability 命令, 审计 EventDescriptor 值
判决准则: MG 上报相应的值
测试说明: 无

测试编号: MGC_AuditCapability-Descriptor_4
测试项目: 信号描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 AuditCapability 命令, 审计 SignalsDescriptor 值
判决准则: MG 上报相应的值
测试说明: 无

测试编号: MGC_AuditCapability-Descriptor_6
测试项目: 观察事件描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 AuditCapability 命令, 审计 ObservedEventsDescriptor 值
判决准则: MG 上报相应的值
测试说明: 无

测试编号: MGC_AuditCapability-Descriptor_7
测试项目: 统计描述符
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 AuditCapablity 命令, 审计 StatisticsDescriptor 值
判决准则: MG 上报相应的值
测试说明: 无

5.4 通配符的应用

测试编号: MGC_WildCard_1
测试项目: MGC 对终结点应用通配符 ALL
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 验证 MGC 支持使用通配符 ALL 表示所有满足条件的终结点。例如: MGC 向 MG 发 Modify 命令对所有终结点检测摘机事件并下发号码表
判决准则: MG 命令设置相应的值
测试说明: 无

测试编号: MGC_WildCard_2
测试项目: MGC 对终结点应用通配符 CHOOSE
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 验证 MGC 支持使用通配符 CHOOSE 表示从满足条件的终结点中选择一个。例如: MGC 向 MG 发送 Modify 命令请求选择到被叫用户的中继
判决准则: MG 执行相应的命令
测试说明: 无

测试编号: MGC_WildCard_3
测试项目: MGC 对关联应用通配符 ALL
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发送的命令中 ContextID=ALL; 步骤 2 MG 正确对所有存在关联执行同一命令
判决准则: MG 执行相应的命令
测试说明: 无

测试编号: MGC_WildCard_4
测试项目: MGC 对关联应用通配符 CHOOSE
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发送的命令中 ContextID=CHOOSE; 步骤 2 MG 正确产生一个新的关联
判决准则: MG 执行相应的命令
测试说明: 无

测试编号: MGC_WildCard_5
测试项目: Root 终结点——Modify 命令使用
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 验证 MGC 支持 Modify 命令使用 Root 终结点。如: Modify 命令使用 Root 终结点修改网关的参数
判决准则: MG 执行相应的命令
测试说明: 无

测试编号: MGC_WildCard_6
测试项目: Root 终结点——AuditValue 命令使用
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 验证 MGC 支持 AuditValue 命令使用 Root 终结点。如: AuditValue 命令使用 Root 终结点审计网关的参数值
判决准则: MG 执行相应的命令
测试说明: 无

测试编号: MGC_WildCard_7
测试项目: Root 终结点——AuditCapability 命令使用
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 验证 MGC 支持 AuditCapability 命令使用 Root 终结点。如: AuditCapability 命令使用 Root 终结点审计网关的性能
判决准则: MG 执行相应的命令
测试说明: 无

5.5 定时器的应用

测试编号: MGC_Timer_1
测试项目: 渐进退出服务定时器
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MG 向 MGC 发 ServiceChange 命令, 方法为 graceful, Reason=905, 并设 delay; 步骤 2 在 delay 时间后, MGC 向 MG 发送 Subtract 命令, TermA 退出服务
判决准则: 在指定时间后, TermA 停止服务
测试说明: 无

测试编号: MGC_Timer_2
测试项目: MGC 重发定时器
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发消息; 步骤 2 MG 超时 (定时器可设) 没有响应消息; 步骤 3 MGC 重发消息
判决准则: 在 MG 超时后能够接收到 MGC 重新发送的消息
测试说明: 无

测试编号: MGC_Timer_3
测试项目: 接收号码定时器——起始定时器超时
预置条件: MG 中至少预置了一个 digitMap 相关的起始、短和长定时器
测试结构: MGC—MG—用户 A
测试过程: 步骤 1 用户 A 摘机后接收到拨号音。 步骤 2 用户摘机, 在 T 时间内没有拨号。 步骤 3 经过 T 时间, 用户 A 接收到忙音
判决准则: 在 T 时间超时后, 用户听忙音。在步骤 3 中, MG 向 MGC 发送 Notify 命令, 包括: — 事件类型 “digitMapCompletion” (=dd/ce) — observedEventsDescriptor.digitString=empty — observedEventsDescriptor.TerminationMethod=PM (局部匹配)
测试说明: 该测试项中测试仪作为 RG

测试编号: MGC_Timer_4
测试项目: 接收号码定时器——在 ‘起始定时器’ 范围内拨号
预置条件: MG 中至少预置了一个 digitMap 相关的起始、短和长定时器
测试结构: MGC—MG—用户 A
测试过程: 步骤 1 用户 A 摘机后接收到拨号音。 步骤 2 用户摘机, 在 T 时间内拨打正确的号码。 步骤 3 MGC 取消定时器 T
判决准则: 在步骤 2 完成后, MG 向 MGC 发送 Notify 命令, 包括: — 事件类型 “digitMapCompletion” (=dd/ce) — observedEventsDescriptor.digitString=《所拨打的号码》 — observedEventsDescriptor.TerminationMethod=PM (完全匹配)
测试说明: 该测试项中测试仪作为 RG

测试编号: MGC_Timer_5
测试项目: 接收号码定时器——长定时器超时
预置条件: MG 中至少预置了一个 digitMap 相关的起始定时器 (10s)、短定时器 (4s) 和长定时器 (16s)
测试结构: MGC—MG—用户 A
测试过程: 步骤 1 用户摘机, 听到拨号音。 步骤 2 用户在 <i>T</i> 时间范围内开始拨号, 拨打不完整的号码, 等待长定时器超时
判决准则: 在步骤 2 完成的 16s 后, MG 向 MGC 发送 Notify 命令, 包括: — 事件类型 “digitMapCompletion” (=dd/ce) — observedEventsDescriptor.digitString=《所拨打的不完整号码》 — observedEventsDescriptor.TerminationMethod=PM (完全匹配)
测试说明: 该测试项中测试仪作为 RG

测试编号: MGC_Timer_6
测试项目: 接收号码定时器——在长定时器内拨号
预置条件: 测试仪中至少预置了一个 digitMap 相关的起始定时器 (10s)、短定时器 (4s) 和长定时器 (16s)
测试结构: MGC—MG—用户 A
测试过程: 步骤 1 用户摘机, 听到拨号音。 步骤 2 用户开始在 <i>T</i> 时间范围内拨号, 依次间隔 10s 拨打每一位号码, 直到拨打完整的电话号码完成
判决准则: 在步骤 2 完成之后, MG 向 MGC 发送 Notify 命令, 包括: — 事件类型 “digitMapCompletion” (=dd/ce) — observedEventsDescriptor.digitString=《所拨打的完整号码》 — observedEventsDescriptor.TerminationMethod=PM (完全匹配)
测试说明: 该测试项中测试仪作为 RG

测试编号: MGC_Timer_7
测试项目: 接收号码定时器——短定时器超时
预置条件: 测试仪中至少预置了一个 digitMap 相关的起始定时器 (10s)、短定时器 (4s) 和长定时器 (16s)
测试结构: MGC—MG—用户 A
测试过程: 步骤 1 用户 A 摘机, 听到拨号音。 步骤 2 用户拨一位号码, 该号码与 digitMap 中的模式相匹配。 步骤 3 用户不继续拨号, 等待短定时器超时
判决准则: 在步骤 2 完成之后 S 秒, MG 向 MGC 发送 Notify 命令, 包括: — 事件类型 “digitMapCompletion” (=dd/ce) — observedEventsDescriptor.digitString=《所拨打的一位号码》 — observedEventsDescriptor.TerminationMethod=PM (完全匹配)
测试说明: 该测试项中测试仪作为 RG

测试编号: MGC_Timer_8
测试项目: 接收号码定时器——短定时器未超时
预置条件: 测试仪中至少预置了一个 digitMap 相关的起始定时器 (10s)、短定时器 (4s) 和长定时器 (16s)
测试结构: MGC—MG—用户 A
测试过程: 步骤 1 用户 A 摘机, 听到拨号音。 步骤 2 用户拨打数字串 ‘AB’, 等待 S 秒, 但少于 T 秒, AB 与 digitMap 模式相匹配。数字 ‘A’ 在 T 时间内拨入, ‘B’ 在 S 时间内拨入。 步骤 3 用户继续拨号, 拨打完整的号码 ‘ABXXXXXX’
判决准则: 在步骤 3 完成之后, MG 向 MGC 发送 Notify 命令, 包括: — 事件类型 “digitMapCompletion” (=dd/ce) — observedEventsDescriptor.digitString=《ABXXXXXX》 — observedEventsDescriptor.TerminationMethod=UM (准确匹配)
测试说明: 该测试项中测试仪作为 RG

测试编号: MGC_Timer_9
测试项目: 等待应答消息定时器超时
预置条件: MG 已经成功注册到 MGC 并发起呼叫。
测试结构: 用户 A—MG—MGC—MG—用户 B
测试过程: 步骤 1 A 呼叫 B。 步骤 2 B 收到振铃指示。 步骤 3 B 没有应答直至超时。 步骤 4 应答定时器在 90s 之后超时, MGC 向用户 A 发出呼叫释放
判决准则: 1 MGC 向用户 B 发送的 Modify 命令使用的 ‘ringing’ 信号必须为 on/off 类型。 2 当呼叫超时, 用户 B 不发送任何 Notify 命令给 MGC。 3 MGC 发送 Subtract 命令给用户 B 释放呼叫
测试说明: 该测试项中测试仪作为 RG

测试编号: MGC_Timer_10
测试项目: 等待应答消息定时器未超时
预置条件: MG 已经成功注册到 MGC 并发起呼叫
测试结构: 用户 A—MG—MGC—MG—用户 B
测试过程: 步骤 1 A 呼叫 B。 步骤 2 B 收到振铃指示。 步骤 3 B 及时应答
判决准则: 1 MGC 向用户 B 发送的 Modify 命令使用的 ‘ringing’ 信号必须为 on/off 类型。 2 当呼叫超时, 用户 B 不发送任何 Notify 命令给 MGC。 3 MGC 发送 Subtract 命令给用户 B 释放呼叫
测试说明: 该测试项中测试仪作为 RG

测试编号：MGC_Timer_11
测试项目：等待 SS7 ACM 或 CON
预置条件： PSTN/PLMN 交换机不能返回 ACM 或 CON 到 MGC，MGC 中 ISUP 定时器 T7 设置为 30s
测试结构：A—PSTN/PLMN 交换机—MG—MGC—MG—PSTN/PLMN 交换机—B
测试过程： 步骤 1 用户 A 拨打用户 B
判决准则： MGC 中的 ISUP 定时器 T7 在 30s 之后终止，大约在 30s 之后，主叫 A 接收到“congest/busy tone”
测试说明： 该测试项中测试仪作为 TG

测试编号：MGC_Timer_12
测试项目：等待 SS7 ANM
预置条件： MGC 中 ISUP 定时器 T9 设置为 90s
测试结构：A—PSTN/PLMN 交换机—MG—MGC—MG—PSTN/PLMN 交换机—B
测试过程： 步骤 1 用户 A 拨打用户 B，接通后用户 B 不应答。 步骤 2 ISUP 定时器超时后，用户 B 停止振铃
判决准则： MGC 中的 ISUP 定时器 T9 在 90s 之后终止，大约在 90s 之后，主叫 A 接收到“congest/busy tone”
测试说明： 该测试项中测试仪作为 TG

5.6 错误码

测试编号：MGC_Error-Code_1
测试项目：返回错误码 402
预置条件： 在 MGC 中配置可注册的 MG，MG 为无权注册的 MG
测试过程： 步骤 1 MG 向 MGC 注册时，消息头所带的认证码不对。 步骤 2 MGC 返回错误码 402
判决准则： MG 接收到的 Reply 消息中带错误码 402
测试说明： 无

测试编号: MGC_Error-Code_2
测试项目: 在确定事务标识前收到错误码 403
预置条件: MG 已经向 MGC 发起注册
测试过程: 步骤1 MG 向 MGC 发起注册请求, 注册请求中的事务标志语法错误。 步骤2 MGC, (以下同) 返回错误码403
判决准则: MG 接收到的 Reply 消息中带错误码 403
测试说明: 无

测试编号: MGC_Error-Code_3
测试项目: 在确定事务标识前收到错误码 410
预置条件: MG 已经向 MGC 发起注册
测试过程: 步骤1 MG 向 MGC 发起注册请求, 注册请求中具有不正确的标志符, 如 TransactionID 错误。 步骤2 MGC返回错误码410
判决准则: MG 接收到的 Reply 消息中带错误码 410
测试说明: 无

测试编号: MGC_Error-Code_4
测试项目: 返回错误码 444
预置条件: MG 已经向 MGC 发起注册
测试过程: 步骤1 MG 向 MGC 发起注册请求, 注册请求中具有不能识别/错误的描述符。 步骤2 MGC返回错误码444
判决准则: MG 接收到的 Reply 消息中带错误码 444
测试说明: 无

测试编号: MGC_Error-Code_5
测试项目: 返回错误码 445
预置条件: MG 已经向 MGC 发起注册
测试过程: 步骤1 MG 向 MGC 发起注册请求, 注册请求中具有不能识别的 property。 步骤2 MGC返回错误码445
判决准则: MG 接收到的 Reply 消息中带错误码 445
测试说明: 无

测试编号: MGC_Error-Code_6
测试项目: 返回错误码 446
预置条件: MG 已经向 MGC 发起注册
测试过程: 步骤1 MG 向 MGC 发起注册请求, 注册请求中具有不能识别/错误的参数。 步骤2 MGC返回错误码446
判决准则: MG 接收到的 Reply 消息中带错误码 446
测试说明: 无

测试编号: MGC_Error-Code_7
测试项目: 返回错误码 447
预置条件: MG 已经向 MGC 发起注册
测试过程: 步骤1 MG 向 MGC 发起注册请求, 注册请求中具有非法的描述符。 步骤2 MGC返回错误码447
判决准则: MG 接收到的 Reply 消息中带错误码 447
测试说明: 无

测试编号: MGC_Error-Code_8
测试项目: 返回错误码 448
预置条件: MG 已经向 MGC 发起注册
测试过程: 步骤1 MG 向 MGC 发起注册请求, 注册请求中具有重复的描述符。 步骤2 MGC返回错误码448
判决准则: MG 接收到的 Reply 消息中带错误码 448
测试说明: 无

测试编号: MGC_Error-Code_9
测试项目: 返回错误码 505
预置条件: MG 已经向 MGC 发起注册
测试过程: 步骤1 当MG向MGC发送ServiceChange请求, 请求注册。 步骤2 MG接收到Reply, 注册成功。 步骤3 MGC向MG发送ServiceChange请求。 步骤4 MG不应答并向MGC发送ServiceChange请求。 步骤5 MGC返回的错误码505
判决准则: MG 接收到 MGC 返回的的 Reply 消息中带错误码 505
测试说明: 无

5.7 杂项

测试编号: MGC_Other_1
测试项目: MGC 支持的编码方式
预置条件: MG 与 MGC 上都配置二进制编码方式和文本编码方式
测试过程: 步骤1 MG采用二进制编码方式向MGC发送注册请求。 步骤2 MG采用文本方式, 向MGC发起注册请求
判决准则: 在两种编码方式下, MG 都能够接收到 MGC 的 Reply 消息
测试说明: 无

测试编号: MGC_Other_2
测试项目: MGC 对事务请求最多处理一次
预置条件: MG 已经注册到 MGC 上
测试过程: 步骤1 MG 向 MGC 发送多个一样的事务请求。 步骤2 MGC只做一次处理, 回应一个Reply消息
判决准则: MG 只能接收到 MGC 的一个 Reply 消息
测试说明: 无

测试编号: MGC_Other_3
测试项目: MGC 的三次握手过程
预置条件: MG 已经注册到 MGC 上。
测试结构: MGC—MG—PSTN/PLMN 交换机—电话
测试过程: 步骤1 通过话机拨通一次成功的呼叫。 步骤2 根据跟踪的消息, 检查消息来往的内容
判决准则: 证实 MGC 能够正确处理消息, ACK 不是以单独的消息存在, 而是插入下一个事务交互中 (T、P、K)
测试说明: 无

测试编号: MGC_Other_4
测试项目: Context Attributes 中携带 ContextIDList 参数
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MG 上存在两个 CONTEXT, 呼叫通话状态中。 步骤 2 MGC 向 MG 发 AuditValue 命令, 审计 context 中所有端点。 步骤 3 MG 回复的 reply 消息中 contextIDlist 中表示两个 Context ID。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>AuditValue (ContextIDlist (*))</div><div>Reply (ContextIDlist(1,2))</div></div>
判决准则: MGC 下发的消息中携带 ContextIDlist; MG 回复的响应中 ContextIDlist 参数为当前 MG 中全部 ContextID
测试说明: 无

测试编号: MGC_Other_5
测试项目: 命令中支持 TerminationIDList 参数
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发 AuditValue 命令, 审计 TD1/TD2 端点, 命令中携带 TerminationIDList 参数。 步骤 2 MG 回复的 reply 消息中 TerminationIDList 中值为 TD1/TD2。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>AuditValue (TerminationIDList (TD1/TD2))</div><div>Reply (TerminationIDList (TD1/TD2))</div></div>
判决准则: MGC 下发的消息中携带 TerminationIDList; MG 回复的响应中 TerminationIDList 参数为当前 MG 中全部 TerminationIDList
测试说明: 无

测试编号: MGC_Other_6
测试项目: 精确审计—利用过滤条件
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 MGC 向 MG 发送 AuditValue 命令, 设置过滤条件, MG 根据过滤条件返回响应。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div> <----- AuditValue -----> </div><div> ----- Reply -----> </div></div>
判决准则: MG 向 MGC 返回审计值
测试说明: 无

测试编号: MGC_Other_7
测试项目: 根据 notreg 值流控 Notify 消息
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 当 MGC 发生过载, MGC 检测到过载后, 向 MG 发送 Modify 消息并设置 notreg=10, 对所有 MG 发送的 Notify 消息进行抑制。 步骤 2 当 MG 连续产生 Notify 时, 对第一个 Notify 进行抑制, 接受 9 个 Notify 消息后达到 10%的比例, 然后重复这个过程直到 MGC 过载解除。 步骤 3 当 MGC 过载解除后, MGC 向 MG 发送 Modify 消息并且 notreg=0, 解除对 Notify 消息的抑制 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>MGC 检测到过载发生, 要求 MG 限制 10%的 Notify 消息</div><div> <-----Modify (Notreg=10)----- </div><div> -----Reply-----> </div><div>对于MG发送给MGC的Notify消息按照10%的比例进行控制</div><div>MGC检测到过载解除后, 要求MG不限制Notify消息</div><div> <-----Modify (Notreg=0)----- </div><div> -----Reply-----> </div></div>
判决准则: 收到抑制的 Notify 消息和流控比例相等; MGC 过载解除后, Notify 消息可以正常上报, 不会被抑制
测试说明: 无

测试编号: MGC_Other_8
测试项目: 根据 Notify 流控策略流控 Notify 消息
预置条件: MG 已经成功注册到 MGC
测试过程: 步骤 1 当 MGC 发生过载, MGC 检测到过载后, 向 MG 发送 Modify 消息并设置 notreg=10, 对所有 MG 发送的 Notify 消息进行抑制。MGC 可选择通过 Modify 消息下发具体的 Notify 命令流控策略。 步骤 2 根据不同的事件触发采取不同的 Notify 处理策略。 步骤 3 当 MGC 过载解除后, MGC 向 MG 发送 Modify 消息并且 notreg=0, 解除对 Notify 消息的抑制。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>MGC 检测到过载发生, 要求 MG 限制 10%的 Notify 消息</div><div> <----- Modify (Notreg=10) ----- </div><div> -----Reply -----> </div><div>MGC 可选择通过 Modify 消息下发具体的 Notify 命令流控策略, 主要针对 Null 关联 中的部分或所有端点</div><div> <----- Modify ----- </div><div> -----Reply -----> </div><div>根据不同事件的触发, 进行 Notify 流控</div><div>MGC 检测到过载解除后, 要求 MG 不限制 Notify 消息</div><div> <----- Modify (Notreg=0) ----- </div><div> -----Reply -----> </div></div>
判决准则: 根据 NotifyBehaviour 参数值来处理 Notify 消息的流控策略; MGC 过载解除后, Notify 消息可以正常上报, 不会被抑制
测试说明: 无

5.8 安全

测试编号：MGC_Security_1（可选）
测试项目：MGC 支持 IPsec
预置条件： MG 与 MGC 配置支持 IPsec
测试过程： 步骤1 MG向MGC发送经Ipsec加密的注册请求。 步骤2 MGC返回Reply。 <div><div>MG</div><div>MGC</div><div>ServiceChangeRoot</div><div>Reply</div></div>
判决准则： MG 能够成功注册
测试说明： 无

测试编号：MGC_Security_2
测试项目：MGC 支持加密和鉴权机制（过渡性 AH 方案）
预置条件： MG 与 MGC 配置加密机制，MGC 上配置鉴权机制
测试过程： 步骤1 MG1向MGC发送带AH头的加密的注册请求，该MG为未授权用户。 步骤2 MGC返回Reply。 步骤3 MG2向MGC发送带AH头的加密的注册请求，该MG为授权用户。 步骤4 MGC返回Reply。 <div><div>MG</div><div>MGC</div><div>ServiceChangeRoot</div><div>Reply</div></div>
判决准则： MG1 不能成功注册，MG2 能够成功注册
测试说明： 无

6 H.248 MG 测试项目

如果没有特殊说明，本节测试项适用于所有类型的媒体网关设备：中继媒体网关（TG）、用户媒体网关（RG）或综合接入设备（IAD）。

6.1 ServiceChange

测试编号: MG_SC_1
测试项目: MGC 响应 MG 的注册请求
预置条件: MG 加电启动向 MGC 发送注册请求
测试过程: 步骤 1 MG 向 MGC 发送注册请求, ServiceChange 命令中的 method=Restart, Termination=Root。 步骤 2 MGC 接受注册, 返回 Reply, 其中包含 MGCID。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div> -----ServiceChange ROOT Method=Restart-----> </div><div> <-----Reply----- </div></div>
判决准则: MG 应能够在 MGC 上成功注册
测试说明: 无

测试编号: MG_SC_2
测试项目: 再注册
预置条件: MG 向主 MGC 注册时, 主 MGC 不接受, 返回备用 MGC 的地址, MG 再根据所返回地址向备用 MGC 注册或者注册后超时未接收到回答, 则重复发出注册请求
测试过程: 步骤 1 整机连接好电源, 使媒体网关与 MGC 处于断开状态。 步骤 2 将媒体网关与主媒体网关控制器相连接, 媒体网关向主媒体网关控制器注册, 主媒体网关控制器不接受媒体网关的注册, 并在 reply 中返回次媒体网关控制器的地址。 步骤 3 媒体网关继续向次媒体网关控制器进行注册, 次媒体网关控制器接受媒体网关的注册, 并应返回正确的应答。 测试消息流程如下所示: <div><div>MG主 MGC</div><div> ----- ServiceChange ROOT Method= Restart -----> </div><div> <----- Reply (MGCIId) ----- </div><div>MG备用 MGC</div><div> ----- ServiceChange ROOT Method= Restart -----> </div><div> <----- Reply ----- </div></div> 或者 <div><div>MGMGC</div><div> ----- ServiceChange ROOT Method= Restart -----> </div><div> ----- ServiceChange ROOT Method= Restart -----> </div></div>
判决准则: MG 应能够向备用 MGC 发起注册或者反复发送注册请求
测试说明: 无

测试编号: MG_SC_3
测试项目: 报告终结点进入服务
预置条件: MG 向主 MGC 注册成功
测试过程: 步骤 1 参见测试项 MG_SC_4, 人为造成 SCN 侧某个终结点返回业务 (插入被拔出的中继)。 步骤 2 当终结点进入服务时, MG 向 MGC 发 ServiceChange, 其中 method=Restart, ServiceChange Reason=900。 测试消息流程如下所示: <div><div>MG</div><div>主 MGC</div><div> ----- ServiceChange Method= Restart -----> </div><div> <----- Reply ----- </div></div>
判决准则: MGC 应能够接收到 MG 的终结点业务恢复报告并响应
测试说明: 无

测试编号: MG_SC_4
测试项目: 冷启动
预置条件: MG 向主 MGC 注册时, 主 MGC 不接受, 返回备用 MGC 的地址, MG 再根据所返回地址向备用 MGC 注册
测试过程: 步骤 1 整机连接好电源, 使媒体网关与 MGC 处于断开状态。 步骤 2 将媒体网关与主媒体网关控制器相连接, 媒体网关向主媒体网关控制器注册, 主媒体网关控制器不接受媒体网关的注册, 并在 reply 中返回次媒体网关控制器的地址。 步骤 3 媒体网关继续向次媒体网关控制器进行注册, 次媒体网关控制器接受媒体网关的注册, 并应返回正确的应答。MG 向 MGC 发 ServiceChange 消息, method=Restart, Termination=Root, ServiceChangeReason=901。 测试消息流程如下所示: <div><div>MG</div><div>主 MGC</div><div> -----ServiceChange ROOT Method= Restart-----> </div><div> <-----Reply (MGCIId)----- </div><div>MG</div><div>备用 MGC</div><div> -----ServiceChange ROOT Method= Restart-----> </div><div> <-----Reply----- </div></div>
判决准则: MG 应能够在备用 MGC 上成功注册
测试说明: 无

测试编号: MG_SC_5
测试项目: 热启动
预置条件: MG 向 MGC 注册成功, 断开连接后一段时间再次注册
测试过程: 步骤 1 整机连接好电源, 使媒体网关与 MGC 处于断开状态。 步骤 2 将媒体网关与主媒体网关控制器相连接, 媒体网关向主媒体网关控制器注册, 主媒体网关控制器不接受媒体网关的注册, 并在 reply 中返回次媒体网关控制器的地址。 步骤 3 媒体网关与 MGC 断开后, 重新向媒体网关控制器进行注册, MG 向 MGC 发 ServiceChange 消息, method=Restart, Termination=Root , ServiceChangeReason=902。 测试消息流程如下所示: <div><div>MG</div><div>主 MGC</div><div> ----- ServiceChange ROOT Method= Restart -----> </div><div> <----- Reply ----- </div><div>MG 断开与 MGC 的通信</div><div> ----- ServiceChange ROOT Method= Restart -----> </div><div> <----- Reply ----- </div></div>
判决准则: MG 应能够在 MGC 上成功注册
测试说明: 无

测试编号: MG_SC_6
测试项目: 报告终结点故障
预置条件: 在 MG 上建立好一个呼叫
测试过程: 步骤 1 人为制造 MG 上终结点故障 (如拔出已建立呼叫所在的中继板)。 步骤 2 MG 向 MGC 发 ServiceChange 消息, method=Force, Termination 为某个终结点, ServiceChangeReason=904 (终结点故障)。 测试消息流程如下所示: <div><div>MG</div><div>主 MGC</div><div> -----ServiceChange Method= Restart-----> </div><div> <-----Reply----- </div></div>
判决准则: MGC 应能够接收到 MG 的故障报告并响应
测试说明: 无

测试编号: MG_SC_7
测试项目: 报告终结点退出服务
预置条件: MG 向主 MGC 注册成功
测试过程: 步骤 1 MG 呼叫成功后, 人为造成 SCN 侧终结点退出业务 (通过命令行命令强迫终结点退出业务)。 步骤 2 MG 向 MGC 发 ServiceChange 消息, method=Force, Termination 为某个终结点, ServiceChangeReason=905 (终结点退出服务)。 测试消息流程如下所示: <div><div>MG</div><div>主 MGC</div><div> ----- ServiceChange Method= Restart -----> </div><div> <----- Reply ----- </div></div>
判决准则: MGC 应能够接收到 MG 的终结点业务退出报告并响应
测试说明: 无

测试编号: MG_SC_8
测试项目: 报告低层连接丢失
预置条件: MG 向主 MGC 注册成功
测试过程: 步骤 1 MG 呼叫成功后, 人为制造 IP 层或者数据链路层故障, 如使用网络流量发生器制造大流量使 IP 端口拥塞或者拔掉 IP 传输网络侧接口网线 <i>n</i> 秒。 步骤 2 MG 向 MGC 发 ServiceChange 消息, method=Restart, ServiceChangeReason=906 (低层连接丢失)。 测试消息流程如下所示: <div><div>MG</div><div>主 MGC</div><div> ----- ServiceChange Method= Restart -----> </div><div> <----- Reply ----- </div></div>
判决准则: MGC 应能够接收到 MG 低层连接丢失报告并响应
测试说明: 无

测试编号: MG_SC_9
测试项目: 报告传输故障
预置条件: MG 向主 MGC 注册成功
测试过程: 步骤 1 MG 呼叫成功后, 人为制造 IP 层或者数据链路层故障, 如插拔以太网端口使 IP 传输链路产生丢包现象。 步骤 2 MG 向 MGC 发 ServiceChange 消息, method=Restart, ServiceChangeReason=907 (传输故障)。 测试消息流程如下所示: <div><div>MG</div><div>主 MGC</div><div> -----ServiceChange Method= Restart-----> </div><div> <-----Reply----- </div></div>
判决准则: MGC 应能够接收到 MG 的传输故障报告并响应
测试说明: 无

测试编号: MG_SC_10
测试项目: 报告 MG 暂停服务
预置条件: MG 向主 MGC 注册成功
测试过程: 步骤 1 MG 呼叫成功后, 人为制造 MG 故障使 MG 停止业务 (如通过命令行将 MG Disable), 但仍然保持与 MGC 的通信。 步骤 2 MG 向 MGC 发 ServiceChange 消息, method=Force, Termination=Root, ServiceChangeReason=908 (MG impending failure)。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div> ----- ServiceChange ROOT Method= Force -----> </div><div> <----- Reply ----- </div></div>
判决准则: MGC 应能够接收到 MG 的暂停服务报告并响应
测试说明: 无

测试编号: MG_SC_11
测试项目: MGC 暂停服务
预置条件: MG 向主 MGC 注册成功
测试过程: 步骤 1 MG 呼叫成功后, 人为制造 MGC 故障使 MGC 停止业务, 但仍然保持与 MG 的通信。 步骤 2 MGC 向 MG 发 ServiceChange 消息, method=Forced, Termination=Root, ServiceChangeReason=909 (MGC impending failure), MG 发 ServiceChange Reply。 测试消息流程如下所示: <div><div>MG</div><div>主 MGC</div><div> <----- ServiceChange ROOT Method= Forced -----> </div><div> ----- Reply -----> </div></div>
判决准则: MG 应能够正确向 MGC 应答
测试说明: 无

测试编号：MG_SC_12
测试项目：报告媒体能力故障
预置条件： 在媒体网关和 MGC 上配置相应的数据
测试过程： 步骤1 MG 向 MGC 发送 ServiceChange 消息, method=Forced, Termination=Root, ServiceChangereason=910 (Media Capability Failure); 步骤2 MGC 向 MG 应答。 测试消息流程如下所示： <div><div>MG</div><div>MGC</div><div>ServiceChange ROOT ServiceChangeReason=910</div><div>Reply</div></div>
判决准则： MGC 应能够接收到 MG 的故障报告并响应
测试说明： 无

测试编号: MG_SC_13
测试项目: 报告 Modem 能力故障
预置条件: 在媒体网关和 MGC 上配置相应的数据
测试过程: 步骤 1 MG 向 MGC 发送 ServiceChange 消息, method=Forced, Termination=Root, ServiceChangereason=911 (Modem Capability Failure); 步骤 2 MGC 向 MG 应答。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div> ----- ServiceChange ROOT ServiceChangeReason=911 -----> </div><div> <----- Reply ----- </div></div>
判决准则: MGC 应能够接收到 MG 的故障报告并响应
测试说明: 无

测试编号: MG_SC_14
测试项目: 报告 Mux 能力故障
预置条件: 在媒体网关和 MGC 上配置相应的数据
测试过程: 步骤1 MG 向 MGC 发送 ServiceChange 消息, method=Forced, Termination=Root, ServiceChangereason=912 (Mux Capability Failure); 步骤2 MGC 向 MG 应答。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div> ----- ServiceChange ROOT ServiceChangeReason=912 -----> </div><div> <----- Reply ----- </div></div>
判决准则: MGC 应能够接收到 MG 的故障报告并响应
测试说明: 无

测试编号: MG_SC_15
测试项目: 报告信号能力故障
预置条件: 在媒体网关和 MGC 上配置相应的数据
测试过程: 步骤 1 MG 向 MGC 发送 ServiceChange 消息, method=Forced, Termination=Root, ServiceChangereason=913 (Signal Capability Failure); 步骤 2 MGC 向 MG 应答。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div> ----- ServiceChange ROOT ServiceChangeReason=913 -----> </div><div> <----- Reply ----- </div></div>
判决准则: MGC 应能够接收到 MG 的故障报告并响应
测试说明: 无

测试编号: MG_SC_16
测试项目: 报告 Event 能力故障
预置条件: 在媒体网关和 MGC 上配置相应的数据
测试过程: 步骤1 MG 向 MGC 发送 ServiceChange 消息, method=Forced, Termination=Root, ServiceChangereason=914 (Event Capability Failure); 步骤2 MGC 向 MG 应答。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div> ----- ServiceChange ROOT ServiceChangeReason=914 -----> </div><div> <----- Reply ----- </div></div>
判决准则: MGC 应能够接收到 MG 的故障报告并响应
测试说明: 无

测试编号: MG_SC_17
测试项目: 报告状态丢失
预置条件: 在媒体网关和 MGC 上配置相应的数据
测试过程: 步骤 1 MG 向 MGC 发送 ServiceChange 消息, method=Forced, Termination=Root, ServiceChangereason=915 (State Loss); 步骤 2 MGC 向 MG 应答。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div> ----- ServiceChange ROOT ServiceChangeReason=915 -----> </div><div> <----- Reply ----- </div></div>
判决准则: MGC 应能够接收到 MG 的故障报告并响应
测试说明: 无

测试编号: MG_SC_18
测试项目: MG 请求终结点渐进退出服务 delay≠null
预置条件: MG 已经在 MGC 上注册成功
测试过程: 步骤 1 MG 发 ServiceChange 命令, 方法为 graceful, 并设 delay 且 delay≠null。 步骤 2 在 delay 时间后, MG 回应 MGC 的 Audit 消息, 上报 TermA 状态为 OutOfService。MGC 发出删除终结点 TermA 命令, TermA 退出服务。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>ServiceChange TermA Method=graceful Delay=x</div><div>Reply</div><div>在 x 秒后, TermA 退出业务</div><div>Subtract TermA</div><div>Reply</div></div>
判决准则: 在 delay 时间后 MGC 上看不到该 MG TermA 的信息
测试说明: 无

测试编号: MG_SC_19
测试项目: MG 请求终结点渐进退出服务 delay=null
预置条件: MG 已经在 MGC 上注册成功
测试过程: 步骤 1 MG 发 ServiceChange 命令, 方法为 graceful, 并设 delay 且 delay=null。 MG 回应 MGC 的 Audit 消息, 上报 TermA 状态为 OutOfService, 收到 MG 的请求后, MGC 立即发出删除终结点 TermA 命令, TermA 退出服务。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>ServiceChange TermA Method=graceful Delay=null</div><div>Reply</div><div>Subtract TermA</div><div>Reply</div></div>
判决准则: 在 MGC 上看不到该 MG TermA 的信息
测试说明: 无

测试编号: MG_SC_20
测试项目: MG 请求终结点强制退出服务
预置条件: MG 已经在 MGC 上注册成功
测试过程: 步骤 1 MG 产生 ServiceChange 命令, 方法为 Forced。分为指定关联和不指定关联 (contextID=null) 两种情况; 步骤 2 MGC 立即发出删除终结点 TermA 命令, TermA 退出服务。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>ServiceChange TermA Method=Forced</div><div>Reply</div><div>立即使 TermA 退出业务</div><div>Subtract TermA</div><div>Reply</div></div>
判决准则: MGC 上看不到该 MG TermA 的信息
测试说明: 无

测试编号: MG_SC_21
测试项目: MG 终结点请求重新进入服务
预置条件: MG 已经脱离业务
测试过程: 步骤 1 MG 产生 ServiceChange 命令, 方法为 Restart, 终结点在 NULL 关联; 步骤 2 MGC 应答, TermA 立即进入服务。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>ServiceChange TermA Method=Restart</div><div>Reply</div></div>
判决准则: 在 MG 上可以查询到 TermA 已经进入业务运行态
测试说明: 无

测试编号：MG_SC_22
测试项目：MGC 要求 MG 的终结点返回服务
预置条件： MG 已经脱离业务
测试过程： 步骤 1 MGC 产生 ServiceChange 命令，方法为 Restart，终结点在 NULL 关联； 步骤 2 MG 应答，TermA 立即返回服务。 测试消息流程如下所示： <div><div>MG</div><div>MGC</div><div>ServiceChange TermA Method=Restart</div><div>Reply</div></div>
判决准则： 在 MG 上可以查询到 TermA 已经进入业务运行态
测试说明： 无

测试编号: MG_SC_23
测试项目: MG 和 MGC 失去正常通信后又恢复通信
预置条件: MG 已经在 MGC 上注册成功, 在过段时间后断开 MG 与 MGC 之间的通信然后恢复
测试过程: 步骤 1 MG 失去和 MGC 的正常通信后又恢复; 步骤 2 MG 通过 ServiceChange 向 MGC 报告失去联系, method=Disconnected; 步骤 3 MGC 向 MG 发出 Audit 消息请求, 要求获得 MG 当前的资源状态。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>MG 失去与 MGC 的通信</div><div>ServiceChange ROOT Method=disconn</div><div>Reply</div><div>Audit</div><div>Reply</div></div>
判决准则: MGC 能够获得该 MG 的统计信息
测试说明: 无

测试编号：MG_SC_24
测试项目：MGC 切换后 MG 重新注册
预置条件： MG 已经在 MGC 上注册成功
测试过程： 步骤 1 媒体网关控制器发送 ServiceChange 消息，ServiceChange 采用 ROOT 终结点标志符，Method 设置为 Handoff,并指定尝试 MGC 的 MgcIdToTry。 步骤 2 媒体网关应答媒体网关控制器的命令。 步骤 3 MG 向由主要 MGC 指定的 MGC 发送 ServiceChange 消息，消息中 Method 设置为 Handoff,Reason 可设置为"MG Directed Change"。 步骤 4 替代 MGC 响应 MG 的请求。 测试消息流程如下所示： <pre>sequenceDiagram participant MG participant MGC1 participant MGC2 MGC1->>MG: ServiceChange ROOT Method=Handoff MG->>MGC1: Reply MG->>MGC2: ServiceChange ROOT Method=Handoff MGC2->>MG: Reply</pre>
判决准则： MG 可以在 MGC2 上注册成功
测试说明： 无

测试编号: MG_SC_25
测试项目: 主备 MG 倒换 (可选)
预置条件: 两个备用 MG 已经在 MGC 上注册成功并配置为冗余备份
测试过程: 步骤 1 主 MG 退出服务, MGC 收到 MG 的 ServiceChange 消息, 其中 method=failover; 步骤 2 MGC 将备用 MG 作为 Active 的 MG. 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div> -----ServiceChange ROOT Method= failover-----> </div><div> <-----Reply----- </div></div>
判决准则: 该 MG 上的业务仍然能够被保持
测试说明: 无

测试编号：MG_SC_26
测试项目：MGC 拒绝注册后 MG 重注册
预置条件： 配置两个备用 MGC
测试过程： 步骤 1 MG 向主 MGC 注册时，主 MGC 不接受，返回备用 MGC 的地址； 步骤 2 MG 再根据所返回地址向备用 MGC 注册。ServiceChange 命令中 method=Restart。 测试消息流程如下所示： <pre>sequenceDiagram participant MG participant MGC1 participant MGC2 MG->>MGC1: ServiceChange ROOT Method=Restart MGC1-->>MG: Reply (mgcidtotry) MG->>MGC2: ServiceChange ROOT Method= Restart MGC2-->>MG: Reply</pre>
判决准则： MG 能够在 MGC2 上注册成功
测试说明： 无

测试编号：MG_SC_27
测试项目：协议版本协商
预置条件： 在媒体网关和 MGC 上配置相应的数据
测试过程： 步骤 1 MG 以 ServiceChange 命令向 MGC 注册，其中带了所支持的版本信息； 步骤 2 MGC 发现 MG 的版本号比自己高，返回自己所支持的版本号。 测试消息流程如下所示： <div><div>MG</div><div>主 MGC</div><div> ----- ServiceChange ROOT (ServiceChangeVersion1) -----> </div><div> <----- Reply (ServiceChangeVersion2) ----- </div></div>
判决准则： MGC 返回自己所支持的版本号
测试说明： 无

测试编号：MG_SC_28（适用于 H.248v3 版本）
测试项目：协议版本协商
预置条件： MGC 支持 H.248 版本 3/2/1，MG 支持 H.248 版本 1/2
测试过程： 步骤 1 MG 以 ServiceChange 命令向 MGC 注册，其中带了所支持的版本信息； 步骤 2 MGC 发现 MG 的版本号比自己高，返回自己所支持的版本号。 测试消息流程如下所示： <div><div>MG</div><div>主 MGC</div><div> -----ServiceChange ROOT (Version1)-----> </div><div> <-----Reply (Version1)----- </div><div> </div><div> -----ServiceChange ROOT (Version2)-----> </div><div> <-----Reply (Version2)----- </div></div>
判决准则： MGC 返回自己所支持的版本号，MG 根据协商结果使用的 H.248 版本
测试说明： 无

测试编号: MG_SC_29 (适用于 H.248v3 版本)
测试项目: MGC 响应 MG 的注册请求 (MGC Directed Change)
预置条件: MG 加电启动向 MGC 发送注册请求
测试过程: 步骤 1 MG 向 MGC 发送注册请求, ServiceChange 命令中的 method=Handoff, Termination=Root, ServiceChangeReasons=903 (MGC Directed Change); 步骤 2 MGC 接受注册, 返回 Reply. 测试消息流程如下所示: <div><div>MG</div><div>MG</div><div> ----- ServiceChange ROOT Method= Handoff -----> </div><div> <----- Reply ----- </div></div>
判决准则: MG 应能够在 MGC 上成功注册
测试说明: 无

测试编号: MG_SC_30
测试项目: Service Restoration
预置条件: MG 已经成功注册在 MGC 上
测试过程: 步骤 1 在运行期间, 主 MGC 向 MG 发 ServiceChange 命令中 method= Restart, ServiceChangeReason=900 (“Service Restored”); 步骤 2 MG 收到 MGC 的 ServiceChange 命令, 也通过 ServiceChange 命令发起 Service Restored, ServiceChangeMethod= “Restart”。
测试消息流程如下所示: <div><div>MG</div><div>MGC1</div><div><div>ServiceChange ROOT Method= Restart</div><div>Reply</div><div>ServiceChange ROOT Method= Restart</div><div>Reply</div></div></div>
判决准则: MG 发起根据 MGC 下发的 Service Restoration 发起重注册, 注册成功
测试说明: 无

测试编号: MG_SC_31
测试项目: MG 请求 ROOT 根节点渐进退出服务 delay≠null
预置条件: MG 已经在 MGC 上注册成功
测试过程: 步骤 1 MG 发 ServiceChange 命令, 方法为 graceful, 并设 delay 且 delay≠null。 步骤 2 在 delay 时间后, MG 将所有端点的状态全部置为 out of service。 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div>ServiceChange root Method=graceful Delay=x</div><div>Reply</div></div>
判决准则: MG 上报所有的端点不可用, MG 退出服务
测试说明: 无

测试编号: MG_SC_32
测试项目: Capability Change
预置条件: MG 已经在 MGC 上注册成功
测试过程: 步骤 1 MG 发送 ServiceChange, ServiceChangeReason= 916 ("Packages Change") or 917 ("Capabilities Change"); 步骤 2 MGC 发送 Audit 消息到 MG 检测 MG 的 capabilities. 测试消息流程如下所示: <div><div>MG</div><div>MGC</div><div> ----- ServiceChange ROOT Method= Restart -----> </div><div> <----- Reply ----- </div><div> <----- AuditCapability ----- </div><div> ----- Reply -----> </div></div>
判决准则: MG 上报业务能够被 MGC 正常处理
测试说明: 无

6.2 各种包的定义和应用

6.2.1 Generic 包

测试内容同 5.2.1 节，但测试项中均由测试仪充当 MGC。

6.2.2 基本 Root 包

测试内容同 5.2.2 节，但测试项中均由测试仪充当 MGC。

6.2.3 基本导通测试包

测试内容同 5.2.3 节，但测试项中均由测试仪充当 MGC。

6.3 描述符的定义和应用

6.3.1 本端控制描述符

测试内容同 5.3.1 节，但测试项中均由测试仪充当 MGC。

6.3.2 事件描述符

测试内容同 5.3.2 节，但测试项中均由测试仪充当 MGC。

6.3.3 Modify 命令所支持的描述符

测试内容同 5.3.3 节，但测试项中均由测试仪充当 MGC。

6.3.4 Add 命令所支持的描述符

测试内容同 5.3.4 节，但测试项中均由测试仪充当 MGC。

6.3.5 Move 命令所支持的描述符

测试内容同 5.3.5 节，但测试项中均由测试仪充当 MGC。

6.3.6 Subtract 命令所支持的描述符

测试内容同 5.3.6 节，但测试项中均由测试仪充当 MGC。

6.3.7 AuditValue 命令所支持的描述符

测试内容同 5.3.7 节，但测试项中均由测试仪充当 MGC。

6.3.8 AuditCapability 命令所支持的描述符

测试内容同 5.3.8 节，但测试项中均由测试仪充当 MGC。

6.4 通配符的应用

测试内容同 5.4 节，但测试项中均由测试仪充当 MGC。

6.5 定时器的应用

测试内容同 5.5 节，但测试项中均由测试仪充当 MGC。

6.6 错误码

测试编号: MG_Error-Code_1
测试项目: 返回错误码 400
预置条件: MG 已经向 MGC 注册
测试过程: 步骤 1 MGC 向 MG 发送命令 H.248 消息, 消息中的事务请求有语法错误。 步骤 2 MG 丢弃该消息并向 MGC 报告错误码 400 (Bad Request)
判决准则: MGC 接收到的 Reply 消息中带错误码 400
测试说明: 无

测试编号: MG_Error-Code_2
测试项目: 返回错误码 403
预置条件: MG 已经向 MGC 注册
测试过程: 步骤 1 MGC 向 MG 发送带有空事务请求的 H.248 消息。 步骤2 MG处理失败, 返回错误码403 (Syntax Error in Transaction)
判决准则: MGC 接收到的 Reply 消息中带错误码 403
测试说明: 无

测试编号: MG_Error-Code_3
测试项目: 返回错误码 406
预置条件: MG 已经向 MGC 注册
测试过程: 步骤 1 MGC 向 MG 发送 H.248 消息, 消息中的版本参数为 MG 所不支持的。 步骤2 MG返回错误码406 (Version Not Supported)
判决准则: MGC 接收到的 Reply 消息中带错误码 406
测试说明: 无

测试编号: MG_Error-Code_4
测试项目: 返回错误码 410
预置条件: MG 已经向 MGC 注册
测试过程: 步骤 1 MGC 向 MG 发送 H.248 消息, 消息中的 mId, transactionId, contextId, TerminationId, PropertyId, EventId, SignalId, StatisticsId, ParameterId 等长度不对或有非法字符。 步骤2 MG返回错误码410 (Incorrect identifier)
判决准则: MGC 接收到的 Reply 消息中带错误码 410
测试说明: 无

测试编号: MG_Error-Code_5
测试项目: 返回错误码 411
预置条件: MG 已经向 MGC 注册
测试过程: 步骤 1 MGC 向 MG 发送 H.248 消息, 消息中的 ContextID 不能识别。 步骤2 MG返回错误码411 (The transaction refers to an unknown ContextId)
判决准则: MGC 接收到的 Reply 消息中带错误码 411
测试说明: 无

测试编号: MG_Error-Code_6
测试项目: 返回错误码 412
预置条件: MG 已经向 MGC 注册且 MG 的 SCN 侧资源全部被占用
测试过程: 步骤 1 设置 MG 所支持的最大的上下文数为 1。 步骤 2 模拟 MGC 向 MG 发送 Add 命令, 请求建立一个新的关联, 然后再试图建立一个新的关联。 步骤 3 MG 返回错误码 412 (No ContextIDs available)
判决准则: MGC 接收到的 Reply 消息中带错误码 412
测试说明: 无

测试编号: MG_Error-Code_7
测试项目: 返回错误码 421
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 H.248 消息, 收到的事务请求中含有不能识别的动作或非法的动作组合。 步骤2 MG返回错误码421(Unknown action or illegal combination of actions)
判决准则: MGC 接收到的 Reply 消息中带错误码 421
测试说明: 无

测试编号: MG_Error-Code_8
测试项目: 返回错误码 422
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 H.248 消息, 收到的请求中含有非法/语法错误的 ContextID。 步骤2 MG返回错误码422(Syntax Error in Action)
判决准则: MGC 接收到的 Reply 消息中带错误码 422
测试说明: 无

测试编号: MG_Error-Code_9
测试项目: 返回错误码 430
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 ADD 命令添加一个终结点到一个上下文中。 步骤2 MGC 向 MG 发送 H.248 命令 (如 Modify、ServiceChange 等), 修改该终结点的 TermnationID 为一个未知值。 步骤3 MG返回错误码430 (Unknown TerminationID)
判决准则: MGC 接收到的 Reply 消息中带错误码 430
测试说明: 无

测试编号: MG_Error-Code_10
测试项目: 返回错误码 431
预置条件: MG 已经向 MGC 注册
测试过程: 步骤 1 MGC 向 MG 发送 Audit 命令, 在所有空关联和非空关联中审计某一带有通配符的 TerminationID, 该 TerminationID 在 MG 中并不存在。 步骤2 MG返回错误码431 (No TerminationID matched a wildcard)
判决准则: MGC 接收到的 Reply 消息中带错误码 431
测试说明: 无

测试编号: MG_Error-Code_11
测试项目: 返回错误码 432
预置条件: MG 已经向 MGC 注册且 MG 上的可用 TerminationID 已经被分配完
测试过程: 步骤 1 设置 MG 所支持的最终结点数为 1。 步骤 2 模拟 MGC 向 MG 发送 Add 命令, 占用一个终结点, 然后再试图占用另一个终结点。 步骤 3 MG 返回错误码 432 (Out of TerminationIDs or No TerminationID available)
判决准则: MGC 接收到的 Reply 消息中带错误码 432
测试说明: 无

测试编号: MG_Error-Code_12
测试项目: 返回错误码 433
预置条件: MG 已经向 MGC 注册
测试过程: 步骤 1 在 MG 上占用一个 TerminationA。 步骤 2 MGC 向 MG 发送 Add 命令且该命令包含 TerminationA (即发送同样的 H.248 消息, 但消息的 TransactionID 不一样)。 步骤3 MG返回错误码433 (TerminationID is already in a Context)
判决准则: MGC 接收到的 Reply 消息中带错误码 433
测试说明: 无

测试编号: MG_Error-Code_13
测试项目: 返回错误码 440
预置条件: MG 已经向 MGC 注册且 MG 上的可用 TerminationID 已经被分配完
测试过程: 步骤1 MGC 向 MG 发送 H.248 消息, 该消息中的 property、parameter、event、signal 或 statistic 标识中的包名是 MG 所不支持的。 步骤2 MG 返回错误码440 (Unsupported or unknown Package)
判决准则: MGC 接收到的 Reply 消息中带错误码 440
测试说明: 无

测试编号: MG_Error-Code_14
测试项目: 返回错误码 442
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 H.248 消息, 该命令中的语法不正确。 步骤2 MG 返回错误码442 (Syntax Error in Command)
判决准则: MGC 接收到的 Reply 消息中带错误码 442
测试说明: 无

测试编号: MG_Error-Code_15
测试项目: 返回错误码 443
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送无法确认的消息, 如 InvalidCommand。 步骤2 MG 返回错误码443 (Unsupported or Unknown Command)
判决准则: MGC 接收到的 Reply 消息中带错误码 443
测试说明: 无

测试编号: MG_Error-Code_16
测试项目: 返回错误码 444
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 H.248 消息, 消息请求中含有不能识别的描述符。 步骤2 MG返回错误码444 (Unsupported or Unknown Descriptor)
判决准则: MGC 接收到的 Reply 消息中带错误码 444
测试说明: 无

测试编号: MG_Error-Code_17
测试项目: 返回错误码 445
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 H.248 消息, 消息中含有不能识别的属性。 步骤2 MG返回错误码445 (Unsupported or Unknown Property)
判决准则: MGC 接收到的 Reply 消息中带错误码 445
测试说明: 无

测试编号: MG_Error-Code_18
测试项目: 返回错误码 446
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 H.248 消息, 消息请求中含有不能识别的参数或漏掉与描述符相关的命令。 步骤2 MG返回错误码446 (Unsupported or Unknown Parameter)
判决准则: MGC 接收到的 Reply 消息中带错误码 446
测试说明: 无

测试编号: MG_Error-Code_19
测试项目: 返回错误码 447
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 AuditCapabilities 命令, 命令中含有 MG 不支持的 PackagesDescriptor。 步骤2 MG返回错误码447 (Descriptor not legal in this command)
判决准则: MGC 接收到的 Reply 消息中带错误码 447
测试说明: 无

测试编号: MG_Error-Code_20
测试项目: 返回错误码 448
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 H.248 消息, 消息请求中含有重复的描述符。 步骤2 MG返回错误码448 (Descriptor appears twice in a command)
判决准则: MGC 接收到的 Reply 消息中带错误码 448
测试说明: 无

测试编号: MG_Error-Code_21
测试项目: 返回错误码 451
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 H.248 消息, 消息中的 Event Name 是 MG 所支持包的版本不支持的。 步骤2 MG返回错误码451 (No such event in this package)
判决准则: MGC 接收到的 Reply 消息中带错误码 451
测试说明: 无

测试编号: MG_Error-Code_22
测试项目: 返回错误码 452
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 H.248 消息, 消息中的 Signal Name 是 MG 所支持包的版本不支持的。 步骤2 MG返回错误码452 (No such signal in this package)
判决准则: MGC 接收到的 Reply 消息中带错误码 452
测试说明: 无

测试编号: MG_Error-Code_23
测试项目: 返回错误码 453
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 H.248 消息, 消息中的 Statistic Name 是 MG 所支持包的版本不支持的。 步骤2 MG返回错误码453 (No such statistic in this package)
判决准则: MGC 接收到的 Reply 消息中带错误码 453
测试说明: 无

测试编号: MG_Error-Code_24
测试项目: 返回错误码 454
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 H.248 消息, 消息中的 parameter value 是 MG 所支持包的版本不支持的。 步骤2 MG返回错误码454 (No such parameter value in this package)
判决准则: MGC 接收到的 Reply 消息中带错误码 454
测试说明: 无

测试编号: MG_Error-Code_25
测试项目: 返回错误码 455
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 H.248 消息, 消息中的 property 有不合法的参数。 步骤2 MG 返回错误码455 (Parameter illegal in this Descriptor)
判决准则: MGC 接收到的 Reply 消息中带错误码 455
测试说明: 无

测试编号: MG_Error-Code_26
测试项目: 返回错误码 456
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 H.248 消息, 消息中的 parameter 或 property 在描述符 property 中出现两次。 步骤2 MG 返回错误码 456 (Parameter or Property appears twice in this Descriptor)
判决准则: MGC 接收到的 Reply 消息中带错误码 456
测试说明: 无

测试编号: MG_Error-Code_27 (可选)
测试项目: 返回错误码 471
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 ADD 命令, 在某一个上下文中加入一个复用描述符终结点, 该终结点不可用。 步骤2 MG 返回错误码471(Implied Add for Multiplex failure)
判决准则: MGC 接收到的 Reply 消息中带错误码 471
测试说明: 无

测试编号: MG_Error-Code_28
测试项目: 返回错误码 501
预置条件: MG 向 MGC 注册
测试过程: 步骤1 MGC向MG发送Modify消息, 其中event 为al/fl, 结果未被执行。 步骤2 MG返回错误码502 (Not Implemented)
判决准则: MGC 接收到的 Reply 消息中带错误码 502
测试说明: 无

测试编号: MG_Error-Code_29
测试项目: 返回错误码 502
预置条件: MG 未向 MGC 注册
测试过程: 步骤1 MG向MGC注销, MG还未向MGC发送注册请求。 步骤2 MGC向MG发送H.248消息 (如Add、Modify、ServiceChange命令等)。 步骤3 MG返回错误码502 (Not ready)
判决准则: MGC 接收到的 Reply 消息中带错误码 502
测试说明: 无

测试编号: MG_Error-Code_30
测试项目: 返回错误码 505
预置条件: MG 未向 MGC 注册
测试过程: 步骤1 MG向MGC注销。 步骤2 MG向MGC发ServiceChange注册请求, MGC在应答MG注册请求之前向MG发送其他的H.248消息。 步骤3 MG返回错误码505 (Command Received before Restart Response)
判决准则: MGC 接收到的 Reply 消息中带错误码 505
测试说明: 无

测试编号: MG_Error-Code_31
测试项目: 返回错误码 515
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 ADD 命令, 命令中指示媒体流类型 ST=1。 步骤2 MG响应MGC, 返回Reply且指示ST=1。 步骤3 MGC向MG发送Modify命令, 命令中指示ST≠1。 步骤4 MG返回错误码515 (Unsupported Media Type)
判决准则: MGC 接收到的 Reply 消息中带错误码 515
测试说明: 无

测试编号: MG_Error-Code_32
测试项目: 返回错误码 517
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MGC 向 MG 发送 H.248 消息, 消息中的 Media 描述符或 LocalControl 描述符中设置的模式是 MG 所不支持的。 步骤2 MG返回错误码517 (Unsupported or invalid mode)
判决准则: MGC 接收到的 Reply 消息中带错误码 517
测试说明: 无

测试编号: MG_Error-Code_33
测试项目: 返回错误码 520
预置条件: MG 已经向 MGC 注册
测试过程: 步骤1 MG 发起呼叫, MGC 向 MG 发送 Modify 命令, 命令中的 digit map 名在 MG 中没有定义。 步骤2 MG返回错误码520 (DigitMap Media Gateway does not have a digit map)
判决准则: MGC 接收到的 Reply 消息中带错误码 520
测试说明: 无

6.7 杂项

测试编号: MG_Other_1
测试项目: MG 支持的编码方式
预置条件: MG 与 MGC 上都配置二进制编码方式和文本编码方式
测试过程: 步骤1 MG采用二进制编码方式向MGC发送注册请求。 步骤2 MG采用文本方式, 向MGC发起注册请求
判决准则: 在两种编码方式下, MG 都能够接收到 MGC 的 Reply 消息
测试说明: 该测试项只适用于 RG

测试编号: MG_Other_2
测试项目: MG 对事务请求最多处理一次
预置条件: MG 已经注册到 MGC 上
测试过程: 步骤1 MGC 向 MG 同时发送多个一样的事务请求; 步骤2 MG只做一次处理, 回应一个Reply消息
判决准则: MGC 只能接收到 MG 的一个 Reply 消息
测试说明 无

测试编号: MG_Other_3
测试项目: MG 的三次握手过程
预置条件: MG 已经注册到 MGC 上
测试过程: 步骤1 MGC向MG发送Modify消息。 步骤2 MG返回Reply
判决准则: 观察 MG 上的 TranscationID
测试说明 无

测试编号: MG_Other_4
测试项目: NULL 关联
预置条件: MG 已经注册到 MGC 上
测试过程: 步骤1 终结点 Term A 未使用, 处于空关联中 步骤2 MGC向MG发送Audit命令, 查询MG上未使用的终结点的关联状态
判决准则: 未使用的终结点处于 NULL 关联
测试说明: 无

测试编号: MG_Other_5
测试项目: 提供点到点的 MG
预置条件: MG 已经注册到 MGC 上
测试过程: 步骤1 MG只支持点到点的连接, 即一个关联中只有2个终结点。 步骤2 MGC向MG发送Add命令, 在一个关联中加入两个终结点A、B。 步骤3 MGC在向该关联中加入另外一个终结点C
判决准则: MG 向 MGC 发送 434 响应
测试说明: 无

测试编号: MG_Other_6(可选)
测试项目: 提供多点会议的 MG
预置条件: MG 已经注册到 MGC 上
测试过程: 步骤1 MG只支持多点连接, 一个关联中可以有3个以上的终结点。 步骤2 MGC向MG发送Add命令, 在一个关联中加入两个终结点A、B。 步骤3 MGC在向该关联中加入另外一个终结点C
判决准则: 终结点加入成功, 可以查询到所有终结点的状态
测试说明: 无

测试编号: MG_Other_7
测试项目: 处于 NULL 中的终结点
预置条件: MG 已经注册到 MGC 上
测试过程: 步骤1 MG发起呼叫, 呼叫成功。 步骤2 MGC发起命令修改呼叫所使用的Termination的属性。 步骤3 MGC发起命令修改NULL关联中的任意Termination的属性
判决准则: 在通话中的 Termination 的属性不能被修改
测试说明: 无

测试编号: MG_Other_8
测试项目: Topology 描述符 (可选)
预置条件: MG 已经注册到 MGC 上
测试过程: 步骤1 MGC向MG发送Add命令, 加入两个终结点A和B, 其中的Topology描述符 {Term1,Term2, Bothway }。 步骤2 MG回应Add命令
判决准则: 在 MG 的回应中有 Topology 描述符
测试说明: 无

6.8 安全

测试内容同 5.8 节, 但测试项中均由测试仪充当 MGC。

中 华 人 民 共 和 国
通 信 行 业 标 准
基于 H.248 的媒体网关控制协议测试方法
YD/T 2422-2012

*

人民邮电出版社出版发行
北京市崇文区夕照寺街 14 号 A 座
邮政编码：100061
宝隆元（北京）印刷技术有限公司印刷
版权所有 不得翻印

*

开本：880×1230 1/16 2013 年 3 月第 1 版
印张：8.5 2013 年 3 月北京第 1 次印刷
字数：237 千字

15115 • 38

定价：95 元

本书如有印装质量问题，请与本社联系 电话：(010)67114922