

# 中国联通软交换试验网 媒体网关控制协议（MGCP）技术规范

信息产业部电信传输研究所  
2002. 12



## 目次

前 言.....	i
1 范围.....	1
2 规范性引用文件.....	1
3 术语和定义.....	1
4 缩略语.....	2
5 呼叫模型 (model) .....	3
6 命令.....	7
7 协议消息格式.....	12
8 协议传送要求.....	20
9 协议安全要求.....	22
10 包 (Package) .....	22
附录 A (规范性附录) 本标准协议语法的 ABNF 描述.....	24



## 前 言

MGCP 协议是下一代分组网中语音业务、数据业务和视频业务呼叫、控制、业务提供的控制设备（软交换设备和 MGC）与受控制设备（MG）之间的接口协议。本标准是以国际电联、计算机标准化组织、软交换论坛制定的相关标准为基础，结合国内网络的实际情况和相关国内标准制定的。

本标准由中国联合通信有限公司提出并归口。

本标准起草单位：

本标准主要起草人：



# 中国联通媒体网关控制协议（MGCP）技术规范

## 1 范围

本标准规定了媒体网关和媒体网关控制器（或软交换设备）之间使用 MGCP 协议进行通信时的协议要求。

本标准适用于媒体网关、媒体网关控制器和软交换设备的研制、开发和引进。

## 2 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本标准，然而，鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本标准。

YDC 003-2001	软交换设备总体技术要求
RFC 3435(2002)	MGCP协议
RFC 2327(1999)	SDP协议
RFC 1819	RTP 协议
RFC 2327	SDP 协议
RFC 2401	IP 协议安全机制
RFC 2402	AH 协议
RFC 2406	ESP 协议
RFC 2234	ABNF 语法定义规则

## 3 术语和定义

**软交换设备 (Softswitch)：**是电路交换网向分组网演进的核心设备，也是下一代电信网络的重要设备之一，它独立于底层承载协议，主要完成呼叫控制、媒体网关接入控制、资源分配、协议处理、路由、认证、计费等主要功能，并可以向用户提供现有电路交换机所能提供的业务以及多样化的第三方业务。

**媒体网关控制器 (Media Gateway Controller)：**负责对媒体网关中与媒体通道连接控制相关的呼叫状态部分进行控制。

**媒体网关 (Media Gateway)：**媒体网关将一种网络中的媒体转换成另一种网络所要求的媒体格式。例如：媒体网关能够在电路交换网的承载通道和分组网的媒体流之间进行转换、可以处理音频、视频或者T. 120，也可以具备处理这三者的任意组合的能力、能够进行全双工的媒体翻译、可以演示视频/音频消息，实现其它IVR功能、也可以进行媒体会议等。

**驻地网关 (Residential Gateway)：**位于用户侧，将模拟线与分组网络相连的媒体网关。

**中继网关 (Trunk Gateway)：**在电路交换网和分组网络之间的媒体网关，用来终结大量的数字电路。

**信令网关 (Signaling Gateway )：**连接七号信令网与IP网的设备，主要完成PSTN/ISDN侧的七号信令与IP网侧信令的转换功能。

**端点 (Endpoint)：**媒体网关中的数据源或数据宿，可以是物理端点，如模拟线、中继电路等，也可以是虚拟端点，如语音服务器的信号源。

**连接 (Connection)** :互相发送数据端点之间的一种关联, 连接可以是点到点或点到多点, 连接可以建立在IP网/ATM网上。

**事物交互(Transaction)**: MG 与 MGC 之间的一组命令请求和命令响应构成事物交互。

#### 4 缩略语

AAL	ATM 适配层
ABR	可用比特率
AH	认证头
AS	公告服务器
ATM	异步传输模式
AUCX	审计连接
AUEP	审计端点
BRI	基本速率接口
CAS	随路信令
CBR	恒定比特率
CRCX	创建连接
DC	数字通道
DLCX	删除连接
DTMF	双音多频
ESP	封装安全净荷
EPCF	端点配置
IANA	因特网号码分配机构
IKE	密钥交换
IP	因特网协议
IVR	交互式语音应答
MGCP	媒体网关控制协议
MDCX	修改连接
MF	多频
MG	媒体网关
MGC	媒体网关控制器
NTFY	通知
PSTN	公共电话交换网
QoS	服务质量
RQNT	通知请求
RTP	实时传输协议
RSIP	重启动进展
SCN	电路交换网
SDP	会话描述协议
SDU	业务数据单元
SG	信令网关
SS7	7号信令系统
TCP	传输控制协议
TDM	时分(多路)复用
UDP	用户数据报协议

VCI	虚通道识别码
VPC	虚通道连接
VPI	虚通道识别码

## 5 呼叫模型 (model)

### 5.1 概述

本协议定义的呼叫模型主要包括端点 (Endpoint) 和连接 (Connection) 两个逻辑概念。通常，一次呼叫是由一个或多个连接组成的。事件 (Event) 和信号 (Signal) 是呼叫建立和释放过程中所需要的两个重要概念，事件和信号组合成包 (package)，本标准规定特定端点支持特定类型的包。

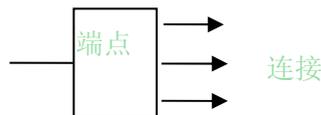


图 1 呼叫连接模型示意图

### 5.2 端点 (Endpoint)

本协议主要定义了8种端点类型：

- 1) 数字通道 (Digit Channel)
- 2) 模拟线 (Analog line)
- 3) 公告服务器接入点 (Announcement sever access point)
- 4) 交互式语音响应 (IVR) 接入点 (Interactive Voice Response access point)
- 5) 会议桥接入点 (Conference bridge access point)
- 6) 分组中继 (Packet Relay)
- 7) ATM中继 (ATM Trunk Side Interface)

除以上7种主要的端点类型外，本协议还定义了检测终端和帧中继端点两种端点类型，其中检测端点用于检测网络质量，帧中继端点用于对帧中继虚电路上的音频信道的复用进行管理。

#### 5.2.1 数字信道 (Digit channel)

数字信道是指位于数字中继 (Trunk) 和ISDN接口中的一条64Kbit/s的数字电路，支持该端点的网关能够把从该信道接收到A/μ率信号转换成音频数据分组。本协议规定，如果媒体网关同时支持网络接入服务 (NAS) 功能，则媒体网关应能接收该信道上的话带数据 (Modem连接) 或二进制数据 (ISDN连接)，并转换为分组数据。

本协议规定，如果媒体网关中的该信道用于传送信令，如ISDN的D信道或No.7信令的F链路，则媒体网关应能执行信令网关的功能，媒体网关应使用“BACKHAUL”过程与MGC交互信令数据包，“BACKHAUL”过程参见IETF的SIGTRAN工作组。如果所使用的信令为随路信令 (如MF R2)，则媒体网关应能检测并产生相应的信令信号。

#### 5.2.2 模拟线 (Analog Line)

模拟线是指模拟用户线或中继线，可提供电话连接、modem连接。本协议规定，如果媒体网关同时支持NAS功能，则媒体网关应能接收该信道上的话带数据 (Modem连接) 并转换为分组数据。

媒体网关能够在端点和分组网之间或同一网关的多个端点之间建立多个连接，多个连接产生的数据信号必须根据连接模式进行混合。端点所支持的最大连接数由媒体网关的特性决定，支持点到点连接的媒体网关中的端点只能支持一个连接，支持点到多点连接的媒体网关中的端点能支持2个或2个以上的连接，以提供“呼叫等待”或“三方呼叫”等补充业务。

### 5.2.3 公告服务器接入点 (Announcement Sever Access Point)

公告服务器接入点是指用于提供公告接入服务的端点。当接收到来自 MGC 的请求时，公告服务器将播放一条指定的公告。本协议规定，MGC 用于发起公告播放请求的流程应遵循本协议所规定信令流程。

公告服务器接入点端点所建立的连接是单向的，如果一个端点上同时建立了多个连接，则应向所有的连接同时播放相同的公告。通常，采用单工或半双工的方式与公告服务器建立连接，公告服务器接入端点并不监听来自连接的音频信号。

### 5.2.4 IVR 接入点

IVR 接入点是指用于提供交互式语音应答业务的端点。当接收到来自 MGC 的请求时，IVR 服务器将播放公告并监听来自用户的应答响应，例如 DTMF 信号或语音消息。本协议规定，MGC 用于发起 IVR 请求的流程应遵循本协议所规定信令流程。

通常，一个 IVR 接入点同时只能支持一个连接。如果一个 IVR 接入点同时建立了多个连接，则应向所有的连接同时播放相同的公告。

### 5.2.5 会议桥接入点 (Conference Bridge)

会议桥接入点是指能为专门会议提供接入的端点。

媒体网关能够在端点和分组网之间或同一网关内的多个端点之间建立多个连接，从这多个连接起源的信号应该按照连接模式 (Connection Mode) 进行混合。端点所支持的最大连接数由媒体网关的属性决定。

### 5.2.6 分组中继 (Packet Relay)

分组中继是一类只支持两个连接的特殊会议桥端点。通常，该端点位于私网和公网之间的防火墙中，或者位于码型转换服务器中用于提供不兼容网关之间适配和互操作，这些不兼容网关如压缩算法不兼容的网关或分别位于 IP 和 ATM 网络的网关。

### 5.2.7 ATM 中继 (ATM Trunk)

ATM 中继指用一个或多个 ATM PVC。当采用 AAL2 承载方式时，同一虚电路上可复用多个通道，每个通道对应一个 ATM 中继端点。

## 5.3 端点标识

端点标识由两部分组成：本地名和端点所在网关的域名，端点名称的定义方式为<本地名@域名>。域名的定义方式参见 RFC1034，除此之外，还可以采用 IPv4 地址代替域名。本协议规定本地名不区分大小写，具体的本地端点名取决于端点类型，每类端点本地名具有相同的层次结构，以标识所在物理网关的描述符开始，以表示断点的描述符结束。端点本地名描述必须遵循以下规则：

- 1) 描述项之间必须用“/”分隔。
- 2) 每个描述项均为由字母，数字或其他可打印的字符组成的字符串，这些字符除了分隔符“/”和“@”，通配符“\*”和“\$”以及空格键之外的字符。
- 3) 本地名可使用通配符“\*”和“\$”。“\*”表示媒体网关中配置的所有值，这些端点即可以处于服务状态也可以处于非服务状态；“\$”表示媒体网关中配置的任意值，且只能为处于服务状态的端点。

本协议建议 MGC 识别端点标识中的通配符采用“右向性”规则，即如果端点本地名某个单元使用了通配符，则该单元右边的所有单元被通配。

#### 5.4 呼叫和连接 (Calls and Connections)

呼叫由建立在端点上的连接组成，例如数字信道端点 (EP1 和 EP2) 之间建立呼叫，则 MGC 将为这两个端点建立两个连接 (Connection1 和 Connection2)，如图 2 所示。本协议规定连接由唯一的连接标识 (ConnectionID) 进行标识，且每个连接由相应的连接属性来定义。



图 2 呼叫和连接关系示意图

##### 5.4.1 呼叫名称

本协议规定，呼叫由唯一的“呼叫标识符”属性(CallID)来进行标识，该 CallID 应独立于 MGC。本协议规定 MGC 为一个呼叫建立了多个连接后，这些连接应具有相同的 CallID，这些连接可以分别位于相同或不同的媒体网关中。此外，CallID 可以作为计费和管理标识。

##### 5.4.2 连接名称

当 MGC 请求 MG 创建连接时，媒体网关应创建连接标识符。本协议建议连接标识符由十六进制字符串组成，最长可为 32 位，且当某个连接终结后，至少过三分钟后 MG 才能将已经使用过的连接标识符分配给同一端点上建立的新连接，通过这种方式，媒体网关可实现连接标识符在媒体网关内的唯一。

#### 5.5 资源管理和连接属性

资源类型和连接有关，本协议将资源分为两类：

- 1) 外部可见资源：将影响网络中的数据流格式，在连接中必须和对端交换此信息。
- 2) 内部资源：决定端点接收信号如何处理和将何种信号发往连接。

连接资源由媒体网关根据 MGC 的命令来进行分配，MGC 所使用的命令由两套参数组成：

- 1) 本地指示命令网关选择连接资源。
- 2) 其它连接提供的“会话描述”，即远端会话描述。

该命令包括的会话参数有：连接模式、优选编码方式或分组化方法，回声抵消或静音压缩等。对每种参数，MGC 可以一个值、一个值范围或者值为空。对应不同程度的控制，网关有不同程度的选择自由度。网关以本地会话描述所确定的参数值来决定某次连接的资源分配。在允许的情况下，网关应会选择与远端会话描述相一致的值，但并不要求参数绝对相同。

资源分配完成后，网关应创建一个“会话描述”来说明发送和接收分组的方式，会话描述可以给定一个范围值，如一个编码算法的列表。

#### 5.6 本地连接的专用事件

大型网关通常包含大量不同类型的端点，因此，有时会在同一网关的两个端点之间创建连接，称为本地连接。本地连接比网间连接简单。有时，本地连接会通过本地互连设备建立，比如 TDM 总线。同一网关内两个端点只需 MGC 发送一条创建连接命令，就可创建本地连接。

#### 5.7 MGC 命令方式

为了提高网络可靠性，网络中可部署冗余的 MGC。MGC 命令方式与端点命令方式相似，由本地名和域名两部分组成。本标准建议采用以下规则来提高网络可靠性。

- 1) 可以由域名而不是 IP 地址来鉴别 MGC 或端点等网络实体，一个域名可对应多个地址。当命令响应或命令请求不能前传至某个网络地址，就试图使用其它网络地址。
- 2) 网络实体可转移至另一平台。域名和实际平台的关联将保留在 DNS 服务器中。MGC 和 MG 必须从 DNS 服务器读取关联的生存期，超过生存期的关联应进行刷新。

除了使用域名和 DNS 之外，使用“Notifiedentity”命令参数也可以实现可靠性。“Notifiedentity”对于端点而言就是当前控制该端点的 MGC。本标准规定一个端点在任一时刻只和一个“Notifiedentity”相关联。端点向 MGC 发送命令请求时，必须向当前的“Notifiedentity”发送命令。MG 发生重启动时，“Notifiedentity”应被设成缺省值。MGC 向 MG 发送的多数命令都能用“Notifiedentity”参数来明确消息的接收方。“Notifiedentity”的值将被保持到端点接收到新的“Notifiedentity”参数为止，或者端点进行冷热启动为止。

如果消息中“Notifiedentity”参数为空，则就被视为空值。如果端点的“Notifiedentity”参数为空或未明确指定，则默认的将接收到最后一条非审计命令的源地址作为“Notifiedentity”。审计命令不应改变“Notifiedentity”的值。本标准建议不使用“Notifiedentity”为控制的消息。

## 5.8 数字映射 (DigitMap)

MGC 可请求 MG 收集用户拨打的号码。本标准规定，MG 把收集的拨号号码置于一个缓冲区，等收集号码完全后再一起发送。本标准定义的 DigitMap 就是用于定义一个号码映射规则，用于实现对拨号号码的收集发送。当号码采集规则如下所示：

11X	紧急呼叫和特服呼叫
6XXXXXXXX	本地号码
0	长途号码
00	国际长途
*xx	补充业务

则 DigitMap 所定义的数字规则可以如下描述：

{11x | 6 XXXXXXXX | 0[1—9]XXX. | 00XXX. | Exx}

数字映射语法规则厂家附录 A。其中，支持基本数字映射的字母是必要的，支持扩展数字映射的字母是任选的。当 MG 接收到不支持的扩展数字映射将返回错误代码为 537 的错误响应。数字映射由一系列数字字符串组成，其所允许的字符包括：

- 1) 数字：0~9
- 2) 定时字母：“T”表示在该处检测到定时器超时。
- 3) DTMF：0~9, T, 或 A, B, C, D, #,
- 4) 通配符：x 可为 0~9 中任意一个。
- 5) 范围：[ ] 中的一个或多个 DTMF 字符。
- 6) 子域：包含用“—”相隔的两个数字并匹配。子域只在表示范围时使用。
- 7) 位置：“.”表示紧随其前的字母可以出现任意多个，包括零个。

网关用以下步骤检测事件是否与数字映射相匹配。

- 1) 将号码事件加到“当前拨号串”的内部状态变量后作标记。
- 2) 查找数字映射表，是否有与“当前拨号串”相符的表达式。
- 3) 如果查找到有匹配项（与且仅与一个部分相匹配），就将当前拨号串送往 MGC。

数字字符串可以完全匹配，也可以“不匹配”（impossible matches）。当定时器超时，则 MG 判定为“不匹配”。当 MG 检测到完全匹配和不匹配地拨号事件都会向 MG 发送通知请求消息。

## 5.9 封包 (Packages)

本标准规定包可以实现对 MGCP 协议的扩展。一个给定的端点可以支持一个或多个封包，不同类型的封包可以支持不同类型的事件和信号，MGCP 中的包定义包含以下几个部分：

- 1) BearerInformation
- 2) LocalConnectionOptions
- 3) ExtensionParameters
- 4) ConnectionModes

- 5) Events
- 6) Signals
- 7) Actions
- 8) DigitMapLetters
- 9) ConnectionParameters
- 10) RestartMethods
- 11) ReasonCodes
- 12) ReturnCodes

支持一种或多种类型包的端点可以将其中的一种类型定义成默认包类型。端点默认包中的事件和信号名称的使用是任选的。除DigitMap以外的其它类型的包扩展名，应通过包名进行识别，扩展名之前必须加上封包名。包名是由字母、“-”字符和数字组成的字符串，且不区分大小写，第一和最后一个位置不能为“-”字符。实验包的名称必须以字符“x-”开头，IANA不能注册以这两个字符开头的封包，以“x+”开头的也被保留。如果网关收到不支持的包类型应必须返回错误代码为518的响应消息。

## 5.10 事件和信号 (Event and Signal)

事件和信号分别在包中进行定义，事件和信号可用“包/事件名(或信号名)”进行标识。包名和事件名(或信号名)都是由数字字母串进行标识，其定义与DigitMap的定义方式类似，且可使用通配符。

包中定义的事件，应描述以下信息：

- 1) 事件描述及目的
- 2) 事件的详细特征
- 3) 事件的典型时长和最大时长

信号的类型可分为三类：

1) 通/断型 (On/Off)：信号一旦加载，这些信号永远存在直至信号被断开为止。信号断开可能是由于另一事件发生或者由一个新的信号请求而引起。

2) 超时型 (TO)：信号一旦加载，这些信号将持续到被断开或者规定的事件结束未知。信号断开可能是由于另一事件发生或者由一个新的信号请求而引起。信号超时终止可生成一个信号操作完成事件。

3) 短暂型 (BR)：信号持续的事件非常短。它们会自行终止。即使另外一个事件产生，该信号也不会终止，但是当一个新的信号请求到达时，该信号将终止。

## 6 命令

### 6.1 概述

命令主要由连接处理命令和端点处理命令组成，本协议共定义了九个命令：

- 1) EndpointConfiguration
- 2) NotificationRequest
- 3) Notify
- 4) CreateConnection
- 5) ModifyConnection
- 6) DeleteConnecation
- 7) AuditEndpoint
- 8) AuditConnection
- 9) RestartInProgress

### 6.2 EndpointConfiguration

该命令由MGC发送给媒体网关，用于规定端点所接收信号的编码方式。

EndpointConfiguration (EndpointID,  
BearerInformation)

EndpointID是媒体网关中执行该命令的端点名称，本协议规定在该命令中EndpointID不可以使用通配符“any of”。当EndpointID使用通配符“all of”时，该命令应作用于所有满足通配条件的端点。BearerInformation（承载信息）用于定义端点接收数据的编码方式，该参数可包含一系列子参数，目前仅定义A/μ率编码方式。

### 6.3 NotificationRequest

该命令由MGC向MG发送，用于规定媒体网关所要监视/报告的在端点上发生的事件。

NotificationRequest( EndpointId,  
[NotifiedEntity,]  
[RequestedEvents,]  
RequestIdentifier,  
[DigitMap,]  
[SignalRequests,]  
[QuarantineHandling,]  
[DetectEvents,]  
[encapsulated EndpointConfiguration])

本协议规定EndpointID不能使用通配符“any of”。

### 6.4 Notify

该命令用于 MG 通知 MGC 报告观察事件的发生，端点在检测到指定观察事件时向 MGC 发送此命令。

Notify( EndpointId,  
[NotifiedEntity,]  
RequestIdentifier,  
ObservedEvents)

### 6.5 CreateConnection

该命令由 MGC 发送给 MG，用于在两个端点间创建一个连接。

CreateConnection(CallId,  
EndpointId,  
[NotifiedEntity,]  
[LocalConnectionOptions,]  
Mode,  
[{RemoteConnectionDescriptor |  
SecondEndpointId},]  
[Encapsulated NotificationRequest,]  
[Encapsulated EndpointConfiguration])

### 6.6 ModifyConnection

该命令用于修改连接特征，包括修改连接的本地特征和远端特征。

ModifyConnection(CallId,  
EndpointId,  
ConnectionId,

```
[NotifiedEntity, ]
[LocalConnectionOptions, ]
[Mode, ]
[RemoteConnectionDescriptor, ]
[Encapsulated NotificationRequest, ]
[Encapsulated EndpointConfiguration])
```

## 6.7 DeleteConnection

该命令可由 MGC 向 MG 发送，或由 MG 向 MGC 发送。该命令的使用方式如下分别所示。

### 1) 发自 MGC

MGC 通过发出该命令来终结连接，同时还可以通过它收集连接执行结果的数据。通常情况下，一个连接有两端，MGC 应该向涉及连接的两个媒体网关都发出 DeleteConnection 命令，若连接使用 IP 组播，则可以一个一个地独立删除。当端点上的所有连接均被删除以后，此端点应处于 inactive 模式。

```
DeleteConnection (CallId,
                  EndpointId,
                  ConnectionId,
                  [Encapsulated NotificationRequest, ]
                  [Encapsulated EndpointConfiguration])
```

### 2) 发自 MG

MG 通过发出该命令来删除连接。如果 MG 失去与连接相关的资源或者发现端点不能或者不愿意发送或接收语音时，可以向 MGC 发出 DeleteConnection 命令来终结连接。

```
DeleteConnection( CallId,
                  EndpointId,
                  ConnectionId,
                  Reason-code,
                  Connection-parameters)
```

### 3) 删除多个连接

该命令由 MGC 向 MG 发送，通过对 DeleteConnection 功能进行改动，可以同时删除多个连接。

```
DeleteConnection (CallId,
                  EndpointId)
```

## 6.8 AuditEndpoint

该命令由 MGC 向 MG 发送，用于查看 MG 中端点的状态。

```
AuditEndpoint (EndpointID,
               [RequestedInfo])
```

## 6.9 AuditConnection

该命令由 MGC 向 MG 发送，用于查看 MG 中与连接相关的参数。

```
AuditConnection (EndpointId,
                 ConnectionId,
                 RequestedInfo)
```

## 6.10 RestartInProgress

该命令由 MG 向 MGC 发送，用于向 MGC 报告 MG 中的端点退出服务或进入服务。

RestartInProgress (EndpointID,  
RestartMethod,  
[RestartDelay,]  
[Reason-code])

本协议定义了重启动方式 (RestartMethod)参数包括以下几种方式:

- 1) Graceful: 规定时延过后将指定端点置于退出服务状态。对于已建立的连接没有影响, 但MGC不能建立新的连接, 并且删除现有连接。
- 2) Forced: 将指定的端点立即强制置于退出服务状态, 已建立的连接也将被丢失。
- 3) Restart: 端点上的服务将在“RestartDelay”参数规定的时延后恢复。
- 4) Disconnected: 端点已经断开连接且正试图在建立新的连接, “RestartDelay”参数规定了端点断开连接的时间, 单位为秒, 且已建立的连接将不受影响。
- 5) Cancel-graceful: 指示MG取消先前的Graceful重启动方式, 端点仍处于服务状态。

“RestartDelay”参数为任选参数, 单位为秒。如果该参数在RestartInProgress中省略则认为空。当采用“Graceful”方式时, “RestartDelay”参数为空指示MGC不仅不能建立新的连接且必须等待现有连接的自然结束。当采用“Forced”和“cancel-graceful”方式中, “RestartDelay”参数总为空。当MG发送的采用“Restart”或者“graceful”的RestartInProgress命令中包含的“RestartDelay”参数为非零值, 则重启动时延结束后MG必须发送更新的RestartInProgress命令。

当采用“Restart”方式的RestartInProgress中所包含的“RestartDelay”参数为非零值, 则指示该断代理的服务已被恢复, 如MG加电启动或重启动。为了避免重启动造成网关IP地址的改变所带来的影响, MGC即可以采用将它本身的域名来代替MG的域名的方式, 也可以通过查询DNS服务器来对MG的域名进行解析。

“Reason-code”参数为可选参数, 用于指示MG发生重启动的原因。

如果重启动成功, 则返回200 OK成功响应, 端点返回的被通知实体就是新的被通知实体; 如果重启动未成功, 则MGC返回一个错误响应。

## 6.11 响应代码和错误代码

本协议规定命令采用请求—证实的方式来进行交互, 因此, 命令的证实响应必须携带响应代码, 用于指示命令的执行状态。响应代码为整数, 目前本协议规定已定义5个范围的响应代码值。

- 1) 000~099: 响应证实;
- 2) 100~199: 临时响应;
- 3) 200~299: 成功响应;
- 4) 400~499: 临时性错误响应;
- 5) 500~599: 永久性错误响应;
- 6) 800~899: 包(Package)指定的响应代码。

本协议已定义的响应代码如表1所示。

表 1 错误代码列表

错误代码	含义
000	响应证实
100	事务正在被处理, 命令成功响应将稍后发送
101	事务正排队处理, 命令成功响应将稍后发送
200	事务被正常处理, 用于指示成功响应
250	用于对DeleteConnection命令的成功响应
400	不确定的临时性错误
401	电话已摘机
402	电话已挂机

403	端点没有足够的资源
404	带宽不足
405	端点正处于重启状态
406	事务处理超时，事务在指定时间内无法正常完成
407	由于外部命令的到来导致事务处理失败，如MDCX命令被DLCX命令所终结
409	MG内部超载
500	端点未知
501	端点处于未准备状态以及“Out of Service”状态
502	端点没有足够资源
503	通配符“All of ”过于复杂
504	未知或不支持的命令
508	未知或不支持“quarantinehandling”参数
509	RemoteConnectionDescriptor错误
510	不能确定的协议错误（该错误代码建议为最终的错误响应）
511	命令包含不能识别的扩展（如无法识别的以“x+”为首位的关键扩展参数）
512	MG不能检测要求检测的事件
513	MG不能检测要求发送的事件
514	MG不能播放指定的通知
515	事务交互指向不正确的ConnectionID（该ConnectionID可能被删除）
516	事务交互指向不正确的CallID
517	不支持或非法的模式 (Mode)
518	不支持或未知的包（建议应答消息应包括PackageList参数）
519	端点不包含Digitmap
520	端点正在重启动
521	端点正重定向到另外一个MGC
522	不支持的事件或信号
523	未知的action和非法的action组
524	LocalConnectionOptions参数内不一致
525	LocalConnectionOptions参数中未知的扩展（如无法识别的以“x+”为首位的关键扩展参数）
526	带宽不足（若为临时性错误，应以404代替）
527	RemoteConnection描述符丢失
528	协议版本不匹配
529	内部硬件错误
530	CAS信令协议错误
531	中继群错误
532	LocalConnectionOptions命令参数中不支持的值
533	响应消息过长
534	编码协商失败
535	不支持的“Packetization period”参数值
536	未知或不支持的重启动模式
537	未知或不支持的DigitMap扩展
538	Event/Signal参数错误
539	非法或不支持的命令参数
540	超过端点允许的最大连接数目

本标准规定，如果实体接收到未知或者不支持的响应代码应遵循以下原则进行处理。

- 1) 未知的0XX代码应被视做000
- 2) 未知的1XX代码应被视做100
- 3) 未知的2XX代码应被视做200
- 4) 未知的3XX代码应被视做521
- 5) 未知的4XX代码应被视做400
- 6) 未知的5XX-9XX代码应被视做510

## 7 协议消息格式

### 7.1 概述

本标准规定MGCP协议通过事务交互(Transaction)来实现MGC对MG之间的控制管理。一个事务交互由命令请求和命令响应组成。MGCP协议共定义了9个命令请求消息，参见本标准6.2-6.10节。

所有命令请求消息由命令头和会话描述两部分组成，命令响应消息由响应头和会话描述两部分组成，其中会话描述部分为可选项，采用SDP协议进行描述，具体规则参见RFC 2327。消息中命令头（或响应头）和会话描述都由若干文本行组成，并通过空行进行分隔。

本标准规定MGCP协议采用TransactionID来关联命令请求和命令响应，其中，TransactionID编码作为命令请求中命令头的一部分，并在相应的命令响应中重复出现。

本标准规定MGCP协议消息采用ABNF编码方式进行编码（参见本标准附录A），ABNF的编码规则参见RFC2234。

### 7.2 命令头

本标准规定MGCP协议命令请求消息的命令头由一个命令行和若干个参数行两部分组成。

#### 7.2.1 命令行 (command line)

本标准规定命令行由命令请求名称、TransactionID、EndpointID和协议版本号四个部分组成，其中，每个部分都是ASCII字符串，且相互间采用空格分隔，其中，空格可采用ASCII空格字符（0x20）和TAB字符(0x09)来进行标识。命令行结构示意图如下所示，[ ]表示空格字符。

命令请求名称 [ ] TransactionID [ ] EndpointID [ ] 协议版本号

##### 7.2.1.1 命令请求名称

本标准规定命令请求名称为4个字母的ASCII编码，且不区分大小写。MGCP协议所定义的命令请求名称ASCII编码如表2所示。

表2 命令请求名称 ASCII 编码

命令请求名称	ASCII编码
EndpointConfiguration	EPCF
CreatConnection	CRXC
ModifyConnection	MDCX
DeleteConnection	DLCX
NotificationRequest	RQNT
Notify	NTFY
AuditEndpoint	AUEP
AuditConnection	AUCX
RestartInProgress	RSIP

### 7.2.1.2 TransactionID

本标准规定 MGCP 协议采用 TransactionID 来关联命令请求和响应响应。本标准建议 TransactionID 由一个九位数来进行标识,TransactionID 的取值范围为 1~999999999,且首位不能为 0。TransactionID 在命令行中位于命令请求名称之后,且用一个空格字符进行分隔。本标准规定,在一个 Transaction 完成后三分钟之内,消息实体不允许选用该 Transaction 所采用的 TransactionID。

### 7.2.1.3 EndpointID (端点标识符)

EndpointID 用于标识协议消息的被通知实体。本标准建议 EndpointID 采用 Email 地址的形式进行标识,Email 地址结构的定义参见 RFC821。EndpointID 的参考结构为“用户名@域名:端口号”,其中,域名标识端点所属的系统,用户名则标识该系统的指定端点或者实体,端口号为可选项,用于标识协议消息所使用的端口号,当端口号省略时应采用 MGCP 协议默认的消息被通知实体的端口号。

### 7.2.1.4 协议版本号

本标准规定协议版本号由一个主版本号和一个副版本号组成,且中间由“.”进行分隔,主版本和副版本号都是由数字进行标识。本标准所定义的协议版本号为 1.0。当一个消息实体接收到不支持的协议版本命令消息时,必须返回一个响应代码为 528 (协议版本不兼容)的响应消息。

## 7.2.2 参数行 (Parameter line)

本标准规定参数行结构如下所示:

参数名: [空格] 参数值

通常,参数名由一个和两个大写字母或数字组成,且首位必须为大写字母。本标准定义的命令参数如表 3 所示。

表 3 命令参数列表

命令参数名称	编 码	参数值
BearerInformation	B	参见7.2.2.1
CallId	C	参见7.2.2.2
Capabilities	A	参见7.2.2.3
ConnectionId	I	参见7.2.2.4
ConnectionMode	M	参见7.2.2.5
ConnectionParameters	P	参见7.2.2.6
DetectEvents	T	参见7.2.2.7
DigitMap	D	数字映射编码
EventStates	E S	参见7.2.2.8
LocalConnectionOptions	L	参见7.2.2.9
MaxMGCPDatagram	M D	参见7.2.2.10
NotifiedEntity	N	Email地址标识,Email地址编 码规则参见RFC821
ObservedEvents	O	参见7.2.2.11
PackageList	P L	参见7.2.2.12
QuarantineHandling	Q	参见7.2.2.13
ReasonCode	E	参见7.2.2.14
RequestedEvents	R	参见7.2.2.15
RequestedInfo	F	参见7.2.2.16

RequestIdentifier	X	参见7.2.2.17
ResponseAck	K	参见7.2.2.18
RestartDelay	R D	十进制整数，单位秒
RestartMethod	R M	参见7.2.2.19
SecondConnectionId	I 2	ConnectionID
SecondEndpointId	Z 2	EndpointID
SignalRequests	S	参见7.2.2.20
SpecificEndPointId	Z	Email地址形式
RemoteConnection-Descriptor or	R C	会话能力描述（SDP）
LocalConnection-Descriptor	L C	会话能力描述（SDP）

表4为命令参数和命令关联表，其中，M表示该参数为必选项，Q表示该参数为任选项，F表示该参数为禁止项。若无特别说明，一个命令参数只能出现一次。

表4 命令参数和命令关联表

参数名称	E PCF	C RCX	M DCX	D LCX	R QNT	N TFY	A UEP	A UCX	R SIP
BearerInformation	0	0	0	0	0	F	F	F	F
CallId	F	M	M	0	F	F	F	F	F
Capabilities	F	F	F	F	F	F	F	F	F
ConnectionId	F	F	M	0	F	F	F	M	F
ConnectionMode	F	M	0	F	F	F	F	F	F
ConnectionParameters	F	F	F	0	F	F	F	F	F
DetectEvents	F	0	0	0	0	F	F	F	F
DigitMap	F	0	0	0	0	F	F	F	F
EventStates	F	F	F	F	F	F	F	F	F
MaxMGCPDatagram	F	F	F	F	F	F	F	F	F
NotifiedEntity	F	0	0	0	0	0	F	F	F
ObservedEvents	F	F	F	F	F	M	F	F	F
PackageList	F	F	F	F	F	F	F	F	F
QuarantineHandling	F	0	0	0	0	F	F	F	F

ReasonCode	F	F	F	0	F	F	F	F	0
RequestedEvents	F	0	0	0	0	F	F	F	F
RequestIdentifier	F	0	0	0	M	M	F	F	F
RequestedInfo	F	F	F	F	F	F	0	M	F
ResponseAck	0	0	0	0	0	0	0	0	0
RestartDelay	F	F	F	F	F	F	F	F	0
RestartMethod	F	F	F	F	F	F	F	F	M
SecondConnectionId	F	F	F	F	F	F	F	F	F
SecondEndpointId	F	0	F	F	F	F	F	F	F
SignalRequests	F	0	0	0	0	F	F	F	F
SpecificEndpointId	F	F	F	F	F	F	F	F	F
LocalConnectionOptions	F	F	F	F	F	F	F	F	F
RemoteConnectionOptions	F	0	0	F	F	F	F	0	F

注：

- 1) BearerInformation 参数只有在 EPCF 命令中为任选项。
- 2) RequestedID 参数在 MDCX、CRCX、DLCX 命令中为任选项，若命令消息中包含被封装的通知请求，此参数应为必选项。
- 3) RequestedEvents 和 SignalRequests 参数在 RQNT 命令中为任选项，若命令中该参数省略，则响应列表就被认为是空。
- 4) ConnectionParameters 参数只有在 MG 发送的 DLCX 命令中才有效。

本标准规定命令参数可通过包扩展参数和厂家扩展参数进行扩展，且建议优选包扩展方式。包扩展参数在包的定义中进行定义。本标准规定扩展参数名必须由“X-”或“X+”为首位的字符串组成，且“X+”为首位的参数名为关键参数扩展名，“X-”开头的参数名是非关键参数扩展名。当消息实体接收到无法识别的关键参数扩展名时，应将拒绝执行命令，并返回错误代码为 511 的响应消息。消息实体接收到无法识别的非关键参数扩展名时，则对该参数不进行任何处理。

#### 7.2.2.1 BearerInformation (承载信息)

BearerInformation 属性用于识别语音流的编码方法，目前，本标准暂时规定可采用 A/ $\mu$ 率的编码方式。BearerInformation 属性可通过包进行扩展。

#### 7.2.2.2 CallID (呼叫标识)

本标准规定 CallID 为一个十六进制的字符串，最大字符长度为 32 位。

#### 7.2.2.3 Capabilities (能力)

当MGC发出审计请求时，Capabilities参数用于向MGC报告端点的会话能力。Capabilities参数中属性的编码方式应与LocalConnectionOptions参数中属性的编码方式相同，而Capabilities参数用“A”进行标识。除此之外，Capabilities参数还可以包含支持包类型的列表和支持模型类型的列表。

Capabilities参数包含以下属性参数：

- 1) 支持编码方式类型的列表
- 2) 打包周期 (Packetization Period)

该属性参数用于指定一个打包周期的取值范围。

- 3) 带宽 (Bandwidth)

该属性参数用于指定与打包周期取值范围相对应的一个取值范围。

#### 4) 回声抑制 (Echo Cancellation)

该属性参数为“ON”表示端点支持回声抑制，“OFF”表示端点不支持回声抑制。该参数缺省值为支持回声抑制。

#### 5) 静音压缩 (Silence Suppression)

该属性参数为“ON”表示端点支持静音压缩，“OFF”表示端点不支持静音压缩。该参数缺省值为支持静音压缩。

#### 6) 增益控制 (Gain Control)

该属性参数为“0”表示端点不支持增益控制，除0之外的任意值表示端点支持增益控制。该参数省略值为支持增益控制。

#### 7) 业务分级 (Type of Service)

该属性参数为“0”表示端点不支持业务等级，除0之外的任意值表示端点支持业务等级。该参数缺省值为支持业务等级。

#### 8) 资源预留 (Resource Reservation Service)

该属性参数为“g”表示网关支持已确保和受控的负载业务，“cl”表示支持受控的负载业务。

#### 9) 密钥 (Encryption Key)

该属性参数为任意值表示支持加密传输，该属性参数缺省值为不支持加密传输。

#### 10) 网络类型 (Type of network)

该属性参数为可选项，用“nt”进行标识，用于指定所支持的网络类型。

#### 11) 包 (Package)

该属性参数用于指定端点所支持的包类型，该属性参数用“v”进行标识。如果该参数指定了一个包类型列表，则该列表的第一个包类型为端点所支持的缺省包类型。

#### 12) 模式 (Modes)

该属性参数用“m”标识，用于指定端点支持的连接模式。

以下为 Capabilities 参数的示例：

A: a:PCMU;G728, p:10-100, e:on, s:off, t:l, v:L, m:sendonly; recvonly;sendrecv;inactive

本标准规定 Capabilities 参数中删除了一个属性参数，则表示端点不支持该属性参数所对应的功能，且如果需要返回多个 Capabilities 参数，则每一个 Capabilities 参数都应分行罗列。

由于 LocalConnectionOptions 可以进行扩展，因此 Capabilities 参数也可以进行扩展。单个扩展可以定义能力的定义方法。如果扩展未指定扩展能力定义方法，则遵循以下规则：

- 1) 包扩展属性：不用于上报单个属性，而仅仅在 Package 属性参数中列出扩展的包类型。
- 2) 厂家扩展属性：上报不携带属性值的扩展属性的属性名。
- 3) 其他扩展属性：上报不携带属性值的扩展属性的属性名。

### 7.2.2.4 事件编码 (Event)

本标准规定事件(Event)的编码方式由包名和事件名组成，包名和事件名之间用“/”字符进行分隔，如“包名/事件名”。其中，包名为可选项，且包名可以使用通配符“\*”用于表示全部包类型。除此之外，事件编码还可使用 ConnectionID，用于指定该事件所关联的连接。ConnectionID 和事件名之间用“@”字符进行分隔，如“包名/事件名@ConnectionID”。事件编码可用 RequestedEvents, SignalRequests, ObservedEvents, DetectEvents, 和 EventStates 命令参数。此外，RequestedEvents 和 DetectEvents 命令参数中的事件编码可使用取值范围或通配符。通配符“all”为预留给值，用于表示包中的全部事件或信号。当事件编码中的 ConnectionID 为通配符“\*”，则表示该事件用于所有连接，“\$”表示用于当前连接。表 5 为事件编码的示例。

表 5 事件编码示例

事件编码	含义
------	----

M/[0-9]	MF包中的0~9号码
hf	模拟线(al)包中的Hook-flash事件
[0-9*#A-D]	DTMF包中的所有数字和字符
T/all	中继(Trunk)包中的所有事件
R/qa@*	所有连接的quality alert事件
G/rt@\$	当前连接的回铃音

#### 7.2.2.5 ConnectionId (连接标识)

该命令参数用一个十六进制字符串进行标识，字符串最大长度为 32 位。

#### 7.2.2.6 ConnectionMode (连接模式)

该命令参数用于定义连接的操作模式，本标准定义了 9 种连接模式，分别为 sendonly、recvonly、sendrecv、confrnce、inactive、loopback、conttest、netwloop 和 netwttest，各连接模式的含义如表 6 所示。

表 6 连接模型列表

连接模式	含义
sendonly	网关只能发送媒体流
recvonly	网关只能接收媒体流
sendrecv	网关可发送和接收媒体流
confrnce	网关处于会议连接模式
inactive	网关既不能发送和接收媒体流
loopback	网关置中继电路于环回状态
conttest	网关置中继电路于测试状态
netwloop	网关置网络连接于网络环回状态
netwttest	网关置网络连接于连续测试状态

#### 7.2.2.7 ConnectionParameters (连接参数)

ConnectionParameters 参数编码由参数类型和参数值两部分组成，参数类型还可以进行扩展。其中，参数类型用两个大写字母组成的字符串标识，参数值用十进制整数进行标识。本标准定义的 ConnectionParameters 参数如表 7 所示。

表 7 ConnectionParameters 列表

连接参数名称	编码	参数值
Packets sent	PS	连接上已经发送的包数目
Octets sent	OS	连接上已经发送的字节数目
Packets received	PR	连接上已经接收的包数目
Octets received	OR	连接上已经接收的字节数目
Packets lost	PL	连接上丢失的包数目
Jitter	JI	平均的包抖动时延 (单位毫秒)
Latency	LA	平均时延 (单位毫秒)

ConnectionParameters 参数可采用包和设备提供方扩展两种方式进行扩展。当MGC接收到无法识别的包或设备提供方扩展参数时，则不应对该参数进行任何处理。

#### 7.2.2.8 DetectEvents (检测事件)

DetectEvents 参数由事件列表组成，其中，事件的编码方式参见 7.2.2.2 节。

例如：T: L/hu, L/hd, L/hf, D/[0-9#\*]

#### 7.2.2.9 EventStates (事件状态)

EventStates参数由事件列表组成，其中，事件的编码方式参见7.2.2.2节。

例如：ES: L/hu

#### 7.2.2.10 LocalConnectionOptions (本地连接)

LocalConnectionOptions参数用于在MGC向MG发送的连接处理命令中（如CRCX和MDCX）定义连接的可操作性参数，如下所示：

- 1) 编码方式：该参数用“a”标识。
- 2) 打包周期：该参数用“p”标识，单位毫秒。
- 3) 带宽：该参数用“b”标识。
- 4) 业务等级：该参数用“t”标识。
- 5) 回声抑制：该参数用“e”标识，参数值为“on”或“off”。
- 6) 增益控制：该参数用“gc”标识，参数值为“auto”或一个十进制数的增益分贝。
- 7) 静音压缩：该参数用“s”标识，参数值为“on”或“off”。
- 8) 资源预留：该参数用“r”标识，参数名为“g”、“cl”或“be”。
- 9) 密钥：该参数用“k”标识，此参数定义同SDP中定义的参数“K”。
- 10) 网络类型：该参数用“nt”标识，参数值为“IN”、“ATM”或“LOCAL”。

LocalConnectionOptions 参数的示例如下：

L: p:10, a:PCMU

L: p:10, a:G726-32

L: p:10-20, b:64

L: b:32-64, e:off

设备提供方扩展属性由属性名和属性值组成，属性名和属性值之间用“,”字符进行分隔，且属性值为任选项。本标准规定强制的扩展属性应以字符“X+”为首位进行标识，非强制扩展属性应以“X-”为首位进行标识。当MG接收到一个无法识别的强制扩展属性，MG应拒绝执行该命令并返回错误码代码为525的响应消息。

#### 7.2.2.11 MaxMGCPDatagram (最大协议包)

MaxMGCPDatagram参数只能用于审计命令的响应消息中，该参数用一个最大长度为9位的十进制数进行标识，且首位不能为0。

例如：MD: 8100。

#### 7.2.2.12 ObservedEvents (检测事件)

ObservedEvents参数用于定义检测事件的列表。其中，根据DigitMap收集的号码事件可以由一个字符串进行标识。当在手机号码事件的过程中，检测到其他事件发生，则这些号码事件应被当作孤立事件逐个报告。例如：

0: L/hu

0: D/8295555T

0: D/8, D/2, D/9, D/5, D/5, L/hf, D/5, D/5, D/T

0: L/hf, L/hf, L/hu

#### 7.2.2.13 PackageList (包列表)

PackageList参数只能用于对审计命令的响应消息，响应消息中包含PackageList用于指示端点所

支持的包类型列表。响应消息返回的 PackageList 参数第一个包类型为端点所支持的缺省包。

例如：PL: L:1;G:1;D:0;FOO:2;T:1

#### 7.2.2.14 QuarantineHandling

QuarantineHandling 参数包含了一个关键词列表。关键词“process”和“discard”用于指示对隔离和 ObservedEvents 的处理。如果“process”和“discard”都不具备，假设为“process”。关键词“step”或者“loop”用于指示每一个允许的通知请求命令的最多一个通知，或者每一个允许的通知请求命令的多个通知。如果“step”或者“loop”都不具备，假设为“step”。

#### 7.2.2.15 ReasonCode (原因值)

ReasonCode 参数用一个三位数值进行标识，ReasonCode 可通过包进行扩展。

#### 7.2.2.16 RequestedEvents (请求事件)

RequestedEvents 参数用于定义被请求事件的列表。当请求的事件发生时，本标准规定事件操作方式及编码方式如表 8 所示。

表 8

操作方法	编码
立即上报	N
Accumulate	A
根据DigitMap进行处理	D
Swap	S
Ignore	I
保持信号音	K
内嵌通知请求	E

在内嵌通知请求命令中，“内嵌通知请求”参数是由一个包含三个参数组的列表组成。其中，第一个参数为任选参数组，由字母“R”标识，是内嵌 RequestedEvents 的参数值；第二个参数组为任选参数组，由字母“S”标识，是内嵌 RequestedInfo 的参数值；第三个参数组为任选参数组，由字母“D”标识，是 DigitMap 的参数值。

如果请求命令中不包含 RequestedEvents 参数，则该值应被设为空。如果请求命令中不包含 SignalRequests 参数，则该值应被设为空。如果请求命令中不包含 DigitMap 参数，则应使用当前值。当事件携带附加的事件参数要求时，检测事件发生过程中，只有当事件附加参数也满足要求时，该事件才发生。

#### 7.2.2.17 RequestedInfo (请求信息)

RequestedInfo 参数用于审计命令中用于定义请求的参数，该参数由一个参数代码列表组成，其中各参数由“,” 字符进行分隔。扩展参数也可以被审计。

例如：F: N, X, R, S, D, Q, T。

#### 7.2.2.18 RequestedID (请求标识)

RequestedID 参数用于关联 Notify 命令和触发该命令的 RQNT 命令。RequestedID 由十六进制字符串组成，最大长度为 32 位。RequestedID 等于“0”被保留用于报告持续事件。

#### 7.2.2.19 ResponseAck (请求应答)

ResponseAck 参数被用于管理“at-most-once”设备，该参数由一个“已证实事务 ID 范围”列表组成。每个“已证实事务 ID 范围”由一个十进制数或者两个十进制数组成，这两个代表范围中的高层

和底层事务 ID。一个 ResponseAck 的例子： K: 6234-6255, 6257, 19030-19044

### 7.2.2.20 RestartMethod (重启动模式)

RestartMethod命令参数被编码成关键词“graceful”，“forced”，“restart”，“disconnected”或者“cancel-graceful”之一。例如： RM: restart。RestartMethod可以由封包扩展。

### 7.2.2.21 SignalRequests

SignalRequests参数提供了RequestedInfo的名称。

## 7.3 响应头

本标准规定响应头由一个响应行和若干个参数行组成，其中响应参数为可选项。响应行的格式如下所示：

响应码[空格]TransactionID[空格]注释

例如： 200 1203 OK

响应行以响应代码开头，响应代码由一个十进制的三位整数标识，响应代码和 TransactionID 用空格字符进行标识。当响应码为包中定义的响应码时，则用“包/响应码”进行标识。当响应头中包含参数行时，响应头中允许出现的命令参数如表 9 所示。其中，F 表示该参数为禁止项，M 表示该参数为必选项，O 表示该参数为可选项。

表 9 响应头中命令参数列表

参数名称	E PCF	C RCX	M DCX	D LCX	R QNT	N TFY	A UEP	A UCX	R SIP
BearerInformation	F	F	F	F	F	F	O	F	F
CallId	F	F	F	O	F	F	F	O	F
Capabilities	F	F	F	F	F	F	O	F	F
ConnectionId	F	O	F	F	F	F	O	F	F
ConnectionMode	F	F	F	F	F	F	F	O	F
ConnectionParameters	F	F	F	O	F	F	F	O	F
DetectEvents	F	F	F	F	F	F	O	F	F
DigitMap	F	F	F	F	F	F	O	F	F
EventStates	F	F	F	F	F	F	O	F	F
LocalConnectionOptions	F	O	O	F	F	F	F	O	F
MaxMGCPDatagram	F	F	F	F	F	F	O	F	F
NotifiedEntity	F	F	F	F	F	F	O	O	O
ObservedEvents	F	F	F	F	F	F	O	F	F
PackageList	F	F	F	F	F	F	O	F	F
QuarantineHandling	F	O	O	O	O	F	F	F	F

ReasonCode	F	F	F	F	F	F	0	F	F
RequestedEvents	F	F	F	F	F	F	0	F	F
RequestIdentifier	F	F	F	F	F	F	0	F	F
RequestedInfo	F	F	F	F	F	F	F	F	F
ResponseAck	0	0	0	0	0	0	0	0	0
RestartDelay	F	F	F	F	F	F	F	F	0
RestartMethod	F	F	F	F	F	F	F	F	M
SecondConnectionId	F	0	F	F	F	F	F	F	F
SecondEndpointId	F	0	F	F	F	F	F	F	F
SignalRequests	F	F	F	F	F	F	0	F	F
SpecificEndpointId	F	0	F	F	F	F	0	F	F
RemoteConnectionOptions	F	F	F	F	F	F	F	0	F

## 8 协议传送要求

为了加快协议消息的传送要求，本标准建议 MGCP 协议消息采用 UDP 进行传送。命令请求消息将送往指定端点在 DNS 中登记的一个 IP 地址，命令响应消息则应回送到命令的发起方。如果端点未指定的端口号，则由 MGC 发往 MG 的缺省 MGCP 端口为 2427，由 MG 发往 MGC 的缺省 MGCP 端口为 2727。由于 UDP 本身不具备提供可靠服务的功能，因此，本标准规定 MGCP 协议采用以下规则来确保协议消息的可靠传送和正确处理。

### 8.1 “At-Most-Once”机制

由于协议消息采用 UDP 进行传送，则 MGCP 消息可能发生丢失。因此，如果消息发送方未及时收到响应消息，则会重发此命令。然而，重复执行命令可能会造成大多数命令执行发生混乱，例如，多吃执行 CRCX 命令会使 MG 状态成为不可预测。为了避免此类情况发生，本标准规定协议消息传送过程中必须确保同一命令只能出现一次，也就是“**At-Most-Once**”机制。

本协议规定消息实体应在其存储器中保存它们对最近事务回送的响应以及目前正在执行的事务的 TransactionID。当消息实体接收到一条命令之后应将此命令的 TransactionID 与最近响应的 TransactionID 进行比较，如果最新收到的事务的 TransactionID 与存储器中某一响应的 TransactionID 相同，则不应执行最新收到的事务，而是重发此响应。否则，应继续和当前执行事务进行比较，如果最新收到的事务的 TransactionID 与当前正在执行的最新收到的事务的 TransactionID 相同，则丢弃此命令，既不执行此事务也不发送事务响应，等待当前事务执行完毕后再发送响应消息。

当命令响应发出 30 秒之后，或者消息发送方通过“**ResponseAck**”参数告知消息接收方此命令响应已经接收到时，消息实体可以将存储的命令响应进行删除。

### 8.2 TransactionID 和三次握手

本标准规定 TransactionID 的取值范围为 0~999999999 的十进制整数。MGC 可以为它所管辖的每个 MG 设置一个独立的取值范围，也可以为若干个 MG 设置一个取值范围。在具体实现时，MGC 可以有多个独立的进程按照负荷分担的原则管理一个大型的 MG，因此，本标准建议这些进程应共享一个 TransactionID 的取值空间，所有事务都应由一个唯一的 TransactionID 进行标识。

MGCP 协议消息中所有的命令请求可以携带“**ResponseAck**”命令参数，该参数用于标识已收到的响应的 TransactionID 取值范围，通过“**ResponseAck**”命令参数，协议消息发送方和接收方可以实现三次握手的过程。当 MG 发出最后一次响应 30 秒之后，或当 MG 继续执行该事务操作时，MG 所发送消息中所包含的“**ResponseAck**”命令参数则不应再包含此事务的 TransactionID。

### 8.3 重传定时机制

本标准规定命令请求消息发送方在发出命令请求之后应设置定时器，当定时器超时消息发送方未接收到相应的命令响应，应重新发送此命令请求。当消息多次重发之后仍未接收到响应，则应由消息发送方决定清除已建立的连接或者通过冗余设备提供服务。本标准不对定时器的具体数值进行规定，重传定时器的取值与网络状况有关，本标准建议根据消息往返时延来进行估算。

### 8.4 最大 MGCP 包长度

理论上，IP 包最大长度为 65535 比特，其中包含 20 比特的 IP 包头和 8 比特的 UDP 包头，因此，当 MGCP 协议使用 UDP 进行传送时，最大 MGCP 包长度的理论值为 65507 比特。然而，IP 层规定主机所能接收的最大包长度为 576 比特。因此，本标准规定，MGCP 协议消息实体所能发送的最大 MGCP 包长度至少为 4000 比特。当 IP 层采用 Ipsec 机制时，最大 MGCP 包长度的值至少应大于 4000 比特。

如果未特别说明，MGC 和端点所能支持的最大 MGCP 包长度的值至少应大于 4000 比特。MGC 可以向端点发送审计命令用于审计端点所能支持的最大 MGCP 包长度，但是，端点不能审计 MGC 所能支持的最大 MGCP 包长度。

### 8.5 消息携带传送

本标准规定 MGC 可以在一个 UDP 包中向 MG 同时传送多个命令请求或响应消息，即消息携带发送机制。此时，每个消息之间用一个且仅有一个“.”字符文本行进行分隔。当 MG 接收到此类消息时，应将携带消息视为多个同时到达的消息。

### 8.6 临时响应

某些事务的执行时间较长，执行事件过长可能会导致重传定时器超时，从而导致消息传送发生混乱。本协议建议当 MG 和 MGC 预测到执行事务的时间过长时必须先发送一个响应代码为 100 的临时响应消息。本标准规定消息接收方需要与外部交换信息才能完成事务执行时应发送一个临时响应，此外，当消息接收方接收到未完成的重复事务请求时也应发出一个临时响应。而 MG 或 MGC 需要处理多个事务时则应发送一个响应代码为 101 的临时响应，指示该事务请求正处于被排队处理状态。

为了减少呼叫建立时延，消息接收方接收到 CRCX 命令请求时，它所发送临时响应中可以包括一个 ConnectionId 和会话描述。如果 MDCX 命令请求的响应中返回了会话描述部分，该会话描述也应包含在临时响应中。当事务完成后，临时响应中返回的信息必须在最终响应中被重复。在成功完成的响应中，如果临时响应信息未重复或者被改变，则协议发生错误。

当端点接收到 DLCX 命令，当前正在执行的 CRCX 命令或 MDCX 命令都将被取消执行，并返回错误代码 407 为命令响应。如果消息实体检测到事务取消，则应返回最终的事务取消响应。

当事务执行完成后，消息实体应发送最终响应并删除临时响应。为了确保迅速检测到最终响应的丢失，本标准规定最终响应必须被确认，即端点必须在最终响应中包括一个空的“ResponseAck”参数。最终响应中的“ResponseAck”参数将触发一个“Response Acknowledgement”响应，该响应将发回至端点。Response Acknowledgement 响应将包括响应头中的 TransactionID。

## 9 协议安全要求

为了防止未经授权的实体利用本协议建立非法呼叫，或者干涉合法呼叫，需要对本协议的传输建立安全机制。当在 IP 网络上传输本协议时，本协议建议使用 IPsec [RFC2401]到[RFC2411]对协议传输进行安全保护。

认证头 (AH) 协议对在 MG 和 MGC 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发送保护，AH 协议的内容参见 RFC2402。封装安全净荷 (ESP) 协议则对 MG 和 MGC 之间传送的消息提

供加密，ESP 协议内容参见 RFC2406。例如，当 SDP 所携带的会话描述用语协议会话密钥时，应提供 ESP 加密业务。本协议规定，在本协议中实现 AH 协议应遵循 RFC2402 第 5 节，实现 ESP 协议应遵循 RFC2406 第 5 节。

本标准建议使用密钥交换（IKE）协议提供 MG 和 MGC 之间进行密钥协商。在本协议中实现 IKE 协议应支持 RSA 签名和 RSA 公共密钥加密。

当 MGCP 协议采用 Ipsec 机制进行传送时，其协议栈如图 2 所示。



图 3 Ipsec 机制传送 MGCP 协议消息

### 9.1 保护媒体连接

本协议支持软交换设备或 MGC 向 MG 提供会话密钥，用来对音频消息进行加密以防止窃听。

分组网络的一个缺陷就是容易受到由非法实体发起的恶意攻击。例如，非法实体可以通过向一个指定的 IP 地址和 UDP 端口发送大量 IP 包进行攻击。针对这种攻击进行保护的一种方法是仅接受来自获得授权实体的分组，例如，仅接收 RemoteConnectionOptions 参数中列出的源 IP 地址和 UDP 端口号。然而这种方法可能会受到地址欺骗的影响以及导致连接建立的延迟。另外一种进行保护的方法是在呼叫建立过程中传送密钥对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以及可以有效地防止地址欺骗。

## 10 包（Package）

本标准定义了包用于实现对 MGCP 协议的扩展。包必须由一个唯一的包名进行定义，其中包名必须为在 IANA 进行注册的正式名称。其中，符号“X-”或“X+”为首位开头的扩展包之外，这种扩展包用于实验包。

本标准规定包必须有一个版本号进行标识，该版本号为一个非负整数表示。包的缺省版本号为 0，下一个版本是 1，依次类推，且新版本的包必须提供完全的后向兼容。例如，新版本的包中应禁止重新定义或删除早期版本的包中定义的扩展，否则，应重新定义一个新的包类型。

本标准规定包的定义包括以下几个部分：

- 1) Actions
- 2) BearerInformation
- 3) ConnectionModes
- 4) ConnectionParameters
- 5) DigitMap
- 6) Events and Signals
- 7) ExtensionParameters
- 8) LocalConnectionOptions
- 9) ReasonCodes
- 10) RestrartMethods
- 11) Return codes

当需要定义包中的各个部分的扩展类型时，必须应包含一个扩展描述。

附 录 A  
(规范性附录)  
本标准协议语法的 ABNF 描述

本附录规定了协议消息使用 ABNF 进行编码的描述方式，ABNF 的相关内容参见 RFC2234。

A.1 ABNF语法定义

```
MGCPMessage = MGCPCommand / MGCPResponse
MGCPCommand = MGCPCommandLine 0*(MGCPPParameter) [EOL *SDPinformation]
MGCPCommandLine = MGCPVerb 1*(WSP) <transaction-id> 1*(WSP)
                  <endpointName> 1*(WSP) MGCPversion EOL
MGCPVerb = "EPCF" / "CRCX" / "MDCX" / "DLCX" / "RQNT"
          / "NTFY" / "AUEP" / "AUCX" / "RSIP" / extensionVerb
extensionVerb = ALPHA 3(ALPHA / DIGIT) ; experimental starts with X
transaction-id = 1*9(DIGIT)
endpointName = localEndpointName "@" DomainName
LocalEndpointName = LocalNamePart 0*("/") LocalNamePart
LocalNamePart = AnyName / AllName / NameString
AnyName = "$"
AllName = "*"
NameString = 1*(range-of-allowed-characters) ; VCHAR except "$", "*", "/", "@"
range-of-allowed-characters = %x21-23 / %x25-29 / %x2B-2E
                          / %x30-3F / %x41-7E
DomainName = 1*256(ALPHA / DIGIT / "." / "-") ; as defined
            / "#" <number> / "[" <dotnum> "]" ; in RFC 821
            / "[" Ipv6address "]" ; see RFC 2373
MGCPversion = "MGCP" 1*(WSP) 1*(DIGIT) "." 1*(DIGIT)
             [1*(WSP) ProfileName]
ProfileName = VCHAR *( WSP / VCHAR) ; Was range-of-allowed-characters
MGCPPParameter = ParameterValue EOL
; Check infoCode if more parameter values defined
; Most optional values can only be omitted when auditing
ParameterValue = ("K" ":" 0*(WSP) [ResponseAck])
                / ("B" ":" 0*(WSP) [BearerInformation])
                / ("C" ":" 0*(WSP) CallId)
                / ("I" ":" 0*(WSP) [ConnectionId])
                / ("N" ":" 0*(WSP) [NotifiedEntity])
                / ("X" ":" 0*(WSP) [RequestIdentifier])
                / ("L" ":" 0*(WSP) [LocalConnectionOptions])
                / ("M" ":" 0*(WSP) ConnectionMode)
                / ("R" ":" 0*(WSP) [RequestedEvents])
                / ("S" ":" 0*(WSP) [SignalRequests])
                / ("D" ":" 0*(WSP) [DigitMap])
```

/ ("O" ":" 0\*(WSP) [ObservedEvents])  
 / ("P" ":" 0\*(WSP) [ConnectionParameters])  
 / ("E" ":" 0\*(WSP) ReasonCode)  
 / ("Z" ":" 0\*(WSP) [SpecificEndpointID])  
 / ("Z2" ":" 0\*(WSP) SecondEndpointID)  
 / ("I2" ":" 0\*(WSP) SecondConnectionID)  
 / ("F" ":" 0\*(WSP) [RequestedInfo])  
 / ("Q" ":" 0\*(WSP) QuarantineHandling)  
 / ("T" ":" 0\*(WSP) [DetectEvents])  
 / ("RM" ":" 0\*(WSP) RestartMethod)  
 / ("RD" ":" 0\*(WSP) RestartDelay)  
 / ("A" ":" 0\*(WSP) [Capabilities])  
 / ("ES" ":" 0\*(WSP) [EventStates])  
 / ("PL" ":" 0\*(WSP) [PackageList] ; Auditing only  
 / ("MD" ":" 0\*(WSP) MaxMGCPDatagram ; Auditing only  
 / (extensionParameter ":" 0\*(WSP) [parameterString])

; A final response may include an empty ResponseAck

ResponseAck = confirmedTransactionIdRange  
 \*( "," 0\*(WSP) confirmedTransactionIdRange )

confirmedTransactionIdRange = 1\*9DIGIT ["-" 1\*9DIGIT]

BearerInformation = BearerAttribute 0\*( "," 0\*(WSP) BearerAttribute)

BearerAttribute = ("e" ":" <BearerEncoding>)  
 / (BearerExtensionName [":" BearerExtensionValue])

BearerExtensionName = PackageLCOExtensionName

BearerExtensionValue = LocalOptionExtensionValue

BearerEncoding = "A" / "mu"

CallId = 1\*32(HEXDIG)

; The audit request response may include a list of identifiers

ConnectionId = 1\*32(HEXDIG) 0\*( "," 0\*(WSP) 1\*32(HEXDIG))

SecondConnectionID = ConnectionId

NotifiedEntity = [LocalName "@" ] DomainName [ ":" portNumber ]

LocalName = LocalEndpointName ; No internal structure  
 ; was 1\*32(suitableCharacter)

portNumber = 1\*5(DIGIT)

RequestIdentifier = 1\*32(HEXDIG)

LocalConnectionOptions = LocalOptionValue 0\*(WSP)  
 0\*( "," 0\*(WSP) LocalOptionValue 0\*(WSP))

LocalOptionValue = ("p" ":" <packetizationPeriod> )  
 / ("a" ":" <compressionAlgorithm> )  
 / ("b" ":" <bandwidth> )  
 / ("e" ":" <echoCancellation> )  
 / ("gc" ":" <gainControl> )  
 / ("s" ":" <silenceSuppression> )  
 / ("t" ":" <typeOfService> )  
 / ("r" ":" <resourceReservation> )

```

        / ("k" ":" <encryptiondata>)
        / ("nt" ":" ( <typeOfNetwork> / <supportedTypeOfNetwork> )
        / (localOptionExtensionName[":" <localOptionExtensionValue>])
Capabilities = CapabilityValue 0*(WSP) 0*("," 0*(WSP) CapabilityValue 0*(WSP))
CapabilityValue = LocalOptionValue
        / ("v" ":" <supportedPackages>)
        / ("m" ":" <supportedModes> )
PackageList = pkgNameAndVers 0*("," pkgNameAndVers)
pkgNameAndVers = packageName ":" packageVersion
packageVersion = 1*(DIGIT)
packetizationPeriod = 1*4(DIGIT)["-" 1*4(DIGIT)]
compressionAlgorithm = algorithmName 0*("," algorithmName)
algorithmName = 1*32(SuitableLCOCharacter)
bandwidth = 1*4(DIGIT)["-" 1*4(DIGIT)]
echoCancellation = "on" / "off"
gainControl = "auto" / ["-"]1*4(DIGIT)
silenceSuppression = "on" / "off"
typeOfService = 1*2(HEXDIG) ; 1 hex only for capabilities
resourceReservation = "g" / "cl" / "be"
;encryption parameters are coded as in SDP (RFC 2327)
;NOTE that encryption key may contain an algorithm
;as specified in RFC 1890
encryptiondata = ( "clear" ":" <encryptionKey> )
        / ( "base64" ":" <encodedEncryptionKey> )
        / ( "uri" ":" <URIToObtainKey> )
        / ( "prompt" ) ; defined in SDP, not usable in MGCP!
; was 1*(SuitableCharacter / SP)
encryptionKey = 1*(SuitableLCOCharacter) / quotedString
; See RFC 2045
encodedEncryptionKey = 1*(ALPHA / DIGIT / "+" / "/" / "=")
URIToObtainKey = 1*(SuitableLCOCharacter) / quotedString
typeOfNetwork = "IN" / "ATM" / "LOCAL" / OtherTypeOfNetwork
; Registered with IANA - see RFC 2327
OtherTypeOfNetwork = 1*(SuitableLCOCharacter)
supportedTypeOfNetwork = typeOfNetwork *("," typeOfNetwork)
supportedModes = ConnectionMode 0*("," ConnectionMode)
supportedPackages = packageName 0*("," packageName)
packageName = 1*(ALPHA / DIGIT / HYPHEN) ; Hyphen neither first or last
localOptionExtensionName = VendorLCOExtensionName
        / PackageLCOExtensionName
        / OtherLCOExtensionname
VendorLCOExtensionName = "x" ("+" / "-" ) 1*32(SuitableExtLCOCharacter)
PackageLCOExtensionName = packageName "/" 1*32(SuitablePkgExtLCOCharacter)
; must not start with "x-" or "x+"
OtherLCOExtensionName = 1*32(SuitableExtLCOCharacter)

```

```

localOptionExtensionValue = (1*32(SuitableLCOCharacter) / quotedString)
                           *(";" (1*32(SuitableLCOCharacter) / quotedString))

;Note: No "data" mode.
ConnectionMode = "sendonly" / "recvonly" / "sendrecv"
                / "confrnce" / "inactive" / "loopback"
                / "conttest" / "netwloop" / "netwtest"
                / ExtensionConnectionMode

ExtensionConnectionMode = PkgExtConnectionMode
PkgExtConnectionMode    = packageName "/" 1*(ALPHA / DIGIT)
RequestedEvents = requestedEvent 0*(";" 0*(WSP) requestedEvent)
requestedEvent  = (eventName ["(" requestedActions ")"])
                  / (eventName "(" requestedActions ")" "(" eventParameters ")")
eventName       = [(packageName / "*" ) "/" ] (eventId / "all" / eventRange / "*" / "#") ; for DTMF
                  ["@" (ConnectionId / "$" / "*")]

; Was 1*(SuitableCharacter)
eventId         = 1*(ALPHA / DIGIT / HYPHEN) ; Hyphen neither first or last
eventRange      = ["[" 1*(DigitMapLetter / (DIGIT "-" DIGIT) / (DTMFLetter "-" DTMFLetter)) "]" ]
DTMFLetter      = "A" / "B" / "C" / "D" ; Was undefined
requestedActions = requestedAction 0*(";" 0*(WSP) requestedAction)
requestedAction = "N" / "A" / "D" / "S" / "I" / "K"
                / "E" "(" EmbeddedRequest ")"
                / ExtensionAction

ExtensionAction = PackageExtAction
PackageExtAction = packageName "/" Action ["(" ActionParameters ")"]
Action           = 1*ALPHA
ActionParameters = eventParameters ; May contain actions

;NOTE: Should tolerate different order when receiving, e.g. for NCS.
EmbeddedRequest = ( "R" "(" EmbeddedRequestList ")"
                  ["," 0*(WSP) "S" "(" EmbeddedSignalRequest ")"]
                  ["," 0*(WSP) "D" "(" EmbeddedDigitMap ")"] )
                / ( "S" "(" EmbeddedSignalRequest ")"
                  ["," 0*(WSP) "D" "(" EmbeddedDigitMap ")"] )
                / ( "D" "(" EmbeddedDigitMap ")" )

EmbeddedRequestList = RequestedEvents
EmbeddedSignalRequest = SignalRequests
EmbeddedDigitMap     = DigitMap

SignalRequests = SignalRequest 0*(";" 0*(WSP) SignalRequest )
SignalRequest  = eventName [ "(" eventParameters ")" ]
eventParameters = eventParameter 0*(";" 0*(WSP) eventParameter)
eventParameter = eventParameterValue / eventParameterName "=" eventParameter
                / eventParameterName "(" eventParameters ")"

eventParameterString = 1*(SuitableEventParamCharacter)
eventParameterName   = eventParameterString
eventParameterValue  = eventParameterString / quotedString
DigitMap             = DigitString / "(" DigitStringList ")"

```

```

DigitStringList      = DigitString 0*( "|" DigitString )
DigitString          = 1*(DigitStringElement)
DigitStringElement  = DigitPosition ["."]
DigitPosition       = DigitMapLetter / DigitMapRange
; NOTE "X" is now included
DigitMapLetter      = DIGIT / "#" / "*" / "A" / "B" / "C" / "D" / "T" / "X" / ExtensionDigitMapLetter
ExtensionDigitMapLetter = "E" / "F" / "G" / "H" / "I" / "J" / "K" / "L" / "M" / "N" / "O" / "P" / "Q" / "R"
                    / "S" / "U" / "V" / "W" / "Y" / "Z"
; NOTE "[x]" is now allowed
DigitMapRange       = "x" / "[" 1*DigitLetter "]"
DigitLetter         = *((DIGIT "-" DIGIT) / DigitMapLetter)
ObservedEvents      = SignalRequests
EventStates         = SignalRequests
ConnectionParameters = ConnectionParameter0*( "," 0*(WSP) ConnectionParameter )
ConnectionParameter = ( "PS" "=" packetsSent )
                    / ( "OS" "=" octetsSent )
                    / ( "PR" "=" packetsReceived )
                    / ( "OR" "=" octetsReceived )
                    / ( "PL" "=" packetsLost )
                    / ( "JI" "=" jitter )
                    / ( "LA" "=" averageLatency )
                    / ( ConnectionParameterExtensionName
                        "=" ConnectionParameterExtensionValue )

packetsSent         = 1*9(DIGIT)
octetsSent          = 1*9(DIGIT)
packetsReceived     = 1*9(DIGIT)
octetsReceived      = 1*9(DIGIT)
packetsLost         = 1*9(DIGIT)
jitter              = 1*9(DIGIT)
averageLatency      = 1*9(DIGIT)
ConnectionParameterExtensionName = VendorCPEExtensionName
                        / PackageCPEExtensionName
VendorCPEExtensionName = "X" "-" 2*ALPHA
PackageCPEExtensionName = packageName "/" CPName
CPName = 1*(ALPHA / DIGIT / HYPHEN)
ConnectionParameterExtensionValue = 1*9(DIGIT)
MaxMGCPDatagram    = 1*9(DIGIT)
ReasonCode          = 3DIGIT [1*(WSP) "/" packageName]; Only for 8xx
                    [WSP 1*(%x20-7E)]

SpecificEndpointID = endpointName
SecondEndpointID   = endpointName
RequestedInfo       = infoCode 0*( "," 0*(WSP) infoCode)
infoCode           = "B" / "C" / "I" / "N" / "X" / "L" / "M" / "R" / "S"
                    / "D" / "O" / "P" / "E" / "Z" / "Q" / "T" / "RC" / "LC"

```

```

    / "A" / "ES" / "RM" / "RD" / "PL" / "MD" / extensionParameter
QuarantineHandling = loopControl / processControl
    / (loopControl ", 0*(WSP) processControl )
loopControl      = "step" / "loop"
processControl   = "process" / "discard"
DetectEvents    = SignalRequests
RestartMethod    = "graceful" / "forced" / "restart" / "disconnected"
    / "cancel-graceful" / extensionRestartMethod
extensionRestartMethod = PackageExtensionRM
PackageExtensionRM      = packageName "/" 1*32(ALPHA / DIGIT / HYPHEN)
RestartDelay = 1*6(DIGIT)
extensionParameter = VendorExtensionParameter
    / PackageExtensionParameter
    / OtherExtensionParameter
VendorExtensionParameter = "X" ("-" / "+") 1*6(ALPHA / DIGIT)
PackageExtensionParameter = packagename "/" 1*32(ALPHA / DIGIT / HYPHEN)
; must not start with "x-" or x+"
OtherExtensionParameter = 1*32(ALPHA / DIGIT / HYPHEN)
parameterString = 1*(%x20-7F) ; First and last must not be white space
    / quotedString
MGCPResponse = MGCPResponseLine 0*(MGCPPParameter) *2(EOL *SDPInformation)
MGCPResponseLine = <responseCode> 1*(WSP) <transaction-id>
    [1*(WSP) "/" packageName] ; Only for 8xx
    [WSP <responseString>] EOL
responseCode = 3DIGIT
responseString = *(%x20-7E)
SuitableCharacter = DIGIT / ALPHA / "+" / "-" / "_" / "&"
    / "!" / "" / "|" / "=" / "#" / "?" / "/"
    / "." / "$" / "*" / ";" / "@" / "[" / "]"
    / "^" / "`" / "{" / "}" / "~"
SuitablePkgExtLCOCharacter = SuitableLCOCharacter
SuitableExtLCOCharacter = DIGIT / ALPHA / "+" / "-" / "_" / "&"
    / "!" / "" / "|" / "=" / "#" / "?" /
    / "." / "$" / "*" / ";" / "@" / "[" / "]"
    / "^" / "`" / "{" / "}" / "~"
SuitableLCOCharacter = SuitableExtLCOCharacter / "/"
SuitableEventCharacter = DIGIT / ALPHA / "+" / "-" / "_" / "&"
    / "!" / "" / "|" / ";" / "?" / "/"
    / "." / "$" / "*" / ";" / "@" / "[" / "]"
    / "^" / "`" / "{" / "}" / "~"
; VCHAR except "", "(", ")", and ", "
SuitableEventParamCharacter = %x21 / %x23-27 / %x2A-2B / %x2D-7E
; NOTE: UTF8 encoded
quotedString = DQUOTE 0*(quoteEscape / quoteChar) DQUOTE
quoteEscape = DQUOTE DQUOTE

```

quoteChar = (%x00-21 / %x23-FF)

EOL = CRLF / LF

HYPHEN = "-"

SDPinformation = ;See RFC 2327

