了解IOS和IOS XE呼叫路由

目录

简介

先决条件

- 要求
- 使用的组件

背景信息

通用定义

命令和功能路线图

Cisco IOS/Cisco IOS XE呼叫路由基础知识

- 语音拨号对等体类型
- 入站拨号对等体匹配
  - 当不存在匹配项时/Default Dial-Peer 0 peer_tag=0, pid:0
- 出站拨号对等体匹配
- 号码字符串拨号对等体寻线
- URI拨号对等体寻线
  - 语音类URI
  - 入站URI拨号对等体匹配
  - 常见使用案例
  - 配置示例
  - 出站URI拨号对等体匹配
  - 配置示例
  - 配置示例
- 拨号对等体通配符
- 拨号对等体状态
- 虚拟路由和转发(VRF)和拨号对等体寻线
  - 与VRF匹配的入站拨号对等体
  - 出站拨号对等体与VRF匹配
  - VRF和拨号对等体组配置示例

高级呼叫路由技术

- 拨号对等体组
  - 配置示例
- E164 — 模式映射
  - CLI配置示例 — 呼叫号码
  - CLI配置示例 — 被叫号码
  - 闪存配置示例
- 目标服务器组
  - 配置示例 — 普通
- 目标服务器组和选项Keepalive
- 呼出代理
- POTS中继组
  - 配置示例
- 语音类租户

- 租户的命令首选项顺序
- 多租户配置示例
- ILS URI呼叫CUBE（语音类路由字符串）
- 配置示例CUCM - SIP - CUBE - SIP - CUCM

传统呼叫路由技术

- DNIS映射
- 配置示例
- 中继组标签
- 配置示例
- 编号类型
- 配置示例
- 拨号对等体数据
- 配置示例
- Voice Source-Group
- 拨号对等体权限

URI和数字处理

- 通过POTS拨号对等体进行数字处理
- 通过语音转换规则和配置文件进行数字处理
- 语音类e164 — 转换
- 通过ISDN映射进行数字处理
- 配置示例
- 通过号码扩展(num-exp)进行数字处理
- 配置示例
- 入站/出站SIP配置文件
- SIP复制列表

特别说明

- 协议信令和媒体绑定
- 配置示例
- DNS和VoIP拨号对等体
- 最大连接和带宽
- 配置示例
- 直接拨入（DID）
- 单阶段拨号
- 配置示例
- 两阶段拨号
- 配置示例
- 阻止呼叫
- 双应答配置示例
- ISDN overlap-receiving命令
- 空被叫号码
- 示例输出
- 限制类别
- Cisco Unified Communications Manager Express(CUCME)拨号对等体
- 带有拨号对等体的MGCP和SCCP
- 带有拨号对等体的SIP DSAPP

呼叫路由故障排除和验证

简介

本文档介绍 Cisco IOS® 和 Cisco IOS XE 呼叫路由。

先决条件

要求

虽然阅读本文档无需任何正式前提条件，但撰写本文档时希望读者已经具备一些用于建立和连接电话呼叫的基础语音信令协议知识。这些协议自始至终被引用多次。

信令协议：会话初始协议(SIP)、H323(h225/h245)、媒体网关控制协议(MGCP)、瘦客户端控制协议(SCCP)、ISDN Q931、E1 R2。

媒体协议：实时协议(RTP)、语音编解码器、视频编解码器。

模拟技术：Ear and Mouth(E&M)、Foreign Exchange Subscriber(FXS)和Foreign Exchange Office(FXO)。

使用的组件

本文档中的信息基于以下软件和硬件版本：

- Cisco IOS和Cisco IOS XE网关
- 2800 / 3800 / 2900 / 3900 / 4300 / 4400 / CSR1000v / CAT8000v / ASR100X / C8200 / C8300 / ISR1100

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

本文档讨论使用普通老式电话服务(POTS)和IP语音(VoIP)网络呼叫段进行入站和出站拨号对等体匹配背后的机制。

除拨号对等体信息外，本文档还介绍了与呼叫路由相关的重要主题。其中包括数字处理、会话初始

协议(SIP)消息处理的快速概述、限制呼叫功能的几种方法、快速媒体和信令绑定概述，以及最后一点故障排除。

本文档使用配置示例以及debug和show命令输出作为参考点。本文档中的许多功能都清楚地标有该功能在Cisco IOS和Cisco IOS XE中引入的版本。此信息也可在命令和功能路线图部分快速参考。如果有明显的缺陷，它会与文本链接，以便读者了解。

通用定义

属性	描述
数字字符串	也称为号码字符串、电话号码、号码或E164号码。完全由数字0到9组成，带可选的加号前导(+)。 示例： 8675309 123456789 +1972525222 +442084445555 +85225353333
拨号号码识别服务(DNIS)	这是呼叫的被叫号码或目标号码。
自动号码识别(ANI)	这是呼叫的主叫号码或始发主叫号码。这也称为呼叫线路标识符(CLID)，也可以命名为呼叫方ID。
统一资源标识符 (URI)	URI是sip:或电话：最常用于VoIP协议SIP和H323的字符串。 URL示例： sip:user@example.com。 sip:user@sub.example.com。 sip:user@192.0.2.1 sip:user@2001:4860:4860::8888 电话：8675309 sip:example.com。
运营商ID	CID示例： cid:orange@example.com。

属性	描述
	cid:orange@sub.example.com。 cid:orange@192.0.2.1 cid:orange@2001:4860:4860::8888 注意：Cisco Bug ID CSCua14749 Carrier-ID在Cisco IOS XE平台上不起作用。
路由字符串	用于SIP的ILS路由字符串的思科专有报头。 示例： X-cisco-dest-route-string:<sip:configured-value>
枚举	ENUM是一种使用域名服务(DNS)将E164电话号码转换为URI的协议。本文档不对此进行介绍。
PSTN	公共交换电话网
ITSP	互联网电话服务提供商
SBC	会话边界控制器。 此设备作为LAN和ITSP/PSTN网络之间的分界点

命令和功能路线图

功能	Cisco IOS version（Cisco IOS 版本）	Cisco IOS XE版本
号码扩展(num-exp) 拨号对等体（POTS和VOIP） answer-address destination-pattern	11.3(1)公吨	all

incoming called-number 会话目标（IPv4和DNS） 最大连接数（最大连接数） direct-inward-dial 转发数字(POTS) 前缀(POTS) timeouts inter-digit(voice-port)		
dial-peer terminator	12.0	all
huntstop	12.0(5)T	all
ISDN映射	12.0(6)公吨	all
拨号对等体搜寻方案	12.0(7)XK	all
语音转换规则和配置文件 translate-outgoing numbering-type 数字条带(POTS)	12.0.(7)XR1	all
会话目标(sip-server)	12.1(1)公吨	all
POTS中继组	12.1(3)T	all
DNIS-Map（出站）	12.2(2)XB	all
中继组标签	12.2(11)吨	all
拨号对等体（数据）	12.2(13)吨	all

语音类URI（出站）	12.3(4)公吨	all
出站代理	12.4(15)吨	all
会话目标(IPv6)	12.4(22)吨	all
SIP配置文件（出站）	15.0(1)米	all
语音类URI（入站） voice source-group	15.1(2)公吨	3.8秒
SIP复制列表 会话目标（注册器）	15.1(3)公吨	3.6秒
call-route(url)	15.2(1)公吨	3.3秒
max-bandwidth	15.2(2)公吨	3.7秒
E164-Pattern-Maps（出站）	15.2(4)米	3.7秒
语音类Route-String call-route(dest-route-string)	15.3(3)米	3.10秒
拨号对等体组(VOIP) E164 — 模式映射（入站） 目标服务器组 requri-passing 会话目标(sip-uri)	15.4(1)公吨	3.11秒
拨号对等体调配策略	15.4(2)公吨	3.12秒

SIP配置文件（入站）		
拨号对等体组(POTS)	15.5(1)公吨	3.14秒
语音类租户	15.6(2)公吨	16.3.1
拨号对等体的VRF过滤	15.6(3)米	16.3.1
e164转换	不适用	16.8.1
SIP DSAPP	不适用	16.12.1
服务器组的Huntstop	不适用	17.4.1
租户的SIP侦听端口和拨号对等体的租户过滤	不适用	17.8.1
基于DNS SRV的保持连接选项	不适用	17.9.1

Cisco IOS/Cisco IOS XE呼叫路由基础知识

Cisco IOS和Cisco IOS XE网关利用拨号对等体的概念来控制呼叫路由和每个呼叫段功能的协商。呼叫支路是两个呼叫代理之间的双向通信。呼叫代理是发起、处理或转发电话呼叫的设备。这可以而且不限于电话提供商设备、Cisco网关、IP电话、Cisco Unified Communication Manager(CUCM)、Cisco Unity connection(CUC)等。有太多呼叫代理要列出。

场景:呼叫从另一个呼叫代理到达思科网关，是入站呼叫段（内置）。 网关处理呼叫，并根据其处理将呼叫发送到下一个呼叫代理。这是出站呼叫段（出站）。

图1显示通过思科语音网关从PSTN到CUCM路由的呼叫以及相应的入站和出站呼叫段信息。

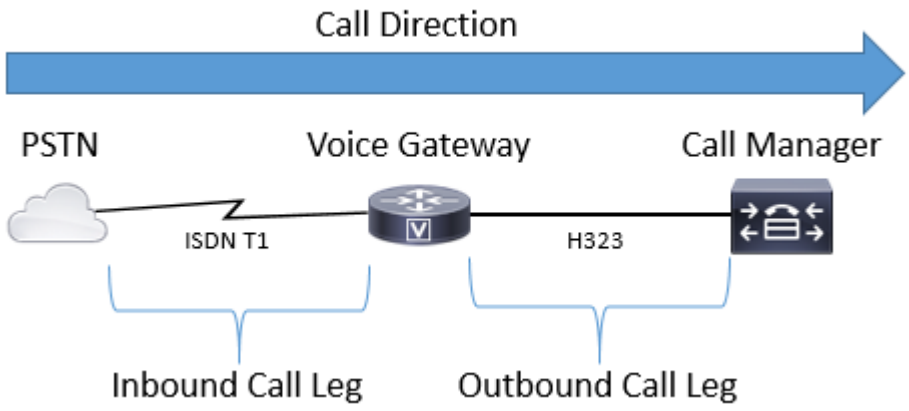


图1 — 图示的入站和出站呼叫段

通过Cisco Gateway ALWAYS的成功呼叫（请参阅注释）与入站或出站拨号对等体匹配以正确路由。入站和出站拨号对等体类似于前面提到的呼叫段。在图1中，呼叫从PSTN到达思科网关，需要匹配入站拨号对等体。然后，网关使用出站拨号对等体将呼叫路由到下一个呼叫代理。请务必记住，这些术语是从思科网关的角度定义的。

通过为呼叫的每一端匹配拨号对等体，管理员可以控制每个特定呼叫段的许多方面。这些设置包括语音编解码器、DTMF首选项、数字处理、呼叫路由位置以及许多其他设置。可以使用入站和出站匹配语句配置拨号对等体，因此，如果有效的入站和出站匹配配置应用到特定拨号对等体，则有可能为内支路和外支路匹配同一拨号对等体。

注意：此规则的例外是MGCP和SCCP语音端口。在呼叫路由过程中，这些信令协议不遵循正常的拨号对等体匹配机制。有关详细信息，请参阅SCCP和MGCP部分。

图2显示了与图1相同的呼入和呼出呼叫段，但是通过思科语音网关对从PSTN到CUCM路由的呼叫使用各自的拨号对等体。

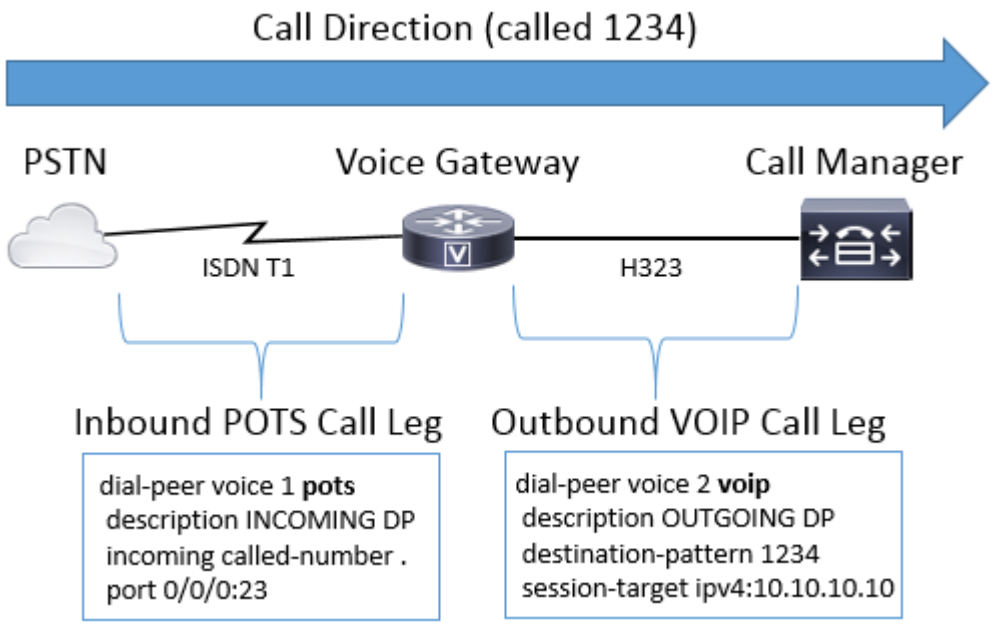
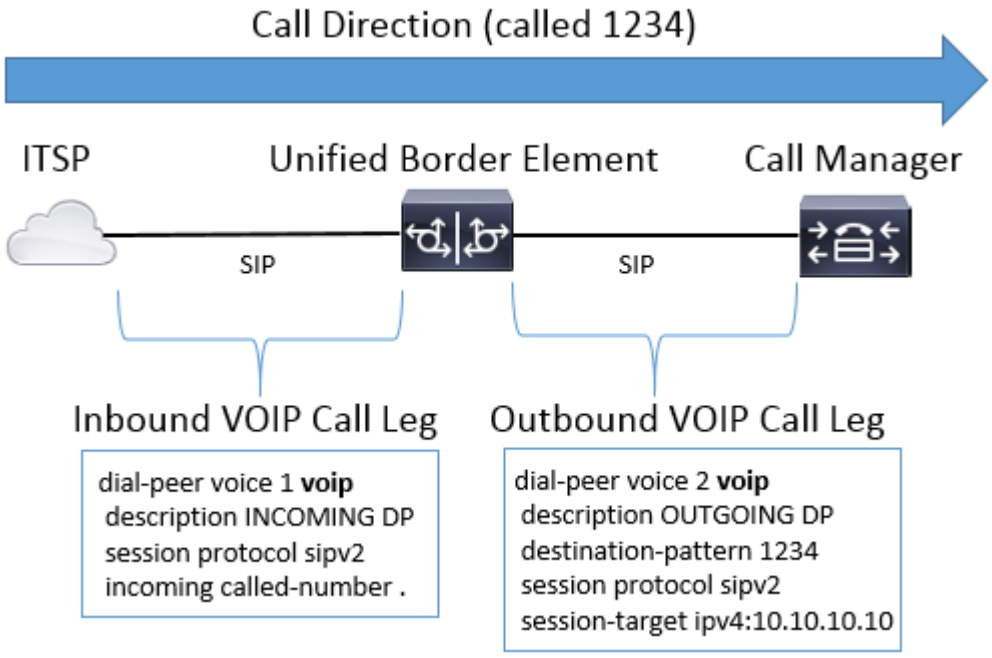


图2 — 图示的入站和出站拨号对等体

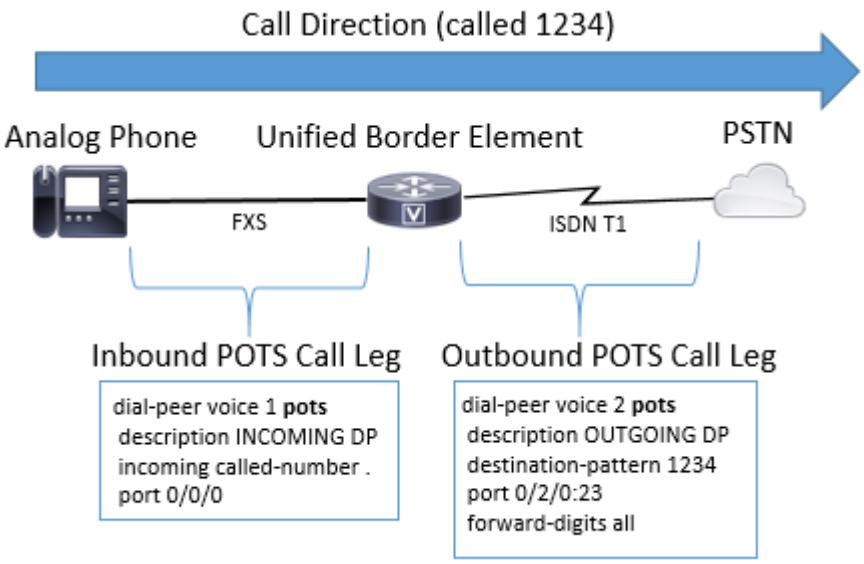
Cisco语音网关可以集成多种不同类型的语音呼叫和协议，包括IP到IP、POTS到POTS和IP到POTS，反之亦然。

图3显示了通过思科统一边界要素(CUBE)进行的VoIP到VoIP呼叫。



图像3 - Voip到VoIP呼叫的入站和出站拨号对等体

图4显示通过思科网关的POTS到POTS呼叫。



映像4 - POTS到POTS呼叫的入站和出站拨号对等体

语音拨号对等体类型

POTS	普通老式电话服务拨号对等体匹配模拟连接，例如模拟FXS、FXO、ISDN T1/E1、E1 R2以及耳和嘴(E&M)连接。 这些设备向网关上的物理语音端口发送呼叫，或从网关上的物理语音端口接收呼叫。
VOIP	IP语音拨号对等体主要用于控制进出网关的H323和SIP连接。 这些拨号对等体使用域名系统(DNS)发送和接收来自IPv4和IPv6地址以及完全限定域名(FQDN)的信令。 — VoIP拨号对等体还可用于帧中继语音(VoFR)、ATM语音(VoATM)、高级数据链路控制语音(VoHDLC)以及这些拨号对等体的注册、准入和状态(RAS)信令和会话目标，还可以包括结算和ENUM值。 注意：其中一些配置是较旧的技术，在较新的网络中看不到，而对于Cisco IOS XE，有些配置不再受支持。因此，本文档中并未涉及这些内容。
MMOIP	Multimedia Mail Over IP拨号对等体用于将电子邮件发送到交换服务器。 这些主要用于t37入匝道/出匝道传真。这些拨号对等体类型不在本文档的讨论范围内。

注:思科网关上可配置的最大拨号对等体数量取决于可用内存(DRAM)。 每个拨号对等体消耗大约6KB的内存，因此请确保网关至少为其他CPU进程保留总内存的20%。配置的大量拨号对等体可能会增加路由呼叫的延迟。当思科语音应用从上到下查看拨号对等体时(类似于访问控制列表(ACL))，这一点可能很重要。 对于较新的Cisco网关，这通常不是问题。

示例错误：

```
May 26 12:59:46.406: %DIALPEER_DB-3-ADDPEER_MEM_THRESHOLD: Addition of dial-peers limited by available m
```

入站拨号对等体匹配

当思科网关收到呼叫建立请求时，网关开始搜索此呼叫的适用传入拨号对等体。这不是逐位分析；而是使用完整消息来确定选择了哪个入站拨号对等体。检查的消息中的项目顺序在很大程度上取决于呼叫的协议，如表1、表2和表3中定义的首选项列表所示。拨号对等体只需满足其中一个匹配条件。不需要在拨号对等体中配置所有属性或每个属性与呼叫建立信息匹配。根据第一个匹配条件搜索所有拨号对等体。仅当找不到匹配项时，网关才会转到下一个条件。

表1.入站SIP拨号对等体选择首选项

首选项	匹配条件	Dial-peer命令
1	URI	通过<uri-tag>传入的uri
2	URI	传入uri请求<uri-tag>
3	URI	<uri-tag>的传入uri
4	URI	来自<uri-tag>的传入uri
5	被叫号码	incoming called-number <number-string> incoming called e164-pattern-map <pattern-map-number>
6	主叫号码	来电e164-pattern-map <pattern-map-number>

		answer-address <number-string>
7	Destination-pattern(ANI)	destination-pattern <number-string>
8	运营商ID	carrier-id source <string>



注意：合格入站拨号对等体可以按VRF或租户进行过滤。如果配置了适用的功能。有关详细信息，请参阅虚拟路由和转发(VRF)以及拨号对等体搜寻和语音类租户部分。

表2.入站H323拨号对等体选择首选项

首选项	匹配条件	Dial-peer命令
1	URI	传入uri called <uri-tag> 传入uri calling <uri-tag>
2	被叫号码	incoming called-number <number-string> incoming called e164-pattern-map <pattern-map-number>
3	主叫号码	来电e164-pattern-map <pattern-map-number> answer-address <number-string>
4	Destination-pattern(ANI)	destination-pattern <number-string>
5	运营商ID	carrier-id source <string>

表 3. 入站汇聚POTS拨号对等体选择首选项

首选项	匹配条件	Dial-peer命令
1	被叫号码	incoming called-number <number-string>

2	主叫号码	answer-address <number-string>
3	Destination-pattern(ANI)	destination-pattern <number-string>
4	语音端口	port <voice-port-number>

当不存在匹配项/默认拨号对等体0 peer_tag=0时，pid:0

当POTS或VoIP呼叫的入站拨号对等体没有符合条件的匹配项时，网关将分配拨号对等体0。这不是理想的选择，因为拨号对等体0功能有限，并且可能导致呼叫问题。SCCP和MGCP协议是此协议的异常值，它们不使用拨号对等体来路由呼叫。有关详细信息，请参阅MGCP和SCCP部分。

dial-peer 0功能

- 没有dtmf中继机制。
- 通告了VoIP呼叫的所有语音编解码器。
- 传真速率语音.
- 已启用语音活动检测(VAD)。
- 不支持 rsvp.
- 不支持POTS呼叫的IVR应用。
- 已启用直接拨入。
- 不支持VRF。

出站拨号对等体匹配

出站拨号对等体用于将POTS或VoIP呼叫从网关路由到下一个呼叫代理。与入站拨号对等体匹配类似，网关可以根据特定协议的首选项顺序使用项目列表来匹配拨号对等体。但是，与入站拨号对等体不同，如果没有合格的出站拨号对等体来路由呼叫，则呼叫将失败。与入站拨号对等体匹配一样，将根据第一个匹配条件搜索所有拨号对等体。仅当找不到匹配项时，网关才会转到下一个条件。

表4.出站SIP拨号对等体选择首选项

首选项	匹配条件	Dial-peer命令
1	拨号对等体组拨号对等体	destination dpg <dpg-tag> （在入站拨号对等体上配置DPG）

2	拨号对等体调配策略URI	destination uri-from <uri-tag> destination uri-to <uri-tag> destination uri-via <uri-tag> destination uri-distribution <uri-tag> destination uri-referred-by <uri-tag> （在入站拨号对等体上配置DPP）
3	ILS路由字符串	destination route-string <route-string-tag>
4	URI和运营商ID	destination uri <uri-tag> AND carrier-id target <string>
5	被叫号码和运营商ID	destination-pattern <number-string> AND carrier-id target <string>
6	URI	目标uri <uri-tag>
7	被叫号码	destination-pattern <DNIS-number> 目标e164-pattern-map <pattern-map-number> dnis-map <dnis-map-number>
8	主叫号码	destination calling e164-pattern-map <pattern-map-number>

注意： 使用calling e164-pattern-map配置传入和传出拨号对等体时，必须将拨号对等体调配策略应用于传入拨号对等体，以避免呼叫失败。

表5.出站H323拨号对等体选择首选项

首选项	匹配条件	Dial-peer命令
1	拨号对等体组拨号对等体	destination dpg <dpg-tag> （在入站拨号对等体上配置）
2	URI和运营商ID	destination uri <uri-tag> AND carrier-id target <string>

3	被叫号码和运营商ID	destination-pattern <number-string> AND carrier-id target <string>
4	URI	目标uri <uri-tag>
5	被叫号码	destination-pattern <number-string> 目标e164-pattern-map <pattern-map-number> dnis-map <dnis-map-number>
6	主叫号码	destination calling e164-pattern-map <pattern-map-number>

表6.出站POTS拨号对等体选择首选项

首选项	匹配条件	Dial-peer命令*
1	拨号对等体组拨号对等体	destination dpg <dpg-tag>（在入站拨号对等体上配置）
2	URI和运营商ID	destination uri <uri-tag> AND carrier-id target <string>
3	被叫号码和运营商ID	destination-pattern <number-string> AND carrier-id target <string>
4	URI	目标uri <uri-tag>
5	被叫号码	destination-pattern <DNIS-number>dnis-map <map-number>

注意：Number String Dial-Peer Hunting和URI Dial-Peer Hunting部分将介绍网关如何评估每个匹配条件行的潜在命令列表，然后再移至下一个匹配条件。例如，它会在检查呼叫号码命令之前评估所有潜在的destination-pattern匹配和目标e164-pattern-map匹配命令。

号码字符串拨号对等体寻线

数字字符串首选项：

与URI具有用于评估匹配项的特定操作顺序类似，在评估数字数字字符串时也会使用一组规则。思科网关的默认拨号对等体寻线方案设置为0。这意味着网关搜索匹配时间最长（最具体）的模式。如果有两个匹配长度相同的拨号对等体，网关会查看明确定义的拨号对等体首选项。最后，如果两者相同，它会随机选择一个。

还有其他可用于配置的拨号对等体寻线方案；但是，大多数部署都保留默认值0。


提示：如果在默认顺序之外匹配拨号对等体，管理员可以检查非默认拨号对等体寻线方案的运行配置。

```
Gateway(config)# dial-peer hunt ?
  <0-7>  Dial-peer hunting choices, listed in hunting order within each choice:
  0 - Longest match in phone number, explicit preference, random selection.
  1 - Longest match in phone number, explicit preference, least recent use.
  2 - Explicit preference, longest match in phone number, random selection.
  3 - Explicit preference, longest match in phone number, least recent use.
  4 - Least recent use, longest match in phone number, explicit preference.
  5 - Least recent use, explicit preference, longest match in phone number.
  6 - Random selection.
  7 - Least recent use.
```

最长的匹配号码字符串拨号对等体算法查找序列中具有最多号码的拨号对等体，该序列与号码字符串中的号码序列完全匹配。此概念在后面的场景中进行了说明。

场景:符合条件的拨号对等体已配置这些可能的匹配项，网关正在评估数字字符串2001。拨号对等体1可以匹配从2000到2999的任何数字，而拨号对等体2可以匹配2000到2009。此呼叫将匹配拨号对等体2，因为它是在使用默认拨号对等体搜寻机制（拨号对等体寻线0）时数字字符串2001的最长匹配项（最具体的）。 换句话说，编号序列200是与编号字符串2001中的编号序列完全匹配的最大序列。

```
!
dial-peer voice 1 voip
 destination-pattern 2...
!
dial-peer voice 2 voip
 destination-pattern 200.
!
```

首选项定义为管理员为每个拨号对等体定义的权重。管理员可以配置首选项，以便呼叫始终先于其它呼叫使用特定的拨号对等体。默认情况下，所有拨号对等体都是首选项0。具有首选项0的拨号对等体在另一个具有首选项1至10的拨号对等体之前匹配。大多数管理员设置多个拨号对等体以将呼

叫发送到特定CUCM用户，备份用户或使用具有较低首选项（配置了较高号码）的另一个拨号对等体配置的另一个呼叫代理进行配置。

场景:两个拨号对等体配置有相同的匹配长度，用于匹配数字字符串2001。管理员定义显式首选项。由于两个拨号对等体的匹配长度相同，因此网关的评估结果相同。但是，管理员将拨号对等体1设置为较高的首选项，以便选择拨号对等体作为路由呼叫时使用的第一个拨号对等体。如果第一个拨号对等体发生故障，Dial-Peer 2将保留为辅助选项。

```
!
dial-peer voice 1 voip
 destination-pattern 2...
 preference 1
!
dial-peer voice 2 voip
 destination-pattern 2...
 preference 2
!
```

Cisco网关每次仅尝试通过一个符合条件的出站拨号对等体路由呼叫。如果在第一个选定的拨号对等体上观察到故障情况，则网关会尝试将呼叫路由到下一个符合条件的拨号对等体。这会一直持续到呼叫成功或失败，因为没有更多符合条件的拨号对等体可以尝试。拨号对等体搜寻和故障的常见症状是进行呼叫时回铃明显延迟。通常需要调试来验证指定拨号对等体上呼叫失败的确切原因。如果管理员不希望网关在发现故障情况时查找另一个拨号对等体，则可以在拨号对等体上使用huntstop命令。

场景:两个拨号对等体配置有相同的匹配长度，用于匹配数字字符串2001。管理员已定义明确的首选项，不希望与此特定呼叫的拨号对等体2匹配。由于有两个具有相同match-length的拨号对等体，因此该首选项用于确定拨号对等体。Dial-Peer 1具有最低配置的首选项号码，因此用于路由呼叫。如果使用dial-peer 1的出站呼叫段发生故障情况，则网关会立即停止拨号对等体寻线，因为配置了huntstop命令。在此方案中，拨号对等体2从未用于出站路由。

```
!
dial-peer voice 1 voip
 destination-pattern 2...
 preference 1
 huntstop
!
dial-peer voice 2 voip
 destination-pattern 2...
 preference 2
!
```



注意：huntstop和preference命令也可以与URI匹配语句一起使用，因为它们是通用的拨号对等体配置命令。此外，语音类服务器组配置可以在17.4.1a中使用huntstop命令。有关此问题

的详细信息，请参阅目标服务器组部分。

URI拨号对等体寻线

网关查看每个匹配条件，并在转到下一个匹配条件之前将其耗尽。例如，在入站SIP呼叫中。根据表1.入站SIP拨号对等体选择首选项，思科网关首先检查的是URI并评估所有可能的URI命令，以找到适合的URI命令。如果没有匹配项，或者没有配置任何匹配项，则网关将移至下一个匹配项，并对该条件进行评估。此过程会重复，直到呼叫根据匹配路由或网关超出匹配条件进行检查为止。

当使用URI命令配置入站或出站拨号对等体时，网关会检查在多个报头中收到的URI，以查找可能的匹配项。匹配首选项基于最具体的匹配项，确切的首选项为完整URI匹配、主机部分、用户部分或电话URI。了解URI匹配的操作顺序可以极大地帮助拨号对等体与SIP和CUBE部署进行匹配。

可以使用voice class uri sip preference命令操作此首选项顺序，以将用户ID指定为第一个选项，而不是主机。

URI首选项：

1. URI的主机部分。示例：(@<ip_address>或@example.com。)
2. URI的用户部分。示例：(sip:8675309或sip:user)
3. 电话URI。示例：(电话：18005532447)
4. 与完整URI完全匹配。示例：(user@host.domain.name、user@<ip_address>、8675309@example.com.、8675309@example.com。)

支持文档：Cisco Unified Border Element配置指南 — Cisco IOS XE 17.6及以上版本

场景:管理员已配置此拨号对等体并将呼叫发送到网关。收到的Invite中的From信头是From:<sip:testuser@192.0.2.1>。网关可能基于此报头匹配两个不同的拨号对等体。拨号对等体1基于用户部分，拨号对等体2基于主机部分。但是，由于主机匹配优先于用户匹配，因此拨号对等体2用于呼叫中的入站拨号对等体。

```
!
voice class uri URI1 sip
 user-id testuser
!
voice class uri URI2 sip
 host ipv4:192.0.2.1
!
dial-peer voice 1 voip
 sess protocol sipv2
 incoming uri FROM URI1
!
dial-peer voice 2 voip
```

```
 sess protocol sipv2
 incoming uri FROM URI2
!
```

语音类URI

入站和出站拨号对等体的URI匹配使管理员能够灵活地在消息中支持URI的VoIP协议的多个电话号码字符串上执行匹配。在Cisco IOS 15.4(1)T和Cisco IOS-XE 3.11S之前，请求URI必须包含字母数字user@host，否则思科网关将拒绝带有4xx消息的呼叫。现在URI可以只包含主机部分，网关仅根据提供的主机路由呼叫。例如，sip:cisco.com。

此外，在Cisco IOS 15.4(1)T和Cisco IOS-XE 3.11S voice-class URI user-id之前只能是数字e.164值(sip:1234@example.com.)。 已更改此设置，以便管理员可以在CUBE(sip:user@example.com)上配置字母数字用户ID。

语音类uri的主机或用户部分可以包含正则表达式(regex)模式，这些模式大大扩展了可以匹配的可能值。

```
Gateway(config-voice-uri-class)# user-id .)
% unmatched ()user-id pattern can be of format ^([][0-9A-Za-z\|\/()*+^$&?#--.])*$

Gateway(config-voice-uri-class)# host .)
% unmatched ()host pattern can be of format ^([][0-9A-Za-z\|@\/()*+^$&?#--.])*$

Gateway(config-voice-uri-class)# pattern .)
% unmatched ()pattern pattern can be of format ^([][0-9A-Za-z\|@;:=%!~\/()*+^$&?#--.])*$
```

示例：语音类URI

```
!
voice class uri HOST sip
 host example.com.
 host dns:cisco.example.com.
 host ipv4:10.50.244.2
 host ipv6:[2001:4860:4860::8888]
!
voice class uri USER sip
 user-id username
!
voice class uri PATTERN sip
 pattern 8675309
!
voice class uri HostRegex sip
```

```
 host (.*)example.com.
!
voice class uri ipRegex sip
 host 172\.18\.110\.20[567]
!
voice class uri PatternRegex sip
 pattern 555(.*)
!
voice class uri ipRegex sip
 pattern (172\.18\.110\.10[134]|10\.10\.10\.10)
 ! One Line that matches 172.18.110.101, 172.18.110.103, 172.18.110.104 OR 10.10.10.10
!
voice class uri UserRegex sip
 user-id test(.*)
!
```

每个语音类URI只能配置10台主机、1个模式或1个用户ID，如以下示例所示。如果需要匹配更多项目，建议使用Regex。

```
Gateway(config)# voice class uri TEST sip
Gateway(config-voice-uri-class)#host ipv4:10.1.1.1
Gateway(config-voice-uri-class)#host ipv4:10.2.2.2
Gateway(config-voice-uri-class)#host ipv4:10.3.3.3
Gateway(config-voice-uri-class)#host ipv4:10.4.4.4
Gateway(config-voice-uri-class)#host ipv4:10.5.5.5
Gateway(config-voice-uri-class)#host ipv4:10.6.6.6
Gateway(config-voice-uri-class)#host ipv4:10.7.7.7
Gateway(config-voice-uri-class)#host ipv4:10.8.8.8
Gateway(config-voice-uri-class)#host ipv4:10.9.9.9
Gateway(config-voice-uri-class)#host ipv4:10.10.10.10
Gateway(config-voice-uri-class)#host ipv4:10.11.11.11
Error:Maximum of 10 hosts can only be configured.

Gateway(config)# voice class uri TEST2 sip
Gateway(config-voice-uri-class)#host dns:1.com
Gateway(config-voice-uri-class)#host dns:2.com
Gateway(config-voice-uri-class)#host dns:3.com
Gateway(config-voice-uri-class)#host dns:4.com
Gateway(config-voice-uri-class)#host dns:5.com
Gateway(config-voice-uri-class)#host dns:6.com
Gateway(config-voice-uri-class)#host dns:7.com
Gateway(config-voice-uri-class)#host dns:8.com
Gateway(config-voice-uri-class)#host dns:9.com
Gateway(config-voice-uri-class)#host dns:10.com
Gateway(config-voice-uri-class)#host dns:11.com
Error:Maximum of 10 hosts can only be configured.

Gateway(config)# voice class uri TEST3 sip
Gateway(config-voice-uri-class)#user-id 8675309
Gateway(config-voice-uri-class)#user-id 123456789
Gateway(config-voice-uri-class)#do sh run | s TEST3
voice class uri TEST3 sip
user-id 123456789

Gateway(config)# voice class uri TEST4 sip
Gateway(config-voice-uri-class)#pattern 8675309
```

```
Gateway(config-voice-uri-class)#pattern 123456789
Gateway(config-voice-uri-class)#do sh run | s TEST4
voice class uri TEST4 sip
pattern 123456789
```

入站URI拨号对等体匹配

此功能已添加到Cisco IOS 15.1(2)T和Cisco IOS-XE 3.8S中，并使用已配置并应用于入站拨号对等体的语音类uri。许多用户通过传统的incoming called-number语句为SIP呼叫采用传入URI，因为它是选择入站拨号对等体时检查的第一个匹配条件。该命令还使管理员能够更好地匹配来自特定呼叫座席或用户的呼叫。

完整文档：[思科统一边界要素配置指南 — Cisco IOS XE 17.6及以上版本]

常见使用案例

1. URI主机部分上的入站拨号对等体匹配，用于应答来自CUCM的OPTIONS ping请求。
2. 在URI的主机部分上匹配的入站拨号对等体，用于控制来自Internet电话服务提供商(ITSP)的入站呼叫
3. 在URi的用户ID部分上匹配的入站拨号对等体，用于某些用户或号码的呼叫处理。

配置示例

此示例输出匹配来自语音类URI中定义的两个主机IP的任何SIP请求的拨号对等体777。标头监视被定义为拨号对等体上的From标头；但是，管理员可以定义许多其他属性，包括VIA、TO和REQUEST（请求URI）。 如果CUCM现在向CUBE发送OPTIONS ping命令，匹配拨号对等体777，并从指定接口向OPTIONS发出my 200 OK应答。如果CUCM向CUBE发送Invite，则将拨号对等体777匹配为入站拨号对等体。

```
!
voice class uri CUCM sip
 host ipv4:10.50.244.2
 host ipv4:10.50.244.20
!
dial-peer voice 777 voip
 description INCOMING URI
 session protocol sipv2
 incoming uri from CUCM
 voice-class sip bind control source-interface Loopback777
 voice-class sip bind media source-interface Loopback777
!
```

出站URI拨号对等体匹配

Cisco IOS网关可以通过将voice class uri应用于出站拨号对等体并将呼叫路由url添加到全局配置来匹配使用URI的出站拨号对等体。如果存在此项，CUBE可以尝试根据请求URI路由呼叫。此功能已添加到Cisco IOS 12.3(4)T中，并且存在于所有Cisco IOS XE版本中。请注意，默认情况下，传出SIP Request-URI和To报头URI具有出站拨号对等体的会话目标。可以使用命令requri-passing禁用此功能，该命令允许网关将内置URI主机部分传递到外置URI主机部分，而不是使用session-target替换URI主机部分。命令requri-passing添加在15.4(1)T和Cisco IOS XE 3.11S中。

配置示例

```
voice service voip
 sip
  call-route url
  requri-passing
!
voice class uri CUCM sip
 host dns:.*.com
!
dial-peer voice 777 voip
 description OUTGOING URI
 session protocol sipv2
 destination uri CUCM
 session target sip-uri
!
```

来源：[思科统一边界要素配置指南 — Cisco IOS XE 17.6或更高版本]

除语音类URI外，管理员可以使用拨号对等体调配策略(DPP)匹配出站拨号对等体匹配的站内URI。此功能已添加到Cisco IOS 15.4(2)T和Cisco IOS XE 3.12S。拨号对等体调配策略需要定义主要匹配属性，而次要匹配属性是可选的。调配策略应用于入站拨号对等体，当选择该拨号对等体用于入站呼叫段时，调用该策略。结果是根据拨号对等体调配策略中的属性选择出站拨号对等体。

出站匹配可以是单个报头或多个报头，这些报头必须全部为真才能与拨号对等体匹配。

在本例中，From和To报头有一个语音类URI。对于OR匹配，配置包含两个首选项的拨号对等体调配策略。From信头是第一个首选项，To信头是备份首选项。拨号对等体1234旨在为入站匹配应用调配策略。然后，建立分别应用destination uri-from和destination uri-to命令的拨号对等体11111和22222。这些命令指向它们的语音类URI。对于呼叫，您可以接收邀请、匹配拨号对等体1234并检查调配策略。然后，设备可以尝试首先在From报头上进行路由，该报头作为拨号对等体上的适用匹配11111。如果失败，您还可以尝试使用22222在to报头上进行路由。

该示例还详细说明了如何实现And匹配拨号对等体调配策略。假设收到相同的Invite，您可以在一个

首选项下定义两个信头，并将此信头应用于入站拨号对等体。

现在，收到邀请后，它可以检查符合条件的出站拨号对等体，这些拨号对等体同时满足调配策略中定义的匹配条件。因此，在本示例中，您的出站拨号对等体需要同时定义TO和FROM报头才能匹配。如果两者中任一者不是有效匹配，则不使用12345拨号对等体。



注意：虽然我们将呼叫路由到From信头上，但离开网关的Invite仍然具有原始请求URI。我们仅使用Dial-peer provision-policy匹配出站拨号对等体，而不更改请求URI。

配置示例

<#root>

```
### Received INVITE

Received:
INVITE sip:8675309@172.18.110.58:5060 SIP/2.0
From: sipp <sip:sipp@172.18.110.65>;tag=1
To: sut <sip:cube@172.18.110.58:5060>



### Common Configurations


!
voice class uri FROM sip
 user-id sipp
!
voice class uri TO sip
 user-id cube
!


### OR Match


!
voice class dial-peer provision-policy 1
 description match from header. If false, try to header
 preference 1 from
 preference 2 to
!
dial-peer voice 1234 voip
 session protocol sipv2
 destination provision-policy 1
 incoming called-number .
!
```

```
dial-peer voice 11111 voip
 destination uri-from FROM
 session protocol sipv2
 session target ipv4:172.18.110.48
!
dial-peer voice 22222 voip
 destination uri-to TO
 session protocol sipv2
 session target ipv4:172.18.110.48
!


### AND Match


!
voice class dial-peer provision-policy 2
 description match from AND to headers
 preference 1 from to
!
dial-peer voice 1234 voip
 session protocol sipv2
 destination provision-policy 2
 incoming called-number .
!
dial-peer voice 12345 voip
 destination uri-from FROM
 destination uri-to TO
 session protocol sipv2
 session target ipv4:172.18.110.48
!
```

来源：通过Cisco IOS XE 17.5的《思科统一边界元素配置指南》

session target sip-uri

在Cisco IOS 15.4(1)T和Cisco IOS XE 3.11S之前，如果URI的主机部分不同，但用户相同，则需要两个单独的出站拨号对等体。

在此版本之后，管理员可以配置一个拨号对等体，以便为同一用户的多个主机提供服务。例如，testuser@example.com。和testuser@example.net。使用会话目标sip-uri触发传入邀请请求URI域的DNS解析，并动态确定会话目标IP。

配置示例:

网关收到包含这些报头的两个SIP邀请Invite sip:testuser@example.com.:5060 SIP/2.0 Invite sip:testuser@example.net:5060 SIP/2.0网关匹配testuser@example.com的传入SIP请求。和testuser@example.net，因为incoming URI命令和user-id定义都匹配testuser。voice-class sip call-route url is present命令意味着您可以基于此入站邀请的请求URI评估出站拨号对等体。由于与拨号

对等体1（testuser的用户ID）匹配的原因相同，因此您匹配拨号对等体2。此拨号对等体的会话目标是会话目标sip-uri定义的原始sip-uri，即FQDN。进行DNS解析后，请更改example.com。将example.net添加到用于第3层路由的IP中，您将消息从网关发送出去。

```
!
ip host example.com 192.0.2.10
ip host example.net 192.0.2.10
!
voice class uri TEST-IN sip
 user-id testuser
!
dial-peer voice 1 voip
 description INCOMING dial-peer
 incoming uri request TEST
 session protocol sipv2
 voice-class sip call-route url
!
dial-peer voice 2 voip
 description OUTBOUND dial-peer
 destination uri TEST
 session protocol sipv2
 session target sip-uri
!
```

验证：

```
show voice class uri <uri-name>
show voice class dial-peer provision-policy <number>
debug voip uri
```

拨号对等体通配符

管理员在定义涉及数字字符串的入站和出站匹配机制时，可以使用拨号对等体通配符。其中包括destination-pattern、incoming called-number、e164-pattern-maps和answer-address以及prefix命令。拨号对等体通配符是可用于配置的正则表达式(regex)，在匹配拨号对等体时具有更大的灵活性。

通配符表

字符	定义	Examples

*	在拨号对等体上，这是键盘上的文字值*（星型）。	12345 *
#	在拨号对等体上，数字键盘上的文字值为#（磅）。	8675309#
,	在数字之间插入1秒的暂停。也可以使用逗号在方括号[ ]中来分隔连续范围。	9,,,,55591[1-3,5-9]8675309
.	用于匹配任意值0-9、A-F和*、#、+的Regex字符 每个拨号对等体最多可以定义15个点字符，尽管CLI允许管理员根据需要配置尽可能多的点字符。 如果需要超过15个点，请使用T。	2.... 91[2-9]..[2-9].....
%	前一位出现零次或更多次的Regex。	
+	当在字符串的开头使用时，它表示在E164数字中使用的文字+。 当在字符串的其他地方使用时，它是发生一次或多次的前一个数字的正则表达式值。	+19191112222
?	上一个数字出现零次或一次的regex。	(206)?5015111 (0)?(1)?(1)?21933...
^	Regex字符，表示在括号外使用时字符串的开头 在方括号中使用时，该语句被视为exclude或DO NO MATCH语句 在后续版本中不再需要此项，因为网关在处理没有^的regex字符串时自动假定^。	^8675309 91[^135]555
$	Regex字符，表示字符串的结尾。	8675309$
\	转义字符表示文字值	
[ ]	方括号定义单个位置的字符范围。 必须使用逗号来分隔连续字符串。	[1-5]000 [2,5-8]000
()	括号定义一组字符。	9(258) 7777

T	最大为32位的可变长度匹配。 路由器在路由呼叫之前等待数字间超时。 数字间超时的默认值为10秒，可通过语音端口上的数字间超时进行修改。 管理员可以使用键盘上的#终止数字间超时。可通过全局配置的命令dial-peer terminator进行修改。 它还引用T302计时器。	9011T
-	在方括号中用于定义范围。	[5-9]1234

来自网关的输出，显示可能的正则表达式输入。

```
Gateway(config-dial-peer)# destination-pattern asdfqw4r3~2
Incorrect format for E.164 Number
        regular expression must be of the form  ^[][^0-9,A-F#*.?+%()-]*T?(\$)?$
```

拨号对等体状态

拨号对等体可以处于两种运行状态之一。

1. Up
2. 关闭

拨号对等体要处于有效运行状态，并且符合与呼叫路由配合使用的条件，它必须处于UP状态。对于出站VOIP拨号对等体，这意味着可以存在有效的出站匹配机制以及有效的会话目标来路由呼叫。对于出站POTS拨号对等体，可以配置有效的出站匹配机制以及有效的语音端口。仅使用入站拨号对等体，必须配置有效的入站匹配机制。

当拨号对等体配置了keepalive机制且远程目标未通过该keepalive机制的参数时，将显示忙出状态。然后，网关将拨号对等体移动到忙出状态，以便不再用于呼叫路由决策，并且当再次执行keepalive机制时，网关将拨号对等体重新置于启用状态。如果拨号对等体被选为出站拨号对等体，并且此拨号对等体处于忙出状态，则网关使用原因代码188使呼叫失败。

除操作状态外，还有管理状态。

1. Up
2. 关闭

管理员可以通过在拨号对等体上输入shutdown命令来禁用拨号对等体，而无需将其从配置中删除。要重新启用dial-per，请输入no shutdown。

注意：语音端口处于关闭、关闭或不运行的拨号对等体保持运行状态Up，但Out状态显示为Down。

确认

```
Gateway# show dial-peer voice summary
dial-peer hunt 0
             AD                                      PRE PASS                 OUT
TAG    TYPE  MIN  OPER PREFIX    DEST-PATTERN       FER THRU SESS-TARGET     STAT PORT    KEEPALIVE
1      voip  up   up                                  0  syst
777    voip  up   up              9...                0  syst ipv4:10.50.244.2
555    voip  up   down            555                 0  syst
888    pots  up   up              888                 0                       up   0/2/0
999    pots  up   up              999                 0                       down 0/2/0
123    voip  up   up              123                 0  syst ipv4:10.10.10.10            busyout
```

虚拟路由和转发(VRF)和拨号对等体寻线

与VRF匹配的入站拨号对等体

从Cisco IOS 15.6(3)M和Cisco IOS-XE 16.3.1开始，Cisco网关可以使用VRF ID匹配入站拨号对等体。要利用这一点，管理员必须将入站拨号对等体绑定到接口，而接口又将拨号对等体绑定到指定接口上的VRF ID。绑定完成后，入站呼叫由思科网关过滤，仅包括符合条件的入站拨号对等体，这些拨号对等体与接收数据包的接口的VRF ID匹配。在此处，入站拨号对等体根据常规拨号对等体匹配操作顺序进行匹配。

在这些思科IOS/思科IOS-XE版本之前，思科网关会根据常规的入站拨号对等体匹配做出入站选择，而无需任何过滤。这意味着VRF1呼叫可以与VRF2拨号对等体匹配。此外，由于在这些版本之前，H323和SIP仅支持一个VRF，因此尝试使用多VRF功能时会出现其他问题。将单个VRF用于语音应用称为VRF感知配置。

完整的VRF感知文档：安全和VPN配置指南

完整多VRF文档：Cisco Unified Border Element配置指南 — Cisco IOS XE 17.6及以上版本

出站拨号对等体与VRF匹配

思科网关能够桥接跨VRF的呼叫，而无需配置路由泄漏。这意味着，如果满足正常的出站拨号对等体匹配选择，则可以在VRF2的拨号对等体上将VRF1上的入站呼叫路由到出站。拨号对等体组可用于强制思科网关将呼叫保留在同一VRF内。

VRF和拨号对等体组配置示例

此配置示例中的VRF1和VRF2具有两个重叠的IP范围和两个重叠的电话号码范围。

使用VRF绑定确保匹配的入站拨号对等体正确，使用Dial-peer Groups确保匹配的出站拨号对等体正确VRF绑定。如果8675309呼叫的SIP数据包到达gig0/0/1.2，则网关会根据VRF2 ID过滤掉所有可用的入站拨号对等体。这意味着您无法匹配拨号对等体10。现在当您检查数字字符串时，您可以匹配拨号对等体20。拨号对等体20有一个拨号对等体组，它告诉网关唯一可以匹配的出站拨号对等体也是拨号对等体20。此拨号对等体组允许您避免匹配拨号对等体10并将来自VRF1的呼叫传递到VRF2。从那里，呼叫可以正常进行。

```
!
interface GigabitEthernet0/0/1.1
 description VRF1
 encapsulation dot1Q 10
 ip vrf forwarding VRF1
 ip address 10.10.10.10 255.255.255.0
!
interface GigabitEthernet0/0/1.2
 description VRF2
 encapsulation dot1Q 20
 ip vrf forwarding VRF2
 ip address 10.10.10.10 255.255.255.0
!
voice service voip
 no ip address trusted authenticate
 media-address voice-vrf VRF1
 media-address voice-vrf VRF2
 allow-connections sip to sip
 sip
!
voice class dpg 10
 description INBOUND VRF1 to OUTBOUND VRF1
 dial-peer 10 preference 1
!
voice class dpg 20
 description INBOUND VRF2 to OUTBOUND VRF2
 dial-peer 20 preference 1
!
```

```
dial-peer voice 10 voip
 description VRF1
 destination-pattern 8675309
 session protocol sipv2
 session target ipv4:10.10.10.20
 destination dpg 10
 incoming called-number 8675309
 voice-class sip bind control source-interface GigabitEthernet0/0/1.1
 voice-class sip bind media source-interface GigabitEthernet0/0/1.1
!
dial-peer voice 20 voip
 description VRF2
 destination-pattern 8675309
 session protocol sipv2
 session target ipv4:10.10.10.20
 destination dpg 20
 incoming called-number 8675309
 voice-class sip bind control source-interface GigabitEthernet0/0/1.2
 voice-class sip bind media source-interface GigabitEthernet0/0/1.2
!
```

确认

```
Gateway# show vrf brief | i VRF
  VRF1                             1:1                  ipv4        Gi0/0/1.1
  VRF2                             2:2                  ipv4        Gi0/0/1.2
```

```
Gateway# show dial-peer voice summary
TAG    TYPE  MIN  OPER PREFIX    DEST-PATTERN      FER THRU SESS-TARGET    STAT PORT    KEEPALIVE    VR
10     voip  up   up             8675309            0  syst ipv4:10.10.10.20                        VR
20     voip  up   up             8675309            0  syst ipv4:10.10.10.20                        VR
```

```
Gateway# show voice class dpg 10
Voice class dpg: 10      AdminStatus: Up
 Description: INBOUND to OUTBOUND VRF1
 Total dial-peer entries: 1

 Peer Tag       Pref
 --------       ----
 10             1
-------------------------------------
```

```
Gateway# show voice class dpg 20
Voice class dpg: 20      AdminStatus: Up
 Description: INBOUND to OUTBOUND VRF2
 Total dial-peer entries: 1

 Peer Tag       Pref
```

```
--------        ----
20              1
-------------------------------------
```

高级呼叫路由技术

随着业务需求的增长，公司不断扩展，需要更多的DID，企业管理员会发现基本拨号对等体无法很好地满足扩展需求。可能需要解决开机和关机问题，或者一般而言拨号对等体数量过多。拥有数千个拨号对等体并不能简化管理和故障排除。为每个特定CUCM服务器或呼叫代理设置拨号对等体开始使拨号对等体过多的问题复杂化，因为现在管理员需要为每个数字字符串配置拨号对等体。如果有多个SIP提供商连接到网关，或者有几个不同的人员使用同一个CUBE，则隔离特定租户非常困难。

思科已接受此反馈并创建了一系列可以解决这些问题和其他问题的项目。拨号对等体组、语音类租户、目标服务器组、e164-pattern-maps和POTS中继组使管理员能够解决列出的所有问题以及更多未列出的问题。

拨号对等体组

在Cisco IOS 15.4(1)T中添加了拨号对等体组，在Cisco IOS 15.5(1)T和Cisco IOS-XE 3.14S中添加了Cisco IOS-XE 3.11S和POTS拨号对等体作为选项。拨号对等体组允许管理员根据匹配的入站拨号对等体为出站路由指定准确的拨号对等体。匹配配置了拨号对等体组的入站拨号对等体后，即使目标模式不匹配，呼叫也会使用拨号对等体组中定义的拨号对等体。唯一的前提条件是出站拨号对等体必须处于Up状态，因此必须配置出站匹配方法，但实际上并未将其用于路由呼叫。

描述拨号对等体组的最佳方法是将其与路由表中的静态路由概念相类似。这些是静态的入站到出站路由决策，会消除网关的一些猜测，因为它们告诉网关具体如何路由呼叫。

完整文档：[思科统一边界要素配置指南 — Cisco IOS XE 17.6及以上版本]

配置示例

在本例中，被叫号码为8675309。这根据incoming called-number语句匹配拨号对等体1234。此拨号对等体配置有拨号对等体组，该组表示如果拨号对等体2失败，呼叫现在可以路由出拨号对等体2，然后路由出3，最后路由出1。这是网关，因此，现在尝试将呼叫路由出拨号对等体2，就像通过拨号对等体组明确告诉它一样，这是它可以执行的操作。



注意：拨号对等体1、2和3上的目标模式不是被叫号码8675309。这是正常的，呼叫仍然路由而不出现问题。

请记住，如“拨号对等体状态”部分所述，您需要将某项/任何内容配置为出站匹配语句。在这种情况下，目标模式只是使拨号对等体进入启用操作状态，并且不会计算该命令的数字字符串。建议配置类似目标模式AAAA的模式，因为这是有效的目标模式。由于这在技术上是一个有效的拨号对等体，因此其他呼叫可能会与之匹配。因此，AAAA数字字符串意味着，除涉及拨号对等体组的特定场景外，您永远不能将其用于任何其他场景，因为呼叫进入AAAA的可能性非常非常低。

```
!
dial-peer voice 1 voip
 description Server 1
 destination-pattern ^1234$
 session target ipv4:192.0.2.1
!
dial-peer voice 2 voip
 description Server 2
 destination-pattern ^5678$
 session target ipv4:192.0.2.2
!
dial-peer voice 3 voip
 description Server 3
 destination-pattern AAAA
 session target ipv4:192.0.2.3
!
voice class dpg 1
 description Dial-peer Group for specific called number 8675309
 dial-peer 2 preference 1
 dial-peer 3 preference 2
 dial-peer 1 preference 3
!
dial-peer voice 1234 voip
 description INCOMING dial-peer with DPG
 incoming called-number ^8675309$
 destination dpg 1
!
```

确认

```
Gateway# show voice class dpg 1
Voice class dpg: 1       AdminStatus: Up
 Description: Dial-peer Group for specific called number 1234
 Total dial-peer entries: 3

 Peer Tag       Pref
 --------       ----
 2              1
 3              2
 1              3
-------------------------------------
```

E164 — 模式映射

此功能使管理员能够将许多可能的号码匹配（目标模式、传入呼叫号码等）组合到一个模式映射中，从而减少拨号对等体总数。在Cisco IOS 15.2(4)M和Cisco IOS-XE 3.7S中添加了出站拨号对等体e164-pattern-map支持，而在Cisco IOS 15.4(1)T和Cisco IOS-XE 3.11S中添加了入站拨号对等体e164-pattern-map支持。

e164-pattern-map可以通过CLI配置或预配置并另存为.cfg文件。然后将.cfg文件添加到网关的闪存中，然后在配置命令的其余部分时引用该文件。.cfg文件可利用5000个条目。

两种配置方法中的条目都可以使用所有普通拨号对等体通配符进行进一步汇聚！

注意： 使用calling e164-pattern-map配置传入和传出拨号对等体时，必须将拨号对等体调配策略应用于传入拨号对等体，以避免呼叫失败。

完整文档：[思科统一边界要素配置指南 — Cisco IOS XE 17.6及以上版本]

CLI配置示例 — 呼叫号码

```
!
voice class e164-pattern-map 1
 description E164 Pattern Map for calling numbers
 e164 919574100.
 e164 919574300.
 e164 8675309
!
dial-peer voice 1 voip
 description INBOUND Dial-peer based on CALLING #
 incoming calling e164-pattern-map 1
!
dial-peer voice 11 voip
 description OUTBOUND Dial-peer based on CALLING #
 destination calling e164-pattern-map 1
!
```

CLI配置示例 — 被叫号码

```
!
voice class e164-pattern-map 2
 description E164 Pattern Map for called 800 numbers
 e164 91800T
 e164 91855T
```

```
 e164 91888T
!
dial-peer voice 2 voip
 description INBOUND Dial-peer based on CALLED #
 incoming called e164-pattern-map 2
!
dial-peer voice 22 voip
 description OUTBOUND Dial-peer based on CALLED #
 destination e164-pattern-map 2
!
```

闪存配置示例

```
!
voice class e164-pattern-map <tag>
 description FILEPATH for E164 Pattern Map
 url flash:<filepath>/e164-pattern-list.cfg
!
dial-peer voice ### voip
 description E164 Pattern Map Dial-peer
 incoming calling e164-pattern-map <tag>
!

voice class e164-pattern-map load
```

确认

```
Gateway# show voice class e164-pattern-map 1

e164-pattern-map 1
----------------------------------------
  Description: CUCM phones
  It has 3 entries
  It is not populated from a file.
  Map is valid.

E164 pattern
-------------------
8675309
1...
[2-5]...$
```

值得注意的缺陷

Cisco Bug ID CSCva64393 e164-pattern-map不解析配置文件的最后一行。

目标服务器组

服务器组使管理员能够在同一VOIP拨号对等体上配置多个目标（会话目标）。默认情况下，排序顺序是在服务器组条目中定义的首选项。当您使用命令hunt-scheme round-robin时，可以采用轮询寻线。服务器组已添加到Cisco IOS 15.4(1)T和Cisco IOS XE 3.11S。在Cisco IOS XE 17.4.1中，语音类服务器组配置中添加了可配置的huntstop错误代码。也就是说，您可以配置单个错误代码，例如404 Not Found，并且SIP错误通常会触发设备尝试服务器组中的下一个选项。在server-group内使用config huntstop 1 resp-code 404;狩猎可以停止。也可以针对以下范围进行配置：huntstop 1 resp-code 401到599。

注意：每个服务器组的最大条目数为5。

完整文档：思科统一边界要素配置指南 — Cisco IOS XE 17.6及以上版本

配置示例 — 普通

```
!
voice class server-group 1
 hunt-scheme round-robin
 ipv4 10.50.244.2 port 5060 preference 1
 ipv4 10.50.244.62
 ipv6 2010:AB8:0:2::1 port 2323 preference 3
 ipv6 2010:AB8:0:2::2 port 2222
!
dial-peer voice 1 voip
 session protocol sipv2
 destination-pattern 8675309
 session server-group 1
!
```

确认

```
Gateway# show voice class server-group 1
Voice class server-group: 1
 AdminStatus: Up OperStatus: Up
 Hunt-Scheme: round-robin Last returned server:
 Description:
 Total server entries: 4
```

```
 Pref Type IP Address IP Port
 ---- ---- ---------- -------
 1 ipv4 10.50.244.2     5060
 0 ipv4 10.50.244.62
 3 ipv6 2010:AB8:0:2::1 2323
 0 ipv6 2010:AB8:0:2::2 2222
[..truncated..]
```

目标服务器组和选项Keepalive

请注意，服务器组不遵循正常的对话外选项Keepalive机制。它们使用称为option-keepalive profile的功能。这样，网关就可以监控在特定server-group中定义的每个呼叫代理。

服务器组的Option-keepalive示例

```
!
voice class server-group 1
 hunt-scheme round-robin
 ipv4 10.50.244.2
 ipv4 10.50.244.62
!
dial-peer voice 1 voip
 session protocol sipv2
 session server-group 1
 voice-class sip options-keepalive profile 1
!
```

确认

<#root>

```
Gateway#
```

show voice class sip-options-keepalive 1

```
Voice class sip-options-keepalive: 1             AdminStat: Up
 Description:
 Transport: system               Sip Profiles: 0
 Interval(seconds) Up: 5                 Down: 5
 Retry: 5

  Peer Tag      Server Group    OOD SessID      OOD Stat        IfIndex
  --------      ------------    ----------      --------        -------
  1             1                               Active          87

  Server Group: 1                OOD Stat: Active
   OOD SessID   OOD Stat
   ----------   --------
   1            Active
```

```
   2            Active

 OOD SessID: 1                   OOD Stat: Active
  Target: ipv4:10.50.244.2
  Transport: system              Sip Profiles: 0

 OOD SessID: 2                   OOD Stat: Active
  Target: ipv4:10.50.244.62
  Transport: system              Sip Profiles: 0
```

呼出代理

SIP出站代理配置可以添加到语音服务voip、语音类租户或拨号对等体配置，以指定第3层SIP数据包的目标。

即，拨号对等体上的会话目标可用于创建SIP数据包，但出站代理可能是在第3层发送数据包的位置。

```
!
voice service voip
 sip
  outbound-proxy dns:la01.sipconnect-us10.cisco-bcld.com
!
voice class tenant 100
 outbound-proxy dns:la01.sipconnect-us10.cisco-bcld.com
!
dial-peer voice 100 voip
 session target ipv4:192.168.1.1
 voice-class sip outbound-proxy dns:la01.sipconnect-us10.cisco-bcld.com
!
```

请注意，拨号对等体的默认配置是voice-class sip outbound-proxy system，这可能使拨号对等体使用全局语音服务voip > sip配置。

可以禁用此行为，并强制拨号对等体回退并使用会话目标作为每个拨号对等体的第3层目标，此配置如下：

```
dial-peer voice 777 voip
 no voice-class sip outbound-proxy
```

POTS中继组

中继组是具有类似信令功能的物理语音端口的集合。此功能可用于减少需要配置的POTS拨号对等体的总数。中继组在12.1(3)T中引入到Cisco IOS中，并且存在于所有Cisco IOS XE版本中。

完整文档：网关中继和基于运营商的路由增强功能

配置示例

```
!
trunk group PSTN
 description PSTN voice-ports
!
trunk group FXO
 description FXO voice-ports
!
voice-port 0/2/0
 trunk-group PSTN 1
!
voice-port 0/2/1
 trunk-group PSTN 2
!
voice-port 0/2/2
 trunk-group FXO 1
!
voice-port 0/2/3
 trunk-group FXO 2
!
dial-peer voice 1234 pots
 trunkgroup PSTN 1
 trunkgroup FXO 2
!
```

语音类租户

Cisco IOS 15.6(2)T和Cisco IOS XE 16.3.1引入了语音类租户，允许每个租户拥有自己的配置。租户可以是电话提供商、Cisco Unified Communication Manager(CUCM)或管理员希望具有的特定全局设置的任何其他第三方呼叫代理。首先，管理员创建语音类租户并定义参数。然后，语音类租户将应用于特定拨号对等体或选择。此新配置使管理员对除拨号对等体和全局配置之外的呼叫拥有更高级别的控制权。

对于17.8.1a，可以使用sip-listen命令（与相应的SIP控制绑定命令相结合）配置语音类租户配置，以定义该租户的非安全或安全端口。这意味着租户1可以在UDP 5060 + VRF红色上侦听不安全的SIP，而租户2在TCP TLS 5070 + VRF蓝色上侦听SIP。根据侦听端口+绑定+可选vrf入站拨号对等体匹配租户后，将过滤为应用了租户的租户。

完整文档：思科统一边界要素配置指南 — Cisco IOS XE 17.6及以上版本

不带租户的命令首选项的正常顺序

1. 拨号对等体命令
2. 全局命令（语音服务voip和sip-ua）

租户的命令首选项顺序

1. Dial-peer命令
2. 租户命令
3. 全局命令（语音服务voip和sip-ua）

多租户配置示例

您有两个租户777和999。您已对它们配置略有不同的配置，并将它们应用于拨号对等体。这意味着使用不同拨号对等体的呼叫具有基于拨号对等体的配置以及租户特定配置。列出的选项只是语音类租户强大功能的片段。请参阅文档，了解可以在租户上配置哪些内容。建议使用严格的匹配机制，例如语音类URI或具有特定号码字符串的标记号码，以分隔租户拨号对等体匹配，甚至配置VRF，使租户A不会与租户B重叠，意外匹配它们无法匹配的拨号对等体。

```
!
voice class tenant 999
 asymmetric payload full
 bind control source-interface GigabitEthernet0/0/0.228
 bind media source-interface GigabitEthernet0/0/0.228
 g729 annexb-all
!
voice class tenant 777
 sip-server ipv4:192.168.1.2
 bind control source-interface Loopback0
 bind media source-interface Loopback0
 pass-thru content sdp
!
dial-peer voice 999 voip
 destination-pattern 8675309
 session protocol sipv2
 incoming called-number 8675309
 voice-class sip tenant 999
!
dial-peer voice 777 voip
 destination-pattern 8675309
 session protocol sipv2
 session target sip-server
 voice-class sip tenant 777
!
```

确认

目前，没有单独的命令可查看语音类租户配置。此命令足以将运行配置过滤为仅包含租户信息。

```
show run | sec tenant
```



注意：Cisco Bug ID CSCvf28730是show sip-ua register status不反映语音类租户上SIP中继注册状态的位置。

ILS URI呼叫CUBE（语音类路由字符串）

路由字符串与CUCM集群间查找服务(ILS)一起使用，可以配置为允许思科网关通过从运行ILS服务的CUCM 9.5+收到的SIP Invite中包含的路由字符串来路由VoIP呼叫。此功能已添加到Cisco IOS 15.3(3)M和Cisco IOS XE 3.10S。大多数ILS连接都是从CUCM到CUCM的，管理员无需费心使用CUBE进行集群间中继。但是，如果您需要执行中间有CUBE的函数，则有相应的选项。CUCM需要在SIP配置文件上启用应用于SIP中继的设置Send ILS Learned Destination Route String，以便将x-cisco-dest-route-string报头发送到CUBE

完整文档：H.323-to-SIP和SIP-to-SIP的企业应用互操作性配置指南，Cisco IOS版本15M&T

配置示例CUCM - SIP - CUBE - SIP - CUCM

```
!
voice service voip
 sip
  call-route dest-route-string
!
voice class route-string rt1
 pattern example.com.
!
voice class sip route-string rt2
 pattern *.eu
!
voice class sip-hdr-passthrulist hdr1
 passthru-hdr x-cisco-dest-route-string
!
dial-peer voice 1 voip
 description INBOUND dial-peer
 session protocol sipv2
 voice-class sip pass-thru headers hdr1
 incoming called-number .
!
dial-peer voice 2 voip
```

```
 description OUTBOUND dial-peer
 destination route-string rt2
 session protocol sipv2
 session target ipv4:172.16.104.178
!
```

确认

```
show voice class route-string
```

传统呼叫路由技术

本部分涵盖的项目被视为传统技术。虽然在Cisco网关中仍能配置这些命令，但不建议在现代配置中使用这些命令。本文档仅介绍它们，因为在使用传统配置或执行升级时可能会遇到它们。

DNIS映射

DNIS映射可以被视为现在的E164模式映射的前身。DNIS映射是在12.2(2)XB中添加到Cisco IOS的，并且一直存在于Cisco IOS XE中。

如果配置了DNIS映射，则值得将它们转换为更强大的e164模式映射功能。

命令语法：Cisco IOS语音命令参考 — D至I

配置示例

```
!
voice dnis-map 34
 dnis 8675309
!
dial-peer voice 88 voip
 dnis-map 34
!
```

中继组标签

中继组标签添加在Cisco IOS 12.2(11)T中，并且存在于所有Cisco IOS XE版本中。trunk-group-label的用途与运营商ID类似，因为它可用于增强拨号对等体的匹配。这可用于POTS中继组、VOIP和POTS拨号对等体以及语音源组中的配置。在现代Cisco网关配置中，中继组标签的使用很少见。

命令语法：Cisco IOS语音命令参考 — T至Z

配置示例

```
!
dial-peer voice 112 pots
 trunk-group-label source north3
 trunk-group-label target east17
!
```

编号类型

通过ISDN Q.931集成，可以根据Q.931 SETUP消息中的主叫或被叫号码以及特定ITU号码类型来匹配拨号对等体。这可以通过VOIP或POTS拨号对等体上的numbering-type命令进行配置。编号类型不能单独使用，并且必须与目标模式、应答地址或传入被叫号码一起使用。这意味着入站/出站匹配语句的条件以及号码类型必须匹配才能使拨号对等体成功用于入站和出站呼叫路由。

Numbering-match可视为拨号对等体过滤机制，而不是匹配机制。这是因为如果不应用管理员首选项，则无论是否应用编号类型命令，拨号对等体都被视为相同的默认首选项权重。这与运营商ID不同，当运营商将ID应用于拨号对等体时，如果两个条件都成立，则向拨号对等体添加首选项而不是其他匹配机制。

编号类型匹配是在Cisco IOS 12.0(7)XR1中添加的，并且存在于所有Cisco IOS XE版本中。随着协作网络中部署的传统POTS ISDN线路的减少，编号类型的使用在现代部署中很少见。

命令语法：Cisco IOS语音命令参考 — K至R

配置示例

只有当ISDN号码类型为National时，此dial-peer才能匹配4085150000至4085159999。

```
!
dial-peer voice 408 voip
 numbering-type national
 destination-pattern 408515....
 session target ipv4:10.1.1.2
!
```

可能的号码类型：

缩写	此网络支持的完整号码的缩写表示
国际	呼叫到另一个国家/地区用户的号码
国内	呼叫的号码，用于接通同一国家/地区但不在本地网络的用户
网络	特定于服务网络的管理或服务编号
预留	保留给分机
用户	到达同一本地网络中用户的被叫号码
未知	号码类型为网络未知

拨号对等体数据

Cisco IOS 12.2(13)T引入了数据拨号对等体，这种拨号对等体的使用用于Cisco网关上的传入数据调制解调器呼叫。此拨号对等体仅用于入站方向，在现代部署中很少见。

命令语法：Cisco IOS语音命令参考 — D至I

配置示例

```
!
dial-peer data 100 pots
 incoming called-number 100
!
```

Voice Source-Group

此功能在15.1(2)T中添加，但是在许多现代部署中并未实施。通常部署Cisco IOS/CUBE的其他安全方法。

CUBE应用安全概述可从本白皮书的第4.2节开始。

思科统一边界元素(CUBE)管理和可管理性规范

命令语法:Voice Source-Group功能

拨号对等体权限

此配置允许管理员将拨号对等体限制为仅允许入站连接(term / terminate)或出口连接(orig / originate)。 这类似于将入站拨号对等体显式配置为仅用于入站呼叫，而将出站拨号对等体配置为出站呼叫。任何拨号对等体的默认设置为允许入站和出站连接。此CLI在现代部署中不常部署。

```
Router(config)# dial-peer voice 1 voip
Router(config-dial-peer)# permission ?
 both allow both orig/term on this dialpeer
 none no orig/term allowed on this dialpeer
 orig allow only orig on this dialpeer
 term allow only term on this dialpeer
```

URI和数字处理

在协作部署中的某些时刻，管理员可能需要处理数字或URI/SIP报头。Cisco网关有多种数字处理的方法，使管理员能够完全控制数字处理的方式和时间。但是，这并不总是容易实现，一些人被不同的选项弄得不知所措，或者管理员不知道有哪个选项存在。

通过POTS拨号对等体进行数字处理

POTS拨号对等体具有一些独特的数字处理技术，而VOIP拨号对等体不具备这些技术。

第一个是删除目标模式中显式定义的左对齐数字。可以在POTS拨号对等体上使用no digit-strip命令禁用此功能。

示例：

在本示例中，9011T被定义为目标模式的字符串。

通过此设置，您可以接收90113227045555的呼叫。这与出站呼叫路由的拨号对等体匹配，并且在将呼叫从语音端口路由出去之前去除明确定义的数字9011。

```
!
dial-peer voice 1 pots
 destination-pattern 9011T
 port 0/0/0:23
!
```

此示例显示没有设置数字条带的配置。

如果呼叫了相同的号码，则发送9011。

```
!
dial-peer voice 1 pots
 destination-pattern 9011T
 port 0/0/0:23
 no digit-strip
!
```

第二个功能是指定您想在POTS拨号对等体上转发多少位。

在本示例中，您收到来自CUCM的918005532447呼叫。在这种情况下，您需要删除9，但发送以1开头的剩余号码。

如果在POTS拨号对等体上配置forward-digits命令，则可以精确指定发送的数字数量。

```
!
dial-peer voice 1 pots
 destination-pattern 918005532447
 forward-digits 11
 port 0/2/0
!
```

最后，POTS拨号对等体可以使用prefix命令在路由出语音端口之前向呼叫添加数字。此示例将明确定义的91和前缀007去掉到号码，然后发出语音端口呼叫。

```
!
dial-peer voice 1 pots
 destination-pattern 91T
 prefix 007
 port 0/1/0:15
!
```

通过语音转换规则和配置文件进行数字处理

语音转换规则是用于转换数字的正则表达式(regex)。转换规则和配置文件已添加到12.0(7)XR1中的Cisco IOS。转换规则应用于语音转换配置文件，然后应用于拨号对等体或语音端口。转换规则包含匹配输入和修改输出。与号码上的匹配输入一起为ISDN计划和类型提供了匹配和修改输入。匹配号码字符串、计划和类型的组合被视为匹配项。这意味着所定义的所有匹配输入都必须为true才能进行转换。

转换规则能够在ISDN、SIP和H323信令协议中更改被叫号码、主叫号码、重定向被叫号码、重定向目标号码和回拨号码。转换规则基于自上而下的搜索进行匹配，因此规则的顺序至关重要。如果在更高级的规则中找到匹配项，网关将立即停止搜索并处理转换。转换规则无法更改非数字SIP报头，例如testuser@10.10.10.10。若要进行此处理，请使用SIP配置文件。

过渡规则可用于阻止思科网关上的呼叫。

转换配置文件选择首选项

1. 语音端口上的传入语音转换配置文件
2. 应用于串行接口的中继组上的传入语音转换配置文件
3. 入站拨号对等体上的传入语音转换配置文件
4. 通过语音服务pots定义的传入语音转换配置文件
5. 通过全局“voip-incoming translation-profile”定义的传入语音转换配置文件
6. 通过语音服务pots定义的outbound voice translation-profile
7. 出站拨号对等体上的outbound voice translation-profile或translate-outgoing
8. 应用于串行接口的中继组上的出站语音转换配置文件
9. voice-port上的出站voice translation-profile

除拨号对等体regex和wilcards转换规则外，转换规则还拥有自己的regex字符。

字符	定义

*	在转换规则中使用时，这是对前面字符的0个或更多字符的regex。 要匹配文本*，请使用转义字符：*
\	常用于转义规则中的转义集\(\)
&	和号用于为修改集提供初始匹配集中的任何匹配项
()	用括号括起来的项被视为集合。
^	定义字符串的显式开头。 与拨号对等体转换规则不同，转换规则不定义字符串的开头。 这意味着定义没有^的字符串可能会匹配输入字符串中的任何内容，从而导致数字中间出现不需要的转换。

修改集

- 集合指定为\0、\1、\2等。
- \0匹配第一个匹配集之间的任何内容。这也可通过“和”字符来实现：&.
- \1匹配match-set中的第一组()
- \2与match-set中的第二组()匹配
- 如此等等。

具有两个集合的转换规则示例

在本例中，您可以检查数字000111000222。

您要从数字中删除0并实现最终的111222数。

为此，您可以配置设置1和设置2，分别抓取111和222，同时丢弃0。

```
!
voice translation-rule 333
 rule 1 /000\(111\)000\(222\)/ /\1\2/
!
voice translation-profile SET-EXAMPLE
 translate called 333
!

Gateway# test voice translation-rule 333 000111000222
Matched with rule 1
```

```
Original number: 000111000222   Translated number: 111222
Original number type: none      Translated number type: none
Original number plan: none      Translated number plan: none
```

从被叫号码中删除9外拨模式的示例

```
!
voice translation-rule 9
 rule 1 /^9\(.*\)/ /\1/
!
voice translation-profile STRIP-9
 translate called 9
!
dial-peer voice 9 voip
 translation-profile outgoing STRIP-9
!
voice-port 0/0/0
 translation-profile outgoing STRIP-9
!

Gateway# test voice translation-rule 9 918675309
Matched with rule 1
Original number: 918675309      Translated number: 18675309
Original number type: none      Translated number type: none
Original number plan: none      Translated number plan: none
```

将被叫号码截断为4位

```
!
voice translation-rule 4
 rule 1 /.*\(....\)/ /\1/
!
voice translation-profile STRIP-TO-4
 translate called 4
!

Gateway# test voice translation-rule 4 8675309
Matched with rule 1
Original number: 8675309        Translated number: 5309
Original number type: none      Translated number type: none
Original number plan: none      Translated number plan: none
```

从被叫号码删除加号+

```
!
voice translation-rule 999
 rule 1 /\+\(.*\)/ /\1/
!
voice translation-profile STRIP-PLUS
```

```
 translate called 999
!
Gateway# test voice translation-rule 999 +8675309
Matched with rule 1
Original number: +8675309       Translated number: 8675309
Original number type: none      Translated number type: none
Original number plan: none      Translated number plan: none
```

转换规则也可以直接应用于拨号对等体，无需首先应用于转换配置文件。

```
!
voice translation-rule 1
 rule 1 /1234/ /8678309/
!
voice translation-rule 2
 rule 2 /^4...$/ /1408515\0/
!
dial-peer voice 1 voip
 translate-outgoing called 1
!
dial-peer voice 2 voip
 translate-outgoing calling 2
!
```

中继组上的转换配置文件

```
!
trunk group <name>
 translation-profile incoming <profile-name>
 translation-profile outgoing <profile-name>
!
```

调试语音转换规则和配置文件

```
debug voip ccapi inout
debug voice translation
debug dialpeer
test voice translation-rule <number> <string> type <type> plan <plan>
```

语音类e164 — 转换

语音类e164转换功能是较新的Cisco IOS XE功能，允许管理员创建匹配语句列表并修改要通过配置文件从闪存或网络目录加载的语句。这与本文档中讨论的e164-pattern-map功能的概念相似。这样

，管理员可以在一个配置文件中配置最多100个转换，并在单个转换配置文件中应用这些转换。有关详细信息，请参阅Cisco IOS语音命令参考

对于.cfg文件，请遵循以下语法：

```
pattern1_to_be_matched<tab>replaced_pattern<space><enter>
pattern1_to_be_matched<tab>replaced_pattern<space><enter>
```

注意：后面的空格非常重要，如果没有额外的格式化步骤，导入可能会失败。

示例.cfg

```
+111111    8897
+222222    8312
928747    +123456789
737362    +987654321
```

然后，此文件按如下方式引用：

```
voice class e164-translation 164
 url ftp://username:password@10.10.10.10/sample.cfg
```

现在，您可以正常应用到转换配置文件，然后使用正常的转换配置文件语法应用到拨号对等体。

```
voice translation-profile e164
 translate calling voice-class e164-translation 164
 translate called voice-class e164-translation 164
```

命令show voice class e164-translation e164-translation-number可用于查看转换配置文件的内容。

通过ISDN映射进行数字处理

ISDN MAPS是一种较旧的数字修改技术。这已添加到Cisco IOS 12.0(6)T中，大多数新配置未使用此功能，因为它们不如语音转换规则和配置文件功能强大。ISDN映射在串行接口下定义。

配置示例

```
Serial0/0/0:23
 isdn map address ^911 plan isdn type unknown
 isdn map address ^1.......... plan isdn type national
 isdn map address ^2......... plan isdn type national
 isdn map address ^3......... plan isdn type national
 isdn map address ^4......... plan isdn type national
 isdn map address ^5......... plan isdn type national
 isdn map address ^6......... plan isdn type national
 isdn map address ^7......... plan isdn type national
 isdn map address ^8......... plan isdn type national
 isdn map address ^9......... plan isdn type national
```

通过号码扩展(num-exp)进行数字处理

与ISDN映射类似，号码扩展是Cisco IOS 11.3(1)T中增加的一项较旧技术，在新网络中很少使用。此功能是在语音转换规则和配置文件存在之前添加的。号码扩展是应用于思科网关上所有拨号对等体的全局数字更改。在拨号对等体匹配之后，以及在将呼叫发送到下一个呼叫代理之前，将修改应用于被叫号码。

配置示例

```
num-exp 4... 18005554...
num-exp 1234 8675309
```

入站/出站SIP配置文件

SIP配置文件是功能强大的正则表达式(regex)Match语句，允许管理员更改包含SDP和SIP报头的SIP消息的任何方面。可以全局启用，按拨号对等体或按租户启用。从Cisco IOS 15.4(2)T和Cisco IOS XE 3.12S开始，SIP配置文件可用于入站修改。由于SIP配置文件非常强大，本文档仅介绍几个具体示例。SIP配置文件还增加了在Cisco IOS 15.5(2)T和Cisco IOS-XE 3.13S中修改或添加自定义SIP报头的功能。

有关入站SIP配置文件与出站SIP配置文件的要点

- 入站SIP配置文件在CUBE处理呼叫路由消息之前更改消息。
- 出站SIP配置文件在CUBE处理呼叫路由之后以及在消息发送到下一跳之前更改消息。

有关sip-profile配置的其他说明：

- SIP配置文件无法处理m=image SDP属性。此功能的增强功能在Cisco Bug ID CSCsr下列出20474 此外，SIP配置文件无法删除值或向SDP添加值。只有您可以修改这些值。但是，可以通过指定整个值，然后将输出设置为一组无空格的空引号来将SDP值修改为空值。
- 在voice class sip-profile中输入命令时，不会执行检查以查看当前输入的命令是否已存在或该命令的某个版本是否已经存在。如果管理员将线路7次粘贴到sip配置文件中，该线路将在运行配置中显示7次。建议删除正在修改的命令，然后在编辑sip-profiles时输入新命令，以避免出现多个命令。

完整文档：思科统一边界要素配置指南 — Cisco IOS XE 17.6及以上版本

CSA上的SIP配置文件测试工具：SPT工具

入站/出站SIP配置文件示例语法

```
!
voice class sip-profiles <number>
 request <message-type> sip-header <header> modify "match-pattern" "replace-pattern"
 request <message-type> sip-header <header> add "add-pattern"
 request <message-type> sip-header <header> remove

 request <message-type> sdp-header <header> modify "match-pattern" "replace-pattern"
 request <message-type> sdp-header <header> add "add-pattern"
 request <message-type> sdp-header <header> remove

 response <number> sip-header <header> modify "match-pattern" "replace-pattern"
 response <number> sip-header <header> add "add-pattern"
 response <number> sip-header <header> remove

 response <number> sdp-header <header> modify "match-pattern" "replace-pattern"
 response <number> sdp-header <header> add "add-pattern"
 response <number> sdp-header <header> remove
!
```

带编号的入站/出站SIP配置文件示例

```
voice class sip-profiles 200
 rule 1 response ANY sip-header Remote-Party-ID modify "match-pattern" "replace-pattern"
 rule 2 response ANY sdp-header Audio-Attribute modify "match-pattern" "replace-pattern"
```

出站SIP配置文件应用方法

<#root>

```
! Global Application

voice service voip
 sip
  sip-profiles <number>
!

! Tenant Application


voice class tenant <tag>
 sip-profiles <tag>
!

! Dial-peer Application


dial-peer voice <tag> voip
 voice-class sip profiles <number>
!
```

入站SIP配置文件应用方法



注意：无论使用的是全局应用、租户还是拨号对等体应用，都需要启用语音服务voip sip下的sip-profile inbound。

<#root>

```
! Global Application

voice service voip
 sip
  sip-profiles inbound
  sip-profiles <number> inbound
!
```

<#root>

```
! Tenant Application


voice service voip
 sip
  sip-profiles inbound
!
voice class tenant <tag>
 sip-profiles <tag> inbound
```

```
!
```

<#root>

```
! Dial-Peer Application


voice service voip
 sip
  sip-profiles inbound
!
dial-peer voice <tag> voip
 voice-class sip profiles <number> inbound
!
```

修改OPTIONS keepalive消息的SIP配置文件示例。

```
!
voice class sip-options-keepalive 200
 transport tcp tls
 sip-profiles 299
!
```

修改主机、域或URI两个部分的SIP配置文件示例。

<#root>

```
! Host


!
voice class sip-profiles 1
 request ANY sip-header Contact modify "sip:(.*)@" "sip:8675309@"
!

! Domain


!
voice class sip-profiles 2
 request ANY sip-header Contact modify "10.67.138.241:5060" "cisco.com"
!

! Note: Port is optional


!

! Modify Both User and Host
```

```
!
voice class sip-profiles 3
 request ANY sip-header Contact modify "sip:(.*)>" "sip:8675309@cisco.com>"
!
```

添加、修改或删除转移报头的SIP配置文件示例。

```
<#root>

! Add


!
voice class sip-profiles 777
 request INVITE sip-header Diversion add "Diversion: <sip:1234@cisco.com>"
!
!

! Modify


!
voice class sip-profiles 888
 request INVITE sip-header Diversion modify "sip:(.*)>" "sip:1234@cisco.com>"
!
!

! Remove


!
voice class sip-profiles 999
 request INVITE sip-header Diversion remove
!
```

修改SIP报头呼叫方ID名称部分的示例SIP配置文件。

```
!
voice class sip-profiles 123
 request INVITE sip-header From modify "\".*\"" "\"TEST CLID*\""
!
```

将183会话更改为180振铃的SIP配置文件示例。

```
!
voice class sip-profiles 789
 response 183 sip-header SIP-StatusLine modify "SIP/2.0 183 Session in Progress" "SIP/2.0 180 Ringing"
!
```

用于与提供商进行单向或单向音频互操作性的SIP配置文件示例。

<#root>

```
!
voice class sip-profiles 200
 request ANY sdp-header Audio-Attribute modify "a=inactive" "a=sendrecv"
 request ANY sdp-header Audio-Connection-Info modify "0.0.0.0" "10.10.10.10"
!

! where 10.10.10.10 is CUBE's provider facing IP address
```

用于删除互操作性问题的UPDATE方法的SIP配置文件示例。

```
!
voice class sip-profiles 200
 request ANY sip-header Allow-Header modify ", UPDATE" ""
!
```

显示SIP配置文件中的SET use的SIP配置文件示例。这与语音转换规则部分中描述的集合概念相同。

```
!
voice class sip-profiles 1
 request ANY sip-header Contact modify "sip:(.*)@" "sip:\1@"
!
```

配置IF逻辑和换行符与SIP配置文件断开。

SIP配置文件支持换行符，但是，这些配置文件只有一个非常具体的使用案例。由于SIP配置文件不具有任何If、Then、Else逻辑，因此现在提供了一种根据来自另一个报头的输入对一个报头进行修改的方法。例如，如果FROM报头包含1234@cisco.com，管理员只想修改转移报头。利用换行符，我们可以伪装SIP配置文件中的IF语句。请参阅配置示例：您匹配From信头中任何域的1234。然后，将第一组换成并添加新的换行符\x0D\x0AD。最后，添加所需的报头。请注意，此方法仅允许ADD报头。无法修改另一个报头。因此，这仅部分满足管理员先前希望达到的要求。

```
!
voice class sip-profiles 1
 request INVITE sip-header From modify “(.*sip:1234@.*)” “\1\x0D\x0ADiversion: <sip:5678@example.com.>”
!
```

具有OR逻辑的SIP配置文件示例。

```
!
voice class sip-profiles 123
 request ANY sdp-header Audio-Attribute modify "(a=sendonly|a=recvonly|a=inactive)" "a=sendrecv"
 response ANY sdp-header Audio-Attribute modify "(a=sendonly|a=recvonly|a=inactive)" "a=sendrecv"
!
```

通过SIP配置文件进行第7层SIP检测的示例。

<#root>

```
### Usage

10.21.15.10 replace with private IP of CUBE
a.b.c.d replace with public IP
-----------------------------------------------------


### Inbound from ITSP Layer 7 Fixup

!
voice class sip-profiles 888
 request INVITE sip-header SIP-Req-URI modify "@.*;" "@10.21.15.100;"
!
voice service voip
 sip
  sip-profiles inbound
!


### Outbound Layer 7 Fixup

!
voice class sip-profiles 777
 request ANY sip-header Contact modify "<sip:(.*)@10.21.15.100:5060>" "<sip:\1 a.b.c.d:5060>"
 response ANY sip-header Contact modify "<sip:(.*)@10.21.15.100:5060>" "<sip:\1 a.b.c.d:5060>"
 request ANY sip-header Via modify "SIP(.*) 10.21.15.100(.*)" "SIP\1 a.b.c.d\2"
 request ANY sdp-header Session-Owner modify "(.*IP4 ).*" "\1a.b.c.d"
 request ANY sdp-header Connection-Info modify "IN IP4 10.21.15.100" "IN IP4 a.b.c.d"
 request ANY sdp-header Audio-Connection-Info modify "IN IP4 10.21.15.100" "IN IP4 a.b.c.d"
 response ANY sdp-header Session-Owner modify "(.*IP4 ).*" "\1a.b.c.d"
 response ANY sdp-header Audio-Connection-Info modify "IN IP4 10.21.15.100" "IN IP4 a.b.c.d"
 response ANY sdp-header Connection-Info modify "IN IP4 10.21.15.100" "IN IP4 a.b.c.d"
 request ANY sip-header Remote-Party-ID modify "<sip:(.*)@10.21.15.100>" "<sip:\1 a.b.c.d>"
 response ANY sip-header Remote-Party-ID modify "<sip:(.*)@10.21.15.100>" "<sip:\1 a.b.c.d>"
!
```

```
### Apply to dial-peers for the side of the CUBE facing the ITSP


!
dial-peer voice 1 voip
 voice-class sip profiles 777
 voice-class sip profile 888 inbound
!
dial-peer voice 2 voip
 voice-class sip profiles 777
 voice-class sip profile 888 inbound
!
```

SIP复制列表

SIP Copylists是SIP配置文件的扩展，它允许网关从呼叫的支路复制报头，然后将PASTE复制到支路上的出口SIP消息中的另一个位置。Cisco IOS 15.1(3)T和Cisco IOS XE 3.6S中增加了SIP复制列表支持。这是一种非常强大的根据来自呼叫支路的其他报头创建动态报头的方法。

最常见的使用案例是将FROM报头动态复制到不同的报头，如转移或p-asserted-id，使得用户部分的值是from用户。这主要是为了身份验证和呼叫方ID目的。

完整文档：[思科统一边界要素配置指南 — Cisco IOS XE 17.6及以上版本]

SIP复制列表示例

<#root>

```
!

! Create Copylist to copy the FROM header on the inbound message specified later.


!
voice class sip-copylist <number>
 sip-header From
!

! Apply this to the inbound dial-peer of the call.


!
dial-peer voice <tag> voip
 voice-class sip copy-list <number>
!

! Create SIP Profile to take From (peer-header) stored as variable "u01" and apply to a header of choice
! This example modifies the user portion of the Contact by copying the user portion of the From header t


!
```

```
voice class sip-profiles <number>
 request INVITE peer-header sip From copy "<sip:(.*)@" u01
 request INVITE sip-header Contact modify "<sip:(.*)>" "<sip:\u01@10.50.244.2>"
!

! Apply the SIP profile to an outbound dial-peer

!
dial-peer voice <tag> voip
 voice-class sip profiles <number>
!
```

调试SIP配置文件和复制列表

```
debug voip ccapi inout
debug ccsip mess
debug ccsip info
debug ccsip feature sip-profile
```

示例SIP复制列表的调试输出

```
### Ingress from CUCM
Received:
INVITE sip:1001@10.50.228.61:5060 SIP/2.0
Via: SIP/2.0/TCP 10.50.244.3:5060;branch=z9hG4bKaad21bc3ae7e
From: "5001" <sip:5001@10.50.244.3>;tag=100442~cdffff43-5020-4e79-a10b-99d406971010-36470319
Contact: <sip:5001@10.50.244.3:5060;transport=tcp>

### Copylist Details
000440: Mar  8 18:59:49.796: //-1/xxxxxxxxxxxx/SIP/Info/info/64/sip_profiles_application_peer_copy_patte
000441: Mar  8 18:59:49.797: //187/D6138E000000/SIP/Info/info/64/sip_profiles_application_peer_copy_pat
000442: Mar  8 18:59:49.797: //-1/xxxxxxxxxxxx/SIP/Info/info/64/sip_profiles_prefix_slash_in_copy_var_v
000443: Mar  8 18:59:49.797: //187/D6138E000000/SIP/Info/info/64/sip_profiles_application_peer_copy_pat
000444: Mar  8 18:59:49.797: //-1/xxxxxxxxxxxx/SIP/Info/info/64/sip_profiles_application_modify_remove_
000445: Mar  8 18:59:49.797: //187/D6138E000000/SIP/Info/info/64/sip_profiles_check_and_get_variables_i
000446: Mar  8 18:59:49.797: //187/D6138E000000/SIP/Info/info/64/sip_profiles_check_and_get_variables_i
000448: Mar  8 18:59:49.797: //187/D6138E000000/SIP/Info/info/64/sip_profiles_check_and_get_variables_i
000449: Mar  8 18:59:49.797: //-1/xxxxxxxxxxxx/SIP/Info/info/64/sip_profiles_app_modify_header: Passing
000450: Mar  8 18:59:49.798: //-1/xxxxxxxxxxxx/SIP/Info/info/64/sip_profiles_application_modify_remove_
000451: Mar  8 18:59:49.798: //187/D6138E000000/SIP/Msg/ccsipDisplayMsg:

### Egress from CUBE
Sent:
INVITE sip:1001@14.50.228.63:5060 SIP/2.0
Via: SIP/2.0/UDP 10.50.228.61:5060;branch=z9hG4bK3C7CD
Remote-Party-ID: "5001" <sip:5001@10.50.228.61>;party=calling;screen=yes;privacy=off
From: "5001" <sip:5001@10.50.228.61>;tag=34C458-D6
Contact: <sip:5001@192.168.0.1>
```

特别说明

协议信令和媒体绑定

所有信令协议都允许管理员将信令绑定到特定接口。默认情况下，网关没有静态定义的绑定，则网关会从数据包经过的物理接口为呼叫分配信令。在拨号对等体上绑定时，数据包包含来自指定接口的源报头、消息和数据包，但实际数据包仍然通过物理接口路由。拨号对等体绑定始终使用会话发起协议(SIP)取代语音类租户和全局语音服务voip绑定。

很多时候，管理员将信令绑定到环回。这是逻辑接口，意味着没有数据包通过此接口。为了执行数据包捕获，必须在物理接口上执行捕获。命令show ip cef <remote-ip>显示数据包用来路由到目标/远程IP的物理接口，即使配置已绑定到虚拟接口也是如此。

媒体和信令绑定并不总是需要相同的IP。如果管理员需要绑定到特定接口，以便向/从CUCM发出信令，但电话和网关之间的音频/媒体可能需要绑定到另一个接口。

配置示例

此示例显示绑定到loopback 1的拨号对等体并接收来自CUCM的呼叫。

即使介质和信令（控制）绑定到环回1,show ip cef命令也会显示发送到CUCM的所有数据包都保留在物理接口GigabitEthernet0/0/1上。

```
!
dial-peer voice 2 voip
 description "Incoming call from CUCM"
 session protocol sipv2
 incoming called-number .
 voice-class sip bind control source-interface Loopback1
 voice-class sip bind media source-interface Loopback1
!
```

第7层应用绑定的操作顺序

1. 根据匹配的入站/出站拨号对等体上的bind语句。
2. 根据分配给匹配的入站/出站拨号对等体的语音类租户下的绑定。
3. 根据全局绑定语句。
4. 根据路由表，预计数据包将根据物理第3层接口退出。

SIP绑定命令

<#root>

```
! Per Dial-peer
```

```
!
dial-peer voice 1 voip
 voice-class sip bind control source-interface <interface>
 voice-class sip bind media source-interface <interface>
!
```

```
! Global Binding
```

```
!
voice service voip
 sip
   bind control source-interface <interface>
   bind media source-interface <interface>
!
```

MGCP绑定命令

```
!
mgcp bind control source-interface <interface>
mgcp bind media source-interface <interface>
!
```

SCCP绑定命令

```
!
sccp local <interface>
!
sccp ccm group <number>
 bind interface <interface>
!
```

H323 Binding命令

<#root>

```
!
inteface <interface>
```

```
 !

 ! Media Bind Command:

 h323-gateway voip interface
 !

 ! Signaling Bind Command:

 h323-gateway voip bind srcaddr <a.b.c.d>
!
```

DNS和VoIP拨号对等体

与任何其他DNS解决方案一样，使用带VOIP的DNS。常见配置是使用会话目标dns:example.com...

即使未全局配置网关上的ip域查找，Cisco网关也会执行DNS解析。这意味着即使您正在禁用DNS，VOIP拨号对等体仍然会解析DNS条目。但是,最近，在Cisco IOS XE 3.16S中，Cisco IOS XE平台中的整体DNS功能发生了一些更改。

完成此更改后，使用会话目标dns:example.com配置的拨号对等体。现在请遵守以下事实：DNS已禁用，且未进行ip域查找。
建议始终确保在使用DNS时配置“ip domain lookup”命令，以避免此问题。

对于出站SIP连接，CUBE会按此顺序执行DNS解析操作。

1. SRV查询查找
2. 记录查找
3. AAAA记录查找

有关如何创建SRV或如何跳过SRV并在会话目标上执行A记录查询的信息，请参阅完整文档：Cisco Unified Border Element配置指南 — Cisco IOS XE 17.6及更高版本

对于Cisco IOS网关需要解析报头以响应消息的入站SIP连接，网关可以使用此操作顺序进行DNS解析

1. 记录查找
2. AAAA记录查找

在Cisco IOS XE 17.9.1中，CUBE可以通过保持连接机制选项来检查DNS会话目标的可达性。请参

阅完整文档：

[Cisco Unified Border Element配置指南 — Cisco IOS XE 17.6或更高版本]

Cisco IOS DNS配置示例

```
ip host _sip._udp.cucmgroup.lab.local srv 1 50 5060 cucm1.lab.local
ip host _sip._udp.cucmgroup.lab.local srv 1 50 5060 cucm2.lab.local
ip host _sip._udp.cucmgroup.lab.local srv 1 50 5060 cucm3.lab.local

ip host cucm1.lab.local 10.0.0.1
ip host cucm2.lab.local 10.0.0.2
ip host cucm3.lab.local 10.0.0.3

ip domain name lab.local
ip name-server 8.8.8.8
```



注意：15.6(1)S / 3.17.00.S及更高版本支持Cisco IOS XE上的DNS SRV。

DNS调试和验证命令

<#root>

```
show host
clear host all *
!
debug ip dns view
debug ip domain
debug ccsip info
debug ccsip error
```

DNS测试3.15S及更高版本

<#root>

```
### Domain Name Verification

Gateway# sh run | s lookup
no ip domain lookup

### Checking the host table for no entry
```

```
Gateway# show host
Name lookup view: Global
Default domain is cisco.com
Name/address lookup uses static mappings

Codes: UN - unknown, EX - expired, OK - OK, ?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined

Host                      Port  Flags      Age Type   Address(es)
```

Verification of no PING on a FQDN

```
Gateway# ping cucm.cisco.com
Translating "cucm.cisco.com"
% Unrecognized host or address, or protocol not running.
```

Made a test call here

Checking logs to see if it worked

```
Gateway# sh log | s INVITE sip:
INVITE sip:9001@14.50.228.70:5060 SIP/2.0
INVITE sip:5001@cucm.cisco.com:5060 SIP/2.0
```

Host Table now has an entry

```
Gateway# sh host
Name lookup view: Global
Default domain is cisco.com
Name/address lookup uses static mappings

Codes: UN - unknown, EX - expired, OK - OK, ?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined

Host                      Port  Flags      Age Type   Address(es)
cucm.cisco.com            None  (temp, OK)  0   IP    10.50.244.2
```

CCSIP All output showing a proper DNS Query for the FQDN on the dial-peer.

```
001338: Mar  9 15:29:07.437: //-1/xxxxxxxxxxxx/SIP/Info/info/1024/httpish_msg_free: Freed msg=0x7FE9873
001339: Mar  9 15:29:07.437: //-1/xxxxxxxxxxxx/SIP/Info/notify/8192/sip_dns_type_srv_query: TYPE SRV qu
001340: Mar  9 15:29:07.438: //-1/xxxxxxxxxxxx/SIP/Info/info/8192/sip_dns_type_a_aaaa_query: DNS query
001341: Mar  9 15:29:07.441: //-1/xxxxxxxxxxxx/SIP/Info/notify/8192/sip_dns_type_a_query: TYPE A query
001342: Mar  9 15:29:07.441: //-1/xxxxxxxxxxxx/SIP/Info/info/8192/sip_dns_type_a_query: ttl for A recor
001343: Mar  9 15:29:07.441: //-1/xxxxxxxxxxxx/SIP/Info/info/8192/sip_dns_type_a_aaaa_query: IP Address
001344: Mar  9 15:29:07.441: //-1/xxxxxxxxxxxx/SIP/Info/info/8192/sip_dns_type_a_aaaa_query: 10.50.244.
```

DNS测试3.16S及更高版本。

<#root>

Checking the command is present

```
Gateway# sh run | s lookup
no ip domain lookup
```

Verifying the gateway cannot ping a FQDN

```
Gateway# ping cucm.cisco.com
% Unrecognized host or address, or protocol not running.
```

Checking the Host Table for entries

```
Gateway# sh host
Default domain is cisco.com
Name servers are 10.50.244.52
NAME   TTL   CLASS    TYPE      DATA/ADDRESS
-----------------------------------------
```

Made a test call here

CCSIP All Outbound showing the failed call

```
000974: *Mar  9 15:53:01.222: //-1/xxxxxxxxxxxx/SIP/Info/info/1024/httpish_msg_free: Freed msg=0x7FF31D
000975: *Mar  9 15:53:01.222: //-1/xxxxxxxxxxxx/SIP/Info/notify/8192/sip_dns_type_srv_query: TYPE SRV q
000976: *Mar  9 15:53:01.224: //-1/xxxxxxxxxxxx/SIP/Info/info/8192/sip_dns_type_a_aaaa_query: DNS query
000977: *Mar  9 15:53:01.225: //-1/xxxxxxxxxxxx/SIP/Error/sip_dns_type_a_query:
 TYPE A query failed for cucm.cisco.com
000978: *Mar  9 15:53:01.225: //-1/xxxxxxxxxxxx/SIP/Error/_send_dns_fail:
 DNS Query for cucm.cisco.com failed
000984: *Mar  9 20:53:01.225: %VOICE_IEC-3-GW: SIP: Internal Error (DNS query fail): IEC=10.1.128.7.47.
```

最大连接和带宽

默认情况下，VOIP和POTS拨号对等体允许无限连接（呼叫）和带宽（仅VOIP拨号对等体）。 对于可使用的呼叫数量或带宽有限的中继，使用max-conn或max-bandwidth命令非常有用。max-conn在Cisco IOS 11.3(1)T中添加，并且存在于所有Cisco IOS XE版本中，而max-bandwidth在15.2(2)T和Cisco IOS-XE 3.7S中添加。

配置示例

此处，您告知网关使用max-conn 30命令将拨号对等体1呼叫限制为30个呼叫。
拨号对等体2正在限制该拨号对等体的带宽，以使我们不超过分配的限制。

```
!
dial-peer voice 1 voip
 description ITSP SIP Trunk - 30 Max Calls!
 session protocol sipv2
 sess target ipv4:10.10.10.10
 destination-pattern 8675309$
 max-conn 30
!
dial-peer voice 2 voip
 description SIP Trunk with Bandwidth Restrictions!
 session protocol sipv2
 sess target ipv4:10.10.10.10
 destination-pattern 123456789$
 max-bandwidth 400
!
```

超过最大连接阈值时的示例错误。

```
000308: Oct  5 19:01:02.603: %CALL_CONTROL-6-MAX_CONNECTIONS: Maximum number of connections reached for

000309: Oct  5 19:01:02.603: %VOICE_IEC-3-GW: CCAPI: Internal Error (Dial-peer connections exceeded): I
000310: Oct  5 19:01:02.604: %SIP-3-MAXCONNCAC: Call rejected due to CAC based on maximum number of con

000311: Oct  5 19:01:02.604: //17084/86B070800000/SIP/Msg/ccsipDisplayMsg:
Sent:
SIP/2.0 503 Service Unavailable
Via: SIP/2.0/TCP 10.50.244.62:5060;branch=z9hG4bKb78c35aa21b0
From: <sip:9001@10.50.244.62>;tag=72531~2e8ca155-3f0b-4f07-a1b2-b14ef77ceb7f-26250846
To: <sip:1234@10.50.245.70>;tag=3E19564D-1684
Date: Thu, 05 Oct 2017 19:01:02 GMT
Call-ID: 86b07080-9d61816e-b762-3ef4320e@10.50.244.62
CSeq: 101 INVITE
Allow-Events: telephone-event
Warning: 399 10.50.245.70 "Maximum Number of Connections reached for dial-peer 1"
Server: Cisco-SIPGateway/IOS-15.4.3.S4
Content-Length: 0
```

直接拨入（DID）

单阶段拨号

在POTS拨号对等体上启用直接拨入后，入站消息传送可以包含路由呼叫所需的所有数字。思科网关无法执行后续数字收集。当路由器或网关搜索出站拨号对等体时，设备使用整个传入拨号字符串。默认情况下，这种匹配是变长的。这种匹配不会按数字逐个进行，因为根据 DID 定义，已经收到了所有数字。这是POTS拨号对等体的默认配置。

完整文档[：了解IOS语音数字(T1/E1)接口上的直接拨入(DID)]

配置示例

```
!
dial-peer voice 1 pots
 incoming called-number 8675309
 voice-port 0/0/0
 direct-inward-dial
!
```

两阶段拨号

如果传入的POTS拨号对等体配置为no direct-inward-dial，则路由器或网关将进入数字收集模式（数字在带内收集）。 出站拨号对等体匹配以数字为单位完成。路由器或网关在设备收到每个数字后检查拨号对等体匹配，然后在完全匹配时路由呼叫。

配置示例

```
!
dial-peer voice 1 pots
 incoming called-number 8675309
 voice-port 0/0/0
 no direct-inward-dial
!
```

阻止呼叫

每种协议处理呼叫阻塞的方式略有不同。大多数协议都可以使用translation-rule reject模式，该模式基于数字字符串进行阻止。如果管理员仍要应用入站转换配置文件进行正常数字处理，但不阻止其中的任何数字，则可以选择使用call-block translation-profile命令实施呼叫阻止。

```
!
voice translation-rule 164
 rule 1 reject /8675309/
!
voice translation-profile CALLBLOCK
```

```
 translate calling 164
!
dial-peer voice 1 pots
 desc INCOMING VOICE-PORT with BLOCK
 translation-profile incoming ANOTHER
 call-block translation-profile incoming CALLBLOCK
 call-block disconnect-cause incoming invalid-number
 incoming called-number .
 port 0/0/0:23
!

Gateway#test voice translation-rule 164 8675309
8675309 blocked on rule 1
```

在E1 R2中，管理员可以阻止收集呼叫。这主要用于巴西部署，但可以通过任何cas自定义组进行配置。

这两个选项是：

1. 根据从电信交换机接收的II-8类别阻止传入的收集呼叫。这是默认机制，所有Cisco网关无需任何配置即可执行此操作。这种屏蔽方法需要更新的电信公司交换机，该交换机支持基于类别的标记。要禁用此方法，请在cas-custom组上使用collect-call-enable命令。
2. 在r2数字E1 R2 cas-custom组上使用double-answer功能。双应答配置会禁用基于类别的阻塞，并将其替换为双应答功能。首先应答来电，然后启动1秒计时器。1秒后，网关以CLEAR BWD命令的形式发送释放。然后，电信公司可以将CLEAR FWD发送到网关，并且呼叫可以断开连接。计时器在网关发送CLEAR BWD后启动，如果此计时器过期，网关将发送另一个ANSWER信号，因此假设呼叫不是收集呼叫，从此呼叫继续正常进行。可以在cas-custom组内使用cc-reanswer-to配置此计时器。

Category II-8 Block消息(debug vpm signal)

<#root>

```
009228: Nov 21 12:02:00.955 GMT: //-1/BF12BE36BAC8/VTSP:(0/0/0:0):-1:1:2/vtsp_report_cas_digit:
   Begin Digit=8, Mode=CC_TONE_R2_MF_BACKWARD_MODE
009229: Nov 21 12:02:00.955 GMT: htsp_digit_ready_up(0/0/0:0(2)):

Rx digit='8'

009230: Nov 21 12:02:00.955 GMT: R2 Incoming Voice(0/0): DSX (E1 0/0/0:1): STATE: R2_IN_CATEGORY R2 Got
009231: Nov 21 12:02:00.955 GMT: Enter r2_comp_category
009232: Nov 21 12:02:00.955 GMT:

R2 Event : 8

009233: Nov 21 12:02:00.955 GMT:

#######R2_II8 TRUE########
```

```
009234: Nov 21 12:02:00.955 GMT:

####### collect_call_enable = 0

009235: Nov 21 12:02:00.955 GMT:

############sending B7 ##########

009236: Nov 21 12:02:00.955 GMT: r2_reg_generate_digits(0/0/0:0(2)):

Tx digit '7'

009237: Nov 21 12:02:01.055 GMT: //-1/BF12BE36BAC8/VTSP:(0/0/0:0):-1:1:2/vtsp_report_cas_digit:
   End Digit=8, Mode=CC_TONE_R2_MF_BACKWARD_MODE
009238: Nov 21 12:02:01.055 GMT: htsp_digit_ready(0/0/0:0(2)): Rx digit='#'
009239: Nov 21 12:02:01.055 GMT: R2 Incoming Voice(0/0): DSX (E1 0/0/0:1): STATE: R2_IN_CATEGORY R2 Got
009240: Nov 21 12:02:01.055 GMT: Enter r2_comp_category
009241: Nov 21 12:02:01.055 GMT: r2_reg_generate_digits(0/0/0:0(2)): Tx digit '#'
009242: Nov 21 12:02:01.359 GMT: htsp_dsp_message: SEND_SIG_STATUS: state=0x8 timestamp=22365 systime=2
009243: Nov 21 12:02:01.359 GMT: htsp_process_event: [0/0/0:0(2), R2_Q421_IC_WAIT_ANSWER, E_DSP_SIG_100
009244: Nov 21 12:02:01.359 GMT: r2_q421_ic_clr_fwd_idle(0/0/0:0(2)) Rx CLEAR FWD
009245: Nov 21 12:02:01.359 GMT: r2_reg_channel_disconnected(0/0/0:0(2))
009246: Nov 21 12:02:01.359 GMT: R2 Incoming Voice(0/0): DSX (E1 0/0/0:1): STATE: R2_IN_CATEGORY R2 Got
009247: Nov 21 12:02:01.359 GMT: Enter r2_comp_category
009248: Nov 21 12:02:01.359 GMT: htsp_timer - 2000 msec
009249: Nov 21 12:02:01.359 GMT: htsp_process_event: [0/0/0:0(2), R2_Q421_IC_CLR_FWD, E_HTSP_RELEASE_REQ
009250: Nov 21 12:02:01.359 GMT: r2_q421_null_release(0/0/0:0(2)) E_HTSP_RELEASE_REQ
009251: Nov 21 12:02:01.359 GMT: r2_reg_process_event: [0/0/0:0(2), R2_REG_COLLECTING, E_R2_REG_DISCONN
009252: Nov 21 12:02:01.359 GMT: r2_reg_disconnect_collect(0/0/0:0(2))
009253: Nov 21 12:02:01.359 GMT: r2_reg_timer_stop(0/0/0:0(2))
009254: Nov 21 12:02:01.711 GMT: htsp_process_event: [0/0/0:0(1), R2_Q421_IC_CLR_FWD, E_HTSP_EVENT_TIME
009255: Nov 21 12:02:01.711 GMT: htsp_timer_stop
009256: Nov 21 12:02:01.711 GMT: r2_q421_clr_fwd_idle(0/0/0:0(1)) Tx IDLEvnm_dsp_set_sig_state:[R2 Q.42
009257: Nov 21 12:02:01.711 GMT: r2_reg_channel_disconnected(0/0/0:0(1))
009258: Nov 21 12:02:01.711 GMT: //682206/0C63B263B9C9/VTSP:(0/0/0:0):0:1:1/vtsp_do_call_history:
   Coder Rate=5
009259: Nov 21 12:02:01.711 GMT: r2_reg_process_event: [0/0/0:0(1), R2_REG_IDLE, E_R2_REG_DISCONNECT(91
```

双应答配置示例

```
!
controller e1 0/0/0
 ds0-group 0 timeslots 1-15,17-31 type r2-digital r2-compelled ani
 cas-custom 0
  country brazil
  double-answer
  cc-reanswer-to 3000
!
```

双应答调试(debug vpm signal)

<#root>

Answer the call and start a 1 second timer

May 23 09:52:59.180 BR: r2_q421_ic_answer(0/0/0:0(18))

Tx ANSWER

 seizure: delay 0 ms,elapsed 12404 msvnm_dsp_set_sig_state:[R2 Q.421 0/0/0:0(18)] set signal state = 0x
May 23 09:52:59.180 BR: r2_reg_channel_connected(0/0/0:0(18))
May 23 09:52:59.180 BR:

htsp_timer - 1000 msec

May 23 09:52:59.180 BR: //23899578/92233E71B421/CCAPI/cc_api_voice_mode_event:
   Call Id=23899578
May 23 09:52:59.180 BR: //23899578/92233E71B421/CCAPI/cc_api_voice_mode_event:
   Call Entry(Context=0x1E73AD8)
May 23 09:52:59.180 BR: htsp_process_event: [0/0/0:0(18), R2_Q421_IC_DOUBLE_ANS_ANS, E_HTSP_VOICE_CUT_T
May 23 09:52:59.184 BR: //23899578/92233E71B421/CCAPI/cc_process_notify_bridge_done:
   Conference Id=0x10AD1, Call Id1=23899578, Call Id2=23899579
May 23 09:52:59.184 BR: r2_reg_process_event: [0/0/0:0(18), R2_REG_WAIT_FOR_CONNECT, E_R2_REG_CONNECT(9
May 23 09:52:59.184 BR: r2_reg_connect(0/0/0:0(18))

One Second Passes and we clear the call and start a 2 second timer

May 23 09:53:00.180 BR: htsp_process_event: [0/0/0:0(18), R2_Q421_IC_DOUBLE_ANS_ANS, E_HTSP_EVENT_TIMER
May 23 09:53:00.180 BR: r2_q421_ic_d_answ_answ_to(0/0/0:0(18)) E_TIMER_EVENT
May 23 09:53:00.180 BR: htsp_timer - 2000 msec
May 23 09:53:00.180 BR: r2_q421_ic_d_answ_answ_to(0/0/0:0(18))

Tx CLEAR BWD

vnm_dsp_set_sig_state:[R2 Q.421 0/0/0:0(18)] set signal state = 0xC
May 23 09:53:00.824 BR: htsp_process_event: [0/0/0:0(18), R2_Q421_IC_DOUBLE_ANS_RLS, E_DSP_SIG_1000]
May 23 09:53:00.824 BR: r2_q421_ic_answer_clr_fwd(0/0/0:0(18))

Rx CLEAR FWD

May 23 09:53:00.824 BR: r2_reg_channel_disconnected(0/0/0:0(18))
May 23 09:53:00.824 BR:

htsp_timer - 2000 msec

May 23 09:53:00.824 BR: r2_reg_process_event: [0/0/0:0(18), R2_REG_CONNECTED, E_R2_REG_DISCONNECT(91)]
May 23 09:53:00.824 BR: r2_reg_disconnect_idle(0/0/0:0(18))
May 23 09:53:00.824 BR: R2 Incoming Voice(0/0): DSX (E1 0/0/0:17): STATE: R2_IN_IDLE R2 Got Event R2_ST
May 23 09:53:00.824 BR: r2_reg_timer_stop(0/0/0:0(18))

2 second passes and the gateway release the call

May 23 09:53:02.824 BR: htsp_process_event: [0/0/0:0(18), R2_Q421_IC_CLR_FWD, E_HTSP_EVENT_TIMER]
May 23 09:53:02.824 BR: htsp_timer_stop

```
May 23 09:53:02.824 BR: r2_reg_channel_disconnected(0/0/0:0(18))
May 23 09:53:02.824 BR: //23899578/92233E71B421/VTSP:(0/0/0:0):17:1:1/vtsp_cc_call_disconnected:
   Cause Value=16
May 23 09:53:02.824 BR: //23899578/92233E71B421/CCAPI/cc_api_call_disconnected:
   Cause Value=16, Interface=0xB41CEBC, Call Id=23899578
```

ISDN overlap-receiving命令

在ISDN接口上配置isdn overlap-receiving命令时，会对入站拨号对等体匹配产生影响。在ISDN层收到每个数字后，将对拨号对等体进行匹配检查。如果发现了完全匹配，则不等待更多数字，就立即路由此呼叫（本示例中为路由到会话应用程序）。T终结器可用于挂起逐位匹配，并强制路由器或网关等待，直到收到所有数字。T是指ISDN级别的T302数字间计时器，可在与ISDN接口关联的串行接口下配置。ISDN还提供其他表示数字结尾的机制，例如在Q.931信息消息中设置发送完整信息元素(IE)。

空被叫号码

当拨号对等体配置有传入被叫号码T时，将显示显示的警告消息。

示例输出

```
Gateway(config)# dial-peer voice 1 pots
Gateway(config-dial-peer)# incoming called-number T
Warning: Pattern T defines a match with zero or more digits and hence could
match with an empty number. If this is not the desired behaviour please
configure pattern .T instead to match on one or more digits
```

有关传入拨号对等体与空被叫号码匹配的特殊说明。

- 空被叫号码与语音端口和/或某些情况下的应答地址相比被认为资格较低。因此，仅当没有基于answer-address或port-number的匹配时，才会发生基于空被叫号码的匹配。
- 在重叠拨号的情况下，空被叫号码与传入被叫号码T不匹配，因为未发生超时。
- 空的被叫号码只能在ENBLOCK的情况下与传入的被叫号码T匹配，并且由于answer-address和port-number原因没有匹配。管理员配置传入被叫号码T时显示的警告指此特定情况。

限制类别

限制类别(COR)是一种限制思科网关上呼叫的方法。COR 经常被描述为锁和密钥机制。锁分配给具有传出COR列表的拨号对等体。密钥被分配给具有传入COR列表的拨号对等体。应用COR列表时，可用的出站拨号对等体是密钥可以解锁的拨号对等体。此过滤发生在检查其他出站拨号对等体匹配方法之前。

具有限制类的两条重要规则：

1. 如果没有应用传出COR列表，则始终路由呼叫。
2. 如果没有传入COR列表，呼叫始终被路由。

限制类别(COR)、逻辑分区限制类别(LPCOR)和带强制授权码(FAC)的LPCOR的配置不在本文档的讨论范围之内，但可以参考这些文档进行进一步阅读。

COR	配置限制类别（COR）
带CME的LPCOR	带LPCOR的CME配置示例
LPCOR，带CME和FAC	Cisco Unified Communications Manager Express 系统管理员指南

Cisco Unified Communications Manager Express(CUCME)拨号对等体

CME为电子电话和语音注册池创建系统拨号对等体。这些在运行配置中无法看到。要更改CME拨号对等体，需要在实际的ephone或语音注册池中进行更改。当查看show dial-peer voice summary输出时，以2000开头的拨号对等体是SCCP电话，而以4000开头的拨号对等体是SIP语音注册池。此拨号对等体显示为来自CME注册电话的呼叫的入站拨号对等体，显示为用于CME注册电话的呼叫的出站拨号对等体的调试。

使用CME的show dial-peer voice summary的输出示例。

```
Gateway# show dial-peer voice sum | s 2000|4000
20001  pots  up   up             1001$              0                              50/0/1
20002  pots  up   up             4001$              0                              50/0/2
20003  pots  up   up             4002$              0                              50/0/3
20004  pots  up   up             7001$              0                              50/0/4
20005  pots  up   up             3009$              0                              50/0/5
20006  pots  up   up             8810....$          0                              50/0/10
20007  pots  up   up             8811....$          0                              50/0/11
40001  voip  up   up             14085151111$       0  syst ipv4:14.50.214.67:50
40002  voip  up   up             19725252222$       0  syst ipv4:14.50.214.67:50
40003  voip  up   up             85225353333$       0  syst ipv4:14.50.214.67:50
40004  voip  up   up             442084445555$      0  syst ipv4:14.50.214.67:50
40005  voip  up   up             911$               0  syst ipv4:14.50.214.67:50
40006  voip  up   up             18005550100$       0  syst ipv4:14.50.214.67:50
```

```
40008  voip  up   up            2001$             0  syst ipv4:14.50.214.51:50
```

show voice register dial-peers with SIP CME的输出示例。

```
Gateway# show voice register dial-peers
Dial-peers for Pool 2:

dial-peer voice 40006 voip
 destination-pattern 14085151111$
 session target ipv4:14.50.214.67:5060
 session protocol sipv2
 dtmf-relay rtp-nte
 digit collect kpml
 codec  g711ulaw bytes 160
 no vad
  call-fwd-all         8888
  after-hours-exempt   FALSE

dial-peer voice 40005 voip
 destination-pattern 19725252222$
 session target ipv4:14.50.214.67:5060
 session protocol sipv2
 dtmf-relay rtp-nte
 digit collect kpml
 codec  g711ulaw bytes 160
 no vad
  after-hours-exempt   FALSE
```

带有拨号对等体的MGCP和SCCP

MGCP和SCCP遵循自己的拨号对等体规则。他们使用的唯一概念是，必须为呼叫配置所需的语音端口。其余部分由STCAPP和MGCPAPP过程处理。当您检查这些拨号对等体的配置时，它们使用service mgcpapp或service stcapp命令。这些功能使选择的应用程序能够使用拨号对等体，并告知应用程序它可以处理哪个拨号对等体。

在调试这些协议时，输出不会显示入站拨号对等体匹配。这始终可以显示为拨号对等体0。因为它不存在。处理应用程序的呼叫代理已选择要将呼叫发送到哪个端口，入站拨号对等体匹配没有用处，因为网关无法控制呼叫的该段。但是，可以观察到出站拨号对等体匹配。这仅仅是为了显示，因为最终处理该过程的呼叫代理也控制呼叫的那一侧。

请记住，拨号对等体只告诉应用程序要控制哪个物理语音端口。由于大多数呼叫由外部呼叫代理和网关控制，因此它只需按照要求执行操作。您将跳过本节的基本操作方法，并提供一些配置以开始使用。

MGCP配置示例[使用CUCM自动配置*]

```
!
mgcp call-agent 10.10.10.10
mgcp
!
ccm-manager mgcp [codec-all]
ccm-manager config server 10.10.10.10
ccm-manager config
ccm-manger redundant-host 10.10.10.20
!
voice-port 0/0/0
 description The MGCP port to register
 no shut
!
dial-peer voice 1 pots
 description Defining the Port for the MGCP application
 service mgcpapp
 port 0/0/0
!
hostname myrouter
ip domain name cisco.com
ip name server 10.10.10.30
!
ip tftp source-interface gig0/0/0
!
```

完整的MGCP文档：[Cisco Unified Communications Manager和互操作性配置指南，Cisco IOS版本15M&T]

SCCP/STCAPP配置示例[使用CUCM自动配置*]

```
!
stcapp ccm-group 1
stcapp
!
sccp local gig0/0/0
sccp ccm 10.10.10.10 id 1 priority 1 version 7.0+
sccp ccm 10.10.10.20 id 1 priority 2 version 7.0+
sccp
!
sccp ccm group 1
 bind interface gig0/0/0
 associate ccm 1 priority 1
 associate ccm 2 priority 2
!
ccm-manager config server 10.10.10.10
ccm-manager sccp local gig0/0/0
ccm-manager sccp
!
voice-port 0/0/0
 description The SCCP port to register
 no shut
!
dial-peer voice 1 pots
 description Defining the Port for the SCCP application
 service stcapp
```

```
 port 0/0/0
!
ip tftp source-interface gig0/0/0
!
```

如果管理员不希望CUCM配置网关，只需删除ccm-manager命令。包括拨号对等体配置是为了明确该概念的工作原理。如果存在ccm-manager配置，CUCM会根据CUCM中的端口配置创建这些拨号对等体，因此无需实际定义拨号对等体。CUCM创建的拨号对等体通常以999开头，然后是另外三个数字。

带有拨号对等体的SIP DSAPP

SIP DSAPP已添加到Cisco IOS XE 16.12.1+和CUCM 12.5.1SU+中

通过此功能，模拟语音端口（如FXS）可以由CUCM注册和管理。使用DSAPP的呼叫路由与MGCP或SCCP略有不同，因为拨号对等体仍然正常匹配。即，网关可以从FXS端口收集数字，并在VOIP拨号对等体上执行拨号对等体查找。找到匹配项后，会将INVITE发送到CUCM封装以供CUCM执行进一步的数字分析。

SIP DSAPP配置示例[使用CUCM自动配置*] | Cisco IOS-XE 16.12.1+和CUCM 12.5.1SU+

```
!
dsapp line
!
voice service voip
 sip
  bind control source-interface GigabitEthernet0/0/0
  bind media source-interface GigabitEthernet0/0/0
  session transport tcp
!
application
 service dsapp
  param dialpeer 777
!
global
 service default dsapp
!
ccm-manager config server 10.10.10.10
ccm-manager sipana auto-config local GigabitEthernet0/0/0
!
dial-peer voice 777 voip
 destination-pattern 9T
 session protocol sipv2
 session target ipv4:10.10.10.10
 session transport tcp
 incoming called-number .
 voice-class sip extension gw-ana
 voice-class sip bind control source-interface GigabitEthernet0/0/0
 dtmf-relay rtp-nte
 codec g711ulaw
```

```
!
dial-peer voice 19990100 pots
 service dsapp
 destination-pattern 7776
 voice-class sip extension gw-ana
 port 0/1/0
!
sip-ua
 registrar ipv4:10.10.10.10 expires 3600 tcp
!
```

完整的SIP DSAPP文档：Cisco VG450语音网关软件配置指南。

呼叫路由故障排除和验证

请参阅本文档了解更多详细信息。

- 为统一边界要素(CUBE)和时分复用(TDM)网关配置调试集合

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。