配置CUBE和TDM网关的调试集合

目录

简介

先决条件

- 要求
- 使用的组件

背景信息

TDM语音网关与CUBE

Cisco IOS/IOS XE语音调试的集合

- 如何通过命令行界面(CLI)访问Cisco IOS/IOS XE路由器
- 如何设置终端监控器以收集show命令或调试
- 从CLI收集基本show命令输出
- 从CLI收集调试输出
  - 内存检查
  - 中央处理器(CPU)检查
  - 当前激活的呼叫检查
  - 日志记录缓冲区设置
  - 配置系统日志设置
  - 调试收集

可在语音路由器中启用哪些调试

- 内部呼叫控制API(CCAPI)调试
- SIP呼叫流
  - 基本SIP调试
  - 高级SIP调试
- 数字(PRI、BRI)呼叫流
  - 基本数字调试
  - 高级数字调试
- 模拟呼叫流
- MGCP呼叫流
  - 基本调试
  - CCM-Manager调试
  - 高级MGCP调试
- H323呼叫流
  - 基本H323调试
  - 高级H323调试
- SCCP媒体资源
  - 基本SCCP调试
  - 高级SCCP调试

VoIP跟踪

- 限制
- 如何启用VoIP跟踪
- 如何禁用VoIP跟踪
- 配置内存限制

如何显示VoIP跟踪数据
Show VoIP Trace All
Show VoIP Trace Cover-Buffers
Show VoIP Trace Call-Id
显示VoIP跟踪统计信息
其他Show命令

简介

本文档介绍在Cisco IOS®/IOS XE®语音路由器中收集语音调试的一些最佳实践。

先决条件

要求

- 在ISR路由器中执行命令的特权访问。
- 需要具有IP语音(VoIP)协议方面的经验。
- 对于VoIP跟踪，至少需要Cisco IOS XE 17.4.1或17.3.2。
- 集成多业务路由器(ISR)内部的Cisco IOS/IOS XE基础知识。

使用的组件

本文档中使用的组件包括：

- PuTTY
- 思科ISR 3925
- 思科ISR 4451

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

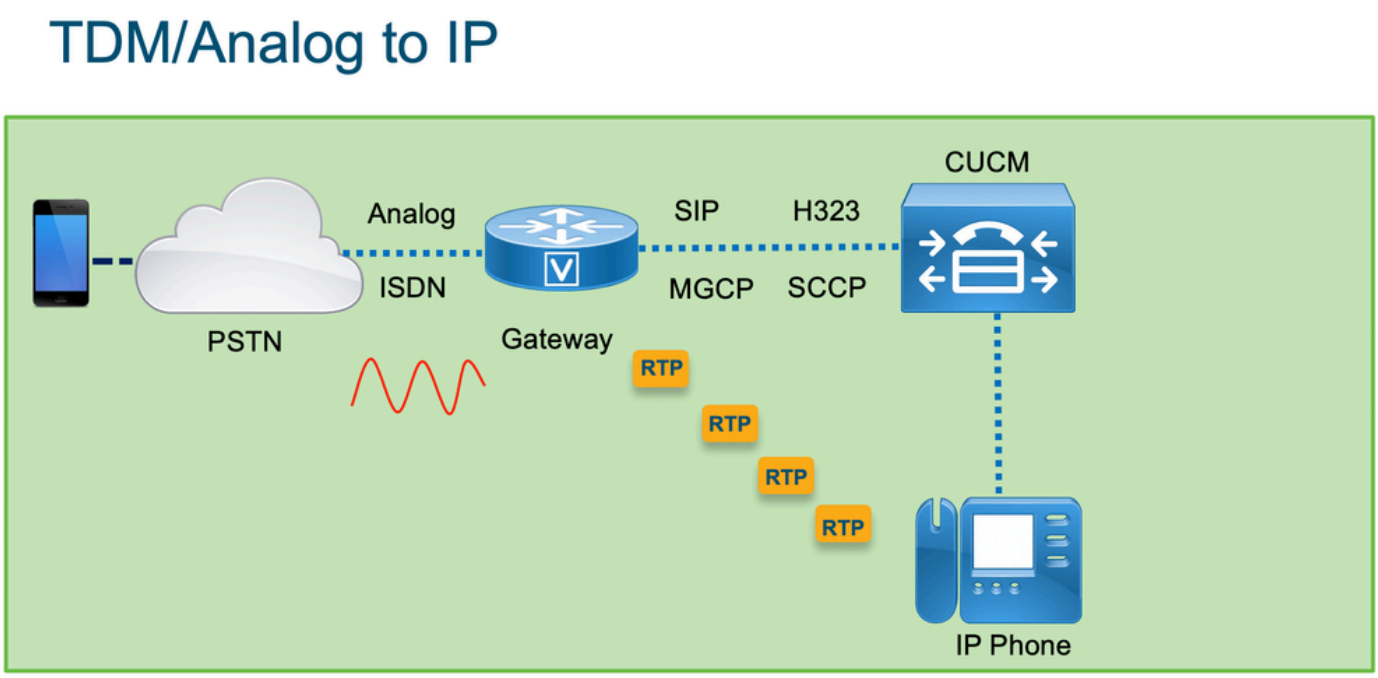
这些平台中的调试收集过程存在挑战，并且可能会影响设备的性能。当语音路由器中建立多个活动

呼叫时，挑战和风险会增加。在某些情况下，如果未正确收集调试，则会导致CPU使用率过高，从而造成路由器容量损坏，甚至触发软件崩溃。本文档讨论思科统一边界要素(CUBE)和时分复用(TDM)/模拟网关之间的区别。

TDM语音网关与CUBE

TDM语音网关主要用于将内部电话系统与另一个专用交换机(PBX)或公共交换电话网络(PSTN)互连。 TDM网关中使用的连接类型是T1/E1控制器（ISDN或CAS）和模拟电路，例如FXS和FXO端口。数字信号处理器(DSP)将音频从原始形式转换为RTP数据包。类似地，在DSP处理了RTP数据包并将音频发送到特定电路之后，RTP数据包会转换为原始音频。这些网关可与VoIP端和TDM端的H323、MGCP或瘦呼叫控制协议(SCCP)进行互操作。其ISDN PRI电路或模拟作为到PSTN或终端的最常见连接。

如图所示，TDM网关在您的内部VoIP基础设施与模拟或ISDN服务提供商之间提供了一个桥梁：



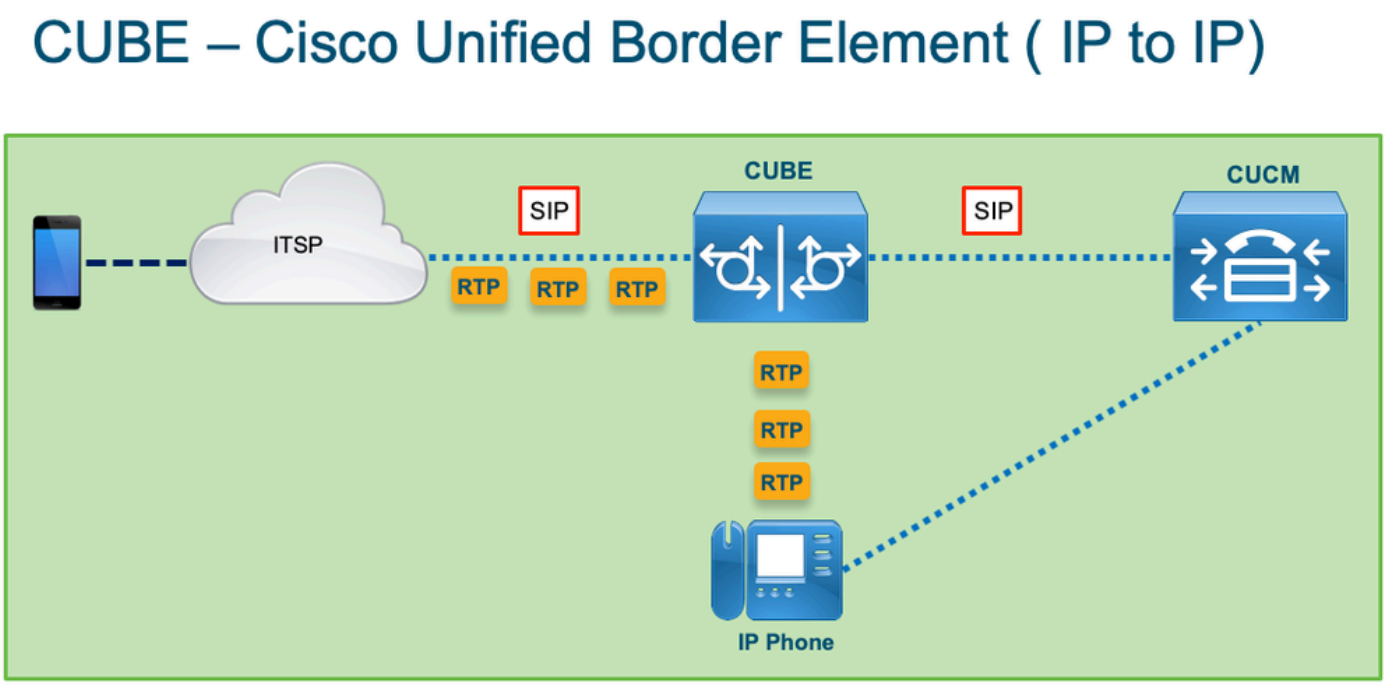
随着VoIP的引入，客户迅速将其传统系统转变为现代VoIP基础设施。在运营商端也发生了同样的情况，他们现在使用连接将本地电话服务与运营商VoIP基础设施互联，并扩展其功能以提供更好的服务。目前最常用的VoIP协议是会话发起协议(SIP)，目前被全球的客户和互联网电话服务提供商(ITSP)广泛使用。

引入CUBE是为了通过将SIP作为主要VoIP协议的ITSP，提供将这些内部VoIP系统与外部世界互连

的方法。CUBE只是一个IP-IP网关，它不再需要T1/E1控制器或模拟端口等任何TDM类型的连接。CUBE与TDM网关在同一平台上运行。

最常用的VoIP协议是SIP（用于呼叫建立和呼叫断开）和RTP（用于媒体传输）。在CUBE中，不需要使用DSP，除非需要转码器。RTP流量从ITSP端到端传输到终端，CUBE充当中间人，地址隐藏是其提供的众多功能之一。

如图所示，CUBE将您的内部VoIP基础设施与SIP ITSP区分开来：

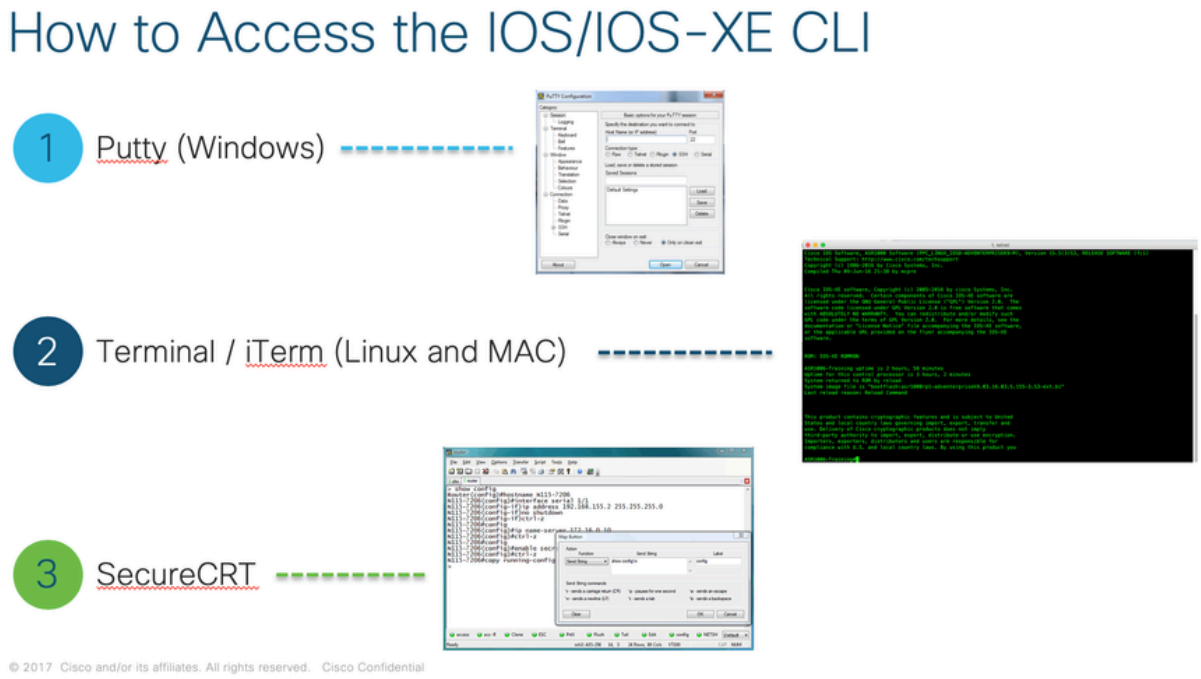


Cisco IOS/IOS XE语音调试的集合

语音功能在不同平台列表上运行，例如ISR、ASR、CAT8Ks等；但是，它们使用的是通用软件，即Cisco IOS或Cisco IOS XE（本文未介绍Cisco IOS和Cisco IOS XE之间的区别）。 以下是有关如何访问Cisco IOS路由器的基本信息。

如何通过命令行界面(CLI)访问Cisco IOS/IOS XE路由器

与任何其他基于CLI的设备一样，路由器需要终端监控器才能通过Secure Shell(SSH)或Telnet运行命令。SSH是现今用来访问给定设备的最常见的协议。它提供到设备的安全加密连接。用于访问路由器CLI的一些常用终端监控器包括：

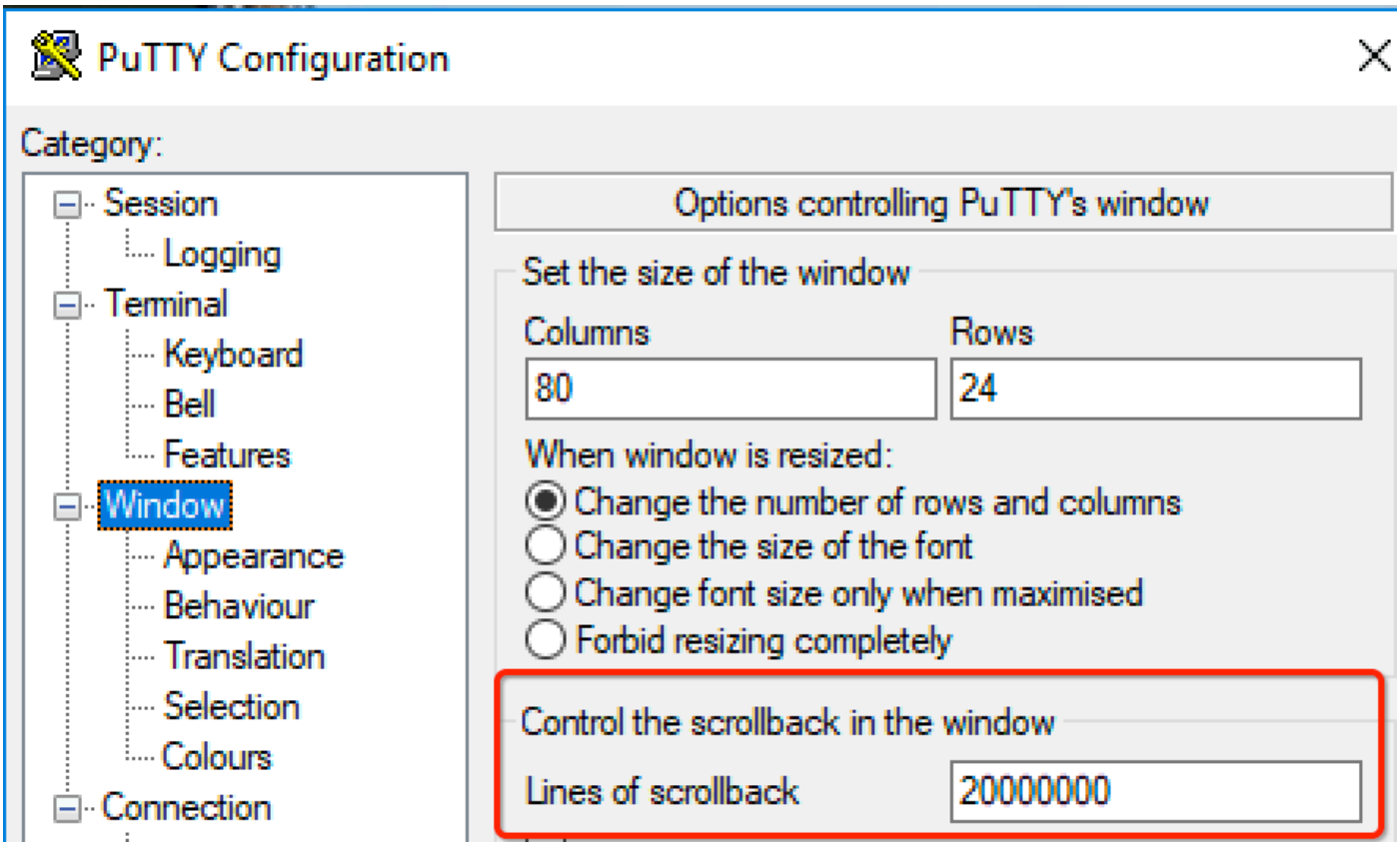


如何设置终端监控器以收集show命令或调试

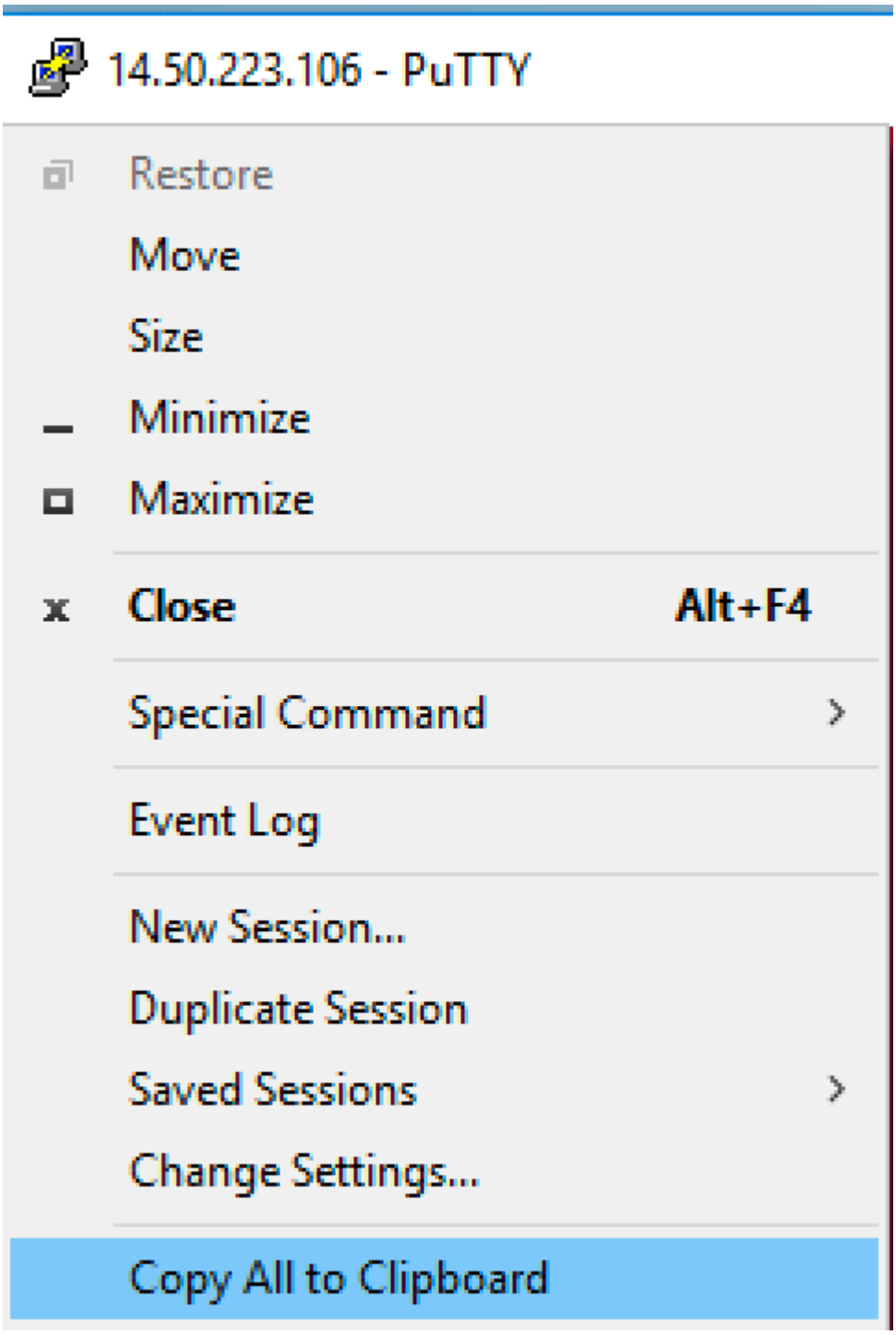
收集CLI的输出有多种方法。建议从路由器的CLI将信息导出到单独的文件中。这样可以更轻松地与外部各方共享信息。

收集设备输出的方法有以下几种：

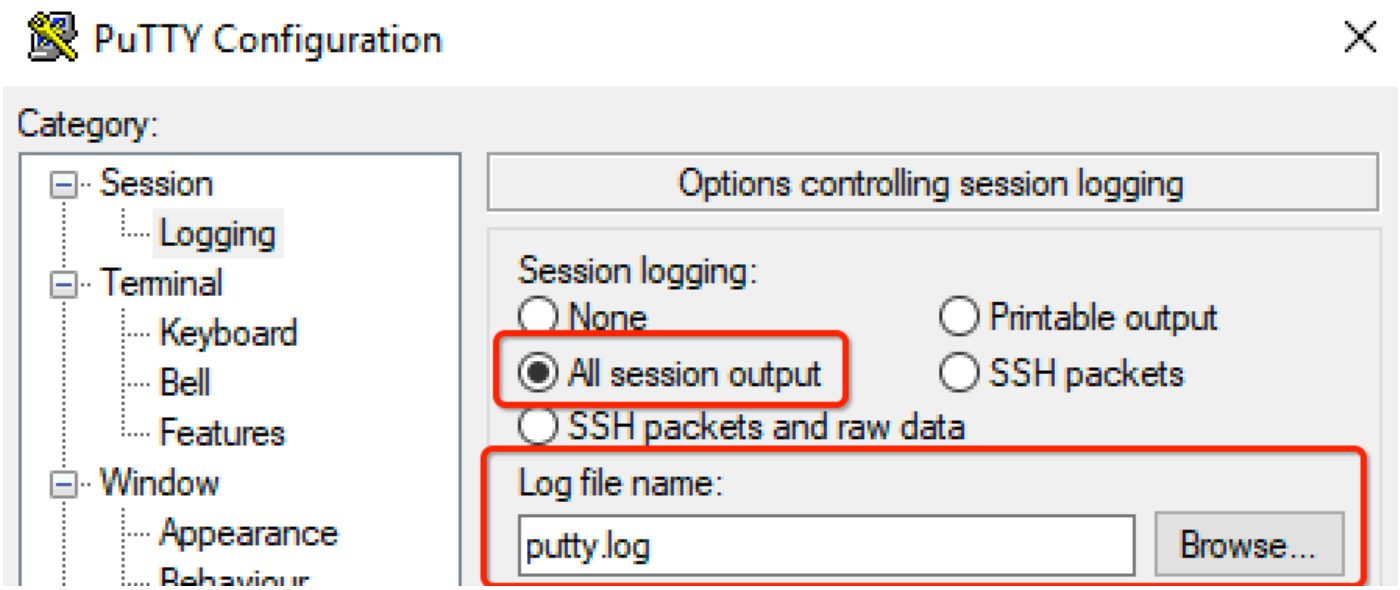
- 转储终端中的所有输出。您需要确保有足够的回滚行，否则，回滚将忽略输出的前几个部分，并且数据可能不完整。要在Putty中增加滚动行，请导航到Putty Configuration > Window > Lines of Scrollback。通常，该值设置为非常高的值，以便获得足够的回滚输出：



稍后，您可以使用“将全部复制到剪贴板”选项从终端监视器收集信息并将输出粘贴到文本文件中：



- 另一个选项是将整个会话输出记录到.txt文件中。使用此选项，输入的所有命令和收集的输出将立即记录到文本文件中。这是一种在会话中记录所有输出的常见做法。要将所有会话输出记录到Putty中的文件，请导航到Putty Configuration > Session > Logging，然后选择All Session Output，如下所示：



注意：如果未指定其他名称，则使用默认日志文件名称。单击Browse按钮可准确了解文件的保存位置，以便稍后查找。此外，请确保不要覆盖同一文件路径中的另一个putty.log文件。

从CLI收集基本show命令输出

在进行任何debug收集之前，需要使用show命令从路由器收集基本信息。Show命令可以快速收集，而且大部分不会对路由器的性能产生任何影响。可以通过运行show命令立即开始隔离问题。

连接到路由器后，终端长度可以设置为0。这样可以更快地收集以同时显示所有输出，并避免使用空格键。用于收集路由器上详细信息的命令是show tech，或者，您可以收集show tech voice，它显示路由器上启用的语音功能的更多具体数据：

<#root>

```
Router#

terminal length 0


Router#

show tech


!or
Router#

show tech voice
```

```
Router#

terminal default length

 !This cmd restores the terminal length to default
```

从CLI收集调试输出

Cisco IOS/IOS XE中的调试输出收集有时可能是一个挑战，因为存在路由器崩溃的风险。下面几节将介绍一些最佳实践，以避免其中一些问题。

内存检查

启用任何调试之前，必须确保有足够的内存将输出存储在缓冲区中。

运行show process memory命令以确定可以分配多少内存来记录缓冲区中的所有输出：

 提示：使用命令terminal length default或terminal length返回终端中显示的有限行数。

```
<#root>
Router#

show process memory


Processor Pool Total: 8122836952 Used: 456568400

Free: 7666268552


lsmpi_io Pool Total: 6295128 Used: 6294296 Free: 832
```

在本例中，有7666268552字节(7.6 GB)可供路由器使用。此内存由路由器在所有系统进程之间共享。这意味着您不能使用所有可用的空闲内存将输出记录到缓冲区中，但您可以根据需要使用大量系统内存。

大多数情况至少需要10 MB才能收集足够的调试输出，然后输出才会丢失或被覆盖。在极少数情况下，需要收集更多数据。在这些情况下，您可以在缓冲区中接收50 MB到100 MB的输出，或者如果有可用内存，则您可以提高输出大小。

如果可用内存不足，则可能存在内存泄漏问题。如果出现这种情况，请让架构TAC团队修正导致内存不足的原因。

中央处理器(CPU)检查

CPU受系统中处于活动状态的进程、功能和呼叫数量的影响。系统中激活的功能或呼叫越多，CPU就越繁忙。

一个好的基准是确保路由器的CPU使用率不超过30%，这意味着您可以安全地启用从基本到高级的调试（使用高级调试时始终关注CPU）。 如果路由器CPU的占用率大约为50%，则可以运行基本调试，但始终要监控CPU。如果CPU使用率超过80%，请立即停止调试（如本文稍后所示），并联系TAC寻求帮助。

运行show process cpu sorted | exclude 0.00命令查看最后5秒、60秒和5分钟的CPU值以及排名靠前的进程。

<#root>

```
Router#

show processes cpu sorted | exclude 0.00


CPU utilization for

five seconds: 1%/0%; one minute: 0%; five minutes: 0%


PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
211 4852758 228862580 21 0.15% 0.06% 0.07% 0 IPAM Manager
84 3410372 32046994 106 0.07% 0.04% 0.05% 0 IOSD ipc task
202 3856334 114790390 33 0.07% 0.05% 0.05% 0 VRRS Main thread
```

在输出中，路由器没有太多活动，CPU使用率低，可以安全地启用调试。

⚠ 警告：密切注意处于活动状态的顶级CPU进程。如果CPU为50%或更高，且顶部进程仅是语音进程，则可以启用基本调试。使用命令持续监控CPU，以确保路由器的整体性能不受影响。

当前激活的呼叫检查

每台路由器具有不同的容量阈值。检查路由器中有多少呼叫处于活动状态，以确保其未接近最大容量，这一点非常重要。Cisco Unified Border Element Version 12 Data Sheet提供了有关每个平台容量的信息以供参考。

运行show call active total-calls命令，了解系统中处于活动状态的呼叫数：

<#root>

```
Router#

show call active total-calls


Total Number of Active Calls : 0
```

运行show call active voice summary命令以接收有关处于活动状态的呼叫类型的更多详细信息：

<#root>

```
Router#

show call active voice summary


Telephony call-legs: 0
SIP call-legs: 0
H323 call-legs: 0
Call agent controlled call-legs: 0
SCCP call-legs: 0
STCAPP call-legs: 0
Multicast call-legs: 0
Total call-legs: 0
```

一些常见值包括：

- 电话呼叫段：TDM网关呼叫，包括模拟呼叫和PRI/ISDN呼叫。
- SIP呼叫段：SIP呼叫总数；如果这是CUBE路由器，则显示每个呼叫两条呼叫段。将总呼叫数除以2可得到准确号码。
- H323呼叫段：H323呼叫总数。
- SCCP呼叫段：Cisco Unified Communications Manager(CUCM)受控媒体资源在路由器中使用，如转码器和MTP。

日志记录缓冲区设置

要将路由器配置为在缓冲区中存储调试输出，需要进入configure terminal模式以手动调整CLI中的设置。此配置对路由器没有影响，但是，如前文部分所示，如果必须回滚配置中的事件，则需要在路由器上执行show tech或show running-config命令。

这是一个配置示例，它是TAC工程师使用的通用基线。这将分配10 MB的缓冲内存，但可以根据需要增加：

```
<#root>
#
configure terminal


service timestamps debug datetime msec localtime show-timezone year


service timestamps log datetime msec localtime show-timezone year


service sequence-numbers


logging buffered 10000000


no logging console
```

```
no logging monitor
```

```
logging queue-limit 10000
```

```
logging rate-limit 10000
```

```
voice iec syslog
```

这些命令可完成以下任务：

- service timestamps debug or log:确保本地路由器时间以毫秒的精度写入每个记录的消息。这对于根据时间查找呼叫非常有用。毫秒时间戳，当两行在同一毫秒内发生时，允许您将调试行分组为逻辑相关事件。
- service sequence-numbers:在行中写入调试的序列号。当日志被转发到系统日志服务器时，此功能非常有用（必需）。这对于确定网络中是否丢弃了任何Syslog服务器的调试消息非常有用。序列号是调试中在时间戳和日志消息之前的第一项。这不同于系统日志服务器可以在其文件中本地写入的时间戳/序列号。
- 日志缓冲器:告知路由器将调试发送到本地缓冲内存。缓冲区大小以字节为单位设置。在配置中，缓冲区大小设置为10 MB。
- no logging console和no logging monitor:控制台或终端监控器中不打印日志消息。如果未配置这些命令，则会损害路由器的性能和调试输出的准确性。
- 语音iec系统日志：启用语音内部错误代码消息以确定断开原因。

配置系统日志设置

有时，问题可能是随机的，并且需要一种持续收集调试直到事件发生的方法。在缓冲区中存储调试时，它会持续收集调试信息。限制为可分配的内存量，一旦达到该内存量，缓冲区就会循环并丢弃最早的消息，从而导致隔离问题所需的有价值信息不完整。

使用syslog，路由器可以将所有调试消息发送到外部服务器，系统日志服务器软件将外部服务器存储在文本文件中。虽然这是收集debug输出的好方法，但它不是日志收集的首选方法。由于服务器拥塞，系统日志服务器倾向于跳过或丢弃已接收输出的行。因为调试输出可能使服务器不堪重负

，或者数据包可能由于网络状况而丢弃。但是，在某些情况下，系统日志是在问题上取得进展的唯一方法。

如果可能，请使用可靠的传输方法（如TCP）以避免信息丢失，作为建议，请将系统日志服务器连接到路由器所连接的同一台交换机上，或者尽可能靠近路由器。这不能保证所有数据都存储在文件中，但会降低数据丢失的几率。

默认情况下，系统日志服务器使用UDP作为端口514上的传输协议：

```
<#root>

#

configure terminal



service timestamps debug datetime msec localtime show-timezone year



service timestamps log datetime msec localtime show-timezone year



service sequence-numbers



!Optional in case you still want to store debug output in the buffer.

logging buffered 10000000



no logging console



no logging monitor



logging trap debugging
```

```
!Replace the 192.168.1.2 with the actual Syslog Server IP Address

logging host 192.168.1.2 transport [tcp|udp] port
```

配置命令后，路由器会立即将消息转发到syslog服务器IP地址。

调试收集

启用调试后，必须先清除缓冲区，然后才能重现问题。这样做是为了确保输出尽可能干净，并避免分析时不需要的其他数据。运行clear log命令，因为这样可以确保清除缓冲区。如果路由器中有其他呼叫处于活动状态并且启用了调试，输出将立即显示在缓冲区中：

<#root>

```
Router#

clear log


Clear logging buffer [confirm]
Router#
```

重现问题后，立即禁用调试以停止缓冲区中的其他输出，然后收集日志。您可以使用以下命令转储终端中的所有输出：

<#root>

```
Router#

undebug all


Router#

terminal length 0


Router#
```

```
show log
```

有时PuTTY会关闭，因为它无法同时处理所有输出。这是正常现象，并不意味着发生了故障。如果发生这种情况，请重新打开会话，然后正常继续。在日志记录缓冲区过大或终端监控程序由于必须打印的数据量而崩溃的情况下，请使用show log将缓冲区输出直接复制到外部设备 | redirect命令：

<#root>

Router#

show log | redirect ftp://username:password@192.168.1.2/debugs.txt

---

此命令将整个缓冲区输出复制到的FTP IP地址：192.168.1.2，文件名debug.txt。必须始终指定文件名。可用于导出数据的其他目标包括：

<#root>

Router#

sh log | redirect ?

```
bootflash: Uniform Resource Locator
flash: Uniform Resource Locator
ftp: Uniform Resource Locator
harddisk: Uniform Resource Locator
http: Uniform Resource Locator
https: Uniform Resource Locator
nvram: Uniform Resource Locator
tftp: Uniform Resource Locator
```

可在语音路由器中启用哪些调试

每个呼叫流程和功能类型（TDM、CUBE或SCCP媒体资源）不同，需要启用特定调试。必须同时启用所有需要的调试。如果一次仅捕获一个调试，则它无效，并在分析数据时造成更多混乱。

在CLI执行提示级别Router#中启用调试，该级别要求您具有特权执行模式权限。

有基本和高级调试，基本调试用于收集SIP、H323或MGCP中的信令信息。这显示了路由器与其对等设备的会话。

高级调试非常详细，用于在内部堆栈错误无法显示基本调试时收集其他信息。这些调试通常是CPU密集型的。

 提示：启用调试后，请记住运行clear logging命令。此命令清除缓冲区，以生成更干净的调试输出。

内部呼叫控制API(CCAPI)调试

在每个Cisco IOS/IOS XE路由器中，都有一个呼叫控制API，负责不同VoIP应用或协议之间的通信。数据平面组件（如RTP、DSP、语音卡等）捕获来自此层的数据。可以使用一个特定的调试：

<#root>

```
debug voip ccapi inout
```

此调试还有其他选项；但是，运行debug voip ccapi inout命令会涵盖所有基本拨号方案和呼叫建立信息，这些信息通常足以了解该层的状态。

 提示：debug voip ccapi inout通常对路由器的CPU影响极小，建议与任何信令调试一起启用，以便提供包含呼叫及其不同状态信息的完整日志集。

SIP呼叫流

这些调试通常用于SIP呼叫流，可以在CUBE和TDM网关内启用，并在路由器和CUCM或任何其他SIP服务器或代理之间使用SIP支路。

基本SIP调试

<#root>

```
debug ccsip messages
```

```
debug ccsip error
```

```
debug ccsip non-call
 !Optional, applies for SIP OPTIONS and SIP REGISTER Messages.
```

高级SIP调试

<#root>

```
debug ccsip all
```

```
debug ccsip verbose
```

```
debug voice ccapi inout
```

数字(PRI、BRI)呼叫流

以下调试适用于主速率接口(PRI)T1/E1或基本速率接口(BRI):

基本数字调试

<#root>

```
debug isdn q931
```

高级数字调试

<#root>

debug isdn q921

模拟呼叫流

当模拟电路涉及外部交换用户(FXS)或外部交换局(FXO)端口时，使用以下调试：

<#root>

debug vpm signal

debug voip vtsp all

MGCP呼叫流

当MGCP用作语音网关和CUCM之间的语音协议时，使用这些调试。

基本调试

<#root>

debug mgcp packets

debug mgcp errors

CCM-Manager调试

ccm-manager调试用于跟踪配置下载以及CUCM和语音网关之间的MoH和PRI/BRI回传消息。这些调试根据需要使用，并且取决于故障场景：

<#root>

debug ccm-manager backhaul

 !For PRI and BRI Deployments

debug ccm-manager errors

debug ccm-manager events

debug ccm-manager config-download

 !Troubleshoot Configuration download issues from CUCM TFTP

debug ccm-mananger music-on-hold

 !Troubleshoot internal MoH Process

高级MGCP调试

<#root>

debug mgcp all

H323呼叫流

虽然H323并未得到广泛使用，但仍有一些配置了H323的部署：

基本H323调试

<#root>

debug h225 asn1

debug h245 asn1

debug h225 events

debug h245 events

高级H323调试

<#root>

debug cch323 h225

debug cch323 h245

debug cch323 all

SCCP媒体资源

这些调试用于解决涉及CUCM服务器注册的媒体终端点(MTP)或转码器的SCCP媒体资源问题：

基本SCCP调试

<#root>

debug sccp messages

```
debug sccp events
```

```
debug sccp errors
```

高级SCCP调试

<#root>

```
debug sccp all
```

VoIP跟踪

随着Cisco IOS XE 17.4.1和17.3.2的推出，在思科统一边界元素(CUBE)内有一个捕获语音日志的新选项。 这一新功能称为VoIP跟踪。这是一个新的可维护性框架，用于记录SIP信令和事件，无需启用任何调试。

默认情况下，VoIP跟踪处于启用状态，可以根据需要随时禁用。VoIP跟踪仅捕获SIP呼叫的特定信息：

- SIP错误
- 拨号对等体匹配
- 分配的RTP端口
- 与SIP信令相关的IEC错误
- SIP中继到中继呼叫的SIP消息
- 有限状态机(FSM)状态和事件
- 从SIP层到CUBE中其他层的事件和API调用
- 呼叫控制（由CUBE处理的统一通信呼叫流）

限制

VoIP跟踪不会记录与对话外SIP消息相关的信息：

- 注册
- 选项
- 订阅/通知
- 信息

支持HA中的VoIP跟踪，但以下警告适用：

- 备用路由器默认启用VoIP跟踪。在备用进程变为活动状态之前，只显示适用的跟踪信息。
- 一旦待机处于活动状态，它就不会包含来自检查点调用的完整跟踪，并且只包含新呼叫。
- show voip trace <key>在备用路由器上仍然工作并显示呼叫的覆盖缓冲区和媒体流数据。

如何启用VoIP跟踪

如前所述，此功能默认启用。要启用此功能，请运行以下命令：

<#root>

```
Router#

configuration terminal


Router(config)#

voice service voip


Router(conf-voi-serv)#

trace

Router(conf-serv-trace)#
```

如何禁用VoIP跟踪

要禁用此功能，请运行以下命令：

<#root>

```
Router(conf-serv-trace)#
```

```
no trace


!or
Router(conf-serv-trace)#

shutdown
```

⚠ 警告：禁用VoIP Trace后，将清除所有内存并丢失信息。

跟踪配置模式中可用的命令包括：

<#root>

```
Router(conf-serv-trace)#

?


default       Set a command to its defaults
  exit          Exit from voice service voip trace mode
  memory-limit  Set limit based on memory used
  no            Negate a command or set its defaults
  shutdown      Shut Voip Trace debugging
```

配置内存限制

内存限制决定了VoIP跟踪用于存储数据的内存大小。默认情况下，它是平台中可用内存的10%，但可以将其调整为最大1 GB，最小10 MB。内存是动态分配的，因为该功能仅在需要时使用内存，并且取决于呼叫量。达到最大可用内存后，它会绕着原有条目转圈并删除旧条目。

当内存限制被修改为大于10%的可用内存时，CLI上会显示一条消息：

<#root>

```
Router(conf-serv-trace)#

memory-limit 1000
```

```
Warning: Setting memory limit more than 10% of available platform memory (166 MB) will affect system pe
```

要将默认内存使用率设置为10%，请运行memory-limit platform命令：

<#root>

```
Router(conf-serv-trace)#
```

memory-limit platform

```
 Reducing the memory-limit clears all VoIP Trace statistics and data.
 If you wish to copy this data first, enter 'no' to cancel,
 otherwise enter 'yes' to proceed. Continue? [no]:
```

⚠ 警告：当内存限制降低时，所有VoIP跟踪数据都将丢失。必须先收集数据的备份，然后才能减少内存。

如何显示VoIP跟踪数据

要显示VoIP跟踪的数据，必须使用特定的show命令。数据可以在同一终端会话中显示，也可以通过syslog发送到出厂设置的syslog服务器。

注意：从收到呼叫的BYE时起，32秒后转储跟踪。

SIP信令按支路显示，不会像常规调试那样组合。常规调试（例如debug CCSIP消息）以事件发生的确切顺序显示呼叫的SIP信令。在VoIP跟踪中，每个支路是独立的，要确定正确的顺序，需要使用时间戳。

要查看可用数据，请运行以下命令：

<#root>

```
Router#
```

show voip trace ?

```
  all             Display all VoIP Traces
  call-id         Filter traces based on Internal Call Id
  correlator      Filter traces based on FPI Correlator
  cover-buffers   Display the summary of all cover buffers
  session-id      Filter traces based on SIP Session ID
  sip-call-id     Filter traces based on SIP Call Id
  statistics      Display statistics for VoIP Trace
```

Show VoIP Trace All

此命令显示缓冲区中可用的所有VoIP跟踪数据。使用此命令会影响路由器的性能。输入命令后，如果继续，将显示警告消息以警告风险：

<#root>

```
Router#
```

show voip trace all

```
 Displaying 11858 cover buffers
 This may severely impact system performance.
 Continue? [yes/no] no
```

Show VoIP Trace Cover-Buffers

此命令显示VoIP跟踪下报告的所有呼叫的呼叫详细信息的概述。每个呼叫段都创建了包含已记录呼叫摘要的封面缓冲区：

<#root>

```
Router#
```

show voip trace cover-buffers

```
------------------ Cover Buffer ---------------
Search-key = 8845:3002:659
Timestamp = *Sep 30 01:17:33.615
Buffer-Id = 1
CallID = 659
Peer-CallID = 661
Correlator = 4
Called-Number = 3002
Calling-Number = 8845
SIP CallID = 20857880-1ec12085-13b930-411b300a@10.48.27.65
```

```
SIP Session ID = 2b1289c400105000a0002c3ecf872659
GUID = 208578800000
-----------------------------------------------

------------------ Cover Buffer ---------------
Search-key = 8845:3002:661
Timestamp = *Sep 30 01:17:33.634
Buffer-Id = 2
CallID = 661
Peer-CallID = 659
Correlator = 4
Called-Number = 3002
Calling-Number = 8845
SIP CallID = 8D6DEC28-1F111EB-829FD797-1B22F6DB@10.48.55.11
SIP Session ID = 0927767800105000a0005006ab805584
GUID = 208578800000
-----------------------------------------------
```

有关每个字段的详细信息，请参阅下表：

字段	描述
Search-Key	包含呼叫、被叫号码和呼叫ID的组合
时间戳	覆盖缓冲区的创建时间
缓冲区ID	覆盖缓冲区的缓冲区ID
呼叫ID	到覆盖缓冲区的各个呼叫支路的呼叫ID
对等呼叫ID	对等分支的呼叫ID
相关器	呼叫的FPI相关器
被叫号码	覆盖缓冲区的各个呼叫段的被叫号码
主叫号码	覆盖缓冲区的各个呼叫支路的呼叫号码
SIP呼叫ID	覆盖缓冲区的各个呼叫段的SIP呼叫ID
Sip会话ID	覆盖缓冲区的各个呼叫段的SIP会话ID

GUID	封面缓冲区的相应调用的GUID
锚脚	如果各自的呼叫段是呼叫分流流或媒体代理部署中的锚段，则锚段设置为“是”
分叉腿	如果相应的呼叫分支是呼叫分支流或媒体代理部署中的锚点，则分叉支路设置为“是”
关联的Call-ID	关联的分支腿的呼叫ID

要过滤盖缓冲区，请运行include和section命令：

<#root>

```
Router#

show voip trace cover-buffers | include Search-key | 8845 | 3002


Search-key = 8845:3002:661
!or
Router#

show voip trace cover-buffers | section Search-key | 8845 | 3002


Search-key = 8845:3002:661
```

Show VoIP Trace Call-Id

与前面的命令结合使用，show voip trace call-id可用于查找呼叫。确定call-id后，可以运行下一个命令以显示有关特定呼叫段的所有信息：

<#root>

```
Router#

show voip trace cover-buffers | include Search-key | 8845 | 3002


Search-key = 8845:3002:661
Router# show voip trace call-id 661
```

显示VoIP跟踪统计信息

此show命令显示有关状态、内存消耗、错误/故障调用、成功调用、最新和最早条目的时间戳等信息的详细输出：

<#root>

```
Router#

show voip trace statistics


VoIP Trace Statistics
Tracing status              : ENABLED at *Sep 12 06:44:02.349
Memory limit configured     : 803209216 bytes
Memory consumed             : 254550928 bytes (31%)
Total call legs dumped      : 2
Oldest trace dumped         : *Sep 12 07:29:21.077 Search-key: 9898:30000:64
Latest trace dumped         : *Sep 12 07:29:21.010 Search-key: 9898:30000:63
Total call legs captured    : 11858
Total call legs available   : 11858
Oldest trace available      : *Sep 12 06:57:23.923, Search-key: 5250001:4720001:11
Latest trace available      : *Sep 13 05:08:25.353, Search-key: 19074502232:30000:13177
Total traces missed         : 0
```

有关每个字段的详细信息，请参阅下表：

字段	描述
跟踪状态	显示跟踪状态，包括启用VoIP跟踪的时间和日期。
配置的内存限制	显示配置的内存限制。这是处理器池内存大小的10%。
消耗的内存	显示VoIP跟踪动态消耗的内存量。
转储的呼叫段总数	显示转储到日志记录缓冲区的失败呼叫段数。转储呼叫是指与IEC错误相关的呼叫段。
已转储最早的跟踪	显示自启用VoIP跟踪以来最旧的失败呼叫的时间戳和搜索密钥。
最新跟踪已转储	显示自启用VoIP跟踪以来的最新失败呼叫的时间戳和搜索密钥。

捕获的呼叫段总数	显示启用VoIP跟踪后捕获的总支路。
可用呼叫段总数	显示历史记录中可用的呼叫段总数。与捕获的总呼叫段相比，这可以相同或不同，这取决于内存限制。
可用的最早跟踪	显示内存中可用的最旧覆盖缓冲区的时间戳和搜索键。
最新跟踪可用	显示内存中可用的最新封面缓冲区的时间戳和搜索键。
丢失的跟踪总数	显示由于内存限制而错过的呼叫段数。

其他Show命令

字段	使用率	描述
show voip trace correlator <correlator>	show voip trace correlator 4	过滤并显示从覆盖缓冲区开始的特定呼叫ID的VOIP跟踪。
show voip trace session-id <session-id>	show voip trace session-id 87003120822b5dbd8fd80f62d8e57c48	根据SIP会话ID过滤并显示呼叫的VOIP跟踪。SIP消息的会话ID报头中的本地或远程UUID可用于显示呼叫的两段。
show voip trace sip-call-id <call-id>	show voip trace sip-call-id 01e60dfa9d8442848336d79e3155a8a1	根据SIP Call-ID过滤和显示VOIP跟踪。

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）。