Unified Border Element(CUBE)および時分割多重(TDM)ゲートウェイのデバッグコレクションの設定

内容

概要

このドキュメントでは、Cisco IOS/IOS-XE音声ルータで音声デバッグを収集するためのベストプラクティスについて説明します。

前提条件

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

要件

- サービス統合型ルータ(ISR)内のCisco IOS/IOS-XEに関する基礎知識。
- ISRルータでコマンドを実行するための特権アクセス。
- Voice-over-IP(VoIP)プロトコルに関する経験が必要です。
- VoIPトレースには、Cisco IOS-XE 17.4.1または17.3.2以上が必要です。

使用するコンポーネント

このドキュメントでは、次のコンポーネントを使用します。

- Cisco ISR 3925
- Cisco ISR 4451
- PuTTY

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景

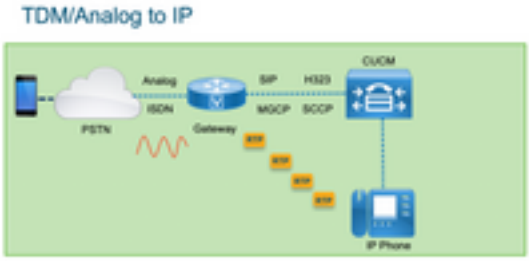
これらのプラットフォームでのデバッグ収集のプロセスには課題があり、デバイスのパフォーマンスに影響する可能性があります。音声ルータで複数のアクティブコールが確立されると、課題

とリスクが増大します。一部のシナリオでは、デバッグが正しく収集されない場合、CPUの使用率が高くなり、ルータのキャパシティが低下し、ソフトウェアクラッシュが発生する可能性があります。このドキュメントでは、Cisco Unified Border Element(CUBE)とTDM/アナログゲートウェイの違いについて説明します。

TDM音声ゲートウェイとCUBE

TDM音声ゲートウェイは主に、内部電話システムを別の構内交換機(PBX)または公衆電話交換網(PSTN)と相互接続するために使用されます。 TDMゲートウェイで使用される接続のタイプは、T1/E1コントローラ（ISDNまたはCAS）、およびFXSポートやFXOポートなどのアナログ回線です。Digital Signal Processor（DSP；デジタル信号プロセッサ）は、オーディオをそのraw形式からRTPパケットに変換します。同様に、DSPがRTPパケットを処理した後、RTPパケットは生の音声に変換され、その音声が特定の回線に送信されます。これらのゲートウェイは、VoIP側ではH323、MGCPまたはSCCPと相互運用でき、TDM側では、PSTNまたはエンドポイントへの最も一般的な接続としてISDN PRI回線またはアナログと相互運用できます。

図に示すように、TDMゲートウェイは、内部VoIPインフラストラクチャとアナログまたはISDNサービスプロバイダー間のブリッジを提供します。



VoIPの導入により、お客様はレガシーシステムを最新のVoIPインフラストラクチャに迅速に変更し始めました。サービスプロバイダー側でも同じことが起こり、接続を使用してオンプレミステレフォニーサービスとサービスプロバイダーのVoIPインフラストラクチャを相互接続し、より優れたサービスを提供するために機能を拡張しています。現在最も一般的に使用されているVoIPプロトコルはSession Initiation Protocol(SIP)で、現在は世界中のお客様やInternet Telephony Service Provider(ITSP)で広く使用されています。

CUBEは、プライマリVoIPプロトコルとしてSIPを使用するITSPを介して、これらの内部VoIPシステムを外部ネットワークに相互接続する方法を提供するために導入されました。CUBEは単なるIP-IPゲートウェイであり、T1/E1コントローラやアナログポートのようなTDMタイプの接続は必要ありません。CUBEは、TDMゲートウェイと同じプラットフォームで動作します。

最も一般的に使用されるVoIPプロトコルは、コールの確立とティアダウン用のSIPと、メディア転送用のRTPです。CUBEでは、トランスコーダが必要でない限り、DSPは必要ありません。RTPトラフィックフローはITSPからエンドポイントまでエンドツーエンドで流れ、CUBEはアドレス隠蔽を備えた中間者として機能し、多くの機能の1つとして提供されます。

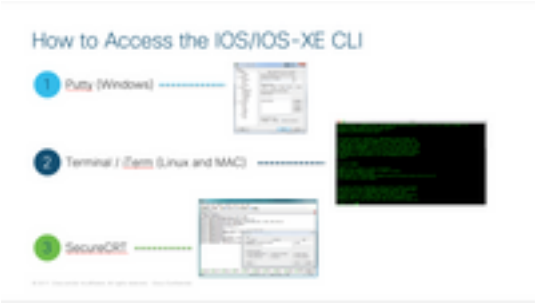
図に示すように、CUBEは内部VoIPインフラストラクチャとSIP ITSPの間の分割を提供します。

Cisco IOS/IOS-XE音声デバッグの収集

音声機能は、ISR、ASR、CAT8Kなどの異なるプラットフォームのリストで動作しますが、共通のソフトウェアであるCisco IOSまたはCisco IOS-XEのいずれかを使用します（Cisco IOSとCisco IOS-XEの違いについてはこの記事では説明しません）。 まず、Cisco IOSルータへのアクセス方法の基本から説明します。

コマンドラインインターフェイス(CLI)からCisco IOS/IOS-XEルータにアクセスする方法

ルータは、他のCLIベースのデバイスと同様に、Secure Shell(SSH)またはTelnetを介してコマンドを実行するには、ターミナルモニタにアクセスする必要があります。SSHは、デバイスへの安全で暗号化された接続を提供するため、現在デバイスにアクセスするために使用されている最も一般的なプロトコルです。ルータのCLIへのアクセスに使用される一般的な端末モニタには、次のものがあります。

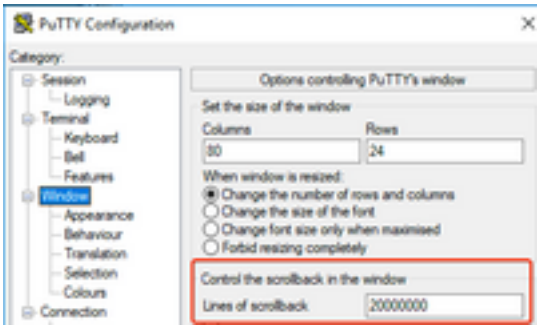


showコマンドまたはデバッグを収集するようにターミナルモニタを設定する方法

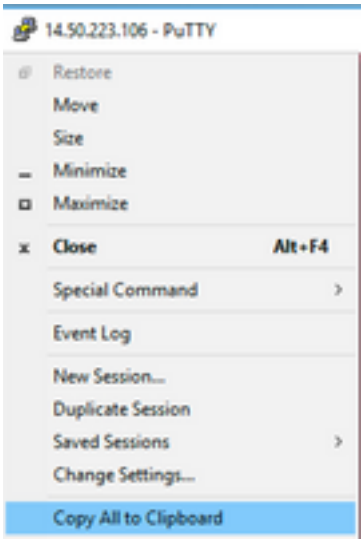
CLIから出力を収集するには、さまざまな方法があります。情報をルータのCLIから別のファイルにエクスポートすることを推奨します。これにより、情報を外部の関係者と共有しやすくなります。

デバイスからの出力を収集する方法は、次のとおりです。

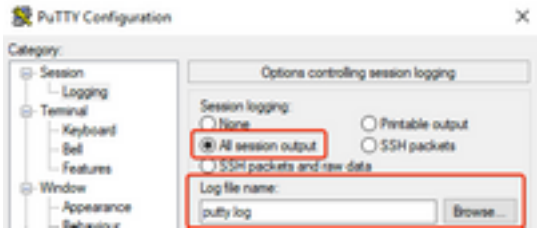
- 端末ですべての出力をダンプします。そのためには、十分な数のscrollback行があることを確認する必要があります。そうしないと、scrollbackが出力の最初のセクションを失い、データが不完全になる可能性があります。Puttyでscrollback行を増やすには、[Putty Configuration] > [Window] > [Lines of Scrollback]に移動します。通常、scrollbackの出力を十分に得るために、この値は非常に高い値に設定されます。



後で、[Copy All to Clipboard] オプションを使用してターミナルモニタから情報を収集し、出力をテキストファイルに貼り付けることができます。



- もう1つのオプションは、セッション出力全体を.txtファイルに記録することです。このオプションを使用すると、入力されたすべてのコマンドと収集された出力が即座にテキストファイルに記録されます。これは、セッションのすべての出力を記録するための一般的な方法です。Puttyのファイルにすべてのセッション出力を記録するには、[Putty Configuration] > [Session] > [Logging] に移動し、次のように[All Session Output]を選択します。



> **注**：他の名前が指定されていない場合は、デフォルトのログファイル名が使用されます。[Browse]ボタンをクリックして、後でファイルを検索するためにファイルの保存場所を正確に確認します。また、同じファイルパスにある別のputty.logファイルを上書きしないでください。

CLIからの基本的なshowコマンド出力の収集

debug収集を実行する前にルータから基本情報を収集するには、showコマンドが必要です。showコマンドは収集が速く、ほとんどの場合、ルータのパフォーマンスには影響しません。問題の切り分けは、showコマンドの出力だけで即座に開始できます。

ルータに接続すると、端末の長さを0に設定できます。これにより、すべての出力を一度に表示し

、スペースバーを使用しないようにするために、収集を高速化できます。ルータに関する詳細情報を収集する1つのコマンドは「show tech」です。または、ルータで有効になっている音声機能に固有のデータを表示するshow tech voiceを収集することもできます。

```
Router# terminal length 0
Router# show tech
!or
Router# show tech voice
Router# terminal default length !This cmd restores the terminal length to default
```

CLIからのデバッグ出力の収集

Cisco IOS/IOS-XEのデバッグ出力の収集は、ルータのクラッシュのリスクがあるため、困難な場合があります。ただし、問題を回避するために、次のセクションでベストプラクティスの一部を説明します。

メモリチェック

デバッグを有効にする前に、出力をバッファに格納するのに十分なメモリがあることを確認する必要があります。

show process memoryコマンドを実行して、バッファ内のすべての出力を記録するために割り当てることができるメモリの量を調べます。

> ヒント：コマンドterminal length defaultまたはterminal length <num_lines>を使用して、端末に表示される限られた数の回線に戻ります。

```
Router# show process memory
Processor Pool Total: 8122836952 Used: 456568400 Free: 7666268552
lsmpi_io Pool Total: 6295128 Used: 6294296 Free: 832
```

この例では、ルータが使用できる空き容量は7666268552バイト(7.6 GB)です。このメモリは、すべてのシステムプロセス間でルータによって共有されます。つまり、バッファの出力をログに記録するために空きメモリ全体を使用することはできませんが、必要に応じて十分な量のシステムメモリを使用することは可能です。

ほとんどのシナリオでは、十分なデバッグ出力を収集するために少なくとも10 MBが必要で、出力が失われたり上書きされたりします。まれに、より多くのデータを収集する必要がある場合があります。このような特定のシナリオでは、バッファに50 MB ~ 100 MB相当の出力を取得するか、メモリが使用可能な限り高い値を取得できます。

空きメモリが少ない場合は、メモリリークの問題が発生している可能性があります。その場合は、アーキテクチャTACチームに連絡して、このようなメモリ不足の原因を修正してください。

中央処理装置(CPU)チェック

CPUは、システムでアクティブなプロセス、機能、およびコールの量に影響されます。システムでアクティブな機能やコールが多いほど、CPUの使用率が高くなります。

適切なベンチマークは、ルータのCPU使用率を30 %以下にすることです。つまり、基本デバッグから高度デバッグまで安全に有効にできます（高度なデバッグを使用する場合は常にCPUを監視

します）。 ルータのCPU使用率が約50 %の場合は、基本的なデバッグを実行して、CPUを注意深く監視できます。CPU使用率が80 %を超えた場合は、すぐにデバッグを停止し（この記事の後半で説明）、TACにサポートを依頼してください。

show process cpu sorted | exclude 0.00コマンドを使用して、上位プロセスと共に最後の5秒間、60秒間、および5分間のCPU値を確認します。

```
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
```

出力では、ルータのアクティビティは多くなく、CPUは低く、デバッグを安全に有効にできます。

> **注意**：CPUが50 %以上で、トッププロセスが音声プロセスの場合は、基本的なデバッグだけを有効にできる場合は、アクティブな上位CPUプロセスに特に注意してください。コマンドを使用してCPUを継続的に監視し、ルータの全体的なパフォーマンスが影響を受けないことを確認します。

現在のアクティブコールの確認

各ルータのキャパシティしきい値はそれぞれ異なります。ルータ内でアクティブなコールの数を確認し、最大容量に近づかないようにすることが重要です。Cisco Unified Border Elementバージョン12データシートには、各プラットフォームのキャパシティに関する参照情報が記載されています。

show call active total-callsコマンドを使用して、システム内でアクティブなコールの数を把握します。

```
Router# show call active total-calls
Total Number of Active Calls : 0
```

アクティブな特定のコールタイプの詳細情報を取得するには、**show call active voice summary**コマンドを使用します。

```
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
```

一般的な値には次のものがあります。

- **テレフォニーコールレッグ**:TDMゲートウェイコール（アナログおよびPRI/ISDNコールを含む）
- **SIPコールレッグ**:SIPコールの合計。これがCUBEルータの場合、コールごとに2つのコール

レッグが表示されます。正確な数を得るには、ここに示すコールの合計を2で割ります。
- **H323コールレッグ**:H323コールの合計。
- **SCCPコールレッグ**:トランスコーダやMTPなど、ルータで使用されるCUCM制御メディアリソース。

ロギングバッファ設定

デバッグ出力をバッファに保存するようにルータを設定するには、configure terminalモードに入り、CLIの設定を手動で調整します。この設定はルータには影響しませんが、前のセクションに示すように、設定をロールバックする必要がある場合には、ルータからの**show tech**または**show running-config**コマンドが必要です。

次に、TACエンジニアが使用する一般的なベースラインである設定例を示します。この例では10 MBのバッファメモリを割り当てますが、必要に応じて増やすことができます。

```
# 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
```

これらのコマンドは、次のタスクを実行します。

- **service timestamps debug or log**:ローカルルータの時間が、ミリ秒単位の精度で、ログに記録されたすべてのメッセージに書き込まれるようにします。これは、時間に基づいてコールを検索するのに便利です。ミリ秒のタイムスタンプを使用すると、同じミリ秒以内に2つの回線が発生した場合に、デバッグ回線を論理的な関連イベントにグループ化できます。
- **service sequence-numbers:**デバッグのシーケンス番号を行に書き込みます。これは、ログがsyslogサーバに転送される場合に便利です（基本的に必須）。これは、syslogサーバへのデバッグメッセージがネットワークでドロップされたかどうかを識別するために非常に便利です。シーケンス番号は、タイムスタンプと実際のログメッセージの前の、デバッグの最初の項目です。これは、syslogサーバがファイルにローカルで書き込むことができるタイムスタンプ/シーケンス番号とは異なることに注意してください。
- **logging buffer**:ローカルバッファメモリにデバッグを送信するようにルータに指示します。バッファサイズはバイト単位で設定されます。設定では、バッファサイズは10 MBに設定されています。
- **no logging console**および**no logging monitor:**コンソールまたは端末モニタにログメッセージが表示されない。これらのコマンドが設定されていない場合、ルータのパフォーマンスとデバッグ出力の精度が低下する可能性があります。
- **voice iec syslog:**Voice Internal Error Codes（Vエラーコード）メッセージを有効にして、接続解除の原因を特定します。

syslog 設定の編集

問題がランダムに発生し、イベントが発生するまでデバッグを継続的に収集する方法が必要になる場合があります。バッファにデバッグを保存すると、デバッグが継続的に収集されます。割り

当て可能なメモリ量に制限され、そのメモリ量に達すると、バッファは最も古いメッセージを循環して廃棄します。これにより、問題の切り分けに必要な不完全で貴重な情報が発生します。

Syslogを使用すると、ルータはすべてのデバッグメッセージを外部サーバに送信できます。外部サーバでは、Syslogサーバソフトウェアがデバッグメッセージをテキストファイルに保存します。これはデバッグ出力を収集する優れた方法ですが、ログ収集には適していません。Syslogサーバでは、サーバでの輻輳が原因で、受信した出力の行がスキップされたり、廃棄されたりすることが多くあります。これは、デバッグ出力がサーバに過大な負荷をかけたり、ネットワークの状態が原因でパケットが廃棄されたりするためです。ただし、一部のシナリオでは、Syslogが問題を進行させる唯一の方法です。

可能であれば、TCPなどの信頼性の高い転送方式を使用して情報の損失を回避し、提案として、Syslogサーバをルータが接続されているのと同じスイッチ、またはルータにできるだけ近い場所に接続します。すべてのデータがファイルに保存されるとは限りませんが、データが失われる可能性は低くなります。

デフォルトでは、syslogサーバはポート514のトランスポートプロトコルとしてUDPを使用します。

```
#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
```

コマンドが設定されるとすぐに、ルータはメッセージをSyslogサーバのIPアドレスに即座に転送します。

コレクションのデバッグ

デバッグを有効にした後、問題が再現される前にバッファをクリアする必要があります。これは、出力が可能な限りクリーンであり、分析に必要のない余分なデータを避けるために行われます。clear logコマンドを実行して、バッファがクリアされるようにします。ルータで他のコールがアクティブになっていて、デバッグが有効になっている場合、出力はバッファに即座に出力されます。

```
Router# clear log
Clear logging buffer [confirm]
Router#
```

問題が再現されたら、デバッグをただちに無効にして、バッファ内の出力を停止します。次に、

ログを収集します。端末のすべての出力は、次のコマンドでダンプできます。

```
Router# undebug all
Router# terminal length 0
Router# show log
```

すべての出力を一度に処理する必要がないため、PuTTYが終了する場合があります。これは正常な動作であり、障害が発生したことを意味するものではありません。この状態が発生した場合は、セッションを再度開き、正常に続行します。ロギングバッファが大きすぎたり、出力する必要があるデータ量が原因で端末モニタがクラッシュする場合は、**show log**コマンドを使用して、バッファ出力を外部デバイスに直接コピーします **|リダイレクト**:

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

このコマンドは、バッファ出力全体をIPアドレス192.168.1.2のFTP (ファイル名debug.txt) にコピーします。ファイル名は常に指定する必要があります。データを工クスポートできるその他の宛先は次のとおりです。

```
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
```

音声ルータで有効にできるデバッグはどれか？

コールフローと機能のタイプ(TDM、CUBE、またはSCCP (メディアリソース))はそれぞれ異なり、有効にできる特定のデバッグがあります。必要なすべてのデバッグを同時に有効にする必要があります。一度に1つのデバッグだけをキャプチャした場合は効果がなく、データの分析時に混乱が生じます。

デバッグはCLI execプロンプトレベル**Router#**内で有効にします。このレベルでは、特権のある実行モード権限が必要です。

基本的なデバッグと高度なデバッグがあります。基本的なデバッグは、SIP、H323、またはMGCPのいずれかでシグナリング情報を収集するために使用され、ルータとピアデバイスとの間の通信を示します。

高度なデバッグは非常に詳細で、通常は、基本的なデバッグでは表示できない内部スタック工ラーが発生した場合に詳細情報を収集するために使用されます。これらのデバッグは通常、CPUに負荷がかかります。

> **ヒント**：デバッグをイネーブルにした後、**clear logging**コマンドを実行することを忘れないでください。このコマンドにより、デバッグのキャプチャをクリアするためにバッファがクリアされます。

内部コール制御API(CCAPI)のデバッグ

各Cisco IOS/IOS-XEルータ内には、異なるVoIPアプリケーション（プロトコル）間の通信を担当するコール制御APIと、RTP、DSP、音声カードなどのデータプレーンコンポーネントがあります。この層からデータをキャプチャするには、使用できる特定のデバッグがあります。

debug voip ccapi inout

このデバッグには他のオプションもありますが、debug voip ccapi inoutには、すべての基本的なダイヤルプランとコール確立情報が含まれており、これは通常、このレイヤの状態を理解するのに十分な情報です。

> ヒント：debug voip ccapi inoutは通常、ルータのCPUへの影響が最小限で、コールとさまざまな状態の情報を含む完全なログセットを提供するために、シグナリングデバッグとともにイネーブルにすることをお勧めします。

SIPコールフロー

これらのデバッグは、SIPコールフローで最も一般的に使用され、ルータとCUCMまたは他の任意のSIPサーバ/プロキシ間のSIPレッグを使用して、CUBEおよびTDMゲートウェイ内で有効にできます。

基本的なSIPデバッグ

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

高度なSIPデバッグ

debug ccsip all
debug ccsip verbose
debug voice ccapi inout

デジタル(PRI、BRI)コールフロー

次のデバッグは、一次群速度インターフェイス(PRI)T1/E1または基本速度インターフェイス(BRI)に適用されます。

基本的なデジタルデバッグ

debug isdn q931

高度なデジタルデバッグ

debug isdn q921

アナログコールフロー

次のデバッグは、Foreign eXchange Subscriber(FXS)またはForeign eXchange Office(FXO)ポートなどのアナログ回線が関係する場合に使用されます。

```
debug vpm signal
debug voip vtsp all
```

MGCPコールフロー

これらのデバッグは、音声ゲートウェイとCUCM間の音声プロトコルとしてMGCPが使用されている場合に使用されます。

基本的なデバッグ

```
debug mgcp packets
debug mgcp errors
```

CCMマネージャのデバッグ

debug ccm-managerは、CUCMと音声ゲートウェイ間の設定のダウンロード、MoHおよびPRI/BRIバックホールメッセージを追跡するために使用されます。これらのデバッグは必要に応じて使用され、障害のシナリオによって異なります。

```
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
```

高度なMGCPデバッグ

```
debug mgcp all
```

H323コールフロー

H323は広く使用されていませんが、H323が設定された導入は依然として存在します。

基本的なH323デバッグ

```
debug h225 asn1
debug h245 asn1
debug h225 events
debug h245 events
```

高度なH323デバッグ

```
debug cch323 h225
debug cch323 h245
debug cch323 all
```

SCCPメディアリソース

次のデバッグは、メディアターミネーションポイント(MTP)またはCisco Unified Communications Manager(CUCM)サーバに登録されたトランスコーダが関係するSkinny Call Control Protocol(SCCP)メディアリソースの問題をトラブルシューティングするために使用されます。

基本的なSCCPデバッグ

```
debug sccp messages
debug sccp events
debug sccp errors
```

高度なSCCPデバッグ

```
debug sccp all
```

VoIPトレース

Cisco IOS-XE 17.4.1および17.3.2の導入により、Cisco Unified Border Element(CUBE)内で音声ログをキャプチャする新しいオプションが追加されました。 この新機能はVoIPトレースと呼ばれます。これは、デバッグを有効にせずにSIPシグナリングとイベントを記録するために作成された新しいサービスアビリティフレームワークです。

VoIPトレースはデフォルトで有効になっており、必要に応じていつでも無効にできます。VoIPトレースは、SIPコールの特定の情報のみをキャプチャします。

- SIPトランクからトランクコールへのSIPメッセージ
- SIPレイヤからCUBEの他のレイヤへのイベントおよびAPIコール
- SIPエラー
- コール制御（CUBEで処理されるユニファイドコミュニケーションのコールフロー）
- 有限状態マシン(FSM)の状態とイベント
- ダイヤルピアの照合
- 割り当てられたRTPポート
- IECエラーとSIPシグナリングの相関関係

制約事項

- VoIPトレースは、ダイアログ外のSIPメッセージに関連する情報を記録しません。REGISTEROPTIONS登録/通知INFO
- HAでのVoIPトレースはサポートされていますが、次の注意事項が適用されます。 スタンバイルータでは、VoIPトレースがデフォルトで有効になっています。スタンバイプロセスに適用可能なトレースだけが、アクティブになるまで表示されますスタンバイがアクティブになると、チェックポイントされたコールからの完全なトレースは**含まれず**、新しいコールだけが含まれます**show voip trace <key>**はスタンバイルータ上で引き続き動作し、コールのカバーバッファとメディアストリームデータを表示します

VoIPトレースを有効にする方法

すでに説明したように、この機能はデフォルトで有効になっています。この機能を有効にするコマンドは次のとおりです。

```
Router# configuration terminal
Router(config)# voice service voip
Router(conf-voi-serv)# trace
Router(conf-serv-trace)#
```

VoIPトレースを無効にする方法

この機能を無効にするには、次のコマンドを使用します。

```
Router(conf-serv-trace)# no trace
!or
Router(conf-serv-trace)# shutdown
```

注意：VoIPトレースを無効にすると、すべてのメモリがクリアされ、情報が失われます。

トレースコンフィギュレーションモードで使用できるコマンドは次のとおりです。

```
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
```

メモリ制限の設定

memory-limitは、データを保存するためにVoIPトレースが使用するメモリの量を決定します。デフォルトでは、プラットフォームで使用可能なメモリの10 %ですが、最大1 GB、最小10 MBに変更できます。動的に割り当てられるメモリ。つまり、この機能は必要に応じてメモリのみを使用し、コールの量に依存します。使用可能な最大メモリに達すると、古い工ントリが循環して削除されます。

メモリの制限が10 %の使用可能なメモリよりも大きくなるように変更されると、コマンドラインインターフェイス(CLI)にメッセージが表示されます。

```
Router(conf-serv-trace)# memory-limit 1000
Warning: Setting memory limit more than 10% of available platform memory (166 MB) will affect
system performance.
```

デフォルトのメモリ使用率を10 %に設定するには、コマンドmemory-limit platformを使用できます。

```
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]:
```

注意：メモリの制限を減らすと、すべてのVoIPトレースデータが失われます。メモリを削減する前に、データのバックアップを収集する必要があります。

VoIPトレースデータの表示方法

VoIPトレースからのデータを表示するには、特定のshowコマンドを使用する必要があります。データは同じターミナルセッションで表示することも、Syslog経由でオフボックスsyslogサーバに送信することもできます。

注：トレースは、コールに対してBYEを受信してから32秒後にダンプされます。

注：SIPシグナリングはレッグごとに表示され、通常のデバッグのように組み合わされることはありません。**debug ccsip messages**などの通常のデバッグでは、イベントが発生した正確な順序でコールのSIPシグナリングが表示されます。VoIPトレースでは、各レッグは別々に扱われます。正しい順序を決定するために、タイムスタンプが使用されます。

データの表示に使用できるコマンドは次のとおりです。

```
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

このコマンドは、バッファで使用可能なすべてのVoIPトレースデータを表示します。このコマンドの使用は、ルータのパフォーマンスに影響します。コマンドを入力すると、警告メッセージが表示され、リスクに関する警告が表示され、続行を確認します。

```
Router# show voip trace all
 Displaying 11858 cover buffers
 This may severely impact system performance.
 Continue? [yes/no] no
```

show voip trace cover-buffers

このコマンドは、VoIPトレースで報告されたすべてのコールの詳細なコールの概要を表示します。各コールレッグには、ロギングされたコールの要約を含むカバーバッファが作成されます。

```
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
-----------------------------------------------
```

各フィールドの詳細については、次の表を参照してください。

フィールド	説明
検索キー	発信者番号、着信者番号、およびコールIDの組み合わせを含む
タイムスタンプ	カバーバッファの作成時間
バッファID	カバーバッファのバッファID
コールID	カバーバッファへのそれぞれのコールレッグのコールID
ピアコールID	ピアレッグのコールID
相関器	コールのFPI相関子
Called-number	カバーバッファの各コールレッグの着番号
Calling-number	カバーバッファの各コールレッグの発信番号
SipコールID	カバーバッファの各コールレッグのSIPコールID
SipセッションID	カバーバッファの各コールレッグのSIPセッションID
GUID	カバーバッファの各コールのGUID
アンカー脚	それぞれのコールレッグがコール分岐フローまたはメディアプロキシ導入のアンカーレッグである場合、アンカーレッグはyesに設定
ニ股の脚	各コールレッグがコールフォーキングフローまたはメディアプロキシ導入のアンカーレッグである場合、[Forked Leg]は[yes]に設定さ
関連付けられたCall ID	関連するフォークされたレッグのコールID

カバーバッファをフィルタリングするには、include コマンドとsection コマンドを使用できます。

```
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

前のコマンドと組み合わせて、show voip trace call-idを使用してコールを検索できます。コールIDが特定されたら、次のコマンドを使用して、特定のコールレッグに関するすべての情報を表示できます。

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

show voip trace statistics

このshowコマンドは、ステータス、メモリ消費、エラーまたは障害コール、成功したコール、最新および最も古いエントリのタイムスタンプなどに関する詳細な出力を表示します。

```
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
```

各フィールドの詳細については、次の表を参照してください。

フィールド	説明
トレース状態	VoIPトレースが有効になった時刻と日付を含む、トレースのステータスを表示します。
メモリ制限が設定されました	設定されているメモリ制限を表示します。これは、プロセッサプールのメモリサイズの10 %です
消費メモリ	VoIPトレース用に動的に消費されるメモリ量を表示します。
ダンプされたコールレッグの総数	ロギングバッファにダンプされた失敗したコールレッグの数を表示します。ダンプされたコールとは、IEC工ラーに関連するコールレします。
ダンプされた最も古いトレース	VoIPトレースが有効になってから最も古い失敗したコールのタイムスタンプと検索キーを表示します。
最新のトレースのダンプ	VoIPトレースが有効になってから最後に失敗したコールのタイムスタンプと検索キーを表示します。
キャプチャされたコールレッグの総数	VoIPトレースが有効になった後にキャプチャされたレッグの合計を表示します
使用可能なコールレッグの総数	履歴で使用可能なコールレッグの合計を表示します。これは、メモリ制限に応じて、プチャされた合計コールレッグと同じか、異なる場合があります。
使用可能な最も古いトレース	メモリで使用可能な最も古いカバーバッファのタイムスタンプと検索キーを表示しま
使用可能な最新のトレース	メモリで使用可能な最新のカバーバッファのタイムスタンプと検索キーを表示しま
欠落したトレースの総数	メモリの制限によって失われたコールレッグの数を表示します。

その他の show コマンド

フィールド	用途	
show voip trace correlator	show voip trace correlator 4	カバーバッファから始まる特定のコールIDのVoIPトレースを
show voip trace session-id <session-id>	show voip trace session-id 87003120822b5dbd8fd80f62d8e57c48	SIPセッションIDに基づいて、コール
show voip trace sip-call-id <call-id>	show voip trace sip-call-id 01e60dfa9d8442848336d79e3155a8a1	SIPコールIDに基づいてVoIPトレース

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。