



## SDP 帯域幅フィールド機能

Cisco Unified Border Element (SP Edition) は、統合モデルで、帯域幅フィールド インターワーキング機能と、メディア帯域幅割り当てに帯域幅フィールドではなく CODEC を使用するオプション機能をサポートしています。

Session Initiation Protocol (SIP; セッション開始プロトコル) メッセージ交換で、Cisco Unified Border Element (SP Edition) は、メディア ピンホール帯域幅の計算に Session Description Protocol (SDP; セッション記述プロトコル) 帯域幅フィールド (b-line) で定義されたパラメータを使用します。SIP メッセージ交換中に、SDP には帯域幅フィールドと Coder/Decoder (CODEC) 情報が含まれています。この場合、Cisco Unified Border Element (SP Edition) は帯域幅フィールド値を使用してメディア ピンホール用に十分な帯域幅を割り当てます。

配置時に、b-line を使用するのではなくセッション記述プロトコル (SDP) メッセージ内の CODEC 定義を使用して、メディア ピンホール帯域幅を設定したほうがよいエンドポイントがある場合もあります。

メディア帯域幅割り当てに帯域幅フィールドではなく CODEC を使用するオプション機能により、Call Admission Control (CAC; コール アドミッション制御) ポリシー エントリ内にメディア帯域幅フラグを設定して、メディア ピンホール帯域幅の計算で b-line を無視して CODEC を使用することができるようになります。

Cisco Unified Border Element (SP Edition) では、Application Specific Maximum (AS) および Transport Independent Application Specific Maximum (TIAS) 変換を使用した帯域幅行パススルーにより、発信側と着信側に送信されたアウトバウンドセッション記述プロトコル (SDP) での帯域幅行の変換方法を決定できる能力をサポートすることで、帯域幅フィールド インターワーキングをサポートしています。



(注)

Cisco IOS XE Release 2.4 以降では、この機能は統合モデルだけでサポートされます。

Cisco Unified Border Element (SP Edition) は、以前は Integrated Session Border Controller と呼ばれており、このマニュアルでは通常 Session Border Controller (SBC; セッション ボーダー コントローラ) と呼びます。

本章で使用されているコマンドの詳細な説明については、次の場所にある『*Cisco Unified Border Element (SP Edition) Command Reference: Unified Model*』を参照してください。

[http://www.cisco.com/en/US/docs/ios/sbc/command/reference/sbcu\\_book.html](http://www.cisco.com/en/US/docs/ios/sbc/command/reference/sbcu_book.html)

すべての Cisco IOS コマンドの詳細については、<http://tools.cisco.com/Support/CLILookup> にある Command Lookup Tool を使用するか、Cisco IOS マスター コマンドリストを参照してください。

## SDP 帯域幅フィールド機能の機能履歴

リリース	変更内容
Cisco IOS XE Release 2.4	メディア帯域幅割り当てに帯域幅フィールドではなく CODEC を使用するオプション機能は、統合モデルとともに Cisco IOS XR に追加されました。
Cisco IOS XE Release 2.5	帯域幅フィールド インターワーキング機能が Cisco IOS XR に追加されました。
Cisco IOS XE Release 3.1S	メディア帯域幅ポリシー機能が追加されました。
Cisco IOS XE Release 3.2S	隣接単位のコーデック ストリング インターワーキング機能が追加されました。

## 内容

この章の内容は、次のとおりです。

- ・「SDP 帯域幅フィールド機能の機能履歴」(P.388)
- ・「メディア帯域幅割り当てに帯域幅フィールドではなく CODEC を使用するオプション機能」(P.388)
- ・「メディア帯域幅割り当てに帯域幅フィールドではなく CODEC を使用するオプションの設定例」(P.393)
- ・「メディア帯域幅ポリシーについて」(P.394)
- ・「メディア帯域幅ポリシーの設定」(P.395)
- ・「帯域幅フィールド インターワーキング機能」(P.403)
- ・「帯域幅フィールドのインターワーキングの設定例」(P.408)
- ・「隣接単位のコーデック ストリング インターワーキング」(P.409)

## SDP 帯域幅フィールド機能の機能履歴

SDP 帯域幅フィールド機能を実装するためには、次の前提条件を満たす必要があります。

SDP 帯域幅フィールド機能を実装する前に、Cisco Unified Border Element (SP Edition) がすでに設定されている必要があります。

## メディア帯域幅割り当てに帯域幅フィールドではなく CODEC を使用するオプション機能

次の項では、「メディア帯域幅割り当てに帯域幅フィールドではなく CODEC を使用するオプション」機能について説明しています。

- ・「SIP コール内の帯域幅計算について」(P.389)
- ・「メディア帯域幅割り当てに帯域幅フィールドではなく CODEC を使用するオプション機能の設定」(P.389)

- 「メディア帯域幅割り当てに帯域幅フィールドではなく CODEC を使用するオプションの設定例」(P.393)

## SIP コール内の帯域幅計算について

SBC では、コール内の各メディア ストリームを分析して、必要な帯域幅を計算します。SDP を含む SIP コールに対して、SBC は `b=CT`、`b=AS`、または `b=TIAS` 行を検索します。これらの行が存在する場合、メディア ストリームに必要なベースライン帯域幅を計算するためにこれらの行が使用されます。

これらの行が存在しない場合、SBC は、ストリーム内で可能性のある各 CODEC を検査して、これらに基づいてベースライン帯域幅を計算することによって、ベースライン帯域幅を計算します。CODEC ごとに割り当てられる帯域幅を設定できます。

次に、SBC が、必要なパケット化および Real Time Control Protocol (RTCP) 帯域幅オーバーヘッドを考慮するようにベースライン帯域幅を調整します。

たとえば次のような特定のエンドポイントは、SBC がメディア ストリーム用に計算した帯域幅の要求に従いません。

- コールの帯域幅の再ネゴシエーションを開始するエンドポイントは、再ネゴシエーションが完了する前に追加の帯域幅を使用して開始することができます。
- 暗号化に必要な追加のペイロード サイズを考慮しないため、`b-line` を使用するセキュア メディアの帯域幅を間違えて要求するエンドポイント。
- 帯域幅の要求の計算時にすべての形式を考慮せずに、複数の形式のデータを並列に送信するエンドポイント（単一のストリーム内にある異なるペイロードを使用する高精細度ビデオと低精細度ビデオなど）。

パケットの廃棄なしでこれらのエンドポイントの相互運用性を可能にするために、SBC では CODEC 単位の帯域幅を設定できます。これにより、SBC 管理者は、これらのエンドポイントでサポートされている最適な最大値で CODEC をサポートできるようになります。該当するエンドポイントで既知の CODEC セットを使用しているため、これで十分です。

ただし、エンドポイントに明示的な帯域幅 (`b-`) 行が含まれている場合、SBC は最大値ではなくその帯域幅行を使用して帯域幅を計算します。メディア帯域幅割り当てに帯域幅フィールドではなく CODEC を使用するオプション機能では、`media bandwidth-fields ignore` コマンドを使用して、コールアドミッション制御 (CAC) ポリシー エントリ内にメディア帯域幅フラグを設定して、帯域幅の計算で `b-line` を無視して CODEC を使用します。

## メディア帯域幅割り当てに帯域幅フィールドではなく CODEC を使用するオプション機能の設定

この作業は、メディア帯域幅割り当てに帯域幅フィールドではなく CODEC を使用するオプション機能を設定します。

### 手順の概要

1. `configure terminal`
2. `sbc service-name`
3. `sbc`
4. `cac-policy-set policy-set-id`
5. `first-cac-table table-name`

6. **cac-table** *table-name*
7. **table-type** {*policy-set* | **limit** {*list of limit tables*}}
8. **entry** *entry-id*
9. **cac-scope** {*list of scope options*}
10. **media bandwidth-fields ignore**
11. **action** [*next-table goto-table-name* | **cac-complete**]
12. **exit**
13. **exit**
14. **complete**

## 手順の詳細

	コマンドまたはアクション	目的
ステップ1	<b>configure terminal</b>  例: Router# configure	グローバル コンフィギュレーション モードをイネーブルにします。
ステップ2	<b>sbc service-name</b>  例: Router(config)# sbc mysbc	SBC サービスのモードを開始します。  • <i>service-name</i> 引数を使用して、サービスの名前を定義します。
ステップ3	<b>sbe</b>  例: Router(config-sbc)# sbe	SBC サービス内で SBE エンティティのモードを開始します。
ステップ4	<b>cac-policy-set policy-set-id</b>  例: Router(config-sbc-sbe)# cac-policy-set 1	SBE エンティティ内で CAC ポリシー セット コンフィギュレーション モードを開始して、必要に応じて新規ポリシー セットを作成します。
ステップ5	<b>first-cac-table table-name</b>  例: Router(config-sbc-sbe-cacpolicy)# first-cac-table StandardListByAccount	ポリシーのアドミSSION コントロール ステージの実行時に処理する最初のポリシー テーブル名を設定します。
ステップ6	<b>cac-table table-name</b>  例: Router(config-sbc-sbe-cacpolicy)# cac-table StandardListByAccount	SBE ポリシー セットのコンテキスト内で、アドミSSION コントロール テーブル (必要に応じて作成します) のコンフィギュレーション モードを開始します。

コマンドまたはアクション	目的
<p><b>ステップ7</b> <code>table-type {policy-set   limit {list of limit tables}}</code></p> <p><b>例:</b> Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set</p>	<p>SBC ポリシー セットのコンテキスト内で CAC テーブルのテーブル タイプを設定します。</p> <p><i>list of limit tables</i> 引数は、テーブル内にあるエントリの <i>match-value</i> フィールドの構文を制御します。使用可能な制限テーブルは次のとおりです。</p> <ul style="list-style-type: none"> <li>• <i>account</i> : アカウント名を比較します。</li> <li>• <i>adj-group</i> : 隣接グループ名を比較します。</li> <li>• <i>adjacency</i> : 隣接名を比較します。</li> <li>• <i>all</i> : 比較タイプはありません。すべてのイベントがこのタイプと一致します。</li> <li>• <i>call-priority</i> : コール プライオリティと比較します。</li> <li>• <i>category</i> : 番号分析が割り当てられたカテゴリを比較します。</li> <li>• <i>dst-account</i> : 宛先アカウント名を比較します。</li> <li>• <i>dst-adj-group</i> : 宛先隣接グループ名を比較します。</li> <li>• <i>dst-adjacency</i> : 宛先隣接名を比較します。</li> <li>• <i>dst-prefix</i> : 着信ディジット スtring の先頭を比較します。</li> <li>• <i>event-type</i> : CAC ポリシー イベント タイプと比較します。</li> <li>• <i>src-account</i> : 送信元アカウント名を比較します。</li> <li>• <i>src-adj-group</i> : 送信元隣接グループ名を比較します。</li> <li>• <i>src-adjacency</i> : 送信元隣接名を比較します。</li> <li>• <i>src-prefix</i> : 発番号 String の先頭を比較します。</li> </ul> <p>機能は、個々の隣接ごとに行う場合と同じように、CAC 設定から隣接グループごとにイネーブルまたはディセーブルにできます。adj-group テーブル タイプは、送信元または宛先の隣接グループで一致します。</p> <p>policy-set キーワードが指定されている場合、cac-scope コマンドを使用して、CAC ポリシー セット テーブル内で制限が適用される、各エントリ内の範囲を設定します。</p>
<p><b>ステップ8</b> <code>entry entry-id</code></p> <p><b>例:</b> Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</p>	<p>アドミッション コントロール テーブル内のエントリを作成または変更するモードを開始します。</p>

コマンドまたはアクション	目的
<p>ステップ9 <code>cac-scope {list of scope options}</code></p> <p><b>例 :</b>  Router(config-sbc-sbe-cacpolicy-cactable-entry)  # cac-scope src-adjacency</p>	<p>CAC の制限がポリシー セット テーブルの各エントリ内で適用される範囲を選択します。</p> <ul style="list-style-type: none"> <li>• <i>list of scope options</i> : イベント照合に使用される次のいずれかのストリングを指定します。 <ul style="list-style-type: none"> <li>– <b>account</b> : 同じアカウントからのイベント。</li> <li>– <b>adjacency</b> : 同じ隣接からのイベント。</li> <li>– <b>adj-group</b> : 同じ隣接グループのメンバーからのイベント。</li> <li>– <b>call</b> : スコープの限界は単一コール単位です。</li> <li>– <b>category</b> : 同じカテゴリを持つイベント。</li> <li>– <b>dst-account</b> : 同じアカウントに送信されるイベント。</li> <li>– <b>dst-adj-group</b> : 同じ隣接グループに送信されるイベント。</li> <li>– <b>dst-adjacency</b> : 同じ隣接に送信されるイベント。</li> <li>– <b>dst-number</b> : 同じ宛先を持つイベント。</li> <li>– <b>global</b> : スコープの限界はグローバルです。</li> <li>– <b>src-account</b> : 同じアカウントからのイベント。</li> <li>– <b>src-adj-group</b> : 同じ隣接グループからのイベント。</li> <li>– <b>src-adjacency</b> : 同じ隣接からのイベント。</li> <li>– <b>src-number</b> : 同じ送信元番号を持つイベント。</li> <li>– <b>sub-category</b> : このスコープで指定される制限は、同じ加入者カテゴリのメンバーとの間で送受信されるすべてのイベントに適用されます。</li> <li>– <b>sub-category-pfx</b> : このスコープで指定される制限は、同じ加入者カテゴリ プレフィックスのメンバーとの間で送受信されるすべてのイベントに適用されます。</li> <li>– <b>subscriber</b> : このスコープで指定される制限は、個々の加入者（レジストラ サーバで登録されているデバイス）から送受信されるすべてのイベントに適用されます。</li> </ul> </li> </ul>
<p>ステップ10 <code>media bandwidth-fields ignore</code></p> <p><b>例 :</b>  Router(config-sbc-sbe-cacpolicy-cactable-entry)  # media bandwidth-fields ignore</p>	<p>メディア フラグを設定し、<b>b-line</b> を無視して、CODEC を使用してメディア ストリームに必要なベースライン帯域幅を計算します。</p>

	コマンドまたはアクション	目的
ステップ 11	<pre>action [next-table goto-table-name   cac-complete]  例： Router(config-sbc-sbe-cacpolicy-cactable-entry) # action cac-complete</pre>	<p>アドミッション コントロール テーブルのこのエントリの後で実行するアクションを設定します。指定できるアクションは次のとおりです。</p> <ul style="list-style-type: none"> <li><b>next-table</b> キーワードおよび <i>goto-table-name</i> 引数を使用して、処理する次の CAC テーブルを特定します。</li> <li><b>cac-complete</b> キーワードを使用して、このスコープの処理を停止します。</li> </ul>
ステップ 12	<pre>exit  例： Router(config-sbc-sbe-cacpolicy-cactable-entry) # exit</pre>	<p><b>entry</b> モードを終了して、<b>cactable</b> モードに移行します。</p>
ステップ 13	<pre>exit  例： Router(config-sbc-sbe-cacpolicy-cactable)# exit</pre>	<p><b>cactable</b> モードを終了して、<b>cacpolicy</b> モードに移行します。</p>
ステップ 14	<pre>complete  例： Router(config-sbc-sbe-cacpolicy)# complete</pre>	<p>セット全体をコミットしたら、CAC ポリシー セットを完了します。</p>

## メディア帯域幅割り当てに帯域幅フィールドではなく CODEC を使用するオプションの設定例

次に、メディア フラグを設定し、**b-line** を無視して、CODEC を使用してメディア ストリームに必要なベースライン帯域幅を計算する例を示します。

```
Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# first-cac-table StandardListByAccount
Router(config-sbc-sbe-cacpolicy)# cac-table StandardListByAccount
Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# cac-scope src-adjacency
Router(config-sbc-sbe-cacpolicy-cactable-entry)# media bandwidth-fields ignore
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbe-cacpolicy-cactable)# exit
Router(config-sbc-sbe-cacpolicy-cactable-entry)# complete
```

## メディア帯域幅ポリシーについて

これよりも前のリリースでは、コールアドミッション制御 (CAC) によって許可された帯域幅を超えたコールは SBC によってディセーブルにされるか拒否されていました。しかし、TelePresence などの一部のアプリケーションでは他のオプションが必要です。SBC では、オーディオストリームを変更せずに維持したまま、ビデオストリームの帯域幅を減らす機能が提供されるようになりました。

このリリースでは、帯域幅制限が拡張され、**media police** コマンドを使用して 3 種類のメディア帯域幅のオプションのいずれかを設定できるようになりました。

- strip
- reject
- degrade

これらのオプションは、すべてのメディア タイプか、必要に応じてビデオのみに設定できます。

### strip

個々のメディアストリームがコールの帯域幅の制限を超えると、そのメディアストリームはポートをゼロ (0) に設定することでディセーブルになります。上記の手順が完了した後、残りのすべてのストリームの合計帯域幅がコールに対する帯域幅の制限を超えている場合、要求は拒否されます。

ポートがゼロ (0) に設定されている場合、コールが終了し、画面に次のメッセージが表示されます。

```
incompatible sites
```

### reject

個々のメディアストリームがコールに対する帯域幅の制限を超えると、要求は拒否されます。すべてのメディアストリームの合計帯域幅がコールに対する帯域幅の制限を超えると、要求は拒否されます。

### degrade

メディアストリームがコールに対する帯域幅の制限を超えると、ビデオストリームはコールの帯域幅の制限内でメディアストリームを伝送する低い (ゼロでない) 帯域幅にダウングレードされます。

(注) ビデオストリームだけがダウングレードされます。オーディオストリームはダウングレードされません。オーディオストリームがコールの帯域幅を超えている場合、メディアストリームはダウングレードできません。

### 制約事項

degrade オプションは H.323 コールではサポートされません。

degrade オプションを使用すると、パフォーマンスが 2 ~ 5% 低下することがあります。

### 設定

メディア帯域幅ポリシーを設定するには、**media policy** コマンドを使用してメディアポリシーモードを設定し、**bandwidth** コマンドを使用して、アナログからデジタルへのコーデック (enCOder/DECoder) の最小帯域幅を設定します。



(注)

コーデック名は、SBC が認識できるシステムコーデックの 1 つである必要があります。システムコーデックのリストを確認するには、**show sbc sbc sbe codecs** コマンドを使用します。

**max-bandwidth-per-scope** コマンドは、パケットオーバーヘッドを含むすべての方向のすべてのメディアストリーム用の最大帯域幅制限を指定します。



**bandwidth min** コマンドは単方向の、最小帯域幅制限を指定し、パケット オーバーヘッドは含まれていません。

設定手順については、「メディア帯域幅ポリシーの設定」(P.395) を、設定例については「end」(P.416) を参照してください。

## メディア帯域幅ポリシーの設定

ここでは、次の手順について説明します。

- 「メディア ポリシー モードの設定」(P.395)
- 「コーデックの最小帯域幅の設定」(P.396)
- 「スコープごとの最大帯域幅の設定」(P.398)

## メディア ポリシー モードの設定

メディア ポリシー モードを設定するには、次の手順を実行します。

### 手順の概要

1. **config**
2. **sbc sbc-name**
3. **sbe**
4. **cac-policy-set policy-set-id**
5. **cac-table table-name**
6. **table-type policy-set**
7. **entry entry-id**
8. **media police strip | reject | degrade**
9. **end**

### 手順の詳細

	コマンドまたはアクション	目的
ステップ1	<b>config</b>  例： Router# config	グローバル コンフィギュレーション モードを開始します。
ステップ2	<b>sbc sbc-name</b>  例： Router(config)# sbc SBC1	Cisco Unified Border Element (SP Edition) で SBC サービスを作成し、SBC コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ3	<b>sbe</b>  例： Router(config-sbc)# sbe	SBC の Signaling Border Element (SBE) 機能のモードを開始します。
ステップ4	<b>cac-policy-set</b> <i>policy-set-id</i>  例： Router(config-sbc-sbe)# cac-policy-set 1	SBE エンティティ内で CAC ポリシー セット コンフィギュレーション モードを開始して、必要に応じて新規ポリシー セットを作成します。  <i>policy-set-id</i> : ポリシー セットを特定するためにユーザによって選択される整数。指定できる範囲は 1 ~ 2147483647 です。
ステップ5	<b>cac-table</b> <i>table-name</i>  例： Router(config-sbc-sbe-cacpolicy)# cac-table testSecure	SBE ポリシー セットのコンテキスト内で、アドミッション コントロール テーブル (必要に応じて作成します) のコンフィギュレーション モードを開始します。  <i>table-name</i> : アドミッション コントロール テーブル名。
ステップ6	<b>table-type</b> <i>policy-set</i>  例： Router(config-sbc-sbe-cacpolicy-cactable)# table-type <i>policy-set</i>	SBE ポリシー セットのコンテキスト内で、CAC テーブルのテーブル タイプを設定します。
ステップ7	<b>entry</b> <i>entry-id</i>  例： Router(config-sbc-sbe-cacpolicy-cactable)# entry 1	アドミッション コントロール テーブル内のエントリを変更するモードを開始します。  <i>entry-id</i> : テーブル エントリを指定します。
ステップ8	<b>media police strip   reject   degrade</b>  例： Router(config-sbc-sbe-cacpolicy-cactable-entry)# media police degrade	SBC がメディア コールの帯域幅限界を超えたメディア ストリームを処理する方法を設定します。
ステップ9	<b>end</b>  例： Router(config-sbc-sbe-cacpolicy-cactable-entry)# end	コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

## コーデックの最小帯域幅の設定

コーデックの最小帯域幅を設定するには、次の手順を実行します。

### 手順の概要

1. **config**
2. **sbc** *sbc-name*
3. **sbe**

4. `codec custom custom-name id`
5. `type variable`
6. `media video`
7. `bandwidth min bandwidth-value`
8. `end`

## 手順の詳細

	コマンドまたはアクション	目的
ステップ1	<code>configure terminal</code>  例： Router# <code>config</code>	グローバル コンフィギュレーション モードを開始します。
ステップ2	<code>sbc <i>sbc-name</i></code>  例： Router(config)# <code>sbc SBC1</code>	Cisco Unified Border Element (SP Edition) で SBC サービスを作成し、SBC コンフィギュレーション モードを開始します。
ステップ3	<code>sbe</code>  例： Router(config-sbc)# <code>sbe</code>	SBC の Signaling Border Element (SBE) 機能のモードを開始します。
ステップ4	<code>codec custom <i>custom-name id</i></code>  例： Router (config-sbc-sbe)# <code>codec custom h263-c id 96</code>	アナログからデジタルへのカスタム コーデック (enCOder/DECoder) の名前を指定し、コーデック定義モードを開始します。
ステップ5	<code>type variable</code>  例： Router (config-sbc-sbe-codec-def)# <code>type variable</code>	コーデックのタイプを <code>variable</code> に設定します。
ステップ6	<code>media video</code>  例： Router (config-sbc-sbe-codec-def)# <code>media video</code>	メディア タイプを <code>video</code> に設定します。
ステップ7	<code>bandwidth min <i>bandwidth-value</i></code>  例： Router (config-sbc-sbe-codec-def)# <code>bandwidth min 328000</code>	コーデックの最小帯域幅を設定します。  (注) <code>bandwidth min</code> コマンドは単方向の、最小帯域幅制限を指定し、パケット オーバーヘッドは含まれていません。
ステップ8	<code>end</code>  例： Router(config-sbc-sbe-cacpolicy-cactable-entry)# <code>end</code>	コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

## スコープごとの最大帯域幅の設定

すべてのメディア ストリームの帯域幅制限を設定するには、次の手順を実行します。

### 手順の概要

1. **configure terminal**
2. **sbc *sbc-name***
3. **sbe**
4. **cac-policy-set *policy-set-id***
5. **description *description***
6. **first-cac-table *table-name***
7. **first-cac-scope *scope-name***
8. **cac-table *table-name***
9. **table-type policy-set**
10. **entry *entry-id***
11. **max-bandwidth-per-scope *bandwidth***
12. **action cac-complete**
13. **media police degrade**
14. **complete**
15. **codec system *sys-codec id payload id***
16. **type variable**
17. **bandwidth min *bandwidth-value***
18. **end**

### 手順の詳細

	コマンドまたはアクション	目的
ステップ1	<b>configure terminal</b>  例： Router# config t	グローバル コンフィギュレーション モードを開始します。
ステップ2	<b>sbc <i>sbc-name</i></b>  例： Router(config)# sbc SBC1	Cisco Unified Border Element (SP Edition) で SBC サービスを作成し、SBC コンフィギュレーション モードを開始します。
ステップ3	<b>sbe</b>  例： Router(config-sbc)# sbe	SBC の Signaling Border Element (SBE) 機能のモードを開始します。

	コマンドまたはアクション	目的
ステップ4	<p><code>cac-policy-set policy-set-id</code></p> <p>例： Router(config-sbc-sbe)# cac-policy-set 1</p>	<p>SBE エンティティ内で CAC ポリシー セット コンフィギュレーション モードを開始して、必要に応じて新規ポリシー セットを作成します。</p> <p><i>policy-set-id</i> : ポリシー セットを特定するためにユーザによって選択される整数。指定できる範囲は 1 ~ 2147483647 です。</p>
ステップ5	<p><code>description description</code></p> <p>例： Router(config-sbc-sbe-cacpolicy)# description bandwidth degrade</p>	<p>このポリシー セットの説明テキストを設定します。</p>
ステップ6	<p><code>first-cac-table table-name</code></p> <p>例： Router(config-sbc-sbe-cacpolicy)# first-cac-table my_table</p>	<p>処理する最初のポリシー テーブルの名前を設定します。CAC ポリシーには、設定済みのテーブルが数多くあります。CAC ポリシーのアプリケーションを開始するには、使用される最初のテーブルを定義する必要があります。</p> <p><i>table-name</i> : 最初に処理すべきアドミッション コントロール テーブル。</p>
ステップ7	<p><code>first-cac-scope scope-name</code></p> <p>例： Router(config-sbc-sbe-cacpolicy)# first-cac-scope call</p>	<p>ポリシーのアドミッション コントロール ステージの実行時に制限が最初に定義されるべき範囲を設定します。各 CAC ポリシーには、これに適用される範囲があります。この CAC ポリシーは、コール ベース単位で適用されます。</p> <p><i>scope-name</i> には、次のいずれかの値が指定されます。</p> <ul style="list-style-type: none"> <li>• <b>adj-group</b> : 同じ隣接グループのメンバーからのイベントの制限。</li> <li>• <b>call</b> : 制限が単一コール単位です。</li> <li>• <b>category</b> : カテゴリ単位の制限。</li> <li>• <b>dst-account</b> : 同じアカウントに送信されるイベントの制限。</li> <li>• <b>dst-adj-group</b> : 同じ隣接グループに送信されるイベントの制限。</li> <li>• <b>dst-adjacency</b> : 同じ隣接到送信されるイベントの制限。</li> <li>• <b>dst-number</b> : 同一隣接グループ番号を持つイベントの制限。</li> <li>• <b>global</b> : 制限がグローバルです (他のオプションと組み合わせることができません)。</li> <li>• <b>src-account</b> : 同じアカウントからのイベントの制限。</li> <li>• <b>src-adj-group</b> : 同じ隣接グループからのイベントの制限。</li> <li>• <b>src-adjacency</b> : 同じ隣接からのイベントの制限。</li> <li>• <b>src-number</b> : 同じ送信元番号を持つイベントの制限。</li> </ul>

## メディア帯域幅ポリシーの設定

	コマンドまたはアクション	目的
ステップ 8	<pre>cac-table table-name</pre> <p>例 :</p> <pre>Router(config-sbc-sbe-cacpolicy)# cac-table testSecure</pre>	<p>SBE ポリシー セットのコンテキスト内で、アドミッション コントロール テーブル (必要に応じて作成します) の コンフィギュレーション モードを開始します。</p> <p><i>table-name</i> : アドミッション コントロール テーブル名。</p>
ステップ 9	<pre>table-type {policy-set   limit {list of limit tables}}</pre> <p>例 :</p> <pre>Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set</pre>	<p>SBE ポリシー セットのコンテキスト内で、CAC テーブルの テーブル タイプを設定します。ポリシー セット テーブルでは、イベント、コール、メッセージは、このテーブルのすべてのエントリに適用されます。</p>
ステップ 10	<pre>entry entry-id</pre> <p>例 :</p> <pre>Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</pre>	<p>アドミッション コントロール テーブル内のエントリを変更するモードを開始します。</p> <p><i>entry-id</i> : テーブル エントリを指定します。</p>
ステップ 11	<pre>max-bandwidth-per-scope bandwidth</pre> <p>例 :</p> <pre>Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-bandwidth-per-scope 6000000 bps</pre>	<p>アドミッション コントロール テーブルのエントリの帯域幅の上限を bps、Kbps、Mbps、または Gbps 単位で設定します。</p> <p>(注) <b>max-bandwidth-per-scope</b> コマンドは、パケット オーバーヘッドを含むすべての方向のすべてのメディア ストリーム用の最大帯域幅制限を指定します。</p>
ステップ 12	<pre>action cac-complete</pre> <p>例 :</p> <pre>Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete</pre>	<p>アドミッション コントロール テーブルのこのエントリの後で実行するアクションを設定します。この場合、<b>cac-complete</b> キーワードを使用してこのスコープの処理を停止します。</p>
ステップ 13	<pre>media police strip   reject   degrade</pre> <p>例 :</p> <pre>Router(config-sbc-sbe-cacpolicy-cactable-entry)# media police degrade</pre>	<p>SBC がメディア コールの帯域幅限界を超えたメディア ストリームを処理する方法を設定します。この例では <b>degrade</b> です。</p>
ステップ 14	<pre>complete</pre> <p>例 :</p> <pre>Router(config-sbc-sbe-cacpolicy-cactable-entry) complete</pre>	<p>フルセットの確定後、CAC-policy または call-policy セットを終了します。</p>
ステップ 15	<pre>codec system sys-codec id payload id</pre> <p>例 :</p> <pre>Router(config-sbc-sbe-cacpolicy)# codec system H263 id 34</pre>	<p>既存のコーデックを変更するためにコーデック 定義モードを開始します。</p>

コマンドまたはアクション	目的
ステップ 16 <code>type variable</code>  例： <pre>Router (config-sbc-sbe-codec-def) # type variable</pre>	コーデックのタイプを <code>variable</code> に設定します。
ステップ 17 <code>bandwidth min bandwidth-value</code>  例： <pre>Router (config-sbc-sbe-codec-def) # bandwidth min 370000</pre>	コーデックの最小帯域幅を設定します。  <b>(注)</b> <code>bandwidth min</code> コマンドは単方向の、最小帯域幅制限を指定し、パケット オーバーヘッドは含まれていません。
ステップ 18 <code>end</code>  例： <pre>Router (config-sbc-sbe-cacpolicy-cactable-entry) # end</pre>	コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

## メディア ポリシーの設定例

ここでは、次の設定例について説明します。

- 「メディア ポリシー モードの設定例」(P.402)
- 「コーデックの最小帯域幅の設定例」(P.402)
- 「スコープごとの最大帯域幅の設定例」(P.402)

## メディア ポリシー モードの設定例

次に、要求が帯域幅制限を超えたときにメディア ストリームが低い帯域幅に低下するように SBC を設定する例を示します。

```
Router# config t
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# cac-table cac-tbl-1
Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# media police degrade
Router(config-sbc-sbe-cacpolicy-cactable-entry)#
```

## コーデックの最小帯域幅の設定例

次に、メディア コールの最大帯域幅制限を 400,000 bps に設定する例を示します。

```
Router# config t
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# codec system H263 id 34
Router(config-sbc-sbe-codec-def)# bandwidth 400000
```

次に、特にビデオ タイプ メディア コールの最小帯域幅制限を 328,000 bps に設定する例を示します。

```
Router# config t
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router (config-sbc-sbe)# codec custom h263-c id 96
Router (config-sbc-sbe-codec-def)# type variable
Router (config-sbc-sbe-codec-def)# media video
Router (config-sbc-sbe-codec-def)# bandwidth min 328000
```

## スコープごとの最大帯域幅の設定例

次に、すべてのメディア ストリームの帯域幅制限を設定する例を示します。

```
Router# config t
Router(config)# sbc SBC1
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# description bandwidth degrade
Router(config-sbc-sbe-cacpolicy)# first-cac-table my_table
Router(config-sbc-sbe-cacpolicy)# first-cac-scope call
Router(config-sbc-sbe-cacpolicy)# cac-table testSecure
Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set
```



```

Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-bandwidth-per-scope 6000000 bps
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# media police degrade
Router(config-sbc-sbe-cacpolicy-cactable-entry) complete
Router(config-sbc-sbe-cacpolicy)# codec system H263 id 34
Router (config-sbc-sbe-codec-def)# type variable
Router (config-sbc-sbe-codec-def)# bandwidth min 370000
Router (config-sbc-sbe-cacpolicy-cactable-entry)# end

```

## 帯域幅フィールド インターワーキング機能

次の項では、帯域幅フィールド インターワーキング機能について説明します。

- 「帯域幅フィールド インターワーキングについて」 (P.403)
- 「帯域幅フィールド インターワーキングの設定」 (P.404)
- 「帯域幅フィールドのインターワーキングの設定例」 (P.408)

## 帯域幅フィールド インターワーキングについて

Cisco Unified Border Element (SP Edition) は、アウトバウンドセッション記述プロトコル (SDP) で帯域幅行を設定する方法を決定する能力をサポートすることによって、帯域幅フィールド インターワーキングをサポートします。Cisco Unified Border Element (SP Edition) は、SDP での Application Specific Maximum (AS) および Transport Independent Application Specific Maximum (TIAS) 帯域幅修飾子をサポートしています。

SDP には、RFC 3556、セッション記述プロトコル (SDP) 帯域幅修飾子に従って、次の構文のあるオプションの帯域幅属性が含まれています。

**b=<modifier>:<bandwidth-value>**

<modifier> は英数字の単語で、メディアまたはセッションで使用される帯域幅を示します。

<bandwidth-value> のデフォルトは、キロビット/秒です。

AS 帯域幅修飾子は、1つの送信元からの単一メディア ストリームの合計帯域幅を指定するために使用されます。

TIAS 帯域幅値は、IP、または TCP や UDP (RFC 3890) などの他のトランスポート層のカウントなしで SDP セッション レベルまたはメディア ストリームに必要な最大帯域幅です。

Cisco Unified Border Element (SP Edition) は、次のコマンドを使用して各隣接に設定される AS および TIAS 帯域幅形式をサポートしています。

- **caller-bandwidth-field** [*as-to-tias* | *tias-to-as*]
- **callee-bandwidth-field** [*as-to-tias* | *tias-to-as*]

帯域幅行を *as-to-tias* 設定に設定すると、アウトバウンド SDP オファー内の SBC で **b=AS** 行が **b=TIAS** 行に変更されます。複数の **b=AS** 行がある場合、最初の行だけが **b=TIAS** 行に変更され、残りは無視されます。

AS の帯域幅修飾子を TIAS に変更すると、次の状況で便利です。

- AS 帯域幅修飾子だけをサポートしているアップストリーム デバイス进行操作している場合、TIAS 帯域幅修飾子ダウンストリームを使用すると、ネットワーク内の帯域幅計算の精度が増すことがあります。ネットワーク シナリオによっては、AS 帯域幅修飾子を使用すると、たとえば IPv4 と IPv6 ネットワークとの間のルーティングなど、帯域幅計算が不正確になることがあります (RFC3890 を参照)。

- 相互運用性のため：AS 帯域幅修飾子を認識しないダウンストリーム デバイスがある場合。

帯域幅行を `as-to-tias` 設定に設定すると、SDP メディア記述子に関連付けられた `b=AS` 行がまだない場合、アウトバウンド SDP オファー内の SBC で `b=TIAS` 行が `b=AS` 行に変更されます。複数の `b=TIAS` 行がある場合、最初の行だけが `b=AS` 行に変更され、残りは無視されます。

TIAS の帯域幅修飾子を AS に変更すると、次の状況で便利です。

- 相互運用性のため：TIAS 帯域幅修飾子を認識しないダウンストリーム ノードがある場合。

SBC はこれらの 2 つの形式間の変換をサポートします。帯域幅行の変換がオファー側隣接でイネーブルの場合、オファー側に戻される前に指定の形式でアンサーに帯域幅行が含まれます。同様に、帯域幅行変換がアンサー側隣接でイネーブルの場合、アンサー側に戻される前に指定の形式でオファーに帯域幅行が含まれます。

同じ規則が、アンサー内の帯域幅行の変換と、オファー内の帯域幅行の変換に適用されます。次のルールがあります。

- SBC は、どの発信帯域幅行形式が設定されているのかを確認します。発信隣接が特定のスタイルの帯域幅行形式を優先するように設定されている場合、その形式が使用されます。したがって、どの AS または TIAS 帯域幅行もその形式に変換されます。
- オファー側の隣接に帯域幅形式プリファレンスが設定されておらず、アンサー側隣接のプリファレンスに対応するために帯域幅形式の変換がオファー側で実行される場合、アンサー側で逆変換が実行されます。

たとえば、アンサー隣接が TIAS 帯域幅行を変換するために設定されます。オファー側の隣接にはプリファレンスがありません。オファー側が、`b=AS` 行を含む SDP オファーを作成します。これは、SBC によって発信オファーの `b=TIAS` スタイルに変換されます。

アンサー側は、増加した帯域幅の要求を表す `b=TIAS` 行で応答します。この増加した帯域幅行は、オファー側が最後に提示したものであるため、オファー側に送信される前に `b=AS` に変換され、戻されます。

## 帯域幅フィールド インターワーキングの設定

この作業は、帯域幅フィールド インターワーキング機能を設定します。



(注)

この手順では、**caller** コマンドと **callee** コマンドが使用されています。シナリオによっては、**caller** と **callee** のコマンド ペアの代わりに **branch** コマンドを使用できます。**branch** コマンドはリリース 3.5.0 で導入されました。このコマンドの詳細については、「[ダイレクト非制限 CAC ポリシーの設定 \(P.139\)](#)」を参照してください。

### 手順の概要

1. **configure**
2. **sbc service-name**
3. **sbc**
4. **cac-policy-set policy-set-id**
5. **first-cac-table table-name**
6. **cac-table table-name**
7. **table-type {policy-set | limit {list of limit tables}}**
8. **entry entry-id**

9. `cac-scope` {*list of scope options*}
10. `caller-bandwidth-field` [`as-to-tias`] [`tias-to-as`]
11. `callee-bandwidth-field` [`as-to-tias`] [`tias-to-as`]
12. `action` [`next-table goto-table-name` | `cac-complete`]
13. `exit`
14. `exit`
15. `complete`

## 手順の詳細

	コマンドまたはアクション	目的
ステップ1	<code>configure terminal</code>  例： Router# <code>configure</code>	グローバル コンフィギュレーション モードをイネーブルにします。
ステップ2	<code>sbc service-name</code>  例： Router(config)# <code>sbc mysbc</code>	SBC サービスのモードを開始します。  • <code>service-name</code> 引数を使用して、サービスの名前を定義します。
ステップ3	<code>sbe</code>  例： Router(config-sbc)# <code>sbe</code>	SBC サービス内で SBE エンティティのモードを開始します。
ステップ4	<code>cac-policy-set policy-set-id</code>  例： Router(config-sbc-sbe)# <code>cac-policy-set 1</code>	SBE エンティティ内で CAC ポリシーセット コンフィギュレーション モードを開始して、必要に応じて新規ポリシーセットを作成します。
ステップ5	<code>first-cac-table table-name</code>  例： Router(config-sbc-sbe-cacpolicy)# <code>first-cac-table StandardListByAccount</code>	ポリシーのアドミSSION コントロール ステージの実行時に処理する最初のポリシー テーブル名を設定します。
ステップ6	<code>cac-table table-name</code>  例： Router(config-sbc-sbe-cacpolicy)# <code>cac-table StandardListByAccount</code>	SBE ポリシー セットのコンテキスト内で、アドミSSION コントロール テーブル (必要に応じて作成します) のコンフィギュレーション モードを開始します。

コマンドまたはアクション	目的
<p><b>ステップ7</b></p> <pre>table-type {policy-set   limit {list of limit tables}}</pre> <p><b>例:</b></p> <pre>Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set</pre>	<p>SBC ポリシー セットのコンテキスト内で CAC テーブルのテーブル タイプを設定します。</p> <p><i>list of limit tables</i> 引数は、テーブル内にあるエントリの <i>match-value</i> フィールドの構文を制御します。使用可能な制限テーブルは次のとおりです。</p> <ul style="list-style-type: none"> <li>• <i>account</i> : アカウント名を比較します。</li> <li>• <i>adj-group</i> : 隣接グループ名を比較します。</li> <li>• <i>adjacency</i> : 隣接名を比較します。</li> <li>• <i>all</i> : 比較タイプはありません。すべてのイベントがこのタイプと一致します。</li> <li>• <i>call-priority</i> : コール プライオリティと比較します。</li> <li>• <i>category</i> : 番号分析が割り当てられたカテゴリを比較します。</li> <li>• <i>dst-account</i> : 宛先アカウント名を比較します。</li> <li>• <i>dst-adj-group</i> : 宛先隣接グループ名を比較します。</li> <li>• <i>dst-adjacency</i> : 宛先隣接名を比較します。</li> <li>• <i>dst-prefix</i> : 着信ディジット スtring の先頭を比較します。</li> <li>• <i>event-type</i> : CAC ポリシー イベント タイプと比較します。</li> <li>• <i>src-account</i> : 送信元アカウント名を比較します。</li> <li>• <i>src-adj-group</i> : 送信元隣接グループ名を比較します。</li> <li>• <i>src-adjacency</i> : 送信元隣接名を比較します。</li> <li>• <i>src-prefix</i> : 発番号 String の先頭を比較します。</li> </ul> <p>機能は、個々の隣接ごとに行う場合と同じように、CAC 設定から隣接グループごとにイネーブルまたはディセーブルにできます。<i>adj-group</i> テーブル タイプは、送信元または宛先の隣接グループで一致します。</p> <p><i>policy-set</i> キーワードが指定されている場合、<i>cac-scope</i> コマンドを使用して、CAC ポリシー セット テーブル内で制限が適用される、各エントリ内の範囲を設定します。</p>
<p><b>ステップ8</b></p> <pre>entry entry-id</pre> <p><b>例:</b></p> <pre>Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</pre>	<p>アドミッション コントロール テーブル内のエントリを作成または変更するモードを開始します。</p>

コマンドまたはアクション	目的
<p><b>ステップ 9</b> <code>cac-scope {list of scope options}</code></p> <p><b>例 :</b>  <pre>Router(config-sbc-sbe-cacpolicy-cactable-entry) # cac-scope src-adjacency</pre></p>	<p>CAC の制限がポリシー セット テーブル内の各エントリに適用される範囲を選択します。</p> <p><i>list of scope options</i> : イベント照合に使用される次のいずれかのストリングを指定します。</p> <ul style="list-style-type: none"> <li>• <i>account</i> : 同じアカウントからのイベント。</li> <li>• <i>adjacency</i> : 同じ隣接からのイベント。</li> <li>• <i>adj-group</i> : 同じ隣接グループのメンバーからのイベント。</li> <li>• <i>call</i> : スコープの限界は単一コール単位です。</li> <li>• <i>category</i> : 同じカテゴリを持つイベント。</li> <li>• <i>dst-account</i> : 同じアカウントに送信されるイベント。</li> <li>• <i>dst-adj-group</i> : 同じ隣接グループに送信されるイベント。</li> <li>• <i>dst-adjacency</i> : 同じ隣接に送信されるイベント。</li> <li>• <i>dst-number</i> : 宛先が同じイベント。</li> <li>• <i>global</i> : スコープの限界はグローバルです。</li> <li>• <i>src-account</i> : 同じアカウントからのイベント。</li> <li>• <i>src-adj-group</i> : 同じ隣接グループからのイベント。</li> <li>• <i>src-adjacency</i> : 同じ隣接からのイベント。</li> <li>• <i>src-number</i> : 同じ送信元番号を持つイベント。</li> </ul>
<p><b>ステップ 10</b> <code>caller-bandwidth-field [as-to-tias] [tias-to-as]</code></p> <p><b>例 :</b>  <pre>Router(config-sbc-sbe-cacpolicy-cactable-entry) # caller-bandwidth-field as-to-tias</pre></p>	<p>発信側に送信されるアウトバウンドセッション記述プロトコル (SDP) で、<b>b=AS</b> 行形式を <b>b=TIAS</b> 行形式へ、または <b>b=TIAS</b> 行形式を <b>b=AS</b> 行形式へに変換するように SBC を設定します。</p> <p>AS = Application Specific Maximum  TIAS = Transport Independent Application Specific Maximum</p>
<p><b>ステップ 11</b> <code>callee-bandwidth-field [as-to-tias] [tias-to-as]</code></p> <p><b>例 :</b>  <pre>Router(config-sbc-sbe-cacpolicy-cactable-entry) # callee-bandwidth-field tias-to-as</pre></p>	<p>着信側に送信されるアウトバウンドセッション記述プロトコル (SDP) で、<b>b=AS</b> 行形式を <b>b=TIAS</b> 行形式へ、または <b>b=TIAS</b> 行形式を <b>b=AS</b> 行形式へに変換するように SBC を設定します。</p> <p>AS = Application Specific Maximum  TIAS = Transport Independent Application Specific Maximum</p>
<p><b>ステップ 12</b> <code>action [next-table goto-table-name   cac-complete]</code></p> <p><b>例 :</b>  <pre>Router(config-sbc-sbe-cacpolicy-cactable-entry) # action cac-complete</pre></p>	<p>アドミッション コントロール テーブルのこのエントリの後で実行するアクションを設定します。指定できるアクションは次のとおりです。</p> <ul style="list-style-type: none"> <li>• <b>next-table</b> キーワードおよび <i>goto-table-name</i> 引数を使用して、処理する次の CAC テーブルを特定します。</li> <li>• <b>cac-complete</b> キーワードを使用して、このスコープの処理を停止します。</li> </ul>

	コマンドまたはアクション	目的
ステップ 13	<b>exit</b>  例： Router(config-sbc-sbe-cacpolicy-cactable-entry) # exit	<b>entry</b> モードを終了して、 <b>cactable</b> モードに移行します。
ステップ 14	<b>exit</b>  例： Router(config-sbc-sbe-cacpolicy-cactable)# exit	<b>cactable</b> モードを終了して、 <b>cacpolicy</b> モードに移行します。
ステップ 15	<b>complete</b>  例： Router(config-sbc-sbe-cacpolicy-cactable-entry) # complete	セット全体をコミットしたら、CAC ポリシー セットを完了します。

## 帯域幅フィールドのインターワーキングの設定例



(注)

この手順では、**caller** コマンドと **callee** コマンドが使用されています。シナリオによっては、**caller** と **callee** のコマンド ペアの代わりに **branch** コマンドを使用できます。**branch** コマンドはリリース 3.5.0 で導入されました。このコマンドの詳細については、「[ダイレクト非制限 CAC ポリシーの設定 \(P.139\)](#)」を参照してください。

次に、オファァ側隣接（発信側）で AS 帯域幅形式を TIAS 帯域幅形式に変換し、アンサー側隣接（着信側）で TIAS 帯域幅形式を AS 帯域幅形式に変換するように SBC が設定される例を示します。

```
Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# cac-table 1
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# caller-bandwidth-field as-to-tias
Router(config-sbc-sbe-cacpolicy-cactable-entry)# callee-bandwidth-field tias-to-as
```

次に、CAC テーブル 1 のエントリ 1 に対して、発信側および着信側帯域幅フィールド情報を含む、詳細情報をリストする例を示します。

```
Router# show sbc mysbc sbe cac-policy-set 1 table 1 entry 1

SBC Service "mysbc"

CAC Policy Set 1
  Active policy set: Yes
  Description:
  Averaging period: 60 sec
  First CAC table: 1
  First CAC scope: global

Table name: cacTable
  Description:
  Table type: policy-set                               Total call failures: 0
```

```

Entry 1
CAC scope:
Action: CAC complete
Max calls per scope: Unlimited
Max in-call rate: Unlimited
Max reg. per scope: Unlimited
Max channels per scope: Unlimited
Early media: Allowed
Early media timeout: None
Number of calls rejected: 0
Max call rate per scope: Unlimited
Max out-call rate: Unlimited
Max reg. rate per scope: Unlimited
Max updates per scope: Unlimited
Early media direction: Both
Transcoder per scope: Allowed
Callee Bandwidth-Field: TIAS-to-AS
Caller Bandwidth-Field: AS-to-TIAS
Media bypass: Allowed
Media flag: Ignore bandwidth-fields (b=)
Renegotiate Strategy: Delta
Max bandwidth per scope: Unlimited
SRTP Transport: Trusted-Only (by default)
Caller hold setting: Standard
Callee hold setting: Standard
Caller privacy setting: Never hide
Callee privacy setting: Never hide
Caller voice QoS profile: Default
Caller video QoS profile: Default
Caller sig QoS profile: Default
Callee voice QoS profile: Default
Callee video QoS profile: Default
Callee sig QoS profile: Default
Restrict codecs to list: Default
Restrict caller codecs to list: Default
Restrict callee codecs to list: Default
Caller inbound SDP policy: None
Caller outbound SDP policy: None
Callee inbound SDP policy: None
Callee outbound SDP policy: None

```

## 隣接単位のコーデック ストリング インターワーキング

ここでは、隣接単位のコーデック ストリング インターワーキング機能について説明します。

- 「隣接単位のコーデック ストリング インターワーキングについて」 (P.409)
- 「隣接単位のコーデック ストリング インターワーキングの制約事項」 (P.410)
- 「隣接単位のコーデック ストリング インターワーキングの設定」 (P.410)
- 「隣接単位のコーデック ストリング インターワーキングの設定例」 (P.415)

## 隣接単位のコーデック ストリング インターワーキングについて

Cisco IOS Release 3.2S から、SBC は、異なる非標準デバイスが SBC 経由で相互動作できるようにするため、非標準 SDP を解釈して、異なる非標準形式の SDP 間でコーデックを変換したり、標準 SDP を標準 SDP に変換できるようになりました。

SBC は、メッセージの各コーデックを調べ、対応するインバウンド SDP で受信したコーデック名が標準名なのかバリエーション名なのかを判定し、コーデックを変換します。

- コーデックが標準の場合、SBC はバリエーションリストから、その標準コーデックの一致するバリエーションを検索します。一致するバリエーションが見つかった場合は、変換されるかそのままパススルーされます。

- コーデックがバリエーションの場合、SBC はそのバリエーションがリストにあるかどうかを確認するためプロファイルを検索します。
  - リストにあるバリエーションは変更せずにパス スルーされます。
  - リストにないバリエーションは同じ標準コーデックの一致するバリエーションに変換できます。
  - 同じ標準コーデックの一致するバリエーションがなく、リストにないバリエーションは、標準の表現に変換されます。

次のシナリオでは、コーデックの変換を避け、SBC は標準の SDP 形式を使用する必要があります。

- H.248 インターフェイスとの一貫性を保つため、H.248 コマンド中で SBC-SIG からメディア ゲートウェイに送信された SDP フラグメントは、標準の表現を使用する必要があります。MG との相互運用性を確保し、サードパーティ MG が SBC H.248 プロファイルを実装するのを簡単にするため、H.248 インターフェイスを指定する必要があります。
- 課金インターフェイスの一貫性を確保するため、XML 課金レコードに SBC-SIG によって格納される SDP フラグメントは、標準の表現を使用する必要があります。サードパーティ製の課金サーバは XML 課金レコードを簡単に解釈できます。
- コーデック ストリングは H.245 シグナリング メッセージに現れず、列挙型がコーデックを表すために使用されます。したがって、コーデックの変換は、SIP 発信隣接だけに適用されます。

## 隣接単位のコーデック ストリング インターワーキングの制約事項

隣接単位のコーデック ストリング インターワーキング機能には次の制約があります。

- オファーをパス スルーするときに指定したコーデックの特定のバリエーションが使用される場合、同じバリエーションはアンサーをパス スルーするときに使用できません。
- コールの特定の側で、着信 SDP をあるバリエーションを使用して解釈し、別のバリエーションに基づいて発信 SDP を変換するように SBC を設定することはできません。
- SBC は非標準のストリングを使用したバリエーションだけをサポートするため、標準の IANA コーデック ストリングを使用するバリエーションを定義できません。
- 2つのバリエーションが同じ標準コーデックにマッピングされる場合、トランスコードはバリエーション間で変換を行いません。たとえば、エンドポイントが G7231H と G7231L を異なるコーデックであると認識しても、SBC はこれらコーデックの間でトランスコードできません。

## 隣接単位のコーデック ストリング インターワーキングの設定

ここでは、隣接単位のコーデック ストリング インターワーキング機能の次の設定について説明します。

- 「コーデック バリエーション変換の設定」(P.410)
- 「CAC ポリシー セットのコーデックの設定」(P.413)

### コーデック バリエーション変換の設定

この作業では、SBC のコーデック バリエーション変換を設定する方法について説明します。

#### 手順の概要

1. `configure terminal`
2. `sbc service-name`



3. **sbe**
4. **codec system** *sys-codec id payload-id*
5. **fntp** *fntp-string*
6. **exit**
7. **codec variant** **codec** *variant-name*
8. **variant** *variant-codec-encoded-name*
9. **standard** *standard-codec-name*
10. **fntp** *fntp-string*
11. **exit**
12. **codec variant** **profile** *profile-name*
13. **variant** *variant-name*
14. **end**
15. **show sbc** *service-name sbe codecs variant [profile]*

## 手順の詳細

	コマンドまたはアクション	目的
ステップ1	<b>configure terminal</b>  例： Router# configure	グローバル コンフィギュレーション モードをイネーブルにします。
ステップ2	<b>sbc</b> <i>service-name</i>  例： Router(config)# sbc mysbc	SBC サービス モードを開始します。  • <i>service-name</i> 引数を使用して、サービスの名前を定義します。
ステップ3	<b>sbe</b>  例： Router(config-sbc)# sbe	SBC サービス内で SBE エンティティ モードを開始します。
ステップ4	<b>codec system</b> <i>sys-codec id payload-id</i>  例： Router(config-sbc-sbe)# codec system G723 id 4	既存のコーデックを変更するためにコーデック定義モードを開始します。  • <i>sys-codec</i> : SBC に含まれるコーデック。 • <b>id</b> <i>payload-id</i> : スタティック ペイロード ID。有効な値は 0 ~ 96 です。
ステップ5	<b>fntp</b> <i>fntp-string</i>  例： Router(config-sbc-sbe-codec-def)# fntp annexa=yes	SDP の Format-Specific Parameter (FMTP) のデフォルト値を設定します。  • <i>fntp-string</i> : SDP の FMTP スtring を、「名前=値」の形式で指定します。  (注) バリエーションに関連付けられているデフォルトの FMTP 値を表示するには、 <b>show sbc sbe codecs variant</b> コマンドを使用します。

	コマンドまたはアクション	目的
ステップ6	<code>exit</code>  例： Router(config-sbc-sbe-codec-def)# exit	コーデック定義モードを終了し、SBE エンティティ モードを開始します。
ステップ7	<code>codec variant codec variant-name</code>  例： Router(config-sbc-sbe)# codec variant codec G723-H-1	コーデック バリエントを設定、変更、削除するため、コーデック バリエント モードを開始します。  • <i>variant-name</i> : コーデック バリエントの名前。
ステップ8	<code>variant variant-codec-encoded-name</code>  例： Router(config-sbc-sbe-codec-var-codec)# variant G723-H-1	符号化されたコーデック バリエントの名前を定義します。  • <i>variant-codec-encoded-name</i> : バリエントの非標準コーデック スtring。  (注) 「#」は、基本バリエント用に予約されています。したがって、バリエント名は「#」から開始できません
ステップ9	<code>standard standard-codec-name</code>  例： Router(config-sbc-sbe-codec-var-codec)# standard G723	標準のコーデック バリエント名を定義します。  • <i>standard-codec-name</i> : 標準のシステム コーデックの名前。
ステップ10	<code>fntp fntp-string</code>  例： Router(config-sbc-sbe-codec-var-codec)# fntp bitrate=6.3	コーデック バリエントの FMTP パラメータを設定します。  • <i>fntp-string</i> : FMTP スtringを、「名前=値」の形式で指定します。  (注) バリエントに関連付けられているデフォルトの FMTP 値を表示するには、 <b>show sbc sbe codecs variant</b> コマンドを使用します。
ステップ11	<code>exit</code>  例： Router(config-sbc-sbe-codec-var-codec)# exit	コーデック バリエント モードを終了し、SBE エンティティ モードを開始します。
ステップ12	<code>codec variant profile profile-name</code>  例： Router(config-sbc-sbe)# codec variant profile Profile-1	コーデック バリエント プロファイルを設定するためコーデック バリエント プロファイル モードを開始します。  • <i>profile-name</i> : コーデック プロファイル名。
ステップ13	<code>variant variant-name</code>  例： Router(config-sbc-sbe-codec-var-prof)# variant G723-H-1	コーデック バリエントの名前を定義します。  • <i>variant-name</i> : コーデック バリエントの名前。  (注) 「#」は、基本バリエント用に予約されています。したがって、バリエント名は「#」から開始できません

	コマンドまたはアクション	目的
ステップ 14	<pre>end</pre> <p>例 :</p> <pre>Router(config-sbc-sbe-codec-var-prof)# end</pre>	<p>コーデック バリアント プロファイル モードを終了し、特権 EXEC モードを開始します。</p>
ステップ 15	<pre>show sbc service-name sbe codecs variant [profile]</pre> <p>例 :</p> <pre>Router# show sbc mySBC sbe codecs variant</pre>	<p>SBC のコーデック バリアントに関する情報を表示します。</p> <ul style="list-style-type: none"> <li>• <b>profiles</b> : コーデック バリアント プロファイルに関する情報を表示します。</li> </ul>

## CAC ポリシー セットのコーデックの設定

ここでは、コーデックの変換をイネーブルにし、CAC ポリシー セットのコーデック バリアント プロファイルを設定する方法について説明します。コーデック バリアントの変換をイネーブルまたはディセーブルすると、次のイベントが発生します。

- コーデック バリアント変換がディセーブルの場合、SBC は指定されたバリアント プロファイルを考慮しません。渡されたすべてのコーデックは元の表現のままになり、SBC によって追加された新しいコーデックは標準の表現で追加されます。
- コーデックのバリアント変換はイネーブルだが、バリアント プロファイルが設定されていない場合、すべてのコーデックはその標準の表現に変換されます。
- コーデックのバリアントがイネーブルで、バリアント プロファイルが設定されている場合、プロファイルに一致するコーデックが適切なバリアントの表現に変換され、バリアント プロファイルに一致しないコーデックはその標準の表現に変換されます。



(注)

この手順では、**caller** コマンドと **callee** コマンドが使用されています。シナリオによっては、**caller** と **callee** のコマンドペアの代わりに **branch** コマンドを使用できます。**branch** コマンドはリリース 3.5.0 で導入されました。このコマンドの詳細については、「[ダイレクト非制限 CAC ポリシーの設定](#)」(P.139) を参照してください。

### 手順の概要

1. **configure terminal**
2. **sbc service-name**
3. **sbe**
4. **cac-policy-set policy-set-id**
5. **cac-table table-name**
6. **table-type {policy-set | limit {list of limit tables}}**
7. **entry entry-id**
8. **caller codec convert**
9. **callee codec convert**
10. **caller codec profile profile-name**
11. **callee codec profile profile-name**
12. **exit**

13. exit

14. complete

## 手順の詳細

	コマンドまたはアクション	目的
ステップ1	<code>configure terminal</code>  例: Router# configure	グローバル コンフィギュレーション モードをイネーブルにします。
ステップ2	<code>sbc service-name</code>  例: Router(config)# sbc mysbc	SBC サービス モードを開始します。  <i>service-name</i> 引数を使用して、サービスの名前を定義します。
ステップ3	<code>sbe</code>  例: Router(config-sbc)# sbe	SBC サービス内で SBE エンティティ モードを開始します。
ステップ4	<code>cac-policy-set policy-set-id</code>  例: Router(config-sbc-sbe)# cac-policy-set 1	SBE エンティティ内で CAC ポリシーセット コンフィギュレーション モードを開始して、必要に応じて新規ポリシーセットを作成します。
ステップ5	<code>cac-table table-name</code>  例: Router(config-sbc-sbe-cacpolicy)# cac-table StandardListByAccount	SBE ポリシーセットのコンテキスト内で、アドミッションコントロール テーブルを設定するための CAC テーブルモードを開始します。必要な場合はテーブルを作成します。
ステップ6	<code>table-type {policy-set   limit {list of limit tables}}</code>  例: Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set	SBC ポリシーセットのコンテキスト内で CAC テーブルのテーブル タイプを設定します。
ステップ7	<code>entry entry-id</code>  例: Router(config-sbc-sbe-cacpolicy-cactable)# entry 1	アドミッションコントロール テーブル内のエントリを作成または変更するモードを開始します。
ステップ8	<code>caller codec convert</code>  例: Router(config-sbc-sbe-cacpolicy-cactable-entry)# caller codec convert	発信側でコーデック バリエーション変換をイネーブルまたはディセーブルにします。

	コマンドまたはアクション	目的
ステップ9	<code>callee codec convert</code>  例： Router(config-sbc-sbe-cacpolicy-cactable-entry) # callee codec convert	着信側でコーデック バリエント変換をイネーブまたはディセーブにします。
ステップ10	<code>caller codec profile profile-name</code>  例： Router(config-sbc-sbe-cacpolicy-cactable-entry) # caller codec profile	発信側でコーデック バリエント プロファイルを指定します。  • <i>profile-name</i> : コーデック バリエント プロファイルの名前。
ステップ11	<code>callee codec profile profile-name</code>  例： Router(config-sbc-sbe-cacpolicy-cactable-entry) # callee codec profile	着信側でコーデック バリエント プロファイルを指定します。  • <i>profile-name</i> : コーデック バリエント プロファイルの名前。
ステップ12	<code>exit</code>  例： Router(config-sbc-sbe-cacpolicy-cactable-entry) # exit	<b>entry</b> モードを終了し、 <b>cactable</b> モードを開始します。
ステップ13	<code>exit</code>  例： Router(config-sbc-sbe-cacpolicy-cactable)# exit	<b>cactable</b> モードを終了し、 <b>cacpolicy</b> モードを開始します。
ステップ14	<code>complete</code>  例： Router(config-sbc-sbe-cacpolicy)# complete	セット全体をコミットしたら、CAC ポリシー セットを完了します。

## 隣接単位のコーデック スtring インターワーキングの設定例

次に、発信側が G723 ビットレート 6.3 annexa=no コーデックをサポートし、着信側が G.723.1 コーデック バリエントをサポートする場合に、隣接単位のコーデック スtring インターワーキング機能を設定する例を示します。

```
configure terminal
sbc MySBC
sbe
codec variant codec PCMU.1
    standard PCMU
    variant PCMU.1
exit
codec variant profile pcmu-var
    variant PCMU.1
    variant #G.723.1/H
exit
cac-policy-set 2
    first-cac-table codec-convert
    first-cac-scope src-adjacency
cac-table codec-convert
```

```
table-type limit src-adjacency
entry 1
  match-value CallMgrA
  callee codec profile pcmu-var
  callee codec convert
  media police strip
  action cac-complete
complete
end
```