



## DNS-AS を使用した AVC の設定

- [DNS-AS を使用した AVC に関する前提条件](#) (1 ページ)
- [DNS-AS を使用した AVC の制約事項およびガイドライン](#) (1 ページ)
- [DNS-AS を使用した AVC について](#) (2 ページ)
- [DNS-AS を使用した AVC の設定方法](#) (7 ページ)
- [DNS-AS を使用した AVC の監視](#) (22 ページ)
- [DNS-AS を使用した AVC のトラブルシューティング](#) (26 ページ)
- [DNS-AS を使用した AVC の機能履歴および情報](#) (27 ページ)

### DNS-AS を使用した AVC に関する前提条件

- DNS-AS を使用するために、[Cisco ONE for Access](#) を所有している。
- マルチレイヤ スイッチ (MLS) の Quality of Service (QoS) が有効になっている。
- DNS-AS を使用した AVC を有効にする前に、権威 DNS サーバー内にデータを維持しており、到達可能である。
- DNS-AS クライアントがホストから開始されるフォワードルックアップ要求をスヌーピングできる。
- DNS パケットのロギングとスヌーピングを確実に実行するため、**service-policy input** コマンドを使用してインターフェイスにポリシーマップを付加している。

### DNS-AS を使用した AVC の制約事項およびガイドライン

- この機能は Cisco Catalyst 3560-CX シリーズ スイッチ上でのみサポートされています。Cisco Catalyst 2960-CX シリーズ スイッチではサポートされていません。
- フォワードルックアップのみがサポートされています。
- 2 台の DNS サーバーがサポートされます (フェールオーバーの場合)。1 台がプライマリ DNS サーバー、もう 1 台がセカンダリ DNS サーバーと見なされます。

- IPv6 はサポートされていません。AAA 要求、および IPv6 DNS サーバーはサポートされていません。
- DNS-AS を使用した AVC は、物理インターフェイス上の入力方向でのみサポートされています。
- Virtual Routing and Forwarding (VRF) はサポートされていません。
- TCAM (Ternary Content Addressable Memory) に影響するため、バインディングテーブル内の DNS-AS を使用した AVC アプリケーションは最大で 300 個までにすることを推奨します。アプリケーションを追加することによって TCAM にどのように影響するかについては、この章の「DNS-AS を使用した AVC のトラブルシューティング」の項を参照してください。

## DNS-AS を使用した AVC について

信頼できるソースとしてのドメイン ネーム システム (DNS-AS) 機能を使用した Application Visibility Control (AVC) (DNS-AS を使用した AVC) は、組織内の信頼ネットワーク トラフィックの識別と分類を制御する一元化された手段を提供します。これは、対象のドメインに対して権威のある DNS サーバーに格納されたネットワーク メタデータを使用することで行われ、アプリケーションを識別し、サービス品質 (QoS) によって対応するトラフィックを分類して適切なポリシーを適用し、Flexible Netflow (FNF) によって、アプリケーション情報を監視して外部コレクタにエクスポートします。

この機能は以下を提供します。

- アプリケーションの可視性：アプリケーションの可視性を向上させます。

DNS-AS メカニズムは要求をスヌーピングします。これには、CPU 集約型のディープ パケット インスペクション (DPI) は必要ありません。トラフィックの分類は、DPI ではなく、DNS 要求によるものであるため、この機能はネットワーク トラフィックが暗号化されているシナリオに適しています。

- メタデータ駆動：アプリケーションに関する情報を使用します。

ネットワークを全体的にプログラムできるため、自動運転車のように動作します。トラフィックの暗号化の有無に関わらず、ネットワーク内の必要なアプリケーションすべてに関する情報を入手できます。

- 一元管理：クロスドメインアプリケーションを対象にしたポリシー コントローラを使用します。

この機能は、一般的に使用可能な既存のクエリ/応答メカニズムを活用して、権威サーバーとして機能するように組織内のローカル DNS サーバーを有効にし、アプリケーション分類情報をエンタープライズ ネットワーク内の DNS-AS クライアントに伝播させます。

- 管理アクセスなしの制御：コントローラベースのアプローチに代わる手段を提供します。

この機能は、ネットワークがクラウド内にあり、クラウドの所有者ではないという状況もサポートします。この場合も、これらのデバイスに対して管理制御を行えなくても、インターネットを通じてネットワーク デバイスを制御できます。

## DNS-AS を使用した AVC の概要

プロセスは、ネットワーク トラフィックの管理と制御に関連する組織の要件で開始します。ネットワーク内のさまざまなホスト（電話機、PC など）で実行するソフトウェア アプリケーション、このようなデバイスがアクセスするドメイン（Web サイト）およびアプリケーション、ならびに組織内のこれらのドメインやアプリケーションのビジネス関連性を評価することから始めます。

この評価は、組織が「信頼」しているドメインやアプリケーションのリストを作成し、残りのドメインやアプリケーションはすべて信頼できないと指定するのに役立ちます。

ネットワーク上でDNS-ASを有効にし、信頼ドメインのリストを使用することで、ネットワーク内のネットワーキング デバイスやDNS-AS は、ネットワーク トラフィックが属するアプリケーションや、要求されているドメインを識別します。トラフィックが信頼リストに含まれている限り、スイッチはDNSサーバーにメタデータやIPの情報を要求します。この要求はDNSクエリの形式で送信されます。受信されるとすぐに、そのリソース レコードの存続可能時間（TTL）が切れるまで、応答がローカルにキャッシュされます。応答はトラフィックにバインドされ、DNS-ASクライアントが適切にトラフィックを識別、分類、転送できるようになります。

## DNS-AS を使用した AVC の主要概念

概念	意味または定義
メタデータ（RFC6759）	DNS-AS 機能を使用した AVC では、メタデータとしてトラフィック分類情報、アプリケーション識別情報、およびビジネス関連性情報が含まれます。  メタデータはTXTレコード形式で維持されます。次に、所定の形式のメタデータ例を示します： <code>CISCO-CLS=app-name:example app-class:TD business:YES app-id:CU/28202</code>
フォワードルックアップ	ホストから発信される IP アドレスの要求、または「A」レコードの要求。  DNS-AS 機能を使用した AVC には、ネットワーク トラフィック内でこれらのフォワードルックアップをスヌーピングできる必要があります。
ホスト	ユーザーがソフトウェアアプリケーションを実行すると、PC やモバイルは Web サイトなどにアクセスします。  フォワードルックアップ要求はホストから開始されます。

概念	意味または定義
クライアントまたは DNS-AS クライアント	<p>ネットワーク全体に存在するネットワークデバイス。ホストトラフィックは常にこのようなクライアントを通じてルーティングされます。</p> <p>(注) この章では、アクセススイッチとしてのみ導入されている Cisco Catalyst スイッチ上の DNS-AS を使用した AVC の設定について説明します。このドキュメント全体を通じて、「クライアント」および「DNS-AS クライアント」という用語は、DNS-AS を使用した AVC が有効になっているスイッチのことを指します。</p> <p>DNS-AS クライアントは権威 DNS サーバーからメタデータを受信し、この情報のデータベースをレコード形式で維持します。クライアントのデータベースにレコードが維持される期間は、レコードの TTL によって決定されます。</p>
バインディングテーブル	<p>DNS-AS クライアントに存在し、解析済み DNS サーバー応答 (TXT レコードと「A」レコード) のデータベースとして機能するテーブル。</p> <p>各 DNS-AS クライアントには各自のバインディングテーブルがあります。</p> <p>信頼ドメインリストは信頼ドメインのみ含むリストです。このバインディングテーブルと混同しないでください。</p>
「A」レコード	<p>ドメイン名と IP アドレス情報 (IPv4 アドレスのみ) を含むレコード。これは DNS サーバー応答の 1 つであり (もう 1 つは TXT レコード)、期限が事前に定義されています。</p> <p>ホストからのフォワードルックアップ要求は、「A」レコードの要求です。</p>
TXT DNS-AS リソース レコードまたは TXT レコード	<p>メタデータを含むレコード。これは、DNS サーバー応答の 1 つであり (もう 1 つは「A」レコード)、期限が事前に定義されています。</p> <p>TXT レコードは 255 文字までに制限されています。</p> <p>DNS-AS を使用した AVC の場合、TXT 属性は常に CISCO-CLS です。CISCO CLS= で始まるすべての TXT レコードは、DNS-AS を使用した AVC メッセージとして認識できます。このメッセージの形式は次のとおりです。</p> <p>CISCO-CLS =&lt;option&gt;:&lt;val&gt;{ &lt;option&gt;:&lt;val&gt;}*</p>

概念	意味または定義
存続可能時間 (TTL)	<p>バインディングテーブル内の「A」レコードと TXT レコードの期限。</p> <p>TTL 値は DNS サーバー上で設定されます。</p> <p>TTLは、TXT レコードと「A」レコードの両方に適用されますが、DNS クライアントは DNS サーバーからの「A」レコード応答にのみ従います。</p>
権威 DNS サーバー	<p>すべてのクライアント メタデータおよび「A」レコード要求で使用される DNS サーバー。</p> <p>どの DNS ドメインにも、権威 DNS サーバーが 1 つのみ存在します。</p> <p>このサーバーがアプリケーション メタデータのレコードを TXT レコードの形式で維持し、必要な形式で維持されているドメイン名に関するクエリにのみ、応答を返します。</p> <p>次に、所定の形式のメタデータ例を示します：  <code>CISCO-CLS=app-name:example app-class:TD business:YES app-id:CU/28202</code></p>

## DNS-AS プロセス フローを使用した AVC

DNS-AS を使用した AVC の動作には、DNS スヌーピング プロセスと DNS-AS クライアント プロセスが含まれます。この両方は緩やかに結びついていますが、独立したプロセスです。

### DNS スヌーピング プロセス

**ステップ 1** ホストが「A」レコード要求を開始します。

組織のユーザーはオフィスビル内の会議室にいます。ここでは、関連付けられた DNS-AS クライアントはスイッチです（この会議室からのネットワークトラフィックはこのスイッチを通じてルーティングされます）。ユーザーが Web サイトの `www.example.com` を検索し、それにより「A」レコードの要求が開始されます。

**ステップ 2** 権威 DNS サーバーが、「A」レコード応答で応答します。

### DNS-AS クライアント プロセス

**ステップ 1** DNS-AS クライアントは権威 DNS サーバーに DNS クエリ（TXT 要求）を送信します。

図 : DNS-AS プロセス フローを使用した AVC

DNS-AS クライアントは、（信頼ドメインリストのエントリに対応する）要求を継続的にスヌーピングし、ホストのフォワードルックアップ要求を検索します。DNS-AS クライアントはスヌーピングの結果に基づいて TXT 要求を権威 DNS サーバーに送信します。

（注） DNS-AS クライアントはホストの「A」レコード要求のコピーを受信しますが、ホストの元の要求をいかなる方法でも変更しません。

ステップ 2 権威 DNS サーバーは TXT レコード応答で応答します。

ステップ 3 TXT 応答の成功後に「A」レコード要求が続きます。

ステップ 4 権威 DNS サーバーが、「A」レコード応答で応答します。

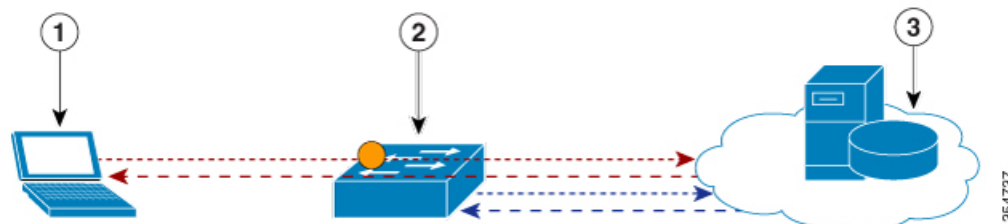
ステップ 5 DNS-AS クライアントは応答を解析し、バインディングテーブルに保存します。

DNS-AS クライアントは TXT レコードと「A」レコードをバインディングテーブルに保存します。応答は、「A」レコードの TTL で指定された期間、バインディングテーブルに保存されたままとなります。システムによって、バインディングテーブル内の完全修飾ドメイン名の重複エントリが自動的に確認され、防止されます。



DNS-AS クライアントは、（DNS サーバーから）受信したメタデータを使用して、QoS ポリシーを適用する必要があるかどうかを決定します。

DNS-AS クライアントは識別したアプリケーションに関する情報を FNF に転送し、この情報をエクスポートできるようにします。

図 : DNS-AS プロセス フローを使用した AVC



1	ホスト	2	DNS-AS クライアント	3	権威 DNS サーバー
<b>パート I : DNS スヌーピング プロセス</b>					
	ホストから DNS サーバーへの「A」レコード要求		DNS サーバーからホストへの「A」レコード応答		
<b>パート II : DNS-AS クライアント プロセス</b>					
	DNS-AS クライアントが保存する「A」レコード要求のコピー	-	-		

	DNS-AS クライアントから DNS サーバーへの TXT レコードと「A」レコード要求		DNS サーバーから DNS-AS クライアントへの TXT レコードと「A」レコード応答
---	---	--	---

## DNS-AS を使用した AVC 用のデフォルト設定

DNS-AS は無効になっています。

## DNS-AS を使用した AVC の設定方法

### メタデータ ストリームの生成

アプリケーション メタデータは、ローカルの権威 DNS サーバーで設定され、保存されます。信頼ドメインごとに、既定の形式（メタデータ ストリーム）で、アプリケーション分類情報を設定します。これは、アプリケーションメタデータを照会されたときにサーバーがスイッチに伝達する情報です。スイッチがアプリケーションに関する TXT クエリを送信すると、DNS サーバーが TXT 応答で関連メタデータを送信します。

メタデータ ストリームを生成するには、次のタスクを実行します。

#### 手順の概要

1. [AVC リソース レコード ジェネレータ](#)に移動します。
2. メタデータ ストリームを生成するオプションのいずれかをクリックします。
  - Generate predefined
  - Generate custom
3. 信頼ドメインとしてマークした DNS ドメインを担う DNS サーバーの対応する TXT リソース レコードにメタデータをコピーします。

#### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<a href="#">AVC リソース レコード ジェネレータ</a> に移動します。 例： <code>CISCO-CLS=app-name:example app-class:ID business:YES app-id:CU/28202</code>	これは、アプリケーションやドメインに TXT レコード形式でメタデータ ストリームを生成するのに役立ちます。 次のメタデータ フィールドを指定できます。 <ul style="list-style-type: none"> <li>• (任意) ドメイン名</li> </ul>

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> <li>• (必須) アプリケーション名：値が必須です。既存のアプリケーション名またはカスタムアプリケーション名を使用できます。</li> <li>• 既存のアプリケーション名 (<b>app-name:</b>) : 標準アプリケーションのリストから選択します。</li> <li>• (任意) カスタム アプリケーション名 (<b>app-name:</b>) : カスタム アプリケーション名を入力する場合は、メタデータ ストリーム内にもトラフィッククラスとビジネス関連性情報を維持する必要があります。</li> <li>• (任意) セレクタ ID (<b>app-id:</b>) : 分類エンジン ID (最初の 8 ビット) とセレクタ ID (次の 24 ビット) から構成されます。 <ul style="list-style-type: none"> <li>• エンジン ID または分類エンジン ID : セレクタ ID のコンテキストを定義します。次のエンジン ID のみが使用できます。 <p>L3 : IANA レイヤ 3 のプロトコル番号</p> <p>L4 : IANA レイヤ 4 のウェルノウン ポート番号</p> <p>L7 : シスコのグローバルアプリケーション ID</p> <p>CU : カスタム プロトコルこのエンジン ID をカスタムアプリケーション名に使用します。</p> </li> <li>• セレクタ ID : 所定の分類エンジン ID のアプリケーション ID。1 ~ 65535 の数値を入力します。 <p>(注) 既存のアプリケーション名にエンジン ID とセレクタ ID を入力する場合は、Network Based Application Recognition (NBAR) の標準に適合させる必要があります。適合させた後でのみ、FNF エクスポートが共通の ID を一貫した方法で報告します。</p> </li> </ul> </li> <li>• (任意) ポート範囲 (<b>server-port:</b>)</li> </ul>



	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> <li>• (任意) トラフィック クラス (<b>app-class:</b>)</li> <li>• (任意) ビジネス関連性 (<b>business:</b>) : yes または no を選択しなかった場合、ビジネス関連性の値は <b>app-class</b> または <b>app-name</b> に基づき、その優先順位の順序で設定されます。</li> </ul> <p>ここでトラフィッククラスとビジネス関連性フィールドが QoS トラフィック分類にマッピングされる方法については、「<a href="#">アプリクラスと QoS トラフィックのマッピング</a>」を参照してください。</p>
ステップ 2	<p>メタデータストリームを生成するオプションのいずれかをクリックします。</p> <ul style="list-style-type: none"> <li>• Generate predefined</li> <li>• Generate custom</li> </ul> <p>例： Generate predefined</p>	<p><b>Generate predefined</b> : 既知のアプリケーションに事前に定義されたメタデータストリームを、ベストプラクティスのデフォルト値を使用して生成します。</p> <p><b>Generate custom</b> : 独自のアプリケーションのカスタムメタデータストリームをカスタム値を使用して生成します。</p>
ステップ 3	<p>信頼ドメインとしてマークした DNS ドメインを担う DNS サーバーの対応する TXT リソースレコードにメタデータをコピーします。</p>	<p>メタデータストリームを Web サイトからコピーし、使用している権威 DNS サーバーに貼り付けます。</p>

## 権威サーバーとしての DNS サーバーの設定

すべての DNS クエリを 1 台の権威 DNS サーバーに送信するように、ネットワーク内のすべての DNS-AS クライアントを設定する必要があります。Cisco Catalyst スイッチで次のタスクを実行します。

### 手順の概要

1. **configure terminal**
2. **ip name-server server-address**

### 手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p><b>configure terminal</b></p> <p>例：  スイッチ# <b>configure terminal</b></p>	<p>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。</p>

	コマンドまたはアクション	目的
ステップ 2	<b>ip name-server server-address</b> 例 : スイッチ(config)# <b>ip name-server server-address 192.0.2.1 192.0.2.2</b>	権威 DNS サーバーの IP アドレスを指定します。ポート番号は常に 53 です。 フェールオーバーに備えて最大 2 台の DNS サーバーを設定できます。 (注) このコマンドを使用すると、最大 6 台のネームサーバー (IPv4 および IPv6) を設定できます。シーケンス内の最初の 2 つ以上の IP アドレスを IPv4 アドレスにします。これは、DNS-AS 機能を使用した AVC がこれらのみを使用するためです。次の例では、最初の 2 つのアドレスが IPv4 (192.0.2.1 および 192.0.2.2)、3 番目のアドレス (2001:DB8::1) は IPv6 アドレスです。DNS-AS を使用した AVC は最初の 2 つを使用します。 スイッチ(config)# <b>ip name-server 192.0.2.1 192.0.2.2 2001:DB8::1</b>

## DNS-AS を使用した AVC の有効化

DNS-AS はデフォルトで無効になっています。Cisco Catalyst スイッチで機能を有効にするには、次のタスクを実行します。

### 手順の概要

1. **configure terminal**
2. **[no] avc dns-as client enable**

### 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 : スイッチ# <b>configure terminal</b>	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>[no] avc dns-as client enable</b> 例 : スイッチ(config)# <b>avc dns-as client enable</b>	スイッチで DNS-AS を使用した AVC (DNS-AS クライアント) を有効にします。 次に、システムによりバインディングテーブルが作成されます。このバインディングテーブルでは、解

	コマンドまたはアクション	目的
		<p>析した DNS サーバー応答が TTL が期限切れになるまで保存されます。</p> <p>(注) DNS パケットロギングやスヌーピングを確実に実行するには、(トラフィッククラスを決定する関連クラスマップを含んでいる) ポリシーマップを <b>service-policy input</b> コマンドを使用してインターフェイスに付加する必要があります。詳細については、<a href="#">DNS-AS を使用した AVC 用 QoS の設定 (12 ページ)</a> を参照してください。</p>

## 信頼ドメインのリストの維持

信頼ドメインは、DNS-AS を使用した AVC が有効になっている DNS-AS クライアントごとに保存されます。DNS-AS クライアントで機能が最初に有効になった時点では、リストは空です。スイッチで信頼すべきドメインを入力する必要があります。スイッチは、このリストに維持されているネットワーク トラフィックのみをスヌーピングします。信頼ドメイン リストにエントリを作成するには、次のタスクを実行します。

### 手順の概要

1. **configure terminal**
2. **[no] avc dns-as client trusted-domains**
3. **[no] domain domain-name**

### 手順の詳細

	コマンドまたはアクション	目的
<b>ステップ 1</b>	<b>configure terminal</b> 例： スイッチ# <b>configure terminal</b>	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。
<b>ステップ 2</b>	<b>[no] avc dns-as client trusted-domains</b> 例： スイッチ(config)# <b>avc dns-as client trusted-domains</b>	信頼ドメイン コンフィギュレーション モードを開始します。
<b>ステップ 3</b>	<b>[no] domain domain-name</b> 例： スイッチ(config-trusted-domains)# <b>domain www.example.com</b> OR	信頼ドメインリストに追加するドメイン名を入力します。これにより、DNS-AS クライアントの信頼ドメインリストの一部が形成されます。残りのすべてのドメインは無視され、デフォルトの転送動作に従います。

	コマンドまたはアクション	目的
	スイッチ (config-trusted-domains) # domain *example.com	最大 50 ドメインを入力できます。  ドメイン名の照合には、正規表現を使用できます。たとえば、組織のすべてのドメインを表現するために Switch (config-trusted-domains) # domain *.example.*、を入力した場合、DNS-AS クライアントは www.example.com、ftp.example.org、および、組織「example」に関連するその他のすべてのドメインを照合します。ただし、このようなエントリは自身の裁量で使用してください。バインディングテーブルのサイズを大幅に拡大させる可能性があります。

## DNS-AS を使用した AVC 用 QoS の設定

信頼できるトラフィックをメタデータストリームに定義されているように分離し、分類するには、クラスマップを作成し（トラフィッククラスごとに1つ）、トラフィッククラスの一致基準とビジネス関連性の一致基準を定義し、ポリシーマップを作成し、クラスマップを追加し、アクションを設定し、ポリシーマップをインターフェイスに付加する必要があります。詳細については、このガイドの「QoS の設定」の章の「」「」 「[分類の概要](#)」の項を参照してください。

### 簡単な QoS モデルのクラス マップの設定

プロビジョニングする必要があるトラフィック クラスの数を特定するには、12クラスの簡単な QoS モデルを使用します。このモデルは、統一された標準ベースの推奨事項を提供し、QoS 設計と導入の組織全体にわたる均一性と一貫性を保証するのに役立ちます。次の出力例では、12クラスの簡単な QoS モデルに従い、トラフィック クラスとビジネス関連性のクラス マップ設定が表示されています。



(注) DNS-AS 機能でのみ、各クラスに 2 つの一致属性を指定できます。

```
class-map match-all VOICE
match protocol attribute traffic-class voip-telephony
match protocol attribute business-relevance business-relevant
class-map match-all BROADCAST-VIDEO
match protocol attribute traffic-class broadcast-video
match protocol attribute business-relevance business-relevant
class-map match-all REAL-TIME-INTERACTIVE
match protocol attribute traffic-class real-time-interactive
match protocol attribute business-relevance business-relevant
class-map match-all MULTIMEDIA-CONFERENCING
match protocol attribute traffic-class multimedia-conferencing
match protocol attribute business-relevance business-relevant
class-map match-all MULTIMEDIA-STREAMING
match protocol attribute traffic-class multimedia-streaming
match protocol attribute business-relevance business-relevant
```

```
class-map match-all SIGNALING
match protocol attribute traffic-class signaling
match protocol attribute business-relevance business-relevant
class-map match-all NETWORK-CONTROL
match protocol attribute traffic-class network-control
match protocol attribute business-relevance business-relevant
class-map match-all NETWORK-MANAGEMENT
match protocol attribute traffic-class ops-admin-mgmt
match protocol attribute business-relevance business-relevant
class-map match-all TRANSACTIONAL-DATA
match protocol attribute traffic-class transactional-data
match protocol attribute business-relevance business-relevant
class-map match-all BULK-DATA
match protocol attribute traffic-class bulk-data
match protocol attribute business-relevance business-relevant
class-map match-all SCAVENGER
match protocol attribute business-relevance business-irrelevant
```

### 簡単な QoS モデルのポリシー マップの定義

次の出力例では、ポリシー マップの定義と、12 クラスの簡単な QoS モデルですべてのトラフィック クラスをマーキングするトラフィック属性が表示されています。

```
policy-map MARKING
class VOICE
set dscp ef
class BROADCAST-VIDEO
set dscp cs5
class REAL-TIME-INTERACTIVE
set dscp cs4
class MULTIMEDIA-CONFERENCING
set dscp af41
class MULTIMEDIA-STREAMING
set dscp af31
class SIGNALING
set dscp cs3
class NETWORK-CONTROL
set dscp cs6
class NETWORK-MANAGEMENT
set dscp cs2
class TRANSACTIONAL-DATA
set dscp af21
class BULK-DATA
set dscp af11
class SCAVENGER
set dscp cs1
class class-default
set dscp default
```

### アプリクラスと QoS トラフィックのマッピング

次の表に、メタデータ ストリーム マップの app-class フィールドをトラフィック分類の 12 クラスの簡単な QoS モデルにマッピングする方法を示します。

## アプリケーションクラスと QoS トラフィックのマッピング

アプリケーションクラスの長いテキスト	アプリケーションクラスの短いテキスト	対応する QoS トラフィッククラス名とビジネス関連性
VOIP-TELEPHONY	VO	トラフィック クラス = voip-telephony ビジネス関連性 = YES
BROADCAST-VIDEO	BV	トラフィック クラス = broadcast-video ビジネス関連性 = YES
REALTIME-INTERACTIVE	RTI	トラフィック クラス = real-time-interactive ビジネス関連性 = YES
MULTIMEDIA-CONFERENCING	MMC	トラフィック クラス = multimedia-conferencing ビジネス関連性 = YES
MULTIMEDIA-STREAMING	MMS	トラフィック クラス = multimedia-streaming ビジネス関連性 = YES
NETWORK-CONTROL	NC	トラフィック クラス = network-control ビジネス関連性 = YES
SIGNALING	CS	トラフィック クラス = Signaling ビジネス関連性 = Yes
OPS-ADMIN-MGMT	OAM	トラフィック クラス = ops-admin-mgmt ビジネス関連性 = YES
TRANSACTIONAL-DATA	TD	トラフィック クラス = Transactional-Data ビジネス関連性 = YES
BULK-DATA	BD	トラフィック クラス = bulk-data ビジネス関連性 = YES

アプリケーションクラスの長いテキスト	アプリケーションクラスの短いテキスト	対応する Qos トラフィック クラス名とビジネス関連性
BEST-EFFORT	BE	トラフィック クラス = <no change> ビジネス関連性 = default
SCAVENGER	SCV	トラフィック クラス = <no change> ビジネス関連性 = NO

### ネットワーク制御トラフィックの分類

次に、ネットワーク制御トラフィックを分類する例を示します。維持する必要がある対応するメタデータは `CISCO-CLS=app-name:example|app-class:NC|business:YES` です。

1. クラス マップを作成し、属性を一致させます。

```

スイッチ# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
スイッチ(config)# class-map NETWORK-CONTROL
スイッチ(config-cmap)# match protocol attribute traffic-class network-control
スイッチ(config-cmap)# match protocol attribute business-relevance business-relevant
スイッチ(config-cmap)# end

```

2. ポリシー マップを作成し、それにクラス マップを付加して、優先順位を指定します。

```

スイッチ# configure terminal
スイッチ configuration commands, one per line. End with CNTL/Z.
スイッチ(config)# policy-map MARKING
スイッチ(config-pmap)# class NETWORK-CONTROL
スイッチ(config-pmap-c)# set dscp ef
スイッチ(config-pmap-c)# end

```

3. インターフェイスにポリシー マップを付加します。

```

スイッチ# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
スイッチ(config)# interface tengigabitethernet 1/0/1
スイッチ(config-if)# service-policy input MARKING
スイッチ(config-if)# end

```

## DNS-AS を使用した AVC 用 FNF の設定

FNFを使用すると、ネットワーク上で実行されているアプリケーションについての可視性が得られ、FNF オプションテンプレートを使用してアプリケーションの ID、説明、および属性の情報をエクスポートできます。DNS-AS クライアント上では、次の FNF 設定を行う必要があります。

- 非キーフィールド **application-name**、キーフィールドの **ipv4 source address** と **ipv4 destination address** を収集するためのフローレコードを設定します。
- フロー エクスポートと 2 つのオプションテンプレートを設定します。オプションテンプレートは、アプリケーションの情報を取得します。

オプションテンプレート **application-table** : DNS-AS クライアントによって解決されたアプリケーションのみをエクスポートします。つまり、バインディングテーブルからアプリケーション ID と名前のみがエクスポートされます。対応するアプリケーションの記述は、標準的なアプリケーションの Network Based Application Recognition (NBAR) からのものです。構築化されたヘルプ文字列は、カスタムアプリケーションに使用されます。

オプションテンプレート **application-attributes** は、アプリケーション名にマッピングすることによって属性情報を取得します。標準的なアプリケーション名を使用した場合、オプションテンプレートは標準的な Network Based Application Recognition (NBAR) 属性の定義が使用されます。カスタムアプリケーション名が使用された場合、ユーザー定義のアプリケーションと特定の属性フィールドのみが値を確実に伝達します。

- フロー モニターを設定し、それをインターフェイスに適用することで、ネットワークトラフィックの監視を有効にします。

DNS-AS を使用した FNF インタラクション : フローテーブルで作成されたすべてのフローで、DNS-AS クライアントは宛先 IP アドレスまたは送信元 IP アドレス (使用できない場合) を使用し、フローのアプリケーション名を解決します (バインディングテーブルにエントリが存在する場合)。

FNF は、対応するアプリケーションにマッピングされているオプションテンプレートデータを設定された間隔 (デフォルトでは 600 秒) で定期的に外部コレクタにエクスポートします。

## オプションテンプレート

**application-table** および **application-attributes** オプションテンプレートがサポートされています。オプションテンプレートにより、外部コレクタにエクスポートする情報が決定されます。

### option application-table

このテンプレートは、アプリケーション名、アプリケーションタグ、および説明を外部コレクタにエクスポートします。

DNS-AS を使用した AVC が有効になっているデバイスでは、DNS-AS クライアントによって解決されたアプリケーションのみがエクスポートされます。ただし、永続的な機能として、**application-table** テンプレートは、この機能が有効になっているかどうかにかかわらず、**unclassified** と **unknown** のアプリケーションをエクスポートします。

- アプリケーション名 : カスタムアプリケーションおよび標準アプリケーションの場合、この情報はバインディングテーブルに保存されている TXT 応答 (**app-name:**) から抽出されます。
- アプリケーションタグ : DNS-AS 機能を使用した AVC では、アプリケーション ID と同じです。エンジン ID とセレクト ID で構成されています。



- エンジン ID または分類エンジン ID : セレクタ ID のコンテキストを定義します。次の値のみがサポートされています。
  - L3 : IANA レイヤ 3 プロトコル番号 (IANA\_L3\_STANDARD、ID : 1)
  - L4 : IANA レイヤ 4 のウェルノウン ポート番号 (IANA\_L4\_STANDARD、ID : 3)
  - L7 : シスコのグローバル アプリケーション ID (CISCO\_L7\_GLOBAL、ID : 13)
  - CU : カスタム プロトコル (NBAR\_CUSTOM、ID : 6)
- セレクタ ID : アプリケーションまたは分類を一意に識別します。

標準アプリケーションの場合、アプリケーションタグ情報は次の送信元から記載されている順に抽出されます。

#### 1. TXT 応答 (app-id:)

#### 2. 標準的なアプリケーションの NBAR 定義 (TXT 応答が値を持たない場合)

カスタム アプリケーションの場合は、アプリケーションタグ情報に次が適用されます。

- TXT 応答 (app-id:) からのみ抽出されます。
- エンジン ID の場合、DNS-AS クライアントが自動的に CU (カスタム プロトコル) を使用します (NBAR\_CUSTOM、ID : 6)。
- セレクタ ID の場合、DNS-AS クライアントがカスタム セレクタ ID を割り当てます。最大 120 のカスタム アプリケーションがサポートされます。その中の 110 のカスタム アプリケーションを DNS-AS クライアントに使用できます。セレクタ ID 値 243 以降は、降順で ID が割り当てられます。割り当てる ID がなくなった場合、エントリーはバインディング テーブルに保存されません。
- 説明 : この情報は、標準アプリケーションの NBAR 定義から抽出されます。カスタム アプリケーションの場合、DNS-AS クライアントはユーザー定義のプロトコル <app-name> を使用します。

### option application-attributes

このテンプレートは、コレクタが属性にアプリケーション名を (オプションの application-table から) マッピングできるようにします。属性は、プロトコルまたはアプリケーションごとに静的に割り当てられ、トラフィックには依存しません。このテンプレートでは、次の属性がサポートされています。

標準アプリケーションの場合 :

- アプリケーション タグ : 上記の [option application-table](#) セクションのアプリケーション タグの情報を参照してください。ここでも同じことが当てはまります。

- **カテゴリ**：一致基準として、各プロトコルのカテゴリ化の最初のレベルに基づいてアプリケーションをグループ化します。類似したアプリケーションが1つのカテゴリにまとめてグループ化されます。たとえば、電子メール カテゴリには、Internet Mail Access Protocol (IMAP)、Simple Mail Transfer Protocol (SMTP)、Lotus Notes などのすべての電子メールアプリケーションが含まれます。
- **サブカテゴリ**：一致基準として、各プロトコルのカテゴリ化の2番目のレベルに基づいてアプリケーションをグループ化します。たとえば、clearcase、dbase、rda、mysql、その他のデータベースアプリケーションはデータベース グループにグループ化されます。
- **アプリケーション グループ**：同じネットワーク キング アプリケーションをまとめてグループ化します。たとえば、Example-Messenger、Example-VoIP-messenger、および Example-VoIP-over-SIP を example-messenger-group の下にまとめてグループ化します。
- **ピアツーピア (p2p)**：p2p テクノロジーを使用するかどうかに基づいてプロトコルをグループ化します。
- **トンネル**：プロトコルが他のプロトコルのトラフィックをトンネルするかどうかに基づいてプロトコルを分類します。NBAR が値を指定しないプロトコルは、未割り当てのトンネルグループに分類されます。たとえば、レイヤ2 トンネリング プロトコル (L2TP) などです。
- **暗号化**：アプリケーションの暗号化と非暗号化のステータスに基づいてアプリケーションをグループ化します。NBAR が値を指定しないプロトコルは、未割り当ての暗号化グループに分類されます。
- **トラフィックのクラス**：所属するトラフィッククラスに基づいてアプリケーションとプロトコルを分類します。たとえば、トラフィッククラス TD のすべてのアプリケーションが挙げられます。トラフィッククラス情報は、次の送信元から記載されている順に抽出されます。

1. TXT 応答 (**app-class:**)

2. 標準的なアプリケーションの NBAR 定義 (TXT 応答が値を持たない場合)

- **ビジネスの関連性**：ビジネスに関連があるとマークされているかどうかに基づいてアプリケーションをグループ化します。たとえば、ビジネス関連性が YES のすべてのアプリケーションが挙げられます。ビジネス関連性情報は、次の送信元から記載されている順に抽出されます。

1. TXT 応答 (**business:**)

2. 標準的なアプリケーションの NBAR 定義 (TXT 応答が値を持たない場合)

カスタム アプリケーションの場合：

application-attributes オプションテンプレートの次の属性のみが値を伝送することが保証されています。

- **アプリケーション タグ**：上記の [option application-table](#) セクションのアプリケーション タグの情報を参照してください。ここでも同じことが当てはまります。

- トラフィック クラス：この情報は TXT 応答 (**app-class:**) から抽出されます。
- ビジネスの関連性：この情報は TXT 応答 (**business:**) から抽出されます。

## DNS-AS を使用した AVC 用 FNF 設定の例

次に、DNS-AS を使用した AVC 用 FNF を設定する例を示します。

パート 1：フロー レコードを作成します。例に示すように設定する必要があります。

- アプリケーション名を解決するための、key フィールドとしての送信元と宛先の IP アドレス。
- フロー レコード内の nonkey フィールドとしてのアプリケーション名の使用。

さらに、フロー内のバイトまたはパケット数を nonkey フィールドとして設定して、コレクタに送信するアプリケーションの数を表示することもできます (オプション)。

```
スイッチ# configure terminal
スイッチ(config)# flow record example-record1
スイッチ(config-flow-record)# match ipv4 source address
スイッチ(config-flow-record)# match ipv4 destination address
スイッチ(config-flow-record)# collect application name
スイッチ(config-flow-record)# collect counter packets
スイッチ(config-flow-record)# exit
```

```
スイッチ# show flow record example-record1
flow record example-record1
match ipv4 source address
match ipv4 destination address
collect application name
collect counter packets
```

パート 2：フロー エクスポートを作成します。

また、**application-table** オプション テンプレートと **application-attributes** オプション テンプレートもエクスポート内に設定します。オプション テンプレートを使用しないと、コレクタは意味のあるアプリケーション情報を取得できません。少なくとも、**application-table** オプションを設定することを推奨します。属性情報の場合は、**application-attribute** オプションも設定します。

また、テンプレートをエクスポートする頻度を秒単位で変更することもできます (許容範囲は 1 ~ 86400 秒、デフォルト値は 600 秒)。

```
スイッチ(config)# flow exporter example-exporter1
スイッチ(config-flow-exporter)# option application-table
スイッチ(config-flow-exporter)# option application-attributes
スイッチ(config-flow-exporter)# template data timeout 500
スイッチ(config-flow-exporter)# exit
```

```
スイッチ# show flow exporter example-exporter1
Flow Exporter example-exporter1:
Description:                User defined
Export protocol:            NetFlow Version 9
Transport Configuration:
```

```

Destination IP address: 192.0.1.254
Source IP address:      192.51.100.2
Transport Protocol:    UDP
Destination Port:      9995
Source Port:           54964
DSCP:                  0x0
TTL:                   255
Output Features:       Not Used
Options Configuration:
  application-table (timeout 500 seconds)
  application-attributes (timeout 500 seconds)

```

```

スイッチ# show flow exporter example-exporter1 statistics
Flow Exporter example-exporter1:
  Packet send statistics (last cleared 00:00:48 ago):
    Successfully sent:      2                (924 bytes)

  Client send statistics:
    Client: Option options application-name
      Records added:        4
      - sent:                4
      Bytes added:          332
      - sent:                332

    Client: Option options application-attributes
      Records added:        2
      - sent:                2
      Bytes added:          388
      - sent:                388

```

パート 3 : フロー モニターを作成します。

フローモニターをインターフェイスに適用し、ネットワークトラフィックの監視を実行します。

また、同じインターフェイスに QoS ポリシーも適用できます。次のれいでは、同じ QoS 設定の一部として作成された QoS ポリシーを適用しています ([ネットワーク制御トラフィックの分類 \(15 ページ\)](#))。

```

スイッチ# configure terminal
スイッチ(config)# flow monitor example-monitor1
スイッチ(config-flow-monitor)# record example-record1
スイッチ(config-flow-monitor)# exporter exporter-export1
スイッチ(config-flow-monitor)# exit
スイッチ(config)# interface tengigabitethernet 1/0/1
スイッチ(config-if)# switchport access vlan 100
スイッチ(config-if)# switchport mode access
スイッチ(config-if)# ip flow monitor example-monitor1 input
スイッチ(config-if)# service-policy input MARKING
スイッチ(config-if)# end

スイッチ# show flow monitor
flow monitor example-monitor1
  record example-record1
  exporter example-exporter1
!
スイッチ# show interface tengigabitethernet1/0/1
interface tengigabitethernet1/0/1
  switchport access vlan 100

```

```

switchport mode access
ip flow monitor example-monitor1 input

スイッチ# show flow monitor example-monitor1 cache
Cache type: Normal
Cache size: 16640
Current entries: 3
High Watermark: 3

Flows added: 6
Flows aged: 3
- Active timeout ( 1800 secs) 0
- Inactive timeout ( 30 secs) 3
- Event aged 0
- Watermark aged 0
- Emergency aged 0

IPV4 SOURCE ADDRESS: 192.0.1.254
IPV4 DESTINATION ADDRESS: 192.51.100.2
counter packets long: 7479
application name: appexample1

IPV4 SOURCE ADDRESS: 192.51.100.11
IPV4 DESTINATION ADDRESS: 203.0.113.125
counter packets long: 445
application name: appexample2

IPV4 SOURCE ADDRESS: 192.51.51.51
IPV4 DESTINATION ADDRESS: 203.0.113.100
counter packets long: 14325
application name: appexample3
Switch#

```

#### パート 4 : その他の関連 show コマンド

```

スイッチ# show avc dns-as client binding-table detail
DNS-AS generated protocols:
Max number of protocols :50
Customization interval [min] :N/A

Age : The amount of time that the entry is active
TTL : Time to live which was learned from DNS-AS server
Time To Expire : Entry expiration time in case device does not see DNS traffic for
the entry host

Protocol-Name : appexample1
VRF : <default>
Host : www.appexample1.com
Age[min] : 2
TTL[min] : 60
Time To Expire[min] : 58
TXT Record : app-name:appexample1|app-class:VO|business:YES
Traffic Class : voip-telephony
Business Relevance : business relevant
IP : 192.0.1.254

Protocol-Name : appexample2
VRF : <default>
Host : www.appexample2.com
Age[min] : 2
TTL[min] : 60
Time To Expire[min] : 58

```

```

TXT Record      : app-name:appexample2|app-class:VO|business:YES
Traffic Class   : voip-telephony
Business Relevance : business relevant
IP              : 192.51.100.11

```

<output truncated>

スイッチ# **show flow exporter option application engines**

```

Engine: prot (IANA_L3_STANDARD, ID: 1)
Engine: port (IANA_L4_STANDARD, ID: 3)
Engine: NBAR (NBAR_CUSTOM, ID: 6)
Engine: cisco (CISCO_L7_GLOBAL, ID: 13)

```

スイッチ# **show flow exporter option application table**

```

Engine: prot (IANA_L3_STANDARD, ID: 1)
appID  Name      Description
-----

```

```

Engine: port (IANA_L4_STANDARD, ID: 3)
appID  Name      Description
-----

```

Engine: NBAR (NBAR\_CUSTOM, ID: 6)

```

appID  Name      Description
-----

```

```

6:28202 appexample1 User defined protocol appexample1

```

Engine: cisco (CISCO\_L7\_GLOBAL, ID: 13)

```

appID  Name      Description
-----

```

```

13:0 unclassified Unclassified traffic

```

```

13:1 unknown      Unknown application

```

```

13:518 appexample2 appexample2, social web application and service

```

## DNS-AS を使用した AVC の監視

設定した、さまざまな DNS-AS を使用した AVC 設定を表示するには、特権 EXEC モードで次のコマンドを使用します。

表 1: DNS-AS を使用した AVC の監視コマンド

コマンド	目的	出力の例
<b>show avc dns-as client status</b>	DNS-AS クライアントの現在のステータスを表示します。このコマンドを使用すると、DNS-AS を使用した AVC が有効になっているかどうかを知ることができます。	例 : <a href="#">show avc dns-as client status</a>
<b>show avc dns-as client trusted-domains</b>	バインディング テーブルに維持されている信頼ドメインのリストを表示します。	例 : <a href="#">show avc dns-as client trusted-domains</a>

コマンド	目的	出力の例
<b>show avc dns-as client binding-table</b> および <b>show avc dns-as client binding-table detail</b>	信頼ドメインと解決済みエントリのリスト用の DNS-AS を使用した AVC のメタデータを表示します。アプリケーション名やドメイン名などで、出力をフィルタリングできます。  どちらのコマンドも、異なる形式で同じ情報を表示します。	<a href="#">例 : show avc dns-as client binding-table</a>
<b>show avc dns-as client statistics</b>	パケット ロギング情報（送信した DNS クエリの数と受信した応答の数）を表示します。	<a href="#">例 : show avc dns-as client statistics</a>
<b>show avc dns-as client name-server brief</b>	メタデータ要求の送信先の DNS サーバーに関する情報を表示します。	<a href="#">例 : show avc dns-as client name-server brief</a>
<b>show ip name-server</b>	維持されているすべてのネームサーバーの IP アドレスを表示します。	<a href="#">例 : show ip name-server</a>
<b>show platform tcam utilization</b>	TCAM の可用性に関する情報を表示します。	<a href="#">例 : show platform tcam utilization</a>

例 : show avc dns-as client status

```
スイッチ# show avc dns-as client status
DNS-AS client is enabled
```

[表 1 : DNS-AS を使用した AVC の監視コマンドに戻る](#)

例 : show avc dns-as client trusted-domains

```
スイッチ# show avc dns-as client trusted-domains
Id | Trusted domain
-----
1 | example.com
2 | www.example.com
3 | example.net
4 | www.example.net
5 | example.org
6 | www.example.org
```

[表 1 : DNS-AS を使用した AVC の監視コマンドに戻る](#)

例 : show avc dns-as client binding-table

```

スイッチ# show avc dns-as client binding-table
Switch# show avc dns-as client binding-table detailed
DNS-AS generated protocols:
Max number of protocols :50
Customization interval [min] :N/A

Age : The amount of time that the entry is active
TTL : Time to live which was learned from DNS-AS server
Time To Expire : Entry expiration time in case device does not see DNS traffic for the
entry host

Protocol-Name : example
VRF : <default>
Host : www.example.com
Age[min] : 2
TTL[min] : 60
Time To Expire[min] : 58
TXT Record : app-name:example|app-class:VO|business:YES
Traffic Class : voip-telephony
Business Relevance : business relevant
IP : 192.0.2.121
   : 192.0.2.254
   : 198.51.100.1
   : 198.51.100.254
   : 192.51.100.12
   : 203.0.113.125
<output truncated>

```

表 1 : DNS-AS を使用した AVC の監視コマンドに戻る

例 : show avc dns-as client statistics



(注) この例では、2 つの DNS サーバーが設定されます。

```

スイッチ# show avc dns-as client statistics
Server details: vrf-id = 0 vrf-name = <default> ip = 192.0.2.1
AAAA Query Error packets 0
AAAA Query TX packets 0
AAAA Response RX packets 0
TXT Query Error packets 0
TXT Query TX packets 8
TXT Response RX packets 0
A Query Error packets 0
A Query TX packets 6
A Response RX packets 0
Server details: vrf-id = 0 vrf-name = <default> ip = 192.0.2.2
AAAA Query Error packets 0
AAAA Query TX packets 0
AAAA Response RX packets 0
TXT Query Error packets 0
TXT Query TX packets 2
TXT Response RX packets 2
A Query Error packets 0
A Query TX packets 4
A Response RX packets 2
Total Drop packets 0

avc_dns_as_pkts_logged = 2
avc_dns_as_q_pkts_processed = 2

```



[表 1 : DNS-AS を使用した AVC の監視コマンドに戻る](#)

例 : show avc dns-as client name-server brief

スイッチ# **show avc dns-as client name-server brief**

```
Server-IP | Vrf-name
-----
192.0.2.1 | <default>
192.0.2.2 | <default>
```

[表 1 : DNS-AS を使用した AVC の監視コマンドに戻る](#)

例 : show ip name-server

スイッチ# **show ip name-server**

```
192.0.2.1
192.0.2.2
2001:DB8::1
```

[表 1 : DNS-AS を使用した AVC の監視コマンドに戻る](#)

例 : show platform tcam utilization



(注) 関連する TCAM エントリは IPv4 qos aces: です。

スイッチ# **show platform tcam utilization**

```
CAM Utilization for ASIC# 0 Max Used
Masks/Values Masks/values

Unicast mac addresses: 16604/16604 24/24
IPv4 IGMP groups + multicast routes: 1072/1072 3/3
IPv4 unicast directly-connected routes: 4096/4096 4/4
IPv4 unicast indirectly-connected routes: 1280/1280 40/40
IPv6 Multicast groups: 1072/1072 18/18
IPv6 unicast directly-connected routes: 4096/4096 1/1
IPv6 unicast indirectly-connected routes: 1280/1280 32/32
IPv4 policy based routing aces: 512/512 14/14
IPv4 qos aces: 512/512 51/51
IPv4 security aces: 1024/1024 78/78
IPv6 policy based routing aces: 256/256 8/8
IPv6 qos aces: 256/256 44/44
IPv6 security aces: 512/512 18/18
```

Note: Allocation of TCAM entries per feature uses a complex algorithm. The above information is meant to provide an abstract view of the current TCAM utilization

[表 1 : DNS-AS を使用した AVC の監視コマンドに戻る](#)

## DNS-AS を使用した AVC のトラブルシューティング

問題	考えられる原因と解決策
バインディングテーブルにエントリがない。	<p>バインドテーブルが、次の理由のいずれか、または両方によって空になっている可能性があります。</p> <ul style="list-style-type: none"> <li>• DNS サーバーでメタデータが維持されていない — 次のタスクを完了してください： <a href="#">メタデータ ストリームの生成 (7 ページ)</a></li> <li>• 信頼ドメインリストでエントリが維持されていない — 次のタスクを完了してください： <a href="#">信頼ドメインのリストの維持 (11 ページ)</a></li> </ul>
DNS スヌーピングまたはパケット ロギングに失敗する。	<p>DNS スヌーピングおよびパケット ロギングを確実に実行するには、ポリシーマップ（トラフィック クラスを決定する関連クラス マップが含まれている）をインターフェイスに付加する必要があります。次の例を参照してください： <a href="#">DNS-AS を使用した AVC 用 QoS の設定 (12 ページ)</a></p>
DNS サーバーが不正な値を返す。	<p>正しい DNS-AS メタデータが DNS システムに維持されていることを確認します。</p> <ul style="list-style-type: none"> <li>• Linux の dig を次のように使用します。 <pre>dig TXT +short www.example.org [dns-server-ip] "CISCO-CLS=app-name:example app-class:TD business:YES app-id:CU/28202"</pre> </li> <li>• Windows nslookup を次のように使用します。 <pre>C:\Windows\system32&gt;NSLookup.exe -q=TXT www.example.org [dns-server-ip] www.example.org text = "CISCO-CLS=app-name:example app-class:TD business:YES app-id:CU/28202"</pre> </li> </ul>
適用した QoS ポリシーがポートから削除されている。	<p>DNS-AS クライアントがアプリケーションを認識し、「A」レコード応答がバインディングテーブルに保存されると、システムは TCAM を使用してそのアプリケーションの IP アドレスを保存します。事実上、単一のアプリケーションが複数の IP アドレスを持つことができ、各アプリケーションが TCAM のスペースをさらに使用します。TCAM が枯渇すると、QoS ポリシーは適用を停止します。</p> <p>この問題を回避するには、定期的に TCAM の使用率を監視します。TCAM の可用性に関する情報を表示するには、<b>show platform tcam utilisation</b> コマンドを特権 EXEC モードで入力します。</p>

問題	考えられる原因と解決策
DNS-AS クライアントが、定義した QoS マッピングを無視し、デフォルトの転送動作を適用します。	<p>次の場合に、DNS-AS クライアントが、QoS マッピングを無視し、デフォルトの転送動作を適用します。</p> <ul style="list-style-type: none"> <li>• トラフィッククラスとビジネス関連性に指定した一致属性が、メタデータストリームに定義したものと一致しない場合、必要に応じて修正してください。</li> <li>• バインディングテーブル エントリがアクティブでなくなっている場合、これはエントリの経過時間を意味します。エントリの経過時間を表示するには、<b>show avc dns-as client binding-table</b> コマンドを使用します。</li> </ul>

## DNS-AS を使用した AVC の機能履歴および情報

次の表に、この章で説明した機能に関するリリース情報を示します。この表は、ソフトウェアリリース トレーンで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェアリリースでもサポートされます。

リリース	変更内容
Cisco IOS リリース 15.2(5)E1	この機能が導入されました。 このリリース以降、この機能は Cisco Catalyst 3560-CX シリーズ スイッチでのみサポートされ、Cisco Catalyst 2960-CX シリーズ スイッチではサポートされません。
Cisco IOS リリース 15.2(5)E2	DNS-AS を使用した AVC 向けに Flexible Netflow (FnF) が導入され、FnF を使用してアプリケーション情報をエクスポートできるようになりました。



## 翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。