スタンドアロン Content Engine の配置シナリオ

この章では、企業環境およびサービス プロバイダー環境でスタンドアロン Content Engine を配置するためのシナリオの例をいくつか説明します。この章の内容は、次のとおりです。

（注） Content Engine を Content Distribution Manager に登録済みのデバイスとして配置する方法については、『*Cisco ACNS Software Configuration Guide for Centrally Managed Deployments*』Release 5.4 を参照してください。

配置するサービスの決定

ネットワークのエッジにコンテンツをプッシュして、コンテンツ配信を高速化し、WAN 帯域幅の使用量を最適化します。これを実行するためのプロセスを、*コンテンツ キャッシング*と呼びます。コンテンツ キャッシングは、*ネットワーク キャッシング*とも呼ばれます。Contet Engine は、Web クライアントとインターネット間の*インライン デバイス*という特殊な位置付けなので、Content Engine は簡単にコンテンツ キャッシング エンジンとして配置できます。頻繁にアクセスされるインターネット コンテンツを、各サイトの Content Engine でローカルにキャッシュして配信することにより、帯域幅の使用量と Web の遅延を大幅に減少できます。

（注）　既存のプロキシ インフラストラクチャと統合するために、ACNS ソフトウェアは、FTP、HTTPS、HTTP 1.0、および HTTP 1.1 を含む、多数のプロキシ プロトコルをサポートしています。サポート対象のネットワーク プロトコルについては、表 B-1 を参照してください。

Content Engine で配置できるサポート対象サービスのタイプは、Content Engine への要求のルーティング方式によって異なります。コンテンツ要求は、直接（直接プロキシ ルーティング）あるいは WCCP ルータまたはレイヤ 4 CSS スイッチ（透過リダイレクション）を使用して、クライアントから Content Engine にルーティングします。

（注）　直接プロキシ　ルーティングまたは透過リダイレクションが、クライアント要求をスタンドアロン Content Engine に転送する場合にサポートされるキャッシングおよびストリーミング サービスのリストについては、表 1-5 および表 1-6 を参照してください。

*直接プロキシ ルーティング方式*が最も簡単で、最も単純なルーティング方式です。直接プロキシ ルーティングでは、クライアント デスクトップ（クライアント ブラウザとメディア プレーヤー）を設定して、そのコンテンツ要求をプロキシ サーバとして機能する特定の Content Engine に直接送信する必要があります。クライアント要求は、クライアントのプロキシ サーバとして設定された Content Engine に直接送信されます。このルーティング方式は、ユーザ デスクトップが厳密に制御される場合に通常使用されます。

*透過リダイレクション方式*の場合、ネットワーク トポロジ およびトラフィック パターンを理解している必要があります。クライアントのデスクトップ設定を変更する必要がないので、企業では通常、透過リダイレクション方式が優先されます。

ただし、クライアントのデスクトップの設定を変更する必要があっても、従来の要件を満たすために直接プロキシ ルーティングが必要になることがあります。また、営業所の WCCP ルータまたはスイッチに必要な設定変更が許可されないために、特定のサービス（HTTPS プロキシ キャッシングなど）に対して直接プロキシ ルーティングを使用しなければならない場合もあります。

ここでは、ACNS 5.x ソフトウェアを実行しているスタンドアロン Content Engine によってサポートされるルーティング方式、および関連するキャッシング サービスとストリーミング サービスについて説明します。

（注）　スタンドアロン Content Engine は、直接プロキシ ルーティング、および WCCP ルーティングおよびレイヤ 4 スイッチングによる透過リダイレクションをサポートします。ただし、ルーティング方式としてコンテンツ ルーティングを使用する場合には、Content Engine を Content Distribution Manager に登録する必要があります。スタンドアロン Content Engine は Content Distribution Manager に登録されないので、コンテンツ ルーティングをサポートできません。コンテンツ ルーティングの詳細は、『*Cisco ACNS Software Configuration Guide for Centrally Managed Deployments*』Release 5.4 を参照してください。

非透過モードでのキャッシングおよびストリーミング サービスの配置

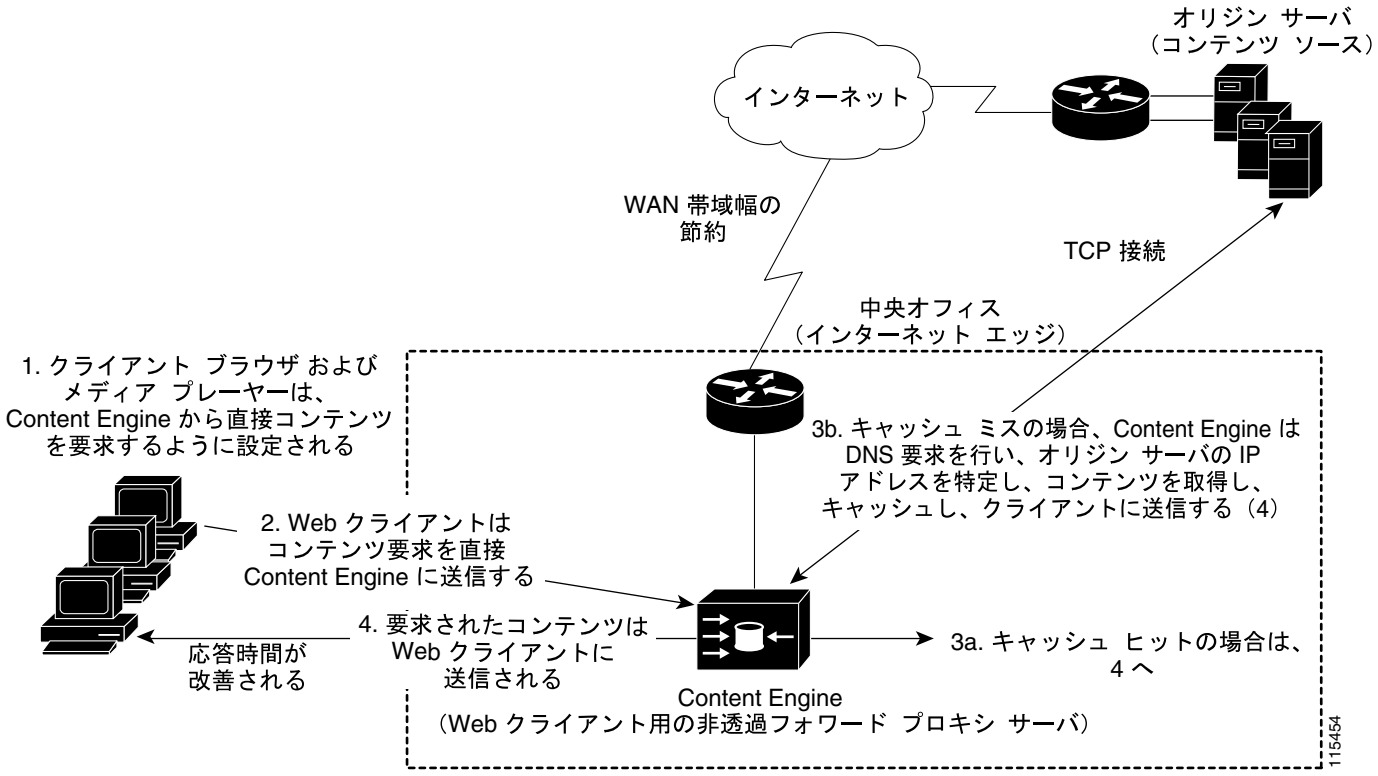
直接プロキシ ルーティングの場合、すべてのブラウザまたはメディア プレーヤーの Web コンテンツ要求の宛先が、スタンドアロン Content Engine になります。これらの要求を、*プロキシスタイル要求*と呼びます。プロキシスタイル要求には Content Engine と同じ宛先 IP アドレスが付加され、Web クライアントにより Content Engine（フォワード プロキシ サーバ）に直接ルーティングされます。

非透過フォワード プロキシ キャッシングの概要

直接プロキシ ルーティングを使用してコンテンツ要求を Content Engine にルーティングする配置の場合（図 3-1 を参照）、Content Engine は Web クライアントの代わりにコンテンツを取得する最適化されたネットワーク ゲートウェイ デバイスとして動作します。

- 要求されたコンテンツが Content Engine のローカル ストレージに保管されている場合（キャッシュ ヒット）、Content Engine はそのコンテンツを Web クライアントに送信します。
- 要求されたコンテンツが　Content Engine　のローカル　キャッシュに保管されていない場合（キャッシュ ミス）、Content Engine はオリジン サーバから要求されたコンテンツを取得し、キャッシングできるコンテンツの場合にはコンテンツのローカル コピーを保管し、要求されたコンテンツを Web クライアントに送信します。Content Engine は、これ以降同じコンテンツに対する要求があった場合、ローカル ストレージからコンテンツを配信します。

図 3-1　スタンドアロン Content Engine を使用した非透過フォワード プロキシ キャッシング



通常、直接プロキシ ルーティングはサービス プロバイダー環境ではなく、企業環境で実装されています。このタイプのキャッシングは、クライアント デスクトップの変更が必要になるからです。直接プロキシ ルーティングでは、クライアント ブラウザおよびメディア プレーヤーが、クライアントのフォワード（非透過）プロキシとして動作している Content Engine を明示的に宛先とするように設定する必要があります。詳細は、「直接プロキシ ルーティング用のクライアント ブラウザとメディア プレーヤーの設定」（p.4-38）を参照してください。

非透過（プロキシ）キャッシング サービスをスタンドアロン Content Engine に配置すると、次のような大きな利点があります。

- エンド ユーザのゲートウェイ デバイスの役割をする Content Engine で、ユーザ グループのインターネット アクセスを規制できる
- すべてのインターネット要求がプロキシ キャッシュ（Content Engine）から発信されているように見えるので、内部ネットワーク アドレスを隠蔽できる
- 頻繁に要求されるキャッシング可能なコンテンツをローカルにキャッシングするため、WAN 帯域幅が節約され、Web クライアントへのコンテンツ配信が高速化される

ACNS ソフトウェア リリース 5.2 以降には、スタンドアロン Content Engine の基本設定（デバイス ネットワーク設定、ディスク設定、通常使用するキャッシング サービスのセット）を効率的に行えるように、Setup ユーティリティが追加されています。この基本設定には、一般的に使用するキャッシング サービスのセットが含まれます。Setup ユーティリティで設定できるキャッシング サービスのリストについては、表 4-2 を参照してください。

非透過モード サービスの着信プロキシ ポートの設定

プロキシ モードで、フォワード プロキシ サーバとして機能している Content Engine は、FTP、HTTP、HTTPS、MMS、および RTSP 要求を待ち受ける着信ポートを最大 8 つまでサポートします。*プロキシ ポート*とは、Content Engine がプロキシスタイル要求を受信し、要求されたコンテンツをクライアントに戻すポートを意味します。着信プロキシ ポートは、透過モード サービス（HTTP 透過キャッシングなど）で使用するポートと同じポートでかまいません。Content Engine の着信プロキシ ポートは、Content Engine で実行中の WCCP サービスをいずれも停止することなく変更できます。

非透過モード サービス（HTTP プロキシ キャッシング、FTP-over-HTTP プロキシ キャッシング、RealMedia プロキシ キャッシング、WMT プロキシ キャッシングなど）の設定の一部として、次の作業をする必要があります。

- クライアント ブラウザまたはメディア プレーヤーが、すべての要求を、Content Engine の着信プロキシ ポートへの要求として転送するように設定します。
- Content Engine が、着信プロキシ ポートでクライアント要求を待ち受けるように設定します。

クライアント ブラウザ、メディア プレーヤー、および Content Engine に着信プロキシ ポートを設定すると、スタンドアロン Content Engine は、クライアント ブラウザまたはメディア プレーヤーから非透過（プロキシ スタイル）要求を直接受け入れます。

クライアント ブラウザまたはメディア プレーヤーの宛先を特定の Content Engine に設定する方法については、表 3-1 を参照してください。

表 3-1　コンテンツ要求の直接プロキシ ルーティングをサポートする、クライアント ブラウザおよびメディア プレーヤーの設定

非透過キャッシング	詳細
HTTP プロキシ キャッシング	スタンドアロン Content Engine をクライアント ブラウザの直接の宛先として指定
WMT MMS 要求の WMT プロキシ キャッシング	Windows Media Player に WMT MMS 要求の直接の宛先としてスタンドアロン Content Engine を指定
WMT RTSP 要求の WMT プロキシ キャッシング	Windows Media 9 Player に WMT RTSP 要求の直接の宛先としてスタンドアロン Content Engine を指定
RealMedia プロキシ キャッシング	スタンドアロン Content Engine を RealMedia Player の直接の宛先として指定

http、**https**、**ftp**、および **rtsp** のグローバル コンフィギュレーション コマンドの **proxy incoming** オプションは、プロトコルごとに最大 8 つのポートをサポートします。着信プロキシ ポートは、コマンドラインの 1 行または複数行で最大 8 つ指定できます。HTTP、FTP、HTTPS、MMS、および RTSP プロトコルのプロキシ スタイル要求は、同一着信プロキシ ポートで受信できます。

（注）　透過要求とプロキシ スタイル要求は、同一ポートで処理できます。

次に、クライアント ブラウザからの HTTP、HTTPS、および FTP-over-HTTP のプロキシ要求をポート 81、8080、および 8081 で直接受信する、スタンドアロン Content Engine の設定例を示します。

```
ContentEngine(config)# http proxy incoming 81 8080 8081
ContentEngine(config)# https proxy incoming 81 8080 8081
ContentEngine(config)# ftp-over-http proxy incoming 81 8080 8081
```

ACNS ソフトウェア リリース 5.3.1 では、FTP-over-HTTP キャッシングと FTP ネイティブ キャッシングを明確に区別するため、**ftp** キーワードが **ftp-ver-http** キーワードと **ftp-native** キーワードに変更されました。したがって、ACNS ソフトウェア リリース 5.3.1 では、**ftp proxy incoming** グローバル コンフィギュレーション コマンドは、**ftp-over-http proxying incoming** および **ftp-native proxy incoming** グローバル コンフィギュレーション コマンドになります。ACNS ソフトウェア リリース 5.3.1 で追加された **ftp-native proxy incoming** コマンドは、FTP クライアント（Reflection X または WS-FTP クライアントなど）からの FTP ネイティブ要求用の着信ポートを設定し、非透過 FTP ネイティブ キャッシングをサポートする場合に使用します。非透過 FTP ネイティブ キャッシングの詳細は、「非透過 FTP ネイティブ キャッシングの設定」（p.7-44）を参照してください。

次に、特定のポートで着信 WMT トラフィックを待ち受けるために、スタンドアロン Content Engine に着信プロキシ ポートを設定する例を示します。これらの WMT 要求は、Content Engine を直接の宛先にするように設定されたクライアントの Windows Media Player から Content Engine（非透過フォワード プロキシ）に直接送信されます。デフォルトの WMT ポートは 1755 で、有効なポート番号は、1 ～ 65535 です。

```
ContentEngine(config)# wmt port incoming portnumber
```

次に、特定のポートで着信 RTSP トラフィックを待ち受けるために、スタンドアロン Content Engine に着信プロキシ ポートを設定する例を示します。これらの RTSP 要求は、Content Engine を直接の宛先とするように設定されたクライアントのメディア プレーヤーから Content Engine（非透過フォワード プロキシ）に直接送信されます。デフォルトの RTSP ポートは 554 で、有効なポート番号は、1 ～ 65535 です。

```
ContentEngine(config)# rtsp port incoming portnumber
```

HTTP、HTTPS、FTP、RTSP、および MMS の着信プロキシ サービスをディセーブルにするには、**no** *protocol* **proxy incoming** グローバル コンフィギュレーション コマンド（**no wmt proxy incoming** グローバル コンフィギュレーション コマンドなど）を使用します。プロキシ モードでポートを追加または削除するには、新規にコマンドを入力して、使用するすべてのポートを指定してください。

SSL トンネリングおよび非透過キャッシングの配置

SSL トンネリング プロトコルを使用して、プロキシ サーバをエンド ユーザとオリジン サーバ間のトンネルとして動作させることができます。クライアントは、HTTP 要求により SSL トンネルを要求します。これにより、Content Engine は https://*url* 形式の CONNECT メソッド要求を処理し、HTTP 経由で SSL トンネリングを提供します。

ヒント　ブラウザが HTTPS-over-HTTP 要求を開始するのは、プロキシが明示的に設定されている場合だけです。ブラウザにプロキシが明示的に設定されていない場合、ブラウザは HTTP-over-SSL 接続を開始します。これは TCP ポート 443 上で行われるので Content Engine が ACNS ソフトウェア リリース 5.1.5 以降を実行している場合に限り、要求が Content Engine により代行受信されます。

ポート 443 上の SSL はエンドツーエンドの暗号化を使用するので、クライアントとオリジン サーバの間の透過デバイスが認識するのはランダム バイトのストリームだけです。

透過モードでのキャッシングおよびストリーミング サービスの配置

ACNS 5.x ソフトウェアでは、スタンドアロン Content Engine で、WCCP ルータおよびレイヤ 4 スイッチから透過的にリダイレクトされたコンテンツ要求を処理できます。透過リダイレクションの場合、スタンドアロン Content Engine は、Web クライアントまたは Web サーバ（本社のサーバファーム内の Web サーバなど）に対して、透過（見えない）プロキシ サーバとして機能します。Content Engine は、Web クライアントのプロキシとして動作する場合、透過フォワード プロキシ サーバになります。Content Engine が Web サーバのプロキシとして動作する場合には、透過リバース プロキシ サーバになります。

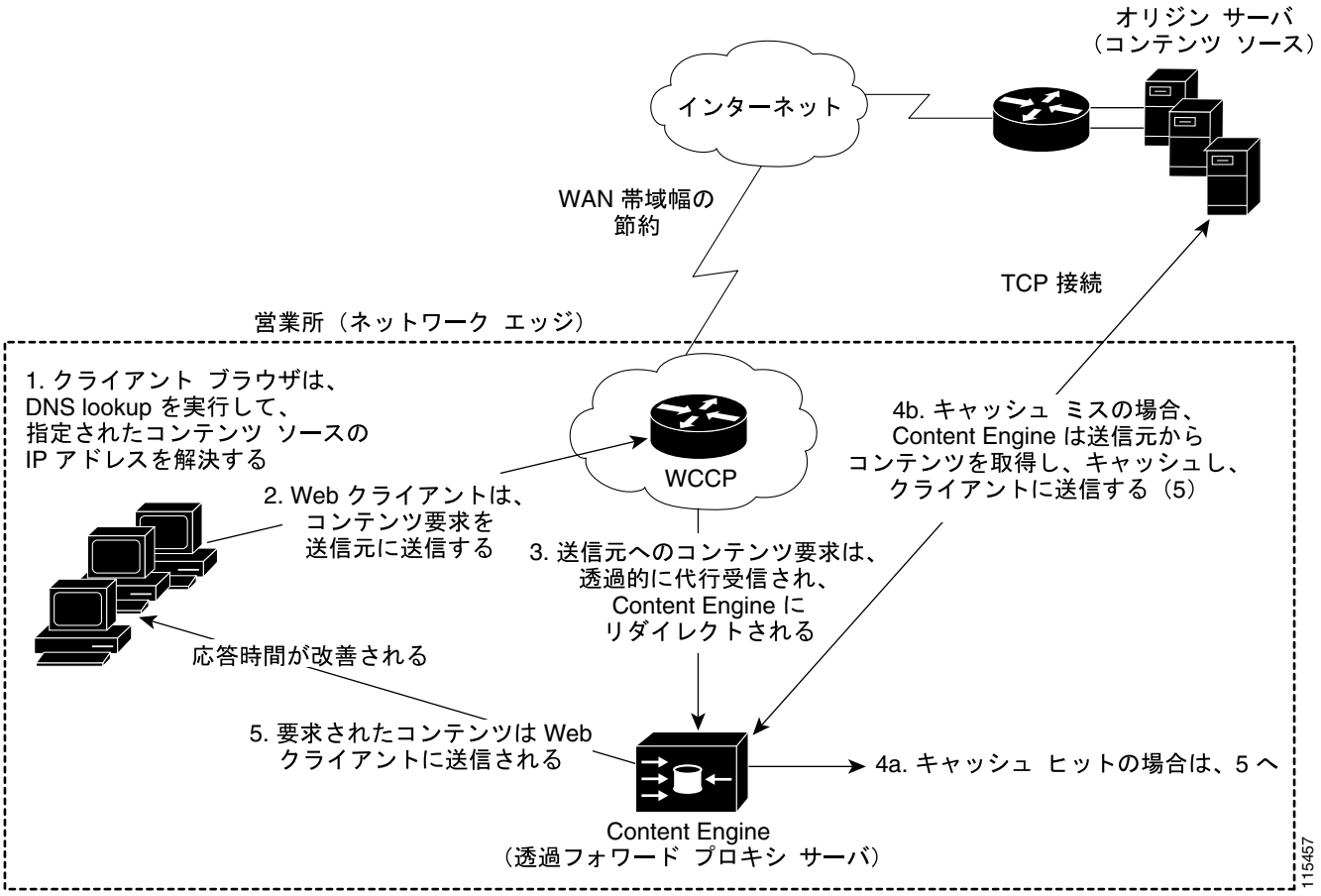
透過フォワード プロキシ キャッシングの概要

スタンドアロン Content Engine を透過キャッシュ（透過フォワード プロキシ サーバ）として配置する場合、Content Engine を Web クライアントとオリジン サーバの間のユーザ グループの近くに置きます。透過キャッシュは、ネットワーク トラフィックのパス上の、すべての出力トラフィックの通過が保証された場所に置きます。

- ネットワーク エッジ（営業所）
- インターネット エッジ（地域本部または本社）

透過リダイレクションの場合、Web クライアントはコンテンツを直接送信先（オリジン サーバ）から要求します。ただし、これらの要求はネットワーク出力ポイントで透過的に代行受信されます。これらの要求は、ネットワーク デバイス（WCCP ルータまたはレイヤ 4 Cisco Content Services Switch [CSS] スイッチ）によって透過的に代行受信され、クライアントの透過キャッシング エンジンとして機能しているスタンドアロン Content Engine にリダイレクトされます。図 3-2 に、WCCP ルータを使用した透過フォワード プロキシ キャッシングの例を示します。

図 3-2　スタンドアロン Content Engine と WCCP ルータを使用した透過フォワード プロキシ キャッシング



（注）　レイヤ 4 スイッチまたは WCCP 対応ルータを使用したフォワード プロキシ キャッシングの実装方法の詳細は、「透過リバース プロキシ キャッシングの概要」（p.3-12）および「透過リバース プロキシ キャッシングの概要」（p.3-12）を参照してください。

スタンドアロン Content Engine は、WCCP Version 2 をサポートするか、またはレイヤ 4 CSS スイッチと相互運用することにより、コンテンツ トラフィックの透過的受信、フォールト トレランス、およびスケーラブルなクラスタなど、基本レベルの透過性を実現できます。

Content Engine は、クライアントおよびネットワーク オペレーションに対して透過的なので、階層的に複数のネットワーク サイトにスタンドアロン Content Engine を容易に配置できます。たとえば、ISP では、インターネットへの主要なアクセス ポイントに スタンドアロン Content Engine（Content Engine A）を配置すると、要求されたコンテンツをインターネットを経由せずに主要アクセス ポイントで取得できるので、Point of Presence（POP）のすべての利点を活用できます。

特定の Web クライアントへのサービスをさらに改善するには、ISP の各 POP にスタンドアロン Content Engine（Content Engine B、C、および D）を配置します。この場合、クライアントがインターネットにアクセスすると、要求はまず、POP の Content Engine にリダイレクトされます。POP Content Engine（Content Engine B、C、および D）がローカル ストレージで要求を満たせない場合、

オリジン サーバに通常の Web 要求を行います。この要求は、アップストリームに送信され、Content Engine A にリダイレクトされます。要求が Content Engine A で満たされると、インターネットのメイン アクセス リンク上のトラフィックは回避され、オリジン サーバへの要求が減少し、クライアントへのネットワーク応答時間が短くなります。同様に、企業ネットワークにも、この階層化透過アーキテクチャを適用できます。

透過キャッシングの配置には、ネットワーク トポロジおよびトラフィック パターンの理解が必要ですが、スタンドアロン Content Engine で透過モード サービスを配置すると、次のような大きな利点があります。

- エンド ユーザの設定が不要 — デスクトップ設定を変更する必要がありません。
- フェールセーフ オペレーション — キャッシュは自動的にフォールト トレラントかつフェールセーフになります。キャッシュ障害が発生しても、エンド ユーザへの DoS は発生しません。
- スケーラビリティ — 複数のキャッシュを配置することにより（キャッシュ クラスタまたは階層化キャッシュ）、キャッシュ サービスを拡張できます。
- 自動バイパス — エンド ユーザの認証が必要なサイト、または HTTP 標準に準拠しないサイトは、自動的に透過キャッシュをバイパスします。

レイヤ 4 スイッチを使用した透過リダイレクションおよびフォワード プロキシ キャッシング

透過リダイレクションにレイヤ 4 スイッチングを使用すると、レイヤ 4 CSS スイッチがコンテンツ要求を透過的に代行受信し、Content Engine にリダイレクトします。CCS スイッチによる透過的な代行受信では、ユーザは、オリジン Web サーバに対する要求が レイヤ 4 CCS スイッチにより Content Engine にリダイレクトされることを認識しません。レイヤ 4 CSS スイッチは、要求をダイナミックに分析し、要求されたコンテンツがキャッシュ可能かどうかを判断するように設定できます。要求されたコンテンツがキャッシュ不可の場合、レイヤ 4 CSS スイッチは、その要求をオリジン サーバに直接送信します。要求されたコンテンツがキャッシュ可能であれば、レイヤ 4 CSS スイッチはその要求を Content Engine に送信します。Content Engine は、ローカル コピーが存在すれば、要求されたコンテンツを戻します。または、新しいコンテンツに対する要求であれば、要求をオリジン Web サーバに送信します。

レイヤ 4 スイッチングを使用してコンテンツ要求を透過的にスタンドアロン Content Engine にリダイレクトする場合、TCP SYN パケットは、レイヤ 4 リダイレクション機能がオンになっているレイヤ 4 CSS スイッチを通過すると、スイッチに接続している Content Engine に転送されます。この場合、レイヤ 4 CSS スイッチは TCP SYN パケットの MAC アドレスを変更し、ゲートウェイまたはオリジン サーバに送信する代わりに、Content Engine の MAC アドレスに変更します。パケットは、レイヤ 4 CSS スイッチにより Content Engine に送信されます。これらの動作はすべて、ハードウェアで実行されます。

Content Engine は、パケットの IP アドレスが Content Engine のアドレスではない場合でも、送信された要求を受け入れるように設定しておきます。Content Engine は、WCCP でリダイレクトされるパケットの処理方法と同様に、TCP SYN パケットを処理します。

スタンドアロン Content Engine およびレイヤ 4 CSS スイッチを使用した透過フォワード プロキシ キャッシングは、次のように実行されます。

1. ユーザ（Web クライアント）がブラウザから Web ページを要求します。
2. レイヤ 4 CCS スイッチが要求を分析し、要求されたコンテンツがキャッシュ可能であるかどうかを判断します。要求されたコンテンツがキャッシュ可能であれば、レイヤ 4 CCS スイッチは要求を Content Engine に透過的にリダイレクトします。

（注）　透過キャッシュ構成に使用可能な Content Engine が存在しない場合、レイヤ 4 CSS スイッチは、すべてのクライアント要求をオリジン Web サーバに送信します。

3. 要求されたコンテンツが Content Engine のローカル キャッシュに保管されていれば、クライアントにそのコンテンツが戻されます。
4. 要求されたコンテンツが Content Engine に保管されていない場合には、次のイベントが発生します。
   a. Content Engine は、コンテンツを取得するために、オリジン Web サーバと別個の TCP 接続をセットアップします。
   b. コンテンツが Content Engine に戻され、保管されます。
5. Content Engine が、要求されたコンテンツを Web クライアントに送信します。以降で同じコンテンツに対する要求があった場合、Content Engine はローカル ストレージ（キャッシュ）から透過的にその要求を処理します。

WCCP ルータを使用した透過リダイレクションおよびフォワード プロキシ キャッシング

透過キャッシング サービスには、適正に設定されたルータが必要です。ルータは WCCP Version 1 または Version 2 をサポートする Cisco IOS ソフトウェアのバージョンを実行している必要があります。ルータ上でキャッシング サポートをイネーブルにし、Content Engine 上で WCCP サポートをイネーブルにすると、両デバイスは相互に通信し、設定されたサービスを配信します。WCCP 対応ルータを使用するには、インターネットに接続しているインターフェイスに IP アドレスを設定し、ネットワーク上の Content Engine がこのインターフェイスを認識できる必要があります。

ヒント　キャッシング サービスを中断するには、個々の Content Engine の電源を切断したり、それ以外の方法でディセーブルにするのではなく、ルータ上でキャッシング サポートをディセーブルにします（たとえば、**no ip wccp** ルータ コマンドを使用してキャッシングをディセーブルにします）。

WCCP を使用すると、ルータは、宛先のホスト サイトではなく、Content Engine 上の特定の TCP ポートに要求を透過的にリダイレクトします。

WCCP は、WCCP 対応ルータと Content Engine 間の Generic Routing Encapsulation（GRE; 汎用ルーティング カプセル化）トンネル内の UDP ポート 2048 で動作します。WCCP 対応ルータは、IP パケットを受信すると、そのパケットが Content Engine に送信すべき要求であるかどうかを判断します。

WCCP Version 1 ルータは、IP ヘッダー内のプロトコル フィールドが TCP で、TCP ヘッダー内の宛先ポートがポート　80　かどうかを確認します。これらの基準を満たしているパケットが、Content Engine にリダイレクトされます。WCCP Version 1 の場合、Content Engine にリダイレクトできるのは、TCP ポート 80 を宛先とする Web キャッシュ情報だけですが、多くのアプリケーションでは、他のポートを宛先とするパケットもリダイレクトする必要があります。たとえば、プロキシ Web キャッシュ処理、FTP プロキシ キャッシング、ポート 80 以外の Web キャッシング、RealAudio、ビデオなどです。

ルータが WCCP Version 1 ではなく WCCP Version 2 用に設定されていれば、Content Engine のポート 80 以外の TCP ポートにトラフィックがリダイレクトされるように設定できます。たとえば、ルータと Content Engine に カスタム Web キャッシュ サービス（サービス 98）を設定する場合、ルータから HTTP トラフィックを Content Engine のポート 80 以外のポートにリダイレクトできます。Content Engine は、リダイレクトされた HTTP 要求のコンテンツをキャッシュするように設定します。このタイプのキャッシングを、WCCP を使用した HTTP 透過キャッシングと呼びます。

透過要求とは、ルータから Content Engine にリダイレクトされた要求です。透過要求内の URL のスタイルは、通常、サーバ スタイルです。ただし、別のプロキシ サーバ宛ての要求を Content Engine が代行受信するときは、プロキシ スタイルの場合があります。サーバスタイルの要求には、プロト

コルとホスト名が含まれません。ただし、RTSP 要求では、サーバスタイルの URL とプロキシスタイルの URL は同じです。サーバスタイルの URL を受信する場合、サポートされるのは HTTP と RTSP だけです（RTSP ユーザ エージェント基準に適合する場合）。プロキシスタイルの URL を受信する場合は、対応するプロキシ サービスの設定により、HTTP、HTTPS、FTP、および RTSP がサポートされます。

リダイレクション方式として WCCP を設定するには、次の作業を実行する必要があります。

1. ルータの WCCP リダイレクションをイネーブルにします（「ルータでの WCCP サービスの設定」[p.6-27]）。
2. スタンドアロン Content Engine 上で WCCP をイネーブルにします。WCCP 対応ルータと Content Engine でサポートする特定の WCCP サービス（Web キャッシュ サービスなど）を設定します。

   詳細は、「WCCP 透過リダイレクションのスタンドアロン Content Engine の設定」（p.6-8）を参照してください。

（注）　サポート対象の WCCP サービスの完全なリストは、表 B-3 を参照してください。

透過リバース プロキシ キャッシングの概要

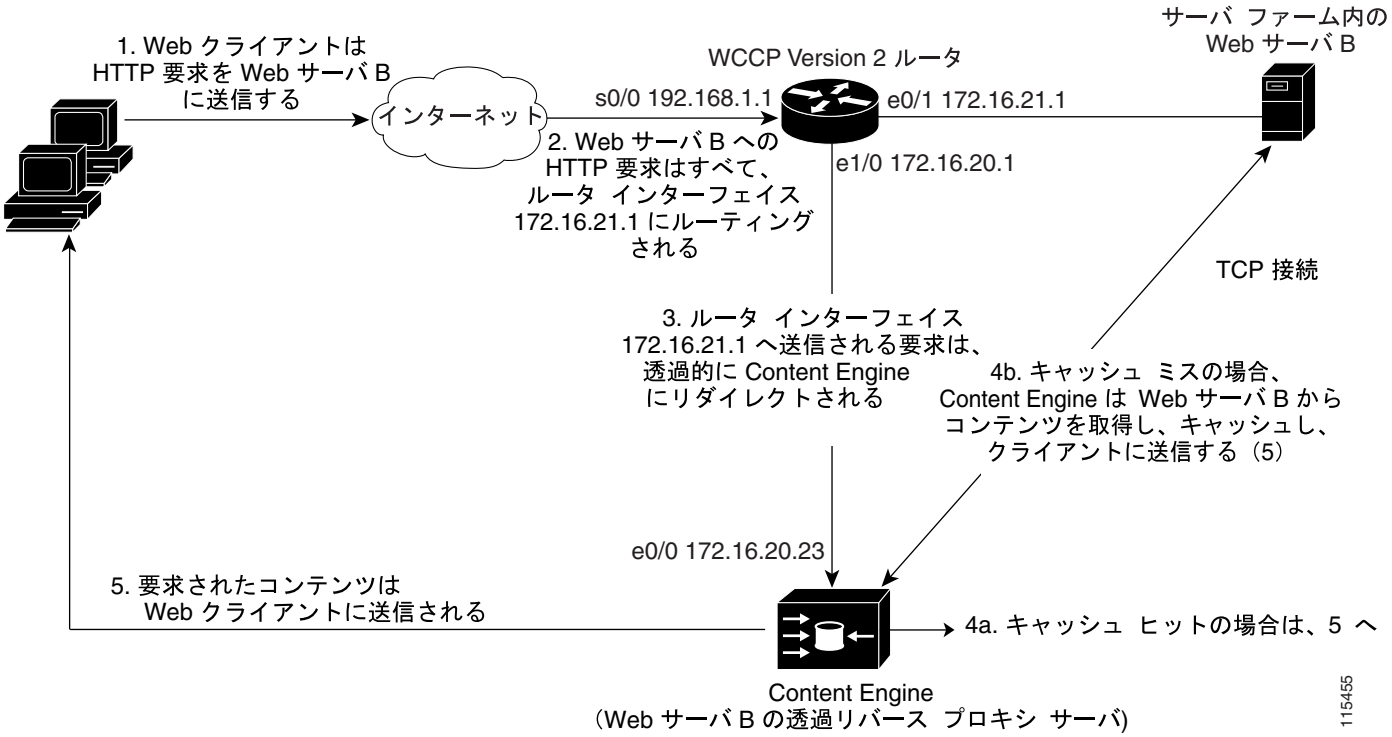
応答時間の短縮、サービス アベイラビリティの最大化を図り、過剰な URL ヒット数、または過剰な帯域幅要求にも対応できるようにするには、Content Engine を、Web サイト サーバ ファームの前面に配置します。この配置は、ファイアウォールやサーバのトラフィック負荷を軽減し、Web サイト全体のインフラストラクチャを最適化します。このタイプの配置方法を、Web サーバの高速化、またはリバース プロキシと呼びます。このように配置した Content Engine を、リバース プロキシ キャッシュと呼びます。この Content Engine は、トランザクションの逆側の、オリジン サーバの手前で動作するからです。

コンテンツに対する着信要求をオリジン サーバ（サーバ ファームのサーバ）で処理するのではなく、Content Engine（リバース プロキシ サーバ）で透過的に処理することにより、Web トラフィックは大幅に削減されます。リバース プロキシ サーバにより、サーバ ファームを効果的に拡張できます。

リバース プロキシ キャッシュ設定では、プロキシ サーバに、インターネット上でルーティング可能な IP アドレスを設定します。クライアントの要求は、ドメイン名の DNS 解決に基づいて、プロキシ サーバに送信されます。クライアント側では、リバース プロキシ サーバは Web サーバのように認識されます。

ACNS 5.x ソフトウェアは、WCCP Version 2 対応ルータまたはレイヤ 4 CSS スイッチの 2 種類のデバイスによりトラフィックのリダイレクションまたは代行受信を実行することにより、リバース プロキシ キャッシングを提供します。図 3-3 に、WCCP 対応ルータを使用した一般的なリバース プロキシ キャッシングの配置を示します。このタイプの配置の場合、Content Engine は WCCP Version 2 対応ルータと相互運用して、Web サーバ環境内でリバース プロキシ サービス（サービス 99）を提供します。Content Engine は、Web サーバ ファームの直前に配置されます。透過および非透過フォワード プロキシ サーバとは異なり、リバース プロキシ サーバは、サーバ ファームの代わりに要求を代理処理して、サーバ ファーム内のサーバからのコンテンツのみをキャッシュします。

図 3-3　スタンドアロン Content Engine と WCCP Version 2 ルータを使用したリバース プロキシ キャッシング



（注）　ルータ上のリダイレクト リスト、または Content Engine 上のスタティック バイパス リストを使用すると、フローが代行受信をバイパスできるようになります。これらのリストは、送信元および宛先の IP アドレスに基づいた基準を使用します。

WCCP ルータまたはレイヤ 4 CSS スイッチをそれぞれ使用したリバース プロキシ キャッシングの実装方法の詳細については、「例 1 — WCCP 透過リダイレクションを使用したリバース プロキシ キャッシングの配置」（p.3-14）および「例 2 — レイヤ 4 スイッチングを使用したリバース プロキシ キャッシングの配置」（p.3-15）を参照してください。

図 3-3 では、インターネットに接続しているルータ インターフェイスの IP アドレスは 192.168.1.1 です。Web サーバ B 宛の HTTP 要求はすべて、172.16.21.1 のルータ インターフェイスにルーティングされます。このインターフェイスで HTTP 要求を受信すると、ルータはこの要求を透過的に代行受信し、IP アドレス 172.16.20.23 の Content Engine にリダイレクトします。このように、Content Engine は論理的に Web サーバ B の手前に配置され、Web サーバの HTTP トラフィックの負荷を軽減します。オリジン サーバのコンテンツを要求している Web クライアントは、リバース プロキシ モードで機能する Content Engine のスタティックな Web ページを受け取ります。これによって、HTTP トラフィックの処理からバック エンド インフラストラクチャが解放されます。

スタンドアロン Content Engine にリバース プロキシ キャッシングを配置する重要な利点は、次のとおりです。

- サーバ ファーム上のスタティック イメージ処理の負担を軽減することで、Web サーバ拡張の代替方法を提供する。
- Content Engine を地理的に離れた場所に配置することにより、これらの場所でコンテンツを複製できる。
- クライアントの設定を変更する必要がない（クライアント ブラウザを設定して、リバース プロキシ サーバとして機能している Content Engine を指定する必要がない）。

例 1 — WCCP 透過リダイレクションを使用したリバース プロキシ キャッシングの配置

次に、（レイヤ 4 スイッチングではなく）WCCP 透過リダイレクションを使用した、リバース プロキシ キャッシング サービスの配置例を示します。

ステップ 1　スタンドアロン Content Engine 上で WCCP Version 2 をイネーブルにします（WCCP Version 1 はリバース プロキシ サービスをサポートしていません）。

```
ContentEngine(config)# wccp version 2
```

ステップ 2　ルータ リストを設定します。

```
ContentEngine(config)# wccp router-list 1 172.16.20.1
```

ステップ 3　リバース プロキシ サービス（サービス 99）を実行するように Content Engine に指示します。

```
ContentEngine(config)# wccp reverse-proxy router-list-num 1
```

ステップ 4　グローバル コンフィギュレーション モードを終了します。

```
ContentEngine(config)# exit
```

ステップ 5　リバース プロキシ サービス（サービス 99）を実行するようにルータに指示します。

```
Router(config)# ip wccp 99
```

ステップ 6　次の手順で、オリジナル サーバに対して出力リダイレクションを実行するリバース プロキシ サービスを設定します。

a. 設定するルータ インターフェイスを指定します。この例では、Ethernet 0/1 が Web サーバへのルータ インターフェイスです。

```
Router(config)# interface Ethernet 0/1
```

b. 指定したインターフェイス宛ての TCP ポート 80 トラフィックを、リバース プロキシ サービスを受け入れる Content Engine にリダイレクトするようにルータに指示します。この例では、ルータは 1 台だけです。

```
Router(config-if)# ip wccp 99 redirect out
```

ステップ 7　次の手順で、スタンドアロン Content Engine に対して入力リダイレクションを実行するリバース プロキシ サービスを設定します。

a. 設定するルータ インターフェイスを指定します。この例では、s0/0 がインターネットへのルータ インターフェイスです。

```
Router(config)# interface s0/0
```

b. 指定のインターフェイスで受信した TCP ポート 80 トラフィックを、リバース プロキシ サービスを受け入れる Content Engine にリダイレクトするようにルータに指示します。

```
Router(config-if)# ip wccp 99 redirect in
```

ステップ 8　インターフェイス コンフィギュレーション モードを終了します。

```
Router(config-if)# exit
```

例 2 — レイヤ 4 スイッチングを使用したリバース プロキシ キャッシングの配置

レイヤ 4 スイッチに基づくリバース プロキシ キャッシングの場合、スタンドアロン Content Engine とレイヤ 4 CSS スイッチを相互運用し、Web サーバ環境内でリバース プロキシ サービスを提供します。この場合、一連の Content Engine のローカル キャッシュを使用して、実際の Web サーバを高速化します。レイヤ 4 CSS スイッチは、仮想 IP アドレス（200.200.200.1 など）を使用する負荷分散スイッチです。この仮想 IP アドレスは、一般ユーザ（サーバに情報を要求する Web クライアント）に公開される Web サーバの IP アドレスです。クライアント要求は、最初にレイヤ 4 スイッチに到達します。スイッチにはレイヤ 4 リダイレクション機能があるので、その要求は、レイヤ 4 スイッチに接続された Content Engine（リバース プロキシ サーバ）にリダイレクトされます。要求がキャッシュ ヒットの場合、Content Engine が要求に応答します。要求がキャッシュ ミスの場合、Content Engine は要求をバック エンドの Web サーバに送信し、その結果をキャッシュして、要求されたコンテンツをクライアントに送信します。

リバース プロキシ キャッシングの場合、Content Engine は一般からも透過的で、通常、クライアントは Content Engine（リバース プロキシ サーバ）の存在を認識しません。

（注）　ここで説明するソリューション例は、特定の Web サーバおよび特定の構成でのみ機能します。

次に、レイヤ 4 CSS スイッチ（CS150）およびスタンドアロン Content Engine（ce1）に基づくリバース プロキシ キャッシングの設定例を示します。

ステップ 1　可能ならば、負荷バイパス機能をディセーブルにして、Content Engine が確実に要求に対応できるようにします。

```
ce1(config)# no bypass load enable
```

ステップ 2　レイヤ 4 スイッチを使用したトラフィックのリダイレクションを受け入れるように Content Engine を設定します。

```
ce1(config)# http l4-switch enable
```

ステップ 3　レイヤ 4 スイッチをリバース プロキシ キャッシング用に設定します。

設定を変更するには、レイヤ 4 スイッチをコンフィギュレーション モードにする必要があります。

（注）　レイヤ 4 CSS スイッチを使用したキャッシングの詳細は、『*CSS Advanced Configuration Guide*』を参照してください。

a. CSS スイッチの所有者を指定します。

```
CS150(config)# owner cisco
```

b. 現在の所有者に対するリバース プロキシ ルールを作成します。

```
CS150(config-owner[cisco])# content RPCRule
Create content RPCRule>, [y/n]:y
```

c. リバース プロキシ ルールにサービスを追加します。この場合、スタンドアロン Content Engine（ce1）がサービスとしてリバース プロキシ ルールに追加されます。

```
CS150(config-owner-content[RPCRule])# add service ce1
```

d. 仮想 IP アドレスを CSS スイッチに割り当てます。

```
CS150(config-owner-content[cisco-RPCRule])# vip address 200.200.200.1
```

e. サービス プロトコルとして TCP を指定します。

```
CS150(config-owner-content[cisco-RPCRule])# protocol tcp
```

f. トラフィックに対する要求がポート 80 に到達するように指定します。

```
CS150(config-owner-content[cisco-RPCRule])# port 80
```

g. キャッシング可能なコンテンツを定義します。

```
CS150(config-owner-content[cisco-RPCRule])# url "/*" eql Cacheable
```

ステップ 4　CSS スイッチでコンフィギュレーション モードを終了します。

```
CS150(config-owner-content[cisco-RPCRule])# exit
```

拡張透過キャッシング機能の使用方法

透過ネットワーク キャッシングの基本原則の 1 つは、Content Engine がエンド ユーザに対して常に透過的でなければならないということです。また、透過キャッシング ソリューションが原因でネットワークに何らかの障害が発生したり、副次的に障害を引き起こしたりしないようにする必要があります。

ACNS ソフトウェアでは、クライアント ブラウザが動作しない場合や、Web サーバが HTTP に準拠していない場合でも、WCCP 対応ルータと各種の先進技術を使用して Content Engine の透過性を確保します。詳細は、第 15 章「スタンドアロン Content Engine の拡張透過キャッシング機能の設定」を参照してください。

スタンドアロン Content Engine へのストリーミング メディア サービスの配置

ストリーミングとは、すべてのメディア パケットの受信が完了する前に、コンテンツにアクセスしたり、コンテンツを表示できる技術です。これに対し、キャッシングの場合、コンテンツにアクセスする前にコンテンツ全体を受信する必要があります。ライブ コンテンツまたはオンデマンド コンテンツとして、ビデオ オン デマンド（VOD）などのストリーミング メディアを配信できます。

Cisco ACNS ソフトウェアは、Microsoft Windows Media Technologies（WMT）ソリューション、RealNetworks RealMedia ソリューションなど、数種類の形式のストリーミング メディア ソリューションをサポートしています。サポート対象のストリーミング メディアのリストは、表 1-3 を参照してください。配置できるストリーミング メディア ソリューションのタイプは、Content Engine が Content Distribution Manager に登録されているか（登録済み Content Engine）、またはスタンドアロン Content Engine であるかによって異なります。登録済み Content Engine の場合、表 1-3 に示すすべてのソリューションがサポートされます。スタンドアロン Content Engine の場合、サポートされるのは WMT ソリューションおよび RealMedia ソリューションだけです。たとえば、RTSP ベースの Cisco Streaming Engine および Apple QuickTime ソリューションは、スタンドアロン Content Engine ではサポートされません。

スタンドアロン Content Engine への Real Media ストリーミング サービスの配置方法については、第 8 章「スタンドアロン Content Engine の RealMedia サービスの設定」を参照してください。スタンドアロン Content Engine への WMT ストリーミング サービスの配置方法については、第 9 章「スタンドアロン Content Engine の WMT ストリーミング メディア サービスの設定」を参照してください。

Content Distribution Manager に登録する Content Engine 上でのストリーミング メディア サービスの設定方法については、『*Cisco ACNS Software Configuration Guide for Centrally Managed Deployments*』Release 5.4 を参照してください。

フィルタリングおよびアクセス制御サービスの配置

スタンドアロン Content Engine にキャッシングおよびストリーミング サービスを設定したあと、特定のコンテンツ サービス（ルール処理、URL フィルタリングなど）を設定できます。ここでは、スタンドアロン Content Engine を設定して、ユーザのインターネット アクセスを制御する例を示します。この例では、スタンドアロン Content Engine を企業のキャッシング エンジンとして配置します。また、コンテンツ要求の処理について、次の特別な要件があるものとします。

- スタンドアロン Content Engine で、選択したサブネットのユーザだけにインターネットへのアクセスを許可し、インターネット アクセスを特定の Web サイトに限定する必要がある。
- スタンドアロン Content Engine で、許可されたサブネット上の特定のユーザに対し、許可された Web サイトへのアクセスをブロックする必要がある。
- スタンドアロン Content Engine で、他のすべてのユーザに対してインターネットへのアクセスをブロックし、Web サイトへのアクセス要求が拒否されたことを通知するカスタム メッセージをユーザに表示する必要がある。
- サイトとユーザは少数のため、サードパーティ製の URL フィルタリング ソリューション（SmartFilter 製品など）を実装することは考えていない。

次に、ACNS 5.x ソフトウェアを実行するスタンドアロン Content Engine を使用して、上記要件を満たすソリューションを実装する例を示します。

ステップ 1　スタンドアロン Content Engine 上でルール処理をイネーブルにします。

```
ContentEngine(config)# rule enable
```

ステップ 2　許可されたサブネットに属する選択したユーザに対し、通常許可されているサイトへのアクセスをブロックします。この場合、Content Engine は、サブネット 192.168.1.50 およびドメイン .foo\com に属するユーザに対し、通常許可されているサイトへのアクセスをブロックします。

```
ContentEngine(config)# rule action block pattern-list 2 protocol all
ContentEngine(config)# rule pattern-list 2 group-type and
ContentEngine(config)# rule pattern-list 2 src-ip 192.168.1.50 255.255.255.0
ContentEngine(config)# rule pattern-list 2 domain .*foo\.com
```

ステップ 3　Content Engine がブロックしたユーザに対して表示するカスタム メッセージを定義します。

a. カスタム メッセージ ファイルを保持するために、local1 または local2 の下に別個のディレクトリを作成します。

b. ブロッキング メッセージを含む block.html という名前の HTML ファイルを作成します。カスタム メッセージの HTML ページに関連したすべての組み込みグラフィックスを、必ず、block.html ファイルと同じディレクトリにコピーしてください。次に、block.html ファイルの例を示します。

```
<TITLE>Cisco Content Engine example customized message for url-filtering</TITLE>
<p>
<H1>
<CENTER><B><I><BLINK>
<FONT COLOR="#800000">P</FONT>
<FONT COLOR="#FF00FF">R</FONT>
<FONT COLOR="#00FFFF">A</FONT>
<FONT COLOR="#FFFF00">D</FONT>
<FONT COLOR="#800000">E</FONT>
<FONT COLOR="#FF00FF">E</FONT>
<FONT COLOR="#00FFFF">P</FONT>
<FONT COLOR="#FF8040">'</FONT>
<FONT COLOR="#FFFF00">S</FONT>
</BLINK>
<FONT COLOR ="#0080FF">Blocked Page</FONT>
</I></B></CENTER>
</H1>
<p>
<p>
<IMG src="/content/engine/blocking/url/my.gif">
<p>
This page is blocked by the Content Engine.
<p>
```

ステップ 4　ブロックしたユーザにカスタム　メッセージが送信されるように、Content　Engine　を設定します。block.html ファイルのディレクトリ（dirname）を指定します。

```
ContentEngine(config)# url-filter http custom-message dirname
```

この例では、Content Engine がブロックされたサイトへの要求を代行受信した場合、block.html ファイルにより、次のカスタム メッセージが表示されます。

```
This page is blocked by the Content Engine
```

ステップ 5　Content　Engine を設定して、badurl.lst ファイルにリストされている URL へのクライアント要求を拒否します。また、goodurl.lst ファイルにリストされている URL への要求だけを許可するように設定します。URL リストの使用は、HTTP、HTTPS、および FTP の形式の要求、および MMS や RTSP などのストリーミング メディア プロトコルの要求に適用できます。次に、特定の HTTP URL を許可し、他のすべての URL を除外する例を示します。

a. goodurl.lst という名前のプレイン テキスト ファイルを作成します。

b. goodurl.lst ファイルに、排他的に許可する URL を入力します。goodurl.lst ファイル内の URL のリストは、http://www.domain.com 形式で入力し、改行キーで区切ります。

c. goodurl.lst ファイルを、Content Engine の /local1 sysfs ディレクトリにコピーします。

（注）　good リストを保持するために、local1 の下に別個のディレクトリ（/local1/filtered_urls など）を作成することを推奨します。

ステップ 6　Content Engine が goodurl.lst ファイル名を参照するように指定します。

```
ContentEngine(config)# url-filter http good-sites-allow file local/local1/goodurl.lst
```

ステップ 7　Content Engine が、good URL へのアクセスだけを許可するように設定します。

```
ContentEngine(config)# url-filter http good-sites-allow enable
```

スタンドアロン Content Engine での URL フィルタリングおよびルールの設定方法については、第 11 章「スタンドアロン Content Engine 上での事前コンテンツ ロードおよび URL フィルタリングの設定」および 第 13 章「スタンドアロン Content Engine の Rules Template の設定」を参照してください。