



# コンテンツ ロード バランシング の概要

---

コンテンツ ロード バランシングは、特定の宛先にコンテンツを要求するとき、CSS が行う処理の方法です。CSS でのロード バランシングを理解するために、この章ではサービス、所有者、およびコンテンツ ルールの関係と、CSS が TCP トラフィックと UDP トラフィックをどのように処理するかについて説明します。次の内容について説明します。

- [サービス、所有者、コンテンツ ルールの概要](#)
- [CSS フローの概要](#)

この章の内容は、特に指定がない限り、すべての CSS モデルに共通です。

## サービス、所有者、コンテンツ ルールの概要

CSS では、サービス、所有者、およびコンテンツ ルールを設定して、コンテンツ要求を特定の宛先サービス（サーバやサーバのポートなど）に送信することができます。サービス、所有者、およびコンテンツ ルールを設定することにより、特定のコンテンツへの要求それぞれについて、CSS の処理方法を最適化し、制御することができます。次に、サービス、所有者、およびコンテンツ ルールについて説明します。

- **サービス**とは、コンテンツが物理的に存在している場所であり、ローカルまたはリモートのサーバとポートで定義されます。サービスをコンテンツ ルールに追加するのはユーザです。サービスをコンテンツ ルールに追加すると、リソース プールに格納され、CSS がコンテンツ要求のロード バランシングを行う場合に使用されます。サービスは、複数のコンテンツ ルールに属することもできます。
- **所有者**とは、通常、Web コンテンツを提供し要求に応じて帯域を割り当てる Web ホスティング サービスを依頼する、個人または企業を指します。所有者は複数のコンテンツ ルールを指定できます。
- **コンテンツ ルール**とは、個別のルールを含む階層形式のルール セットを指します。個別のルールには、Web サイトでアクセスされるコンテンツ（.html ファイルなど）、コンテンツのミラー方法、コンテンツが常駐するサーバ、および CSS でのコンテンツ要求の処理方法が記載されています。各ルールセットには所有者が必要です。

CSS はコンテンツ ルールを使用して、次の内容を判断します。

- コンテンツが物理的に存在する場所（ローカルまたはリモート）
- コンテンツ要求を送信する場所（サービス（複数可））
- 使用するロード バランシング方式

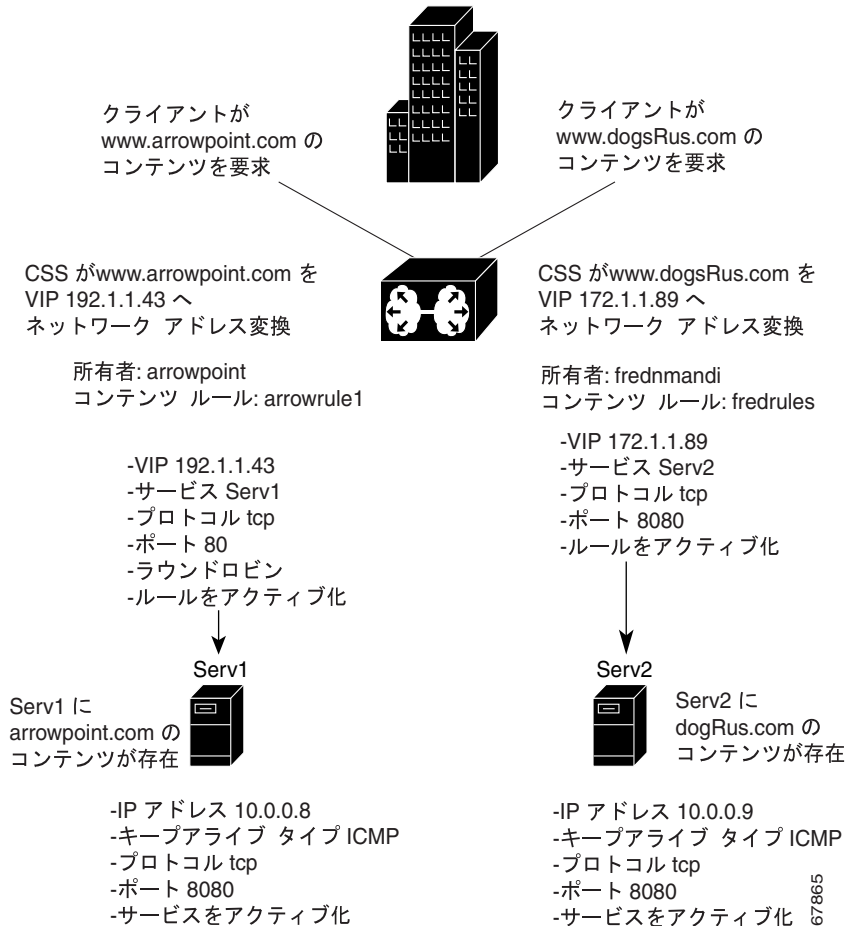
コンテンツの要求があると、CSS は次の処理を行います。

1. 所有者のコンテンツ ルールを使用して、所有者の Virtual IP（VIP; 仮想 IP）アドレスまたはドメイン名を Network Address Translation（NAT; ネットワーク アドレス変換）で、対応サービスの IP アドレスとポートに変換します。
2. コンテンツ要求を満たす利用可能なサービスを調べます。
3. コンテンツ ルールを使用して、コンテンツの要求を処理するのに最適なサービスを選択します。

4. すべてのコンテンツルール（たとえば、ロードバランシング方式、リダイレクト、フェールオーバー、スティッキなど）を適用して、コンテンツ要求を処理します。

図 1-1 に、CSS のサービス、所有者、およびコンテンツルールの概念を示します。

図 1-1 サービス、所有者、コンテンツルール



## ■ サービス、所有者、コンテンツ ルールの概要

次に示す各章で、サービス、所有者、およびコンテンツ ルールの設定についてそれぞれ説明します。

- [第3章「サービスの設定」](#)
- [第9章「所有者の設定」](#)
- [第10章「コンテンツ ルールの設定」](#)

CSS での TCP トラフィックおよびUDP トラフィックのフローについては、[「CSS フローの概要」](#)を参照してください。

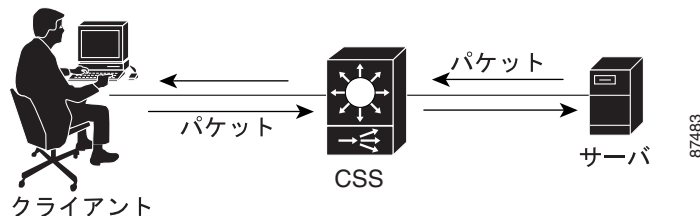
## CSS フローの概要

フローとは、CSS を通って送信元（クライアント）と宛先（サーバ）間の TCP または UDP 接続で転送される一連のパケットの流れをいいます。入力フロー（CSS に入ってくるトラフィック）内のパケットはすべて、次の5つのタプルを共有します。

- 送信元アドレス
- 宛先アドレス
- プロトコル
- 送信元ポート
- 宛先ポート

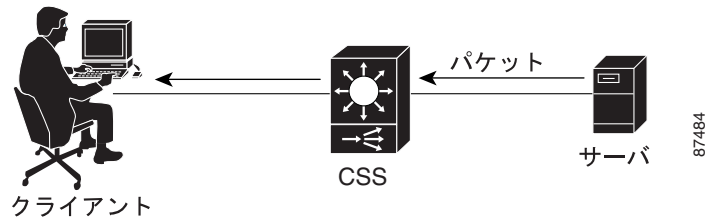
TCP のフローは双方向です（[図 1-2](#) 参照）。パケットは、CSS を通じてクライアントからサーバ、またサーバからクライアントに移動します。厳密に言うと、TCP 接続は各方向にそれぞれ1本、合計2本のフローから成ります。TCP フローは SYN で始まり、ACK の後に FIN/ACK または RST で終了します。

図 1-2 TCP フローの例



通常、UDP フロー（[図 1-3](#) 参照）は単方向です（サーバからクライアントに送信されるストリーミング オーディオなど）。UDP フローには明確な開始と終了がなく、一定期間が経過した後でだけ完了するとみなされます。この一定期間とは、宛先装置が、元のフローで定義されたものと同じアドレス、プロトコル、およびポートを共用するパケットを受信しない期間のことです。

図 1-3 UDP フローの例



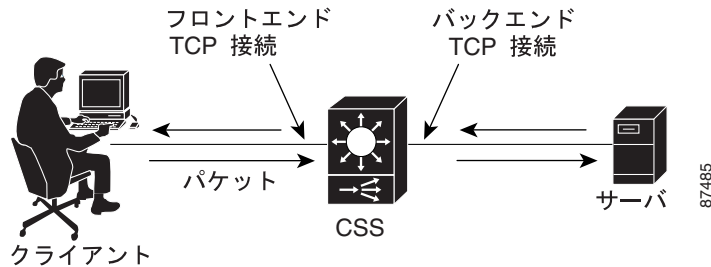
CSS では、Flow Control Block (FCB; フロー制御ブロック) というデータ構造体を使用して入力フローを設定および追跡します。FCB には、CSS がフローを処理および管理するのに必要な情報がすべて含まれています。フロー情報から FCB を作成することをフローマッピングと呼びます。FCB の作成とフローマッピングは、各モジュールセッションプロセッサ内のフローマネージャで実行されます。

単方向フローごとに 1 つの FCB を使用します。したがって、TCP フローでは 2 つの FCB を、UDP フローでは 1 つの FCB を使用します。フロントエンド SSL は TCP で動作しますが、4 つの FCB を必要とします。また、バックエンド SSL ではさらに 2 つの FCB が加わるので、全二重 SSL 接続ごとに合計で 6 つの FCB が使用されます。SSL の詳細については、『*Cisco Content Services Switch SSL Configuration Guide*』を参照してください。

クライアント、CSS、サーバ間の接続は、次の 2 つの部分から成ります (図 1-4 参照)。

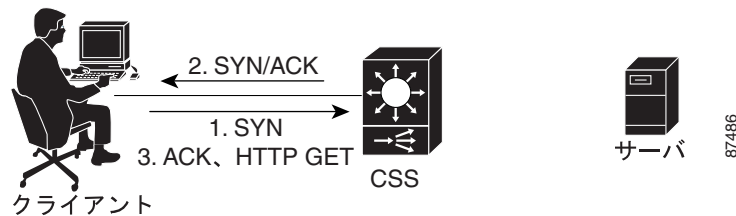
- フロントエンド：クライアントと CSS 間の接続
- バックエンド：CSS とサーバ間の接続

図 1-4 フロントエンド接続およびバックエンド接続による TCP フローの例



レイヤ5フローは、クライアントによるコンテンツの要求で始まります。DプロキシがDNS要求（たとえば、クライアントがWebブラウザでURLを入力する）を解決し、クライアント要求をCSSの仮想IP（VIP）アドレスに転送すると、CSSはTCP 3ウェイハンドシェイクを使用してクライアントとのフロントエンドTCP接続を確立します（図1-5参照）。

図 1-5 フロントエンドTCP接続の設定（遅延バインディング）



レイヤ5フローが確立すると、CSSは、クライアントSYNの宛先装置（サーバ）のプロキシとして機能することでバックエンドTCP接続を「スプーフィング」します。つまり、CSSは、サーバとのバックエンドTCP接続を設定する前にSYN/ACKでクライアントのSYNに応答します。

このプロセスは遅延バインディングと呼ばれます。遅延バインディングによって、クライアントは ACK と HTTP GET 要求で応答します。このプロセスによって、CSS は、コンテンツ要求に対する最適なサービス（コンテンツがあるサーバポートまたは FTP などのサーバで実行されるアプリケーション）を選択するのに必要な情報を収集できます。

CSS は、HTTP 要求メソッド（GET、HEAD、POST など）内の HTTP ヘッダーと URL を調べます。HTTP ヘッダー、URL、および CSS に設定されたコンテンツルールの情報に基づいて、CSS は要求を満たす最適なサイトと最適なサービスを選択します。CSS は次のような要素を使ってサービスの選択（サーバロードバランシング）を行います。

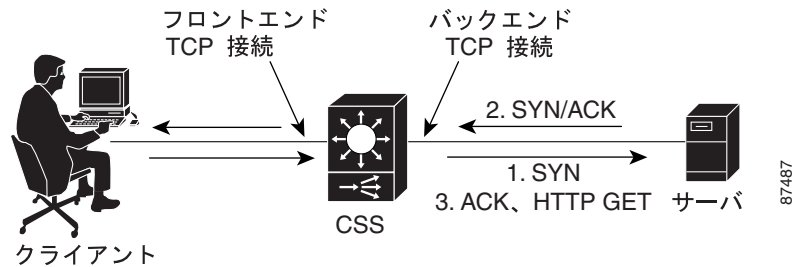
- コンテンツ ルールの一致
- サービスの可用性
- サービスの負荷
- クッキー
- 送信元 IP アドレス

CSS server load balancing (SLB; サーバロードバランシング) の詳細については、「[サービス、所有者、コンテンツルールの概要](#)」を参照してください。

CSS は、要求されたコンテンツをクライアントに提供する最適なサービスを選択すると、TCP 3 ウェイ ハンドシェイクを使用してサービスとのバックエンド接続を確立し、フロントエンド接続とバックエンド接続をつなぎます。CSS は、クライアントのコンテンツ要求をサービスに転送します（[図 1-6](#) 参照）。サービスは CSS を通じてクライアントに応答します。フローの残りの継続期間に、CSS はクライアントとサービスの間でパケットを切り換え、必要に応じてネットワークアドレス変換（NAT）やその他のパケット変換を実行します。



図 1-6 バックエンド TCP 接続の設定 (遅延バインディング)



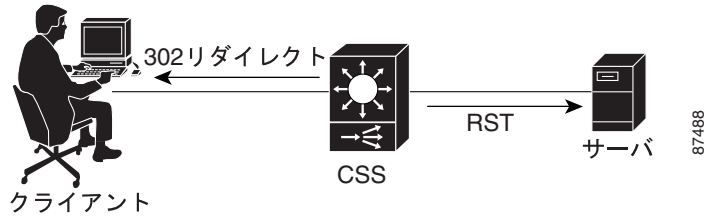
同じ TCP 接続 (HTTP 1.1 以上) で同じクライアントから引き続き出されたコンテンツ要求については、CSS は、デフォルトで最初の HTTP 要求に、コンテンツを提供したサービスとのバックエンド接続を維持しようとします。このことを持続性と呼びます。

固定接続の継続期間中に、CSS は、コンテンツ ルール、ロード バランシング、およびサービスの可用性に基づいて、クライアントの接続を新しいサービスに移動する必要があるかどうかを判断する必要があります。状況によって、クライアントの接続を移動する必要がない場合と、移動が必須になる場合があります。

クライアントを新しいサービスに移動する必要が生じた場合に、次のいずれかの機能を実行するように CSS を設定できます。

- **HTTP リダイレクション : persistence reset redirect** コマンドにより、CSS は、サービスに RST を送信してバックエンド接続を終了する (図 1-7 参照)。CSS は、302 リダイレクトをクライアントのブラウザに送信して同じ DNS 名で再接続することをブラウザに通知します。ただし、この場合、HTTP 要求は異なるコンテンツ ルールに一致します。これにより、クライアントと最適なサービスの間で新しいフローが確立されます。

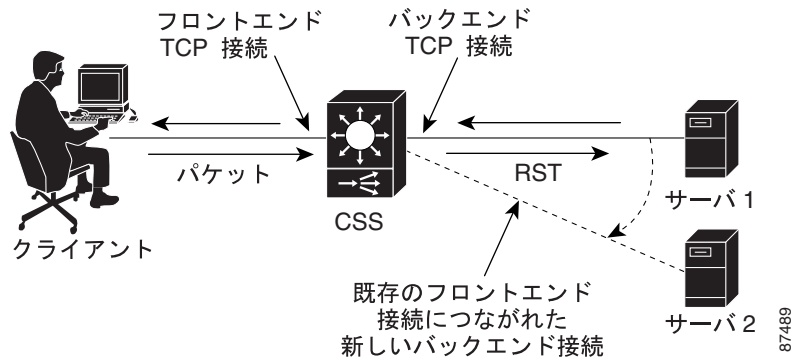
図 1-7 HTTP リダイレクションの例



- サービスの再マッピング：**persistence reset remap** コマンドにより、CSS はサービス（図 1-8 のサーバ 1）に RST を送信してバックエンド接続のみを終了し、サービス サーバ 2 とのバックエンド接続を確立して、フロントエンド接続とバックエンド接続をつなぐ。CSS は、クライアントからのコンテンツ要求をサーバ 2 に転送します。この時点でパケットはクライアント 2 とサーバ 2 間を移動します。

持続性、HTTP リダイレクト、およびサービスの再マッピングの詳細については、第 10 章「コンテンツルールの設定」を参照してください。

図 1-8 バックエンド接続の再マッピングの例



CSS のフロー マネージャはアイドル状態の古いフローを定期的に切断し、システム リソース (FCB) を再要求します。この処理をフロー リソースの再要求と呼びます。また、フローのクリーンアップ、またはガベージ コレクションとも呼びます。フロー リソースを再要求すると、TCP と UDP のリストから FCB が削除されます。CSS では、フローで不要になった FCB を再利用して性能を最適化します。

通常は、CSS で現在アクティブになっているフローの合計数に対するある割合でフローが消去されます。CSS では、UDP フローが必ず消去されます。TCP フローの場合は、FCB の合計数に対する使用済み FCB 数が一定の割合に達すると、リソースが回収されます。また、長時間存続している TCP フローが FIN や RST を受信している場合、またはタイムアウト時間が経過している場合、これらのフローも消去されます。デフォルトのフロー消去動作については、さまざまなコマンドを設定して変更することができます。

アイドル状態の TCP フローを消去しない方がよい場合があります。たとえば、データの送受信が行われない場合でも永続的にアクティブなままにしておく必要があるデータベース サーバに接続する場合です。保持しておく必要のある長時間存続するアイドル接続が CSS によって廃棄されている場合は、次の TCP フロー コマンドを設定できます。

- **flow permanent** コマンド：再要求されない固定 TCP（または UDP）ポートを作成する。
- **flow-timeout-multiplier** コマンド：TCP および UDP フローについて、コンテンツ ルールおよびソース グループの単位でフロー無活動タイムアウト値を設定する。

CSS で、TCP および UDP フローの処理と消去を制御するコマンドについては、[第2章「フローパラメータとポートマッピングパラメータの設定」](#)を参照してください。

