CHAPTER 10

追加機能およびオプションの設定

この章では、コンテント スイッチングの設定方法について説明します。

セッションの持続性（スティッキ性）の設定

セッションの固定（スティッキ性）は、同一のクライアントから同一のサーバへ常時複数の（同時または連続した）接続を送信する機能です。この機能は、特定のロードバランシング環境で、通常、必要です。

アプリケーションのトランザクションを完了する（ブラウザで Web サイトにアクセスし、購入するさまざまな品目を選択してからチェックアウトするなど）には、通常、複数（ときには何百、何千）の同時または連続接続が必要です。こういったトランザクションの多くは、一時的に重要な情報を生成して、それを使用します。この情報はトランザクションを処理する特定のサーバ上で保存したり修正したりします。このトランザクションの完了までに数分から数時間かかる場合があり、クライアントはその間同じサーバに何度も送信する必要があります。

バックエンド共有データベースによる多層設計でも問題の一部は解決できますが、ローカル サーバ キャッシュを利用することで、アプリケーションのパフォーマンスをさらに改善できます。ローカル サーバのキャッシュを使用するとデータベースに接続する必要がなくなり、新しいサーバが選択されるたびにトランザクション固有の情報が得られます。

持続性に関する問題の中でも最も難しいのは、複数の接続にわたって個々のクライアントをどのように特定するかということです。ロードバランシング デバイスはクライアントの識別に使用できるあらゆる情報を保存し、現在トランザクションを処理しているサーバにその情報を関連付ける必要があります。

（注）　CSM-S は、256,000 のエントリからなるスティッキ データベースを維持できます。

CSM-S は個々のクライアントを識別して、次の方法で固定処理を実行します。

- 送信元 IP アドレスの固定

  CSM-S に送信元 IP アドレス全体（32 ビットのネットマスクを含めて）を学習させるか、またはその一部を学習させるかを設定できます。

- SSL 識別情報の固定

  クライアントおよびサーバが Secure Socket Layer（SSL）を介して通信している場合、複数の接続にわたって一意の SSL 識別番号が維持されます。SSL バージョン 3.0 または Transport Layer Security（TLS）1.0 では、クリア テキストでこの識別番号を伝送する必要があります。CSM-S は、この値を使用して特定のトランザクションを識別します。ただし、この SSL ID を再度ネゴシエートできるため、常に正しいサーバへの固定（スティッキ性）が維持できるわけではありません。SSL ID ベースの固定方式を利用すると、常に SSL ID を再利用させることによって、SSL 終端装置のパフォーマンスが向上します。

（注）　CSM-S を Catalyst 6500 SSL モジュールと組み合わせて使用した場合、各 Catalyst 6500 SSL モジュールの MAC アドレスが特定のオフセットで SSL ID 内に挿入されるため、SSL ID の再ネゴシエート後も SSL ID を固定できます。この固定方法は、仮想サーバのコンフィギュレーション サブモードで **ssl-sticky** コマンドを使用して設定できます。

スティッキ接続の設定情報については、『*Catalyst 6500 Series Switch SSL Services Module Configuration Note*』の Chapter 5「Configuring Different Modes of Operation」を参照してください。

ssl-sticky コマンドについては、『*Catalyst 6500 Series Switch Content Switching Module Command Reference*』を参照してください。

- ダイナミック Cookie ラーニング

  特定の Cookie 名を探して、クライアント要求の HTTP ヘッダーまたはサーバの「Set-Cookie」メッセージから自動的にその値を学習するように CSM-S を設定できます。

  CSM-S はデフォルトで、Cookie 値全体を学習します。この機能は CSM-S ソフトウェア リリース 4.1.(1) では、オプションのオフセットおよび長さを取り入れて拡張され、Cookie 値の一部分だけを学習するように CSM-S に対して指示できるようになりました。「Cookie 固定のオフセットおよび長さ」（p.10-4）を参照してください。

  ダイナミック Cookie ラーニングは、同一の Cookie 内にセッション ID またはユーザ ID を複数保存するアプリケーションを扱う場合に役立ちます。スティッキ性に関連があるのは、Cookie 値の特定のバイトだけです。

  CSM-S ソフトウェア リリース 4.1(1) には、ダイナミック Cookie スティッキ機能も追加されています。これは、URL の一部としての Cookie 情報を検索する（さらに学習して固定する）機能です。（「URL ラーニング」[p.10-5] を参照）。URL の学習は、HTTP URL に Cookie 情報を組み込むアプリケーションで有用です。場合によっては、この機能を使用して Cookie を拒否するクライアントに対処できます。

- Cookie 挿入

  CSM-S はサーバに代わって Cookie を挿入するので、サーバが Cookie を設定しない場合でも Cookie 固定を実行できます。Cookie には、CSM-S が特定の実サーバ固定を確実に実行するための情報が含まれています。

固定（sticky）グループの設定

固定（sticky）グループを設定するには、固定方法（送信元 IP、SSL ID、Cookie）とそのグループのパラメータを設定し、さらにポリシーと関連付ける必要があります。固定（sticky）タイムアウトは、スティッキ情報がスティッキ テーブルで維持される期間を指定します。デフォルトの固定タイムアウト値は 1440 分（24 時間）です。特定のエントリの固定（sticky）タイマーは、そのエントリに一致する新規接続が開かれるたびにリセットされます。

特定のエントリの固定（sticky）タイマーは、最後のセッションが終了した時点からリセットされます。タイムアウト ポリシーは、IP_Sticky のみを使用するセッションに適用されます。他の固定形式（cookie および url-hash など）を使用するセッションは、この動作の影響を受けません。

固定（sticky）環境変数を設定するには、次のコマンドを使用します。

```
Router(config-module-csm)# variable NO_TIMEOUT_IP_STICKY_ENTRIES 1
```

（注）　複数のポリシーまたは仮想サーバは、潜在的に同じ固定グループに設定できます。その場合、それらのポリシーまたは仮想サーバへのすべての接続に、その固定処理が適用されます。ポリシー 1 および 2、または仮想サーバ 1 および 2 が同じ固定グループに設定されている場合、ポリシー 1 または仮想サーバ 1 を介してサーバ A に固定されるクライアントが、ポリシー2 または仮想サーバ 2 を介して同一のサーバ A に固定されるため、これらの接続は「buddy 接続」とも呼ばれます。

注意　複数のポリシーまたは仮想サーバで同じ固定グループを使用している場合、同じサーバ ファームか、またはグループ内で同じサーバを指定している異なるサーバ ファームを間違いなく使用することが重要です。

固定グループを設定する手順は、次のとおりです。

コマンド	目的
Router(config-module-csm)# **sticky** *sticky-group-id* {**netmask** *netmask* | **cookie** *name* | **ssl**} [**address** [**source** | **destination** | **both**]] [**timeout** *sticky-time*]	同じポリシーと一致する同じクライアントからの接続で、同じ実サーバが使用されるようにします[1]。

1. デフォルトの設定に戻すには、このコマンドの **no** 形式を使用します。

次に、固定グループを設定して、ポリシーに関連付ける例を示します。

```
Router(config-module-csm)# sticky 1 cookie foo timeout 100
Router(config-module-csm)# serverfarm pl_stick
Router(config-slb-sfarm)# real 10.8.0.18
Router(config-slb-real)# inservice
Router(config-slb-sfarm)# real 10.8.0.19
Router(config-slb-real)# inservice
Router(config-slb-real)# exit
Router(config-slb-sfarm)# exit
Router(config-module-csm)# policy policy_sticky_ck
Router(config-slb-policy)# serverfarm pl_stick
Router(config-slb-policy)# sticky-group 1
Router(config-slb-policy)# exit
Router(config-module-csm)# vserver vs_sticky_ck
Router(config-slb-vserver)# virtual 10.8.0.125 tcp 90
Router(config-slb-vserver)# slb-policy policy_sticky_ck
Router(config-slb-vserver)# inservice
Router(config-slb-vserver)# exit
```

Cookie 挿入

サーバが現在適切な Cookie を設定していない場合にセッション Cookie を固定するには、Cookie 挿入機能を使用します。この機能をイネーブルにすると、CSM-S はクライアントからサーバへの応答に Cookie を挿入します。CSM-S は次にサーバからクライアントへのトラフィック フローに Cookie を挿入します。

次に、固定するために Cookie を指定する例を示します。

```
Cat6k-2(config-module-csm)# sticky 5 cookie mycookie insert
```

Cookie 固定のオフセットおよび長さ

Cookie 値は、クライアント / サーバ間のトランザクションで、一部を残して変更される場合があります。その場合、特定サーバへの連続接続を保つために、変更されない部分を使用できます。接続の連続性を固定または維持するために、Cookie の変更されない部分を **cookie offset** *num* [**length** *num*] コマンドを使用して、オフセットおよび長さの値で指定できます。

オフセット（Cookie 値を先頭のバイトから数える）および長さ（Cookie で使用する部分の長さ）をバイトで指定し、固定接続の維持に使用します。これらの値はスティッキ テーブルに保存されます。

オフセットおよび長さは 0 ～ 4000 バイトの範囲で指定できます。Cookie 値がオフセットより長く、オフセットと Cookie の長さを足したものより短い場合、CSM-S はオフセットの後ろの Cookie 部分に応じて接続を固定します。

次に、Cookie のオフセットおよび長さを指定する例を示します。

```
Cat6k-1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Cat6k-1(config)# module csm 4
Cat6k-1(config-module-csm)# sticky 20 cookie SESSION_ID
Cat(config-slb-sticky-cookie)# cookie offset 10 length 6
```

URL ラーニング

URL ラーニングによる Cookie 固定機能により、CSM-S は Set-Cookie フィールドまたは URL 埋め込み Cookie のセッション情報をキャプチャできます。CSM-S は、サーバ応答の Set-Cookie HTTP ヘッダーに埋め込まれた特定の Cookie 値に基づいて、固定テーブルのエントリを作成します。

URL ラーニングを設定すると、CSM-S は次の 3 通りの方法で Cookie 値を学習できます。

- サーバからクライアント方向の Cookie メッセージ
- クライアント要求内の Cookie
- URL に埋め込まれた Cookie 値

最初の 2 つの方法は標準のダイナミック Cookie ラーニング機能ですでにサポートされています。3 つめの方法は URL ラーニング機能として追加されました。

多くの場合、そのあとの一連の HTTP 要求内において、クライアントは同じ Cookie 値を戻します。CSM-S は、それに一致する値に基づいて同じサーバにクライアントを固定します。ただし、クライアントによってはブラウザで Cookie をディセーブルにしているため、このタイプの Cookie 固定接続ができない場合もあります。URL Cookie ラーニングの新機能で、CSM-S は URL ストリングに埋め込まれた Cookie 名および値を抽出できます。この機能が実行されるのは、サーバが Web ページの URL リンクに Cookie を埋め込んでいる場合だけです。

クライアント要求に Cookie が含まれていない場合、CSM-S は、CSM-S に設定されたセッション ID ストリング（?session-id=）を探します。このストリングに対応する値が、CSM-S がキャッシュ内で探しているセッション ID 番号です。セッション ID は、要求情報が保存されていて、なおかつクライアント要求の送信先であるサーバと一致します。

セッション Cookie および URL セッション ID は異なる可能性があるため、Cisco IOS の **sticky** *id* **cookie** *name* コマンドがアップデートされました。次の例で、正しい構文を示します。

（注）　このリリースの Cookie 固定オフセット機能をサポートするために、アップデートされたコマンドにはオフセットおよび長さを指定する構文が含まれています。「Cookie 固定のオフセットおよび長さ」（p.10-4）を参照してください。

クライアント / サーバの動作およびそのフレーム シーケンスに応じて、HTTP Cookie、Set-Cookie ヘッダー、または URL 埋め込み Cookie で表示されている同一の Cookie 値が、標準の HTTP Cookie に表示される場合があります。また、Cookie が URL に埋め込まれているか、HTTP Cookie ヘッダーに表示されているかによって、Cookie 名は URL によって異なる場合があります。異なる名前の Cookie および URL は、これらの 2 つのパラメータの多くがサーバ上で別々に設定されるために生じます。次に、Set-Cookie 名の例を示します。

```
Set-Cookie: session_cookie = 123
```

次に、URL の例を示します。

```
http://www.example.com/?session-id=123
```

sticky コマンドの *name* フィールドには、Cookie ヘッダーに表示される Cookie 名を指定します。このコマンドに追加された **secondary** *session_id* には、URL に表示される対応する Cookie 名を指定します。

次に、URL ラーニング機能の設定例を示します。

```
Cat6k-1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Cat6k-1(config)# module csm 4
Cat6k-1(config-module-csm)# sticky 30 cookie session_cookie
Cat(config-slb-sticky-cookie)# cookie secondary session-id
Cat(config-slb-sticky-cookie)#
```

RHI の設定

ここでは Route Health Injection（RHI）の設定方法について説明します。

RHI について

ここでは、RHI について説明します。

RHI の概要

RHI は、CSM-S に Virtual IP（VIP; 仮想 IP）アドレスの可用性をネットワーク全体にアドバタイズさせます。また、ネットワーク全体にわたって同一の VIP アドレスおよびサービスを持つ複数の CSM-S 装置を配置できます。ある CSM-S は、ほかの装置でサーバ ロードバランス サービスを利用できなくなった場合に、ほかの装置のサービスを変更できます。この CSM-S は、ほかのサーバ ロードバランシング デバイスよりクライアント システムに論理上近いので、サービスの提供もできます。

（注）　CSM-S は VIP アドレスをホスト ルートとしてアドバタイズしますが、ほとんどのルータはホスト ルート情報をインターネットに伝播しないので、RHI の用途はイントラネットに限定されます。

RHI をイネーブルにするには、次のように CSM-S を設定します。

- 実サーバをプローブし、使用可能な仮想サーバおよび VIP アドレスを識別します。
- 変更が発生するたびに、VIP アドレスの可用性情報を Multilayer Switch Feature Card（MSFC）に正確にアドバタイズします。

（注）　起動時に RHI がイネーブルの場合、各 VIP アドレスが使用可能になるので、CSM-S は MSFC にメッセージを送信します。

MSFC は RHI が提供する VIP アドレスの可用性情報を定期的に伝播します。

（注）　セキュリティ上の理由から、ほとんどのルータはホスト　ルート情報をインターネットに伝播しないので、通常、RHI の用途はイントラネットに限定されます。

RHI を使用しない VIP アドレスへのルーティング

RHI を使用しない場合、トラフィックは VIP アドレスが属すクライアント VLAN へのルートを経由して、VIP アドレスに送信されます。CSM-S の起動時に、MSFC はルーティング テーブルにクライアント VLAN へのルートを作成し、このルート情報をほかのルータと共有します。VIP に到達するために、クライアント システムはルータを使用して、各 VIP アドレスが属すネットワーク サブネット アドレスに要求を送信します。

サブネットまたはセグメントに到達可能であっても、その場所にある CSM-S の仮想サーバが動作していない場合、要求は失敗します。ほかの CSM-S 装置はさまざまな場所に配置することができます。ただし、ルータは単に論理的な距離に基づいてサブネットに要求を送信します。

RHI を使用しないと、VIP アドレスを使用できるかどうかが検証されずに、トラフィックが VIP アドレスに送信されます。この場合は、VIP に接続された実サーバがアクティブではないこともあります。

（注）　デフォルトでは、CSM-S は設定された VIP アドレスをアドバタイズしません。

RHI を使用する VIP アドレスへのルーティング

RHI を使用すると、VIP アドレスが使用可能になって、使用不可能な VIP アドレスのアドバタイズが取り消された場合に、CSM-S は MSFC にアドバタイズを送信します。ルータはルーティング テーブルを参照して、要求をクライアントから VIP アドレスに送信するために必要なパス情報を検索します。RHI 機能が有効な場合、一致した中で最も固有性の高いものが VIP アドレス情報としてアドバタイズされます。クライアントに対する要求は、アクティブな VIP サービスを使用して、CSM-S に到達するパスを経由して送信されます。

VIP アドレスのインスタンスが複数存在する場合、クライアント ルータは VIP アドレスのインスタンスごとに必要な情報（可用性およびホップ数）を受信して、その VIP アドレスに対する最適なルートを判別できます。ルータは CSM-S が論理上、クライアント システムに近くなるようにパスを選択します。

（注）　CSM-S はコンテンツを処理するすべての実サーバをプローブで精査することによって、指定された VIP アドレスに到達できるかどうかを判別します。したがって、RHI を使用する場合はプローブも設定する必要があります。VIP アドレスに到達できるかどうかを判別したあと、CSM-S はこの可用性情報を MSFC と共有します。次に、MSFC はこの VIP 可用性情報をイントラネットのほかの装置に伝播します。

CSM-S が VIP の可用性を判別する仕組み

VIP が使用可能かどうかを CSM-S が判断できるようにするには、プローブ（HTTP、ICMP、Telnet、TCP、FTP、SMTP、または DNS）を設定し、それをサーバ ファームに関連付ける必要があります。プローブが設定されている場合、CSM-S は次の確認を行います。

- プローブ用に設定されたすべてのサーバ　ファーム上のすべての実サーバをプローブで調べます。
- 到達可能な（到達可能な実サーバを 1 台以上含む）サーバ ファームを識別します。
- 到達可能な（到達可能なサーバ ファームを 1 つ以上含む）仮想サーバを識別します。
- 到達可能な（到達可能な仮想サーバを 1 台以上含む）VIP を識別します。

VIP の可用性情報の伝播

RHI を使用する場合、CSM-S は使用可能な VIP アドレスを含むアドバタイズ メッセージを MSFC に送信します。MSFC は、CSM-S から受信する VIP アドレスごとに、ルーティング テーブルにエントリを追加します。MSFC で動作中のルーティング プロトコルは、ほかのルータにルーティング テーブル アップデートを送信します。VIP アドレスが使用不可能になると、そのルートはアドバタイズされなくなり、エントリはタイムアウトし、ルーティング プロトコルは変更を伝播します。

（注）　RHI を CSM-S で動作させるには、CSM-S が搭載されているシャーシ内の MSFC で Cisco IOS Release 12.1.7(E) 以降を稼働させ、その MSFC をクライアント側ルータとして設定する必要があります。

仮想サーバ用 RHI の設定

仮想サーバ用の RHI を設定する手順は、次のとおりです。

ステップ 1　VLAN が設定されていることを確認します。第 4 章「VLAN の設定」を参照してください。

ステップ 2　プローブをサーバ ファームに関連付けます。「ヘルス モニタリング用プローブの設定」（p.11-2）を参照してください。

ステップ 3　実サーバをプローブで調べるように CSM-S を設定します。「ヘルス モニタリング用プローブの設定」（p.11-2）を参照してください。

ステップ 4　**advertise active** Server Load Balancing（SLB; サーバ ロード バランシング）仮想サーバ コマンドを入力して、仮想サーバごとに RHI をイネーブルにします。

```
Router(config-module-csm)# vserver virtual_server_name
Router(config-slb-vserver)# advertise active
```

次に、vserver1 という名前の仮想サーバに対して RHI をイネーブルにする例を示します。

```
Router(config-module-csm)# vserver vserver1
Router(config-slb-vserver)# advertise active
```

環境変数

variable *name string* コマンドを使用して、コンフィギュレーションの環境変数をイネーブルにできます。表 10-1 で CSM-S 環境変数の値について説明します。

表 10-1　CSM-S 環境変数

名前	デフォルト	有効値	説明
ARP_INTERVAL	300	整数（15～31536000）	設定したホストの Address Resolution Protocol（ARP; アドレス解決プロトコル）要求の間隔（秒）
ARP_LEARNED_INTERVAL	14400	整数（60～31536000）	学習したホストの ARP 要求の間隔（秒）
ARP_GRATUITOUS_INTERVAL	15	整数（10～31536000）	gratuitous ARP 要求の間隔（秒）
ARP_RATE	10	整数（1 ～ 60）	ARP 再試行の間隔（秒）
ARP_RETRIES	3	整数（2 ～ 15）	ホスト ダウンのフラグを立てる前に ARP を再試行する回数
ARP_LEARN_MODE	1	整数（0 ～ 1）	CSM-S が応答のみ（0）の MAC アドレスを学習するか、すべてのトラフィック（1）の MAC アドレスを学習するかを指定します。
ARP_REPLY_FOR_NO_INSERVICE_VIP	D	0	整数（0 ～ 1）
ADVERTISE_RHI_FREQ	10	整数（1 ～ 65535）	CSM-S が RHI アップデートをチェックする回数（秒）
AGGREGATE_BACKUP_SF_STATE_TO_VS	0	整数（0 ～ 1）	仮想サーバの状態に、バックアップ サーバ ファームの動作可能状態を含めるかどうかを指定します。
COOKIE_INSERT_EXPIRATION_DATE	Fri, 1 Jan 2010 01:01:50 GMT	ストリング（2 ～ 63 文字）	CSM-S によって挿入される HTTP Cookie の期限切れの時刻および日付を設定します。
DEST_UNREACHABLE_MASK	65535	整数（0 ～ 65535）	Internet Control Message Protocol（ICMP）宛先到達不能コードの転送をビットマスクで定義します。
FT_FLOW_REFRESH_INT	60	整数（1 ～ 65535）	フォールトトレラントのスロー パス フローのリフレッシュ間隔（秒）
HTTP_CASE_SENSITIVE_MATCHING	1	整数（0 ～ 1）	URL（Cookie、ヘッダー）の一致および固定で、大文字と小文字を区別するかどうかを指定します。
HTTP_URL_COOKIE_DELIMITERS	/?&#+	ストリング（1 ～ 64 文字）	URL ストリングの Cookie の区切り文字のリストを設定します。
MAX_PARSE_LEN_MULTIPLIER	1	整数（1 ～ 16）	設定した max-parse-len をこの総数で乗算します。
NAT_CLIENT_HASH_SOURCE_PORT	0	整数（0 ～ 1）	送信元ポートを使用してクライアントの Network Address Translation（NAT; ネットワーク アドレス変換）IP アドレスを取得するかどうかを指定します。

表 10-1　CSM-S 環境変数（続き）

名前	デフォルト	有効値	説明
ROUTE_UNKNOWN_FLOW_PKTS	0	整数（0 ～ 1）	既存のフローと一致しない non-SYN パケットをルーティングするかどうかを指定します。
NO_RESET_UNIDIRECTIONAL_FLOWS	0	整数（0 ～ 1）	設定されている場合、タイムアウト時に単一方向フローをリセットしないように指定します。
SWITCHOVER_RP_ACTION	0	整数（0 ～ 1）	スーパバイザ エンジン Route Processor（RP）のスイッチオーバーが発生したあとで、復旧（0）または停止 / 再起動（1）するかどうかを指定します。
SWITCHOVER_SP_ACTION	0	整数（0 ～ 1）	スーパバイザ エンジン Switch Processor（SP）のスイッチオーバーが発生したあとで、復旧（0）または停止 / 再起動（1）するかどうかを指定します。
SYN_COOKIE_INTERVAL	3	整数（1 ～ 60）	新しい Syn-Cookie キーが生成される間隔を指定します（秒）。
SYN_COOKIE_THRESHOLD	5000	整数（0 ～ 1048576）	Syn-Cookie の動作のスレッシュホールドを指定します（中断されているセッション数）。
TCP_MSS_OPTION	1460	整数（1 ～ 65535）	レイヤ 7 の処理に対して CSM-S が送信できる最大セグメント サイズ（MSS）値を指定します。
TCP_WND_SIZE_OPTION	8192	整数（1 ～ 65535）	レイヤ 7 の処理に対して CSM-S が送信できるウィンドウ サイズ値を指定します。
VSERVER_ICMP_ALWAYS_RESPOND	false	ストリング（1 ～ 5 文字）	「true」の場合、仮想サーバの状態に関わらず ICMP プローブに応答します。
XML_CONFIG_AUTH_TYPE	Basic	ストリング（5 ～ 6 文字）	Xml-Config に対して HTTP 認証タイプを指定します。Basic または Digest です。

コンフィギュレーションの環境変数を表示する例を示します。

```
Router# show mod csm 5 variable

variable                          value
----------------------------------------------------------------
ARP_INTERVAL                      300
ARP_LEARNED_INTERVAL              14400
ARP_GRATUITOUS_INTERVAL           15
ARP_RATE                          10
ARP_RETRIES                       3
ARP_LEARN_MODE                    1
ARP_REPLY_FOR_NO_INSERVICE_VIP    0
ADVERTISE_RHI_FREQ                10
AGGREGATE_BACKUP_SF_STATE_TO_VS   0
DEST_UNREACHABLE_MASK             0xffff
FT_FLOW_REFRESH_INT               60
GSLB_LICENSE_KEY                  (no valid license)
HTTP_CASE_SENSITIVE_MATCHING      1
MAX_PARSE_LEN_MULTIPLIER          1
NAT_CLIENT_HASH_SOURCE_PORT       0
ROUTE_UNKNOWN_FLOW_PKTS           0
NO_RESET_UNIDIRECTIONAL_FLOWS     0
SYN_COOKIE_INTERVAL               3
SYN_COOKIE_THRESHOLD              5000
TCP_MSS_OPTION                    1460
TCP_WND_SIZE_OPTION               8192
VSERVER_ICMP_ALWAYS_RESPOND       false
XML_CONFIG_AUTH_TYPE              Basic
Cat6k-2#
```

コンフィギュレーションの現在の環境変数セットのすべての情報を表示するには、次のように **show module csm** *slot* **variable** [**detail**] コマンドを使用します。

```
Cat6k-2# show mod csm 5 variable detail
Name:ARP_INTERVAL  Rights:RW
Value:300
Default:300
Valid values:Integer (15 to 31536000)
Description:
Time (in seconds) between ARPs for configured hosts

Name:ARP_LEARNED_INTERVAL  Rights:RW
Value:14400
Default:14400
Valid values:Integer (60 to 31536000)
Description:
Time (in seconds) between ARPs for learned hosts

Name:ARP_GRATUITOUS_INTERVAL  Rights:RW
Value:15
Default:15
Valid values:Integer (10 to 31536000)
Description:
Time (in seconds) between gratuitous ARPs

Name:ARP_RATE  Rights:RW
Value:10
Default:10
Valid values:Integer (1 to 60)
Description:
Seconds between ARP retries
```

```
Name:ARP_RETRIES  Rights:RW
Value:3
Default:3
Valid values:Integer (2 to 15)
Description:
Count of ARP attempts before flagging a host as down

Name:ARP_LEARN_MODE  Rights:RW
Value:1
Default:1
Valid values:Integer (0 to 1)
Description:
Indicates whether CSM-S learns MAC address on responses only (0) or all traffic (1)

Name:ARP_REPLY_FOR_NO_INSERVICE_VIP  Rights:RW
Value:0
Default:0
Valid values:Integer (0 to 1)
Description:
Whether the CSM-S would reply to ARP for out-of-service vserver

Name:ADVERTISE_RHI_FREQ  Rights:RW
Value:10
Default:10
Valid values:Integer (1 to 65535)
Description:
The frequency in second(s) the CSM-S will check for RHI updates

Name:AGGREGATE_BACKUP_SF_STATE_TO_VS  Rights:RW
Value:0
Default:0
Valid values:Integer (0 to 1)
Description:
Whether to include the operational state of a backup serverfarm into the state of a
virtual server

Name:DEST_UNREACHABLE_MASK  Rights:RW
Value:0xffff
Default:65535
Valid values:Integer (0 to 65535)
Description:
Bitmask defining which ICMP destination unreachable codes are to be forwarded

Name:FT_FLOW_REFRESH_INT  Rights:RW
Value:60
Default:60
Valid values:Integer (1 to 65535)
Description:
FT slowpath flow refresh interval in seconds

Name:GSLB_LICENSE_KEY  Rights:RW
Value:(no valid license)
Default:(no valid license)
Valid values:String (1 to 63 chars)
Description:
License key string to enable GSLB feature

Name:HTTP_CASE_SENSITIVE_MATCHING  Rights:RW
Value:1
Default:1
Valid values:Integer (0 to 1)
Description:
Whether the URL (Cookie, Header) matching and sticky to be case sensitive

Name:MAX_PARSE_LEN_MULTIPLIER  Rights:RW
Value:1
Default:1
Valid values:Integer (1 to 16)
Description:
Multiply the configured max-parse-len by this amount
```

```
Name:NAT_CLIENT_HASH_SOURCE_PORT  Rights:RW
Value:0
Default:0
Valid values:Integer (0 to 1)
Description:
Whether to use the source port to pick client NAT IP address

Name:ROUTE_UNKNOWN_FLOW_PKTS  Rights:RW
Value:0
Default:0
Valid values:Integer (0 to 1)
Description:
Whether to route non-SYN packets that do not matched any existing flows

Name:NO_RESET_UNIDIRECTIONAL_FLOWS  Rights:RW
Value:0
Default:0
Valid values:Integer (0 to 1)
Description:
If set, unidirectional flows will not be reset when timed out

Name:SYN_COOKIE_INTERVAL  Rights:RW
Value:3
Default:3
Valid values:Integer (1 to 60)
Description:
The interval, in seconds, at which a new syn-cookie key is generated

Name:SYN_COOKIE_THRESHOLD  Rights:RW
Value:5000
Default:5000
Valid values:Integer (0 to 1048576)
Description:
The threshold (in number of pending sessions) at which syn-cookie is engaged

Name:TCP_MSS_OPTION  Rights:RW
Value:1460
Default:1460
Valid values:Integer (1 to 65535)
Description:
Maximum Segment Size (MSS) value sent by CSM-S for L7 processing

Name:TCP_WND_SIZE_OPTION  Rights:RW
Value:8192
Default:8192
Valid values:Integer (1 to 65535)
Description:
Window Size value sent by CSM-S for L7 processing

Name:VSERVER_ICMP_ALWAYS_RESPOND  Rights:RW
Value:false
Default:false
Valid values:String (1 to 5 chars)
Description:
If "true" respond to ICMP probes regardless of vserver state

Name:XML_CONFIG_AUTH_TYPE  Rights:RW
Value:Basic
Default:Basic
Valid values:String (5 to 6 chars)
Description:
HTTP authentication type for xml-config:Basic or Digest
```

連続（persistent）接続の設定

CSM-S では、HTTP ヘッダー内の URL、Cookie、またはその他のフィールドに基づいて HTTP 接続をスイッチングできます。CSM-S の連続（persistent）接続サポートにより、連続接続での後続 HTTP 要求はそれぞれ別々にスイッチング可能です。新しい HTTP 要求が届いた場合、その要求は、前の要求と同じサーバにスイッチングしたり、別のサーバにスイッチングしたり、またはクライアントにリセットしてその要求が完了しないように設定することができます。

ソフトウェア Release 2.1(1) の時点で、CSM-S は HTTP 1.1 の連続機能（persistence）をサポートしています。この機能によって、ブラウザは 1 つの連続接続で複数の HTTP 要求を送信できます。連続接続の確立後、サーバは同じクライアントからさらに要求が届く可能性を想定して、設定可能なインターバルの間、接続をオープンな状態にしておきます。連続接続によって、要求のたびに新しい TCP 接続を確立することに伴うオーバーヘッドが排除されます。

HTTP 1.1 の連続機能はデフォルトとして、レイヤ 7 ポリシーで設定されたすべての仮想サーバでイネーブルです。連続接続をディセーブルにする場合は、**no persistent rebalance** コマンドを入力します。連続接続をイネーブルにする場合は、**persistent rebalance** コマンドを入力します。

次に、連続接続を設定する例を示します。

```
Router# configure terminal
Enter configuration commands, one per line.  End with
CNTL/Z.
Router(config)# mod csm 2
!!! configuring serverfarm
Router(config-module-csm)# serverfarm sf3
Router(config-slb-sfarm)# real 10.1.0.105
Router(config-slb-real)# inservice
!!! configuring vserver
Router(config-slb-real)# vserver vs3
Router(config-slb-vserver)# virtual 10.1.0.83 tcp 80
Router(config-slb-vserver)# persistent rebalance
Router(config-slb-vserver)# serverfarm sf3
Router(config-slb-vserver)# inservice
Router(config-slb-vserver)# end
```

HTTP ヘッダー挿入

HTTP のヘッダー挿入は、CSM-S にクライアント IP アドレスの HTTP ヘッダーへの挿入のような情報を挿入させる機能です。この機能は、CSM-S が送信元 NAT を実行していて、さらにサーバ側のアプリケーションがまだ最初の送信元 IP を要求している場合に特に有効です。

CSM-S は、クライアントからサーバ方向のヘッダーに、クライアントからの送信元 IP アドレスを挿入できます。

HTTP ヘッダーに情報を挿入するには、**insert protocol http header** *name* **header-value** *value* コマンドを使用します。

- *name* **—** HTTP ヘッダーの一般フィールドの文字どおりの名前。名前は 1 ～ 63 文字のストリングです。
- *value* **—** 要求に挿入するヘッダー値のストリングを文字どおり指定します。

  また、ヘッダー値の特殊パラメータ %is および %id を使用できます。%is 値は HTTP ヘッダーに送信元 IP アドレスを挿入し、%id 値は HTTP ヘッダーに宛先 IP アドレスを挿入します。特殊パラメータはそれぞれヘッダー マップごとに 1 度ずつ指定できます。

（注）　ヘッダー マップには、複数の挿入ヘッダーが含まれることがあります。スペースを含む複数のキーワードからなるヘッダー値を挿入する場合、全体を二重引用符で囲む必要があります。

HTTP ヘッダー挿入を設定する場合、ヘッダー マップおよびポリシーを使用する必要があります。HTTP ヘッダー挿入を機能させるには、デフォルトのポリシーは適用できません。

次に、ヘッダー フィールドと値を指定して、要求を検索する例を示します。

```
Cat6k-2(config-module-csm)# natpool TESTPOOL 10.10.110.200 10.10.110.210 netmask
255.255.255.0
!
Cat6k-2(config-module-csm)# map HEADER-INSERT header
Cat6k-2(config-slb-map-header)# insert protocol http header Source-IP header-value %is
Cat6k-2(config-slb-map-header)# insert protocol http header User-Agent header-value
"MyBrowser 1.0"
!
Cat6k-2(config-module-csm)# real SERVER1
Cat6k-2(config-slb-real)# address 10.10.110.10
Cat6k-2(config-slb-real)# inservice
Cat6k-2(config-module-csm)# real SERVER2
Cat6k-2(config-slb-real)# address 10.10.110.20
Cat6k-2(config-slb-real)# inservice
!
Cat6k-2(config-module-csm)# serverfarm FARM-B
Cat6k-2(config-slb-sfarm)# nat server
Cat6k-2(config-slb-sfarm)# nat client TESTPOOL
Cat6k-2(config-slb-real)# real name SERVER1
Cat6k-2(config-slb-real)# inservice
Cat6k-2(config-slb-real)# real name SERVER2
Cat6k-2(config-slb-real)# inservice
!
Cat6k-2(config-module-csm)# policy INSERT
Cat6k-2(config-slb-policy)# header-map HEADER-INSERT
Cat6k-2(config-slb-policy)# serverfarm FARM-B
!
Cat6k-2(config-module-csm)# vserver WEB
Cat6k-2(config-slb-vserver)# virtual 10.10.111.100 tcp www
Cat6k-2(config-slb-vserver)# persistent rebalance
Cat6k-2(config-slb-vserver)# slb-policy INSERT
Cat6k-2(config-slb-vserver)# inservice
```

GSLB の設定

ここでは、CSM Global Server Load Balancing（GSLB; グローバル サーバ ロードバランシング）の拡張機能セット オプション、およびその使用方法について説明します。拡張機能セット オプションを使用する前に、「はじめに」の「ライセンス」（p.xxvii）および表紙裏に記載されているソフトウェア使用許諾書の諸条件を入念に確認してください。

（注）　ソフトウェアをダウンロードまたはインストールすることにより、使用許諾書に同意することになります。この許諾書のすべての条項に同意できない場合は、ソフトウェアをダウンロード、インストール、または使用しないでください。

GSLB 拡張機能セット オプションの使用

GSLB をイネーブルにするには、イネーブル モードで次の手順を実行します。

コマンド	目的
`Router# config t` `Router(config)# mod csm 5`	コンフィギュレーション モードを開始して、特定の CSM-S（たとえば、ここで使用されているモジュール 5 など）の CSM-S コンフィギュレーション モードを開始します。
`Router(config-module-csm)# variable` *name value*	次のように名前と値を指定して、GSLB をイネーブルにします。 Name = [1] Value =
`Router(config-module-csm)# exit` `Router (config)# write mem`	CSM-S モジュール コンフィギュレーション モードを終了して、設定の変更を保存します。
`Router#:hw-module` *slot number* `reset`	CSM-S を再起動して、変更をアクティブにします。

1. GSLB では、別途ライセンスを購入する必要があります。ご使用の GSLB のライセンスを購入する場合は、製品を購入された代理店にご連絡ください。

表 10-2 に、CSM-S で使用する GSLB の環境変数の値を示します。

表 10-2　GSLB 環境変数の値

名前	デフォルト	有効値	説明
GSLB_LICENSE_KEY	（有効なライセンスはありません）	ストリング（1 ～ 63 文字）	GSLB 機能をイネーブルにするライセンス キーのストリング
GSLB_KALAP_UDP_PORT	5002	整数（1 ～ 65535）	GSLB KAL-AP UDP のポート番号を指定します。
GSLB_KALAP_PROBE_FREQ	45	整数（45 ～ 65535）	GSLB KAL-AP プローブの頻度を指定します。
GSLB_KALAP_PROBE_RETRIES	3	整数（1 ～ 65535）	GSLB KAL-AP プローブの最大再試行回数を指定します。
GSLB_ICMP_PROBE_FREQ	45	整数（45 ～ 65535）	GSLB ICMP プローブの頻度を指定します。
GSLB_ICMP_PROBE_RETRIES	3	整数（1 ～ 65535）	GSLB ICMPプローブの最大再試行回数を指定します。
GSLB_HTTP_PROBE_FREQ	45	整数（45 ～ 65535）	GSLB HTTP プローブの頻度を指定します。

表 10-2　GSLB 環境変数の値（続き）

名前	デフォルト	有効値	説明
GSLB_HTTP_PROBE_RETRIES	3	整数（1 ～ 65535）	GSLB HTTP プローブの最大再試行回数を指定します。
GSLB_DNS_PROBE_FREQ	45	整数（45 ～ 65535）	GSLB DNS プローブの頻度を指定します。
GSLB_DNS_PROBE_RETRIES	3	整数（1 ～ 65535）	GSLB DNS プローブの最大再試行回数を指定します。

GSLB の設定

GSLB は、利用可能な負荷に基づき、Domain Name Server（DNS; ドメイン ネーム サーバ）を通してさまざまなサーバ ファームおよび実サーバにクライアント接続を振り分けることによって、散在している複数のホスト サイト間で負荷を分散させます。GSLB は、アクセス リスト、マップ、サーバ ファーム、およびロードバランス アルゴリズムを使用して実行されます。表 10-3 に、CSM-S 上で GSLB を設定するための要件をまとめます。

表 10-3　GSLB の動作

クライアント要求（起点）	ドメイン（目的）	サーバ ファーム（終点）	アルゴリズム（方法）
着信DNS要求のフィルタリングには、アクセス リストを使用できます。着信 DNS 要求に設定されているマップ、クライアント グループ、およびサーバ ファームを関連付けるには、ポリシーを使用します。	マップを設定して、クライアント要求と一致しなければならないドメイン名を指定します。正規表現の構文を使用できます。 たとえば、ドメイン名は cnn.com または yahoo.com であり、これにクライアント要求を突き合せます。ドメイン名が指定されたポリシー マップと一致した場合、プライマリ サーバ ファームに対して、要求に応じる実サーバがあるかどうかを問い合わせます。	サーバ ファームでは、クライアント要求を満たす情報の検索先として、実サーバ グループを指定します。	ターゲット実サーバのアベイラビリティを判別するために、GSLB プローブを使用できます。実サーバに設定されているプローブ タイプを使用します。 GSLB サーバ ファーム プレディクタは、ラウンドロビン式の最小負荷、順序付きリスト、ハッシュ アドレス ソース、ハッシュ ドメイン、およびハッシュ ドメイン アドレス ソースです。

図 10-1 に、GSLB の基本的な設定を示します。

図 10-1　GSLB の設定

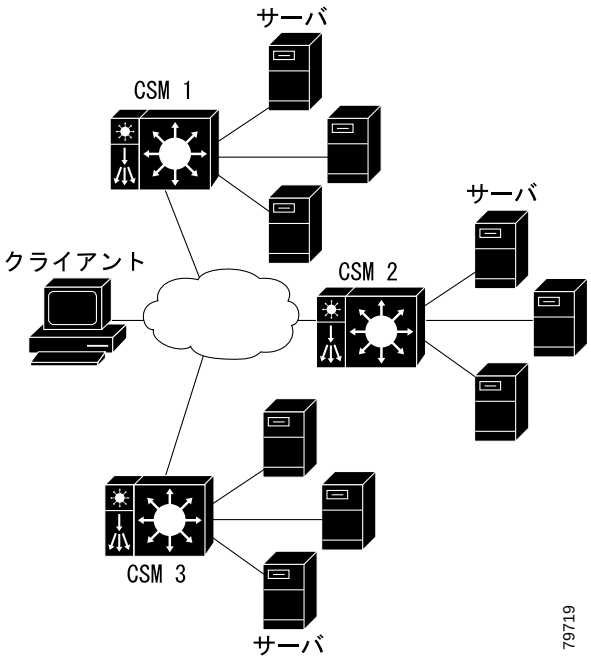


図 10-1 の場合、設定作業と例に関する注意事項は次のとおりです。

- CSM-S 1 は GSLB および SLB の両方を実行しますが、CSM-S 2 および CSM-S 3 が実行するのは SLB だけです。
- CSM-S 1 には、SLB 用の仮想サーバと GSLB 用の仮想サーバがあります。SLB 用の仮想サーバでは、サーバ ファームの実サーバはローカル サーバの IP アドレスです。
- DNS ポリシーではプライマリ サーバ ファームを使用します。実サーバの 1 つがローカルで、ほかの 2 つの実サーバは、それぞれ CSM-S 2 および CSM-S 3 上で設定された仮想サーバです。
- 両方のリモート ロケーション、ローカル実サーバ、および仮想サーバにプローブを追加する必要があります。
- CSM-S 1 の管理用 IP アドレス（CSM-S 1 の VLAN アドレスまたはエイリアス IP）に送信された DNS 要求には、GSLBFARM というサーバ ファームで設定された 3 つの実サーバ IP アドレスのうちの 1 つが応答として与えられます。

GSLB を設定する手順は、次のとおりです。

	コマンド	目的
ステップ 1	`Router(config-slb-vserver)# serverfarm serverfarm-name`	仮想サーバに関連付けるサーバ ファームを作成します。
ステップ 2	`Router(config-module-csm)# vserver virtserver-name`	CSM-S 1 上で SLB 用の仮想サーバを指定し、仮想サーバ サブモードを開始します。
ステップ 3	`Router(config-slb-vserver)# virtual ip-address [ip-mask] protocol port-number [service ftp]`	仮想サーバの属性を設定します。
ステップ 4	`Router(config-slb-vserver)# inservice`	仮想サーバのロードバランシングをイネーブルにします。

	コマンド	目的
ステップ 5	`Router(config-module-csm)# vserver virtserver-name dns`	GSLB 用の仮想サーバを指定し、仮想サーバ サブモードを開始します。
ステップ 6	`Router(config-slb-vserver)# dns-policy [group group-id] [netmask ip-netmask]`	同じクライアントからの接続には同じサーバ ファームが使用されるようにします。
ステップ 7	`Router(config-slb-vserver)# inservice`	仮想サーバの GSLB をイネーブルにします。
ステップ 8	`Router(config-module-csm)# serverfarm GSLBFARM dns-vip`	GSLBFARM サーバ ファーム（実際にはフォワーディング ポリシー）を作成して名前を指定し、サーバ ファーム コンフィギュレーション モードを開始します。
ステップ 9	`Router(config-slb-sfarm)# predictor hash address source`	サーバ ファームにロードバランス プレディクタのハッシュ アドレス ソースを設定します。
ステップ 10	`Router(config-module-csm)# real ip-addres`	実サーバのエイリアス IP アドレスを指定し、実サーバ コンフィギュレーション サブモードを開始します。
ステップ 11	`Router(config-slb-real)# inservice`	仮想サーバのロードバランシングをイネーブルにします。
ステップ 12	`Router(config-module-csm)#map dns-map-name dns`	DNS マップを設定します。
ステップ 13	`Router(config-dns-map)# match protocol dns domain name`	DNS マップに DNS 名を追加します。
ステップ 14	`Router(config-module-csm)#policy policy name`	ポリシーを設定します。
ステップ 15	`Router(config-slb-policy)# dns map map_name`	ポリシーに DNS マップ属性を追加します。
ステップ 16	`Router(config-slb-policy)#serverfarm primary-serverfarm [backup sorry-serverfarm [sticky]]`	サーバ ファームとポリシーを関連付けます。
ステップ 17	`Router(config-module-csm)# vserver virtserver-name`	CSM-S 2 上で仮想サーバを設定し、仮想サーバ サブモードを開始します。
ステップ 18	`Router(config-slb-vserver)# virtual ip-address [ip-mask] protocol port-number [service ftp]`	仮想サーバの属性を設定します。
ステップ 19	`Router(config-slb-vserver)# serverfarm serverfarm-name`	サーバ ファームと仮想サーバを関連付けます。
ステップ 20	`Router(config-slb-vserver)# inservice`	仮想サーバのロードバランシングをイネーブルにします。
ステップ 21	`Router(config-module-csm)# vserver virtserver-name`	CSM-S 3 上で仮想サーバを設定し、仮想サーバ サブモードを開始します。
ステップ 22	`Router(config-slb-vserver)# virtual ip-address [ip-mask] protocol port-number [service ftp]`	仮想サーバの属性を設定します。
ステップ 23	`Router(config-slb-vserver)# serverfarm serverfarm-name`	サーバ ファームと仮想サーバを関連付けます。
ステップ 24	`Router(config-slb-vserver)# inservice`	仮想サーバのロードバランシングをイネーブルにします。

次に、GSLB を設定する例を示します。

CSM 1 上で：

```
Router(config-module-csm)# serverfarm WEBFARM
Router(config-slb-sfarm)# predictor round-robin
Router(config-slb-sfarm)# real 3.5.5.5
Router(config-slb-real)# inservice
Router(config-slb-sfarm)# real 3.5.5.6
Router(config-slb-real)# inservice
Router(config-slb-real)# exit
Router(config-slb-sfarm)# exit

Router(config-module-csm)# vserver WEB
Router(config-slb-vserver)# virtual 10.10.10.10 tcp www
Router(config-slb-vserver)# serverfarm WEBFARM
Router(config-slb-vserver)# inservice

Router(config-module-csm)# serverfarm GSLBSERVERFARM dns-vip
Router(config-slb-sfarm)# predictor round-robine
Router(config-slb-sfarm)# real 10.10.10.10
Router(config-slb-real)# inservice
Router(config-slb-real)# exit
Router(config-slb-sfarm)# real 20.20.20.20
Router(config-slb-real)# inservice
Router(config-slb-real)# exit
Router(config-slb-sfarm)# real 30.30.30.30
Router(config-slb-real)# inservice
Router(config-slb-real)# exit

Router(config-module-csm)# map MAP1 dns
Router(config-dns-map)# match protocol dns domain foobar.com
Router(config-dns-map)# exit

Router(config-module-csm)# policy DNSPOLICY dns
Router(config-slb-policy)# dns map MAP1
Router(config-slb-policy)# serverfarm primary GSLBSERVERFARM ttl 20 responses 1
Router(config-slb-policy)# exit

Router(config-module-csm)# vserver DNSVSERVER dns
Router(config-slb-vserver)# dns-policy DNSPOLICY
Router(config-slb-vserver)# inservice
```

CSM-S 2 上で：

```
Router(config-module-csm)# serverfarm WEBFARM
Router(config-slb-sfarm)# predictor round-robin
Router(config-slb-sfarm)# real 4.5.5.5
Router(config-slb-real)# inservice
Router(config-slb-sfarm)# real 4.5.5.6
Router(config-slb-real)# inservice
Router(config-slb-real)# exit
Router(config-slb-sfarm)# exit

Router(config-module-csm)# vserver WEB
Router(config-slb-vserver)# virtual 20.20.20.20 tcp www
Router(config-slb-vserver)#s erverfarm WEBFARM
Router(config-slb-vserver)# inservice
```

CSM-S 3 上で：

```
Router(config-module-csm)# serverfarm WEBFARM
Router(config-slb-sfarm)# predictor round-robin
Router(config-slb-sfarm)# real 5.5.5.5
Router(config-slb-real)# inservice
Router(config-slb-sfarm)# real 5.5.5.6
Router(config-slb-real)# inservice
Router(config-slb-real)# exit
Router(config-slb-sfarm)# exit
Router(config-module-csm)# vserver WEB
Router(config-slb-vserver)# virtual 30.30.30.30 tcp www
Router(config-slb-vserver)# serverfarm WEBFARM
Router(config-slb-vserver)# inservice
```

ネットワーク管理の設定

ここでは、ネットワーク上での CSM-S の管理方法について説明します。

実サーバの SNMP トラップの設定

SNMP（簡易ネットワーク管理プロトコル）トラップはイネーブルの場合、実サーバのステートが変わるたびに（たとえば、サーバがサービスを開始または停止するたびに）外部の管理装置に送信されます。トラップには、実サーバ トラップであることを示す Object Identifier（OID; オブジェクト識別子）が含まれます。

（注）　実サーバ トラップの OID は 1.3.6.1.4.1.9.9.161.2 です。

トラップには、サーバ ステートが変わった理由を示すメッセージも含まれます。

Catalyst 6500 シリーズ スイッチの SLB 機能に関連付けられたフォールトトレラント トラップをイネーブルまたはディセーブルにするには、**snmp-server enable traps slb ft** コマンドを使用します。フォールトトレラント トラップは、SLB のフォールトトレラントの要素を扱います。たとえば、フォールトトレラント トラップがイネーブルで、SLB 装置がフォールトトレラント ピアの障害を検出した場合、その SLB 装置はスタンバイからアクティブに変わるときに、SNMP トラップを送信します。

実サーバ用の SNMP トラップを設定する手順は、次のとおりです。

	コマンド	目的
ステップ 1	`Router (config)# snmp-server community public`	通知動作で送信される、パスワードと同様のコミュニティ ストリングを定義します。**public** はその一例です。
ステップ 2	`Router (config)# snmp-server host host-addr`	トラップの送信先となる、外部ネットワーク管理装置の IP アドレスを定義します。
ステップ 3	`Router (config)# snmp-server enable traps slb csrp`	実サーバ用の SNMP トラップをイネーブルにします[1]。

1. SNMP フォールトトレラント トラップ機能をディセーブルにする場合は、このコマンドの **no** 形式を使用します。

XML インターフェイスの設定

従来のリリースでは、Cisco IOS CLI（コマンドライン インターフェイス）が CSM-S を設定する唯一の手段でした。XML により、Document Type Definition（DTD）を使用して CSM-S を設定できます（XML DTD の例については、付録 D「CSM XML の DTD」を参照してください）。

CSM-S で XML を使用する場合、注意事項は次のとおりです。

- 同時に使用できるクライアント接続は最大で 5 つです。
- XML の設定は IP SLB モードとは無関係ですが、**csm_module slot='x' sense='no'** コマンドの場合は例外的に所定の結果をもたらし、XML エラーを生成します。
- パイプラインの HTTP POST はサポートされません。
- すべてのクライアント通信が 30 秒でタイムアウトします。

- クライアント証明書が不良だった場合、Cisco IOS のシステム ログにメッセージが送信されます。
- 異なるスロット属性を指定することによって、1 つの CSM-S をほかの CSM-S コンフィギュレーションのプロキシにすることができます。

この機能をイネーブルにすると、ネットワーク管理装置が CSM-S に接続し、新しい設定を装置に送信する場合があります。ネットワーク管理装置は、標準の HTTP プロトコルを使用して、コンフィギュレーション コマンドを CSM-S に送信します。HTTP POST のデータ部分で、XML 文書を CSM-S に送信することによって、新しい設定が適用されます。

HTTP 会話の例を示します。

```
******** Client **************
POST /xml-config HTTP/1.1
Authorization: Basic VTpQ
Content-Length: 95

<?xml version="1.0"?>
<config><csm_module slot="4"><vserver name="FOO"/></csm_module></config>
******** Server **************
HTTP/1.1 200 OK
Content-Length: 21

<?xml version="1.0"?>
******** Client **************
POST /xml-config HTTP/1.1
Content-Length: 95

<?xml version="1.0"?>
<config><csm_module slot="4"><vserver name="FOO"/></csm_module></config>
******** Server **************
HTTP/1.1 401 Unauthorized
Connection: close
WWW-Authenticate: Basic realm=/xml-config
```

表 10-4 に、サポート対象の HTTP 戻りコードを示します。

表 10-4　XML に関する HTTP の戻りコード

戻りコード	説明
200	OK
400	不良要求
401	未承認（要求した証明書が提出されなかった）
403	禁止（無効な証明書が提出され、Syslog も生成された）
404	未検出（「/xml-config」が指定されていない）
408	要求のタイムアウト（受信待ちで 30 秒以上経過）
411	コンテンツ長の脱落（Content-Length フィールドが脱落またはゼロ）
500	内部サーバ エラー
501	実装されていない（POST が指定されていない）
505	サポートされない HTTP バージョン（1.0 または 1.1 が指定されていない）

次の HTTP ヘッダーがサポートされます。

- Content-Length（すべての POST にゼロ以外の値が必要）
- Connection（*close* 値は要求を連続 [persistent] させないことを指定）
- WWW-Authenticate（要求した証明書がない場合にクライアントに送信）
- Authorization（base64 符号化方式による基本証明書を指定するためにクライアントから送信）

XML 機能を動作させるには、ネットワーク管理システムがスイッチ インターフェイスの IP アドレスではなく、CSM-S の IP アドレスに接続する必要があります。

コマンドライン インターフェイスの場合と同様、コンフィギュレーションのマスター コピーを Cisco IOS ソフトウェアに保存しなければならないので、XML コンフィギュレーション要求を受信した CSM-S は、これらの要求をスーパーバイザ エンジンに送らなければなりません。

（注）　XML コンフィギュレーションによって、1 つの CSM を同一スイッチ シャーシに搭載されたすべての CSM-S のプロキシとして動作させることができます。たとえば、ある CSM-S 用のコンフィギュレーションが含まれる XML ページを、同じスイッチ シャーシに搭載された別の CSM-S から正しく提供できます。

現在、一般公開されている DTD は、作成する XML コンフィギュレーション文書の基盤です（付録 D「CSM XML の DTD」を参照）。XML 文書は HTTP POST 要求によって、CSM-S に直接送られます。XML を使用するには、Cisco IOS の CLI を使用して、前もって CSM-S 上で最小限のコンフィギュレーションを作成しておく必要があります。**xml-config** コマンドについては、『*Catalyst 6500 Series Content Switching Module Command Reference*』を参照してください。

応答は要求をミラー化した XML 文書です。問題のある要素にはチャイルド エラー要素でフラグが設定され、エラー コードおよびエラー文字列が示されます。XML 文書でルート要素の属性を使用することによって、無視すべきエラーのタイプを指定できます。

イネーブル / ディセーブル機能とともに、TCP ポート、クライアント アクセス リストのセキュリティ オプション、および HTTP 認証がサポートされます。

CSM-S 上で XML を設定する手順は、次のとおりです。

	コマンド	目的
ステップ 1	`Router(config-module-csm)# module csm slot`	モジュールおよびスロット番号を指定します。
ステップ 2	`Router(config-module-csm)# xml-config`	CSM-S 上で XML をイネーブルにして、XML 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	`Router(config-slb-xml)# port port-number`	CSM-S HTTP サーバが待ち受ける TCP ポートを指定します。
ステップ 4	`Router(config-slb-xml)# vlan id`	CSM-S HTTP サーバが指定された VLAN からの接続だけを受け付けるように制限します。
ステップ 5	`Router(config-slb-xml)# client-group [1-99 | name]`	CSM-S XML コンフィギュレーション インターフェイスが受け付けるのは、クライアント グループと一致する IP アドレスからの接続だけであることを指定します。
ステップ 6	`Router(config-slb-xml)# credentials user-name password`	ユーザ名とパスワードのコンビネーションを 1 つまたは複数設定します。**credentials** コマンドを 1 つまたは複数設定した場合、CSM-S HTTP サーバは RFC 2617 で規定された基本認証方式を使用して、ユーザ アクセスを認証します。
ステップ 7	`Router# show module csm 4 xml stats`	XML 統計情報のリストを表示します。 （注）　統計情報のカウンタは 32 ビットです。

CSM-S 上で XML を設定する例を示します。

```
Router(config-module-csm)# configure terminal
Router(config-module-csm)# module csm 4
Router(config-module-csm)# xml-config
Router(config-slb-xml)# port 23
Router(config-slb-xml)# vlan 200
Router(config-slb-xml)# client-group 60
Router(config-slb-xml)# credentials eric @#$#%%@
Router# show module csm 4 xml stats
```

許容できない XML エラーが発生した場合は、HTTP 応答に 200 というコードが含まれます。エラーが発生した元の XML 文書の一部が、エラー タイプと説明を示したエラー要素とともに戻されます。

仮想サーバ名が脱落している場合のエラー応答の例を示します。

```
      <?xml version="1.0"?>
      <config>
       <csm_module slot="4">
        <vserver>
         <error code="0x20">Missing attribute name in element
vserver</error>
        </vserver>
       </csm_module>
      </config>
```

戻されるエラー コードは、コンフィギュレーション要素のエラー許容属性のビットとも対応しています。戻される XML エラー コードは、次のとおりです。

```
XML_ERR_INTERNAL          = 0x0001,
XML_ERR_COMM_FAILURE      = 0x0002,
XML_ERR_WELLFORMEDNESS    = 0x0004,
XML_ERR_ATTR_UNRECOGNIZED = 0x0008,
XML_ERR_ATTR_INVALID      = 0x0010,
XML_ERR_ATTR_MISSING      = 0x0020,
XML_ERR_ELEM_UNRECOGNIZED = 0x0040,
XML_ERR_ELEM_INVALID      = 0x0080,
XML_ERR_ELEM_MISSING      = 0x0100,
XML_ERR_ELEM_CONTEXT      = 0x0200,
XML_ERR_IOS_PARSER        = 0x0400,
XML_ERR_IOS_MODULE_IN_USE = 0x0800,
XML_ERR_IOS_WRONG_MODULE  = 0x1000,
XML_ERR_IOS_CONFIG        = 0x2000
```

デフォルトの error_tolerance 値は 0x48 です。これは、認識されない属性および要素の無視と対応しています。

SASP の設定

Server Application State Protocol（SASP）によって、CSM-S は Workload Manager（WM）のレジスタからトラフィック ウェイトに関する推奨を受けることができます。さらに、WM から CSM-S に新しいロードバランシング グループ メンバーを推奨できます。

SASP は、Cisco IOS Release 12.1(13)E3 以降のリリースでサポートされます。また、4.1.2 以降のリリースをサポートする Cisco IOS リリースが必要です。

SASP を設定するには、サーバ ファーム（SASP グループなど）および DFP エージェント（SASP Global Workload Manager [GWM] など）に特殊な bind_id を関連付ける必要があります。

SASP グループの設定

SASP グループは、CSM-S 上のサーバ グループに相当します。グループを設定するには、**serverfarm** コンフィギュレーション コマンドを使用します。グループ メンバーはすべて、サーバ ファームに所属するものとして設定された実サーバです。このグループを GWM に関連付けるには、GWM と一致する SASP bind_id を割り当てます。SASP グループを設定するには、次のように、サーバ ファーム コンフィギュレーション サブメニューから **bindid** コマンドを使用します。

```
Router(config-slb-sfarm)# bindid 7
```

GWM の設定

GWM は DFP エージェントとして設定します。GWM を設定するには、CSM-S コンフィギュレーション コマンドから dfp サブメニューを開始する必要があります。次に、DFP エージェントとして GWM を設定する例を示します。

```
Router(config-slb-dfp)# agent ip.address port bind id
```

（注）　CLI から bind_id を入力することはできません。ただし、このエージェントを GWM として設定するには、bind_id が必須です。CLI では、bind_id キーワードを「アクティビティ タイムアウト」または「キープアライブ」として記述します。さらに 2 つの値を追加できます。ただし、SASP 環境のトラブルシューティング時を除き、追加の値は入力しないでください。

代わりに、GWM は次のように設定できます。

```
Router(config-slb-dfp)# agent ip.address port bind id flags
```

または

```
Router(config-slb-dfp)# agent ip.address port bind_id flags keep-alive-interval
```

キープアライブ インターバルは秒数です。デフォルトは 180 です。フラグは CSM-S が GWM にどのように登録するかを制御します。デフォルト値はゼロです。フラグの意味については、表 10-5 を参照してください。

表 10-5　SASP フラグ

フラグ値	意味
0	CSM-S のデフォルトの登録フラグ（37）を使用
32	GWMのデフォルト ロードバランシング登録を指定。ロードバランサは「Get Weights」メッセージを送信して新しいウェイトを取得し、GWM からそのウェイトを引き出します（pull）。 GWM には、このロードバランサにウェイトを送信するときに、（ウェイトが変わらないメンバーを含めて）すべてのグループ メンバーのウェイトを組み込む必要があります。
33	ロードバランサが「Send Weights」メッセージを介してウェイトを受信することを指定（GWM はロードバランサにウェイトを格納 [push]）
34	メンバーが開始した登録 / 登録解消を GWM に信頼させ、送信されたウェイトで登録 / 登録解消をただちに更新させます。
35	33 および 34 と同じ
36	GWM が前回の期間からウェイトが変化していないメンバーを含めてはならないことを指定
37	33 および 36 と同じ
38	34 および 36 と同じ
39	33、34、および 36 と同じ

代替 bind_id の設定

デフォルトでは 1 つの bind_id が SASP bind_id、65520 として設定されます。最初の bind_id は 1 ～ 65525 の任意の値にできます。次に、CSM-S のコンフィギュレーション コマンドを使用して bind_id を設定する例を示します。

```
Router(config-module-csm)# variable SASP_FIRST_BIND_ID value
```

SASP で使用できる bind_id の最大数は 8 です。これは、サポートされる GWM の最大数でもあります。bind_id の最大数は 0 ～ 8 の任意の値に設定できます。次に、使用する SASP bind_id の最大数を設定する例を示します。

```
Router(config-module-csm)# variable SASP_GWM_BIND_ID_MAX value
```

（注）　これらの環境変数を 1 つでも変更した場合は、CSM-S を再起動してください。

CSM-S 固有の ID 設定

CSM-S にはデフォルトで、「Cisco-CSM」という固有の識別ストリングが与えられます。次に、CSM-S のコンフィギュレーション コマンドを使用して、このストリングを設定する例を示します。

```
Router(config-module-csm)# variable SASP_CSM_UNIQUE_ID text
```

（注）　これらの環境変数を 1 つでも変更した場合は、CSM-S を再起動してください。

ウェイト スケーリングの設定

CSM-S 上の実サーバのウェイトは 0 ～ 100 です。メンバーの SASP ウェイトは 0 ～ 65536 です。GWM が CSM-S の範囲内でウェイトを作成するかぎり、スケーリングは不要です。GWM が SASP の全範囲を使用する場合は、この範囲をマッピングする必要があります。次に、SASP ウェイトのスケーリング例を示します。

```
Router(config-module-csm)# variable SASP_SCALE_WEIGHTS value
```

SASP_SCALE_WEIGHTS の範囲は 0 ～ 12 です。0 ～ 11 の値を指定すると、SASP ウェイトが 2 の n 乗で除算されます。値 12 を指定すると、65536 の値全体が CSM-S の 0 ～ 100 のウェイト範囲にマッピングされます。

次に、SASP GWM の詳細の表示例を示します。

```
Router# show module csm 3 dfp detail
DFP Agent 64.100.235.159:3860  Connection state: Connected
   Keepalive = 65521  Retry Count = 33     Interval = 180   (Default)
   Security errors = 0
   Last message received: 03:33:46 UTC 01/01/70
   Last reported Real weights for Protocol any, Port 0
      Host 10.9.10.22       Bind ID 65521  Weight 71
      Host 10.10.12.10      Bind ID 65521  Weight 70
      Host 10.10.12.12      Bind ID 65521  Weight 68
   Last reported Real weights for Protocol any, Port 44
      Host 10.9.10.9        Bind ID 65521  Weight 69

DFP manager listen port not configured
No weights to report to managers.
```

次に、SASP グループの表示例を示します。

```
Router# show module csm 3 serverfarms detail
SVRFARM2, type = SLB, predictor = RoundRobin, nat = SERVER
  virtuals inservice: 0, reals = 4, bind id = 65521, fail action = none
  inband health config: <none>
  retcode map = <none>
  Real servers:
    10.10.12.10, weight = 78, OUTOFSERVICE, conns = 0
    10.10.12.12, weight = 76, OPERATIONAL, conns = 0
    10.9.10.9:44, weight = 77, OPERATIONAL, conns = 0
    10.9.10.22, weight = 79, OUTOFSERVICE, conns = 0
  Total connections = 0
```

次に、SASP 環境変数の表示例を示します。

```
Router# show module csm 3 variable

variable                       value
-------------------------------------------------------------
ARP_INTERVAL                   300
…
ROUTE_UNKNOWN_FLOW_PKTS        0
SASP_FIRST_BIND_ID             65520
SASP_GWM_BIND_ID_MAX           2
SASP_CSM_UNIQUE_ID             paula jones
…
XML_CONFIG_AUTH_TYPE           Basic
```

バックエンドの暗号化

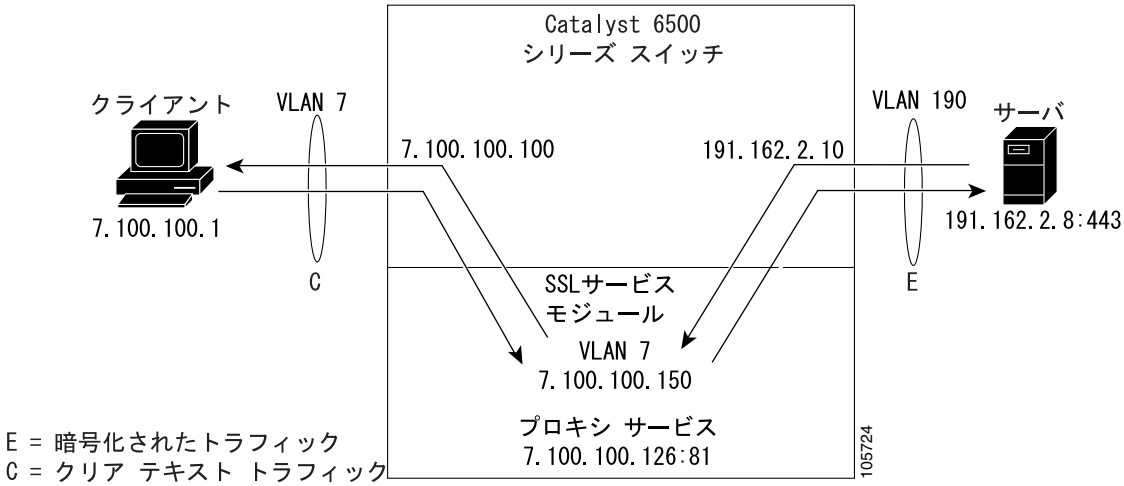
バックエンドの暗号化によって、安全なエンドツーエンド環境が実現します。図 10-2 では、クライアント（7.100.100.1）はスイッチ ポート 6/47 に接続して VLAN 7 にアクセスします。サーバ（191.162.2.8）は、スイッチ ポート 10/2 に接続して VLAN 190 にアクセスします。

SSL プロキシである VLAN 7 の設定は、次のとおりです。

- IP アドレス — 7.100.100.150
- スタティック ルートおよびゲートウェイ：
  - ルート 191.0.0.0
  - ゲートウェイ 7.100.100.100

  ゲートウェイの IP アドレス（MSFC 上の VLAN 7 の IP アドレス）は、未知のネットワークを宛先とするクライアント側のトラフィックがこのアドレスに転送され、そこからクライアントにルーティングされるように設定されています。
- クライアント側ゲートウェイ — 7.100.100.100（MSFC 上で設定された VLAN 7 の IP アドレス）
- クライアント プロキシ サービスの VIP — 7.100.100.150:81
- サーバ IP アドレス — 191.162.2.8

図 10-2　基本的なバックエンド暗号化



クライアント側の設定

次に、SSL プロキシ サービスの設定例を示します。

```
ssl-proxy(config)# ssl-proxy service S1
ssl-proxy(config-ssl-proxy)# virtual ipaddr 10.1.0.21 protocol tcp port 443 secondary
ssl-proxy(config-ssl-proxy)# server ipaddr 10.2.0.100 protocol TCP port 80
ssl-proxy(config-ssl-proxy)# inservice
```

次に、CSM-S 仮想サーバの設定例を示します。

```
Cat6k-2(config-module-csm)# serverfarm SSLfarm
Cat6k-2(config-slb-sfarm)# real 10.1.0.21 local
Cat6k-2(config-slb-real)# inservice

Cat6k-2(config-module-csm)# vserver VS1
Cat6k-2(config-slb-vserver)# virtual 10.1.0.21 tcp https
Cat6k-2(config-slb-vserver)# serverfarm SSLfarm
Cat6k-2(config-slb-vserver)# inservice
```

SSL ロードバランシングは、混在モードの CSM-S および SSL Services Module（SSLSM）上で実行できます。

CSM-S は SSL-ID 固定（sticky）機能を使用して、同じ SSL サービス モジュールに SSL 接続を固定します。CSM-S は、SSL-ID を調べるためにクライアント側の TCP 接続を終端させなければなりません。CSM-S はさらに、ロードバランシングが決定された時点で、SSLSM への TCP 接続を開始しなければなりません。

トラフィック フローには、仮想サーバで受信したあらゆるトラフィックを SSLSM 上で終端する TCP で SSLSM に渡す CSM-S が含まれます。SSL 固定（sticky）機能をイネーブルにすると、CSM-S と SSLSM 間の接続が完全な TCP 接続になります。

次に、混在モードの SSL ロードバランシングを設定する例を示します。

```
Cat6k-2(config-module-csm)# sticky 10 ssl timeout 60
Cat6k-2(config-module-csm)# serverfarm SSLfarm
Cat6k-2(config-slb-sfarm)# real 10.1.0.21 local
Cat6k-2(config-slb-sfarm)# inservice
Cat6k-2(config-slb-sfarm)# real 10.2.0.21
Cat6k-2(config-slb-sfarm)# inservice
Cat6k-2(config-module-csm)# vserver VS1
Cat6k-2(config-slb-vserver)# virtual 10.1.0.21 tcp https
Cat6k-2(config-slb-vserver)# sticky 60 group 10
Cat6k-2(config-slb-vserver)# serverfarm SSLfarm
Cat6k-2(config-slb-vserver)# persistent rebalance
Cat6k-2(config-slb-vserver)# inservice
```

CSM-S がクライアント側の TCP 接続を終端させなければならないときに、SSLSM でトラフィックを転送するコンフィギュレーションを内部生成の形で作成する必要があります。サーバ ファーム *SSLfarm* のローカルな各実サーバの同じ IP アドレスまたはポートを指定して、仮想サーバを作成する必要があります。この仮想サーバは内部で、その仮想サーバ宛てのあらゆるトラフィックを SSLSM に転送するように設定されます。

内部生成の形でコンフィギュレーションを作成しなければならないのは、ローカル実サーバの IP アドレスと CSM-S 仮想サーバのアドレスを一致させなければならないからです。CSM がこのローカル実サーバへの接続を開始すると、CSM-S が SYN（同期）フレームを送受信します。CSM-S が SYN を受信し、宛先 IP アドレスまたはポートが仮想サーバの VS1 と同じだった場合、CSM-S はさらに具体的な仮想サーバが追加されないかぎり、VS1 と一致したとみなします。

サーバ側の設定

SSLSM がバックエンド サーバとして CSM-S を使用する場合、レイヤ 4 およびレイヤ 7 のロードバランシングには、仮想サーバの標準設定を使用します。

次に、SSLSM からのトラフィックだけを受信するように、この仮想サーバに制限を加える例を示します。

```
Cat6k-2(config-module-csm)# serverfarm SLBdefaultfarm
Cat6k-2(config-slb-sfarm)# real 10.2.0.20
Cat6k-2(config-slb-sfarm)# inservice

Cat6k-2(config-module-csm)# vserver VS2
Cat6k-2(config-slb-vserver)# virtual 10.2.0.100 tcp www
Cat6k-2(config-slb-vserver)# serverfarm SLBdefaultfarm
Cat6k-2(config-slb-vserver)# vlan local
Cat6k-2(config-slb-vserver)# inservice
```

次に、バックエンド サーバとして実サーバを設定する例を示します。

```
Cat6k-2(config-module-csm)# serverfarm SSLpredictorforward
Cat6k-2(config-slb-sfarm)# predictor forward

Cat6k-2(config-module-csm)# vserver VS3
Cat6k-2(config-slb-vserver)# virtual 0.0.0.0 0.0.0.0 tcp www
Cat6k-2(config-slb-vserver)# serverfarm SSLpredictorforward
Cat6k-2(config-slb-vserver)# inservice
```

バックエンド サーバとしての CSM-S の設定

仮想サーバおよびサーバ ファームの設定により、実サーバをバックエンド サーバとして使用できます。CSM-S をバックエンド サーバとして使用するには、「クライアント側の設定」（p.10-31）で説明した設定を使用し、さらに SSL ドータカードを設定します。

次に、レイヤ 7 のロードバランシングに対応する CSM-S 仮想サーバの設定を示します。

```
Cat6k-2(config-module-csm)# serverfarm SLBdefaultfarm
Cat6k-2(config-slb-sfarm)# real 10.2.0.20
Cat6k-2(config-slb-real)# inservice

Cat6k-2(config-module-csm)# serverfarm SLBjpgfarm
Cat6k-2(config-slb-sfarm)# real 10.2.0.21

Cat6k-2(config-module-csm)# map JPG url
Cat6k-2(config-slb-map-cookie)# match protocol http url *jpg*

Cat6k-2(config-module-csm)# policy SLBjpg
Cat6k-2(config-slb-policy)# url-map JPG
Cat6k-2(config-slb-policy)#serverfarm SLBjpgfarm

Cat6k-2(config-module-csm)# vserver VS2
Cat6k-2(config-slb-vserver)# virtual 10.2.0.100 tcp www
Cat6k-2(config-slb-vserver)# serverfarm SLBdefaultfarm
Cat6k-2(config-slb-vserver)# slb-policy SLBjpg
Cat6k-2(config-slb-vserver)# inservice
```

次に、レイヤ 4 のロードバランシングに対応する CSM-S 仮想サーバの設定を示します。

```
Cat6k-2(config-module-csm)# serverfarm SLBdefaultfarm
Cat6k-2(config-slb-sfarm)# real 10.2.0.20
Cat6k-2(config-slb-real)# inservice

Cat6k-2(config-module-csm)# vserver VS2
Cat6k-2(config-slb-vserver)# virtual 10.2.0.100 tcp www
Cat6k-2(config-slb-vserver)# serverfarm SLBdefaultfarm
Cat6k-2(config-slb-vserver)# vlan local
Cat6k-2(config-slb-vserver)# inservice
```

バックエンド サーバとしての実サーバの設定

実サーバをバックエンド サーバとした、サーバ側コンフィギュレーションのトラフィック フローは、クライアント側のコンフィギュレーションと同様です。実サーバをバックエンド サーバとして使用するには、「クライアント側の設定」（p.10-31）で説明した設定を使用し、さらに SSLSM を設定します。

SSLSM プロキシ サービスに関して、新しい設定は不要です。次に、設定を内部で開始し、ユーザにわからないようにする例を示します。

```
ssl-proxy(config)# ssl-proxy service S1
ssl-proxy(config-ssl-proxy)# virtual ipaddr 10.1.0.21 protocol tcp port 443 secondary
ssl-proxy(config-ssl-proxy)# server ipaddr 10.2.0.20 protocol TCP port 80
ssl-proxy(config-ssl-proxy)# inservice
```

次に、CSM-S 仮想サーバの設定例を示します。

```
Cat6k-2(config-module-csm)# serverfarm SSLreals

Cat6k-2(config-slb-sfarm)# real 10.2.0.20
Cat6k-2(config-slb-sfarm)# inservice

Cat6k-2(config-module-csm)# serverfarm SSLpredictorforward
Cat6k-2(config-slb-sfarm)# predictor forward

Cat6k-2(config-module-csm)# vserver VS3
Cat6k-2(config-slb-vserver)# virtual 0.0.0.0 0.0.0.0 tcp www
Cat6k-2(config-slb-vserver)# serverfarm SSLpredictorforward
Cat6k-2(config-slb-vserver)# inservice
```