# NSH トラフィックステアリング



# マニュアルの変更履歴

（注） リリース 21.24 よりも前に導入された機能については、詳細な改訂履歴は示していません。

| 改訂の詳細 | リリース |
|---|---|
| このリリースでは、トラフィックステアリングおよび L2 up-appliance-group BFD 設定の後処理ルール条件の照合がサポートされています。 | 21.23.22 |
| このリリースでは、トラフィックステアリングの後処理ルール条件の照合と、インターフェイス名を使用して実行できる L2 up-appliance-group BFD 設定のサポートが追加されています。 | 21.27 |

| 改訂の詳細 | リリース |
| --- | --- |
| 最初の導入。 | 21.24 より前 |

# 機能説明

3GPP EPC アーキテクチャにより、Gi インターフェイス上の各種サービス機能間でデータトラフィックをステアリングできます。トラフィック ステアリング アーキテクチャは、ネットワーク サービス ヘッダー（NSH）サービス チェーン プロトコルに基づいています。EPC ゲートウェイは、NSHをサポートするアプライアンスを含む複数のサービスチェーン全体でトラフィックを誘導するため、トラフィックステアリングを実行する必要があります。

NSH トラフィックステアリングには、次の 2 つのモードがあります。

- スタンドアロン モード
- サンドイッチモード

この機能により、お客様の要件に基づいて、トラフィックの課金とステアリングを互いに独立させることができます。お客様は、最小限の構成拡張によって、トラフィックをステアリングするためのさまざまなトラフィックカテゴリを既存のユースケースシナリオ内に加えることができます。

## トラフィックステアリングの後処理ルール条件の照合

単純なトラフィック分類は、複数のルールベースにまたがる膨大な数の課金ルールにより、トラフィックステアリングの操作および設定プロセスを簡素化するのに役立ちます。

- サービス スキーム フレームワークのトリガー条件では、後処理 ruledef 名の照合がサポートされています。
- トラフィックの後処理ルールとして設定された L3/L4 ruledef がトラフィックステアリングされます。
- トリガーアクションは、トラフィックステアリングの後処理ルールの照合に関するトリガー条件をサポートしています。
- トリガー条件の後処理 ruledef 名は、PFD プッシュと RCM でサポートされています。

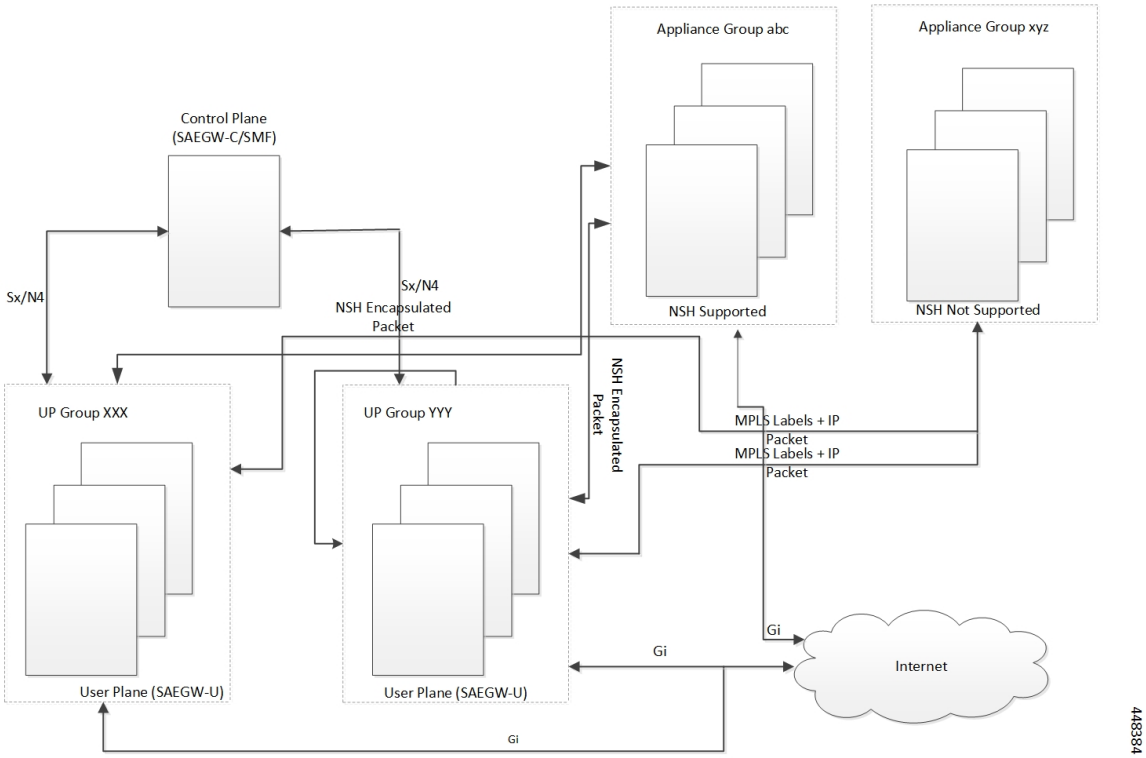
## インターフェイス名を使用した UP アプライアンスグループでの BFD インスタンス ID の設定

トラフィックステアリングの場合、 **up-appliance-group** 内の Bidirectional Forwarding Detection（BFD）インスタンス ID の設定は、IP 設定とともにインターフェイス名を使用して有効になります。

# アーキテクチャ：スタンドアロンモード

次の図は、NSHアプライアンス向けのCUPSベースゲートウェイのアーキテクチャセットアップを示しています。

**図 *1 : NSH* トラフィック ステアリング アーキテクチャ：スタンドアロンモード**




この機能は、NSH がサポートするアプライアンスの Service Function Chaining をサポートします。ゲートウェイは、各アプライアンスのインスタンスまたはグループに基づいて、トラフィックをステアリングするための適切なステアリング方式またはカプセル化方式を選択するように設定されます。

**表 *1 :* 通話フロー**

| ステップ | 説明 |
|---|---|
| 1. | SAEGW-U で受信した UL パケットは、適切な SFC に関連付けられた設定済みポリシーに基づいて分類されます。 |

| ステップ | 説明 |
|---|---|
| 2. | Saegw は、SFP のスティッキ性（MSISDN スティッキ性）またはサービスと負荷の可用性に基づいて SFP の選択を実行します。UL トラフィックは、選択した SFP で NSH（IP-UDP）カプセル化されてステアリングされ、必要に応じてコンテキストヘッダーが入力されます。 |
| 3. | NSH アプライアンスは、NSH パケットを受信すると、IP パケット（場合によってはコンテキストヘッダーも）を処理し、Gi インターフェイスを介してパケットを送信します。 |
| 4. | 接続先サーバーは、Gi インターフェイスから SAEGW-U に DL パケットを送信します。DL トラフィックは、選択した SFP で NSH（IP-UDP）カプセル化されてステアリングされ、必要に応じてコンテキストヘッダーが入力されます。 |
| 5. | NSH アプライアンスは、NSH パケットを受信すると、IP パケット（場合によってはコンテキストヘッダーも）を処理し、パケットを SAEGW-U にヘアピンします。 |
| 6. | NSH パケットを受信した SAEGW-U：<br>• 受信したペイロードのカプセル化を解除します。<br>• IP パケット（場合によってはコンテキストヘッダーも）を処理し、Gn インターフェイスを介してパケットを UE に送信します。 |

# コンポーネント

トラフィック ステアリング アーキテクチャは、次の主要コンポーネントで構成されています。

### コントロールプレーン（SAEGW-C）

CP はサブスクライバのトラフィックのステアリング方法に関する情報を UP に送信します。UP はサブスクライバに対して定義されたポリシーに基づいて、サブスクライバ データ トラフィックのすべてまたは一部のみをステアリングします。さまざまなタイプのサブスクライバ トラフィックをさまざまなサービス機能チェーンに誘導できます。

CP はローカルに設定されたポリシーに基づいて、PCRF から Ts-subscription-scheme AVP を受信した後、サブスクライバのサービスチェーン名を選択します。

### ユーザープレーン（SAEGW-U）

UP は CP から受信したポリシーに基づいて、サブスクライバ データ トラフィックを 1 つ以上のサービス機能チェーンに誘導します。

UP は、次の機能も実行します。

- 特定のサービス機能チェーン（SFC）のサービス機能パス（SFP）を選択します。
- アプライアンスにトラフィックを転送しながら、サブスクライバのスティッキ性を維持します。
- ノードやアプライアンスに障害が発生した場合は、サブスクライバ データ トラフィックを再選択し、新しいノードに誘導します。
- SFP の**インサービス**および **アウトオブサービス**のステータスを管理します。
- SFC 内でサービスを提供できる SFP の数に応じて、SFC ステータスを管理します。

### NSH

NSH アプライアンスの正常性をモニタリングするために、各 SAEGW-U/UPF はアプライアンスの負荷と有用性統計のモニタリングを担当します。

- OAM NSH パケットメカニズムを使用して、アプライアンスのステータスをモモニターします。
- 設定のモニタリング頻度は 1 ～ 20 秒で、デフォルトの間隔は 1 秒です。
- OAM 要求がタイムアウトした場合は、再試行します。タイムアウトと再試行の値については、タイムアウトは 1 ～ 5 秒（デフォルトは 3 秒）、再試行は 1 ～ 3 回（デフォルトは 2 回）の範囲で値を設定できます。
- アプライアンスの有用性ステータスに加えて、アプライアンスの現在の負荷が監視されます。SF のさまざまなインスタンス間で最適なロードバランシングを維持するために、現在の負荷をモニターします。この負荷ステータスは、NSH の OAM 応答パケットを介して返されます。

# 制限事項

NSH トラフィックステアリングには、次の制限があります。

- NSH アプライアンスで、インターフェイスのフラグメンテーションが発生しないようにする必要があります。NSH アプライアンス インターフェイスへの MTU を Gn/Gi インターフェイスよりも大きくします。

- HTTP パイプライン化セッション、ミッドフロー HTTP 部分パケット、および TCP アウトオブオーダーパケットの場合、L7 条件で SFP 再評価が要求されると、NSH アプライアンスに到達しません。
- メイン設定から SFP ID 設定を削除しても、show configuration では依然として SFP ID が表示されます。commit CLI を使用して SFP ID を VPP にコミットすると、SFP ID は削除されます。
- トラフィックステアリング統計は、トラフィックステアリングの候補となるパケットを示します。トラフィックステアリング統計では、クォータの枯渇によってドロップされたパケットもカウントされますが、これらは依然としてトラフィックステアリングの候補です。
- NSH SRC/バインド IP アドレスまたはアプライアンス IP アドレスの変更が NSH アプライアンスのインスタンス設定で必要になった場合は、インスタンスを削除してから、それに関連付けられている SFP を削除し、SFP と新しいインスタンスを変更した IP アドレスとともに配置する必要があります。後でコミットを実行します。
- ノード障害が発生して、連続データが受信されると、ステアリング統計に不一致が生じる可能性があります。ダウンしている SFP でステアリングされたデータは、統計に反映されません。
- マルチ PDN コールの場合、NSH インスタンスのスティッキ性は各サブスクライバセッションに制限されます。
- ICSR や SFP の削除などの設定変更が原因で SAEGW-U の状態が変更された場合、この時間枠でアプライアンスからヘアピンバックされているパケットがドロップされる可能性があります。それ以降のすべての着信パケットは、通常どおりに処理されます。
- フローの最初のパケットが DL パケット（セッションリカバリ）の場合、最初のパケットだけがドロップされますが、再送信されたパケットと後続のすべてのパケットは通常どおりに送信されます。
- NSH 形式のタグが変更された場合、タグタイプ stream-fp-md エンコード、reverse-stream-fp-md 、secondary-srv-path-hdr、および rate-group は、既存のフローではなく、新しいフローに対して有効になります。NSH 形式の残りのタグの変更は新しいセッションに適用されますが、既存のセッションのトラフィックは古い形式のタグで続行されます。このようなケース、特にタグの変更や削除の場合、アプライアンスが NSH パケットで受信したタグ値と一致せず、あいまいな動作につながる可能性があります。そのため、NSH 形式のタイプの変更は慎重に行ってください。
- サーバーによって開始された TCP フローは、トラフィックステアリングでは考慮されません。
- NSH トラフィックをキャプチャするための Monsub サポートは現在使用できません。
- アプライアンスレベルの制限（トラフィックタイプなど）に対処するためには、サービススキームのポリシー選択の設定で、そのようなアプライアンスを含むサービスチェーンの選択対象からそのようなトラフィックを柔軟に除外できます。

- N:M 構成の場合、サービススキームの設定（トリガーアクション、トリガー条件、サービススキーム、サブスクライバクラス、サブスクライバベース）は、UP の Day-0 設定で行う必要があります。UP の一般的な設定では、サービススキームが設定されていると、競合状態になり、ユーザープレーンのセッションマネージャでサービススキームが断続的に設定されなくなり、トラフィックステアリング機能で障害が発生します。
- L2 ステアリングの OAM 統計は部分的にサポートされています。
- HTTP 連結パケットの場合のパケットは、パケット内の最後の HTTP GET で一致したポリシーに基づいてステアリングされたトラフィックを指します。
- アプライアンスがダウンした場合、次のアップリンクパケットがフローで回復したときに、フローを再評価するためにオンロードされます。新しい SFP の選択が行われ、トラフィックが新しいアプライアンスに誘導されます。

# 機能の仕組み：スタンドアロンモード

## パケットフロー

この項では、NSH トラフィック ステアリング アーキテクチャのパケットフローについて説明します。

### アップリンクパケット数

**図 *2:* アップリンクパケットフロー**

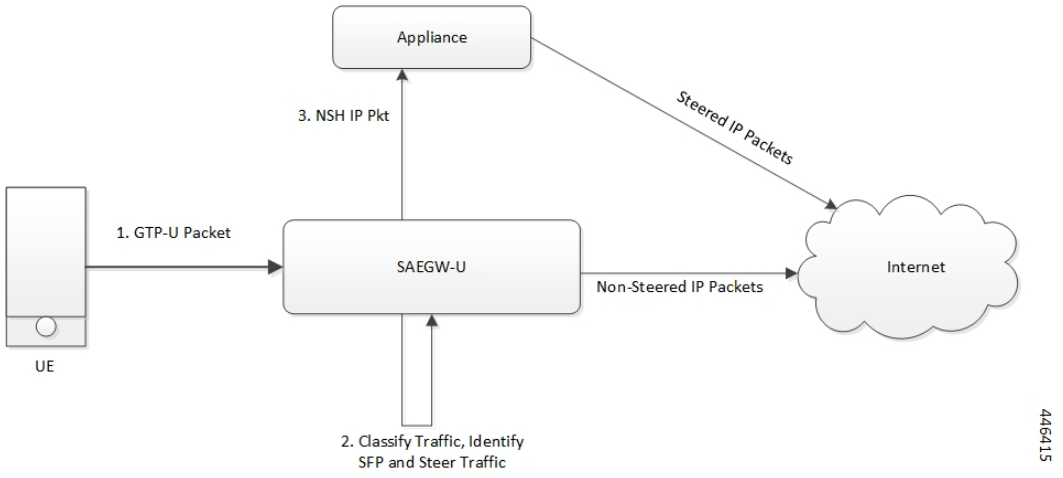


表 *2:* アップリンクパケットフローの説明

| 手順 | 説明 |
|---|---|
| 1 | UE が、サブスクライバのデータパケットを SAEGW-U に送信します。 |
| 2 | SAEGW-U が、サブスクライバポリシーに基づいてサブスクライバのデータトラフィックを分類し、SFC を識別して適宜 SFP を選択します。 |
| 3 | SAEGW-U が、NSH RFC に従って NSH カプセル化を使用してアップリンク（UL）パケットをステアリングし、NSH アプライアンスに送信します。<br>SAEGW-U が、ステアリングされていない IP パケットをサーバーに送信します。 |
| 4 | アップリンクパケットを受信した NSH 対応アプライアンスが、特定の基準に基づいてパケットをサーバーに転送するか決定します。 |

## ダウンリンクパケット数

図 *3:* ダウンリンクパケットフロー

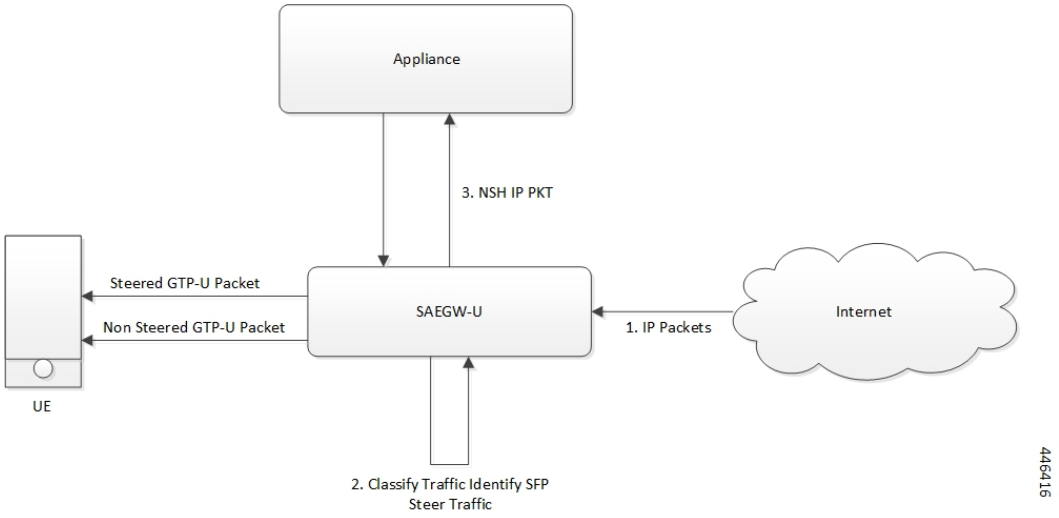



表 *3:* ダウンリンクパケットフローの説明

| 手順 | 説明 |
|---|---|
| 1 | SAEGW-U が、サーバーからダウンリンク（DL）パケットを受信します。 |
| 2 | SAEGW-U が SFP を選択します。 |

| 手順 | 説明 |
|---|---|
| 3 | SAEGW-U が、メタデータを NSH コンテキストヘッダーとして追加し、NSH 対応アプライアンスに転送します。 |
| 4 | NSH 対応アプライアンスが、SAEGW-U によって送信されたいくつかのメタデータタグを使用してパケットを SAEGW-U に送り返します。 |
| 5 | パケットを受信した SAEGW-U が、サブスクライバ課金ポリシーに基づいてサブスクライバのデータトラフィックを分類します。 |
| 6 | SAEGW-U がデータパケットをサブスクライバに送信します。 |

# NSH トラフィックステアリング要件

トラフィック ステアリング ソリューションにおける NSH アプライアンスの統合の動作は次のとおりです。

- SAEGW-U は NSH アプライアンスセッションのスティック性を維持し、サブスクライバセッションのすべてのフローが同じアプライアンスインスタンスを選択するようにします。
- すべてのアプライアンスインスタンスの負荷容量を定義するために設定できるオプション（50%、100% など）があります。NSH アプライアンスによる負荷ステータスがこのしきい値を超えた場合、既存のサブスクライバだけがそれまでのインスタンスを続行できます。このインスタンスは、負荷ステータスがしきい値を下回るまで、新しいサブスクライバに割り当てられません。
- NSH アプライアンスが DEAD 状態と検出された場合、このアプライアンスインスタンスに関与する SFP 上のすべてのトラフィックが再分類され、トラフィックは別のアプライアンスインスタンスに移動します。このようなアプライアンスは ALIVE 状態に戻っても、新しいサブスクライバの選択には使用できません。
- トラフィックステアリングは、セッション中に有効または無効にできます。セッション中にトラフィックステアリングを有効にすると、新しいフローはトラフィックステアリングの対象になります。古いフローは、トラフィックステアリングなしで続行されます。
- SR/ICSR 後のトラフィック ステアリング セッションのスティッキ性に対する SR/ICSR サポートは維持されます 。
- マルチアプライアンス SFP の場合、設定には次の 2 つの形式があります。
  - アプライアンスがトラフィックの開始（TWH パケットなど）を確認する必要がある場合は、すべてのアプライアンスに関与する SFP が選択されます。設定ポリシーに従って分類が行われると、トラフィックは不適格なアプライアンスから脱落する可能性があります。
  - アプライアンスが中間フローに関与する場合、特定のアプライアンスがさらにトラフィック分類の対象になると、アプライアンスが関与するように設定されます。

- トラフィックステアリング統計は、トラフィックステアリングの候補となるパケットを示します。トラフィックステアリング統計では、クォータの枯渇によってドロップされたパケットもカウントされますが、これらはトラフィックステアリングの候補です。
- ノード障害が発生して、連続データが受信されると、ステアリング統計に不一致が生じる可能性があります。ダウンしている SFP でステアリングされたデータは、統計に反映されません。
- NSH アプライアンスインスタンスの設定で NSH リモート IP アドレスまたは SRC バインド IP を変更する場合は、次の手順を実行します。
  - 次に、インスタンスを削除します。
  - 次に、関連付けられている SFP を削除します。
  - 変更後の IP アドレスで SFP と新しいインスタンスを配置します。
  - 後でコミットを実行します。

この機能では、次のトラフィック ステアリング システムの制限値がサポートされています。

| トラフィック ステアリング オブジェクト | 上限 |
|---|---|
| アプライアンスグループの総数 | 16 |
| アプライアンスグループごとのインスタンスの総数 | 256 |
| SFC の総数 | 16 |
| SFP の総数 | 64000 |

### デフォルトのサービスチェーン

オペレータは、トラフィックステアリングが有効になっているサブスクライバのすべてのトラフィックが特定のアプライアンスを通過する必要があるといったユースケースを扱うことがあります。このような要件に対応すると同時に、それを実現するための簡単な設定メカニズムを提供するために、デフォルトのサービスチェーンの概念が導入されました。たとえば、サブスクライバが 2 つのアプライアンス（APP1 と APP2）を持つサブスクライバとコミュニケーションを取り、APP2 ですべてのトラフィックを表示する必要がある場合、APP2 を含むサービスチェーンがデフォルトのサービスチェーンとして設定されます。

したがって、トラフィックステアリングが有効になっているサブスクライバの場合、次のような状況下では、特定のトラフィックについてはサービスチェーン APP1+APP2 を使用できない可能性があります。

- APP1+APP2 サービスチェーンを選択しようとしている特定のフローに適切なポリシーが設定されていない。

- APP1+APP2 サービスチェーンが選択されたが、APP1 インスタンスが最小インスタンスしきい値を下回っている。このような場合、APP1+APP2 サービスチェーンは使用できません。
- APP1+APP2 サービスチェーンが選択されたが、SFP を選択できなかった。

このようにサービスチェーンが使用できない場合、フローは設定済みのデフォルトサービスチェーンにフォールバックし、フローに対する APP2 サービス処理が保証されます。

デフォルトのサービスチェーンが設定されていない場合、トラフィックはステアリングされずに送信されます。

# SFP の選択

SFP の選択は、次のいずれかに基づきます。

- MSISDN スティッキ性（事前設定済み）
- 負荷の可用性

### MSISDN スティッキ性

MSISDN スティッキ性は MS-ISDN に依存し、対応するノードを提供します。ノードが使用可能で、SFP の一部である場合は、その SFP がデータ（UL/DL）用に選択されます。現在、MSISDN スティッキ性は L2 ノードでのみ使用可能であり、L2 ノードのみ、または L2 と NSH が混在するサービスチェーンが存在する可能性があります。サービスチェーンの SFP はすべて、同じノードタイプのセットになっています。このタイプには、L2、L2+NSH、または NSH（のみ）があります。

サブスクライバのスティッキ性（L2 と NSH の両方）は、そのノードが使用可能になるまでサービスチェーン全体でサブスクライバに対して維持され、ノードがダウンするか設定から削除されると、サブスクライバは（SFP 選択に基づいて）別の SFP に移動できます。スティッキ性が失われた場合には、ログとトラップが生成されます。

### 負荷の可用性

負荷の可用性とは負荷のキャパシティであり、現在の負荷は SFP ごとに保持されます（SFP の一部であるすべてのインスタンスの最小値）。SFP は、負荷の可用性に基づいて、使用可能リスト、過負荷リスト、またはブロック対象リストに分類されます。ブロック対象リストはノードがダウンしている SFP を対象としているため、SFP の選択には、使用可能リストと過負荷リストのみが使用されます。使用可能リストの SFP は、古いコール/セッションおよび新しいコール/セッションの両方に使用できます。過負荷リスト（負荷の可用性＝0）は、スティッキ性（存在する場合）を維持する場合にのみ使用されます。つまり、古いコール/セッション専用です。SFP が選択されると、その SFP は負荷に応じて、またスティッキ性の維持のために、過負荷リストに移動する場合があります。古いコール/セッションには同じ SFP が使用され、新しいコールには SFP 選択の使用可能リストにある残りの SFP が使用されます。

# インライン機能とのインターワーキング

次のインライン機能とのインターワーキングのサポートは、既存の実装の範囲には含まれていません。

- IPv4/v6 再アドレス指定
- NAT44 および NAT64
- Next Hop Forwarding
- L2 マーキング

NSH コンテキストヘッダーの評価グループのエンコーディングは、次の予想される動作に合わせてサポートされています。

- エンコードされた評価グループ値は、各パケットが一致するルールに対応するため、単一のフローのパケットでは、異なる評価グループが設定されているか、または設定されていない異なるルール間をフローが移動すると、評価グループが変更されるか、またはエンコードされません。
- SAEGW により、評価グループフィールドに評価グループ値が入力されます（設定されている場合）。コンテンツ ID のみが設定されている場合、この値がフィールドに入力されます。パケットの一致ルールに関連付けられているルールがない場合、評価グループに対応する TLV フィールドは送信されません。
- SAE-GW で遅延ルールの照合が実行され、ルールが一致しない状態でパケットが送信される場合、パケットの評価グループ TLV はエンコードされません。

# L2 および NSH トラフィックステアリング機能の設定：スタンドアロンモード

ここでは、CP と UP の両方で L2 および NSH トラフィックステアリング CUPS 機能を設定するために使用できる CLI コマンドについて説明します。

### コントロールプレーンの設定

CP を設定するには、次の手順を実行します。

1. 次の CLI コマンドは、[active-charging service] で CP を設定するための設定例です。

```
configure
  active-charging service ACS
  policy-control services-framework

  trigger-action ta1
    up-service-chain sc_L3
  exit
  trigger-action ta2
    up-service-chain  L3
```

```
  exit

  trigger-condition tc1
    rule-name = rule1
    rule-name = rule2
    multi-line-or
  exit

  trigger-condition tc2
    any-match = TRUE
  exit

  service-scheme scheme1
    trigger rule-match-change
      priority 1 trigger-condition tc1 trigger-action ta1
    exit
    trigger subs-scheme-received
      priority 1 trigger-condition tc2 trigger-action ta2
    exit

    subs-class class1
      subs-scheme = s1
    exit
    subscriber-base base1
      priority 1 subs-class class1 bind service-scheme scheme1
    exit
end
```

注：

- **subs-scheme**：この名前は、Gx インターフェイスを介して PCRF から受信した subscription-scheme AVP 値と一致する必要があります。
- **up-service-chain SecNet**：この値は、UP で設定されている up-service-chain と一致する必要があります。
- **rule-name**：この値には、静的/事前定義/GoR/ダイナミックルールを使用できます。

2. トラフィックステアリング AVP は現在、Diameter ディクショナリ custom44 でサポートされています。Diameter ディクショナリにより、TS 関連の AVP が Gx インターフェイスを介して受信されるとき、および Sx メッセージで UP に送信されるときに、CP はこれらの AVP を適切に復号できます。

   CP でディクショナリを設定するための設定例を以下に示します。

```
configure
  context ISP1
    ims-auth-service IMSGx
    policy-control
    diameter dictionary dpca-custom44
  exit
end
```

GX を介して CCA-I/CCA-U/RAR で受信する TS 関連 AVP の値の例を以下に示します。

```
    [V] Services:
      [V] Service-Feature:
        [V] Service-Feature-Type: TS (4)
        [V] Service-Feature-Status: ENABLE (1)
        [V] Service-Feature-Rule-Install:
          [V] Service-Feature-Rule-Definition:
```

```
[V] Service-Feature-Rule-Status: ENABLE (1)
[V] Subscription-Scheme: scheme
[V] Profile-Name: Gold
```

### ユーザープレーンの設定

UP を設定するには、次の手順を同じ順序で実行します。

次の CLI コマンドは、L2 および NSH がサポートされているアプライアンスへのデータ送信に使用されるコンテキストに、インターフェイスを追加するための設定例です。

1. L2 および NSH がサポートされているアプライアンスへのデータ送信に使用されるコンテキストにインターフェイスを追加します。

   以下に設定例を示します。

```
configure
require tsmon
end
configure
   context ISP1-UP
   interface <ts_ingress>
   ip address <ip_address>
   ipv6 address <ipv6_address_secondary>
   exit
   end

configure
   context ISP2-UP
   interface <ts_egress>
   ip address <ip_address>
   ipv6 address <ipv6_address_secondary>
   exit
   end
```

2. 新たに追加されたインターフェイスを UP の物理ポートにバインドします。

   次に設定例を示します。

```
configure
   port ethernet 1/11
   vlan 1240
   no shutdown
   bind interface ts_ingress ISP1-UP
   exit
   exit
   port ethernet 1/12
   vlan 1240
   no shutdown
   bind interface ts_egress ISP2-UP
   exit
   exit
   end
```

3. TS 関連の設定を UP に追加します。

   次に設定例を示します。

```
config

ts-bind-ip IP_UP01 ipv4-address 209.165.200.225 ipv6-address 4001::106

  nsh
```

```
node-monitor ipv4-address 209.165.200.226 ipv6-address 4001::107 poll-interval 1
retry-count 2 load-report-threshold 5 (node-monitor  is mandatory for NSH appliances,
 default values are poll-interval=1, retry-count=2, load-report-threshold=5)
    up-nsh-format format1
      tag-value 250 imsi encode
      tag-value 66 msisdn encode
      tag-value 4 rating-group encode
      tag-value 1 stream-fp-md encode decode
      tag-value 2 reverse-stream-fp-md encode decode
      tag-value 76 subscriber-profile encode
      tag-value 3 secondary-srv-path-hdr encode
      tag-value 5 rat-type encode
      tag-value 51 mcc-mnc encode
      tag-value 255 apn encode
      tag-value 25 sgsn-address encode
    #exit
  #exit
  traffic-steering
    up-service-chain sc_L3
      sfp-id 9 direction uplink up-appliance-group L2 instance 1 up-appliance-group
 L3 instance 1
     sfp-id 10 direction downlink up-appliance-group L3 instance 1 up-appliance-group
 L2 instance 1
     sfp-id 11 direction uplink up-appliance-group L2 instance 2 up-appliance-group
 L3 instance 1
     sfp-id 12 direction downlink up-appliance-group L3 instance 1 up-appliance-group
 L2 instance 2
     sfp-id 13 direction uplink up-appliance-group L2 instance 1 up-appliance-group
 L3 instance 2
     sfp-id 14 direction downlink up-appliance-group L3 instance 2 up-appliance-group
 L2 instance 1
     sfp-id 15 direction uplink up-appliance-group L2 instance 2 up-appliance-group
 L3 instance 2
     sfp-id 16 direction downlink up-appliance-group L3 instance 2 up-appliance-group
 L2 instance 2
     sfp-id 17 direction uplink up-appliance-group L2 instance 3 up-appliance-group
 L3 instance 1
     sfp-id 18 direction downlink up-appliance-group L3 instance 1 up-appliance-group
 L2 instance 3
     sfp-id 19 direction uplink up-appliance-group L2 instance 4 up-appliance-group
 L3 instance 1
     sfp-id 20 direction downlink up-appliance-group L3 instance 1 up-appliance-group
 L2 instance 4
     sfp-id 21 direction uplink up-appliance-group L2 instance 3 up-appliance-group
 L3 instance 2
     sfp-id 22 direction downlink up-appliance-group L3 instance 2 up-appliance-group
 L2 instance 3
     sfp-id 23 direction uplink up-appliance-group L2 instance 4 up-appliance-group
 L3 instance 2
     sfp-id 24 direction downlink up-appliance-group L3 instance 2 up-appliance-group
 L2 instance 4
    #exit
    up-service-chain L3
      sfp-id 1 direction uplink up-appliance-group L3 instance 1
      sfp-id 2 direction downlink up-appliance-group L3 instance 1
      sfp-id 3 direction uplink up-appliance-group L3 instance 2
      sfp-id 4 direction downlink up-appliance-group L3 instance 2
    #exit
    up-appliance-group L3
      steering-type nsh-aware
      up-nsh-format format4
      min-active-instance 1
      instance 1 ip address 40.40.40.3
      instance 2 ip address 40.40.40.4
```

```
    #exit
    up-appliance-group L2
      steering-type l2-mpls-aware
      min-active-instance 1
      instance 1 ingress slot/port 1/13 vlan-id 2136 egress slot/port 1/12 vlan-id
2136  ingress-context ingress ip address 4101::1 egress-context egress ip address
4101::2
      instance 2 ingress slot/port 1/13 vlan-id 2137 egress slot/port 1/12 vlan-id
2137  ingress-context ingress ip address 4201::1 egress-context egress ip address
4201::2
      instance 3 ingress slot/port 1/13 vlan-id 2138 egress slot/port 1/12 vlan-id
2138  ingress-context ingress ip address 4301::1 egress-context egress ip address
4301::2
      instance 4 ingress slot/port 1/13 vlan-id 2139 egress slot/port 1/12 vlan-id
2139  ingress-context ingress ip address 4401::1 egress-context egress ip address
4401::2
    #exit
```

4. **show configuration** CLI コマンドを使用して前述の設定を確認します。次に、**commit** CLI コマンドを実行して設定を有効にします。

```
configure
  traffic-steering
    commit
end
```

### 設定時の注意事項

ここでは、この機能を適切に設定するために必要な次のガイドラインについて説明します。

- 前項で説明したのと同じ順序で、UPのTS関連の設定を行います。この方法により、トラフィックを L2 に誘導するために使用されるインターフェイスが設定で適切に適用されます。
- [up-appliance-group] のインスタンスを変更または削除する必要がある場合は、まず [up-service-chain] の関連するすべての sfp-id を削除する必要があります。
- コールの開始後に関連するインスタンスと sfp-id に対して前述の変更を行う必要がある場合は、問題を回避するため、sfp-id を削除して再設定します。
- up-appliance-groupインスタンスを設定する前に、インターフェイスに変更を適用します。インターフェイスへの変更が後になってから適用される場合は、まずup-service-chain設定を削除してから、up-appliance-group設定を削除します。インターフェイスの変更が完了したら、サービスチェーンとアプライアンスグループを再設定します。
- インターフェイスまたはsfpidを削除するために、UPサービスチェーンとアプライアンスグループ全体を削除することはできません。

# N:M トラフィックステアリング

N:M トラフィックステアリングの設定手順は次のとおりです。

1. すべてのアクティブ UP の RCM ホスト固有の設定で TS-bind IP を設定します。

2. RCM の共通設定で必要なアクティブな課金 ruledef、ルールベース設定、およびトラフィックステアリング設定（up-nsh 形式、up-appliance-group および up-service-chain、commit CLI）を設定し、コミットします。
3. 必要な ts-mon、RCM 設定、L3 サーバーモニタリング用のノードモニター CLI、L2 の BFD 関連インターフェイス設定、およびトラフィックステアリング用のサービススキーマ設定（トリガー条件、トリガーアクションなど）が設定されている Day-0 設定を使用して、アクティブおよびスタンバイ UP をリロードします。
4. RCM がすべての UP に設定をプッシュすることを確認します。すべてのサービスがすべての UP で稼働していることを確認します。
5. VPP fastpath テーブルに SST、SSMT、および SST テーブルが作成されていること、また、グローバルテーブルが正しく作成されていることを確認します。
6. up-service-chain の SFP ステータスを確認し、SFP が使用可能な状態であることを確認します。

### 設定

以下に設定例を示します。

- **Day-0 設定**：次の設定は Day-0 設定の一部です。
  - 前述の「設定」の項で説明したように、L3 アプライアンスをモニターするには ts-mon および Node-monitor CLI が必要です。各 UP には、L3 アプライアンスをモニターする独自の物理 IP があります。
  - L2 の BFD 関連のインターフェイス設定。VLAN 設定および IP インターフェイス関連の設定。
  - サービススキーマ設定（トリガー条件、サービススキームなど）。

（注） 最適化により、サービススキーマ設定を共通設定に移動する予定です。現在、サービススキームの設定を変更する必要がある場合は、すべての UP で手動で変更する必要があります。

UP の設定例

**L3 モニタリング**

```
config
require ts-mon
nsh
 node-monitor ipv4-address 209.165.200.227 poll-interval 5 retry-count 5
load-report-threshold 20
exit

    interface ISP1_TO_PDN
      ip address 209.165.200.227 255.255.255.224
```

```
    ipv6 address 4001::254/64 secondary
  #exit
```

（注） UP2 では、IP は 40.40.40.454 にできます。これはその UP に固有の物理 IP アドレスです。

**L2 モニタリング**：

```
config
  context ingress
    bfd-protocol
      bfd multihop-peer 209.165.200.228 interval 50 min_rx 50 multiplier 20
      bfd multihop-peer 209.165.200.229 interval 50 min_rx 50 multiplier 20
      bfd multihop-peer 209.165.200.230 interval 50 min_rx 50 multiplier 20
    #exit
    interface TS_SecNet_v4 loopback
      ip address 209.165.200.231 255.255.255.224

    #exit
    interface TS_SecNet_v4_1 loopback
      ip address 209.165.200.232 255.255.255.224

    #exit
    interface TS_SecNet_v4_2 loopback
      ip address 209.165.200.233 255.255.255.224

    #exit
    interface TS_Secnet_ingress
      ip address 209.165.200.234 255.255.255.224

    #exit
    interface TS_Secnet_ingress1
      ip address 209.165.200.235 255.255.255.224

    #exit
    interface TS_Secnet_ingress2
      ip address 209.165.200.236 255.255.255.224

    #exit

    ip route static multihop bfd bfd1 209.165.200.231 209.165.200.228

    ip route static multihop bfd bfd2 209.165.200.232 209.165.200.229

    ip route static multihop bfd bfd3 209.165.200.233 209.165.200.230

    ip route 209.165.200.228 255.255.255.224 209.165.200.237 TS_Secnet_ingress

    ip route 209.165.200.229 255.255.255.224 209.165.200.238 TS_Secnet_ingress1

    ip route 209.165.200.230 255.255.255.224 209.165.200.239 TS_Secnet_ingress2

  #exit
end
```

```
config
  context egress
    bfd-protocol
    bfd multihop-peer 209.165.200.231 interval 50 min_rx 50 multiplier 20
      bfd multihop-peer 209.165.200.232 interval 50 min_rx 50 multiplier 20
      bfd multihop-peer 209.165.200.233 interval 50 min_rx 50 multiplier 20
```

```
    #exit
    interface TS_SecNet_v4 loopback
      ip address 209.165.200.228 255.255.255.224
    #exit
    interface TS_SecNet_v4_1 loopback
      ip address 209.165.200.229 255.255.255.224
    #exit
    interface TS_SecNet_v4_2 loopback
      ip address 209.165.200.230 255.255.255.224
    #exit
    interface TS_Secnet_egress
      ip address 209.165.200.237 255.255.255.224
    #exit
    interface TS_Secnet_egress1
      ip address 209.165.200.238 255.255.255.224
    #exit
    interface TS_Secnet_egress2
      ip address 209.165.200.239 255.255.255.224
    #exit
    subscriber default
    exit
    aaa group default
    #exit
    ip route static multihop bfd bfd4 209.165.200.228 209.165.200.231
    ip route static multihop bfd bfd5 209.165.200.229 209.165.200.232
    ip route static multihop bfd bfd6 209.165.200.230 209.165.200.233
    ip route 209.165.200.231 255.255.255.224 209.165.200.234 TS_Secnet_egress
    ip route 209.165.200.232 255.255.255.224 209.165.200.235 TS_Secnet_egress1
    ip route 209.165.200.233 255.255.255.224 209.165.200.236 TS_Secnet_egress2
  #exit
end
```

すべてのインターフェイスをポートと VLAN にバインドするための 1 つのインターフェイス設定例。

```
port ethernet 1/11
    vlan 1608
      no shutdown
      bind interface TS_Secnet_ingress ingress
    #exit
    vlan 1609
      no shutdown
      bind interface TS_Secnet_ingress1 ingress
    #exit
    vlan 1610
      no shutdown
      bind interface TS_Secnet_ingress2 ingress
    #exit
  #exit
  port ethernet 1/13
    no shutdown
    vlan 1608
      no shutdown
      bind interface TS_Secnet_egress egress
    #exit
    vlan 1609
      no shutdown
      bind interface TS_Secnet_egress1 egress
    #exit
    vlan 1610
      no shutdown
      bind interface TS_Secnet_egress2 egress
    #exit
```

サービススキーマ設定：

```
trigger-action ta1
      up-service-chain sc_L3
    #exit
    trigger-action default
      up-service-chain default
    #exit
    trigger-condition tc1
      rule-name = udp
      rule-name = http-pkts
      rule-name = tcp
      rule-name = dynamic2
      multi-line-or all-lines
    #exit
    trigger-condition tc2
      rule-name = qci8
      rule-name = qci1
      multi-line-or all-lines
    #exit
   trigger-condition defualt
     any-match = TRUE
   #exit
    service-scheme scheme1
      trigger rule-match-change
        priority 1 trigger-condition tc1 trigger-action ta1
        priority 2 trigger-condition tc2 trigger-action ta1
      #exit
      trigger subs-scheme-received
        priority 1 trigger-condition default trigger-action default
      #exit
     #exit
    subs-class class1
      subs-scheme = gold
    #exit
    subscriber-base base1
      priority 1 subs-class class1 bind service-scheme scheme1
    #exit
```

- **ホスト固有の設定**：次の設定は、ホスト固有の設定の一部です。
  - 各アクティブ UP の TS-bind IP 設定は、RCM におけるホスト固有の設定の一部です。

```
svc-type upinterface
redundancy-group 1
host Active1
host 391 " context ISP1-UP"
host 436 " interface ISP1_TO_PDN_v6 loopback"
host 437 " ipv6 address 4000::106/128"
host 438 " #exit"
host 439 " interface ISP1_TO_PDN_v4 loopback"
host 440 " ip address 209.165.200.240 255.255.255.224"
host 441 " #exit"
host 471 "ts-bind-ip up1 ipv4-address 209.165.200.240 ipv6-address 4000::106"
host 472 " exit"
host Active2
host 600 " context ISP1-UP"
host 601 " interface ISP1_TO_PDN_v6 loopback"
host 602 " ipv6 address 4000::107/128"
host 603 " #exit"
host 604 " interface ISP1_TO_PDN_v4 loopback"
host 605 " ip address 209.165.200.241 255.255.255.224"
host 606 " #exit"
```

```
host 607 "ts-bind-ip up2 ipv4-address 209.165.200.241 ipv6-address 4000::107"
host 608 " exit"
```

（注） TS-bind IP はループバック IP アドレスで、その物理 IP アドレスは、Day-0 設定の一部です。

- **共通設定**：次の設定は、共通設定の一部です。
  - トラフィックステアリング設定（up-nsh format、up-appliance-group、および up-service-chain config）。

（注） 低いVLANで入力が設定されていると仮定すると、アップリンクデータフローの場合、パケットは入力 VLAN ID で SN に送信され、出力 VLAN ID で SN から受信されます。同様に、ダウンリンクデータフローの場合、パケットは出力 VLAN ID で SN に送信され、入力 VLAN ID で SN から受信されます。

```
nsh
    up-nsh-format L3-format
      tag-value 7 imsi encode
      tag-value 4 rating-group encode
      tag-value 1 stream-fp-md encode decode
      tag-value 2 reverse-stream-fp-md encode decode
      tag-value 76 subscriber-profile encode
      tag-value 3 secondary-srv-path-hdr encode
      tag-value 5 rat-type encode
      tag-value 51 mcc-mnc encode
      tag-value 255 apn encode
      tag-value 25 sgsn-address encode
    #exit

  #exit
  traffic-steering
  up-appliance-group L2
steering-type l2-mpls-aware
min-active-instance 1
instance 1 ingress slot/port 1/12 vlan-id 1608 egress slot/port 1/13 vlan-id 1608
ingress-context ingress ip address 209.165.200.231egress-context egress ip address
209.165.200.228 load-capacity 100
instance 2 ingress slot/port 1/12 vlan-id 1609 egress slot/port 1/13 vlan-id 1609
ingress-context ingress ip address 209.165.200.232egress-context egress ip address
209.165.200.229 load-capacity 80
instance 3 ingress slot/port 1/12 vlan-id 1610 egress slot/port 1/13 vlan-id 1610
ingress-context ingress ip address 209.165.200.233egress-context egress ip address
209.165.200.230 load-capacity 90
exit
    up-appliance-group L3_only
      steering-type nsh-aware
      up-nsh-format new
      min-active-instance 1
      instance 1 ip address 209.165.200.242 load-capacity 80
      instance 2 ip address 209.165.200.243 load-capacity 90
    #exit
```

```
    up-service-chain sc_L3
      sfp-id 1 direction uplink up-appliance-group L2 instance 1 up-appliance-group
 L3_only instance 2
      sfp-id 2 direction downlink up-appliance-group L3_only instance 2
up-appliance-group L2 instance 1
     sfp-id 10 direction uplink up-appliance-group L2 instance 2 up-appliance-group
 L3_only instance 2
      sfp-id 11 direction downlink up-appliance-group L3_only instance 2
up-appliance-group L2 instance 2
     sfp-id 12 direction uplink up-appliance-group L2 instance 3 up-appliance-group
 L3_only instance 2
      sfp-id 13 direction downlink up-appliance-group L3_only instance 2
up-appliance-group L2 instance 3
     sfp-id 14 direction uplink up-appliance-group L2 instance 1 up-appliance-group
 L3_only instance 1
      sfp-id 15 direction downlink up-appliance-group L3_only instance 1
up-appliance-group L2 instance 1
     sfp-id 16 direction uplink up-appliance-group L2 instance 2 up-appliance-group
 L3_only instance 1
      sfp-id 17 direction downlink up-appliance-group L3_only instance 1
up-appliance-group L2 instance 2
     sfp-id 18 direction uplink up-appliance-group L2 instance 3 up-appliance-group
 L3_only instance 1
      sfp-id 19 direction downlink up-appliance-group L3_only instance 1
up-appliance-group L2 instance 3
    #exit
 up-service-chain default
sfp-id 200 direction uplink up-appliance-group L3_only instance 1
sfp-id 201 direction downlink up-appliance-group L3_only instance 1
sfp-id 202 direction uplink up-appliance-group L3_only instance 2
sfp-id 203 direction downlink up-appliance-group L3_only instance 2
#exit
commit
exit
```

### 検証用の show CLI

ユーザープレーンと RCM の show CLI を次に示します。

- ユーザープレーン：**Show srp checkpoints stats/ Show srp checkpoints stats debug-info**

  ```
  laas-setup# show srp checkpoint statistics | grep UPLANE_TRAFFIC_STEERING_INFO
  ```

- RCM：**under rcm checkpoint manager**

  ```
  "numTSInfo": 0
  ```

# モニタリングと障害対応：スタンドアロンモード

ここでは、この機能のモニタリングおよび障害対応の方法を説明します。

### コントロールプレーンの show コマンド

ここでは、CP でこの機能をモニターするための show コマンドについて説明します。

**show active-charging sessions full all**

（注） *TS Subscription Scheme Name*：active-charging-service で設定されたサービススキームから適用する必要があるサブスクリプションスキームが表示されます。この active-charging-service は、Gx インターフェイスを介して PCRF から受信されます。

### ユーザープレーンの show コマンド

ここでは、UP でこの機能をモニターするための show コマンドについて説明します。

### 設定の show コマンド

ここでは、この機能の設定の確認に使用できる show コマンドについて説明します。

- **show user-plane-service traffic-steering up-service-chain all**
- **show user-plane-service traffic-steering up-service-chain name** *up-service-chain name*
- **show user-plane-service traffic-steering up-service-chain sfp-id** *sfp-id*

### データ統計の show コマンド

ここでは、この機能に関連するデータ統計の確認に使用できる show コマンドについて説明します。

- **show user-plane-service inline-services traffic-steering statistics up-service-chain all v**
- **show user-plane-service inline-services traffic-steering statistics up-service-chain all**
- **show user-plane-service inline-services traffic-steering statistics up-service-chain sfp-id** *sfp-id*
- **show user-plane-service inline-services traffic-steering statistics up-appliance-group all verbose**
- **show user-plane-service inline-services traffic-steering statistics up-appliance-group name** *appliance group name*
- **show user-plane-service inline-services traffic-steering statistics up-appliance-group name** *appliance group name* **instance** *appliance instance*

### TS のサービスチェーンと SFP 関連付けを確認する show コマンド：

ここでは、サービスチェーンと SFP 関連付けの確認に使用できる show コマンドについて説明します。

- **show subscriber user-plane-only flows**
- **show subscribers user-plane-only callid** *<call-id>* **flows**

### OAM 統計の show コマンド

ここでは、この機能に関連する OAM 統計の確認に使用できる show コマンドについて説明します。

- **show user-plane-service inline-services traffic-steering oam all**
- **show user-plane-service inline-services traffic-steering oam summary**

- **show user-plane-service inline-services traffic-steering oam l3-steering summary**
- **show user-plane-service inline-services traffic-steering oam l3-steering monitors** *<ip address>*
- **show user-plane-service inline-services traffic-steering oam l3-steering monitors all**
- **show user-plane-service inline-services traffic-steering oam l3-steering monitors up-appliance-group***<name>*
- **show user-plane-service inline-services traffic-steering oam l2-steering monitors all**
- **show user-plane-service inline-services traffic-steering oam l2-steering monitors up-appliance-group** *<name>*
- **show user-plane-service inline-services traffic-steering oam l2-steering summary**
- **clear user-plane-service traffic-steering oam statistics**
- **clear user-plane-service traffic-steering oam l3-steering statistics**

現在、BFD はセッション統計をクリアする API を提供していないため、次の traffic-steering OAM clear コマンドが拡張され、l2-steering 統計が追加されました。

- clear user-plane-service traffic-steering
  - OAM：OAM をクリアします。
  - statistics：ユーザー プレーン トラフィック ステアリング統計をクリアします。
- clear user-plane-service traffic-steering OAM
  - L3-steering：L3-steering OAM をクリアします。
  - statistics：OAM 統計をクリアします。

### show configuration コマンド

次の設定は、この機能の **show configuration** コマンドのサンプルスニペットです。

```
nsh
  up-nsh-format format4
    tag-value 250 imsi encode
    tag-value 66 msisdn encode
    tag-value 4 rating-group encode
    tag-value 1 stream-fp-md encode decode
    tag-value 2 reverse-stream-fp-md encode decode
    tag-value 76 subscriber-profile encode
    tag-value 3 secondary-srv-path-hdr encode
    tag-value 5 rat-type encode
    tag-value 51 mcc-mnc encode
    tag-value 255 apn encode
    tag-value 25 sgsn-address encode
  #exit
traffic-steering
  up-service-chain L3
    sfp-id 65535 direction uplink up-appliance-group L3 instance 1
    sfp-id 65536 direction downlink up-appliance-group L3 instance 2
    sfp-id 65537 direction downlink up-appliance-group L3 instance 1
```

```
      sfp-id 65538 direction uplink up-appliance-group L3 instance 2
    #exit
    up-service-chain sc_L3
      sfp-id 9 direction uplink up-appliance-group L2 instance 1 up-appliance-group L3
instance 1
      sfp-id 10 direction downlink up-appliance-group L3 instance 1 up-appliance-group
L2 instance 1
      sfp-id 11 direction uplink up-appliance-group L2 instance 2 up-appliance-group L3
 instance 1
      sfp-id 12 direction downlink up-appliance-group L3 instance 1 up-appliance-group
L2 instance 2
      sfp-id 13 direction uplink up-appliance-group L2 instance 1 up-appliance-group L3
 instance 2
      sfp-id 14 direction downlink up-appliance-group L3 instance 2 up-appliance-group
L2 instance 1
      sfp-id 15 direction uplink up-appliance-group L2 instance 2 up-appliance-group L3
 instance 2
      sfp-id 16 direction downlink up-appliance-group L3 instance 2 up-appliance-group
L2 instance 2
      sfp-id 17 direction uplink up-appliance-group L2 instance 3 up-appliance-group L3
 instance 1
      sfp-id 18 direction downlink up-appliance-group L3 instance 1 up-appliance-group
L2 instance 3
      sfp-id 19 direction uplink up-appliance-group L2 instance 4 up-appliance-group L3
 instance 1
      sfp-id 20 direction downlink up-appliance-group L3 instance 1 up-appliance-group
L2 instance 4
      sfp-id 21 direction uplink up-appliance-group L2 instance 3 up-appliance-group L3
 instance 2
      sfp-id 22 direction downlink up-appliance-group L3 instance 2 up-appliance-group
L2 instance 3
      sfp-id 23 direction uplink up-appliance-group L2 instance 4 up-appliance-group L3
 instance 2
      sfp-id 24 direction downlink up-appliance-group L3 instance 2 up-appliance-group
L2 instance 4
    #exit
    up-appliance-group L3
      steering-type nsh-aware
      up-nsh-format format4
      min-active-instance 1
      instance 1 ip address 209.165.200.225
      instance 2 ip address 4001::3
    #exit
    up-appliance-group L2
      steering-type l2-mpls-aware
      min-active-instance 1
      instance 1 ingress slot/port 1/13 vlan-id 2136 egress slot/port 1/12 vlan-id 2136
  ingress-context ingress ip address 4101::1 egress-context egress ip address 4101::2
load-capacity 100
      instance 2 ingress slot/port 1/13 vlan-id 2137 egress slot/port 1/12 vlan-id 2137
  ingress-context ingress ip address 4201::1 egress-context egress ip address 4201::2
load-capacity 60
      instance 3 ingress slot/port 1/13 vlan-id 2138 egress slot/port 1/12 vlan-id 2138
  ingress-context ingress ip address 4301::1 egress-context egress ip address 4301::2
load-capacity 20
      instance 4 ingress slot/port 1/13 vlan-id 2139 egress slot/port 1/12 vlan-id 2139
  ingress-context ingress ip address 4401::1 egress-context egress ip address 4401::2
load-capacity 100
    #exit
  #exit
  ts-bind-ip nshsrcip ipv4-address 209.165.200.226 ipv6-address 4001::106
  #exit
    context egress
    bfd-protocol
```

```
    bfd multihop-peer 4101::1 interval 50 min_rx 50 multiplier 20
    bfd multihop-peer 4201::1 interval 50 min_rx 50 multiplier 20
    bfd multihop-peer 4301::1 interval 50 min_rx 50 multiplier 20
    bfd multihop-peer 4401::1 interval 50 min_rx 50 multiplier 20
  #exit
  interface ts_egress1
    ipv6 address 4101::2/64
    ip mtu 1600
  #exit
  interface ts_egress2
    ipv6 address 4201::2/64
    ip mtu 1600
  #exit
  interface ts_egress3
    ipv6 address 4301::2/64
    ip mtu 1600
  #exit
  interface ts_egress4
    ipv6 address 4401::2/64
    ip mtu 1600
  #exit
  subscriber default
  exit
  aaa group default
  #exit
  gtpp group default
  #exit
  ipv6 route static multihop bfd bfd1 4101::2 4101::1
  ipv6 route static multihop bfd bfd2 4201::2 4201::1
  ipv6 route static multihop bfd bfd3 4301::2 4301::1
  ipv6 route static multihop bfd bfd4 4401::2 4401::1
  ip igmp profile default
  #exit
#exit
context ingress
  bfd-protocol
    bfd multihop-peer 4101::2 interval 50 min_rx 50 multiplier 20
    bfd multihop-peer 4201::2 interval 50 min_rx 50 multiplier 20
    bfd multihop-peer 4301::2 interval 50 min_rx 50 multiplier 20
    bfd multihop-peer 4401::2 interval 50 min_rx 50 multiplier 20
  #exit
  interface ts_ingress1
    ipv6 address 4101::1/64
    ip mtu 1600
  #exit
  interface ts_ingress2
    ipv6 address 4201::1/64
    ip mtu 1600
  #exit
  interface ts_ingress3
    ipv6 address 4301::1/64
    ip mtu 1600
  #exit
  interface ts_ingress4
    ipv6 address 4401::1/64
    ip mtu 1600
  #exit
  subscriber default
  exit
  aaa group default
  #exit
  gtpp group default
  #exit
  ipv6 route static multihop bfd bfd1 4101::1 4101::2
```

```
  ipv6 route static multihop bfd bfd2 4201::1 4201::2
  ipv6 route static multihop bfd bfd3 4301::1 4301::2
  ipv6 route static multihop bfd bfd4 4401::1 4401::2
  ip igmp profile default
  #exit
#exit
  context ISP1-UP
  ip access-list IPV4ACL
    redirect css service ACS any
    permit any
  #exit
  ipv6 access-list IPV6ACL
    redirect css service ACS any
    permit any
  interface TO-ISP12
    ipv6 address 4001::106/64
    ip address 209.165.200.226 255.255.255.224 secondary
    ip mtu 2000
  #exit
    port ethernet 1/12
  no shutdown
  vlan 2135
    no shutdown
    bind interface TO-ISP12 ISP1-UP
  #exit
  vlan 2136
    bind interface ts_egress1 egress
  #exit
  vlan 2137
    no shutdown
    bind interface ts_egress2 egress
  #exit
  vlan 2138
    no shutdown
    bind interface ts_egress3 egress
  #exit
  vlan 2139
    no shutdown
    bind interface ts_egress4 egress
  #exit
#exit
port ethernet 1/13
  no shutdown
  vlan 2137
    no shutdown
    bind interface ts_ingress2 ingress
  #exit
  vlan 2138
    no shutdown
    bind interface ts_ingress3 ingress
  #exit
  vlan 2139
    no shutdown
    bind interface ts_ingress4 ingress
  #exit
  vlan 2136
    no shutdown
    bind interface ts_ingress1 ingress
  #exit
#exit
```

### ユーザープレーンの 1:1 冗長性に関する show コマンド

**show srp checkpoint statistics | grep ts-sfp**

```
call-recovery-uplane-internal-audit-ts-sfp-failure: 0
```

### SFP の可用性に関する show コマンド

**show user-plane traffic-steering up-service-chain <all> <name> <sfp-id>**

# SNMP トラップ

この機能をサポートするために、次の SNMP トラップが追加されました。

- UPlaneTsMisConfig：アプライアンスグループに関連付けられている SFP がない場合。
- UPlaneTsNoSelectedSfp：SFP を選択できない場合。
- UPlaneTsServiceChainOrApplianceDown：サービスチェーンまたはアプリケーションノードが使用できなくなった場合。アプリケーショングループの最小インスタンスが使用できなくなると、サービスチェーンは使用できなくなります。
- UPlaneTsServiceChainOrApplianceUp：サービスチェーンまたはアプリケーションノードインスタンスが使用可能になったため、アプライアンスのノードステータスが更新されたとき。

# バルク統計

### UP サービスチェーンスキーマ

| 変数名 | データ型 | カウンタタイプ | 説明 |
|---|---|---|---|
| up-svc-chain-name | 文字列 | Info | UP サービスチェーンの名前 |
| up-svc-chain-status | Int32 | Info | UP サービスチェーンのステータス |
| up-svc-chain-load-status | Int32 | ゲージ | UP サービスチェーンの負荷ステータス |
| up-svc-chain-sfp-stickiness-miss-count | Int32 | Counter | UP サービスチェーンの SFP スティッキ性の欠落数 |
| up-svc-chain-sfp-not-selected-count | Int32 | Counter | SFP が選択されていない UP サービスチェーンの数 |
| up-svc-chain-associated-calls | Int32 | ゲージ | UP サービスチェーンの関連コール |

| 変数名 | データ型 | カウンタタイプ | 説明 |
|---|---|---|---|
| up-svc-chain-associated-flows | Int32 | ゲージ | UP サービスチェーンの関連フロー |
| up-svc-chain-total-uplink-pkts | Int64 | Counter | UP サービスチェーンの合計アップリンクパケット数 |
| up-svc-chain-total-uplink-bytes | Int64 | Counter | UP サービスチェーンの合計アップリンクバイト数 |
| up-svc-chain-total-downlink-pkts | Int64 | Counter | UP サービスチェーンの合計ダウンリンクパケット数 |
| up-svc-chain-total-downlink-bytes | Int64 | Counter | UP サービスチェーンの合計ダウンリンクバイト数 |

**Up-appliance-group Schema**

| 変数名 | データ型 | カウンタタイプ | 説明 |
|---|---|---|---|
| up-appl-group-name | 文字列 | Info | UP アプライアンスのグループ名 |
| up-appl-group-status | Int32 | Info | UP アプライアンスグループのステータス |
| up-appl-group-load-status | Int32 | ゲージ | UP アプライアンスグループの負荷ステータス |
| up-appl-group-node-down-count | Int32 | Counter | UP アプライアンスグループのノードダウン数 |
| up-appl-group-associated-sfps | Int32 | ゲージ | UP アプライアンスグループに関連付けられた SFP |
| up-appl-group-num-times-loaded-state | Int32 | Counter | UP アプライアンスグループのノードダウン状態の回数 |
| up-appl-group-total-uplink-pkts | Int64 | Counter | UP アプライアンスグループの合計アップリンクパケット数 |
| up-appl-group-total-uplink-bytes | Int64 | Counter | UP アプライアンスグループの合計アップリンクバイト数 |
| up-appl-group-total-downlink-pkts | Int64 | Counter | UP アプライアンスグループの合計ダウンリンクパケット数 |
| up-appl-group-total-downlink-bytes | Int64 | Counter | UP アプライアンスグループの合計ダウンリンクバイト数 |

次の CLI コマンドは、この機能のバルク統計情報の設定例です。

```
config
   bulkstats collection
   bulkstats mode
   file 1
   up-service-chain schema TS format "\nup-service-chain-name = %up-svc-chain-name%
   \nup-service-chain-status=%up-svc-chain-status%\nup-service-chain-load-status =
   %up-svc-chain-load-status%\nup-service-chain-associated-calls =
   %up-svc-chain-associated-calls%\nup-service-chain-associated-flows =
   %up-svc-chain-associated-flows%\nup-service-chain-total-uplink-pkts =
   %up-svc-chain-total-uplink-pkts%\nup-service-chain-total-uplink-bytes =
   %up-svc-chain-total-uplink-bytes%\nup-service-chain-total-downlink-pkts =
   %up-svc-chain-total-downlink-pkts%\nup-service-chain-total-total-downlink-bytes
   = %up-svc-chain-total-downlink-bytes%\n\n"
```

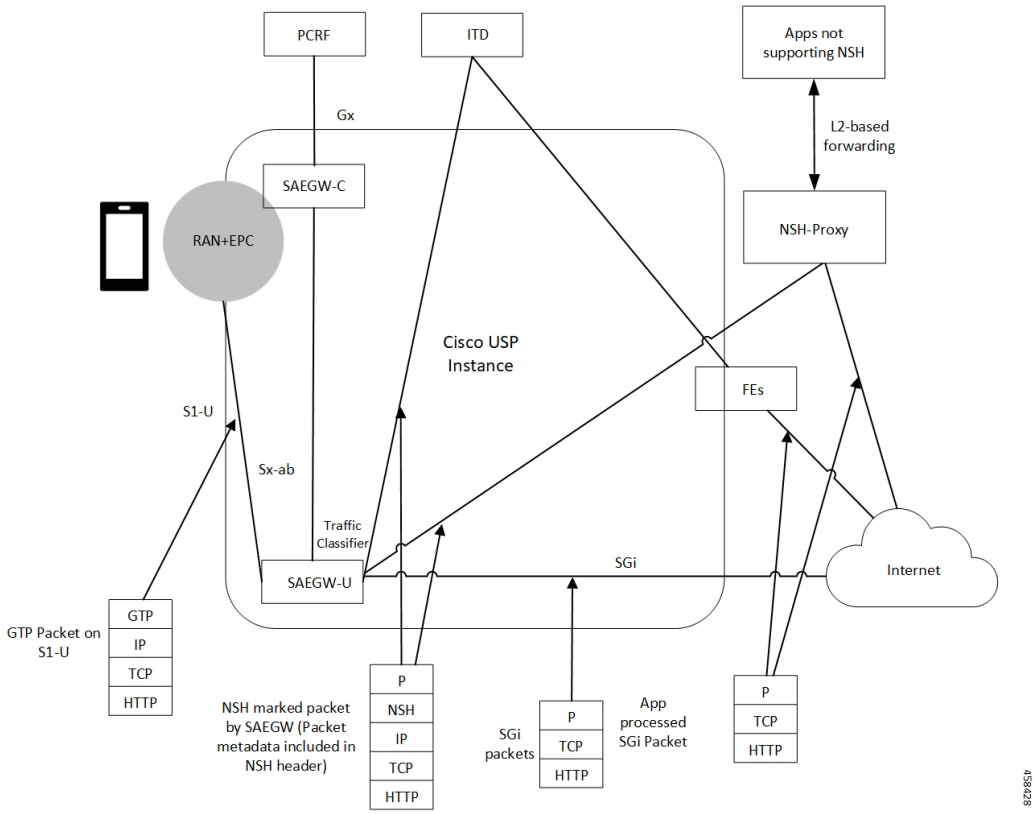
# 機能説明：サンドイッチモード

サンドイッチモードは、NSH ベースのトラフィックステアリング（TS）アプローチに対応して、サービス機能アプライアンスのフォワーディングエンジン（FE）ノードに必要なメタデータを提供します。

サンドイッチ モード ソリューションは、Cisco USP インスタンスで Cisco Nexus 9000 シリーズ NX-OS Intelligent Traffic Director（ITD）を活用します。ITD の詳細については、Cisco Nexus 9000 Series NX-OS Intelligent Traffic Director コンフィギュレーション ガイド [英語] を参照してください。

# アーキテクチャ：サンドイッチモード

次の図は、外部サービス機能アプライアンスとシスコの SAEGW-U（ユーザープレーン）の統合を示しています。




サンドイッチ モード ソリューションには、次の機能が含まれています。

- SAEGW-U は、アップリンク方向でのみ Gi パスを出る該当パケットに NSH ベースの該当メタデータを追加します。
- サンドイッチモードで実行中の ITD は、これらのパケットを（送信元 IP に基づいて）FE にロードバランシングできます。
- SAEGW-U は、FE に対して正常性チェックを実行せず、その存在を認識しません。
- ITD ノードは、セッションレベルで「スティック性」を維持できます。ITD は、NSH-Outer-IP-SRC-Header を調べてこれを行います。
- アップリンク方向では、送信元 IP は「UE-IP」（内部 IP ヘッダーのコピー）です。宛先 IP は「server-IP-internet」です。
- ダウンリンク方向では、NSH ヘッダーはなく、パケットはインターネットから FE に直接送信されます。SAEGW-U では、送信元 IP は「Server-IP」、宛先 IP は「UE-IP」です。
- SAEGW-U はトラフィック分類を実行し、特定のフローのサービスチェーンを選択します。
- SAEGW-U のサービスチェーンには複数のアプライアンスを含めることができ、ステアリング機能はこれらのアプライアンスを処理できます。
- SAEGW-U は、アップリンクパケットの NSH ヘッダーのみをエンコードします。

- SAEGW-U は、元の UE-IP ヘッダーから送信元 IP の詳細を直接コピーします。SAEGW-U は、外部ヘッダー SRC および DEST ポートに NSH ポート 6633 を使用します。宛先 IP（設定されている場合）はアプライアンス IP です。
- NSH ヘッダーを持つダウンリンク パケットを受信すると、SAEGW-U はそのようなパケットをドロップします。
- SAEGW-U は、FE や ITD の正常性チェックを実行しません。SAEGW-U は、ITD を常に使用可能として扱います。
- SAEGW-U は、NSH ベースヘッダー、サービスヘッダー、およびコンテキストヘッダー（メタデータを含む）を使用して、ITD に向かうすべてのアップリンクパケット（サービス機能アプライアンスによって認定）をエンコードします。
- TS アプリケーションは、一度に 1 つのモード（サンドイッチモードまたはスタンドアロンモード）でのみ動作します。

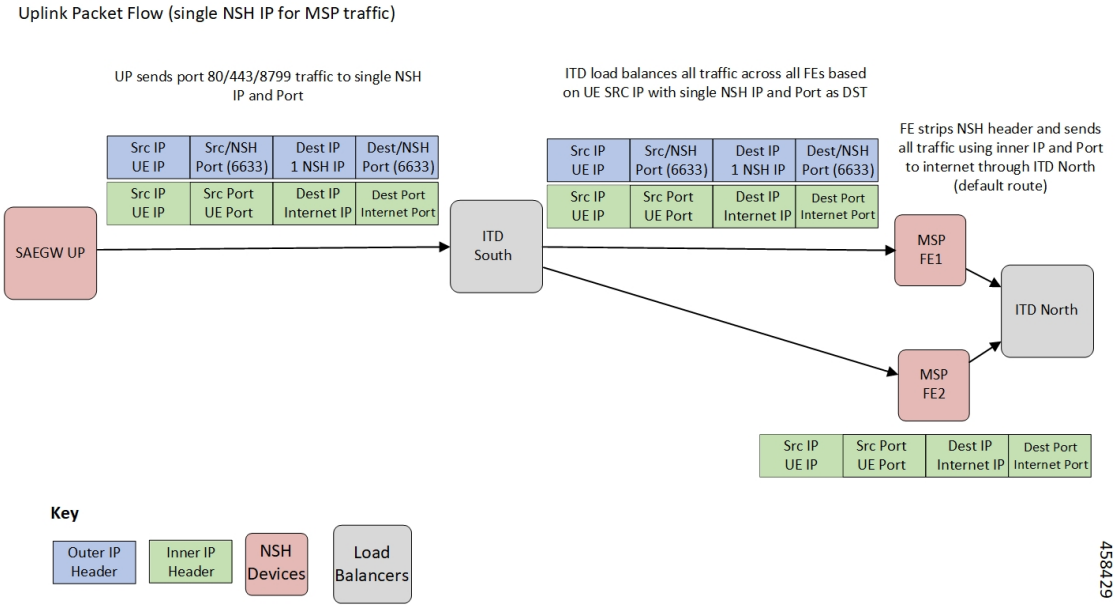
（注）
- サンドイッチモードの場合、**require tsmon** CLI コマンドを設定しないでください。
- サンドイッチモードからスタンドアロンモード、およびその逆に変更した場合は、再起動が必要です。

# 機能の仕組み：サンドイッチモード

## サンドイッチモードのパケットフロー

### アップリンクパケット数

次の図は、アップリンクパケットフローを示しています。




次に、パケットフローについて説明します。

1. GTP-U パケットが SAEGW-U に到着すると、SAEGW-U が GTP ヘッダーのカプセル化を解除し、フローのサブスクライバを特定します。
2. SAEGW-Uがトラフィック分類を実行し、フローのサービスチェーンを関連付けます。サービス機能アプライアンス（ITD）を含むサービスチェーンを、TCP/UDP/HTTP/HTTPS に応じて分類されるトラフィックに関連付けるように SAEGW-U が設定されます。
3. サービス機能アプライアンスに送信される NSH 変数ヘッダーのパラメータをエンコードするために、サービスチェーンに関連付けられた NSH 形式を SAEGW-U が検索します。

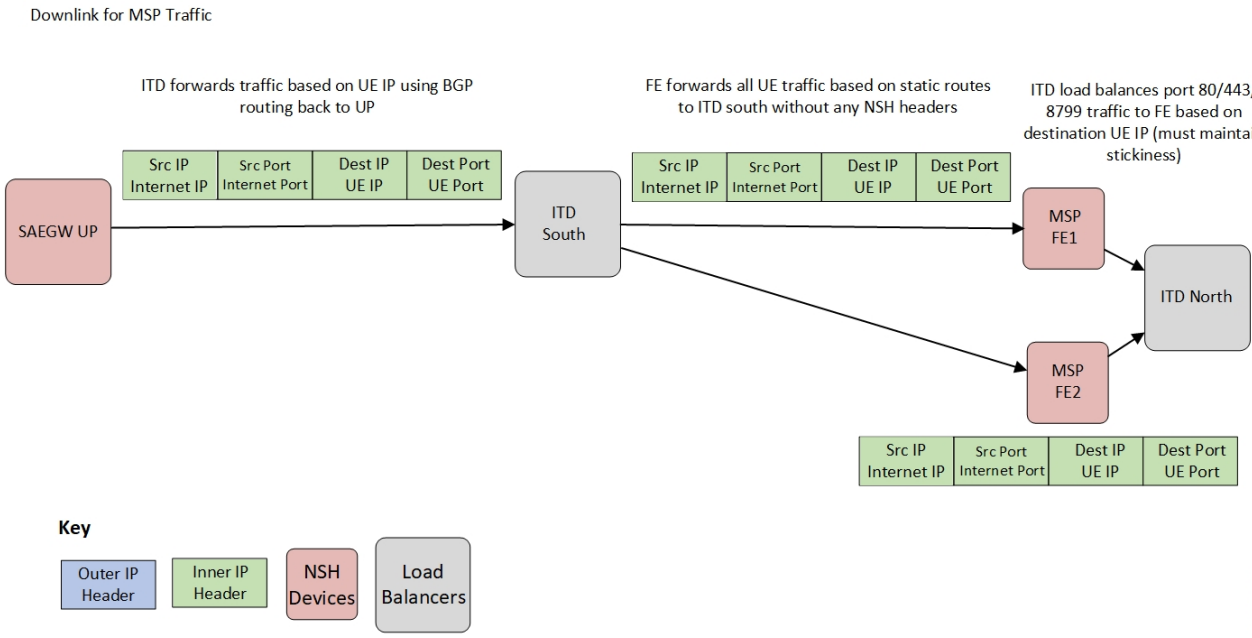
   次に、アップリンクパケット用に選択された SFP が 200 である NSH ヘッダーの例を示します。

```
**************NSH Base Header**************
                   Version: 0
                   OAM Bit: 0
                    Length: 4
                   MD Type: 2
               Next Protocol: 1

**************NSH Service Header**************
   Service Path Identifier: 200
             Service Index:   1

**************Start NSH Context Header**************
                  TLV Type: <MSISDN tag configured in UP>
                  TLV Len: 15
                  TLV Value: 123456789012340 (unencrypted msisdn)

                  TLV Type: <MCCMNC tag configured in UP>
                  TLV Len: 6
                  TLV Value: 404122 (mcc-mnc value)

                  TLV Type: <RAT TYPE tag configured in UP>
                  TLV Len: 1
```

```
                        TLV Value: 3 (rat type value)

                        TLV Type: <APN tag configured in UP>
                        TLV Len: 64
                        TLV Value: APN1 (apn value)

                        TLV Type: <Sub Profile tag configured in UP>
                        TLV Len: 32
                        TLV Value: Profile-1 (Sub Profile name)

                        TLV Type: <SGSN addr tag configured in UP>
                        TLV Len: 4
                        TLV Value: 169090600 (SGSN Addr(in network byte order))

**************End NSH Context Header**************
```

### ダウンリンクパケット数

次の図は、ダウンリンクパケットフローを示しています。




次に、パケットフローについて説明します。

1. パケットはインターネットサーバーから FE に直接送信されます。FE がパケットを処理し、SAEGW-U にパケットを送信します。

   SRC IP/ポートはサーバー IP/ポートで、DEST IP/ポートは UE IP/ポートです。

2. SAEGW-U がパケットを処理し、サービスチェーン内に他のサービス機能アプライアンスがある場合は、追加処理のためにパケットを送信します。サービスチェーンが完了すると、パケットはルールの照合や分類と課金のために通常のダウンリンクパケット処理パスに送信されます。

3. SAEGW-U が、GTP-U ヘッダーがあるパケットをカプセル化し、UE に送信します。

（注） ダウンリンクパケットはNSHエンコードできません。SAEGW-Uでは同様のパケットはすべてドロップされます。

### TCP および UDP トラフィック

#### アップリンクトラフィック

- アプライアンス向けのステアリングに適したすべてのTCPトラフィックとUDPトラフィックは同様に扱われます。
- ULパケットは、設定されたNSHコンテキストヘッダー要素を使用してアプライアンスにステアリングされます。NSHサービスヘッダーはSI=1でエンコードされるため、SI推論がさらに実行され、SI=0の場合、パケットはGiインターフェイスを介して送信されます。
- 外部ヘッダーの SRC IP は、内部ヘッダーの SRC IP（つまり、UE SRC IP）と同じです。
- 外部ヘッダーの SRC ポートは、NSH ポート 6633 です。
- 外部ヘッダーの DST IP は、設定されたアプライアンス IP です。
- 外部ヘッダーの DST ポートは、NSH ポート 6633 です。

#### ダウンリンクトラフィック

ダウンリンクパケットはITDを介してFEから受信されるため、FEにステアリングされることなく、通常の IP パケットとして処理されます。

- SAEGW-Uで受信したULパケットは、適切なSFCに関連付けられた設定済みポリシーに基づいて分類されます。
- SAEGW-Uは、アプライアンスインスタンスと選択されたステアリングのサービスと負荷の可用性に基づいてSFPの選択を実行します。アップリンクトラフィックはNSH（IP-UDP）でカプセル化され、必要に応じて、入力されたコンテキストヘッダーを使用して、選択した SFP でステアリングされます。
- NSH アプライアンスは、NSH パケットを受信すると、IP パケット（場合によってはコンテキストヘッダー）を処理し、Gi インターフェイスを介してパケットを送信します。
- ダウンリンクパケットは、Gi インターフェイスを介して接続先サーバーから SAEGW-U に送信されます。

# トラフィックステアリングのサービススキームの選択

service-scheme は、次の 2 つの方法のいずれかで選択できます。

1. Gx/PCRF：

   PCRF は、次の AVP を介してトラフィックステアリングを有効にします。

```
        [V] Services:
          [V] Service-Feature:
            [V] Service-Feature-Type: TS (4)
            [V] Service-Feature-Status: ENABLE (1)
            [V] Service-Feature-Rule-Install:
              [V] Service-Feature-Rule-Definition:
                [V] Service-Feature-Rule-Status: ENABLE (1)
                [V] Subscription-Scheme: gold
                [V] Profile-Name: L3_profile
```

続いて、TSプロファイルとTSサブスクライバスキームがSxメッセージングを介してユーザープレーンに送信されます。

```
        SUBSCRIBER PARAMS:
...
...
...
            TS-Profile: L3_profile

            TS-Subscriber-Scheme: gold
```

Gx/PCRF ベースのトラフィックステアリングの場合、service-scheme 設定で **trigger subs-scheme-received** CLI コマンドが必要です。

2. Service-scheme フレームワーク（Gx/PCRF AVP なし）：

   トラフィックステアリングは、Subscription-scheme AVP がなくても PCRF から有効にできます。

   **trigger sess-setup** CLI コマンドは、**up-service-chain** を指定したトリガーアクションでは必須です。次に設定例を示します。

```
service-scheme scheme1
trigger sess-setup
  priority 1 trigger-condition subs-scheme-check trigger-action ta2
exit

trigger-condition subs-scheme-check
  any-match = TRUE
exit

trigger-action ta1
  up-service-chain SN-L3_profile

exit
```

# デフォルトのサービスチェーン

TS 対応サブスクライバの場合、次の状況では、特定のトラフィックでサービスチェーン（APP1+APP2）が使用できなくなる可能性があります。

- APP1+APP2 サービスチェーンを選択しようとしている特定のフローに適切なポリシーが設定されていない。
- APP1+APP2 サービスチェーンが選択されたが、APP1 インスタンスが最小インスタンスしきい値を下回っている。このような場合、APP1+APP2 サービスチェーンは使用できません。

- APP1+APP2 サービスチェーンが選択されたが、SFP を選択できなかった。

このようにサービスチェーンが使用できない場合、フローは設定済みのデフォルトサービスチェーンにフォールバックし、フローに対する APP2 サービス処理が保証されます。

デフォルトのサービスチェーンが設定されていない場合、トラフィックはステアリングされずに送信されます。

Gx/PCRF を介した TS 対応の場合、デフォルトのサービスチェーンは **trigger subs-scheme-received** で定義されます。

Gx/PCRF AVP を使用しないサービス スキーム フレームワークを介した TS 対応の場合、デフォルトのサービスチェーンは **trigger sess-setup** で定義されます。

# SFP の選択

**NSH ベースのアプライアンスのみを使用するサービスチェーンの場合**：

ダウンリンクパケットの場合、NSH アプライアンスがないため、SFP はありません。

**L2 および NSH ベースのアプライアンスが混在するサービスチェーンの場合**：

SFP は、L2 の「スティッキネス」に基づいて選択されます。同じ NSH ベースのアプライアンスが存在し、SFP の選択に常に使用できます。

ダウンリンクパケットの場合、SFP の選択は L2 アプライアンスのみに基づいています。

NSH ベースのアプライアンスの負荷の可用性に基づいた SFP の選択は実行されません。NSH とアプライアンスは常に使用可能と見なされます。

# 制限事項と制約事項

この機能には次の既知の制限事項があります。

- スタンドアロンモードからサンドイッチモードに、またはその逆に変更するには、リロードと設定変更が必要です。
- トラフィックステアリングが PCRF から、またはサービススキームフレームワークを使用してローカルで有効になっている場合、そのセッションでトラフィックステアリングを無効にすることはできません。
- マルチ アプライアンス サービス チェーン（L2 および L3 ステアリング）の場合、V4 および V6 トラフィックの SFP は異なります。ただし、両方の SFP は L2 アプライアンスの MSISDN ベースのスティッキ性を維持します。

# NSH トラフィックステアリングの設定：サンドイッチモード

この項では、CP と UP の両方で NSH トラフィックステアリング：サンドイッチモードを設定するために使用できる CLI コマンドについて説明します。

## CP の設定

CP を設定するには、次の手順を実行します。

1. アクティブ課金サービスを設定します。

   次に設定例を示します。

```
configure
  active-charging service ACS
  policy-control services-framework

  trigger-action ta1
   up-service-chain sn-L3-sc  <<<< (This should match the up-service-chain configured
 on UP)
  exit

  trigger-action ta2
   up-service-chain  L3-sc    <<<< (This should match the up-service-chain configured
 on UP)
  exit

  trigger-condition tc1
    rule-name = rule1           <<<<< (This can be static/predef/gor/dynamic rules)
    rule-name = rule2
    multi-line-or
  exit

  trigger-condition tc2
    any-match = TRUE
  exit

  service-scheme scheme1
    trigger rule-match-change
      priority 1 trigger-condition tc1 trigger-action ta1
    exit
    trigger subs-scheme-received      <<<<< (For default service chain selection)
      priority 1 trigger-condition tc2 trigger-action ta2
    exit

    subs-class class1
      subs-scheme = gold  <<<<<<< (This name should match the subscription-scheme
AVP value received from PCRF over Gx)
    exit

    subscriber-base base1
      priority 1 subs-class class1 bind service-scheme scheme1
    exit
end
```

2. トラフィックステアリング AVP は現在、Diameter ディクショナリ custom44 でサポートされています。Diameter ディクショナリにより、TS 関連の AVP が Gx インターフェイスを介して受信されるとき、および Sx メッセージで UP に送信されるときに、CP はこれらの AVP を適切に復号できます。

   CP でディクショナリを設定するための設定例を以下に示します。

```
configure
  context ISP1
    ims-auth-service IMSGx
    policy-control
    diameter dictionary dpca-custom44
  exit
end
```

CCA-I/CCA-U/RAR の Gx を介して受信する TS 関連 AVP の値の例を以下に示します。

```
    [V] Services:
      [V] Service-Feature:
        [V] Service-Feature-Type: TS (4)
        [V] Service-Feature-Status: ENABLE (1)
        [V] Service-Feature-Rule-Install:
          [V] Service-Feature-Rule-Definition:
            [V] Service-Feature-Rule-Status: ENABLE (1)
            [V] Subscription-Scheme: gold
            [V] Profile-Name: L3
```

# UP の設定

UP を設定するには、次の手順を同じ順序で実行します。

1. サービス チェーン アプライアンスにデータを送信するために使用されるコンテキストにインターフェイスを追加します。

   次に設定例を示します。

```
configure
  context ISP1-UP
    interface ts_ingress
    ip address 209.165.200.225 255.255.255.224
    ipv6 address 4101::1/64 secondary
  exit
end

configure
  context ISP2-UP
    interface ts_egress
    ip address 209.165.200.225 255.255.255.224
    ipv6 address 4101::2/64 secondary
  exit
end
```

2. 新たに追加されたインターフェイスを UP の物理ポートにバインドします。

   次に設定例を示します。

```
configure
  port ethernet 1/11
    vlan 1240
      no shutdown
```

```
      bind interface ts_ingress ISP1-UP
    exit
  exit
  port ethernet 1/12
    vlan 1240
      no shutdown
      bind interface ts_egress ISP2-UP
    exit
  exit
end
```

3. TS 関連の設定を UP に追加します。

   次に設定例を示します。

```
configure
  ts-bind-ip IP_UP01 ue-src-ip ipv4-address 209.165.200.225     <<<< See Notes below


  nsh
    up-nsh-format nfo
      tag-value 1  apn encode
      tag-value 2  imsi encode
      tag-value 3  mcc-mnc encode
      tag-value 4  msisdn encode
      tag-value 5  rat-type encode
      tag-value 10 rating-group encode
      tag-value 11 sgsn-address encode
      tag-value 12 subscriber-profile encode
    exit
  exit

  traffic-steering
    up-service-chain L3
      sfp-id 1 direction uplink up-appliance-group L3 instance 1
    exit

    up-service-chain sn_L3
     sfp-id 3  direction uplink   up-appliance-group L2 instance 1 up-appliance-group
 L3 instance 1
      sfp-id 4  direction downlink up-appliance-group L2 instance 1
     sfp-id 5  direction uplink   up-appliance-group L2 instance 2 up-appliance-group
 L3 instance 1
      sfp-id 6  direction downlink up-appliance-group L2 instance 2
     sfp-id 7  direction uplink   up-appliance-group L2 instance 3 up-appliance-group
 L3 instance 3
      sfp-id 8  direction downlink up-appliance-group L2 instance 3
     sfp-id 9  direction uplink   up-appliance-group L2 instance 4 up-appliance-group
 L3 instance 3
      sfp-id 10 direction downlink up-appliance-group L2 instance 4

    exit
    up-appliance-group L3
      steering-type nsh-aware
      up-nsh-format nfo
      min-active-instance 1
      instance 1 ip address 40.40.40.3
    exit
    up-appliance-group L2
      steering-type l2-mpls-aware
      min-active-instance 1
      instance 1 ingress slot/port 1/13 vlan-id 2136 egress slot/port 1/12 vlan-id
2136  ingress-context ingress ip address 4101::1 egress-context egress ip address
4101::2
```

```
      instance 2 ingress slot/port 1/13 vlan-id 2137 egress slot/port 1/12 vlan-id
2137  ingress-context ingress ip address 4201::1 egress-context egress ip address
4201::2
      instance 3 ingress slot/port 1/13 vlan-id 2138 egress slot/port 1/12 vlan-id
2138  ingress-context ingress ip address 4301::1 egress-context egress ip address
4301::2
      instance 4 ingress slot/port 1/13 vlan-id 2139 egress slot/port 1/12 vlan-id
2139  ingress-context ingress ip address 4401::1 egress-context egress ip address
4401::2
    exit
```

注：

- **ts-bind-ip** *name* **ue-src-ip { ipv4-address** *ipv4_address* **| ipv6-address** *ipv6_address* **}**：パケットが ITD に送信される UP インターフェイスの IP アドレスを指定します。

4. **show configuration** CLI コマンドを使用して前述の設定を確認します。次に、**commit** CLI コマンドを実行して設定を有効にします。

```
configure
  traffic-steering
    commit
end
```

# スタンドアロンとサンドイッチの両モードでの後処理 Ruledef の設定

**up-service-chain** トリガーアクションは、トリガー条件を含めて、トラフィックをステアリングするために rulebase 内の ruledef の後処理設定で使用されます。複数の課金 ruledef がある場合でも、HTTP、HTTPS、およびその他のプロトコルのポート番号を使用して単一の後処理 ruledef が定義されます。この単一の後処理 ruledef 名は、トラフィックステアリングで使用されるトリガー条件で照合されます。

次の設定を使用して、トラフィックをステアリングするための ruledef の後処理を設定します。

```
configure
  active-charging service service_name
    rulebase rulebase_name
      post-processing priority priority_number ruledef ruledef_name
charging-action charging_action_name
      end
```

次の設定を使用して、後処理 ruledef のトリガー条件を設定します。

```
configure
  trigger-condition trigger_condition_name
    rule-name rule_name
    post-processing-rule-name post_processing_rule_name
    end
```

# UPアプライアンスグループでのインターフェイス名を使用したBFDインスタンスIDの設定

トラフィックステアリング中に、**up-appliance-group** 内で、インターフェイス名と IP 設定を使用して BFD インスタンス ID が設定されます。

次の設定を使用して、トラフィックをステアリングするための BFD インスタンス ID を設定します。

```
configure
  traffic-steering
    up-appliance-group up_appliance_group_name
      steering-type steering_type
      instance instance_id ingress slot/port slot_or_port_number vlan-id
vlan_id egress slot/port slot_or_port_number vlan-id vlan_id ingress-context
ingress interface-name interface-name egress-context egress interface-name
 interface-name
      end
```

（注）

- 特定の L2 **up-appliance-group** では、BFD インスタンス ID は IP アドレスを使用して、または対応するインターフェイス名を使用し、特定の **ingress** または **egress** に関する **interface-name** を使用して設定されます。
- **interface-name** を使用した BFD モニタリングの **up-appliance-group** 設定が完了し、BFD の登録が完了するまで最大 5 分かかります。
- BFD の登録が成功すると、IP アドレスと **interface-name** が **show user-plane traffic-steering up-appliance-group all** の出力で使用可能になります。
- BFD モニタリングを使用して **up-appliance-group** で使用されている **interface-name** の IP アドレスが変更された場合は、**up-appliance-group** を再設定する必要があります。

# NSHトラフィックステアリングのモニタリングとトラブルシューティング：サンドイッチモード

ここでは、この機能のモニタリングと障害対応で使用できる CLI コマンドについて説明します。

SNMP トラップの詳細については、この章の「SNMP トラップ （28 ページ）」の項を参照してください。

バルク統計情報の詳細については、この章の「バルク統計 （28 ページ）」の項を参照してください。

# コマンドの表示

この項では、この機能をサポートするために使用可能な show CLI コマンドについて説明します。

### CP コマンド

機能をモニターおよび障害対応するには、CP で次の show CLI コマンドを使用します。**show active-charging sessions full all**

TS Subscription Scheme Name：active-charging-service で設定されたサービススキームから適用する必要があるサブスクリプションスキームが表示されます。この active-charging-service は、Gx インターフェイスを介して PCRF から受信されます。

### UP コマンド

機能をモニターおよび障害対応するには、UP で次の show CLI コマンドを使用します。

- トラフィックステアリングの設定チェック
  - **show user-plane-service traffic-steering up-service-chain all**
  - **show user-plane-service traffic-steering up-service-chain name** *up_service_chain_name*
  - **show user-plane-service traffic-steering up-service-chain sfp-id** *sfp_id*
  - **show user-plane traffic-steering up-appliance-group name** *name* **instance-id** *id*
  - **show user-plane traffic-steering up-appliance-group name** *name*
  - **show user-plane traffic-steering up-appliance-group all**
  - **show user-plane traffic-steering up-service-chain name** *name*
  - **show user-plane traffic-steering up-service-chain sfp-id** *id*
  - **show user-plane traffic-steering up-service-chain all**
- トラフィックステアリングの統計情報
  - **show user-plane-service inline-services traffic-steering statistics up-service-chain all verbose**
  - **show user-plane-service inline-services traffic-steering statistics up-service-chain all**
  - **show user-plane-service inline-services traffic-steering statistics up-service-chain sfp-id** *sfp_id*
  - **show user-plane-service inline-services traffic-steering statistics up-appliance-group name** *appliance_group_name*
  - **show user-plane-service inline-services traffic-steering statistics up-appliance-group name** *appliance_group_ name* **instance** *appliance instance*
  - **show user-plane-service statistics trigger-action all**
- サービスチェーンと SFP 関連付け

- **show subscriber user-plane-only flows**
- **show subscribers user-plane-only callid** *call_id* **flows**

## show user-plane traffic-steering up-appliance-group all

機能のモニタリングや障害対応には、次の show CLI コマンドを使用します。

- **show in interface-name out interface-name**

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。