

ディープ パケット インスペクションとインラインサービス

- 機能の概要と変更履歴, on page 1
- 機能説明, on page 2
- 機能の仕組み （4 ページ）
- サポートされるインラインサービス （8 ページ）
- 静的ルールと事前定義済みルールを構成する, on page 62
- DPI 向け L7 プロトコルの ACS Ruledef を構成する （63 ページ）
- DPI 向け L7 プロトコルの課金アクションの構成 （65 ページ）

機能の概要と変更履歴

要約データ

該当製品または機能エリア	5G-UPF
該当プラットフォーム	VPC-SI SMI
機能のデフォルト設定	無効：設定が必要
このリリースでの関連する変更点	N/A
関連資料	*UCC 5G UPF* 構成およびアドミニストレーション ガイド

マニュアルの変更履歴

改訂の詳細	リリース
• N4 インターフェイスの NAT サポートが追加されました。 • TCP アイドルタイムアウトのサブスクライバ ファイアウォールのサポートが追加されました。 • 2 つの新しい属性を使用してデータストールの問題をキャプチャするためのサポートが追加されました。	2023.04.0
次の機能に対してサポートが追加されました。 • IP 再アドレス指定 • RTP ダイナミックフローの検出 • ベアラー固有フィルタのルール照合 • QUIC IETF の導入	2021.02.0
ディープ パケット インスペクション（DPI）んお一部として、新しい L7 プロトコルが導入されました。	2021.01.0
TCP 用に次の EDR 属性が追加されました。 • SYN および SYN-ACK パケット • SYN-ACK および ACK パケット	2021.01.0
EDR に新しい DNS 属性が導入されました。	2021.01.0
最初の導入。	2020.02.0

機能説明

Cisco 5G-UPF の主要な製品機能の 1 つとして、統合されたディープ パケット インスペクション（DPI）ベースのサービスが挙げられます。DPI は、Uniform Resource Identifier（URI）情報を含むレイヤ 7（L7）の検査です。場合によっては、トリガー条件を識別するレイヤ 3（L3）およびレイヤ 4（L4）アナライザは、課金目的では不十分であるため、レイヤ 7（L7）検査が使用されます。

DPI は、L4 検査よりも後のパケットインスペクションを実行し、通常、L7 で URI 情報（DNS、HTTP、HTTPS、RTP、RTSP URL など）を検出するためにデプロイされます。

DNS サーバーアドレス再指定

不正な DNS サーバーを使用する場合は常に、要求を変更して DNS IP のアドレスを再指定し、許可されたサーバーを使用することができます。ruledef は、パケットが DNS クエリに属するかどうかを判別します。また、DNS クエリが一連の許可 DNS サーバーに属しているかどうかも判別します。DNS クエリが許可された DNS サーバーに属していない場合、フローアクションは、アドレス再指定サーバーリストから DNS サーバーを取得します。

active-charging サービスで **readdress-server-list** コマンドを構成できます。フローが、**ruledef** と一致すると、フローアクションを構成して、**readdress-server-list** からサーバーを使用できます。

次の構成を使用して、active-charging サービスでアドレス再指定サーバーリストを構成します。

```
configure
  active-charging service service_name
    readdress-server-list name_of_list
      server ipv4_address [ port ]
      server ipv6_address [ port ]
```

（注） アドレス再指定サーバーリストには、最大 10 のサーバーを構成でき、active-charging サービスには、最大 10 のアドレス再指定サーバーリストを構成できます。同じ **readdress-server-list** 内に、IPv4 アドレスと IPv6 アドレスの両方を設定できます。

次のいずれかのメソッドを使用して、リストから **readdress-server-list** を選択します。

- ラウンドロビン：新しいフローごとにラウンドロビン方式でサーバーが選択されます。選択時に、リスト内の非アクティブサーバーは考慮されません。

  このメソッドが、デフォルトの選択です。

- 階層型：このアプローチでは、サーバーは **readdress-server-list** で定義されている順序に応じて、プライマリ、セカンダリ、ターシャリというようにタグ付けされます。すべてのフローのアドレス再指定先は、プライマリサーバーが使用可能である限り、プライマリサーバーです。プライマリサーバーがダウンした場合は、フローのアドレス再指定先はセカンダリサーバーとなり、その後も同じロジックが繰り返されます。プライマリサーバーが再びアクティブになると、フローのアドレス再指定先はプライマリサーバーに戻ります。

DNS アドレス再指定サーバーリストを設定するには、active-charging サービスで次の CLI 構成を使用します。

```
configure
  active-charging service service_name
    readdress-server-list name_of_list
      server ipv4_address [ port ]
      server ipv6_address [ port ]
      consecutive-failures integer_value
      response-timeout integer_value
```

```
reactivation-time integer_value

charging-action action_name
  flow action readdress server-list name_of_list
  exit
```

注：

- **consecutive-failures**：1 ～ 10 の整数を指定します。デフォルト値は *5* です。
- **response-timeout**：1 ～ 10000 ミリ秒の整数を指定します。デフォルト値は *1000* です。
- **reactivation-time**：1 ～ 1800 秒の整数を指定します。デフォルト値は *300* です。

アドレス再指定先サーバーの状態

この項では、アドレス再指定サーバーの状態について説明します。

- アクティブ状態：一度設定すると、すべてのサーバーが [Active] としてマークされます。
- 非アクティブ状態：アドレス再指定先サーバーからの応答が受信されない場合、サーバーは [Inactive] としてマークされます。
- アクティブ保留状態：サーバーが [Active-Pending] 状態になると、アドレス再指定要求を受け付け可能となります。この状態の場合、リクエストがこのサーバーにアドレス再指定されると、レスポンスが返り、サーバー状態が、[アクティブ（Active）] に変わります。それ以外の場合は、[Inactive] 状態に戻ります。

機能の仕組み

この項では、以下の DPI の機能について説明します。

- ダウンリンクおよびアップリンクパケットの DSCP マーキング。
- 通信のアドレス再指定またはリダイレクト

ダウンリンクおよびアップリンクパケットの DSCP マーキング

トランスポートレベルのマーキングは、Differentiated Services Code Point（DSCP; DiffServ コードポイント）値を使用して UPF で通信をマーキングするプロセスです。QoS フローごとに実行されるトランスポートレベルのマーキングは、5QI からのマッピングと、SMF からのオプションの割り当ておよび保持ポリシー（ARP）構成に基づいています。

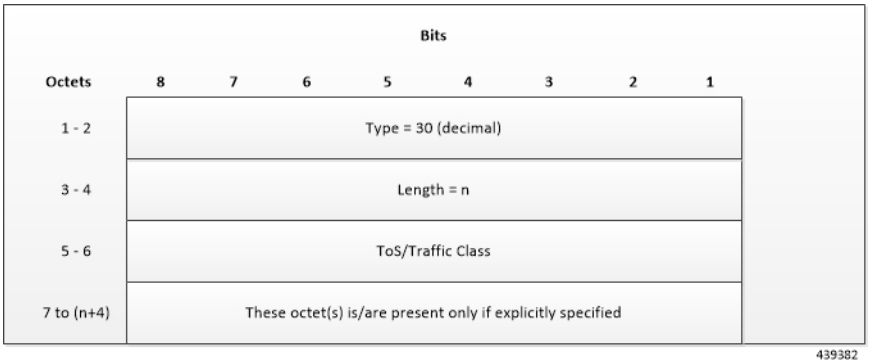
SMF は、マークされる通信と一致する PDR に関連付けられている FAR の Transport Level Marking IE の ToS（IPv4）または Traffic Class（IPv6）の DSCP を指定することで、通信レベルのマーキングを制御します。UPF は、検出された通信に対して、トランスポートレベルのマーキングを実行し、マークされたパケットをピアエンティティに送信します。SMF は、関連する FAR の Transport Level Marking IE を変更して、トランスポートレベルのマーキングを変更できます。

UPF は次の IE をサポートしています。

- Inner Packet Marking IE：内部パケットマーキングのトンネルパケットをマークします。
- Transport Level Marking Options IE：内部パケットから外部 IP パケットに DSCP をコピーします。

Transport Level Marking IE

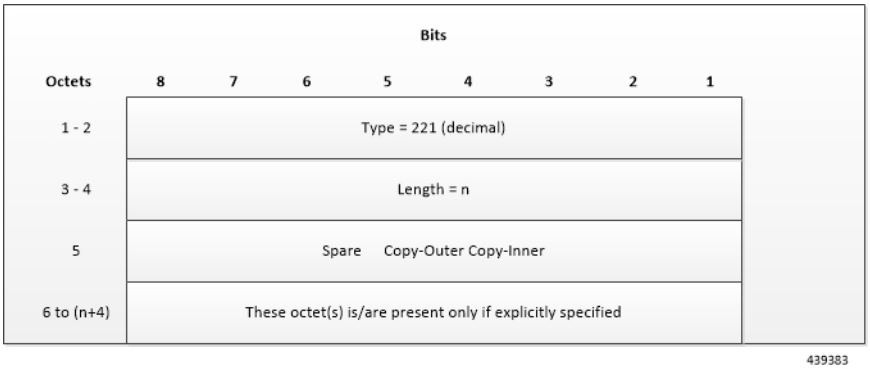
「Transport Level Marking」 IE のタイプは、以下の図に示すようにエンコードされています。これは、downlink transport-level marking の DSCP 値を示しています。



Type-of-Service（ToS）または Traffic Class のエンコーディングは、OctetString として 2 つのオクテット形式で実行されます。最初のオクテットには、[IPv4 Type-of-Service] フィールドまたは [IPv6 Traffic-Class] フィールドの DSCP 値が含まれ、2 番目のオクテットには、*0xFC*に設定されてる [ToS/Traffic Class] マスクフィールドが含まれます。

Transport Level Marking Options IE

「Transport Level Marking Options」 IE のタイプは、以下の図に示すようにエンコードされています。ダウンリンクトランスポート レベル マーキングの DSCP は、内部パケットからコピーされます。

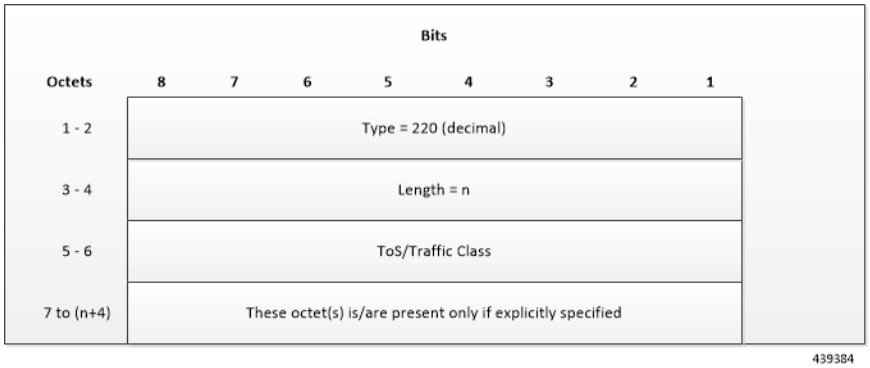


Copy-Inner および Copy-Outer フラグは、オクテット 5 のビット 0 とビット 1 に存在します。ISP から受信するパケットには外部ヘッダーが存在しないため、ダウンリンクパケットには

Copy-Outer フラグが使用されません。Copy-Inner フラグが存在する場合、UPF は内部パケットの DSCP 値を使用してトランスポートレベルの IP ヘッダーをマークします。

Inner Packet Marking IE

次の図に示すように、「Inner Packet Marking」IE タイプがエンコードされます。ダウンリンクの Inner Packet Marking の DSCP 値を示します。



ToS/通信クラスのエンコーディングは、OctetString として 2 つのオクテットの形式で実行されます。最初のオクテットには IPv4 ToS または IPv6 Traffic Class フィールドの DSCP 値が含まれ、2 番目のオクテットには ToS/Traffic Class マスクフィールドが含まれます。このフィールドは、*0xFC* に設定されています。

注：

- ユーザープレーンパケットの IP ヘッダー内の元の ECN ビットは、Transport Level Marking または Inner Packet Marking の適用後は変更されません。
- 「Transport Level Marking」IE、「Inner Packet Marking」IE または両方の IE が、アップリンク FAR に関連付けられている場合、次のリリースがアップリンクパケットマーキングに適用されます。
  - 「Transport Level Marking」IE または「Inner Packet Marking」IE が存在する場合、その DSCP 値が使用されます。
  - 「Transport Level Marking」IE および「Inner Packet Marking」IE の両方が存在する場合、アップリンクパケットマーキングには、「Transport Level Marking」IEからの値が使用されます。

通信のアドレス再指定またはリダイレクト

通信リダイレクトは、アップリンクアプリケーション通信をリダイレクト接続先にリダイレクトするプロセスです。たとえば、一部の HTTP フローをサービス プロビジョニング ページにリダイレクトします。リダイレクト接続接続先は、PCF によって提供されるか、SMF または UPF で事前設定済みです。

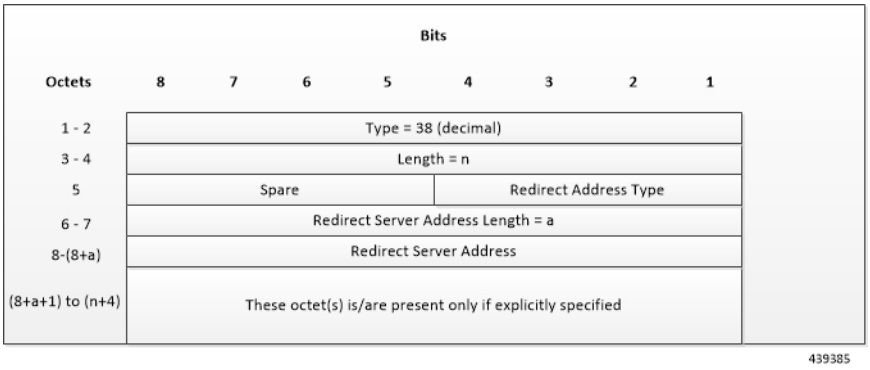
通信リダイレクトの適用は、SMF または UPF がサポートする通信が、通信リダイレクトの対象となる場合、UPF で適用されます。UPF は、「UP-Function Features」IE 経由で UPF での通信リダイレクトの適用をサポートしているかどうかを SMF に報告します。

UPF で通信リダイレクトを適用するために、SMF は次のアクションを実行します。

- 通信リダイレクトを表すために必要な PDR が存在しない場合は、作成します。
- 次を使用して FAR を作成します。
  - リダイレクト接続先を含む「Redirect Information」IE（通信が SMF によって提供されるリダイレクト接続先へのリダイレクトが必要な場合）。SMF からのリダイレクト接続先は、UPF で事前設定済みのリダイレクト接続先よりも優先されます。
    - HTTP 通信リダイレクトの場合、リダイレクトアドレスタイプは「URL」に設定され、SMF は FAR の「Destination Interface」IE を（リダイレクトを示す状態コードとともに HTTP 応答メッセージを転送する）「Access」に設定します。他のタイプの通信リダイレクトの場合、FAR の「Destination Interface」IE は「Core」に設定されます。
- FAR を PFCP セッションの上記の PDR に関連付けます。

Redirect Information IE

リダイレクト情報は、次のようにエンコードされます。



リダイレクトアドレスのタイプを示す「Redirect Address Type」：

Redirect Address Type	値（10 進数）
IPv4 アドレス	0
IPv6 アドレス	1
URL	2
SIP URI	3
スペア（将来使用向け）。	4 ～ 15

「リダイレクト サーバー アドレス長」は、「リダイレクト サーバー アドレス」の長さを示します。「リダイレクトサーバーアドレス」は、UTF8String フォーマットでエンコードされており、エンドユーザーと接続するリダイレクトサーバー（例：HTTP リダイレクトサーバー、SIP サーバー）のアドレスが含まれます。

Important このリリースでは、FAR が クォータ期限切れの URR と関連付けられている場合に、ダイナミックルールに対して、リダイレクトアドレスタイプ URL がサポートされます。

サポートされるインラインサービス

アプリケーション検出および制御

ADC インラインサービスは、通信を分析することでピアツーピアプロトコルを検出するために使用されます。ソーシャルネットワーキングやゲームミングアプリケーションなど、大量のインターネット通信を生成するその他の一般的なアプリケーションも検出できます。

ADC と呼ばれるインライン サービスは、「P2P」とも呼ばれます。ピアツーピア（P2P）とは、2 つのわずかに異なるコンテキストで使用される用語です。機能レベルでは、クライアント/サーバー方式とは対照的に、ピアリング方式で対話するプロトコルを意味します。ノードの機能の間に明確な区別はありません。どのノードもクライアント、サーバー、またはその両方として機能できます。プロトコルはこの 2 つを明確に区別しない場合があります。たとえば、ピアリング交換には、クライアントとサーバーの機能や情報の送受信が同時に含まれることがあります。

（注） ADC のサポートは、ライセンスが必要な機能です。ライセンスの入手方法の詳細については、シスコのアカウントまたはサポート担当者にお問い合わせください。

QUIC IETF の導入

現在のフレームワークでは、プラグイン到達時に、フロー内のすべてのパケットに対してディープ パケット インスペクション（DPI）が実行されます。DPI は、パケットを分析し、確定的なパターンを抽出することによって実行されます。DPI は順番に実行され、アプリケーションの検出とそのサブタイプの分類を行います。プラグインは、DPI 後のフローを除外します。フローは検出後にオフロードされます。QUIC IETF の一部として、初回の QUIC ハンドシェイクパケット（Client Hello/Server Hello）はネットワーク上で暗号化されます。したがって、アプリケーションの検出に使用できる確定的なパターンはありません。p2p プラグインによる、検出用 SNI（Server Name Indication）の復号と取得が新たにサポートされるようになりました。

QUIC IETF の設定

QUIC IETF 復号を有効または無効にするには、次の設定を使用します。

```
configure
  active-charging service acs_service_name
    p2p-detection debug-param protocol-param p2p_quic_ietf_decrypt 1
    end
```

（注）
- デフォルトでは、CLI は無効になっており、TLS 復号によるパフォーマンスへの影響は最小限です。
- ランタイム変更の構成は、既存のフローに影響しません。変更は、新しいフローにのみ適用されます。

統計情報

show user-plane-service statistics analyzer name p2p

この show CLI コマンドを使用して、QUIC とアプリケーションに関して分析するパケットを判断します。

コンテンツ フィルタリング

コンテンツフィルタリングは、3GPP および 3GPP2 ネットワークで使用可能なインラインサービスです。HTTP リクエスト内の URL に基づいてモバイルサブスクライバからの HTTP リクエストをフィルタ処理します。これにより、オペレータは個々のサブスクライバがアクセスできるコンテンツをフィルタ処理して制御できるため、サブスクライバが常識的に容認されないコンテンツや望まないコンテンツに思いがけずさらされることはありません。

コンテンツフィルタリングの構成

コンテンツフィルタリングを有効にするには、次の追加設定を使用します。

```
configure
  require user-plane content-filtering
    content-filtering category database directory path path_address
    content-filtering category database max-version version_number
    end
```

（注）
コンテンツフィルタリングを有効にするには、ブート時に上記の構成を UPF で構成する必要があります。ユーザープレーン設定後に上記の設定を定義すると、エラーや不整合が発生します。

show コマンドの入力と出力

show subscribers user-plane-only callid *call_id* full all

SMF は、Session Establishment/Modification 要求で、コンテンツ フィルタリング ポリシー ID を指定します。この機能をサポートするために、次のフィールドが表示されます。

- SUBSCRIBER パラメータ
  - コンテンツ フィルタリング ポリシー ID

DNS スヌーピング

課金

DNS スヌーピングの課金は SM-P で実行されます。

ルール定義

ルール定義のホスト名（domain-names）とホスト名の一部を指定するには、次の CLI コマンドを使用します。

```
ruledef <ruledef_name>
        ip [server-domain-name {contains|=|ends-with|starts-with}
<url_string>]
        ip [server-domain-name {contains|=|ends-with|starts-with}
<url_string>]
        multi-line-OR enabled
```

ip server-domain-name のルールラインを削除するには、この CLI の no バージョンを使用します。

```
ruledef <ruledef_name>
  no ip [server-domain-name {contains|=|ends-with|starts-with}
<url_string>]
  exit
```

ECS レベルで DNS エントリについて設定可能なタイマーには、次の CLI を使用します。

```
configure
         active-charging service service_name
            ip dns-resolved-entries timeout <value_secs>
            end
```

ip server-domain-name キーワードを含む ruledef が定義され、ルールベースで使用されるたびに、インスタンス単位でルールベースごとに ip-table が作成されます。

ルールの照合

この機能は、非 CUPS アーキテクチャの機能と同じです。

show CLI

次の CLI を使用して、DNS IP エントリ：**show user-plane-service [statistics dns-learnt-ip-addresses {summary | sessmgr instance <id> |all [verbose] }]** のテーブルを確認します。

バルク統計情報

DNS スヌーピング機能をサポートするために、次のバルク統計情報を使用できます。

- ecs-dns-learnt-ipv4-entries
- ecs-dns-flushed-ipv4-entries
- ecs-dns-replaced-ipv4-entries
- ecs-dns-overflown-ipv4-entries
- ecs-dns-learnt-ipv6-entries
- ecs-dns-flushed-ipv6-entries
- ecs-dns-replaced-ipv6-entries
- ecs-dns-overflown-ipv6-entries

前述のバルク統計情報は、非 CUPS アーキテクチャと同様に ECS スキーマに追加されます。

（注） SNMP トラップ生成コマンドは、CUPS DNS スヌーピング機能ではサポートされていません。

イベントデータ レコード

表 *1*：機能の履歴

機能名	リリース情報	機能説明
EDR の最終アップリンクおよびダウンリンクパケット時間属性	2023.04	EDR 設定では、データ停止の問題を特定するために、2 つの新しい属性 **sn-last-uplink-pkt-time** と **sn-last-downlink-pkt-time** がサポートされています。

機能説明

Event Data Records（EDR）は、システム管理ユーザーによるコンテンツ情報、形式、および生成トリガーの構成をサポートする、使用状況レコードです。

フローが終了すると、UPF は終了したフローの詳細情報を使用して EDR を生成します。

機能の仕組み

EDRは、フロー終了時にユーザープレーンから生成されます。コールセットアップとコール変更中、EDR 生成に必要なすべてのコール固有属性は、Sx Establishment/Modification 要求メッセージの 「Subscriber Params」IE の一環として、SMF から UPF に送信されます。

フローの終了時に、VPP から課金カウンタが取得されます。EDR 形式の設定で設定されたすべてのコールレベル属性は、課金やボリュームカウンタの属性とともに CDRMOD proclet に送信されます。この proclet により、該当レコードがファイルまたはディスクに書き込まれ、設定された外部サーバーに転送されます。

TCP Fast Open

TCP Fast Open（TFO）は、2 つのエンドポイント間の連続する TCP 接続をより迅速に開始するための拡張機能です。そのためにこの機能では、クライアントに保存され、サーバーとの初回接続時に設定される暗号化 Cookie である TFO Cookie（TCP オプション）を使用します。クライアントが再接続する際、認証のために初回 SYN パケットとともに TFO Cookie データも送信します。認証が成功すると、サーバーは、3 ウェイハンドシェイクの最後の ACK パケットを受信する前であっても、クライアントへのデータ送信を開始できます。このため、EDR の TCP フローの制御パケット間の時間差を計算して記録するために、次のパケット間の差が記録されます。

- SYN および SYN-ACK パケット
- SYN-ACK および ACK パケット

EDR 内の情報をキャプチャするために追加されるルール変数の詳細については、「その他の *TCP* フィールドを構成する」項を参照してください。

トランザクション完了 EDR

HTTP トランザクションが完了すると、HTTP EDR に対してTransaction Complete EDR が生成されます。完了すると、VPP から課金カウンタがフェッチされます。EDR 形式の設定で設定されたすべてのコールレベル属性は、課金やボリュームカウンタの属性とともにCDRMOD proclet に送信されます。この proclet により、該当レコードがファイルまたはディスクに書き込まれ、設定された外部サーバーに転送されます。

次の EDR 属性がサポートされています。

- sn-start-time
- sn-end-time
- sn-start-time format MM/DD/YYYY-HH:MM:SS:sss
- sn-end-time format MM/DD/YYYY-HH:MM:SS:sss
- radius-calling-station-id
- radius-called-station-id
- bearer 3gpp imsi

- bearer 3gpp imei
- bearer 3gpp rat-type
- bearer 3gpp user-location-information
- ip subscriber-ip-address
- ip dst-address
- sn-ruledef-name
- sn-subscriber-port
- sn-server-port
- sn-app-protocol
- sn-volume-amt ip bytes uplink
- sn-volume-amt ip bytes downlink
- sn-flow-start-time format seconds
- sn-flow-end-time format seconds
- sn-volume-amt ip pkts uplink
- sn-volume-amt ip pkts downlink
- sn-direction
- traffic-type
- p2p protocol
- p2p app-identifier tls-cname
- p2p app-identifier tls-sni
- p2p app-identifier quic-sni
- bearer 3gpp sgsn-address
- sn-rulebase
- sn-charging-action
- flow tethered-ip-ttl
- flow ttl
- flow ip-control-param
- bearer qci
- tcp flag
- ip server-ip-address
- sn-flow-id
- sn-closure-reason

- sn-duration
- ip src-address
- IP プロトコル
- sn-charge-volume ip bytes uplink
- sn-charge-volume ip bytes downlink

次の HTTP EDR 属性がサポートされています。

- http url length 2000
- http request method
- http content type
- http user-agent length 255
- http reply code
- http referer
- http host
- http cookie
- http header-length
- transaction-uplink-bytes
- transaction-downlink-bytes

次の DNS EDR 属性がサポートされています。

- dns answer-ip-list
- dns answer-name
- dns previous-state
- dns query-name
- dns query-type
- dns return-code
- dns state
- dns tid

データの停滞を特定する EDR 属性

EDR は、データ停滞の問題を特定するために、次の属性をサポートしています。

- **sn-last-uplink-pkt-time**
- **sn-last-downlink-pkt-time**

アップリンクまたはダウンリンク報告の最後のパケットと**sn-end time**（フローリリース時間）間で平均差が大きい場合、これは、データの停滞を示しています。

これらのEDR変数は、制御パケット（FIN/RST/FIN-ACK/TCP TEAR DOWN ACK）のパケット時間はキャプチャしません。これらのパケットのパケット時間は、**sn-end time**を使用してセッションマネージャでキャプチャされます。

オフロードされたフローの場合、属性**sn-last-uplink-pkt-time**と**sn-last-downlink-pkt-time**は、fastpathを通過するパケットに対して更新されます。

制限事項

UPDのEDR機能には、次の制限があります。

- EDRは、フロー終了条件（アイドルタイムアウト、HAGR、通常フロー終了、およびセッション終了中）に対してのみ生成されます。
- 課金アクションベースのEDR設定はサポートされません。
- EDRのレポートはサポートされません。
- コントロールプレーンのパケット時間は、EDRのlast_uplink_packet_time and last_downlink_packet_time except mid session TCP ACKには更新されません。

イベントデータレコードを構成する

UPFでEDRを構成する

UPFでEDR属性を構成するには、次の構成を使用します。

```
active-charging service service_name
   rulebase rulebase_name
      flow end-condition { timeout | normal-end-signaling | session-end
 | hagr } charging-edr charging_edr_format_name
      edr transaction-complete { http | dns } charging-edr
charging_edr_format_name
      exit
      edr-format format_name
         attribute attribute_name format time_format priority priority_val
```

注：

- **flow end-condition**：このコマンドを使用すると、ユーザーセッションに関連するセッションフローの終了条件を設定し、EDR生成をトリガーできます。
- **timeout**：タイムアウト条件が原因でフローが終了するたびに、指定されたEDR形式でEDRを作成します。
- **normal-end-signaling**：フロー終了が正常に通知されるたびに、指定されたEDR形式でEDRを作成します。たとえば、TCPフローのFINとACKを検出し、指定されたEDR形式を使用してフローのEDRを作成します。

- **session-end**：サブスクライバセッションが終了するたびに、指定された EDR 形式で EDR を作成します。このオプションを使用すると、セッションマネージャは、セッション終了時にフローに対して最後の EDR が作成されてからアクティビティがあったすべてのフローに対して、指定された形式名で EDR を作成します。
- **charging-edr** *charging_edr_format_name*：課金 EDR 形式を指定します。
- **hagr**：シャーシ間 Session Recovery アクションによりフローが終了するたびに、指定された EDR 形式で EDR を作成します。
- **http**：HTTP プロトコル関連の設定を指定します。
- **dns**：DNS プロトコル関連の設定を指定します。
- **edr-format** *format-name*：構成する必要がある EDR を指定します。
- **attribute** *attribute_name* **format** *time_format* **priority** *priority_val*：EDR でキャプチャする具体的な属性を指定します。**format** *time_format*は、タイムスタンプ形式を指定します。次の時間形式をサポートしています。
  - MM/DD/YY-HH:MM:SS
  - MM/DD/YY-HH:MM:SS:sss
  - MM/DD/YYYY-HH:MM:SS
  - MM/DD/YYYY-HH:MM:SS:sss
  - YYYY/MM/DD-HH:MM:SS
  - YYYY/MM/DD-HH:MM:SS:sss
  - YYYYMMDDHHMMSS
  - YYYYMMDDHHMMSSsss
  - 秒

  priority *priority_val* は、EDR レコード内の値の位置の優先順位を指定します。値は、1 ～ 65535 の整数で指定する必要があります。

EDR モジュールを有効にするための構成

EDR モジュールを有効にするには、次の構成を使用します。

```
configure
  context context_name
    edr-module active-charging-service
    end
```

追加の TCP フィールドの設定

次の CLI コマンドを使用して EDR に追加の TCP フィールドを設定する前に、他のすべての EDR 設定が完了していることを確認します。

```
configure
  active-charging service service_name
    edr-format edr_format_name
      rule-variable tcp syn-synack-rtt priority priority_value
      rule-variable tcp synack-ack-rtt priority priority_value
      end
```

モニタリングおよびトラブルシューティング

show user-plane-service statistics rulebase name *rulebase_name*

この機能をサポートするために、次のフィールドが表示されます。

- ルールベース名
  - EDR
  - Charge Volume
    - Uplink Pkts
    - Uplink Bytes
    - Downlink Pkts
    - Downlink Bytes
  - Charging EDRs
    - Total Charging EDRs generated
    - EDRs generated for handoff
    - EDRs generated for timeout
    - EDRs generated for normal-end-signaling
    - EDRs generated for session end
    - EDRs generated for rule match
    - EDRs generated for hagr
    - EDRs generated for flow-end content-filtering
    - EDRs generated for flow-end url-blacklisting
    - EDRs generated for content-filtering
    - EDRs generated for url-blacklisting
    - EDRs generated for any-error packets
    - EDRs generated for firewall deny rule match
    - EDRs generated for transaction completion
    - EDRs generated for voip call end
    - EDRs generated for dcca failure handling

- EDRs generated for TCP optimization on
- EDRs generated for tethering signature change
- EDRs generated for interim interval
- Total Flow-Overflow EDRs
- Total zero-byte EDRs suppressed

show user-plane-service edr-format all

EDR 機能の追加の TCP フィールドをサポートするために、次のフィールドが表示されます。

- Service Name
  - EDR 形式名
    - rule-variable tcp syn-synack-rtt priority 1
    - rule-variable tcp synack-ack-rtt priority 2

ファイアウォールのサポート

表 *2:* 機能の履歴

機能名	リリース情報	説明
ファイアウォール TCP アイドル タイムアウト アクション	2023.04	ファイアウォール機能は、IP セッションベースのアクセス制御を実行するサブスクライバ通信を検査して、セキュリティ攻撃からサブスクライバを保護します。 UPF は、サブスクライバフローをドロップするか、TCP タイムアウトの期限切れ時にリセットを送信する TCP アイドル タイムアウト アクションをサポートします。

機能説明

UPF 2.0のファイアウォール機能は、サブスクライバ通信を検査し、サブスクライバセッションの IP セッションベースのアクセスコントロールｗｐ実行して、悪意のあるセキュリティ攻撃からサブスクリプションを保護します。UPF 2.0 は、サブスクライバフローをドロップする

か、TCP タイムアウトの期限切れ時にリセットを送信する TCP アイドル タイムアウト アクションをサポートします。

ファイアウォールの設定により、システムはサブスクライバ データ セッションの各パケットを検査できます。また、セキュリティ脅威を評価し、アップリンクおよびダウンリンクトラフィックに設定されたポリシーを適用します。ファイアウォールは、フローレベルごとおよびパケットごとの検証をサポートします。

ファイアウォール機能は、次の機能をサポートしています。

- DoS および DDoS 攻撃からの保護
- アプリケーションレベル ゲートウェイ
- ステートフル パケット インスペクションおよびフィルタリング
- ステートレス パケット インスペクションおよびフィルタリング
- SNMP のしきい値設定

サブスクライバ ファイアウォールの構成

ファイアウォール構成は、UPF のアクティブな課金構成で使用できます。

- Access-Ruledefs
- Firewall-NAT Policy

IPv4 および IPv6 のデフォルトファイアウォールを有効にする

次の構成を使用して、IPv4 および IPv6 のデフォルトファイアウォールを構成します。

```
configure
  active-charging service service_name
    fw-and-nat policy policy_name
      firewall policy ipv4-and-ipv6
      end
```

TCP アイドル タイムアウト アクションを構成する

TCP アイドル タイムアウト アクションを構成するには、次の構成を使用します。

```
configure
  active-charging service service_name
    fw-and-nat policy policy_name
      firewall tcp-idle-timeout-action { drop | reset }
      end
```

注：

- **firewall tcp-idle-timeout-action { drop | reset }**：TCP アイドルタイマーの期限切れ時に実行されるステートフル ファイアウォール アクションを指定します。
  - **drop**：サブスクライバフローは、TCP タイムアウトの期限切れ時にリセットを送信せずにクリアまたはドロップされます。

- **reset**：TCP タイムアウトの期限切れ時にリセットを送信するように指定します。これはデフォルト値です。

- **firewall tcp-idle-timeout-action reset** CLI コマンドは、ファイアウォールにのみ適用されます。
- UPF はフローマッピングをサポートしません。

前述のサービス構成に加えて、次の構成は、サービス内のさまざまなファイアウォール関連 CLI のデフォルト CLI 動作です。

```
Dos-Protection:
      Source-Route                          : Disabled
      Win-Nuke                              : Disabled
      Mime-Flood                            : Disabled
      FTP-Bounce                            : Disabled
      IP-Unaligned-Timestamp                : Disabled
      Seq-Number-Prediction                 : Disabled
      TCP-Window-Containment                : Disabled
      Teardrop                              : Disabled
      UDP Flooding                          : Disabled
      ICMP Flooding                         : Disabled
      SYN Flooding                          : Disabled
      Port Scan                             : Disabled
      IPv6 Extension Headers Limit          : Disabled
      IPv6 Hop By Hop Options               : Disabled
      Hop By Hop Router Alert Option        : Disabled
      Hop By Hop Jumbo Payload Option       : Disabled
      Invalid Hop By Hop Options            : Disabled
      Unknown Hop By Hop Options            : Disabled
      IPv6 Destination Options              : Disabled
      Invalid Destination Options           : Disabled
      Unknown Destination Options           : Disabled
      IPv6 Nested Fragmentation             : Disabled

    Max-Packet-Size:
      ICMP                              : 65535
      Non-ICMP                          : 65535
    Flooding:
      ICMP limit                        : 1000
      UDP limit                         : 1000
      TCP-SYN limit                     : 1000
      Sampling Interval                 : 1

    TCP-SYN Flood Intercept:
      Mode                              : None
      Max-Attempts                      : 5
      Retrans-timeout                   : 60
      Watch-timeout                     : 30
    Mime-Flood Params:
      HTTP Header-Limit                 : 16
      HTTP Max-Header-Field-Size        : 4096

    No Firewall Ruledef Match Action:
      Uplink Action                     : permit
      Downlink Action                   : deny

    TCP RST Message Threshold     : Disabled
    ICMP Dest-Unreachable Threshold    : Disabled
    Action upon receiving TCP SYN packet with ECN/CWR Flag set    : Permit
    Action upon receiving a malformed packet  : Deny
```

```
Action upon IP Reassembly Failure   : Deny
Action upon receiving an IP packet with invalid Options : Permit
Action upon receiving a TCP packet with invalid Options : Permit
Action upon receiving an ICMP packet with invalid Checksum: Deny
Action upon receiving a TCP packet with invalid Checksum: Deny
Action upon receiving an UDP packet with invalid Checksum: Deny
Action upon receiving an ICMP echo packet with id zero : Permit
TCP Stateful Checks : Enabled
First Packet Non-SYN Action: Drop
ICMP Stateful Checks: Enabled
TCP Partial Connection Timeout: 30
```

IP リアセンブル

この項では、リアセンブル機能について説明します。

- 非CUPSアーキテクチャでは、デフォルトのファイアウォール構成の場合、フラグメントは最大64Kバイトまでバッファリングされます。64Kを超えると、バッファリングされたすべてのフラグメントと後続のフラグメントがドロップされます。UPF では、最大 9K バイトのパケットサイズを最大 6 つのフラグメントにリアセンブルできます。
- **firewall ip-reassembly-failure** CLI は、ティアドロップ攻撃、ネストされたフラグメンテーション、および一般的な ip-reassembly-failure を構成します。最大 IP パケットサイズは。6 つのフラグメント（9000 バイトまで）に制限されています。
- ファイアウォール統計のカウンタを次に示します。このカウンタは、リアセンブルに関連するすべての攻撃に対して増分されます。
  - IPv4 リアセンブルの失敗が原因でドロップされたダウンリンクパケット数
  - IPv4 リアセンブルの失敗が原因でドロップされたアップリンクパケット数
  - IPv4 リアセンブル失敗時のダウンリンクドロップバイト数
  - IPv4 リアセンブル失敗時のアップリンクドロップバイト数
  - IPv6 リアセンブルの失敗が原因でドロップされたダウンリンクパケット数
  - IPv6 リアセンブルの失敗が原因でドロップされたアップリンクパケット数
  - IPv6 リアセンブル失敗時のダウンリンクドロップバイト数
  - IPv6 リアセンブル失敗時のアップリンクドロップバイト数

モニタリングおよびトラブルシューティング

この項では、この機能をサポートする show コマンドとその出力について説明します。

show config active-charging service name acs verbose

```
fw-and-nat policy SFW_NAT_TEST
      no firewall dos-protection source-router
      no firewall dos-protection winnuke
      no firewall dos-protection mime-flood
      no firewall dos-protection ftp-bounce
      no firewall dos-protection ip-unaligned-timestamp
```

```
  no firewall dos-protection tcp-window-containment
  no firewall dos-protection teardrop
  no firewall dos-protection flooding udp
  no firewall dos-protection flooding icmp
  no firewall dos-protection flooding tcp-syn
  no firewall dos-protection port-scan
  no firewall dos-protection ipv6-extension-hdrs
  no firewall dos-protection ipv6-hop-by-hop
  no firewall dos-protection ipv6-hop-by-hop router-alert
  no firewall dos-protection ipv6-hop-by-hop jumbo-payload
  no firewall dos-protection ipv6-hop-by-hop invalid-options
  no firewall dos-protection ipv6-hop-by-hop unknown-options
  no firewall dos-protection ipv6-dst-options
  no firewall dos-protection ipv6-dst-options invalid-options
  no firewall dos-protection ipv6-dst-options unknown-options
  no firewall dos-protection ipv6-frag-hdr nested-fragmentation
  no firewall dos-protection ip-sweep tcp-syn
  no firewall dos-protection ip-sweep udp
  no firewall dos-protection ip-sweep icmp
  firewall max-ip-packet-size 65535 protocol icmp
  firewall max-ip-packet-size 65535 protocol non-icmp
  firewall flooding protocol icmp packet limit 1000
  firewall flooding protocol udp packet limit 1000
  firewall flooding protocol tcp-syn packet limit 1000
  firewall flooding sampling-interval 1
  firewall tcp-syn-flood-intercept mode none
  firewall tcp-syn-flood-intercept watch-timeout 30
  firewall mime-flood http-headers-limit 16
  firewall mime-flood max-http-header-field-size 4096
  no firewall icmp-destination-unreachable-message-threshold
  access-rule no-ruledef-matches uplink action permit
  access-rule no-ruledef-matches downlink action deny
  firewall tcp-idle-timeout-action reset
  no firewall tcp-reset-message-threshold
  firewall tcp-syn-with-ecn-cwr permit
  firewall malformed-packets drop
  firewall ip-reassembly-failure drop
  no firewall validate-ip-options
  firewall tcp-options-error permit
  firewall icmp-echo-id-zero permit
  firewall icmp-checksum-error drop
  firewall tcp-checksum-error drop
  firewall udp-checksum-error drop
  firewall tcp-fsm first-packet-non-syn drop
  firewall icmp-fsm
  firewall policy ipv4-and-ipv6
  firewall tcp-partial-connection-timeout 30
  no nat policy
  no nat binding-record
  no nat pkts-drop edr-format
  no nat pkts-drop timeout
  default nat suppress-aaa-update
  nat private-ip-flow-timeout 180
  nat check-point-info basic limit-flows 100
  nat check-point-info sip-alg
  nat check-point-info h323-alg
  nat max-chunk-per-realm single-ip
#exit
```

show CLI

この項では、UPF でサポートされている show コマンドについて説明します。

- **show subscribers user-plane-only full all**

- **show subscribers user-plane-only flows**
- **show user-plane-service inline-services firewall statistics verbose**
- **show user-plane-service statistics rulebase all**
- **show alarm outstanding all**
- **show alarm outstanding all verbose**
- **show alarm statistics**
- **show user-plane-service statistics rulebase name** *rulebase_name*

SNMP トラップ

この項では、SNMP トラップについて説明します。トラップを有効にするには、ユーザープレーンでそれぞれの trap CLI を使用します。

- **Dos 攻撃**：DoS攻撃の数が設定されたしきい値を超えると、SNMP トラップが生成されます。設定された時間間隔内に数値がしきい値を下回ると、トラップがクリアされます。
- **パケットドロップ**：ドロップされたパケット数がしきい値を超えると、SNMP トラップが生成されます。設定された時間間隔内に数値がしきい値を下回ると、トラップがクリアされます。
- **拒否ルール**：拒否ルールの数がしきい値を超えると、SNMP トラップが生成されます。設定された時間間隔内に数値がしきい値を下回ると、トラップがクリアされます。
- **ルールなし**：ルールなしの数がしきい値を超えると、SNMP トラップが生成されます。設定された時間間隔内に数値がしきい値を下回ると、トラップがクリアされます。

フローアイドルタイムアウトのランダム化

セッションマネージャは、2秒ごとに、セッションマネージャインスタンスからの最新パケットの時刻、または fastpath ストリームをポーリングして、アイドルフローを判断します。短い距離のフローは、パケット数が少ないため、短時間で 5 ～ 10 秒以内でアイドル状態になります。その結果、所定の2秒のポーリングサイクルでのタイムアウトにより、多数のアイドルフローを削除する必要があります。アイドルフローの削除は、統計の調整、EDRの生成、および高速パスストリームの削除をともなうため、CPUに負荷がかかります。短期フローは積極的にクリアされるため、この機能でより多くのフローに対応できます。

ACS でフローアイドルタイムアウトのランダム化を構成する

アイドルタイムアウトフローをランダム化するには、次の構成を使用します。

```
configure
active-charging service service_name
  idle-timeout randomize-range range
    { default | no } idle-timeout randomize-range
      end
```

注：

- **idle timeout**：フローがアイドル状態を維持できる最大時間を秒単位で指定します。秒は、5～30の整数で指定する必要があります。その後、フローはランダムな値に基づいて終了します。
- **randomize-range**：期間を秒単位で指定します。適用されるアイドルタイムアウトは、フローごとに異なります。

  次に例を示します。

  ```
  idle-timeout randomize-range 20
  ```

  0～20の整数のランダム番号が生成されます。この数値は、設定されたアイドルタイムアウト値に追加され、2秒間のタイマー処理でフローがアイドルになったかどうかを確認します。設定されたアイドルタイムアウトが60秒に設定されている場合、各フローに適用される実際のタイムアウトは *60 + 20* 秒の範囲でランダム化され、フローが段階的に削除されます。
- **no**：アイドルタイムのランダム化を無効にします。このコマンドは、デフォルトでディセーブルになっています。
- **default**：idle timeout randomization コマンドをを秒単位のデフォルト設定で構成します。秒は、0～30の整数で指定する必要があります。デフォルト範囲は、0～30秒です。

  たとえば、**default idle-timeout randomize-range** は、**idle-timeout randomize-range 30** と同じです。

HTTP URL フィルタリング機能

HTTP URL フィルタリング機能は、URL 検出に使用されるルール定義を簡素化します。

HTTP リクエストパケットには、プロキシ（プレフィックス付き）URL と実際の URL を含めることができます。プロキシURLがHTTPリクエストパケットで見つかった場合、HTTP URL フィルタリング機能は解析された情報からこのURLを切り捨て、実際のURLのみがルール照合とイベントデータレコード（EDR）の生成に使用されます。

HTTP URL フィルタリング機能の設定

ここでは、HTTP URL フィルタリング機能の設定方法について説明します。

プレフィックス付き URL のグループの設定

プレフィックス付き URL のグループを設定するには、次の CLI コマンドを使用します。

```
configure
  active-charging service ecs_service_name
    group-of-prefixed-urls prefixed_urls_group_name
    end
```

プレフィックス付き URL のグループ内 URL の設定

プレフィックス付き URL のグループでフィルタリング対象の URL を設定するには、次の CLI コマンドを使用します。

```
configure
  active-charging service ecs_service_name
    group-of-prefixed-urls prefixed_urls_group_name
      prefixed-url url_1
      ...
      prefixed-url url_10
      end
```

ルールベースでのプレフィックス付き URL のグループの有効化

プレフィックス付き URL を処理するためにルールベースでプレフィックス付き URL のグループを有効にするには、次の CLI コマンドを使用します。

```
configure
  active-charging service ecs_service_name
    rulebase rulebase_name
      url-preprocessing bypass group-of-prefixed-urls
prefixed_urls_group_name_1
      ...
      url-preprocessing bypass group-of-prefixed-urls
prefixed_urls_group_name_64
      end
```

コントロール プレーン シャーシのこの設定は、「group-of-prefixed-urls」と「rulebase-url-preprocessing」の PFD メッセージを使用してユーザープレーンにプッシュされます。

プレフィックス付き URL のグループにはプロキシ URL のリストがあり、このリストは削除する必要があります。rulebase には、プレフィックス付き URL の複数のグループが含まれており、フィルタリングする必要があります。課金 ruledef には、プレフィックス付き URL グループ内の URL を削除してから検索する必要がある実際の URL のルールが含まれています。

（注）
- プレフィックス付き URL の 1 グループあたり、最大 10 個のプレフィックス付き URL を追加できます。
- 最大 64 のプレフィックス付き URL グループを作成および設定できます。

Show コマンド

show user-plane-service group-of-prefixed-urls all | name *group_name*

この show コマンドをユーザープレーンで使用すると、プレフィックス付き URL のグループがプッシュされているかどうかを確認できます。このコマンドの出力は次のとおりです。

- Name of the group of prefixed URLs
- Prefixed URLs

- Total number of prefixed URLs found

show user-plane-service rulebase name *rbase_name*

この show コマンドをユーザープレーンで使用すると、プレフィックス付き URL のグループが rulebase で設定されているかどうかを確認できます。このコマンドの出力は次のとおりです。

- Name of rulebase
- Name of the groups of prefixed Urls for URL pre-processing

show user-plane-service statistics analyzer name http

このコマンドの出力は次のとおりです。

- Total HTTP Sessions
- Current HTTP Sessions
- Total Uplink Bytes
- Total Downlink Bytes
- Total Uplink Pkts
- Total Downlink Pkts
- Uplink Bytes Retrans
- Downlink Bytes Retrans
- Uplink Pkts Retrans
- Downlink Pkts Retrans
- Total Request Succeed
- Total Request Failed
- GET Requests
- POST Requests
- CONNECT Requests
- PUT requests
- HEAD requests
- Websocket Flows
- Invalid packets
- Wrong FSM packets
- Unknown request method
- Pipeline overflow requests
- Corrupt request packets
- Corrupt response packets

- Unhandled request packets
- Unhandled response packets
- Partial HTTP Header Anomaly prevented
- New requests on closed connection
- Memory allocation failures
- Packets after permanent failure
- Prefixed Urls Bypassed
- FastPath Statistics
- Total FP Flows
- Uplink (Total FP Pkts)
- Downlink (Total FP Pkts)
- Uplink (Total FP Bytes)
- Downlink (Total FP Bytes)



（注） パフォーマンス測定指標として、[Prefixed URLs Bypassed] カウンタが http アナライザ統計に追加されました。削除済みのプレフィックス付き URL 数を表示します。

IP 再アドレス指定

IP アドレス再指定機能により、パケットの宛先 IP アドレスに基づいて、不明なゲートウェイ通信を既知のゲートウェイや信頼できるゲートウェイにリダイレクトできます。

IP アドレス再指定は、課金アクションで定義されたフローアクションで設定されます。IP アドレス再指定は、特定の ruledef に一致する通信に対して機能し、そのため課金アクションが適用されます。IP アドレス再指定は、アップリンクとダウンリンク通信の両方に適用されます。拡張された課金サブシステムでは、アップリンクパケットは、パケット検査やルール照合などの後に変更され、接続先 IP またはポートが決定され、送信される直前にアドレス再指定された IP またはポートに置き換えられます。ダウンリンクパケット（アドレス再指定された IP またはポートを含む）は、受信した時点でパケットインスペクションの前に変更され、送信元 IP またはポートが、元のサーバー IP またはポート番号に置き換えられます。

MS からの 1 つのフローで、1 つのパケットのアドレスが再指定されると、そのフローに含まれるすべてのパケットのアドレスが同じサーバーに再指定されます。DPI や rule-matching などの機能は影響を受けません。各 IP アドレスとポートの組み合わせは、ruledef として定義されます。

IP フラグメンテーションの場合、IP リアセンブルに成功したパケットのアドレスが再指定されます。ただし、IP フラグメンテーションに失敗したパケットのアドレスは、再指定されません。

server-list が charging-action で構成された場合、アドレス再指定サーバーを選択するために、2 つの異なるアプローチがあります。

- **round-robin**：round-robin アプローチでは、サーバーは、新しい各フローで round-robin 方式で選択されます。round-robin では、リスト内のアクティブなサーバーのみが選択対象と見なされます。
- **階層型アプローチ**：階層型アプローチでは、アドレス再指定 server-list で定義されている順に応じて、サーバーがプライマリ、セカンダリ、ターシャリなどでタグ付けされます。すべてのフローは、プライマリサーバーが稼働状態になるまで、プライマリサーバーにアドレス再指定されます。プライマリサーバーがダウンした場合は、フローのアドレス再指定先はセカンダリサーバーとなり、その後も同じロジックが繰り返されます。プライマリサーバーが再びアクティブになると、フローのアドレス再指定先はプライマリサーバーに戻ります。

ユーザーがサーバーの選択に階層型か round-robin アプローチを指定できるその他の CLI が提供されます。round-robin および階層型のサーバー選択アプローチはどちらも、IPv4 ベースおよび IPv6 ベースのサーバーの両方に適用できます。

IP アドレス再指定を構成する

パケットの宛先 IP アドレスに基づいてパケットをアドレス再指定することで、道のゲートウェイ通信から既知または信頼されているゲートウェイにリダイレクトできます。これは、課金アクションでサーバーのアドレス再指定を構成することによって実装されます。

IP アドレス再指定機能を構成するには、次の構成を使用します。

```
configure
  active-charging service  acs_service_name
    charging-action  charging_action_name
      flow action readdress server ipv4_address/ipv6_address [
discard-on-failure ]  [ dns-proxy-bypass ]  [ port port_number [
discard-on-failure ] [ dns-proxy-bypass ] ]
      end
```

アドレス再指定 server-list が charging-action で定義されている場合に、IP アドレス再指定機能を構成するには、次の構成を使用します。

```
configure
  active-charging service  acs_service_name
    charging-action  charging_action_name
      flow action readdress server-list server_list_name [ hierarchy ] [
 round-robin ] [ dns-proxy-bypass ] [ discard-on-failure ]
      end
```

次に、server-list の構成例を示します。

```
readdress-server-list DRE
consecutive-failures 1
response-timeout 10000
server 209.165.200.225 port 53
server 209.165.200.226 port 53
server 2001:420:54fe::1019 port 53
server 2001:420:54fe::1039 port 53
```

```
server 2001:420:54fe::1049 port 53
server 2001:420:54fe::1059 port 53
server 209.165.201.30 port 8080
#exit
```

（注） リストには、最大 10 のサーバーを構成でき、active-charging サービスには、最大 10 のリストを構成できます。

Show コマンド

この項では、IP のアドレス再指定機能のサポートに使用可能な show CLI コマンドについて説明します。

CLI コマンド	説明
show user-plane-service readdress-server-list statistics all **show user-plane-service readdress-server-list statistics instance** *instance* **show user-plane-service readdress-server-list statistics name** *name*	これらの show コマンドを使用して、リアドレスサーバーリストの統計を表示します。
show user-plane-service readdress-server-list statistics	

clear user-plane-service readdress-server-list statistics all CLI を使用して、再アドレスサーバーリスト統計をクリアします。

show user-plane-service statistics charging-action all CLIを使用して、「再アドレスされた」カウンタを確認します。

L7 プロトコル

次の L7 プロトコルが DPI の一部としてサポートされています。

- DNS
- FTP
- HTTP
- HTTPS
- RTP/RTSP
- SIP

DNS

UPF は、L7 アナライザの一部として DNSプロトコルをサポートします。

FTP

UPF 2.0 は、L7 アナライザの一部として FTP プロトコルをサポートします。

HTTP

HTTP リクエストまたは HTTP 応答が完了すると、次の場合に、アップリンクまたはダウンリンクのデータパケットが、VPP にオフロードされます。

- コンテンツ長：ボリュームベースのオフロードは、GETやPOSTなどのメソッドでサポートされます。チャンクエンコーディングによるデータ転送メカニズムを使用したHTTPフローは、HTTP で定義されているメソッドに関係なくオフロードされません。ストリームがコンテンツ長に基づいてオフロードされた場合、もう一方のストリームも、前者がオンロードされなくなるまでオフロードされます。
- CONNECTメソッド：フローがCONNECTにアップグレードされると、アップリンクとダウンリンクの両方のストリームがオフロードされるメソッドです。
- WebSocket メソッド：フローが WebSocket プロトコルとして分類されると、アップリンクとダウンリンクの両方のストリームがオフロードされます。
- ストリームは、次のいずれかの場合に再度、オンロードされます。
  - FIN パケットを受信した場合
  - コンテンツ長に違反している場合
  - PDN の更新

ヘッダー解析

rulebaseに含まれるruledefで定義されているヘッダーフィールドのみが解析されます。X-Header機能の場合は、一部のHTTPヘッダーフィールドと依存関係があるリダイレクトが設定されます。

HTTP 課金

- 完全なパケットは、課金されます。
- 部分的パケットは、完成時に課金されます。部分的パケットを完成させるパケットも、課金されます。
- 連結パケットは、課金されます。
- 遅延課金が有効になっている場合：制御パケットは、遅延課金CLI構成に応じて、アプリケーションベースのルールに対して課金されます。

- 応答ベースの課金が有効になっている場合：HTTP リクエストの応答を受信した後、その HTTP リクエストは応答ルールの CA に対して課金されます。

X-Header の解析とルール照合

x-header ルール行が含まれる ruledef が解析され、照合されます。

WebSocket

HTTP フローが WebSocket フローにアップグレードされた場合、HTTP GET リクエストの後のフローの後続のパケットの課金が関与します。

応答ベースの TRM

トランザクションルールの照合は、HTTP 応答パケットの受信後に行われます。

X-Header の挿入

HTTP リクエストへの X-Header の挿入がサポートされます。

URL ベースのリダイレクト

URL ベースのリダイレクションは、次を実行する UPF 機能です。

- サブスクライバにリダイレクト応答を返し、既存の TCP 接続を終了します。
- サブスクライバの Web ブラウザをトリガーして、元の HTTP パケットを指定された接続先 URL に自動送信します。
- ダイナミックポリシー経由で受信したコンポジット文字列（returnURL など）から特定の接続先 URL を抽出できるようにします。

通信フローが HTTP でない場合、UPF は、リダイレクト URL オプションを無視して、パケットを通常どおり転送します。リダイレクションは、アップリンク方向に送信される特定の HTTP パケット（GET 要求など）についてのみサポートされます。

表 *3*：機能の履歴

機能名	リリース情報	説明
リダイレクト URL の暗号化	2024.01	UPF は、Blowfish および AES 暗号化方式を使用したダイナミックフィールドの URL 暗号化をサポートしています。 デフォルト設定：無効：構成が必要

機能名	リリース情報	説明
リダイレクトの書き換えと TCP FIN/RST	2026.01.1	N7 インターフェイスの ADC ダイナミックルールは、リダイレクト URL サポートを実行します。これは、AES 暗号化でのみサポートされます。

ダイナミックフィールドと暗号化：

- **フィールド挿入**： UPF は、リダイレクトされた URL へのダイナミック サブスクライバ データの追加をサポートしています。
- **セキュリティ**：機密性の高いダイナミックフィールドを、対称ブロック暗号を使用して保護できます。
  - **Blowfish**：暗号化と復号に 128 ビットキーを使用します。
  - **AES**：256 ビットキーによる暗号ブロック連鎖（CBC）モードを使用します。UPF は、AES 暗号化にハードコードされた 10 ラウンドの値を使用します。

（注） N7 インターフェイスのリダイレクト URLは、AES 暗号化でのみサポートされます。

N7 インターフェイスを使用した URL リダイレクションの制限事項

- この機能は、UL L7 パケットのみをサポートします。機能要件は、特に UL L7 のサポート向けであるため、DL L7 パケットはサポートしません。
- 既存の課金アクションの設定は変更しないでください。

リダイレクションでサポートされるダイナミックフィールド

UPF は、リダイレクトされた URL に挿入するための次のダイナミックフィールドをサポートしています。アスタリスク（*）付きのフィールドは、暗号化をサポートします。

- #ACSMGR_BEARER_CALLED_STATION_ID#
- #BEARER.IMSI# *
- #BEARER.MSISDN# *
- #HTTP.HOST#
- #HTTP.URL#
- #HTTP.URI#
- #RTSP.URI#

- #RULEBASE#

UPF は、N7 インターフェイスでリダイレクトされた URL に挿入するための次のダイナミックフィールドをサポートしています。

- #BEARER.IMSI# *
- #HTTP.URL#

UPF のリダイレクションと終了の仕組み

UPF は、service-scheme フレームワークを使用して具体的なポリシーイベントを検出します。次に、SMF から N4 インターフェイス経由で受信した ADC ダイナミックルールに基づいて、リダイレクションまたはフロー終了を実行します。

process_summary

このプロセスに関与する主要なコンポーネントは次のとおりです。

- PCF：複合リダイレクト URL またはブロック指示を含むダイナミックポリシーを提供します。
- SMF：N4 インターフェイス経由で UPF にポリシー情報をリレーします。
- UPF：リダイレクトと暗号化を実行し、ダイナミックデータを挿入し、ブロックされた通信の TCP シグナリング（FIN/RST）を管理します。

process_workflow

N7 インターフェイスによるリダイレクトまたは終了を有効にするワークフロー：

1. **service-scheme トリガー**：service-scheme フレームワークは、セッションの確立または変更中に far-received または pdr-received イベントを検出します。
2. **条件照合**：システムは、リダイレクト URL に returnURL や構成に基づいた具体的なキーワードが含まれているかどうかなどをチェックするなど、trigger-conditions を評価します。
3. **アクションの実行**：UPF は、セッションの確立または変更中に発生した adc ルールに基づいて特定の charging-action を指す、マッピングされた trigger-action を実行します。
4. **URL 処理**：リダイレクトの場合、UPF は実際の URL を抽出し、ダイナミックフィールドに入力し、暗号化を適用します。
5. **フロー管理**：終了の場合、UPF はTCP FIN を Web クライアントに送信します。また、TCP RST を Web サーバーに送信して、接続をグレースフルに終了します。

URL 暗号化を構成する

リダイレクトされた URL のダイナミックフィールドの暗号化を有効にするには、次の構成例を使用します。

```
configure
  active-charging service service_name
    charging-action charging_action_name
      flow action redirect-url redirect_url [ encryption { blowfish128
| aes256 } [ encrypted ] key key ]
    end
```

注：

- **flow action redirect-url** *redirect_url*：リダイレクト URL を指定します。最大 16 のダイナミックフィールドを含めることができます。*redirect_url* は、1 ～ 511 文字の英数字で指定する必要があります。
- **encryption { blowfish128 | aes256 } [encrypted] key** *key*：リダイレクト URL のダイナミックフィールドの暗号化を有効にする場合に指定します。
  - **blowfish128**：ダイナミックフィールドの暗号化に 128 ビットキーで Blowfish 暗号化を使用するために指定します。
  - **aes256**：ダイナミックフィールドの暗号化に 256 ビットキーで AES-CBC 暗号化を使用するように指定します。
  - **encrypted**：キーを暗号化する場合に指定します。
  - **key** *key*：ダイナミックフィールドの暗号化に使用するキーを指定します。*key* は、1 ～ 523 文字の英数字で指定する必要があります。
- リダイレクト先の URL でエンコーディングが有効になっている場合、暗号化された URL は、最大 2300 文字をサポートします。この長さを超えると、2301 文字以降の URL は切り捨てられます。

N7 ベースのリダイレクト フロー アクションを構成する

N7 インターフェイスを使用してダイナミックリダイレクト URL を処理し、暗号化を適用するように UPF を設定します。

Before you begin

- L7 アナライザがアクティブであることを確認します（HTTPS のサポートに必要）。
- AES 暗号化を使用する場合は、キーフレーズ（英数字、1 ～ 523 文字）を指定します。
- リダイレクト URL 列が 1 ～ 511 文字であることを確認します。

リダイレクト フロー アクションを構成するには、次の手順を実行します。

手順

ステップ 1 URL ID の ruledef を定義します。

例：

```
[upf] # ruledef rule_name
 [upf] #  http url contains test.html
 [upf] #  http url !contains abc.html
 [upf] #  multi-line-or all-lines
```

ステップ 2 リダイレクト URL の trigger-conditions を構成します。

例：

```
[upf] # trigger-condition TC1
 [upf] #  redirect-url contains "returnURL="
 [upf] #  redirect-url starts-with "https://snap"
```

ステップ 3 リダイレクションの trigger-actions を定義します。

例：

```
[upf] # trigger-action TA1
    [upf] #  execute-charging-action url_redirect_ca
```

ステップ 4 暗号化パラメータを使用して charging-action を作成します。

例：

```
[upf] # charging-action url_redirect_ca
[upf] #  flow action redirect-url <return url recieved over n4 interface with ADC rule>
```

ステップ 5 トリガーを service-scheme にバインドします。

例：

```
[upf] # service-scheme SS1
 [upf] #  trigger far-received
   [upf] #  priority 1 trigger-condition TC1 trigger-action TA1
```

終了フローアクションを構成する

TCP シグナリングを使用して通信ブロックを実装します。

Before you begin

フロー終了を構成するには、次の手順を実行します。

手順

ステップ 1 通信ブロックの ruledef を定義します。

例：

```
[upf] # ruledef terminate_rule
  [upf] #  ip any-match = TRUE
```

ステップ 2 終了の trigger-conditions を設定します。

例：

```
[upf] # trigger-condition TC2
  [upf] # tdf-appid contains <terminate app id recived with ADC dynamic rule over N4 interface>
```

ステップ 3 終了の trigger-actions を定義します。

例：

```
[upf] # trigger-action TA2
   [upf] #   execute-charging-action terminate_all
```

ステップ 4 フロー終了の charging-action を作成します。

例：

```
[upf] # charging-action terminate_all
   [upf] #  flow action terminate-flow
```

ステップ 5 トリガーを service-scheme にバインドします。

例：

```
[upf] # service-scheme SS1
[upf] #   trigger pdr-received
[upf] #   priority 2 trigger-condition TC2 trigger-action TA2
```

フロー統計のリダイレクトと終了

次のコマンドを使用して、機能の状態を確認し、リダイレクトまたは終了したフローの統計を表示できます。

サブスクライバ FAR の詳細

抽出されたリダイレクトURLと具体的なコールに適用された課金アクションを表示するには、次のシンタックスを使用します。

例：

```
[upf] # show subscribers user-plane-only callid [call_id] far full all
```

サンプル出力：

```
Callid: 00004e29
Interface Type: Sxb
 FAR-ID       : 0x0003
 Destination Interface : Core
 Apply Action     : FORWARD
 Outer Header Creation   :
 Remote TEID     : 0x0
 Transport Level Marking    : N/A
 Transport Level Marking Options :
   Copy Inner:                    No
   Copy Outer:                    No
 Inner Packet Marking           : N/A
 Layer2 Marking                 : N/A
 Remote IP Address   :
 Remote Port    : N/A
 Number of Associated PDRs: 1
 Linked Traffic Endpoint : 0x2
 Redirect Info  :
Redirect Address Type  : URL
Redirect Address Value :
https://snap.mobile.att.net/?nonce=ApolloHONDA&returnURL=https://myvehicle.att.com/%23/start%3Foem=honda
```

```
       Charging Action Name           : url_redirect_ca
```

トリガーアクションの構成

具体的なトリガーアクションにマッピングされている課金アクションを確認するには、次のシンタックスを使用します。

例：

```
[upf] # show user-plane-service trigger-action all
```

サンプル出力：

```
Trigger-Action: TA1
       HTTP Response Based TRM      : none
       HTTP Response Based Charging : none
       Throttle Suppress            : Disabled
       Flow Recovery                : Disabled
       Traffic Optimization         : Disabled
       Step Up GBR                  : Disabled
       Step Down GBR                : Disabled
       TCP Acceleration             : Disabled
       TCP Acceleration Threshold   : Disabled
       Service-Chain                : none
       UP-Service-Chain             : none
       Flow-IP-Readdressing         : Disabled
       EDNS-Encode                  : Disabled
       Charging-Action-Map          : none
       Charging-Action-Name         : url_redirect_ca

Trigger-Action: TA2
       HTTP Response Based TRM      : none
       HTTP Response Based Charging : none
       Throttle Suppress            : Disabled
       Flow Recovery                : Disabled
       Traffic Optimization         : Disabled
       Step Up GBR                  : Disabled
       Step Down GBR                : Disabled
       TCP Acceleration             : Disabled
       TCP Acceleration Threshold   : Disabled
       Service-Chain                : none
       UP-Service-Chain             : none
       Flow-IP-Readdressing         : Disabled
       EDNS-Encode                  : Disabled
       Charging-Action-Map          : none
       Charging-Action-Name         : terminate_all
```

トリガー条件の詳細

課金アクションの選択に使用する条件を確認するには、次のシンタックスを使用します。

例：

```
[upf] # show user-plane-service trigger-condition all
```

サンプル出力：

```
Trigger-Condition: TC1
       redirect-url  contains returnURL=
       redirect-url  starts-with https://snap
       redirect-url  contains nonce=Name
       Multi-line-OR All lines : Disabled
```

```
Trigger-Condition: TC2
       Tdf-AppId  contains terminate_
       Multi-line-OR All lines : Disabled
```

グローバルリダイレクト統計

リダイレクトされたフローのグローバル数を表示するには、次のシンタックスを使用します。

例：

```
[upf] # show user-plane-service statistics all
```

サンプル出力：

```
ADC Redirect Stats:
    ADC Redirected Flows         : 3
```

終了したフロー数

具体的なコールに対して終了したフローの数を表示するには、次のシンタックスを使用します。

例：

```
[upf] # show subscribers user-plane-only full callid <call_id>
```

サンプル出力：

```
Flow Action Terminated Flows: 5
```

PDR 終了の詳細

フローの終了中に一致するパケット数と適用される課金アクションを表示するには、次のシンタックスを使用します。

例：

```
[upf] # show subscribers user-plane-only callid <call_id> pdr full all
```

サンプル出力：

```
Callid: 12345
Interface Type: Sxb
IP address: 11.11.0.1
PDR-ID:                   0x0003
Rule Name:               dynamicfacebook
Charging Action Name:      terminate_all
TDF App Id:   terminate_rule
TDF Notifications:   Enabled
Hits:                     5 Match Bypassed:            0
Matched Bytes:          285 Matched Packets:           5
```

HTTPS

UPF は、L7 アナライザの一部として HTTPS プロトコルをサポートします。

RTP/RTSP

UPF 2.0 は、L7 アナライザの一部として RTP および RTSP プロトコルをサポートします。

SIP

Session Initiation Protocol は、IETF で定義されているシグナリングプロトコルです。Internet Protocol（IP）を介した音声やビデオ通話などの通信セッションの制御に広く使用されています。このプロトコルは、2 者（ユニキャスト）またはマルチパーティ（マルチキャスト）セッションの作成、変更、および終了に使用できます。セッションは、1 つまたは複数のメディアストリームで構成されます。

UPF 2.0 は、L7 アナライザの一部として SIP をサポートしています。

モニタリングおよびトラブルシューティング

この項では、機能のモニタリングとトラブルシューティングのサポートに使用できる CLI コマンドに関する情報を提供します。

show コマンドや出力

この項では、この機能のサポートにおける show コマンドまたはその出力について説明します。

DNS

DNS 関連の統計を取得するには、次の CLI コマンドを使用します。

show user-plane-service statistics analyzer name dns

FTP

FTP 関連の統計を取得するには、次の CLI コマンドを使用します。

show user-plane-service statistics analyzer name ftp

HTTP

HTTPS 関連の統計を取得するには、次の CLI コマンドを使用します。

show user-plane-service statistics analyzer name http

HTTPS

HTTPS 関連の統計を取得するには、次の CLI コマンドを使用します。

show user-plane-service statistics analyzer name secure-http

RTP

RTP 関連の統計を取得するには、次の CLI コマンドを使用します。

show user-plane-service statistics analyzer name rtp

RTSP

RTSP に関連する統計情報を取得するには、次の CLI コマンドを使用します。

- **show user-plane-service statistics analyzer name rtsp**

- **show user-plane-service statistics analyzer name rtsp verbose**

SIP

SIP 関連の統計を取得するには、次の CLI コマンドを使用します。

show user-plane-service statistics analyzer name sip

N4 での NAT サポート

表 *4:* 機能の履歴

機能名	リリース情報	説明
N4インターフェイスでのNATサポート	2023.04	ネットワーク アドレス変換（NAT）機能は、ルーティング不可のプライベート IP アドレスをパブリック IP アドレスのプールからのルーティング可能なパブリック IP アドレスに変換します。 UPFFは、ネットワークアドレスを構成するN4インターフェイスの NAT をサポートし、NAT バインディングレコードを N4 に送信します。NAT ポリシーと NAT パブリック IP アドレスの IP プールは、N4 の UPF で構成されます。

機能説明

ネットワーク アドレス変換（NAT）機能は、ルーティング不可のプライベート IP アドレスをパブリック IP アドレスのプールからのルーティング可能なパブリック IP アドレスに変換します。この機能により、ユーザーは外部ネットワークとの通信に必要なパブリックIPアドレスの数を節約できます。この機能により、内部ネットワーク向けIPアドレススキーマとしてセキュリティを確保でき、外部ホストからマスキングされ、各送受信パケットがこの変換プロセスを通過します。

UPF は、N4 インストレーションで NAT をサポートし、ネットワークアドレスを構成します。NAT IP アドレスと NAT ポートを使用して、UE の送信元 IP または送信元ポートアドレスをカプセル化したデータパケットを自動的に転送するようにシステムを構成できます。

サポートされる NAT の組み合わせは次のとおりです。

- NAT44 オンデマンド多対 1

- NAT44 オンデマンド 1 対 1
- NAT64 オンデマンド多対 1
- NAT 64 オンデマンド 1 対 1
- NAT44 非オンデマンド多対 1
- NAT44 非オンデマンド 1 対 1
- NAT64 非オンデマンド多対 1
- NAT64 非オンデマンド 1 対 1

NAT の補足情報については、StarOS の『*NAT* アドミニストレーション ガイド』を参照してください。

注：StarOS の『*NAT* アドミニストレーション ガイド』に記載されているすべての機能が UPF に当てはまるわけではありません。

制限事項

N4 インターフェイスの NAT のサポートには次の制限事項があります。

- 多対 1 およびオンデマンドモードの NAT44 のみがサポートされます。
- すべての NAT プールは、接続先コンテキストの個別のユーザープレーンで設定されます。
- fw-and-nat ポリシーでの CLI アクション拒否を使用した課金アクション、および active-charging-service での flow-any-error 課金アクションはサポートされていません。
- 外部サーバーからの「dynamic-only」および「static-and-dynamic」ルールを使用して構成された access-rules はサポートされません。
- 同じレルムからの複数の IP はサポートされません。
- NAT プールでのネクストホップ転送はサポートされません。
- NAT プールのポート範囲はサポートされません。
- プライベート IP チェック CLI のスキップはされません。
- RADIUS および Gy で返される Fw-and-nat ポリシーベースの NAT ポリシーの適用はサポートされません。
- ベアラー固有のフィルタは、access-ruledef ではサポートされません。
- Access-rules は、fw-and-nat ポリシーでの open-port ポート範囲設定のトリガーをサポートしません。
- SR/ICSR 後の NAT ポートリカバリ（fw-and-nat アクション）はサポートされません。
- active-charging サービスでは、NAT reassembly timeout CLI はサポートされません。代わりに、UP の汎用コンテキストレベル CLI を使用する必要があります。
- NAT フラグメンテーション リアセンブルのエラーはサポートされません。

- NAT flow-mapping タイマーはサポートされません。
- N:M 冗化性の場合、各 UP ホストのインターフェイス設定の一部として RCM から設定される NAT IP プール、およびプール名は、すべてのアクティブなユーザープレーンで一意である必要があります。fw-and-nat ポリシーで参照される同じ NAT レルムをすべてのユーザープレーンに適用できるように、すべてのプールに NAT グループを使用することが必須になります。
- N:M 冗長化の場合、RCM 経由ですべての UP でまとめて設定される NAT IP プールの総数は、IP プールの最大制限数（2,000）に従う必要があります。すべてのアクティブ UP の累積合計が最大値を超えると、スタンバイユーザープレーンの設定は失敗します。

NAT の設定

NAT ポリシー（Firewall-and-NAT ポリシー）と access-ruledef 構成は、UPF の active-charging サービスで実行されます。

NAT 関連の CLI コマンドの詳細については、『*StarOS NAT* アドミニストレーション ガイド』の「*NAT* 構成」章を参照してください。

（注） 『*StarOS NAT* アドミニストレーション ガイド』の「*NAT* 構成」章に記載されているすべての CLI コマンドと構成が UPF に適用できるわけではありません。

設定例

ユーザープレーン

ISP コンテキストのユーザープレーンでは、次のプール関連の設定が必要です。

```
configure
   context ISP1-UP
      ip pool NAT44_PUBLIC1 209.165.200.225 255.255.255.224 napt-users-per-ip-address 2
 on-demand port-chunk-size 16 max-chunks-per-user 4 group-name NAT44_GRP1
      ip pool NAT44_PUBLIC2 209.165.200.226 255.255.255.224 napt-users-per-ip-address 2
 on-demand port-chunk-size 16 max-chunks-per-user 4 group-name NAT44_GRP1
      ip pool NAT44_PUBLIC3 209.165.200.227 255.255.255.224 napt-users-per-ip-address 2
 on-demand port-chunk-size 8 max-chunks-per-user 1 group-name NAT44_GRP2
      ip pool NAT44_PUBLIC4 209.165.200.228 255.255.255.224 napt-users-per-ip-address 4
 on-demand port-chunk-size 32256 max-chunks-per-user 4 group-name NAT44_GRP2
      ip pool NAT44_PUBLIC5 209.165.200.229 255.255.255.224 napt-users-per-ip-address
8064 on-demand port-chunk-size 8 max-chunks-per-user 2
      end
```

さまざまな NAT プールタイプの NAT プール関連の設定例

```
1-1 on-demand:
----------------------------
config
context ISP1-UP
ip pool NAT44_ipv4_1_1 209.165.200.230 255.255.255.224 nat-one-to-one on-demand
```

```
nat-binding-timer 60
end

N-1 Not-on-demand:
------------------------
config
context ISP1-UP
ip pool NAT44_ipv4_N_1 209.165.200.231 255.255.255.224 napt-users-per-ip-address 2
max-chunks-per-user 2 port-chunk-size 8
end

1-1 Not-on-demand:
------------------------
config
context ISP1-UP
ip pool NAT44_ipv4_NOD_1_1 209.165.200.232 255.255.255.224 nat-one-to-one
end
```

Note コントロールプレーンの構成を追加して、1 つ以上のアクセスルール定義をユーザープレーンで設定された必要な NAT プールまたはグループにマッピングします。

モニタリングおよびトラブルシューティング

NAT 統計の収集

次の表に、NAT 統計の収集に使用できるコマンドを示します。

最初の列には収集する統計、2 つ目の列には使用するコマンドを挙げています。

統計/情報	show コマンド
アクティブまたは休止中のセッションがある、現在のすべてのサブスクライバに関する情報。サブスクライバに関連付けられている IP アドレスを確認します。NAT レルムで使用されているすべての IP アドレスも表示されます。	**show subscribers user-plane-only full all**
NAT サブシステムの統計情報。	**show user-plane-service statistics all**
NAT 関連のすべての統計。	**show user-plane-service statistics nat all**
NAT レルム関連のすべての統計。	**show user-plane-service statistics nat nat-realm all**
NAT IP プールグループ内のすべての NAT IP プールの統計。	**show user-plane-service statistics nat nat-realm** *pool_name*
生成された NAT バインドレコードに関する情報。	**show user-plane-service edr-format statistics all**
UP の APN で fw-and-nat ポリシーの関連付けを確認します。	**show user-plane-service pdn-instance name** *name*

統計/情報	show コマンド
UP での fw-an-nat ポリシーの設定を確認します。	**show user-plane-service fw-and-nat policy all**
ポートチャンクの割り当ておよび解放のために生成された NAT バインドレコードに関する情報。	**show user-plane-service rulebase name** *name*
アクセス ruledef に関する情報。	**show user-plane-service ruledef all**
UP の rulebase で fw-and-nat ポリシーの関連付けを確認します。	**show user-plane-service rulebase name** *name*

clear コマンド

この機能をサポートする、次の clear CLI コマンドを使用できます。

- **clear user-plane-service statistics nat nat-realm all**
- **clear user-plane-service statistics nat all**

NAT パラメータしきい値の SNMP トラップ

NAT パラメータしきい値に対する次の SNMP トラップがサポートされます。

SNMP トラップ	説明
ThreshNATPortChunks	NAT ポートのチャンク使用率が、しきい値により設定された限度に達すると生成されます。
ThreshClearNATPortChunks	NAT ポートのチャンク使用率が、クリアしきい値により設定された限度に達すると生成されます。
ThreshNATPktDrop	NAT パケットドロップが、しきい値により設定された限度に達すると生成されます。
ThreshClearNATPktDrop	NAT パケットドロップが、クリアしきい値により設定された限度に達すると生成されます。
ThreshIPPoolUsed	IP プールで使用されている IP の数が、しきい値により設定された限度に達すると生成されます。
ThreshClearIPPoolUsed	IP プールで使用されている IP の数が、クリアしきい値により設定された限度に達すると生成されます。
ThreshIPPoolFree	IP プールが解放され、しきい値が定める限度に達すると生成されます。
ThreshClearIPPoolFree	IP プールが使用され、クリアしきい値が定める限度に達すると生成されます。

SNMP トラップ	説明
ThreshIPPoolAvail	IP プールが次のフローで使用可能になり、設定されたしきい値に達すると生成されます。
ThreshClearIPPoolAvail	IP プールが使用され、設定されたしきい値に達すると生成されます。

注：これらのトラップを有効にするには、それぞれのCLIをユーザープレーンで設定する必要があります。

バルク統計

NAT レルムスキーマ

NAT レルムはユーザープレーンで設定され、統計情報はコンテキストごと、レルムごとに保存されます。これらの統計変数（累積とスナップショットの両方）は、NAT レルムスキーマで使用できます。

***Table 5: NAT* レルムスキーマ**

変数名	データ型	カウンタタイプ	説明
Vpnname	文字列	情報	コンテキスト名
Realmname	文字列	情報	レルム名。
nat-rlm-bind-updates	Int64	カウンター	送信された暫定AAA NBUの合計。
nat-rlm-bytes-txferred	Int64	カウンター	レルムによって転送されたNAT44およびNAT64バイトの合計数（アップリンク+ダウンリンク）。
nat-rlm-bytes-nat44-tx	Int64	カウンター	レルムによって転送されたNAT44バイトの合計数。
nat-rlm-bytes-nat64-tx	Int64	カウンター	レルムによって転送されたNAT64バイトの合計数。
nat-rlm-ip-flows	Int64	カウンター	レルムで使用されたNAT44およびNAT64フローの総数。
nat-rlm-nat44-flows	Int64	カウンター	レルムによって処理されたNAT44フローの総数。
nat-rlm-nat64-flows	Int64	カウンター	レルムによって処理されたNAT64フローの総数。

変数名	データ型	カウンタタイプ	説明
nat-rlm-ip-denied	Int32	カウンター	NAT IP アドレスが拒否された NAT44 および NAT64 フローの総数。
nat-rlm-ip-denied-nat44	Int64	カウンター	IP が拒否された NAT44 フローの総数。
nat-rlm-ip-denied-nat64	Int64	カウンター	IP が拒否された NAT64 フローの総数。
nat-rlm-port-denied	Int32	カウンター	ポートが拒否された NAT44 および NAT64 フローの総数。
nat-rlm-port-denied-nat44	Int64	カウンター	ポートが拒否された NAT44 フローの総数。
nat-rlm-port-denied-nat64	Int64	カウンター	ポートが拒否された NAT64 フローの総数。
nat-rlm-memory-denied	Int64	カウンター	メモリが拒否された NAT44 および NAT64 フローの総数。
nat-rlm-memory-denied-nat44	Int64	カウンター	メモリが拒否された NAT44 フローの総数。
nat-rlm-memory-denied-nat64	Int64	カウンター	メモリが拒否された NAT64 フローの総数。
nat-rlm-ttl-ips	Int32	ゲージ	NAT レルムあたりのコンテキストごとの NAT パブリック IP アドレスの総数。スタティック値です。
nat-rlm-ips-in-use	Int32	ゲージ	NAT レルムあたりのコンテキストごとに、現在使用されている NAT IP アドレスの総数。
nat-rlm-current-users	Int32	ゲージ	NAT レルムを現在使用しているサブスクライバの総数。
nat-rlm-ttl-port-chunks	Int32	ゲージ	NAT レルムあたりのコンテキストごとのポートチャンクの総数。スタティック値です。
nat-rlm-chunks-in-use	Int32	ゲージ	NAT レルムあたりのコンテキストごとに現在使用されているポートチャンクの総数。

変数名	データ型	カウンタタイプ	説明
nat-rlm-port-chunk-size	Int32	ゲージ	NAT レルムのポートチャンクのサイズ。
nat-rlm-port-chunk-average-usage-tcp	Int32	ゲージ	割り当てられた TCP ポートの平均 TCP ポート使用率。つまり、割り当てられた TCP ポートのうち、使用された数。パーセンテージ値ではありません。
nat-rlm-port-chunk-average-usage-udp	Int32	ゲージ	割り当てられた UDP ポートの平均 UDP ポート使用率。つまり、割り当てられた UDP ポートのうち、使用された数。パーセンテージ値ではありません。
nat-rlm-port-chunk-average-usage-others	Int32	ゲージ	割り当てられた他のポートでの他の（ICMP または GRE）ポートの平均使用率（つまり、割り当てられた「他の」ポートのうち、使用された数）。パーセンテージ値ではありません。
nat-rlm-max-port-chunk-subs	Int64	カウンター	最大数のポートチャンクを使用したサブスクライバの総数。
nat-rlm-max-port-chunk-used	Int32	カウンター	使用された最大ポートチャンク数。
nat-rlm-max-cur-port-chunk-subs	Int64	ゲージ	ポートチャンクの最大数を使用している現在のサブスクライバ数。
nat-rlm-max-cur-port-chunk-used	Int32	ゲージ	アクティブなサブスクライバによって使用された最大ポートチャンク数。

NAT バインドレコード

NAT IP アドレスまたは NAT ポートチャンクがサブスクライバとの間で割り当てまたは割り当て解除されるたびに、NAT バインドレコード（NBR）を生成できます。NBR の生成は、ファイアウォールと NAT ポリシーの設定で設定できます。

NBR の例

```
#sn-start-time,sn-end-time,ip-protocol,ip-subscriber-ip-address,ip-server-ip-address,sn-subscriber-port,
sn-server-port,sn-nat-ip,sn-nat-port-block-start,sn-nat-port-block-end,sn-subscriber-nat-flow-ip,sn-subscriber-nat-flow-port,
sn-nat-realm-name,sn-nat-subscribers-per-ip-address,sn-nat-binding-timer,sn-nat-gmt-offset,sn-nat-port-chunk-alloc-dealloc-flag,
```

```
sn-nat-port-chunk-alloc-time-gmt,sn-nat-port-chunk-dealloc-time-gmt,sn-nat-no-port-packet-dropped,sn-closure-reason
,,,209.165.200.225,,,,209.165.201.1,1024,1039,,,NAT44_PUBLIC2,2,60,+0530,1,02/18/2020
06:41:08,,,
,,,209.165.200.225,,,,209.165.201.2,1024,1031,,,NAT44_PUBLIC5,8064,60,+0530,1,02/18/2020
 06:41:14,,,
,,,209.165.200.225,,,,209.165.201.3,1024,1039,,,NAT44_PUBLIC2,2,60,+0530,0,02/18/2020
06:41:08,02/18/2020 06:42:12,,
,,,209.165.200.225,,,,209.165.201.14,1024,1031,,,NAT44_PUBLIC5,8064,60,+0530,0,02/18/2020
 06:41:14,02/18/2020 06:44:24,,
```

EDR

通常の EDR では、次の NAT 固有の属性がサポートされています。

- sn-nat-subscribers-per-ip-address：NAT IP アドレスごとのサブスクライバ
- sn-subscriber-nat-flow-ip：NAT 対応サブスクライバの NAT IP アドレス
- sn-subscriber-nat-flow-port：NAT 対応サブスクライバの NAT ポート番号

EDR の例

```
#sn-start-time,sn-end-time,ip-protocol,ip-subscriber-ip-address,ip-server-ip-address,sn-subscriber-port,sn-server-port,
sn-nat-ip,sn-nat-port-block-start,sn-nat-port-block-end,sn-subscriber-nat-flow-ip,sn-subscriber-nat-flow-port,sn-nat-realm-name,
sn-nat-subscribers-per-ip-address,sn-nat-binding-timer,sn-nat-gmt-offset,sn-nat-port-chunk-alloc-dealloc-flag,sn-nat-port-chunk-alloc-time-gmt,
sn-nat-port-chunk-dealloc-time-gmt,sn-nat-no-port-packet-dropped,sn-closure-reason
02/18/2020 12:11:11:630,02/18/2020
12:11:11:632,1,209.165.200.225,209.165.201.1,0,0,,,,209.165.200.230,1024,,2,,,,,,0,0
02/18/2020 12:11:08:672,02/18/2020
12:11:09:671,6,209.165.200.225,209.165.201.1,1001,3000,,,,209.165.200.230,1034,,2,,,,,,0,0
02/18/2020 12:11:14:499,02/18/2020
12:11:14:499,17,209.165.200.225,209.165.201.1,1001,3000,,,,209.165.200.240,1025,,8064,,,,,,0,0
```

パケットドロップ EDR

パケットドロップ *EDR* の例

```
#sn-nat-no-port-packet-dropped,sn-start-time,sn-end-time,sn-subscriber-imsi
2,03/13/2020 08:28:24,03/13/2020 08:28:54,123456789012345
```

テザリング検出

機能説明

テザリングとは、モバイルスマートフォンを USB ドングルまたはモデムとして使用して、スマートフォンのデータプランで実行されている PC デバイス（ラップトップ、PDA、タブレットなど）へのインターネット接続を提供することを指します。通常、スマートフォンユーザーの場合、ほとんどのオペレータが無制限のデータプランを設定しており、その使用量はモバイルデバイスとしてのスマートフォンから行うことを目的としています。ただし、一部のサブスクライバは、低コストまたは無制限の使用量データプランを使用して、ブロードバンドまたは Wi-Fi を介した通常のインターネット接続がコストかかるか、利用できないか、または安全でない場所でラップトップにインターネット接続ています。

テザリング検出機能を使用すると、PC デバイスからモバイルスマートフォンにテザリングされたサブスクライバデータ通信の検出が可能になります。また、サービスプロバイダーがこのような通信量を管理し、それに応じてサブスクライバに請求できるよう、効率的なレポートを提供します。テザリング検出は、IPv4（TCP）および IPv6 通信フローでサポートされています。

このリリースでは、IP-TTL ベースのテザリングがサポートされています。この機能は、rulebase レベルで設定でき、rulebase 内に IP-TTL 設定を持つすべてのサブスクライバのすべてのフローに適用されます。

テザリングサポートを構成する

この項では、テザリングサポート機能の構成方法について説明します。

テザリングサポート機能の構成には、以下の手順が含まれます。

- テザリングの rulebase 構成
- テザリングの Ruledef の構成
- テザリングの EDR の構成

テザリングの rulebase 構成

次のコマンドを使用して、テザリングに向けて rulebase パラメータを構成します。

```
configure
  active-charging service service_name
    rulebase rulebase_name
      tethering-detection ip-ttl valuettl_value
      end
```

注：

- **tethering-detection**：このコマンドを使用すると、現在の rulebase および使用する具体的なデータベースに対してテザリング機能を有効または無効にできます。
- **ip-ttl value** *ttl_value*：IP-TTL 構成を使用して、テザリング検出を実行するかを指定します。*ttl_value* は、1 ～ 255 の整数で指定し、テザリングフローの TTL 値を設定します。

テザリングの Ruledef の構成

次のコマンドを使用して、テザリングに向けて ruldef パラメータを構成します。

```
configure
  active-charging serviceservice_name
    ruledef ruledef_name
      ip any-match operator_condition
      tethering-detection ip-ttl flow-tethered
      end
```

注：

- **ip any-match** *operator_condition*：このコマンドは、すべての IPv4 および IPv6 パケットと一致するようにルール式を定義します。
- **ip-ttl**：各 IP-TTL 値ごとにテザリングされたフローまたはテザリングされていないフローを選択することを指定します。
- **flow-tethered**：テザリングがフローで検出された場合、照合するように指定します。

テザリングの EDR の構成

次のコマンドを使用して、テザリング用に EDR を設定します。

```
configure
  active-charging service service_name
    edr-format format_name
      rule-variable flow tethered-ip-ttl priority priority_value
      rule-variable flow ttl priority priority_value
      end
```

注：

- **edr-format** *format_name*：EDR フォーマットを構成します。
- **flow**：EDR でフィールド関連のフローを構成します。
- **tethered-ip-ttl**：フローで IP-TTL ベースのテザリングが検出されます。
- **ttl**：フローの最初のパケットで受信した Time To Live/最大ホップ数の値。

モニタリングおよびトラブルシューティング

この項では、機能のモニタリングとトラブルシューティングのサポートに使用できる CLI コマンドに関する情報を提供します。

show コマンドや出力

この項では、この機能のサポートにおける show コマンドまたはその出力について説明します。

show user-plane-service statistics tethering-detection

この機能をサポートするために、次のフィールドが表示されます。

- 現在のテザリングされたサブスクライバの数
- テザリングされたサブスクライバの総数
- スキャンされたフローの総数
- 検出されたテザリングフローの総数
- 回復されたテザリングフローの総数
- スキャンのためにバイパスされたフローの総数
- テザリング検出統計（ip-ttl）

- スキャンされたフローの総数
- テザリングされたフローの検出数
- テザリングされたアップリンクパケット
- テザリングされたダウンリンクパケット

show user-plane-service statistics rulebase name rulebase_name

この機能をサポートするために、次のフィールドが表示されます。

- テザリング検出（ip-ttl）
  - スキャンされたフローの総数
  - テザリングされたフローの検出数
  - テザリングされたアップリンクパケット
  - テザリングされたダウンリンクパケット

RTP ダイナミックフローの検出

rtp dynamic-flow-detection CLI コマンドは、[ACS Rulebase Configuration] モードで、Real Time Streaming Protocol（RTSP）および Session Description Protocol（SDP）アナライザが子 RTP および RTCP フローを検出できるようにします。RTSP/SIP および SDP アナライザを設定し、**rtp dynamic-flow-detection** CLI が存在していれば、RTP/RTCP の明示的な設定は必要ありません。**rtp dynamic-flow-detection** CLI コマンドを使用すると、子 RTP または RTCP フローが親 RTSP/SIP-SDP フローと相互に関連付けられます。

親フロー（RTSP/SIP-SDP）がクリアされると、子 RTP/RTCP フローもクリアされます。この CLI がない場合、RTP および RTCP の L7 レイヤ分析には、別途アナライザの設定が必要です。RTP/RTCP フローと RTSP/SIP-SDP フローとの相関関係はありません。

ベアラー固有フィルタのルール照合

ベアラー固有フィルタのルール照合機能には、次のようなものがあります。

- IMSI ベースのルールは、サブスクライバの IMSI に従って照合されます。
- APN ベースのルールを使用すると、ベアラーフローのアクセスポイント名（APN）と一致するルール式を定義できます。
- RAT タイプを使用すると、ベアラーフロー内の無線アクセス技術（RAT）に一致するルール式を定義できます。

IMSI プールを構成する

IMSI プールを設定するには、次の CLI コマンドを使用します。

```
configure
  active-charging service service_name
    imsi-pool pool_name
      imsi { imsi_number | range start_imsi to end_imsi }
```

imsi-pool には、IMSI 値または IMSI の範囲を含めることができます。

Rule-line ACS ruledef を構成する

次の CLI コマンドを使用して、ACS Ruledef で構成モードで rule-line を設定します。

```
configure
  active-charging service service_name
    ruledef ruledef_name
      bearer 3gpp imsi { = imsi_value } | { range imsi-pool pool_name }
      bearer 3gpp apn operator apn_name
      bearer 3gpp rat-type operator rat_type
```

IMSI 範囲は、IMSI プールを使用してルール内で設定できます。

bearer 3gpp rat-type *operator rat_type*：

- *operator* は次のいずれかになります。
  - **!+**：等しくない
  - **=**：等しい
- **注意**：このリリースでは、**wlan** は、適格な *rat_type* です。

HTTP Content-Type を構成する

次の CLI コマンドを使用して、HTTP Content-Type entity-header フィールドの値に一致するルール式を定義します。

```
configure
  active-charging service service_name
    ruledef ruledef_name
      http content type [ case-sensitive ] operator content_type
```

show CLI

UPF で **show user-plane-service imsi-pool name** *pool_name* CLI を使用して、サービスで構成された IMSI プール に関する情報を表示します。

アップリンクパケットの IP 送信元違反検出の有効化

表 *6 :* 機能の履歴

機能名	リリース情報	説明
アップリンクパケットの IP 送信元違反検出を有効にする	2024.03.0	cnSGWc のこの機能により、UPF は、アップリンクパケットの発信元が正しいサブスクライバからのものであるかどうかをチェックできます。 この機能は次をサポートします： • この機能により、権限のない関係者へのデータの漏洩を防止することで、セキュリティとサブスクライバのプライバシーが強化されます。この機能は、合法的傍受要件に準拠することにより、サービスプロバイダーの法的および規制的問題のリスクをさらに軽減します。 • cnSGW がセッションに割り当てたトンネルエンドポイント識別子（TEID）の再利用をセッション終了まで遅延させることで、誤ったサブスクライバのタッピング、パケット誤アソシエーションリスク、および合法的傍受（LI）機関による報告を防止します。 この機能は、既存の CLI コマンド **show subscribers user-plane-only full all | callid** *callid* を使用して、構成された **IP Source Violation Action** オプションを表示します。 **デフォルト設定**：無効：有効にするには設定が必要。

IP 送信元違反検出機能が cnSGW で有効になっています。この情報は、cnSGWc から UPF に伝達されます。パケットを無視または破棄する cnSGW の構成に基づいて、ユーザープレーンはソース違反パケットに対処します。

UE がネットワークに接続したら、cnSGW は新しいセッションに一意のトンネルエンドポイント識別子（TEID）を割り当て、UPF への Create Session 要求メッセージに TEID を含めます。

UPF はセッションの確立を確認し、TEID を使用してユーザー データパケットをルーティングします。

この機能により、cnSGW がサブスクライバごとのセッションに割り当てるトンネルエンドポイント識別子（TEID）は、セッションが終了するまで別のサブスクライバですぐに再利用されません。遅延 TEID 再利用により、LI エージェンシーによる誤ったサブスクライバのタップと報告が回避されます。

ユーザープレーンでの IP 送信元違反は、**show subscribers user-plane-only full all | callid** *call_id* CLI コマンドを使用して確認できます。

IP 送信元違反のアップリンクパケットの検証の詳細については、『*cnSGWc* 構成とアドミニストレーション ガイド』の「IP 送信元違反のアップリンクパケットの検証」を参照してください。

アップリンクパケットの IP 送信元違反の有効化の仕組み

この機能を有効または無効にする場合の重要なポイントは次のとおりです。

- デフォルトでは、この機能はディセーブルになっています。
- この機能が IGNORE オプションとともに有効になっている場合、パケットは無効な送信元 IP アドレスに対して検証されます。対応する統計がそれに応じて増加します。ただし、IP が無効であるため、パケットに対するアクションはありません。
- この機能が DISCARD オプションとともに有効になっている場合、無効な送信元 IP アドレスを持つパケットは廃棄されます。対応する統計がそれに応じて増加します。

アップリンクパケットの IP 送信元違反を確認する

UPF でこの手順を使用して、cnSGWc で構成した値で、IP 送信元違反機能が有効になっているか、無効になっているかを確認します。

手順

show subscribers user-plane-only full all | callid <>を入力します。

例：

```
show subscribers user-plane-only full all | callid <>
```

新しいカウンタ **ip source violations ignored:<>** が追加され、無視または転送された IP 送信元違反パケットの数を表示できるようになりました。

以下は、**show subscribers user-plane-only full all | callid <>** CLI コマンドの出力例です。ここでは、**IP Source Violation Action**、**ip source violations** および **ip source violations ignored** の値が構成されています。

```
show subscribers user-plane-only full all
  Local SEID      : [0x0004000000000000] 1125899906842624
  Remote SEID     : [0x0004000000000005] 1125899906842629
  State           : Connected
  Connect Time    : Wed Jun  5 05:09:22 2024
  Idle time       : 00h11m19s
```

```
Access Type: n/a                        Network Type: IP
user-plane-service-name: UP21
active-service-scheme-name:
Callid: 00004e21
Rulebase:
Interface Type: Sxa
eMPS Session: No
eMPS Session Priority: 0
Session-Type: normal
Precedence-order: 0
Data Queue: 0
Card/Cpu: 1/0                           Sessmgr Instance: 1
IP address: n/a
Next Hop Ip Address: n/a
Source context: EPC2-UP                 Destination context: EPC2-UP
PDN-Instance: n/a
User-plane-Sx-addr: 20.20.20.106
Control-plane-Sx-addr: 20.20.20.101
Number of associated PDRs : 2
Number of associated ADC PDRs : 0
Number of associated FARs : 2
Number of associated QERs : 0
Number of associated BARs : 0
Number of associated URRs : 1
Uplink APN AMBR (bps) : 0       Downlink APN AMBR (bps) : 0
active input acl: n/a                active output acl: n/a
active input ipv6 acl: n/a             active output ipv6 acl: n/a
Bandwidth Policy: n/a
FW-and-NAT Policy: n/a
FW-and-NAT Policy ID: n/a
Firewall Policy IPv4: n/a
Firewall Policy IPv6: n/a
NAT Policy NAT44: n/a
NAT Policy NAT64: n/a
Converged Session: No                  Converged Peer Callid:      n/a
Visited Call: No
IP Source Violation Action: DISCARD
Subscriber Parameters:
IMSI: 404005123456789
IMEI: 112233445566778
MSISDN: 9890098900
Rat Type: 6
Old Rat Type: n/a
CF Policy ID: n/a
GX CF Policy ID: n/a
Roaming Status: n/a
Old Roaming Status: n/a
input pkts: 5                                   output pkts: 0
input bytes: 250                                output bytes: 0
input bytes dropped: 123                        output bytes dropped: 0
input pkts dropped: 3                           output pkts dropped: 0
CF Buffered Uplink Packets: 0                   CF Buffered Downlink Packets: 0
CF Buffered Uplink Bytes: 0                     CF Buffered Downlink Bytes: 0
Uplink Packets in Buffer: 0                     Downlink Packets in Buffer: 0
Buff Over-limit Uplink Pkts: 0                  Buff Over-limit Downlink Pkts: 0
DDN buffered pkts : 0                           DDN buffered bytes : 0
DDN buffer overflow drop pkts : 0               DDN buffer overflow drop bytes : 0
pk rate from user(bps): 0                       pk rate to user(bps): 0
ave rate from user(bps): 0                      ave rate to user(bps): 0
sust rate from user(bps): 0                     sust rate to user(bps): 0
pk rate from user(pps): 0                       pk rate to user(pps): 0
ave rate from user(pps): 0                      ave rate to user(pps): 0
sust rate from user(pps): 0                     sust rate to user(pps): 0
ipv4 bad hdr: 0                                 ipv4 ttl exceeded: 0
```

```
  ipv4 fragments sent: 0                          ipv4 could not fragment: 0
  ipv4 bad length trim: 0
  ipv4 input mcast drop: 0                        ipv4 input bcast drop: 0
  input pkts dropped (0 mbr): 0                   output pkts dropped (0 mbr): 0
  ipv4 input acl drop: 0                          ipv4 output acl drop: 0
  ipv6 input acl drop: 0                          ipv6 output acl drop: 0
  ip source violations: 3                         ipv4 output no-flow drop: 0
  ipv6 bad hdr: 0                                 ipv6 bad length trim: 0
  ip source violations ignored: 0
  ipv4 icmp packets dropped: 0
  APN AMBR Uplink Pkts Drop: 0                     APN AMBR Downlink Pkts Drop: 0
  APN AMBR Uplink Bytes Drop: 0                    APN AMBR Downlink Bytes Drop: 0
  APN AMBR Uplink Pkts IP pref lowered: 0          APN AMBR Downlink Pkts IP pref lowered: 0

  APN AMBR Uplink Bytes IP pref lowered: 0         APN AMBR Downlink Bytes IP pref lowered: 0

  ITC Uplink Pkts Drop: 0                          ITC Downlink Pkts Drop: 0
  ITC Uplink Bytes Drop: 0                         ITC Downlink Bytes Drop: 0
  ITC Uplink Pkts IP pref lowered: 0               ITC Downlink Pkts IP pref lowered: 0
  ITC Uplink Bytes IP pref lowered: 0              ITC Downlink Bytes IP pref lowered: 0
  ITC Terminated Flows:: 0                         ITC Redirected Flows: 0
  Flow Action Terminated Flows: 0                  Flow Action Redirected Flows: 0
  ToS marked Uplink Pkts: 0                        ToS marked Downlink Pkts: 0
  CC Dropped Uplink Pkts: 0                        CC Dropped Downlink Pkts: 0
  CC Dropped Uplink bytes: 0                       CC Dropped Downlink Bytes: 0
  Uplink Inflight Pkts: 0                          Downlink Inflight Pkts: 0
  QFI Mismatch Uplink Pkts Drop: 0                 QFI Mismatch Downlink Pkts Drop: 0
  Total QoS-Group(s) Active: 0
  DNS-to-EDNS Uplink Pkts: 0                     DNS-to-EDNS Uplink Bytes: 0
  EDNS Response Received: 0

  Flow information:
    Current Active Flows:
      TCP: 0
      UDP: 0
    Total Flows:
      TCP: 0
      UDP: 0
      FP:  0

Static & Predef Rule Match stats:
Rule Name            Pkts-Down  Bytes-Down Pkts-Up    Bytes-Up   Hits       Match-Bypassed
FP-Down(Pkts/Bytes)  FP-Up(Pkts/Bytes)
-------------------- ---------- ---------- ---------- ---------- ---------- --------------
-------------------  ------------------

Dynamic Rule Match stats:
PDR Id   Rule Name            Pkts-Down  Bytes-Down Pkts-Up    Bytes-Up   Hits       Match-Bypassed
  FP-Down(Pkts/Bytes)  FP-Up(Pkts/Bytes)
-------- -------------------- ---------- ---------- ---------- ---------- ---------- --------------
  -------------------  ------------------
0x0002   N/A                         0          0          5        250          0              0
       0/0               5/250

Post-Processing Rule Match stats:
Rule Name            Pkts-Down  Bytes-Down Pkts-Up    Bytes-Up   Hits       Match-Bypassed
FP-Down(Pkts/Bytes)  FP-Up(Pkts/Bytes)
-------------------- ---------- ---------- ---------- ---------- ---------- --------------
-------------------  ------------------

Firewall-NAT Access Rule Match stats:
Firewall-Ruledef Name Pkts-Down Bytes-Down    Pkts-Up   Bytes-Up Dr-Pkts-Dn Dr-Byts-Dn Dr-Pkts-Up
Dr-Byts-Up       Hits
-------------------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
```

```
---------- ----------

QoS-Group Statistics:
QGR Name             Pkts-Down  Bytes-Down Pkts-Up    Bytes-Up   Hits       Match-Bypassed
FP-Down(Pkts/Bytes)  FP-Up(Pkts/Bytes)
-------------------- ---------- ---------- ---------- ---------- ---------- --------------
-------------------  -----------------

SGACL Match stats:
ACL Name             Pkts-Down  Bytes-Down Pkts-Up    Bytes-Up   Pkts dropped
-------------------- ---------- ---------- ---------- ---------- -------------

  CIoT APN Rate Control:
    Allowed UL Limit:   0                          Allowed DL limit:    0
    Remaining UL Limit: 0                          Remaining DL limit:  0
    Allowed Time unit: unrestricted                Status Validity Time: N/A

Total subscribers matching specified criteria: 1
```

URL のブロックリスト登録登録

機能説明

URL ブロックリスト登録機能は、URL または URI がブロックリストに登録されている Web サイトからコンテンツを表示またはダウンロードするためのサブスクライバのアクセスを規制します。この機能では、検出された URL がブロック対象として分類されているかどうかを示す URL のリストを記録するデータベースが使用されます。

機能の仕組み

UFP で URL ブロックリスト登録機能を有効にするには、フラッシュ、SFTP、またはそのサブディレクトリ内に「optblk.bin」という名前の URL ブロックリスト登録データベースが存在する必要があります。このデータベースディレクトリのパスは、ユーザープレーンのサービス起動後に、ユーザープレーンで設定する必要があります。

HTTP アナライザの URL ブロックリスト登録を有効にする必要があります。HTTP アナライザは、受信した HTTP リクエストデータパケットから URL 情報を抽出します。抽出された URL コンテンツが、URL ブロックリスト登録データベースと比較されます。受信した HTTP データパケットの URL がデータベースの URL エントリと一致すると、その URL はブロックリスト登録された URL として扱われ、該当する HTTP パケットに対して次のいずれかのアクションが実行されます。

- フローの終了
- パケットの破棄

URL ブロックリスト登録構成は、コントロールプレーン（CP）の Active Charging Service の Rulebase 設定で構成する必要があります。さらに、Active Charging Service レベルの構成では、Exact と Generic の 2 つのブロックリスト登録メソッドがサポートされています。



重要 ブロックリスト登録データベースは、IWF（Internet Watch Foundation）およびNCMEC（National Center for Missing and Exploited children）によって提供されます。UPF は常に、最適化されたフォーマットでブロックリスト登録データベースを受信します。

URL ブロックリスト登録データベースのアップグレード

次の 2 つの方法による URL データベースのアップグレードがサポートされます。

- タイマーベースのアップグレードまたは自動アップグレード
- CLI ベースのアップグレードまたは手動アップグレード

タイマーベースのアップグレードまたは自動アップグレード

データベースがシャーシに初めてロードされると、5 分間のタイマーが開始されます。このプロセスは、データベースを自動アップグレードするために開始されるものです。

タイマー終了時に、ディレクトリパスに有効なデータベースのより上位のバージョンがある場合には、データベースのアップグレード手順が開始され、新しいバージョンのデータベースが UPF にロードされます。

URLブロックリスト登録データベースをアップグレードするには、「optblk_f.bin」という名前の有効な URL ブロックリスト登録データベースの上位バージョンが、現在のデータベース「optblk.bin」と同じディレクトリ内にある必要があります。

データベースが正常にアップグレードされると、以前の「optblk.bin」ファイルの名前が「optblk_0.bin」に変更され、「optblk_f.bin」ファイルの名前が「optblk.bin」に変更されます。すると、「optblk_0.bin」ファイルは、古いデータベースのバックアップファイルとして扱われます。

追加のアップグレードが実行されると、「optblk_0.bin」ファイルの名前は「optblk_1.bin」に変更され、現在の「optblk.bin」ファイルの名前は「optblk_0.bin」に変更されます。

データベースに保存するバックアップファイルの数を構成するには、「*UPF* で *URL* ブロックリスト登録データベースを読み込む」 項を参照してください。

CLI ベースのアップグレードまたは手動アップグレード

現在のデータベースを新しいバージョンにアップグレードするには、「*URL* ブロックリスト登録 データベースをアップグレードする」項を参照してください。

URL ブロックリスト登録を構成する

UPF で URL ブロックリスト登録データベースを読み込む

次の構成を使用して、UPF で URL ブロックリスト登録データベース読み込みます。

2022.01.0 より前のリリース：

```
configure
  url-blacklisting database directory path database_directory_path
  url-blacklisting database max-versions max_version_value
  end
```

2022.01.0 以降のリリース：

```
configure
  url-blockedlisting database directory path database_directory_path
  url-blockedlisting database max-versions max_version_value
  end
```

注：

- **database directory path**：データベースのディレクトリパスを設定します。

  database_directory_path は、1 ～ 255 文字の文字列です。

- **max-versions**：データベースの最大アップグレードバージョンを設定します。

  max_version_value は 0 ～ 3 の整数です。

URL ブロックリスト登録データベースをアップグレードする

この構成を利用して、URLブロックリスト登録データベースを手動でアップグレードします。

2022.01.0 より前のリリース：

```
upgrade url-blacklisting database
end
```

2022.01.0 以降のリリース：

```
upgrade url-blockedlisting database
end
```

URL ブロックリスト登録を有効にするための構成

次の構成を使用して、UPF で URL ブロックリスト登録機能を有効にします。

2022.01.0 より前のリリース：

```
configure
  active-charging service service_name
    url-blacklisting match-method [ exact | generic ]
    rulebase rulebase_name
      url-blacklisting action [ discard | terminate-flow ]
      end
```

2022.01.0 以降のリリース：

```
configure
  active-charging service service_name
    url-blockedlisting match-method [ exact | generic ]
    rulebase rulebase_name
      url-blockedlisting action [ discard | terminate-flow ]
      end
```

注：

- **match-method [exact | generic]**：URL ブロックリスト登録に使用する一致メソッドを指定します。
  - **exact**：URL ブロックリスト登録が、URL の完全一致を実行します。
  - **generic**：URL ブロックリスト登録が、URL の汎用一致を実行します。
- **url-blockedlisting action [discard | terminate-flow]**:
  - **discard**：受信した HTTP パケットを破棄します。
  - **terminate-flow**：受信した HTTP パケットのフローを終了します。

モニタリングおよびトラブルシューティング

この項では、機能のモニタリングとトラブルシューティングのサポートに使用できる CLI コマンドに関する情報を提供します。

show コマンドや出力

この項では、この機能のサポートにおける show コマンドまたはその出力について説明します。

show user-plane-service url-blacklisting database

この機能をサポートするために、次のフィールドが表示されます。

- URL ブラックリスト静的評価データベース：
  - 前回のアップグレードステータス
  - パス
    - データベースステータス
    - DB 内の URL の数
    - タイプ
    - バージョン
    - 作成時間
    - ホスト名
    - コメント
    - 最終アクセス時刻
    - 最終変更時刻
    - ステータスの最終変更時刻

show user-plane-service url-blacklisting database url database_directory_path

この機能をサポートするために、次のフィールドが表示されます。

- URL ブラックリスト静的評価データベース：
  - 前回のアップグレードステータス
  - パス
    - データベースステータス
    - DB 内の URL の数
    - タイプ
    - バージョン
    - 作成時間
    - ホスト名
    - コメント
    - 最終アクセス時刻
    - 最終変更時刻
    - ステータスの最終変更時刻

show user-plane-service url-blacklisting database facility sessmgr all

この機能をサポートするために、次のフィールドが表示されます。

- URL-Blacklisting SessMgr Instance Based Database Configuration
  - SessMgr Instance
  - BL DB Load Status
  - BL DB Version
  - Number of URLs
  - チェックサム

show user-plane-service rulebase name rulebase_name

この機能をサポートするために、次のフィールドが表示されます。

- URL-Blacklisting Action
- URL-Blacklisting Content ID

show user-plane-service inline-services info

この機能をサポートするために、次のフィールドが表示されます。

- URL ブラックリスト：有効
  - URL ブラックリストの照合方法：汎用

show user-plane-service inline-services url-blockedlisting statistics

この機能をサポートするために、次の情報が表示されます。

- 累積 URL ブロックリスト登録の統計
  - ブロックリストに登録された URL のヒット数
  - ブロックリストに登録されていない URL 数
  - 一致したルールベースの総数

show user-plane-service inline-services url-blacklisting statistics rulebase name rulebase_name

この機能をサポートするために、次のフィールドが表示されます。

- ルールベース名
  - URL ブラックリストの統計情報
  - ブラックリストに登録された URL のヒット数
  - ブラックリストに登録された URL の欠落数
- 一致したルールベースの総数

静的ルールと事前定義済みルールを構成する

この項では、課金アクション構成で静的ルールと事前定義済みルールを構成する方法について説明します。

```
configure
   active-charging service service_name
      charging-action charging_action_name
         flow action { discard [ downlink | uplink ] | redirect-url
redirect_url | terminate-flow }
         end
```

注：

- **flow action**：ルール定義に一致するパケットで実行するアクションを指定します。

- **discard [downlink | uplink]**：ダウンリンクまたはアップリンクのパケットを破棄することを指定します。
- **redirect-url** *redirect_url*：リダイレクトする URL を指定します。例：http://search.com/subtarg=#HTTP.URL#
- **terminate-flow**：終了するフローを指定します。

• redirect-url の場合は、rulebase で HTTP アナライザを構成します。例：

```
route priority 70 ruledef http-port analyzer HTTP
   ruledef http-port
   tcp either-port = 80
   rule-application routing
   exit
```

DPI 向け L7 プロトコルの ACS Ruledef を構成する

は、プロトコル フィールドと状態情報に基づいた複数の L3 ～ L7 プロトコルにおける一連の一致条件を表します。各 ruledef は、アクティブな課金サービス内の複数のルールベースで使用できます。

（注） UPF では、アクティブなコールおよびデータフロー中に ruledef 内のルール行の追加または削除が行われた場合、この構成変更は現在のフローには適用されません。ただし、構成変更は、新しいコールと同じコールの新しいフローに適用されます。

以下は、ACS ルール定義を作成、構成、または削除する方法を示す構成例です。

```
configure
   active-charging service service_name
      ruledef ruledef_name
        dns { any-match value | query-type query_type | query-name query_name
 }
         ip any-match [ = | != ] [ TRUE | FALSE ]
         ip dst-address { operator { { ipv4_address | ipv6_address } | {
ipv4_address/mask | ipv6_address/mask} | address-group ipv6_address } | { !range
 | range } host-pool host_pool_name }
         ip server-ip-address { operator { { ipv4_address | ipv6_address } |
{ ipv4_address/mask | ipv6_address/mask} | address-group ipv6_address } | { !range
 | range } host-pool host_pool_name }
         multi-line-or all-lines
         rule-application { charging | post-processing | routing }
         { tcp | udp } { either-port port_number }
         end
```

注：

- **ruledef** *ruledef_name*：追加、設定、または削除する ruledef を指定します。 *ruledef_name* は ACS ルールデ の名前である必要があります。また、1 ～ 63 文字の英数字の文字列である必要があり、句読点を含めることができます。各 ruledef には一意の名前を付ける必要があります。ホストプール、ポートマップ、IMSI プール、ファイアウォール、ルーティング、および課金のルール定義には一意の名前を付ける必要があります。
- 指定された ruledef が存在しない場合は作成され、CLI モードは ruledef を設定できる ACS Ruledef 構成モードに変わります。
- 指定された ruledef がすでに存在する場合、CLI モードはその ruledef の AACS Ruledef 構成モードに変わります。ACS Ruledef 構成モードを使用して、個々のルール定義（ruledef）でルール式を作成および管理します。
- **ip any-match [= | !=] [TRUE | FALSE]:** ：このコマンドは、IPv4/IPv6 パケットに一致するルール式を定義します。コマンドの 演算子 と 条件 は、次を指定します。
  - 演算子
    - != : 等しくない
    - <=: 等しい
  - 条件
    - FALSE
    - TRUE
- **ip dst-address {** *operator* **{ {** *ipv4_address* **|** *ipv6_address* **} | {** *ipv4_address/mask* **|***ipv6_address/mask* **} | address-group** *ipv6_address* **} | { !range | range } host-pool** *host_pool_name* **}**：このコマンドは、IP ヘッダー内の IP 宛先アドレスフィールドに一致するルール式を定義します。
  - *ipv4_address* **|** *ipv6_address*：発信通信の宛先ノードの IP アドレスを指定します。 *ipv4_address* **|** *ipv6_address* は、IPv4 ドット付き 10 進表記または IPv6 コロン区切り 16 進表記の IP アドレスである必要があります。
  - *ipv4_address/mask* **|** *ipv6_address/mask*：発信通信の宛先ノードの IP アドレスを指定します。 *ipv4_address/mask* **|** *ipv6_address/mask* は、 サブネットマスクビットを含む IPv4 ドット付き 10 進表記または IPv6 コロン区切り 16 進表記の IP アドレスである必要があります。マスク ビットは、サブネット マスクのビット数に対応する数値です。
  - *address-group ipv6_address*：ワイルドカード入力や特殊な範囲入力で設定された IPv6 アドレスのグループを指定します。複数のワイルドカード文字を入力として受け入れることができますが、入力できるのは 1 つの 2 バイト範囲入力だけです。ワイルドカード文字の入力と 2 バイトの範囲の入力の両方を、特定の IPv6 アドレス内で一緒に設定できます。
  - **host-pool***host_pool_name*：ホスト プールの名前を指定します。 *host_pool_name* は、1 ～ 63 文字の英数字の文字列である必要があります。
  - コマンドの 演算子 は、次を指定します。

- !=：等しくない
- <：次の値以下
- =：等しい
- >=: 次の値以上

- **multi-line-or all-lines**：このコマンドは、単一の ruledef で複数の URL 式を指定できます。ruledef が評価されるときに、multi-line-or all-lines コマンドが設定されていると、論理 OR 演算子が ruledef 内のすべてのルール式に適用され、ruledef が一致するかどうかが判断されます。multi-line-or all-lines コマンドが設定されていない場合、論理 AND 演算子がすべてのルール式に適用されます。
- **rule-application { charging | post-processing | routing }**：このコマンドは、ルール定義のルール適用を指定します。**.**
  - **charging**：現在の ruledef が課金目的であることを指定します。
  - **post-processing**：現在の ruledef が後処理用であることを指定します。これにより、パケットのルール照合が無効になっている場合でも、パケットを処理できます。
  - **routing**：現在の ruledef がルーティング用であると指定します。アクティブな課金サービスでのルーティングには、最大 256 のルール定義を定義できます。デフォルトで、ディセーブルになっています。
- **dns { any-match** *value* **| query-type** *query_type* **| query-name** *query_name* **}**：このコマンドを使用すると、すべての DNS パケットまたはクエリタイプまたはクエリ名に基づいてパケットを照合するルール式を定義できます。

  ip server-ip-address *ip_address_value*：このコマンドを使用すると、接続の接続先の IP アドレスと一致するルール式を定義できます。
- **{ tcp | udp } { either-port** *port_number* **}**：このコマンドを使用すると、UDP/TCP ヘッダーの接続先または送信元ポート番号のいずれかと一致するルール式を定義できます。

DPI 向け L7 プロトコルの課金アクションの構成

このセクションでは、課金アクションの構成方法を説明します。課金アクションは、構成されたルールに一致したときに実行されるアクションを意味します。アクションの範囲として、アカウンティング レコード（例：EDR）の生成から IP パケットのドロップなどまでが含まれます。また、課金アクションによって、使用量計算の原則を規定します。再送信されたパケットをカウントするかどうか、および課金情報にどのプロトコルフィールド（L3/L4/L7 など）を使用するかがこれに当たります。

課金アクション構成を使用して、ruledef に関連付けられている QoS と課金関連パラメータを定義します。

```
configure
   active-charging service service_name
     charging-action charging_action
        allocation-retention-priority priority  [ pci pci_value  | pvi
pvi_value
        billing-action egcdr
        cca charging credit [ rating-group coupon_id ] [
preemptively-request ]
        content-id content_id
        flow action { discard [ downlink | uplink ] | redirect-url
redirect_url | terminate-flow }
        flow limit-for-bandwidth { { direction { downlink | uplink }
peak-data-rate bps peak-burst-size bytes violate-action { discard |
lower-ip-precedence } [ committed-data-rate bps committed-burst-size
bytes [ exceed-action { discard | lower-ip-precedence } ] ] } | { id id
 } }
        nexthop-forwarding-address ipv4_address/ipv6_address
        qos-class-identifier qos_class_identifier
        service-identifier service_id
        tft packet-filter packet_filter_name
        tft-notify-ue
        tos { af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 |
af33 | af41 | af42 | af43 | be | ef | lower-bits tos_value } [ downlink
| uplink ]
```

注：

- **charging-action***charge_action_name*：課金アクションの名前を指定します。*charge_action_name* は、1 ～ 63 文字の英数字の文字列である必要があります。また、句読点を含めることができます。各家禽アクションには一意の名前が必要です。
- 指定された課金アクションが存在しない場合は作成され、CLI モードは課金アクションを構成できる ACS 課金アクション構成モードに変わります。
- 指定された課金アクションが存在する場合、CLI モードはその課金アクションの ACS 課金アクション構成モードに変更されます。
- **allocation-retention-priority** *priority* **[pci** *pci_value* **| pvi** *pvi_value* **]**：割り当て保持優先順位（ARP）を構成します。優先順位 は、1 ～ 15 の範囲の整数で指定する必要があります。
  - **pci** *pci_value* ：**Specifies the Preemption Capability Indication (PCI) value. The options are:**
    - MAY_PREMPT：フローをプリエンプション処理できます。これはデフォルト値です。
    - NOT_PREMPT：フローをプリエンプション処理できません
  - **pvi** *pvi_value*：Preemption Vulnerability Indication（PVI）値を指定します。次のオプションがあります：
    - NOT_PREEMPTABLE：フローをプリエンプション処理できません。これはデフォルト値です。

• PREEMPTABLE：フローをプリエンプション処理できます

- **billing-action**：具体的なルール定義を一致するパケットに請求アクションを構成します。
- **cca charging credit**：クレジットコントロール課金クレジット動作を有効または無効にします。
- **content-id**：Rating Group を構成します。
- **flow action**：ルール定義に一致するパケットで実行するアクションを指定します。
  - **discard [downlink | uplink]**：ダウンリンクまたはアップリンクのパケットを破棄することを指定します。
  - **redirect-url** *redirect_url*：リダイレクトする URL を指定します。
  - **terminate-flow**：終了するフローを指定します。
- **flow limit-for-bandwidth**：MBR や GBR などの QoS パラメータを構成します。
  - Peakdatarate（MBR）：デフォルトは 3000 bps です
  - Peakburstsize：デフォルトは 3,000 バイトです
  - [committedDataRate（GBR）]：デフォルトは 144000 bps です
  - commitedBurstSize：デフォルトは 3000 バイト
- **nexthop-forwarding-address** *ipv4_address/ipv6_address* ：ネクストホップ転送アドレスを構成します。
- **qos-class-identifier** *qos_class_identifier* ：課金アクションの QCI を構成します。*qos_class_identifier* は、1 ～ 9 または 128 ～ 254（オペレータ固有）の範囲の整数で指定する必要があります。
- **service_identifier** *service_id*：生成された請求レコードで使用するサービス識別子を構成します。*service_id* は、1 ～ 2147483647 の範囲の整数で指定する必要があります。
- **tft packet-filter** *packet_filter_name*：現在の課金アクションに追加または削除するパケットフィルタを指定します。*packet_filter_name* は、パケットフィルタ名で、1 ～ 63 文字の英数字の文字列で指定する必要があります。
- **tft-notify-ue**：特定のトリガ条件に基づいて UE に対する TFT 更新を制御します。
- **tos**：Type of Service（ToS）オクテットを構成します。

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。

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。