

NSH Traffic Steering

- Revision History, on page 1
- Feature Description, on page 1
- How it Works—Standalone Mode, on page 6
- Configuring the L2 and NSH Traffic Steering Feature—Standalone Mode, on page 10
- Monitoring and Troubleshooting—Standalone Mode, on page 19
- Feature Description—Sandwich Mode, on page 26
- How it Works—Sandwich Mode, on page 28
- Configuring NSH Traffic Steering—Sandwich Mode, on page 33
- Configuring Post Processing Ruledef in Both Standalone and Sandwich Mode, on page 36
- Configuring BFD Instance Id Using Interface Name in UP Appliance Group, on page 36
- Monitoring and Troubleshooting the NSH Traffic Steering—Sandwich Mode, on page 37

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
The support for post processing rule condition match for Traffic Steering and L2 up-appliance-group BFD configuration is available in this release.	21.23.22
First introduced.	Pre 21.24

Feature Description

The 3GPP EPC architecture enables data traffic steering across various service functions on the Gi interface. The traffic steering architecture is based on the Network Service Header (NSH) service chaining protocol. The EPC gateway needs to perform the traffic steering to steer the traffic across the multiple service chains containing the appliances which support NSH.

The following are the two modes of NSH Traffic Steering:

- Standalone Mode
- Sandwich Mode

This feature enables the charging and steering of traffic to be independent of each other based on the customer's requirement. It is possible for customers to include a large set of traffic categories for steering traffic with minimum configurational enhancements within the existing use case scenarios.

Post Processing Rule Condition Match for Traffic Steering

A simple traffic classification helps in simplifying the operation and configuration processes in traffic steering due to the huge number of the charging rules across multiple rulebases.

- Trigger condition in service scheme framework supports post processing ruledef name match.
- The L3/L4 ruledef which is configured as a post processing rule for traffic is traffic-steered.
- Trigger action supports trigger condition of post processing rule match for traffic steering.
- The post processing ruledef name in trigger condition is supported in PFD push and RCM.

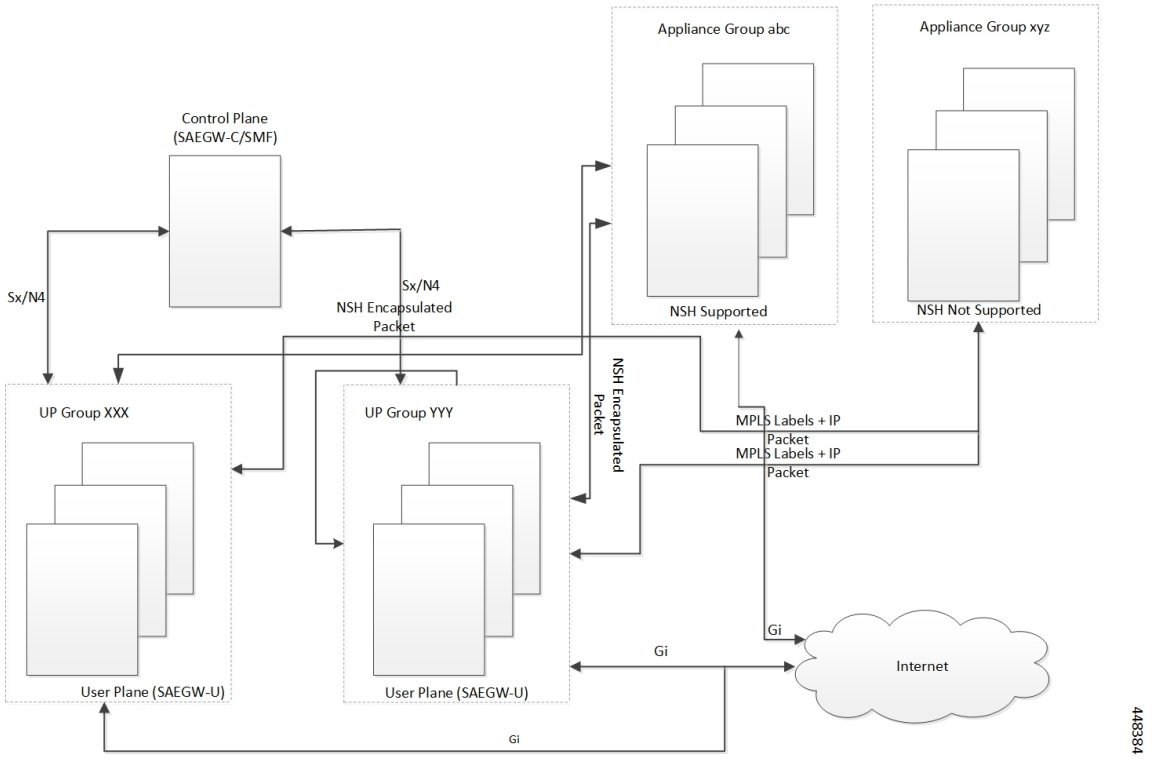
BFD Instance Id Configuration in UP Appliance Group Using Interface Names

For traffic steering, the configuration of Bidirectional Forwarding Detection (BFD) instance id in the **up-appliance-group** is enabled using interface names along with IP configuration.

Architecture—Standalone Mode

The following figure illustrates the architectural setup for CUPS based gateway for NSH appliances.

Figure 1: NSH Traffic Steering Architecture—Standalone Mode



The feature supports a service function chain for NSH supported appliances. The gateway is configured to select a suitable steering or encapsulation method for steering traffic that is based on each appliance instance or group.

Table 1: Call Flow

Step	Description
1.	UL packet received at the SAEGW-U is classified based on the configured policy associated with the appropriate SFC.
2.	The Saegw performs the SFP selection based on the stickiness (MSISDN stickiness) or service and load availability of the SFPs. The UL traffic is NSH (IP-UDP) encapsulated steered on the selected SFP with the context header populated as necessary.
3.	The NSH appliance on receiving the NSH Packet, processes the IP packet (and possibly the context header) and sends the packet over the Gi interface.
4.	Destination server sends the DL packet from the Gi interface to the SAEGW-U. The DL traffic is NSH (IP-UDP) encapsulated steered on the selected SFP with the context header populated as necessary.

Step	Description
5.	The NSH appliance on receiving the NSH Packet, processes the IP packet (and possibly the context header) and hairpin the packet back to the SAEGW-U.
6.	The SAEGW-U on receiving the NSH packet: • Decapsulates the received payload • Processes the IP packet (and possibly the context header) and send the packet over the Gn interface to the UE

Components

The traffic steering architecture comprises of the following main components:

Control Plane (SAEGW-C)

CP sends information to UP on how to steer the subscriber's traffic. The UP steers all or only part of the subscriber data traffic that is based on policies that are defined for the subscriber. It's possible to steer different types of subscriber traffic to different service function chains.

CP selects the service chain name for a subscriber after it receives the Ts-subscription-scheme AVP from PCRF, which is based on locally configured policies.

User Plane (SAEGW-U)

Based on the policy, which UP receives from CP, it steers the subscriber data traffic to one or more service function chains.

UP also performs the following functions:

- Select a Service Function Path (SFP) for a particular Service Function Chain (SFC).
- Maintain subscriber stickiness while forwarding traffic toward the appliances.
- If a node or an appliance fails, reselect and steer the subscriber data traffic to a new node.
- Manage **In-Service** and **Out-of-Service** status for SFPs.
- Manage SFC status depending on the number of serviceable SFPs available within an SFC.

NSH

For monitoring health of the NSH appliance, each SAEGW-U/UPF is responsible for monitoring of the appliance load and serviceability stat.

- Use the OAM NSH packet mechanism to monitor the status of the appliances.
- The monitoring frequency for the configuration is (1-20 secs) with a default interval of 1 sec.
- In case the OAM request times out. Do the retry. The timeout and the retry value are configurable with values of 1-5 secs for timeout (default of 3 secs) and 1-3 retries (default of two retries).

- In addition to the appliance serviceability status, the current load on the appliance is under observation. Monitor the current load in order to maintain the optimum load balancing among various instances of an SF. This load status returns through the NSHs OAM response packet.

Limitations

The NSH traffic steering has the following limitations:

- On NSH appliance, make sure that the interface fragmentation doesn't happen. Keep the MTU towards the NSH appliance interface bigger than Gn/Gi interface.
- For HTTP pipelined sessions, mid flow HTTP partial packets, and TCP Out of order packets, if requires an SFP revaluation with L7 conditions, doesn't reach the NSH appliance.
- If you remove the SFP ID configuration from the main configuration, show configuration still shows the SFP ID. The SFP ID goes away once committed to VPP, using the commit CLI.
- Traffic steering statistics indicate the packets which are candidate for traffic steering. In traffic steering statistics those packets are also counted which are dropped by quota exhaust, though they still are the candidates for traffic steering.
- If modification of NSH SRC/bind IP address OR appliance IP address is required in the configuration for any NSH appliance's instance, then you need to remove the instance, then SFPs associated with it, put the SFPs and new instance with modified IP addresses. Perform the commit afterwards.
- When node failure is done and continuous data is coming, then there can be discrepancy in steering statistics. Data steered on SFPs which is going down is not reflected in statistics.
- For multi PDN call, NSH instance stickiness is restricted to each subscriber session.
- In case of a change in the state at the SAEGW-U due to ICSR or config change like SFP removal in the interim period, there is a possibility that packets which are being hair pinned back from the appliance in this window can be dropped. All further incoming packets are processed as normal
- In case of the first packet of a flow being a DL packet (session recovery), just that first packet is dropped. However, the retransmitted packet and all subsequent packets are sent out as normal.
- In case of change in the NSH format tags, tag types stream-fp-md encode, reverse-stream-fp-md , secondary-srv-path-hdr, and rating-group comes into effect for new flows and not for existing flows. Any changes for the remaining tags in the NSH format applies for new sessions while traffic on existing sessions continue with older format tags. In such cases, particularly in case of modification and deletion of tags, the appliance can mismatch the tag values received in the NSH packets and can lead to ambiguous behavior. So, perform the NSH-format type changes carefully.
- Server initiated TCP Flows are not considered for Traffic steering.
- Monsub support for capturing NSH traffic is not currently available.
- For addressing any appliance level limitation (example - traffic type), policy selection configuration on the service scheme provides the flexibility to filter out such traffic from selecting a service chain containing such appliance.
- For N:M setup, service scheme config (trigger action, trigger condition, service-scheme, subscriber class, and subscriber base) needs to be configured in Day-0 config on UP. Service scheme when configured, in common config on UP, is hitting a race condition leading to service scheme not getting configured on user-plane sessmgrs intermittently, which leads to failure of traffic steering functionality.

- OAM stats for L2 steering is partially supported.
- For HTTP concatenated packet, the packet is traffic steered based on the policy matched by the last HTTP GET in the packet.
- In case a appliance goes down, the flow gets onloaded for revaluation when the next uplink packet is recovered on the flow. Post which the a new SFP selection happens and the traffic is steered to the new appliance.

How it Works—Standalone Mode

Packet Flows

This section describes the packet flows for the NSH traffic steering architecture.

Uplink Packets

Figure 2: Uplink Packet Flow

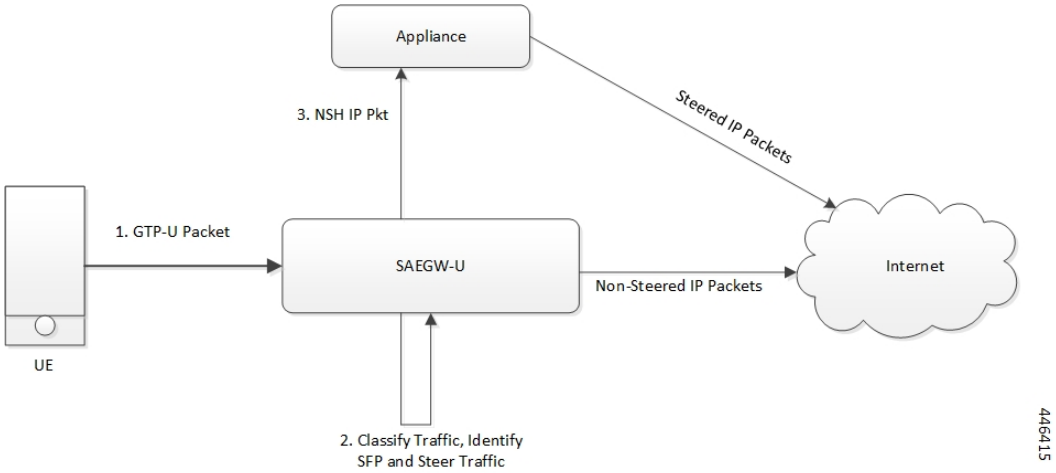


Table 2: Uplink Packet Flow Description

Steps	Description
1	UE sends the subscriber data packets to SAEGW-U.
2	SAEGW-U classifies the subscriber data traffic that is based on subscriber policies, and identifies an SFC to select an SFP accordingly.
3	SAEGW-U steers the Uplink (UL) packets with NSH encapsulation as per NSH RFC and sends to NSH appliance. SAEGW-U sends the non-steered IP packets to the server.
4	NSH supported appliance on receiving uplink packet, takes the decision to forward the packet to server based on certain criteria.

Downlink Packets

Figure 3: Downlink Packet Flow

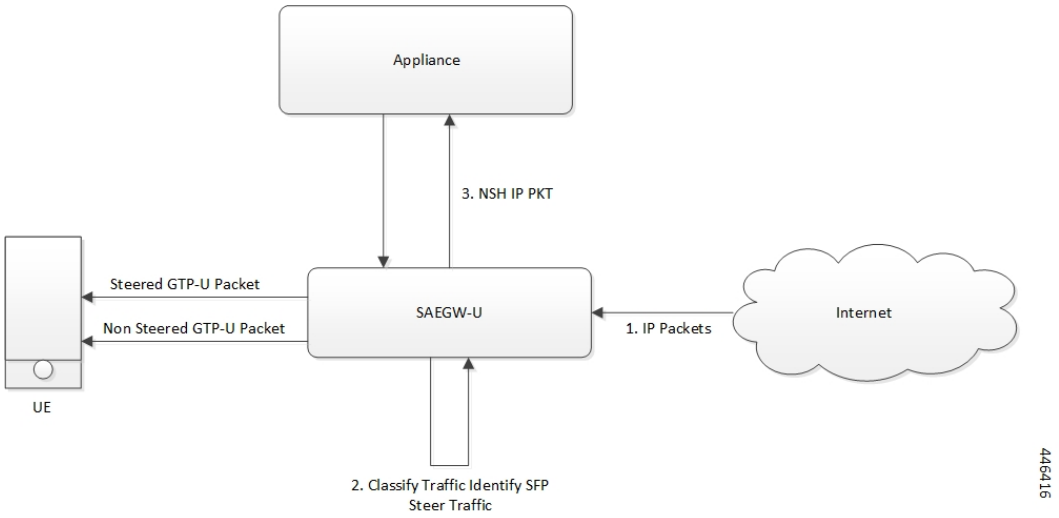


Table 3: Downlink Packet Flow Description

Steps	Description
1	SAEGW-U receives the Downlink (DL) packets from the server.
2	SAEGW-U selects an SFP.
3	SAEGW-U adds metadata as NSH context header and forwards it to the NSH supported appliance.
4	The NSH supported appliance sends back the packets to the SAEGW-U with the some metadata tags,as sent by SAEGW-U.
5	On receiving the packets, SAEGW-U classifies the subscriber data traffic that is based on subscriber charging policies.
6	SAEGW-U sends the data packets to the subscriber.

NSH Traffic Steering Requirements

Following is the behavior for integration of NSH appliances in the Traffic steering solution:

- SAEGW-U maintains the session stickiness of NSH appliance and ensure that all flows of a subscriber session end up selecting the same appliance instance.
- There’s a configurable option to define the load capacity for every appliance instance, example 50%, 100%. If the load status by the NSH appliance exceeds this threshold, only existing subscribers can continue with such instance. This instance doesn’t allocate to any new subscriber until the load status falls below the threshold.

- If NSH appliance detects as DEAD, all traffic on SFPs engaging this appliance instance is reclassified and traffic moves to a different appliance instance. Such appliance isn't available for new subscriber selection once it comes back ALIVE.
- Traffic Steering can be enable/disable in midsession. If you enable the traffic steering in between, then it's applicable to new flows. Old flows continue without traffic-steering.
- SR/ICSR support for traffic-steering Post SR/ICSR session stickiness is maintained.
- In case of multi appliance SFP, there are two forms of configurations:
  - For cases where appliances need to see start of traffic (example - TWH Packets), an SFP is selected which engages all appliances. As per the configuration policies, when the classification happens, the traffic can fall out of ineligible appliances.
  - For cases where appliances engage in mid flow, the configuration is such that appliances engage once the certain appliances become eligible further to traffic classification.
- Traffic steering statistics indicate the packets which are candidate for traffic steering. For traffic steering statistics, those packets are also counted which are dropped by quota exhaust, though they are the candidates for traffic steering.
- When node failure is done and continuous data is coming, then there can be discrepancy in steering statistics. Data steered on SFPs which is going down is not reflected in statistics.
- If you want to modify the NSH remote IP add or SRC bind IP in the configuration for any NSH appliance instance:
  - Then remove the instance.
  - Then remove the SFPs associated with it.
  - Put the SFPs and new instance with modified IP addresses.
  - Perform the commit afterwards.

This feature supports the following Traffic steering system limits:

Traffic steering object	Max Limit
Total Appliance groups	16
Totsl Instances per Appliance Group	256
Total SFCs	16
Total SFPs	64000

Default Service Chain

For operator, there could be certain use cases, where all traffic for a subscriber who has traffic steering enabled, needs to traverse through certain appliance(s). In order to cater to such requirement while providing an easy configuration mechanism to achieve that, the concept of a default service chain has been brought in. For e.g. if the subscriber is engaged on a subscriber with 2 appliances, APP1 and APP2 , where APP2 needs to see all the traffic, a service chain containing APP2 would be configured as the default service chain.

Thus, for a traffic steering enabled subscriber, there could be unavailability of service chain APP1+APP2 for certain traffic due the following conditions:

- There is no suitable policy configured for certain flows which would select the APP1+APP2 service chain.
- APP1+APP2 service chain was selected ,but APP1 instances went down below the min instance threshold. In such case the APP1+APP2 service chain will not be available.
- APP1+APP2 service chain was selected but no SFP could be selected.

Under such cases due to service chain unavailability, the flows would fall back to the configured default service chain thus ensuring APP2 service treatment to the flows.

If a default service chain, however, if not configured, will lead to the traffic being sent out non-steered.

SFP Selection

SFP selectios is based on the:

- MSISDN Stickiness (preconfigured) or
- Load Availability

MSISDN Stickiness

MSISDN Stickiness depends on the MS-ISDN and it provides the corresponding node. If the node is available and is part of the SFP, then that SFP is selected for the data (UL/DL). Presently, MSISDN stickiness is available for the L2 nodes only and there can be a service chain having L2 nodes alone or with a mix of L2 and NSH. All SFPs of the service chain have same set of type of nodes, where type can be of L2 or L2 + NSH or NSH (only).

Subscriber Stickiness (for both L2 and NSH) is maintained for the subscriber across the service chains till that node is available and when node goes down or removed from the config, subscriber can move to a different SFP (based on SFP selection). In case of stickiness miss, logs and traps are generated.

Load Availability

Load availability is load capacity, current load is maintained for each SFP (minimum of all instances that are part of the SFP). The SFPs are classified as part of available, overloaded or blocked list based on load availability. Only available-list and overloaded-list are being used for SFP selection as blocked-list is for SFPs for which node is down. Available-list SFPs are available for both old and new calls/sessions. Overloaded-list (load availability =0) is only used for maintaining the stickiness (if any), that is for old calls/sessions only. SFPs, once selected may move to overloaded-list because of load and for maintaining the stickiness. Same SFPs are used for the old calls/sessions and new calls use the remaining SFPs of the available-list for SFP selection.

Interworking with Inline Features

Support for interworking with the following inline features is not in the scope of the existing implementation.

- IPv4/v6 Readdressing
- NAT44 and NAT64

- Next Hop Forwarding
- L2 Marking

The encoding of rating group in the NSH context header is supported aligned with the following expected behaviour:

- The encoded rating group value corresponds to the rule that each packet matches. So, in a single flow's packets, the rating group either changes or is not encoded as the flow moves across different rules with different rating groups configured/or not configured.
- The SAEGW populates the rating group value, if configured, in the rating group field. If only content id is configured then this value is populated in the field. In the event that none are associated with the packet's matching rule, no TLV field corresponding to the rating group is sent.
- In case SAE-GW performs a deferred rule match and send out the packets without a rule match, it doesn't encode the rating group TLV for such packets.

Configuring the L2 and NSH Traffic Steering Feature—Standalone Mode

The following sections provide information about the CLI commands available to configure the L2 and NSH traffic steering CUPS feature in both CP and UP.

Configuring the Control Plane

Perform the following steps to configure the CP:

1. The following CLI command is a sample configuration to configure CP under the active-charging service.

```
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
```

NOTES:

- **subs-scheme**: The name should match the subscription-scheme AVP value that is received from PCRF over the Gx interface.
- **up-service-chain SecNet**: The value must match the up-service-chain that is configured on UP.
- **rule-name**: The value can be static/predef/gor/dynamic rules.

2. Traffic steering AVPs are currently supported with the Diameter dictionary custom44. The Diameter dictionary enables CP to properly decode the TS-related AVPs when they are received over the Gx interface and sent in Sx message to UP.

   The following is an example configuration to configure the Dictionary in CP.

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

Following are the sample values for TS-related AVPs received over GX in CCA-I/CCA-U/RAR.

```
     [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
```

Configuring the User Plane

Perform the following steps in same sequence to configure the UP:

The following CLI command is a sample configuration to add an interface in the contexts, which are used to send data toward L2 and NSH supported appliance.

1. Add the interface in the contexts which will be used to send data toward the L2 and NSH supported appliance.

   The following is a sample configuration:

```
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. Bind these newly-added interfaces to the physical ports of the UP.

   The following is an example configuration:

```
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. Add the TS-related configuration in the UP.

   The following is an example configuration:

```
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. Verify the above configurations using **show configuration** CLI command. Then, execute the **commit** CLI command for the configurations to be effective.

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

Configuration Guidelines

This section describes the following guidelines that are required to properly configure the feature:

- Configure the TS-related configuration on UP in the same sequence as mentioned in the preceding sections. This method ensures that the interfaces used to steer traffic toward L2 are applied properly in the configuration.
- If the instance under up-appliance-group has to be modified or deleted, then all the associated sfp-id under up-service-chain must be removed or deleted first.
- If the preceding modification must be done to the associated instance and sfp-id after a call is initiated, then remove the sfp-ids and reconfigure them to avoid any issues.
- Apply any changes to the interface before configuring the up-appliance-group instance. If the changes to the interface are applied at a later stage, remove the up-service-chain configuration first and then the up-appliance-group configuration. After the interface modification is complete, reconfigure the service chain and appliance group.
- The entire UP service chain and appliance group must not be removed to remove an interface or sfpid.

N to M Traffic Steering

Following are the configuration steps for the N:M Traffic Steering:

1. Configure TS-bind ip in RCM host specific configuration for all active UPs.
2. Configure the required active charging ruledef, rulebase configurations and traffic steering configurations (up-nsh format, up-appliance-group and up-service-chain, commit CLI) in common configuration in RCM and do commit.
3. Reload the active and standby UP with Day-0 config which has require ts-mon, RCM config, node monitor CLI for L3 server monitoring, BFD related interfaces configuration for L2, and service schema config for traffic steering (trigger condition, trigger action and so on).
4. Check RCM pushes config to all UPs. Check all services are up on all the UPs.
5. Check that the VPP fastpath tables have SST, SSMT, and SST tables created. Also check global tables are created correctly.
6. Check the up-service-chain SFP status and make sure that the SFPs are in available state.

Configuration

Following are the sample configurations:

- **Day-0 config** : The following configurations are part of Day-0 config.
  - Require ts-mon and Node-monitor CLI to monitor L3 appliance as mentioned in the earlier configuration section. Each UP has its own physical IP to monitor L3 appliance.
  - BFD related interfaces configuration for L2. Vlan configuration and IP interface related configuration.
  - Service schema configuration (Trigger condition, service scheme and so on).

Note Optimisation is planned to move service schema configuration to common configuration. Currently if service scheme configuration needs to be modified then changes needs be done manually on all the UPs.

UP Sample Configuration

L3 Monitoring

```
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
```

Note on UP2, IP can be 40.40.40.454, this is physical IP address specific to that UP.

L2 Monitoring:

```
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
```

One sample interface configuraition to bind all interfaces to port and vlan.

```
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
```

service schema configuration:

```
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
```

- **Host Specific configuration**: The following configurations is the part of the host specific configuration.
  - TS-bind IP configuration for each ACTIVE UP is the part of host specific configuration on RCM.

```
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"
```

Note TS-bind IP is a loopback IP address. Its physical IP address is the part of Day-0 configuration.

- **Common configuration**: The following configuration is the part of the common configuration.
  - Traffic steering configuration (up-nsh format, up-appliance-group, and up-service-chain config).

Note Assuming that the ingress is configured with low vLAN, for uplink data flow, the packets are sent to SN at the ingress vLAN Id and received from SN at the egress vLAN Id. Similarly, for the downlink data flow, the packets are sent to SN at the egress vLAN Id and receive from the SN at the ingress vLAN Id.

```
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
```

Show CLI for Verification

Following are the show CLIs for User Plane and RCM:

- User Plane: **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
```

Monitoring and Troubleshooting—Standalone Mode

This section describes how to monitor and troubleshoot this feature.

Show Commands for Control Plane

This section describes the available show command to monitor this feature on CP.

show active-charging sessions full all



Note *TS Subscription Scheme Name*: Displays the subscription scheme that must be applied from the service-scheme configured under the active-charging-service. This active-charging-service is received from PCRF over the Gx interface.

Show Commands for User Plane

This section describes the available show commands to monitor this feature on UP.

Show Commands for Configuration

This section describes the available show commands to check configuration for the feature.

- **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 Commands for Data Statistics

This section describes the available show commands to check data statistics related to the feature.

- **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*

Show Commands to Check the Service Chain and SFP Association for TS:

This section describes the available show commands to check the service chain and SFP association.

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

Show Commands for OAM Statistics

This section describes the available show commands to check OAM statistics related to the feature.

- **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**

Currently BFD doesn't provide an API to clear session stats, so the following traffic-steering OAM clear command is extended to include l2-steering stats.

- clear user-plane-service traffic-steering
  - OAM - Clears OAM
  - statistics - Clears the User-Plane Traffic-steering Statistics
- clear user-plane-service traffic-steering OAM
  - L3-steering - Clear L3-steering OAM
  - statistics - Clears OAM statistics

Show Configuration Command

The following configuration is a snippet of a sample **show configuration** command for this feature.

```
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
```

Show Command for User Plane 1:1 Redundancy

show srp checkpoint statistics | grep ts-sfp

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

Show Commands for SFP availability

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

SNMP Traps

The following SNMP Traps are added in support of this feature:

- UPlaneTsMisConfig : When there is no SFP that is associated with an appliance group.
- UPlaneTsNoSelectedSfp : When an SFP selection is not possible.
- UPlaneTsServiceChainOrApplianceDown : When a service chain or an application node becomes unavailable. The service chain is unavailable when the minimum instance of application group becomes unavailable.
- UPlaneTsServiceChainOrApplianceUp : When the node status of appliance is updated because the service chain or application node instance becomes available.

Bulk Statistics

Up-service-chain Schema

Variable Name	Data Type	Counter Type	Description
up-svc-chain-name	String	Info	Name of up service chain
up-svc-chain-status	Int32	Info	Status of up service chain
up-svc-chain-load-status	Int32	Gauge	Load status of up service chain
up-svc-chain-sfp-stickiness-miss-count	Int32	Counter	SFP stickiness miss count of up service chain
up-svc-chain-sfp-not-selected-count	Int32	Counter	SFP not selected count of up service chain
up-svc-chain-associated-calls	Int32	Gauge	Associated calls of up service chain
up-svc-chain-associated-flows	Int32	Gauge	Associated flows of up service chain
up-svc-chain-total-uplink-pkts	Int64	Counter	Total Uplink packets of up service chain
up-svc-chain-total-uplink-bytes	Int64	Counter	Total Uplink bytes of up service chain
up-svc-chain-total-downlink-pkts	Int64	Counter	Total Downlink packets of up service chain
up-svc-chain-total-downlink-bytes	Int64	Counter	Total Downlink bytes of up service chain

Up-appliance-group Schema

Variable Name	Data Type	Counter Type	Description
up-appl-group-name	String	Info	Name of up Appliance Group
up-appl-group-status	Int32	Info	Status of up appliance group
up-appl-group-load-status	Int32	Gauge	Load status of up appliance group
up-appl-group-node-down-count	Int32	Counter	Node down count of up appliance group
up-appl-group-associated-sfps	Int32	Gauge	Associated sfps of up appliance group
up-appl-group-num-times-loaded-state	Int32	Counter	Number of times node down state of up appliance group
up-appl-group-total-uplink-pkts	Int64	Counter	Total Uplink packets of up appliance group
up-appl-group-total-uplink-bytes	Int64	Counter	Total Uplink bytes of up appliance group
up-appl-group-total-downlink-pkts	Int64	Counter	Total Downlink packets of up appliance group
up-appl-group-total-downlink-bytes	Int64	Counter	Total Downlink bytes of up appliance group

The following CLI command is a sample bulkstats configuration for the feature.

```
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"
```

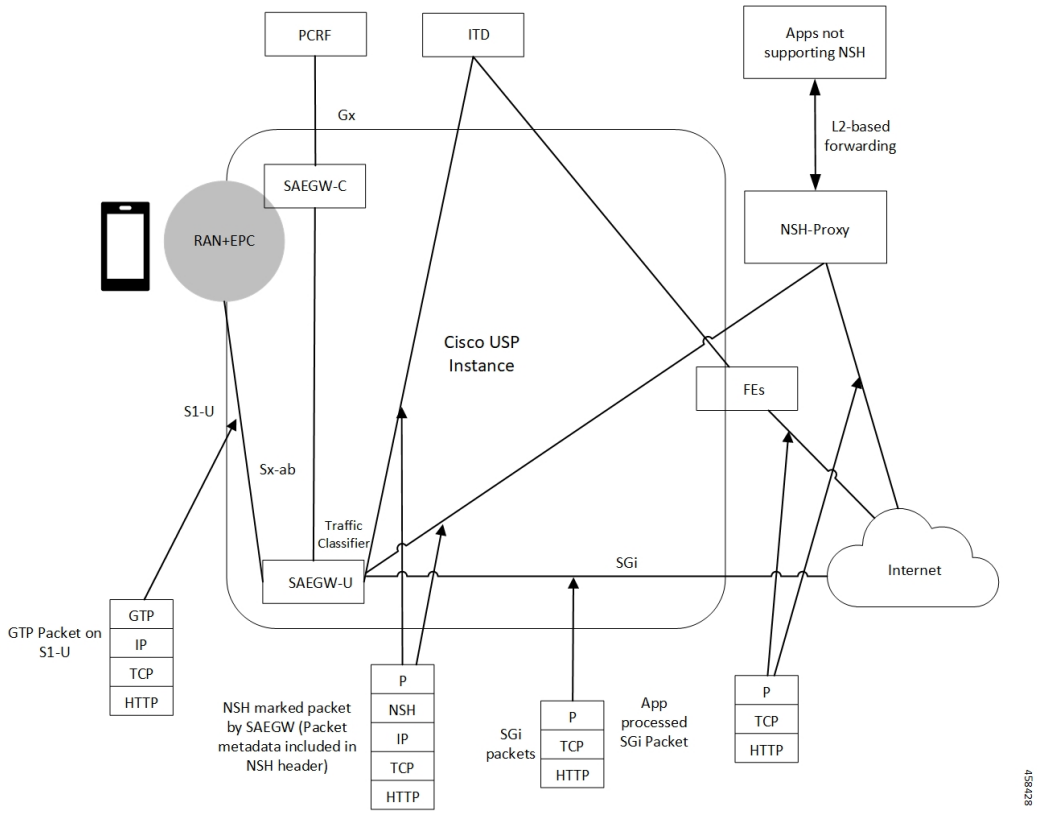
Feature Description—Sandwich Mode

The Sandwich Mode caters to the NSH-based Traffic Steering (TS) approach to provide the metadata needed by the service function appliance's Forwarding Engine (FE) nodes.

The Sandwich Mode solution leverages the Cisco Nexus 9000 Series NX-OS Intelligent Traffic Director (ITD) in the Cisco USP instance. For more details about ITD, refer the *Cisco Nexus 9000 Series NX-OS Intelligent Traffic Director Configuration Guide*.

Architecture—Sandwich Mode

The following figure illustrates the integration of an external service function appliance with Cisco’s SAEGW-U (User Plane).



The Sandwich Mode solution includes the following functionality:

- SAEGW-U adds the relevant NSH-Based-Metadata onto the relevant packets exiting the Gi path only in the Uplink direction.
- The ITD, running in Sandwich Mode may load-balance these packets (based on source-IP) to the FEs.
- SAEGW-U doesn't perform any health checks toward the FEs or aware of its existence.
- The ITD node may maintain the “stickiness” at a session level. The ITD does so by looking at the NSH-Outer-IP-SRC-Header.
- In the Uplink direction, the source IP is the "UE-IP" (Copy of Inner IP header). The destination IP is the "server-IP-internet".
- In the Downlink direction, there's no NSH Header and the packet straight away goes from the Internet into the FEs. At SAEGW-U, the source IP is the "Server-IP", and the destination IP is the "UE-IP".
- SAEGW-U performs the traffic classification and selects the service chain for a given flow.
- Service chain at SAEGW-U can include more than one appliance, and the steering functions can handle these appliances.
- SAEGW-U encodes only the NSH Header on Uplink packets.
- SAEGW-U copies the source IP details directly from the original UE-IP Header. SAEGW-U uses NSH Port 6633 for outer header SRC and DEST Port. The destination IP is the Appliance IP (as configured).
- On receiving Downlink packets with NSH header, the SAEGW-U drops such packets.

- SAEGW-U doesn't perform any health checks for the FEs or the ITD. The SAEGW-U treats the ITD as always available.
- SAEGW-U encodes all Uplink packets (qualified by the service function appliance) towards ITD with NSH Base Header, Service Headers, and Context Headers (with Metadata).
- TS App works only in one mode (either Sandwich mode or Standalone mode) at a time.

Note

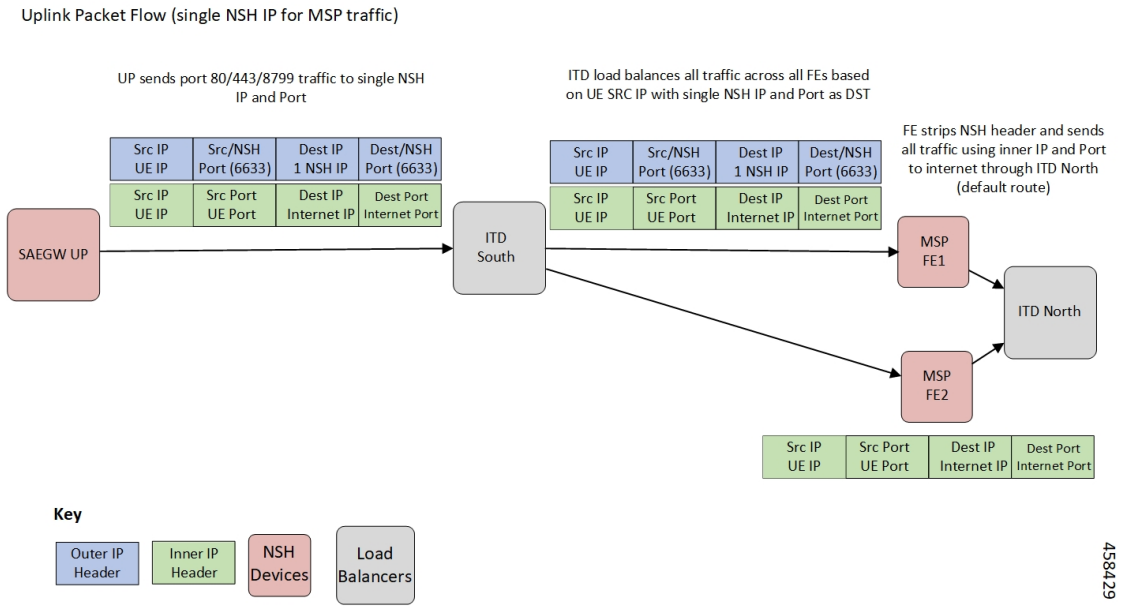
- For Sandwich mode, the **require tsmon** CLI command must not be configured.
- Changing from Sandwich to Standalone mode and conversely, requires a reboot.

How it Works—Sandwich Mode

Packet Flows in Sandwich Mode

Uplink Packets

The following figure illustrates the Uplink packet flow.



The following describes the packet flow:

1. GTP-U packet arrives at SAEGW-U. It decapsulates the GTP header and identifies the subscriber for the flow.

2. SAEGW-U performs traffic classification and associates a service chain for the flow. The SAEGW-U is configured to associate a service chain containing the service function appliance (ITD), with traffic classified depending on TCP/UDP/HTTP/HTTPS.
3. SAEGW-U looks up for NSH format associated with the service chain for encoding the parameters in the NSH variable header to be sent to the service function appliance.

   The following is an example of NSH Header with SFP selected for the Uplink packet is 200.

```
**************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)

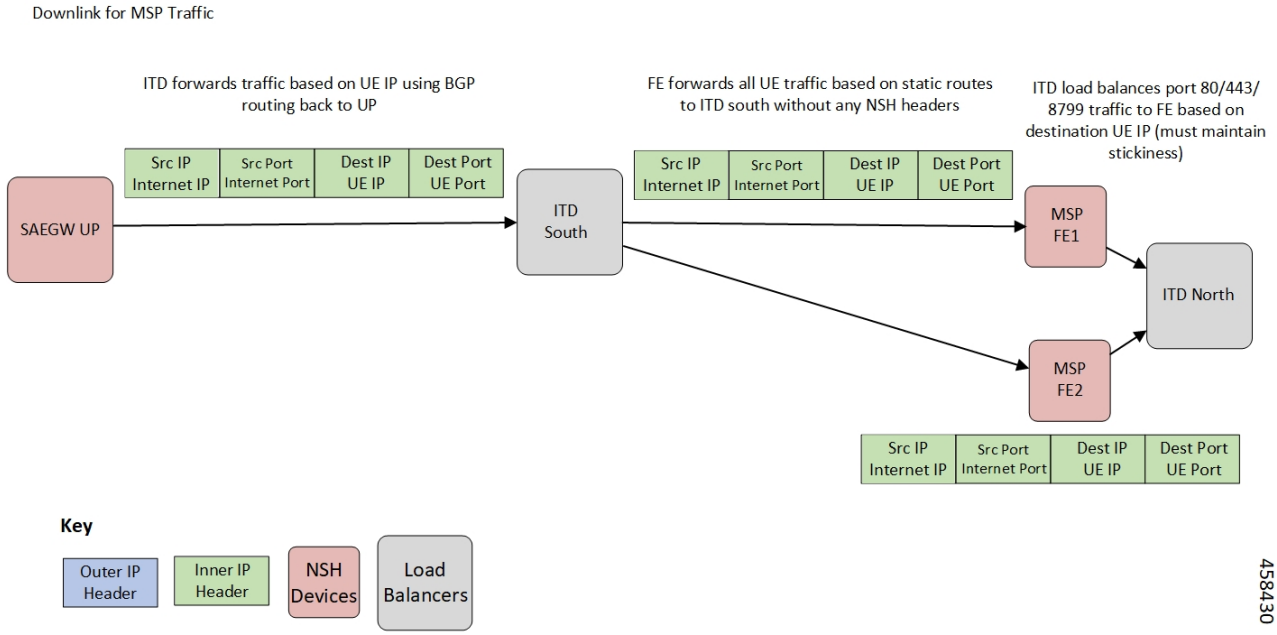
                   TLV Type: <Sub Profile tag configured in UP>
                   TLV Len: 32
                   TLV Value: Profile-1 (Sub Profile name)

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


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

Downlink Packets

The following figure illustrates the Downlink packet flow.



The following describes the packet flow:

1. Packets flow directly from internet server to the FEs. The FE processes the packets and sends it to the SAEGW-U.

   The SRC IP/Port is the server IP/Port and the DEST IP/Port is the UE IP/Port.

2. The SAEGW-U processes the packet, and if there are more service function appliances in the service chain, sends the packet for further processing. If the service chaining is complete, the packet is sent to normal Downlink packet processing path for Rule matching/classification and charging.

3. The SAEGW-U encapsulates the packet with GTP-U header and sends it across to the UE.

Note Downlink packets must not be NSH encoded. Otherwise, SAEGW-U will drop all such packets.

TCP and UDP Traffic

Uplink Traffic

- All TCP and UDP traffic qualified for steering towards the appliance is treated alike.
- UL packets are steered to the appliance with configured NSH context header elements. The NSH Service header is encoded with SI=1. Therefore, further to SI deduction and with SI=0, packet is sent over the Gi interface.
- The outer headers SRC IP is the same as the inner headers SRC IP (that is, UE SRC IP).
- The outer headers SRC Port is NSH port 6633.
- The outer headers DST IP is the configured Appliance IP.
- The outer headers DST PORT is the NSH port 6633.

Downlink Traffic

Downlink packets are received from FEs through ITD and therefore, processed as normal IP packet without being steered toward the FEs.

- UL packet received at the SAEGW-U is classified and based on the configured policy associated with the appropriate SFC.
- The SAEGW-U performs the SFP selection based on the service and load availability of the appliance instances and selected steer. The Uplink traffic is NSH (IP-UDP) encapsulated and steered on the selected SFP with the context header populated as deemed necessary.
- The NSH appliance on receiving the NSH packet, processes the IP packet (and possibly the context header), and sends the packet over the Gi interface.
- Downlink packet is sent by the destination server over the Gi interface to the SAEGW-U.

Service-Scheme Selection for Traffic Steering

You can select service-scheme in one of the following two ways:

1. Gx/PCRF:

   PCRF enables Traffic Steering through the following AVPs:

   ```
   [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
   ```

   TS Profile and TS Subscriber Scheme are then sent to User Plane through Sx messaging:

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

       TS-Subscriber-Scheme: gold
   ```

   For Gx/PCRF based Traffic Steering, the **trigger subs-scheme-received** CLI command is required in the service-scheme configuration.

2. Service-scheme framework (without Gx/PCRF AVPs):

   Traffic Steering can be enabled without Subscription-scheme AVP from PCRF.

   The **trigger sess-setup** CLI command is required with trigger-action pointing to the **up-service-chain**. The following is an example configuration:

   ```
   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
```

Default Service Chain

For a TS-enabled subscriber, the following conditions can cause unavailability of service chain (APP1+APP2) for certain traffic:

- There's no suitable policy configured for certain flows which would select the APP1+APP2 service chain.
- APP1+APP2 service chain was selected, however, APP1 instances went down below the minimum instance threshold. In such case, the APP1+APP2 service chain won't be available.
- APP1+APP2 service chain was selected, however, no SFP could be selected.

Under such cases of service chain unavailability, the flows fall back to the configured default service chain and ensuring APP2 service treatment to the flows.

If a default service chain isn't configured, it leads to the traffic being sent out nonsteered.

For TS-enabled through Gx/PCRF, the default service chain is defined through **trigger subs-scheme-received**.

For TS-enabled through service scheme framework without Gx/PCRF AVPs, the default service chain is defined through **trigger sess-setup**.

SFP Selection

For service chains with only NSH-based appliance:

For Downlink packets, there's no NSH appliance and so, there's no SFP.

For service chains with a mix of L2 and NSH-based appliances:

Any SFP is selected based on L2 "stickiness". Same NSH-based appliance is present, and always available for SFP selection.

For Downlink packets, the SFP selection is based only on L2 appliance.

There's no SFP selection based on Load availability of NSH-based appliance. The NSH/appliance is considered as always-available.

Limitations and Restrictions

The following are the known limitation/restrictions of the feature:

- Changing from Standalone mode to Sandwich mode and vice versa, requires a reload and configuration change.
- When Traffic Steering is enabled from PCRF or locally using the service-scheme framework, then Traffic Steering can't be disabled on that session.
- For multi appliance service chain (L2 and L3 steering), the SFPs for V4 and V6 traffic are different. However, both SFPs maintains the L2 appliances MSISDN based stickiness.

Configuring NSH Traffic Steering—Sandwich Mode

This section provides information about the CLI commands available to configure NSH Traffic Steering—Sandwich Mode in both CP and UP

CP Configuration

Perform the following steps to configure the CP:

1. Configure the Active Charging Service configuration.

   The following is an example configuration:

```
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. Traffic steering AVPs are currently supported with the Diameter dictionary custom44. The Diameter dictionary enables CP to properly decode the TS-related AVPs when they are received over the Gx interface and sent in Sx message to UP.

   The following is an example configuration to configure the Dictionary in CP.

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

The following are the sample values for TS-related AVPs received over Gx in CCA-I/CCA-U/RAR.

```
     [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 Configuration

Perform the following steps in same sequence to configure the UP:

1. Add the interface in the contexts which will be used to send data toward the Service chain appliances.

   The following is an example configuration:

```
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. Bind these newly-added interfaces to the physical ports of the UP.

   The following is an example configuration:

```
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. Add the TS-related configuration in the UP.

   The following is an example configuration:

```
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
```

NOTES:

- **ts-bind-ip** *name* **ue-src-ip { ipv4-address** *ipv4_address* **| ipv6-address** *ipv6_address* **}**: Specifies the IP address of the UP interface from which packet is sent out toward ITD.

4. Verify the above configurations using **show configuration** CLI command. Then, execute the **commit** CLI command for the configurations to be effective.

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

Configuring Post Processing Ruledef in Both Standalone and Sandwich Mode

up-service-chain trigger action is used with trigger condition in the configuration of post processing a ruledef in the rulebase for steering the traffic. A single post processing ruledef is defined with port numbers for HTTP, HTTPS and other protocols even when there are multiple charging ruledefs. This single post processing ruledef name is matched in the trigger condition which is used in traffic steering.

Use the following configuration to configure post processing of ruledef for steering traffic:

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

Use the following configuration to configure the trigger condition in post processing ruledef:

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

Configuring BFD Instance Id Using Interface Name in UP Appliance Group

During traffic steering, in the **up-appliance-group**, the BFD instance id is configured using the interface name and IP configuration.

Use the following configuration to configure BFD instance id for steering traffic:

```
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
```

Note

- For any given L2 **up-appliance-group**, the BFD instance id is configured using the IP address or the **interface-name** for the particular **ingress** or **egress** using the corresponding interface names.
- Once the **up-appliance-group** configuration is complete for BFD monitoring using the **interface-name**, the BFD registration takes upto five minutes to complete.
- Once the BFD registration is successful, the IP address and the **interface-name** will be available in the **show user-plane traffic-steering up-appliance-group all** output.
- In case the IP address changes for any **interface-name** used in the **up-appliance-group** with BFD monitoring, then the **up-appliance-group** must be reconfigured.

Monitoring and Troubleshooting the NSH Traffic Steering—Sandwich Mode

This section provides information about the CLI commands available for monitoring and troubleshooting the feature.

For details about SNMP Traps, refer SNMP Traps, on page 25 section of this chapter.

For details about Bulk Statistics, refer Bulk Statistics, on page 25 section of this chapter.

Show Commands

This sections provides information about the show CLI commands that are available in support of the feature.

CP Commands

Use the following show CLI command in CP to monitor and troubleshoot the feature: **show active-charging sessions full all**

TS Subscription Scheme Name: Displays the subscription scheme that must be applied from the service-scheme configured under the active-charging-service. This active-charging-service is received from PCRF over the Gx interface.

UP Commands

Use the following show CLI commands in UP to monitor and troubleshoot the feature.

- Traffic Steering configuration check
  - **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**

- Traffic Steering statistics
  - **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**

- Service chain and SFP association
  - **show subscriber user-plane-only flows**
  - **show subscribers user-plane-only callid** *call_id* **flows**

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

Use the following show CLI command to monitor and troubleshoot the feature.

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