

Network Service Headers (NSH)

This chapter describes the following topics:

- Feature Summary and Revision History, on page 1
- Feature Description, on page 2
- How It Works, on page 2
- Configuring Support for NSH Framework, on page 4
- Show Commands and Outputs, on page 11

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	• P-GW • SAEGW
Applicable Platform(s)	• ASR 5500 • VPC-DI • VPC-SI
Feature Default	Enabled - Always-on
Related Changes in This Release	Not Applicable
Related Documentation	• *Command Line Interface Reference* • *P-GW Administration Guide* • *SAEGW Administration Guide*

Revision History

Revision Details	Release
In this release, NSH-based Traffic Identification with Traffic Steering is supported.	21.9
First introduced.	21.4

Feature Description

Network Services Headers (NSH), a new service chaining protocol, is added to the network traffic in a packet header to create a dedicated service plane that is independent of the underlying transport protocol. In general, NSH describes a sequence of service nodes that a packet is routed through before reaching the destination address. The NSH includes meta-data information about the packet and service chain in an IP packet. The NSH protocol addresses the growing requirement to deploy various services functions external to the gateway.

This feature introduces NSH protocol support for P-GW and SAEGW products and supports the following:

- Encoding and decoding of NSH format in the P-GW/SAEGW.
- Configurable parameters to be included for encoding in the variable header.
- NSH treatment for selective traffic based on configuration.
- Configuring the tag values for parameters present in the variable header.
- Selective configuration of policies for acting on the decode parameters received in the NSH.
- Configuring the intelligence of encoding the NSH information in every packet of a flow or only once per flow.
- NSH-based Traffic Identification with Traffic Steering.



Important In this release, selective encryption of parameters is not supported.

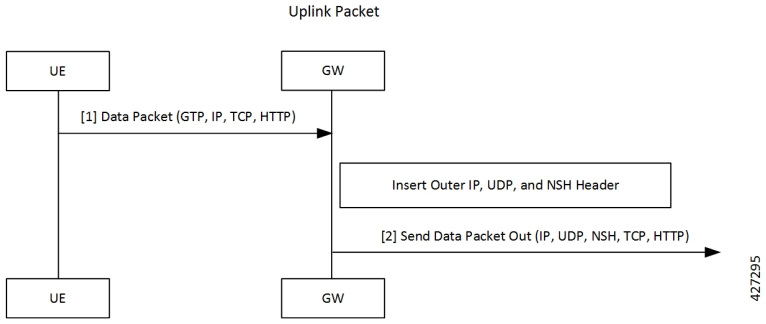
How It Works

This section describes the working of NSH protocol support in Cisco's P-GW/SAEGW products.

- **The Uplink Packet**

  For the uplink packet, P-GW/SAEGW adds the NSH, if the flow matches the specified criteria. NSH has a variable length context header also.

  Following call flow shows the NSH protocol support in the Cisco PGW/SAEGW products for an uplink packet.

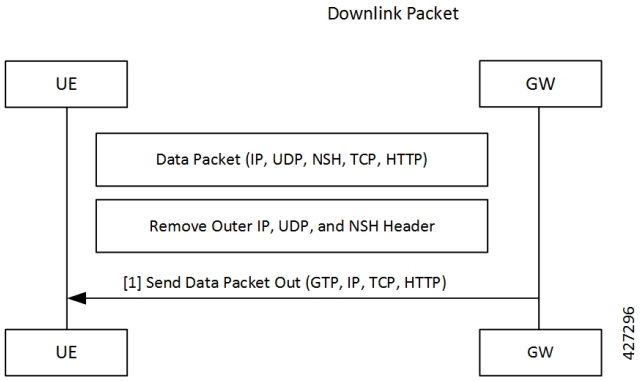


For an uplink packet, if the call flow matches the specified criteria, PGW or SAEGW adds the NSH header to the data packet. NSH header may have variable length context header, which can be encrypted if specified in the configuration.

- **The Downlink Packet**

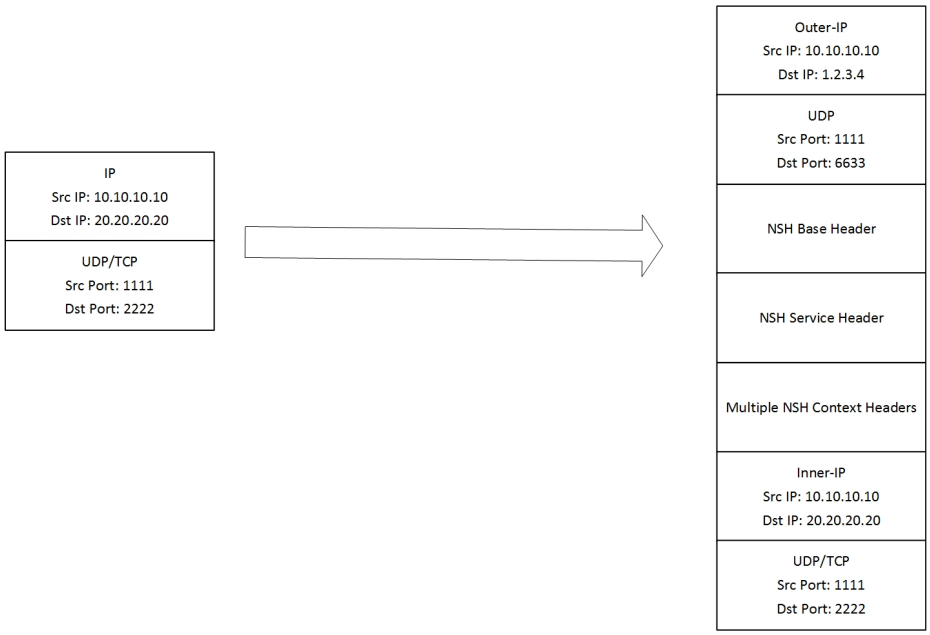
  For the downlink packet, P-GW/SAEGW processes and removes the NSH and applies policies based on the extracted NSH parameters.

  Following call flow shows the NSH protocol support in the Cisco PGW/SAEGW products for a downlink packet.



  For a downlink packet, PGW or SAEGW processes and removes NSH header. Then, PGW or SAEGW apply policies based on the extracted NSH parameters.

- Source and destination IP address for the outer IP packet is taken from the inner IP packet.
- By default, NSH encapsulated packets use the port number 6633.



Configuring Support for NSH Framework

This section covers configuration steps used in this feature for adding support for NSH framework.

Charging Action Association

Service chain is associated to charging action in the following way:

```
configure
  active-charging service service_name
    charging-action charging_action_name
      service-chain service_chain_name
      end
```

Notes:

- **charging-action**: Defines charging action.

  charging_action_name: Specifies name of the charging action. This is entered as an alphanumeric string of 1 through 64 characters.

- **service-chain**: Defines service chain association.

  service_chain_name: Specifies name of the service chain. This is entered as an alphanumeric string of 1 through 64 characters.

Service Chain Association

A new CLI command **nsh-format** is added to the **service-chain** command for service-chain association.

```
configure
   service-chain <service_chain_name>
      nsh-format <nsh_format_name>
      end
```

Notes:

- **service-chain**: Defines service chain association.

  service_chain_name: Specifies name of the service chain. This is entered as an alphanumeric string of 1 through 64 characters.

- **nsh-format**: Associates NSH format with the service chain.

Service Scheme Association

A new CLI command **nsh-response-received** has been added to the **trigger** command to the ACS service scheme configuration mode.

```
configure
  active-charging service service_name
     service-scheme service_scheme_name
        [ no ] trigger { bearer-creation | flow-create | loc-update |
nsh-response-received | sess-setup }
       end
```

Notes:

- **service-scheme**: Enables the association of service-scheme based on subscriber class.

  service_scheme_name: Specifies name of the service scheme. This is entered as an alphanumeric string of 1 through 64 characters.

- **no**: Disables the trigger action for the service-scheme.
- **trigger**: Specifies the trigger action for service-scheme.
- **bearer-creation**: Triggers for every new bearer.
- **flow-create**: Triggers for every new flow.
- **loc-update**: Triggers whenever location changes of the subscriber.
- **nsh-response-received**: Triggers on NSH response packet.
- **sess-setup**: Triggers at session setup.

NSH Configuration Mode

The Network Service Header (NSH) configuration mode is a sub-mode of the Global Configuration mode. This NSH mode is used to encode or decode NSH.

Exec > Global Configuration> Network Service Entity - IP Configuration

```
configure
   nsh
   end
```

Entering the above command sequence results in the following prompt:

[local]host_name(nsh)#

NSH Fields Configuration Mode

The NSH Fields configuration mode is a sub-mode of the NSH Configuration mode. This NSH Fields configuration mode is used to tag value to the NSH fields.

Exec > Global Configuration> Network Service Header > Network Service Header - Fields Configuration

```
configure
   nsh
     nsh-fields fields_name
     end
```

Entering the above command sequence results in the following prompt:

[local]host_name(nsh-nshfields)#

tag-value

This new CLI command is added to the NSH Fields Configuration mode to associate a tag value to a NSH field.

```
configure
  nsh
     nsh-fields fields_name
        tag-value tag_value { content-type | enterprise-id | imei | imsi |
msisdn | rating-group | rulebase | tdf-app-id }
        end
```

Notes:

- **nsh-fields**: Defines NSH fields tag values.

  fields_name: Specifies name of the *nsh-field*. This is entered as an alphanumeric string of 1 through 64 characters.

- **tag-value** Associates a tag to a field.

  tag_value : Tag value for the NSH field.

- **content-type**: Specifies content type of payload.
- **enterprise-id**: Specifies the enterprise-ID to be sent in NSH context header.
- **imei**: Specifies IMEI of the subscriber.
- **imsi**: Specifies IMSI of the subscriber.
- **msisdn**: Specifies MSISDN of the subscriber.

- **rating-group**: Specifies rating-group applied for the traffic.
- **rulebase**: Specifies rule-base of the subscribers.
- **tdf-app-id**: Specifies TDF Application ID applied to the traffic.

NSH Format Configuration Mode

The NSH Format Configuration mode is a sub-mode of the NSH Configuration mode. This NSH Format mode is used to encode or decode NSH.

Exec > Global Configuration> Network Service Header > Network Service Header - Format

```
configure
   nsh
     nsh-formatformat_name
     end
```

Entering the above command sequence results in the following prompt:

[local]host_name(nsh-nshformat)#

encode

This new CLI command is added to the NSH Format configuration mode. This command defines the NSH encoding fields to be associated with the NSH format.

```
configure
  nsh
     nsh-format format_name
        encode nsh-fields fields_name
        end
```

Notes:

- **nsh-format**: Defines format in NSH header.

  format_name: Specifies name of the *NSH format*. This is entered as an alphanumeric string of 1 through 64 characters.

- **encode**: Associates nsh-fields for encoding.
- **nsh-fields**: Defines nsh fields tag value.

  fields_name: Specifies name of the fields. This is entered as an alphanumeric string of 1 through 64 characters.

encoding-frequency

This command defines frequency of encoding the NSH fields to be associated with the NSH format.

```
configure
  nsh
     nsh-format format_name
```

```
encoding-frequency { always | once-per-flow }
end
```

Notes:

- **encoding-frequency**: Defines frequency of encoding nsh-fields.
- **always**: Encodes nsh fields on every hit.
- **once-per-flow**: Encodes nsh fields once per flow.

decode

This command defines the NSH decoding fields to be associated with the NSH format.

```
configure
  nsh
    nsh-format format_name
      decode nsh-fields fields_name
      end
```

Notes:

- **nsh-format**: Defines format in NSH header.

  format_name: Specifies name of the *NSH format*. This is entered as an alphanumeric string of 1 through 64 characters.

- **decode**: Associates nsh-fields for decoding.
- **nsh-fields**: Defines nsh fields tag value.

  fields_name: Specifies name of the fields. This is entered as an alphanumeric string of 1 through 64 characters.

Trigger Condition Configuration Mode Commands

content-type

This command specifies the content type to be matched.

```
configure
  active-charging service service_name
    trigger-condition trigger_condition_name
      content-type { operator condition }
      end
```

Notes:

- **trigger-condition**: Defines ACS trigger conditions.

  trigger_condition_name: Specifies name of the trigger condition. This is entered as an alphanumeric string of 1 through 64 characters.

- **content-type**: Specifies the content type.

- **operator** : Specifies how to match. Operator must be one of the following:
  - **!=**: not equals
  - **!contains**: not contains
  - **!ends-with**: not ends with
  - **!starts-with**: not starts with
  - =: equals
  - **contains**: contains
  - **ends-with**: ends with
  - **starts-with**: starts with
- **condition**: Specifies the condition to match. Condition must be one of the following:
  - FALSE
  - TRUE

tdf-app-id

This command specifies the identifier for application-based rules to be matched.

```
configure
  active-charging service service_name
     trigger-condition trigger_condition_name
        tdf-app-id { operator condition }
        end
```

Notes:

- **trigger-condition**: Defines ACS trigger conditions.

  trigger_condition_name: Specifies name of the trigger condition. This is entered as an alphanumeric string of 1 through 64 characters.
- **tdf-app-id**: Specifies the identifier for application based rules.
- **operator condition**: Specifies how to match. Operator must be one of the following:
  - **!=**: not equals
  - **!contains**: not contains
  - **!ends-with**: not ends with
  - **!starts-with**: not starts with
  - =: equals
  - **contains**: contains
  - **ends-with**: ends with

- **starts-with**: starts with

- **condition**: Specifies the condition to match. Condition must be one of the following:
  - FALSE
  - TRUE

Sample Configuration for NSH Creation

The following is a sample configuration for this NSH service creation:

```
config
       nsh
         nsh-fields xyz
            tag-val 1 imei
            tag-val 2 imsi
         exit
          nsh-fields abc
            tag-val 4 content-type
         exit
         nsh-format format1
            encoding frequency always
            encode nsh-fields xyz
            decode nsh-fields abc
          exit
    exit
  traffic-steering
  appliance-group firewall
   nsh-format format1
   ip address 1.2.3.4
  #exit
 #exit
 service-chain sch1
  sfp direction uplink service-index 1 appliance firewall
 #exit
 exit
config
    active-charging service ACS
        trigger-action ta1
           throttle-suppress
        exit
        trigger-condition tc1
           content-type contains text
        exit
        service-scheme scheme1
            trigger nsh-response-received
                priority 1 trigger-condition tc1 trigger-action ta1
            exit
        exit
        subs-class class1
            any-match = TRUE
        exit
        subscriber-base base1
            priority 1 subs-class class1 bind service-scheme scheme1
        exit
       charging-action ca1
           service-chain xyz
       exit
```

```
    exit
exit
```

Show Commands and Outputs

This section provides information regarding show commands and their outputs in support of the feature.

show nsh statistics

This command has been newly added in this release to display the nsh statistics. Following is the output when you execute this command:

```
Total Encap Successful                :         0
Total Decap Successful                :         0

Total Encap Failed                    :         0
   Memory Allocation                  :         0
   Config Error                       :         0
   Encryption Failed                  :         0

Total Decap Failed                    :         0
   Config Error                       :         0
   Base Header
      Invalid Length                  :         0
      Unsupported Version             :         0
      Unsupported Next Protocol       :         0
      Next Protocol Mismatch          :         0
      Unsupported MD-Type             :         0
   Context Header
      Unsupported MD-Class            :         0
      Unsupported Type                :         0

OAM Packets
   Received                           :         0
   Dropped                            :         0

Unknown Context Header Type           :         0
```

show active-charging trigger-condition statistics

The output of this command includes the following field for this feature:

- NSH-Rsp-Rcvd

This field displays the matching of trigger condition based on NSH response.