UPF Ingress Interfaces

- Feature Summary and Revision History, on page 1
- Feature Description, on page 2
- UPF Ingress Interface Type Configuration, on page 2
- Multiple GTPU IP Addresses Support, on page 8

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Product(s) or Functional Area	5G-UPF
Applicable Platform(s)	VPC-SI SMI
Feature Default Setting	Disabled – Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	*UCC 5G UPF Configuration and Administration Guide*

Revision History

Table 2: Revision History

Revision Details	Release
Added support for binding multiple GTPU IP addresses to ensure even distribution of traffic across the VPP worker threads on UPF.	2023.04.0

Revision Details	Release
Added support for IP separation per interface to allow separate networks for N3 and N9 interfaces.	2023.03.0 2023.02.0
First introduced.	2020.02.0

Feature Description

UPF supports GTP-U ingress interfaces to initiate the user plane service. The UPF supports different GTP-U ingress interfaces to allow a separate network for N3 interface and N9, S5u, and S8u interfaces. The N9, S5u, and S8u interfaces share the same public IP.

The supported GTP-U ingress interfaces include:

- **N3**: N3 is the interface between gNodeB and UPF.
- **N9**: The N9 interface connects two UPFs. It is the interface between intermediate I-UPF and UPF session anchor connecting different PLMN.
- **S5u**: S5u is the interface between SGW-U and PGW-U in the 4G.
- **S8u**: S8u is an inter-PLMN variant of the S5u interface.

The user plane service associates with the GTP-U service using the **associate gtpu-service** CLI command in User Plane Service configuration mode. To enable the **upf-ingress** command, you must configure the **require upf** command on the UPF. You need the UPF license to enable the **require upf** command.

Note UPF does not support multiple GTP-U service selections for a single call.

UPF Ingress Interface Type Configuration

Configuring UPF Ingress Interface Type

To configure the UPF ingress interface type, use the following configuration:

To enable the **upf-ingress** command, you must configure the **require upf** command on the UPF. You need the UPF license to enable the **require upf** command.

```
configure
  context context_name
    user-plane-service service_name
      [ no ] associate gtpu-service gtpu_service_name upf-ingress
interface-type [ n9 | s5u | s8u | n3 | n9-s5u-s8u ]
      end
```

NOTES:

- **associate gtpu-service** *gtpu_service_name*: Associate the GTP-U service with the user plane service.
- **upf-ingress**: Configure the interface type as UPF ingress.
- **interface-type [n9 | s5u | s8u | n3 | n9-s5u-s8u]**: Configure the desired GTP-U ingress interface type.
- If the N3 GTP-U ingress service is not present, UPF uses the default interface.
- If the configuration for N9, S5u, or S8u interfaces is not present, UPF performs the following actions:
  - UPF looks to use the N9 interface first followed by the S5u and S8u interfaces.
  - If there is no GTP-U service, UPF uses the default interface.

Configuration Example

The following is an example configuration:

```
associate gtpu-service n3-ingress upf-ingress
associate gtpu-service n3-ingress upf-ingress interface-type n3
associate gtpu-service n9-ingress upf-ingress interface-type n9
associate gtpu-service s5-ingress upf-ingress interface-type s5
associate gtpu-service s8-ingress upf-ingress interface-type s8
```

Configuring Echo Response for N9 Interface

The echo response in a GTP-U service provides the availability of peer UPF.

The following is a sample echo response configuration for the N9 interface in hUPF:

```
gtpu-service n9-ingress
   echo-interval 60
   echo-retransmission-timeout 1
   bind ipv4-address 60.60.41.5 bearer-type all
   exit
```

The following is a sample echo response configuration for the N9 interface in vUPF:

```
gtpu-service n9-egress
   echo-interval 60
   echo-retransmission-timeout 1
   bind ipv4-address 60.60.42.5 bearer-type all
   exit
```

Separating N3 and N9 BGP Routes

The following is a sample configuration to separate the N3 and N9 BGP routes for hUPF.

```
 config
  context SAEGW
    ip vrf mpls-vrf-1
    #exit
    ip vrf mpls-vrf-2
    #exit
    mpls bgp forwarding
    bfd-protocol
    #exit
    ip prefix-list name up seq 15 permit 50.50.17.4/32
    ip prefix-list name up seq 20 permit 50.50.17.5/32
```

```
ip prefix-list name up seq 25 permit 50.50.17.6/32
ip prefix-list name up seq 40 permit 50.50.17.7/32
ip prefix-list name n9-up1 seq 50 permit 60.60.17.5/32
ip prefix-list name n9-up1 seq 55 permit 60.60.17.6/32
route-map up_routes1 permit 10
  match ip address prefix-list n9-up1
#exit
route-map up_routes permit 10
  match ip address prefix-list up
#exit
router bgp 61435
  maximum-paths ebgp 4
  neighbor 20.14.35.100 remote-as 65000
  neighbor 20.14.35.100 ebgp-multihop
  neighbor 20.14.35.100 update-source 20.14.35.98
  neighbor 2001:4888:121:1435::1 remote-as 65000
  neighbor 2001:4888:121:1435::1 ebgp-multihop
  neighbor 2001:4888:121:1435::1 update-source 2001:4888:121:1435::2
  neighbor 20.14.40.100 remote-as 65000
  neighbor 20.14.40.100 ebgp-multihop
  neighbor 20.14.40.100 update-source 20.14.40.98
  address-family ipv4
    neighbor 20.14.35.100 max-prefix 8192 threshold 50
    neighbor 20.14.35.100 route-map up_routes out
    neighbor 20.14.35.100 send-community extended
    neighbor 20.14.40.100 max-prefix 8192
    neighbor 20.14.40.100 route-map up_routes1 out
    neighbor 20.14.40.100 send-community extended
    redistribute connected
    redistribute static
  #exit
  address-family ipv6
    neighbor 2001:4888:121:1435::1 activate
    redistribute connected
    redistribute static
  #exit
#exit
interface SxC loopback
  ip address 50.50.17.4 255.255.255.255
#exit
interface SxC-ipv6 loopback
  ipv6 address 2001:4888:50:50::17/128
#exit
interface gn-1
  ip address 20.14.35.98 255.255.255.0
  ipv6 address 2001:4888:121:1435::2/64 secondary
#exit
interface gn-2
  ip address 20.14.36.98 255.255.255.0
  ipv6 address 2001:4888:121:1436::2/64 secondary
#exit
interface gn-n9-ingress
  ip address 20.14.40.98 255.255.255.0
#exit
interface n9-ingress-hupf loopback
  ip address 60.60.17.5 255.255.255.255
#exit
interface pgw-gtpu loopback
  ip address 50.50.17.5 255.255.255.255
#exit
interface sgw-gtpu-egress loopback
  ip address 50.50.17.6 255.255.255.255
#exit
interface sgw-gtpu-ingress-phazr loopback
```

```
    ip address 60.60.17.6 255.255.255.255
  #exit
  subscriber default
  exit
  aaa group default
  #exit
  gtpp group default
    gtpp limit-secondary-rat-usage 32
  #exit
  gtpu-service SxC
    echo-interval 60
    echo-retransmission-timeout 1
    bind ipv4-address 50.50.17.4 ipv6-address 2001:4888:50:50::17 bearer-type all
  exit
  gtpu-service n9-ingress
    bind ipv4-address 60.60.17.5 bearer-type all
  exit
  gtpu-service pgw-gtpu
    bind ipv4-address 50.50.17.5 bearer-type all
  exit
  gtpu-service sgw-gtpu-egress
    bind ipv4-address 50.50.17.6 bearer-type all
  exit
  gtpu-service sgw-gtpu-ingress-phazr
    echo-interval 60
    echo-retransmission-timeout 1
    bind ipv4-address 60.60.17.6 bearer-type all
  exit
  sx-service sx-svc
    instance-type userplane
    bind ipv4-address 50.50.17.4 ipv6-address 2001:4888:50:50::17
    sxa max-retransmissions 5
    sxa retransmission-timeout-ms 1000
    sxb max-retransmissions 5
    sxb retransmission-timeout-ms 1000
    sxab max-retransmissions 5
    sxab retransmission-timeout-ms 1000
    sx-protocol heartbeat retransmission-timeout 15
    sx-protocol heartbeat max-retransmissions 3
    sx-protocol association debug-mode debug-reattempt-timeout 1
    no sx-protocol compression
  exit
  user-plane-service user_plane_svc
    associate gtpu-service pgw-gtpu upf-ingress
    associate gtpu-service n9-ingress upf-ingress interface-type n9
    associate gtpu-service sgw-gtpu-ingress-phazr sgw-ingress
    associate gtpu-service sgw-gtpu-egress sgw-egress
    associate gtpu-service SxC cp-tunnel
    associate sx-service sx-svc
    associate fast-path service
    associate control-plane-group SAEGW
    associate userplane-load-control-profile LCP
    associate userplane-overload-control-profile OLCP
  exit
```

The following is a sample configuration to separate the N3 and N9 BGP routes for vUPF.

```
context SAEGW
  router bgp 61441
    neighbor 20.14.41.100 remote-as 65000
    neighbor 20.14.41.100 ebgp-multihop
    neighbor 20.14.41.100 update-source 20.14.41.98
    neighbor 2001:4888:121:1441::1 remote-as 65000
    neighbor 2001:4888:121:1441::1 ebgp-multihop
    neighbor 2001:4888:121:1441::1 update-source 2001:4888:121:1441::2
```

```
  address-family ipv4
    redistribute connected
    redistribute static
  #exit
  address-family ipv6
    neighbor 2001:4888:121:1441::1 activate
    redistribute connected
    redistribute static
  #exit
#exit
interface SxC loopback
  ip address 50.50.18.4 255.255.255.255
#exit
interface SxC-ipv6 loopback
  ipv6 address 2001:4888:50:50::18/128
#exit
interface gn-1
  ip address 20.14.41.98 255.255.255.0
  ipv6 address 2001:4888:121:1441::2/64 secondary
#exit
interface gn-2
  ip address 20.14.42.98 255.255.255.0
  ipv6 address 2001:4888:121:1442::2/64 secondary
#exit
interface n9-egress loopback
  ip address 50.50.18.8 255.255.255.255
#exit
interface pgw-gtpu loopback
  ip address 50.50.18.5 255.255.255.255
#exit
interface pgw-gtpu-ipv6 loopback
  ipv6 address 2001:4888:50:50::18:5/128
#exit
interface sgw-gtpu-egress loopback
  ip address 50.50.18.6 255.255.255.255
#exit
interface sgw-gtpu-egress-ipv6 loopback
  ipv6 address 2001:4888:50:50::18:6/128
#exit
interface sgw-gtpu-ingress-phazr loopback
  ip address 50.50.18.7 255.255.255.255
#exit
interface sgw-gtpu-ingress-phazr-ipv6 loopback
  ipv6 address 2001:4888:50:50::18:7/128
#exit
subscriber default
exit
aaa group default
#exit
gtpp group default
  gtpp limit-secondary-rat-usage 32
#exit
gtpu-service SxC
  echo-interval 60
  echo-retransmission-timeout 1
  bind ipv4-address 50.50.18.4 ipv6-address 2001:4888:50:50::18 bearer-type all
exit
gtpu-service n9-egress
  bind ipv4-address 50.50.18.8 bearer-type all
exit
gtpu-service pgw-gtpu
  bind ipv4-address 50.50.18.5 ipv6-address 2001:4888:50:50::18:5 bearer-type all
exit
gtpu-service sgw-gtpu-egress
```

```
    bind ipv4-address 50.50.18.6 ipv6-address 2001:4888:50:50::18:6 bearer-type all
  exit
  gtpu-service sgw-gtpu-ingress-phazr
    echo-interval 60
    echo-retransmission-timeout 1
    bind ipv4-address 50.50.18.7 ipv6-address 2001:4888:50:50::18:7 bearer-type all
  exit
  sx-service sx-svc
    instance-type userplane
    bind ipv4-address 50.50.18.4 ipv6-address 2001:4888:50:50::18
    sxa max-retransmissions 5
    sxa retransmission-timeout-ms 1000
    sxb max-retransmissions 5
    sxb retransmission-timeout-ms 1000
    sxab max-retransmissions 5
    sxab retransmission-timeout-ms 1000
    sx-protocol heartbeat retransmission-timeout 15
    sx-protocol heartbeat max-retransmissions 3
    sx-protocol association debug-mode debug-reattempt-timeout 1
    no sx-protocol compression
  exit
  user-plane-service user_plane_svc
    associate gtpu-service pgw-gtpu upf-ingress
    associate gtpu-service n9-egress upf-egress
    associate gtpu-service sgw-gtpu-ingress-phazr sgw-ingress
    associate gtpu-service sgw-gtpu-egress sgw-egress
    associate gtpu-service SxC cp-tunnel
    associate sx-service sx-svc
    associate fast-path service
    associate control-plane-group SAEGW
    associate userplane-load-control-profile LCP
    associate userplane-overload-control-profile OLCP
  exit
  ip route 0.0.0.0 0.0.0.0 20.14.41.100 gn-1
  ip igmp profile default
  #exit
#exit
```

Note For RCM UPF, add all the IP prefix-lists in day 0.5 configuration.

Verifying the UPF Ingress Interface Type Configuration

To verify the configuration, use the following command:

- show user-plane-service all, on page 7

show user-plane-service all

The **show user-plane-service all** command displays user-plane service information.

The following is a sample output of this command:

```
show user-plane-service all
Service name                        : user-plane-service
  Service-Id                        : 7
  Context                           : EPC2-UP
  Status                            : NOT STARTED
  UPF Ingress GTPU Service          : sx-gtpu-service
```

```
UPF Ingress N3 Interface Type GTPU Service: n3-ingress
UPF Ingress N9 Interface Type GTPU Service: n9-ingress
UPF Ingress S5U Interface Type GTPU Service: s5u-ingress
UPF Ingress S8U Interface Type GTPU Service: s8u-ingress
UPF Egress GTPU Service           : Not defined
SGW Ingress GTPU Service          : SGW-Ingress
SGW Egress GTPU Service           : SGW-Egress
Control Plane Tunnel GTPU Service : control_gtpu
Sx Service                        : sxu
Control Plane Group               : g1
Fast-Path service                 : Disabled
```

NOTES:

- The UPF configures only one of the interface types, **pgw-ingress** or **upf-ingress** in a single user plane service.

Multiple GTPU IP Addresses Support

Table 3: Feature History

Feature Name	Release Information	Feature Description
Binding Multiple GTPU IP Addresses	2023.04	UPF allows binding of multiple GTPU IP addresses to provide high uplink throughput in Private 5G deployments. UPF creates unique GTPU 3-tuple hash entries to ensure uniform distribution of ingress traffic on all VPP worker threads.

Feature Description

In the Private 5G networks, where the uplink traffic is more than the downlink traffic, the number of gNodeBs are limited. UPF achieves the optimal distribution of subscriber traffic using the Receive Side Scaling (RSS) concept. However, due to limited number of VPP worker threads on UPF, the ingress traffic does not get evenly distributed across them. It limits the UPF from optimally utilizing all the worker threads, and therefore impacts the subscriber's uplink throughput.

UPF allows utilizing the VPP worker threads in an optimal manner by creating the unique GTPU 3-tuple hash entries on the UPF's forwarding plane. It ensures an optimum distribution of ingress traffic on the N3 interface. It enables UPF to perform traffic processing to its maximum capacity ensuring high uplink throughput for Private 5G networks.

How it Works

The even distribution of ingress traffic across all the available worker threads on the UPF is ensured by creating the unique 3-tuple hash entries.

The GTPU header source IP is the gNodeB's N3 service egress IP, while the source port is the standard port in the absence of port randomization support on the gNodeB. With these two values being constants per gNodeB, the only variable that can be achieved in the GTPU header is the destination IP (N3 service ingress IP address).

The following points describe how the unique 3-tuple hash entries are created:

- GTPU addresses are assigned in FTEID in Sx Session Establish Response for a subscriber session in a round-robin manner. However, this round-robin mechanism is ensured only within the session manager.
- When a session manager is selected, the GTPU address in FTEID is allocated in a round-robin manner to all the sessions landing on the selected session manager.
- Deletion of IP address from the GTPU service or the interface does not cause the GTPU service to stop. The existing calls will be deleted. However, the uplink traffic of the subscriber to which the deleted IP address was assigned will not be received at UPF.
- If every IP address is configured under a separate interface, the interface must be configured as a loopback (multiple non-loopback interfaces cannot share the same subnet). If any interface goes down and comes alive again, the GTPU service does not stop as the interfaces are configured as loopback.
- Every address pair is one GTPU endpoint for a GTPU service. Every endpoint should belong to the same subnet.

Note

- For optimum performance, the number of GTPU bind addresses must be equal to the number of VPP worker threads.
- A maximum of 12 IP address pairs (either IPv4 or IPv6 or IPv4+IPv6) can be configured per GTPU service.

Limitations

This feature has the following known limitations::

- If the GTPU Echo is enabled and the first bind IP address is deleted, the Echo request from the remote peer will not reach the UPF causing the remote peer to detect path failure. The subsequent downlink traffic from the UPF for that subscriber results in an error indication received at UPF.

  The GTPU services stop only if all the bind IP addresses are deleted. These IP addresses can be under single interface or separate interfaces.
- UPF does not allow to configure different IP address types to be configured per GTPU service i.e., IPv4 + IPv6 or IPv4 + IPv4/IPv6 combinations are not allowed. Only the combinations of IPv4 or IPv6 or IPv4+IPv6, are allowed to be configured.

Configuring Multiple Bind Addresses on N3 Interface

To configure multiple bind addresses on the N3 interface, use the following sample configuration:

```
config
  context context_name
    gtpu-service service_name
      bind { ipv4-address ipv4_address | ipv6-address ipv6_address [
bearer-type { non-ims-media | ims-media | all } ] }
      exit
```

NOTES:

- **bind { ipv4-address** *ipv4_address* **| ipv6-address** *ipv6_address* **}**—Bind the GTPU service to the IPv4 address or IPv6 address on the N3 interface.
- **[bearer-type { non-ims-media | ims-media | all }]**—Specify the type of bearer to be associated with the bind address. **bearer-type** allows three types of media traffic:
  - **non-ims-media**—Configure bind address for non-ims media only.
  - **ims-media**—Configure bind address for ims-media traffic only.
  - **all**—Configure bind address to handle all types of bearer traffic. This is the default behavior.

Note If GTPU echo is enabled in GTPU service, the Echo request will be started with the first bind address configured in GTPU service. The bind address configuration is stored in the sorted list. As a result, the GTPU echo starts with the numerically lowest bind IP address.

Configuration Example

The following is an example configuration for binding multiple GTPU IP addresses on the N3 interface:

```
config
context EPC2-UP
cups enable
interface uplane_interface
    ip address <ip_address 1> <subnet mask>
    ip address <ip_address 2> <subnet mask> secondary
    ip address <ip_address 3> <subnet mask> secondary
    ip address <ip_address 4> <subnet mask> secondary
    ip address <ip_address 5> <subnet mask> secondary
    ip address <ip_address 6> <subnet mask> secondary
    ip address <ip_address 7> <subnet mask> secondary
#exit
gtpu-service N3U-service
    bind ipv4-address <ip_address 1> bearer-type all
    bind ipv4-address <ip_address 2> bearer-type all
    bind ipv4-address <ip_address 3> bearer-type all
    bind ipv4-address <ip_address 4> bearer-type all
    bind ipv4-address <ip_address 5> bearer-type all
    bind ipv4-address <ip_address 6> bearer-type all
exit
```

Configuration Verification

To verify the configuration, use the following command:

show gtpu-service name N3U-service

The following is a sample output of this command:

```
show gtpu-service name N3U-service

Service name:             N3U-service
  Context:                  EPC2-UP
  State:                    Started
  Echo Interval:            Disabled
  Sequence number:          Disabled
  Include UDP Port Ext Hdr: FALSE
```

```
Max-retransmissions:         4
Retransmission Timeout:   5 (secs)
IPSEC Tunnel Idle Timeout:60 (secs)
GTPU IP QOS DSCP value:10
Allow Error-Indication:   Disabled
Address List:
               <ip_address 1>      all
               <ip_address 2>      all
               <ip_address 3>      all
               <ip_address 4>      all
               <ip_address 5>      all
               <ip_address 6>      all
               <ip_address 7>      all
GTPU UDP Checksum:  Enabled - Attempt Optimize Default Mode
Path Failure Detection
on gtp echo msgs:         Set
Path Failure Clear Trap : non-echo
GTPU Source-port Configuration : Use Standard Port
  GTPU Source-port Offset      : N/A
GTPU Peer Statistics Threshold : 16000
    1
```

OAM Support

This section describes operations, administration, and maintenance support for this feature.

Monitoring Support

This section describes feature-level monitoring support information.

show user-plane-service gtpu local-addresses

This command displays the number of active bearers/sessions using specific bind address.

For Single Converged 4G Call:

```
show user-plane-service gtpu local-addresses
  Service name:         N3U-service
  Address List:                                                    GTPv1 Sessions:
     <ip_address 1>                                                    1
  Service name:         SGW-Egress
  Address List:                                                    GTPv1 Sessions:
     <ip_address 2>                                                    1
  Service name:         SGW-Ingress
Address List:                                                   GTPv1 Sessions:
     <ip_address 3>                                                    1
```

Note

- In 4G calls, there will be one GTPU session per bearer.
- This CLI output will not display the total number of unique subscribers per GTPU address.

For 5G Call:

```
show user-plane-service gtpu local-addresses
  Service name:         N3U-service
  Address List:                                                     GTPv1 Sessions:
     <ip_address 1>                                                    1
  Service name:         SGW-Egress
```

```
  Address List:                                                    GTPv1 Sessions:
No active sessions exists
  Service name:         SGW-Ingress
  Address List:                                                    GTPv1 Sessions:
No active sessions exists
```

show user-plane-service gtpu statistics local-address

This command displays the statistics per GTPU bind address.

```
show user-plane-service gtpu statistics local-address < Local IPv4/IPv6 Address>
Session Stats:
    Current:                                    1
    Current (IMS-media):                        0
    Total Setup:                                2
    Total Setup (IMS-media):                    0
    Current gtpu v0 sessions:                   0
    Current gtpu v1 sessions:                   2

  Total Data Stats:
    Uplink Packets:                   0  Uplink Bytes:                0
    Downlink Packets:                 0  Downlink Bytes:              0
    Packets Discarded:                0  Bytes Discarded:             0
    Uplink Packets (IMS-media):                 0  Uplink Bytes (IMS-media):
    0
    Downlink Packets (IMS-media):               0  Downlink Bytes (IMS-media):
    0
    Packets Discarded (IMS-media):              0  Bytes Discarded (IMS-media):
    0

 QoS Stats:

    QCI 1:
         Uplink Packets:              0  Uplink Bytes:                0
         Downlink Packets:            0  Downlink Bytes:              0
         Packets Discarded:           0  Bytes Discarded:             0

    QCI 2:
         Uplink Packets:              0  Uplink Bytes:                0
         Downlink Packets:            0  Downlink Bytes:              0
         Packets Discarded:           0  Bytes Discarded:             0

    QCI 3:
         Uplink Packets:              0  Uplink Bytes:                0
         Downlink Packets:            0  Downlink Bytes:              0
         Packets Discarded:           0  Bytes Discarded:             0

    QCI 4:
         Uplink Packets:              0  Uplink Bytes:                0
         Downlink Packets:            0  Downlink Bytes:              0
         Packets Discarded:           0  Bytes Discarded:             0

    QCI 5:
         Uplink Packets:              0  Uplink Bytes:                0
         Downlink Packets:            0  Downlink Bytes:              0
         Packets Discarded:           0  Bytes Discarded:             0

    QCI 6:
         Uplink Packets:              0  Uplink Bytes:                0
         Downlink Packets:            0  Downlink Bytes:              0
         Packets Discarded:           0  Bytes Discarded:             0

   QCI 7:
         Uplink Packets:              0  Uplink Bytes:                0
```

```
     Downlink Packets:            0  Downlink Bytes:              0
     Packets Discarded:           0  Bytes Discarded:             0

QCI 8:
     Uplink Packets:              0  Uplink Bytes:                0
     Downlink Packets:            0  Downlink Bytes:              0
     Packets Discarded:           0  Bytes Discarded:             0

QCI 9:
     Uplink Packets:              0  Uplink Bytes:                0
     Downlink Packets:            0  Downlink Bytes:              0
     Packets Discarded:           0  Bytes Discarded:             0

QCI 65:
     Uplink Packets:              0  Uplink Bytes:                0
     Downlink Packets:            0  Downlink Bytes:              0
     Packets Discarded:           0  Bytes Discarded:             0

QCI 66:
     Uplink Packets:              0  Uplink Bytes:                0
     Downlink Packets:            0  Downlink Bytes:              0
     Packets Discarded:           0  Bytes Discarded:             0

QCI 69:
     Uplink Packets:              0  Uplink Bytes:                0
     Downlink Packets:            0  Downlink Bytes:              0
     Packets Discarded:           0  Bytes Discarded:             0

 QCI 70:
     Uplink Packets:              0  Uplink Bytes:                0
     Downlink Packets:            0  Downlink Bytes:              0
     Packets Discarded:           0  Bytes Discarded:             0

QCI 80:
     Uplink Packets:              0  Uplink Bytes:                0
     Downlink Packets:            0  Downlink Bytes:              0
     Packets Discarded:           0  Bytes Discarded:             0

QCI 82:
     Uplink Packets:              0  Uplink Bytes:                0
     Downlink Packets:            0  Downlink Bytes:              0
     Packets Discarded:           0  Bytes Discarded:             0

QCI 83:
     Uplink Packets:              0  Uplink Bytes:                0
     Downlink Packets:            0  Downlink Bytes:              0
     Packets Discarded:           0  Bytes Discarded:             0

  Non-Std QCI(Non-GBR):
      Uplink Packets:             0  Uplink Bytes:                  0
      Downlink Packets:           0  Downlink Bytes:          0
      Packets Discarded:          0  Bytes Discarded:           0
      Uplink Packets:             0  Uplink Bytes:                  0
      Downlink Packets:           0  Downlink Bytes:          0
      Packets Discarded:          0  Bytes Discarded:           0

  Total uplink packets GBR QCI's:               0
  Total uplink Bytes GBR QCI's:                 0
  Total Downlink packets GBR QCI's:             0
  Total Downlink Bytes GBR QCI's:               0
  Total uplink packets Non-GBR QCI's:           0
  Total uplink Bytes Non-GBR QCI's:             0
  Total Downlink packets Non-GBR QCI's:         0
  Total Downlink Bytes Non-GBR QCI's:           0
```

```
 Path Management Messages:
   Echo Request Rx:                  0  Echo Response Rx:            0
   Echo Request Tx:                  0  Echo Response Tx:            0
   SuppExtnHdr  Tx:                  0  SuppExtnHdr   Rx:            0

 Peer Stats:
   Total GTPU Peers:                              1
   Total GTPU Peers with Stats:                   1

Tunnel Management Messages:
    Error Indication Tx:                           0
    Error Indication Rx:                           0
    Error Indication Rx Discarded:                 0
```

show npu utilization table

This command displays the VPP worker thread CPU utilization:

```
[local]UPF1# show npu utilization table
          ---------vpp--------
   lcore      now   5min  15min
--------  ------ ------ ------
 01/0/01     15%    11%    12%
 01/0/02     15%    13%    14%
 01/0/03     23%    23%    24%
 01/0/04     19%    21%    20%
 01/0/05     22%    20%    20%
 01/0/06     12%    11%    11%
 01/0/07     16%    16%    16%
 01/0/08     15%    17%    17%
 01/0/09     11%    12%    12%
```