



Configuring GRE Tunnels

Generic Routing Encapsulation (GRE) is a tunneling protocol that provides a simple generic approach to transport packets of one protocol over another protocol by means of encapsulation. This module provides information about how to configure a GRE tunnel.

- [Configuring GRE Tunnels, on page 1](#)
- [Single Pass GRE Encapsulation Allowing Line Rate Encapsulation, on page 4](#)

Configuring GRE Tunnels

Tunneling provides a mechanism to transport packets of one protocol within another protocol. Generic Routing Encapsulation (GRE) is a tunneling protocol that provides a simple generic approach to transport packets of one protocol over another protocol with encapsulation. GRE encapsulates a payload, that is, an inner packet that needs to be delivered to a destination network inside an outer IP packet. The GRE tunnel behaves as a virtual point-to-point link that has two endpoints identified by the tunnel source and tunnel destination address. The tunnel endpoints send payloads through GRE tunnels by routing encapsulated packets through intervening IP networks. Other IP routers along the way do not parse the payload (the inner packet); they only parse the outer IP packet as they forward it towards the GRE tunnel endpoint. Upon reaching the tunnel endpoint, GRE encapsulation is removed and the payload is forwarded to the packet's ultimate destination.

Encapsulation by the outer packet takes place at the tunnel source whereas decapsulation of the outer packet takes place at the tunnel destination. Encapsulation and decapsulation data is collected periodically or on demand. Encapsulation statistics provide us the number of packets encapsulated at the tunnel source. Decapsulation statistics provide us the number of packets that are decapsulated at the tunnel destination. This data is stored as statistics in logical tables that are based on statistics type in the route processor. The different statistics types include L2 Interface TX Stats, L3 Interface TX Stats, TRAP stats, and so on. Encapsulation statistics can help you to infer the source of the traffic, and decapsulation statistics provide you the destination of the traffic. Decapsulation statistics also help you to detect the type of traffic as well.

Guidelines and Restrictions for Configuring GRE Tunnels

The following restrictions apply while configuring GRE tunnels:

- The router supports up to 500 GRE tunnels.
- Only up to 16 unique source IP addresses are supported for the tunnel source.
- 2-pass to Single-pass migration, which means converting the same GRE tunnel, is not possible in a single configuration step. You must first delete the 2-pass tunnel and then add the Single-pass tunnel.

- Configurable MTU is not supported on Single-pass GRE interface, but supported on 2-pass GRE interface.
- L3VPN over GRE is not supported.
- The Cisco NCS 5500 series router support only IPv4 GRE tunnels. IPv6 GRE tunnels are not supported.
- The IPv4 GRE tunnels supports IPv4 and IPv6 payloads.
- To use the outer IPv4 GRE header for IP tunnel decapsulation in the hashing algorithm for ECMP and bundle member selection, use the **hw-module profile load-balance algorithm** command.

Table 1: GRE Tunnels with Supported MTU and TOS Hardware Profiles

Supported Hardware	Profile Type	Maximum Supported Profile
NC55-36x100G NC55-18H18F NC55-24x100G-SE NC55-24H12F-SE NC55-36x100G-S NC55-6x200-DWDM-S	MTU	3
NC55-36x100G-A-SE NC55-MOD-A-S NC55-MOD-A-SE-S NC55-32T16Q4H	MTU	3
NC57-24DD NC57-18DD-SE NC57-36H-SE NC57-36H6D NC57-MOD-S	MTU	7
NC55-36x100G NC55-18H18F NC55-24x100G-SE NC55-24H12F-SE NC55-36x100G-S NC55-6x200-DWDM-S	TOS	8

Supported Hardware	Profile Type	Maximum Supported Profile
NC55-36x100G-A-SE NC55-MOD-A-S NC55-MOD-A-SE-S NC55-32T16Q4H	TOS	8



Note If the configured MTU and Tunnel TOS profile exceeds the supported hardware limit, the system displays SDK-Out of Memory error.

Configuration Example

Configuring a GRE tunnel involves creating a tunnel interface and defining the tunnel source and destination. This example shows how to configure a GRE tunnel between Router1 and Router2. You need to configure tunnel interfaces on both the routers. Tunnel source IP address on Router1 will be configured as the tunnel destination IP address on Router2. Tunnel destination IP address on Router1 will be configured as the tunnel source IP address on Router2. In this example, OSPF is used as the routing protocol between the two routers. You can also configure BGP or IS-IS as the routing protocol.

```
RP/0/RP0/CPU0:Router1# configure
RP/0/RP0/CPU0:Router1(config)# interface tunnel-ip 30
RP/0/RP0/CPU0:Router1(config-if)# tunnel mode gre ipv4
RP/0/RP0/CPU0:Router1(config-if)# ipv4 address 10.1.1.1 255.255.255.0
RP/0/RP0/CPU0:Router1(config-if)# tunnel source 192.168.1.1
RP/0/RP0/CPU0:Router1(config-if)# tunnel destination 192.168.2.1
RP/0/RP0/CPU0:Router1(config-if)# exit
RP/0/RP0/CPU0:Router1(config)# interface Loopback 0
RP/0/RP0/CPU0:Router1(config-if)# ipv4 address 10.10.10.1
RP/0/RP0/CPU0:Router1(config-if)# exit
RP/0/RP0/CPU0:Router1(config)# router ospf 1
RP/0/RP0/CPU0:Router1(config-ospf)# router-id 192.168.4.1
RP/0/RP0/CPU0:Router1(config-ospf)# area 0
RP/0/RP0/CPU0:Router1(config-ospf-ar)# interface tunnel-ip 30
RP/0/RP0/CPU0:Router1(config-ospf-ar)# interface Loopback 0
RP/0/RP0/CPU0:Router1(config-ospf-ar)# commit

RP/0/RP0/CPU0:Router2# configure
RP/0/RP0/CPU0:Router2(config)# interface tunnel-ip 30
RP/0/RP0/CPU0:Router2(config-if)# tunnel mode gre ipv4
RP/0/RP0/CPU0:Router2(config-if)# ipv4 address 10.1.1.2 255.255.255.0
RP/0/RP0/CPU0:Router2(config-if)# tunnel source 192.168.2.1
RP/0/RP0/CPU0:Router2(config-if)# tunnel destination 192.168.1.1
RP/0/RP0/CPU0:Router2(config-if)# exit
RP/0/RP0/CPU0:Router2(config)# interface Loopback 0
RP/0/RP0/CPU0:Router2(config-if)# ipv4 address 2.2.2.2
RP/0/RP0/CPU0:Router2(config)# router ospf 1
RP/0/RP0/CPU0:Router2(config-ospf)# router-id 192.168.3.1
RP/0/RP0/CPU0:Router2(config-ospf)# area 0
RP/0/RP0/CPU0:Router2(config-ospf-ar)# interface tunnel-ip 30
RP/0/RP0/CPU0:Router2(config-ospf-ar)# interface Loopback 0
RP/0/RP0/CPU0:Router2(config-ospf-ar)# commit
```

Single Pass GRE Encapsulation Allowing Line Rate Encapsulation

Single Pass GRE Encapsulation Allowing Line Rate Encapsulation feature, also known as Prefix-based GRE Tunnel Destination for Load Balancing feature, enables line rate GRE encapsulation traffic and enables flow entropy. Data-plane forwarding performance supports full line rate, which is adjusted to consider added encapsulation. GRE tunnel goes down if the destination is not available in RIB. Routing over GRE Single-pass tunnel is not supported in Release 6.3.2, so the traffic that is eligible for GRE encapsulation is identified using an ACL filter that is based on GRE encapsulation. GRE tunnel destination address is an anycast address. All of the GRE encapsulation must be assigned based upon either an ACL or a policy-map, or both. Destinations may be individual addresses or /28 prefixes.

Configure GRE Single-Pass Entropy

Perform the following tasks to configure the GRE Single-Pass Entropy feature:

- GRE Single-pass
- GRE Entropy(ECMP/UCMP)

```

/* GRE Single-Pass */

Router# configure
Router(config)# interface tunnel-ip30016
Router(config-if)# ipv4 address 216.1.1.1 255.255.255.0
Router(config-if)# ipv6 address 216:1:1::1/64
Router(config-if)# ipv6 enable
Router(config-if)# tunnel mode gre ipv4 encap
Router(config-if)# tunnel source Loopback22
Router(config-if)# tunnel destination 170.170.170.22
Router(config-if)# commit
Router(config-if)# exit

/* GRE Entropy(ECMP/UCMP) */

ECMP (ISIS)

Router# configure
Router(config)# router isis core
Router(config)# apply-group ISIS-INTERFACE
Router(config-isis)# is-type level-2-only
Router(config-isis)# net 49.1111.0000.0000.002.00
Router(config-isis)# nsr
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# metric 2
Router(config-isis-af)# mpls traffic-eng level-2-only
Router(config-isis-af)# mpls traffic-eng router-id Loopback0
Router(config-isis-af)# maximum-paths 5
Router(config-isis-af)# commit
!

/* UCMP (ISIS) */

```

```
Router# configure
Router(config)# router isis core
Router(config)# apply-group ISIS-INTERFACE
Router(config-isis)# is-type level-2-only
Router(config-isis)# net 49.1111.0000.0000.002.00
Router(config-isis)# nsr
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide
Router(config-isis-af)# ucmp
Router(config-isis-af)# metric 2
Router(config-isis-af)# mpls traffic-eng level-2-only
Router(config-isis-af)# mpls traffic-eng router-id Loopback0
Router(config-isis-af)# maximum-paths 5
Router(config-isis-af)# redistribute connected
Router(config-isis-af)# commit
Router(config-isis-af)# exit
!

Router# configure
Router(config)# interface Bundle-Ether3
Router(config-if)# apply-group ISIS-INTERFACE
Router(config-if)# address-family ipv4 unicast
Router(config-af)# metric 20
Router(config-af)# commit
Router(config-af)# exit
!

Router# configure
Router(config)# interface Bundle-Ether111
Router(config-if)# apply-group ISIS-INTERFACE
Router(config-if)# address-family ipv4 unicast
Router(config-af)# metric 15
Router(config-af)# commit
Router(config-af)# exit
!

/* ECMP (OSPF) */

Router# configure
Router(config)# router ospf 3
Router(config-ospf)# nsr
Router(config-ospf)# maximum paths 5
Router(config-ospf)# address-family ipv4 unicast
Router(config-ospf-af)# area 0
Router(config-ospf-af-ar)# interface Bundle-Ether3
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether4
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether111
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether112
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Loopback23
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface HundredGigE0/7/0/23
Router(config-ospf-af-ar-if)# commit
```

```

Router(config-ospf-af-ar-if) # exit

/* UCMP (OSPF) */

Router# configure
Router(config) # router ospf 3
Router(config-ospf) # nsr
Router(config-ospf) # maximum paths 5
Router(config-ospf) # ucmp
Router(config-ospf) # address-family ipv4 unicast
Router(config-ospf-af) # area 0
Router(config-ospf-af-ar) # interface Bundle-Ether3 cost 2
Router(config-ospf-af-ar-if) # exit
!
Router(config-ospf-af-ar) # interface Bundle-Ether4
Router(config-ospf-af-ar-if) # exit
!
Router(config-ospf-af-ar) # interface Bundle-Ether11
Router(config-ospf-af-ar-if) # exit
!
Router(config-ospf-af-ar) # interface Bundle-Ether112 cost 2
Router(config-ospf-af-ar-if) # exit
!
Router(config-ospf-af-ar) # interface Loopback23
Router(config-ospf-af-ar-if) # exit
!
Router(config-ospf-af-ar) # interface HundredGigE0/7/0/23
Router(config-ospf-af-ar-if) # commit
Router(config-ospf-af-ar-if) # exit

/* ECMP (BGP) */
Router# configure
Router(config) # router bgp 800
Router(config-bgp) # bgp bestpath as-path multipath-relax
Router(config-bgp) # address-family ipv4 unicast
Router(config-bgp-af) # network 170.170.170.3/32
Router(config-bgp-af) # network 170.170.170.10/32
Router(config-bgp-af) # network 170.170.170.11/32
Router(config-bgp-af) # network 170.170.172.3/32
Router(config-bgp-af) # network 180.180.180.9/32
Router(config-bgp-af) # network 180.180.180.20/32
Router(config-bgp-af) # network 180.180.180.21/32
Router(config-bgp-af) # network 180.180.180.24/32
Router(config-bgp-af) # network 180.180.180.25/32
Router(config-bgp-af) # commit
!
Router# configure
Router(config) # router bgp 800
Router(config-bgp) # neighbor 4.1.1.2
Router(config-bgp-nbr) # remote-as 300
Router(config-bgp-nbr) # address-family ipv4 unicast
Router(config-bgp-nbr-af) # address-family ipv4 unicast
Router(config-bgp-nbr-af) # route-policy pass-all in
Router(config-bgp-nbr-af) # route-policy pass-all out
Router(config-bgp-nbr-af) # commit
!

/* UCMP (BGP) */

Router# configure
Router(config) # router bgp 800

```

```

Router(config-bgp)# bgp bestpath as-path multipath-relax
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# maximum-paths ebgp 5
Router(config-bgp-af)# network 180.180.180.9/32
Router(config-bgp-af)# network 180.180.180.20/32
Router(config-bgp-af)# network 180.180.180.21/32
Router(config-bgp-af)# network 180.180.180.24/32
Router(config-bgp-af)# network 180.180.180.25/32
Router(config-bgp-af)# commit
!
Router# configure
Router(config)# router bgp 800
Router(config-bgp)# neighbor 7.1.5.2
Router(config-bgp-nbr)# remote-as 4000
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy TRANSITO_IN in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# next-hop-self
Router(config-bgp-nbr-af)# commit
!
Router# configure
Router(config)# router bgp 800
Router(config-bgp)# 4.1.111.2
Router(config-bgp-nbr)# remote-as 4000
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy TRANSITO_IN in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# next-hop-self
Router(config-bgp-nbr-af)# commit
!

/* Configure roupte policy */

Router# configure
Router(config)# route-policy TRANSITO_IN
Router(config-rpl)# if destination in (170.170.170.24/32) then
Router(config-rpl-if)# set extcommunity bandwidth (2906:1250000)
Router(config-rpl-if)# else
Router(config-rpl-else)# pass
Router(config-rpl-else)# endif
Router(config-rpl)# end-policy
!

Router# configure
Router(config)# route-policy TRANSIT1_IN
Router(config-rpl)# if destination in (170.170.170.24/32) then
Router(config-rpl-if)# set extcommunity bandwidth (2906:37500000)
Router(config-rpl-if)# else
Router(config-rpl-else)# pass
Router(config-rpl-else)# endif
Router(config-rpl)# end-policy

```

Running Configuration

```

/* GRE Single-Pass configuration */

interface tunnel-ip30016

```

```

ipv4 address 216.1.1.1 255.255.255.0
ipv6 address 216:1:1::1/64
ipv6 enable
tunnel mode gre ipv4 encaps
tunnel source Loopback22
tunnel destination 170.170.170.22
!

/* GRE Entropy (ECMP/UCMP) */

ECMP (ISIS)

router isis core
apply-group ISIS-INTERFACE
is-type level-2-only
net 49.1111.0000.0000.002.00
nsr
log adjacency changes
address-family ipv4 unicast
metric-style wide
metric 2
mpls traffic-eng level-2-only
mpls traffic-eng router-id Loopback0
maximum-paths 5
!

/* UCMP (ISIS) */

router isis core
apply-group ISIS-INTERFACE
is-type level-2-only
net 49.1111.0000.0000.002.00
nsr
log adjacency changes
address-family ipv4 unicast
metric-style wide
ucmp
metric 2
mpls traffic-eng level-2-only
mpls traffic-eng router-id Loopback0
maximum-paths 5
redistribute connected
!
interface Bundle-Ether3
apply-group ISIS-INTERFACE
address-family ipv4 unicast
metric 20
!

interface Bundle-Ether111
apply-group ISIS-INTERFACE
address-family ipv4 unicast
metric 15
!

!

/* ECMP (OSPF) */

router ospf 3
nsr
maximum-paths 5
address-family ipv4 unicast
area 0

```



```
interface Bundle-Ether3
!
interface Bundle-Ether4
!
interface Bundle-Ether111
!
interface Bundle-Ether112
!
interface Loopback23
!
interface HundredGigE0/7/0/23
!
!
!
/* UCMP (OSPF) */

router ospf 3
nsr
maximum paths 5
ucmp
address-family ipv4 unicast
area 0
interface Bundle-Ether3
cost 2
!
interface Bundle-Ether4
!
interface Bundle-Ether111
!
interface Bundle-Ether112
cost 2
!
interface Loopback23
!
interface HundredGigE0/7/0/23
!
!
!

/* ECMP (BGP) */

router bgp 800
bgp bestpath as-path multipath-relax
address-family ipv4 unicast
maximum-paths ebgp 5
network 170.170.170.3/32
network 170.170.170.10/32
network 170.170.170.11/32
network 170.170.172.3/32
network 180.180.180.9/32
network 180.180.180.20/32
network 180.180.180.21/32
network 180.180.180.24/32
network 180.180.180.25/32
!
neighbor 4.1.1.2
remote-as 300
address-family ipv4 unicast
route-policy PASS-ALL in
route-policy PASS-ALL out
next-hop-self
!
!
```

```

/* UCMF(BGP) */

router bgp 800
  bgp bestpath as-path multipath-relax
  address-family ipv4 unicast
  maximum-paths ebgp 5
  network 180.180.180.9/32
  network 180.180.180.20/32
  network 180.180.180.21/32
  network 180.180.180.24/32
  network 180.180.180.25/32
  !

  neighbor 7.1.5.2
  remote-as 4000
  address-family ipv4 unicast
  route-policy TRANSITO_IN in
  route-policy PASS-ALL out
  next-hop-self
  !
  !
  neighbor 4.1.111.2
  remote-as 4000
  address-family ipv4 unicast
  route-policy TRANSIT1_IN in
  route-policy PASS-ALL out
  next-hop-self
  !
  !

/* Configure rouverte policy */

route-policy TRANSITO_IN
  if destination in (170.170.170.24/32) then
  set extcommunity bandwidth (2906:1250000)
  else
  pass
  endif
end-policy
!
route-policy TRANSIT1_IN
  if destination in (170.170.170.24/32) then
  set extcommunity bandwidth (2906:37500000)
  else
  pass
  endif
end-policy
!

```

Verification

Verify if the tunnel mode GRE encapsulation is enabled.

```

Router# show int tunnel-ip2

interface tunnel-ip2
  ipv4 address 80.80.82.1 255.255.255.0
  ipv6 address 2000:80:80:82::1/64
  load-interval 30
  tunnel mode gre ipv4 encap
  tunnel source Loopback4
  tunnel destination 11.4.2.2
  !

```

```
RP/0/RP0/CPU0:PE1_5516#show int tunnel-ip2
tunnel-ip2 is up, line protocol is up
  Interface state transitions: 1
  Hardware is Tunnel
  Internet address is 80.80.82.1/24
  MTU 1500 bytes, BW 100 Kbit (Max: 100 Kbit)
    reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation TUNNEL_IP, loopback not set,
  Last link flapped 1d18h
  Tunnel TOS 0
  Tunnel mode GRE IPV4, encap
  Keepalive is disabled.
  Tunnel source 11.11.12.1 (Loopback4), destination 11.4.2.2/32
  Tunnel TTL 255
  Last input never, output never
  Last clearing of "show interface" counters 14:53:37
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 bytes, 0 total input drops
    0 drops for unrecognized upper-level protocol
  Received 0 broadcast packets, 0 multicast packets
    0 packets output, 0 bytes, 0 total output drops
  Output 0 broadcast packets, 0 multicast packets
```

Verify if the tunnel mode GRE encapsulation and decapsulation are enabled.

```
Router# sh interfaces tunnel-ip 5 accounting
Wed May 16 01:50:57.258 UTC
tunnel-ip5
  Protocol          Pkts In      Chars In     Pkts Out     Chars Out
  IPV4_UNICAST      489          55746        0             0
  IPV6_UNICAST      489          55746        0             0
  MPLS              587          69266        0             0
```

Verify if the recycle of the packets are not done under Recycle VoQ: 48:

```
Router# show tunnel ip ea summary location 0/7/CPU0

Number of tunnel updates to retry: 0
Number of tunnel updates retried: 0
Number of tunnel retries failed: 0
Platform:
Recycle VoQ: 48
  ReceivedBytes  ReceivedPackets  ReceivedKbps
  DroppedBytes  DroppedPackets   DroppedKbps

NPU 0:0
  0 0 0
  1 0 0
  2 0 0
  3 0 0
  ...
NPU 1:0
  0 0 0
  1 0 0
  2 0 0
  3 0 0
```

```

NPU 2:0      0          0          0
              0          0          0
1            0          0          0
              0          0          0
2            0          0          0
              0          0          0
3            0          0          0
              0          0          0

```

Verify if the tunnel mode GRE encapsulation is enabled.

```
Router# show interfaces tunnel-ip * brief
```

```

Thu Sep 7 00:04:39.125 PDT
Intf Intf LineP Encap MTU BW
Name   State   State   Type      (byte) (Kbps)
-----
ti30001 down    down    TUNNEL_IP 1500  100
ti30002 up      up      TUNNEL_IP 1500  100

```

Verify the tunnel endpoint route in RIB.

```
Router# show route 10.1.1.1
```

```

Routing entry for 10.0.0.0/8
Known via "static", distance 1, metric 0 (connected)
Installed Oct 2 15:50:56.755 for 00:39:24
Routing Descriptor Blocks
directly connected, via tunnel-ip109
Route metric is 0, Wt is 1
No advertising protos.

```

Verify if the tunnel mode GRE encapsulation is enabled.

```
Router# show tunnel ip ea database tunnel-ip 109 location 0/7/CPU0
```

```

----- node0_0_CPU0 -----
tunnel ifhandle 0x80022cc
tunnel source 161.115.1.2
tunnel destination 162.1.1.1/32
tunnel transport vrf table id 0xe0000000
tunnel mode gre ipv4, encap
tunnel bandwidth 100 kbps
tunnel platform id 0x0
tunnel flags 0x40003400
IntfStateUp
BcStateUp
Ipv4Caps
Encap
tunnel mtu 1500
tunnel tos 0
tunnel ttl 255
tunnel adjacency flags 0x1
tunnel o/p interface handle 0x0
tunnel key 0x0, entropy length 0 (mask 0xffffffff)
tunnel QT next 0x0
tunnel platform data (nil)
Platform:
Handle: (nil)
Decap ID: 0
Decap RIF: 0
Decap Recycle Encap ID: 0x00000000
Encap RIF: 0
Encap Recycle Encap ID: 0x00000000
Encap IPv4 Encap ID: 0x4001381b
Encap IPv6 Encap ID: 0x00000000

```

```
Encap MPLS Encap ID: 0x00000000
DecFEC DecRcyLIF DecStatsId EncRcyLIF
```

Verify if the QoS table is updated properly.

```
Router# show controllers npu stats voq base 48 instance all location
0/0/CPU0
```

```
Asic Instance = 0
VOQ Base = 48
```

	ReceivedPkts	ReceivedBytes	DroppedPkts	DroppedBytes
COS0 = 0	0	0	0	0
COS1 = 0	0	0	0	0
COS2 = 0	0	0	0	0
COS3 = 0	0	0	0	0

```
Asic Instance = 1
VOQ Base = 48
```

	ReceivedPkts	ReceivedBytes	DroppedPkts	DroppedBytes
COS0 = 0	0	0	0	0
COS1 = 0	0	0	0	0
COS2 = 0	0	0	0	0
COS3 = 0	0	0	0	0

```
Asic Instance = 2
VOQ Base = 48
```

	ReceivedPkts	ReceivedBytes	DroppedPkts	DroppedBytes
COS0 = 0	0	0	0	0
COS1 = 0	0	0	0	0
COS2 = 0	0	0	0	0
COS3 = 0	0	0	0	0

