



## Remote Node Management Using GCC

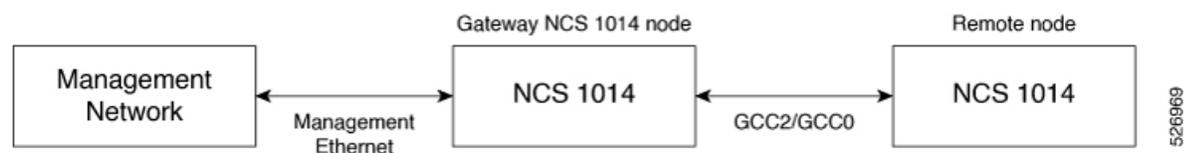
- [Remote node management using GCC, on page 1](#)
- [Configure the GCC0 interface on a QXP card, on page 2](#)
- [Verify the GCC0 interface status and IPv4 configuration, on page 3](#)
- [Configure the MTU to prevent IP fragmentation on GCC0 for SCP, on page 4](#)

## Remote node management using GCC

Remote node management using GCC is a network management method that

- leverages the General Communication Channel (GCC) embedded in optical transport networks,
- delivers reliable, out-of-band communication between centralized controllers and remote network nodes, and
- enables real-time monitoring, configuration, and maintenance activities without requiring direct physical access to each node.

**Figure 1: Remote node management in linear topology**



The remote nodes can be dynamically discovered over the GCC interface using OSPF. The connectivity to the management network can be achieved using OSPF and static routes.

## Supported protocols

These protocols are supported over the GCC interface:

- PING
- SSH
- TELNET
- SCP

- TFTP
- FTP
- SFTP
- HTTP
- HTTPS
- OSPF
- SNMP
- NETCONF

## Configure the GCC0 interface on a QXP card

Use this task to enable and configure the GCC0 interface on a Coherent DSP controller in a QXP card. Assign an IPv4 address to the interface to facilitate configuration.

The GCC0 interface operates at a data rate of 7.7 Mbps on the QXP card.

**Table 1: Feature History**

Feature Name	Release Information	Feature Description
GCC0 interface support on NCS1K4-QXP-K9 card	Cisco IOS XR Release 26.1.1	General Communication Channel (GCC0) interface is supported in 400G TXP and MXP modes (only in TRUNK mode OR) for the DP04QSDD-HK9 pluggable on the NCS1K4-QXP-K9 card. GCC0 interface enables you to remotely manage, monitor, and operate the chassis and line cards, especially in environments without direct Data Communication Network (DCN) access.



**Note** Open Config is not supported.

Follow these steps to configure the GCC0 interface on a QXP card.

### Procedure

**Step 1** Enter configuration mode for the Coherent DSP controller and enable the GCC0 interface.

**Example:**

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller coherentDSP 0/0/0/0
RP/0/RP0/CPU0:ios(config-CoDSP)#gcc0
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
```

**Step 2** Enter the **ipv4 address** *ipv4-address net-mask* command to assign the IPv4 address and subset mask to the GCC0 interface.

**Example:**

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#interface gcc0 0/0/0/0
RP/0/RP0/CPU0:ios(config-CoDSP)#ipv4 address 192.0.2.1 255.255.255.0
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
```

**Step 3** Enter the **interface Loopback R/S/I/P ipv4 address** *ipv4-address* command to configure the interface loopback.

**Example:**

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#interface Loopback0
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 20.1.1.1 255.255.255.255
```

**Step 4** Enter the **ipv4 unnumbered loopback 0** command to configure the GCC0 interface using the loopback IP address.

**Example:**

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#interface GCC0 0/1/0/0
RP/0/RP0/CPU0:ios(config-if)#ipv4 unnumbered loopback 0
RP/0/RP0/CPU0:ios(config-if)#exit
RP/0/RP0/CPU0:ios(config)#exit
```

---

The GCC0 interface is enabled with the specified IPv4 address and configured on the Coherent DSP controller of the QXP card.

## Verify the GCC0 interface status and IPv4 configuration

Use this task to confirm the operational status and assigned IPv4 address of GCC0 interfaces.

### Procedure

---

Run the **show ipv4 interface brief** command to display a summary of IPv4 interfaces.

**Example:**

```
RP/0/RP0/CPU0:ios#show ipv4 interface brief
Tue Sep 16 00:40:52.056 UTC
Interface                IP-Address      Status      Protocol      Vrf-Name
GCC00/0/0/0              198.51.100.51  Up          Up            default
MgmtEth0/RP0/CPU0/0     192.0.2.32     Up          Up            default
MgmtEth0/RP0/CPU0/1     unassigned     Shutdown   Down          default
MgmtEth0/RP0/CPU0/2     unassigned     Shutdown   Down          default
```

The output displays the IPv4 address, status, and protocol for GCC0 interfaces, confirming their configuration.

---

## Configure the MTU to prevent IP fragmentation on GCC0 for SCP

Use this task to prevent IP fragmentation on GCC0 interfaces during SCP protocol operations by limiting the maximum transmission unit (MTU).

IP fragmentation is not supported on GCC0 interfaces for the SCP protocol. To avoid fragmentation, configure the interface to restrict the maximum packet size to less than 1454 bytes, which is the fragmentation limit.

### Procedure

---

Enable the GCC0 interface and enter the **ipv4 mtu** *size* command to set the IPv4 MTU size for the GCC0 interface.

The MTU size must be 1200 bytes.

#### Example:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#interface gcc0 0/0/0/0
RP/0/RP0/CPU0:ios(config-if)#ipv4 mtu 1200
RP/0/RP0/CPU0:ios(config-if)#commit
```

---

The IPv4 MTU is now configured on the GCC0 interface. This setting limits the maximum packet size and prevents SCP-related fragmentation.