

Ports and Port Channels

This chapter contains the following sections.

- Configuring Port Channels, on page 1
- Configuring LLDP, on page 7
- Configuring Administrative Status of a Port, on page 8
- Tracking Changes for a Port, on page 8
- Speed, Duplex and Autonegotiation, on page 9

Configuring Port Channels

Information About Port Channels

Port channels combine individual links into a group to create a single logical link that provides the aggregate bandwidth of up to eight physical links. Creating port channels helps to increase bandwidth and redundancy and to load balance traffic between the member ports. If a member port within a port channel fails, the traffic from the failed port switches to the remaining member ports.

Port channels must have atleast two ports and can be configured using static mode or Link Access Control Protocol (LACP). Configuration changes that are applied to the port channel are applied to each member port of the port channel. A port channel can also be added to a bridge. When a port channel has two or more than two members and the port channel is added to a bridge, a bond is created.

A port can be a member of only one port channel and all the ports in a port channel must be compatible. Each port must use the same speed and operate in full-duplex mode.

Note

- The Physical Network Interface Controllers (PNICs) added to the port channel should be uniform. For example, all the PNICs associated with the port channel must have SRIOV VFs or they should not have SRIOV VFs.
- The Data Plane Development Kit (DPDK) can be associated only with port channels that have no SRIOV VFs attached to them. When a port channel is attached to a bridge and if the port channel has SRIOV VFs attached, the bridge gets automatically downgraded to a non-DPDK bridge.

Port Channels Bond Mode

A port channel can be configured for the following bond modes:

- **`active-backup`**: In this mode, one of the ports in the aggregated link is active and all others ports are in the standby mode.
- **`balance-slb`**: In this mode, load balancing of traffic is done based on the source MAC address and VLAN.
- **`balance-tcp`**: In this mode, 5-tuple (source and destination IP, source and destination port, protocol) is used to balance traffic across the ports in an aggregated link.

Port Channels LACP Mode

A port channel can be configured for the following LACP modes:

- **`off`**: Indicates that no mode is applicable.
- **`active`**: Indicates that the port initiates transmission of LACP packets.
- **`passive`**: Indicates that the port only responds to the LACP packets that it receives but does not initiate the LACP negotiation.

Port Channels Bond Mode

A port channel can be configured for the following bond modes:

- **active-backup**: In this mode, one of the ports in the aggregated link is active and all others ports are in the standby mode.
- **balance-slb**: In this mode, load balancing of traffic is done based on the source MAC address and VLAN.
- **balance-tcp**: In this mode, 5-tuple (source and destination IP, source and destination port, protocol) is used to balance traffic across the ports in an aggregated link.

Port Channels LACP Mode

A port channel can be configured for the following LACP modes:

- **off**: Indicates that no mode is applicable.
- **active**: Indicates that the port initiates transmission of LACP packets.
- **passive**: Indicates that the port only responds to the LACP packets that it receives but does not initiate the LACP negotiation.

Configuring a Port Channel

Creating a Port Channel

To create a port channel:

```
configure terminal
```

```
pnic egroup type port_channel lacp_type active bond_mode balance-tcp trunks 10,20
commit
```

Note Ensure to commit the changes.

Adding a Port to a Port Channel

You can add a port to a new port channel or a port channel that already contains ports. To add a port to a port channel:

Adding GE0-0 and GE0-1 to egroup:

```
configure terminal
pnic GE0-0 member_of egroup
commit
```

Note Ensure to commit the changes.

```
configure terminal
pnic GE0-1 member_of egroup
commit
```

Note Ensure to commit the changes.

Adding a Port Channel to a Bridge

You can add a port channel to a new bridge or an existing bridge. When a port channel is added to a bridge, a bond is added for the port channel.

To add a port channel to a bridge:

```
configure terminal
bridges bridge test-br port egroup
commit
```

Note Ensure to commit the changes.

Deleting a Port Channel

Before deleting a port channel, you must remove all members assigned to the port channel. If the port channel is configured on the bridge, you must remove the port channel from the bridge.

1. Remove ports from port channel. If GE0-0 and GE0-1 are part of port channel pc, remove them from pc first.

```
configure terminal
no pnic pc GE0-0 member_of egroup
commit
```



Note Ensure to commit the changes.

```
configure terminal
no pnic GE0-1 member_of egroup
commit
```



Note Ensure to commit the changes.

2. Remove port channel from the bridge.

```
configure terminal
no bridges bridge test-br port egroup
commit
```



Note Ensure to commit the changes.

3. Delete port channel.

```
configure terminal
no pnic egroup
commit
```



Note Ensure to commit the changes.

Verifying Port Channel Configurations

To verify port channel configurations, use the **show port-channel** command.

```
nfvis# show port-channel

----bond-egroup----
bond_mode: balance-tcp
bond may use recirculation: yes, Recirc-ID : 1
bond-hash-basis: 0
updelay: 0 ms
downdelay: 0 ms
next rebalance: 6921 ms
lacp_status: negotiated >>>this should be negotiated to indicate port channel is active
lacp_fallback_ab: false
active slave mac: 38:90:a5:1b:fe:0d(GE0-1)>>should indicate active slave mac address
```

```
slave GE0-0: enabled
may_enable: true

slave GE0-1: enabled
active slave >>>active slaveport should show active
may_enable: true
```

Port Channel APIs and Commands

APIs	Commands
/api/config/pnics /api/config/pnics/pnic/<pnic_name>/member_of /api/config/pnics/pnic/<pnic_name>/bond_mode /api/config/pnics/pnic/<pnic_name>/trunks	pnic <port_channel_name> type port_channel pnic <pnic_name> member_of <portchannel_name> show port-channel

Adding a Port to a Port Channel

A port channel must have at least two ports. A bond is created when a port channel has more than two members and the port channel is added to a bridge. You can add a port to a new port channel or a port channel that already contains ports.

To add a port to a port channel:

```
configure terminal
pnic eth1 member_of pc
commit
```

Adding a Port to a Port Channel APIs and Commands

APIs	Commands
/api/config/pnics/pnic/*name*/member_of	pnic *name* member_of *portchannel_name*

Adding a Port Channel to a Bridge

You can add a port channel to a new bridge or an existing bridge. When a port channel is added to a bridge, a bond is added for the port channel.

To add a port channel to a bridge:

```
configure terminal
bridges bridge test-br port pc
commit
```

Adding a Port Channel to a Bridge APIs and Commands

APIs	Commands
/api/config/bridges/bridge/*bridgename*	bridges bridge *name* port *portchannel_name*

Deleting a Port Channel

Before deleting a port channel, you must remove all members assigned to the port channel. If the port channel is configured on the bridge, you must remove the port channel from the bridge.

To delete a port channel:

```
configure terminal
no pnic pc
commit
```

Port Channel Deletion APIs and Commands

APIs	Commands
/api/config/pnics/pnic/*portchannel_name*	no pnic *portchannel_name*
/api/operational/pnics	show pnic

Removing a Port from a Port Channel

To remove a port from a port channel:

```
configure terminal
no pnic eth1 member_of pc
commit
```

Removing a Port from a Port Channel APIs and Commands

APIs	Commands
/api/config/pnics/pnic/*name*/member_of	no pnic *name* member_of *portchannel_name*

Removing a Port Channel from a Bridge

To remove a port channel from a bridge:

```
configure terminal
no bridges bridge test-br port pc
commit
```

Removing a Port Channel from a Bridge APIs and Commands

APIs	Commands
/api/config/bridges/bridge/*bridgename*	no bridges bridge *bridgename* port *portname*

Configuring LLDP

Starting from NFVIS 3.7.1 release, LLDP is supported on NFVIS. The Link Layer Discovery Protocol (LLDP) is used by network devices for advertising their identity, capabilities, and neighbors. You can configure LLDP on a PNIC which is not a port channel or a DPDK port. By default, LLDP is disabled for all PNICs.

LLDP information is sent by devices from each of their interfaces at a fixed interval, in the form of an Ethernet frame. Each frame contains one LLDP Data Unit (LLDPDU). Each LLDPDU is a sequence of type-length-value (TLV) structures.

LLDP is enabled in transmit and receive mode. The LLDP agent can transmit the local system capabilities and status information and receive the remote system's capabilities and status information

If LLDP is enabled on two connected devices, they can see each other as neighbors.



Note LLDP packets are not propagated to VMs. LLDP cannot be enabled on port channel or DPDK ports.

To enable LLDP on a PNIC:

```
configure terminal
pnic eth0 lldp enabled
commit
```

To disable LLDP on a PNIC:

```
configure terminal
pnic eth0 lldp disabled
commit
```

Use the **show lldp neighbors** command to display the peer information:

```
nfvis# show lldp neighbors eth0
----------------------------------------------------------------
DEVICE
NAME ID      HOLDTIME  CAPS  PLATFORM  PORTID  DESCRIPTION

----------------------------------------------------------------
eth0 Switch1623 120 Bridge, Router Cisco IOS Software, Catalyst L3 Switch Software
(CAT3K_CAA-UNIVERSALK9-M), Version 15.0(1)EX3, RELEASE SOFTWARE (fc2)Ifname:
Gi1/0/4GigabitEthernet1/0/4
```

Use the **show lldp stats** command to display the tx and rx information:

```
nfvis# show lldp stats eth0
-------------------------------------------------------------------
TX      DISCARD  ERROR  RX      DISCARDED  UNREC
NAME   FRAMES  RX       RX     FRAMES    TLVS       TLVS   AGEOUTS
-------------------------------------------------------------------
eth0    23      0        0       19667     0         0         0
```

LLDP Configuration APIs and Commands

APIs	Commands
/api/config/pnics/pnic/<pnic_name>/lldp /api/operational/lldp/neighbors /api/operational/lldp/stats	pnic <pnic_name> lldp enabled pnic <pnic_name> lldp disabled show lldp neighbors <pnic_name> show lldp stats <pnic_name>

Configuring Administrative Status of a Port

Administrative status provides a mechanism for configuring the administrative status of a port. It can be set to up or down and the default setting is on.

Note Administrative status cannot be enabled on port channel.

To configure the admin status on a pnic for a VM:

```
configure terminal
pnic GE0-1 admin status down
commit
```

Use the **show pnic** command to verify the admin status configuration. Use the **show pnic link_state** command to verify the admin state configuration.

```
nfvis# show pnic GE0-1 link_state
link_state down
```

Admin Status Configuration APIs and Commands

APIs	Commands
/api/config/pnics/pnic/<pnic_name>/adminstatus	pnic <pnic_name> adminstatus

Note Speed and duplex values in **show pnic** and **ethtool** outputs may differ depending on the peer device's interface speed and duplex settings.

Tracking Changes for a Port

Note This feature is supported only on ENCS 5400 starting from NFVIS 3.10.1 release.

In a virtual environment when the PNIC goes down there is no indication to the interfaces inside the VNFs. It is useful to track state changes of PNICs including switch ports to one or more VNF interfaces and accordingly bring down or up the vNICs. This feature brings the appropriate interfaces inside the VNF up or down based on the PNIC state changes. Most of the VNFs support this functionality.

Track state can also be configured for LAN-SRIOV. The LAN network is not physically connected to LAN-SRIOV. Switch ports are connected to an embedded switch on the LAN side. The switch has an int-LAN interface which is a 10G interface the VMs can connect to from the LAN network using VFs (virtual functions). Therefore, the VM is not directly connected to LAN-SRIOV.

Track state configuration on WAN-SRIOV is not needed, as there is a one to one connection between WAN-SRIOV and the VM.

Track state can be configured for monitored and un-monitored VMs. If a track state configuration is deleted, the PNIC or switch port state changes will not be notified to the vNICs or VFs.

The VM has to be first deployed before you can configure PNIC track state for the VM. VNFs or vNICs do not have to be attached to a bridge connected to the PNIC.

To configure track state on a pnic for a VM use the following commands: **pnic <pnic_name> track-state <vm_name> <vnic>** or **pnic <pnic_name> track-state <deploy_name.vm_grp_name> <vnic>**

```
configure terminal
pnic GE0-0 track-state ROUTER 0
end
```

To verify the track state configuration on the VM use the **show interface** or **ethtool** commands or the VM specific command that displays the interface link state.

In the following example, the vedge VM deployed and vNIC 0 is being tracked by GE0-1. The **if-oper-status** command shows the state of the vNIC being tracked by pNIC. When GE0-1 is down, **if-oper-status** also shows as down.

Track StateAPIs and Commands

Track StateAPIs	Track State Commands
• api/config/pnics/pnic/<pnic_name>/track-state	• pnic <pnic_name> track-state <vm_name> <vnic> • pnic <pnic_name> track-state <deploy_name.vm_grp_name> <vnic>

Speed, Duplex and Autonegotiation

NFVIS supports autonegotiation by default on all PNICs. Speed and duplex are set to *auto* mode to indicate autonegotiation is enabled.

Autonegotiation allows a PNIC to communicate with the device on the other end of the link to determine the optimal duplex mode and speed for the connection. Autonegotiation can be turned off by configuring speed and duplex. Supported Ethernet speed is 10 Mbps, 100 Mbps, and 1G and 10 G.

Duplex mode displays the data flow on the interface. Duplex mode on an interface can be full or half duplex. A half-duplex interface can only transmit or receive data at any given time and a full-duplex interface can send and receive data simultaneously.

When autonegotiation is enabled on a port, it does not automatically determine the configuration of the port on the other side of the ethernet cable to match it. Autonegotiation only works if it is enabled on both sides of the link. If one side of a link has auto-negotiation enabled, and the other side of the link does not, then autonegotiation cannot determine the speed and duplex configurations of the other side. If autonegotiation is enabled on the other side of the link, the two devices decide together on the best speed and duplex mode. Each interface advertises the speed and duplex mode at which it can operate, and the best match is selected. Higher speed and full duplex is the preferred mode.

If one side of a link does not have autonegotiation enabled, then the speed and duplex on both sides must match so that the data can transmit without collisions. Autonegotiation fails on 10/100 links, if one side of the link has been set to 100/full, and the other side has been set to autonegotiation which is 100/half.



Note Not all ports on ENCS 5000 series devices support auto-mdix feature. When autonegotiation is disabled, you need to use the correct cable to configure speed and duplex correctly. The cable type depends on the remote system, based on which you can try straight through or cross over cable.

To disable autonegotiation on a PNIC, speed and duplex must be configured:

```
configure terminal
pnic GE0-0 speed 100 duplex full
commit
```

To enable autonegotiation on a PNIC:

```
configure terminal
pnic GE0-0 speed auto duplex auto
commit
```

To configure speed and duplex with non auto values:

```
configure terminal
pnic GE0-0 speed 100 duplex full
commit
```

Use the **show pnic GE0-0 operational-speed**, **show pnic GE0-0 operational-duplex** and **show pnic GE0-0 autoneg** to verify the configurations.

```
nfvis# show pnic GE0-0 operational-speed
operational-speed 100


nfvis# show pnic GE0-0 operational-duplex
operational-duplex full


nfvis# show pnic GE0-0 autoneg
autoneg off
```

To verify the PNIC speed and duplex configurations, use the **show notification stream nfvis Event** command.

```
notification
event Time 2019-12-16T22:52:49.238604+00:00
```

```
nfvisEvent
    user_id admin
    config_change true
    transaction_id 0
    status FAILURE
    status_code 0
    status_message Pnic GE0-1 speed did not update successfully
    details NA
    event_type PNIC_SPEED_UPDATE
    severity INFO
    host_name nfvis
    !
!
notification
event Time 2019-12-16T22:53:05.01598+00:00
nfvisEvent
    user_id admin
    config_change true
    transaction_id 0
    status SUCCESS
    status_code 0
    status_message Pnic GE0-1 duplex updated successfully:full
    details NA
    event_type PNIC_DUPLEX_UPDATE
    severity INFO
    host_name nfvis
    !
!
```

Speed, Duplex, and Autonegotiation APIs and Commands

Speed, Duplex and Autonegotiation APIs	Speed, Duplex and Autonegotiation Commands
/api/config/pnics/pnic/GE0-0/speed	pnic GE0-0 speed auto duplex auto
/api/config/pnics/pnic/GE0-0/duplex	pnic GE0-0 speed 100 duplex full show
/api/operational/pnics/pnic/GE0-0/operational-speed	show pnic GE0-0 operational-speed
/api/operational/pnics/pnic/GE0-0/operational-duplex	show pnic GE0-0 operational-duplex
/api/operational/pnics/pnic/GE0-0/autoneg	show pnic GE0-0 autoneg