VirtualWire™ Configuration Guide

Version 5.2.1

June 2020

Table of Contents

Configuring VirtualWire™ Features

This chapter provides information for understanding and configuring the VirtualWire™ features on a Pluribus switch. This chapter includes:

Note: This feature is supported on all Dell and Freedom/Edgecore platforms .

Prerequisites

To install and configure VirtualWire, ensure the following prerequisites are followed:

- Refer to Adding License Keys to Netvisor ONE section for how to install a license on the switch.

- VirtualWire functionality is available for all supported Pluribus Network transceiver at 1Gbs, 10Gbs, 25Gbs, 40Gbs, or 100 Gbs. For a list of supported transceivers and licenses, please refer to the product data sheet.

- All commands described in this chapter requires a fabric over a management interface. Refer to Configuring the Fabric Over the Management Interface section for information on how to create or join a fabric over a management interface .

- To add the VirtualWire feature to an existing Pluribus Networks switch in your network, you must use the `switch-config-reset` command to erase the current configuration and reset the switch configuration to factory default.

- After re-configuring the initial setup, you must upgrade to the latest version of Netvisor One that supports VirtualWire mode. And then, install the license key for VirtualWire. Refer to the Installing Netvisor ONE and Initial Configuration chapter for details.

- You must re-join the fabric after re-configuring the switch to VirtualWire mode. See the Configuring and Administering the Pluribus Fabric chapter for details.

Adding License Keys to Netvisor ONE

Netvisor ONE binds the license key to the serial number of the switch and when downloading the Netvisor ONE software, the Pluribus Networks Cloud locates the serial number.

To install the license key, use the following syntax:

```
CLI (network-admin@switch) > software-license-install key
license-key
```

The license key has the format of four words separated by commas. For example,

```
    License Key: rental,deer,sonic,solace
```

Once the license key is installed, you can display information about the key using the following command:

```
CLI (network-admin@Leaf1) > software-license-show

switch:              T6001-ON
license-id:          NVOS-CLD-LIC-60D
description:         Pluribus Open Netvisor OS Linux Cloud
Edition License
expires-on:          never
status:              VALID
```

Enabling Administrative Services

There are many features of the Pluribus Networks fabric that require or can be enhanced using remote access. For example, when packets are written to a log file, you may want to transfer that file from a switch to a different system for analysis. Also, if you are creating a NetVM environment, an IOS image of the guest OS must be loaded on the switch.

You can enhance or modify several services such as SSH, NFS, Web, SNMP, SFTP.

To check the status of various services, use the following command:

```
CLI (network-admin@Leaf-1) > admin-service-show

switch:                Leaf-1
if:                    mgmt
ssh:                   on
nfs:                   on
web:                   on
web-ssl:               off
web-ssl-port:          443
web-port:              80
web-log:               off
snmp:                  on
net-api:               on
icmp:                  on

switch:                Leaf-1
if:                    data
ssh:                   on
nfs:                   on
web:                   on
web-ssl:               off
web-ssl-port:          443
web-port:              80
web-log:               off
snmp:                  on
net-api:               on
icmp:                  on
```

Netvisor ONE supports the file transfer method, SFTP and SFTP is enabled by default on Netvisor ONE. Because SFTP relies on Secure Shell (SSH), you must enable SSH before enabling SFTP.

To enable SSH, use the following command

```
CLI (network-admin@Leaf1) > admin-service-modify nic mgmt ssh
```

To enable SFTP, use the following command:

```
CLI (network-admin@Leaf1) > admin-sftp-modify enable
```

```
sftp password: <password>
confirm sftp password: <password>
```

The default SFTP username is sftp and the password can be changed using the `admin-sftp-modify` command:

```
CLI (network-admin@Leaf1) > admin-sftp-modify

sftp password: <password>
confirm sftp password: <password>
```

To display the details, use the following commands:

```
CLI (network-admin@Leaf-1) > admin-service-show

switch  if   ssh  nfs web web-ssl web-ssl-port web-port snmp
net-api icmp
------  ---- ---  --- --- ------- ------------ -------- ----
------- ----
Leaf-1  mgmt  on  off off  off      443          80      on
off     on
Leaf-1  data  on  off off  off      443          80      on
off     on

admin-service-show: Fabric required. Please use fabric-
create/join/show

CLI (network-admin@Leaf1) > admin-sftp-show

switch:     Leaf1
sftp-user:  sftp
enable:     yes
```

Use SFTP from a host to the switch, and login with the username **sftp** and the password configured for SFTP. Then you can download the available files or upload files to the switch.

```
CLI (network-admin@Leaf1) > admin-service-show

switch nic  ssh  nfs web web-port snmp net-api icmp
------ ---  ---  --- --- -------- ---- ------- ----
Leaf1  mgmt  on  off  on   80     off   on      on
```

Configuring Administrative Session Timeout

Netvisor ONE sets the administrator sessions to timeout after 60 minutes (by default) of idle time or no activity, but allows you to change the timeout value to a user desirable time. During the session timeout, you are logged out of the CLI and the Shell prompt and your privileges changes to *root user*. To access the switch, you must login again using the CLI or Shell prompt.

To verify the default or user configured session timeout value, use the command:

```
CLI (network-admin@Spine1) > admin-session-timeout-show

switch: Spine1
timeout: 1h
```

To modify the timeout value, use the command:

```
CLI (network-admin@Spine1) > admin-session-timeout-modify
```

`timeout duration: #d#h#m#s`	Specify the maximum time to wait for user inactivity before terminating login session.

Confirming Connectivity on the Network

After connecting your switch, take the time to ensure connectivity by pinging an external IP address (supports both IPv4 and IPv6), and pinging a domain to ensure domain name resolution.

To ping the external network from the switch, use the `ping` command:

```
CLI (network-admin@switch) > ping 2010::2
```

```
PING 2010::2(2010::2) 56 data bytes
64 bytes from 2010::2: icmp_seq=1 ttl=64 time=1.69 ms
64 bytes from 2010::2: icmp_seq=2 ttl=64 time=0.412 ms
64 bytes from 2010::2: icmp_seq=3 ttl=64 time=0.434 ms
64 bytes from 2010::2: icmp_seq=4 ttl=64 time=0.418 ms
```

To ping an IP address from the switch, use the ping command:

```
CLI (network-admin@switch) > ping 98.138.253.109 : 56 data
bytes
```

```
PING 98.138.253.109 (98.138.253.109) 56(84) bytes of data.
64 bytes from 98.138.253.109: icmp_seq=1 ttl=47 time=51.8 ms
64 bytes from 98.138.253.109: icmp_seq=2 ttl=47 time=51.9 ms
64 bytes from 98.138.253.109: icmp_seq=3 ttl=47 time=53.6 ms
```

To ping a domain, use the ping command again:

```
CLI (network-admin@Leaf1) > ping yahoo.com
```

```
PING yahoo.com (98.138.253.109) 56(84) bytes of data.
64 bytes from ir1.fp.vip.ne1.yahoo.com (98.138.253.109): icmp_seq=1 ttl=47 time=52.2 ms
64 bytes from ir1.fp.vip.ne1.yahoo.com (98.138.253.109): icmp_seq=2 ttl=47 time=52.5 ms
64 bytes from ir1.fp.vip.ne1.yahoo.com (98.138.253.109): icmp_seq=3 ttl=47 time=51.9 ms
64 bytes from ir1.fp.vip.ne1.yahoo.com (98.138.253.109): icmp_seq=4 ttl=47 time=51.8 ms
```

Running Shell Commands or Scripts Using REST API

Netvisor ONE version 5.1.0 provides the ability to run shell commands or scripts using REST API or through CLI commands. As a network administrator or as an admin user, you can run the scripts from the directories */opt/nvOS/bin/pn-scripts* (directory and all files are delivered as part of pn-upgrade-agent package) and */usr/bin/pn-scripts* (backup directory for running custom scripts).

The commands introduced to enable this feature are: `pn-script-show` (to view all the available scripts) and `pn-script-run name <`*`script-name`*`>` (to run a specified script).

Usage Guidelines: To run a custom script,

- You should have permission to run the script.
- You should not have any duplicate scripts in the directories, */opt/nvOS/bin/pn-scripts* and */usr/bin/pn-scripts*. In case of duplicate scripts, the script from the directory, */opt/nvOS/bin/pn-scripts* takes precedence.
- It is not recommended to execute any scripts that are manually copied to the directory.

You can use the CLI commands or the vREST API to run the scripts. To run the scripts using the CLI commands, for example:

To display the available scripts, use the command:

```
CLI (network-admin@pn-lab1) > pn-script-show

switch: pn-lab1
pn-lab1: /opt/nvOS/bin/pn-scripts/:
testscript.sh
pn-testscript.sh
```

To run the script, use the command:

```
CLI (network-admin@pn-lab1) > pn-script-run name testscript.sh

Executing /opt/nvOS/bin/pn-scripts/testscript.sh:
Executing Test PN script!
```

To run the scripts using vREST API, use the following API call:

```
$ curl -u network-admin:test123  http://pn-lab1/vRest/api-docs/run-pn
```

```
{"apiVersion":"1.0.0","swaggerVersion":"1.2","basePath":"/vRes
t","resourcePath":"/run-pn","produces":
["application/json","application/x-ndjson"],"consumes":
["application/json"],"apis":[{"path":"/run-
pn/script","operations":
[{"method":"POST","summary":"","notes":"","type":"result-
list","nickname":"scriptRun","consumes":
["application/json"],"parameters":
[{"name":"body","required":false,"type":"run-pn-
script","paramType":"body","allowMultiple":false}]}]}],"models
":{"result":{"id":"result","required":["api.switch-
name","scope","status","code"],"properties":{"api.switch-
name":{"type":"string"},"scope":{"type":"string","enum":
["local","fabric"]},"status":{"type":"string","enum":
["Success","Failure"]},"code":
{"type":"integer","format":"int32"},"message":
{"type":"string"}}},"result-list":{"id":"result-
list","required":["status","result"],"properties":{"status":
{"type":"string","enum":["Success","Failure"]},"result":
{"type":"array","items":{"$ref":"result"}}}},"run-pn-script":
{"id":"run-pn-script","description":"Run PN
script","required":["name"],"properties":{"name":
{"type":"string","description":"desc=Script to
execute:pattern=^[a-zA-Z0-9_.:-]+$:pattern-help=letters,
numbers, _, ., :, and -"}}}}}

$ curl -u network-admin:test123  http://pn-lab1/vRest/pn-
script

     {"data":[{}],"result":{"status":"Success","result":
[{"api.switch-
name":"local","scope":"local","status":"Success","code":0,"mes
sage":"/opt/nvOS/bin/pn-scripts/:
\ntestscript.sh\n\n/usr/bin/pn-scripts/:\ntestscript.sh\ntest-
usr.sh\n"}]}}

$ curl -u network-admin:test123 -X POST http://pn-
lab1/vRest/pn-script/run -d '{"name":"testscript.sh"}' -H
"Content-Type: application/json"

     {"result":{"status":"Success","result":[{"api.switch-
name":"local","scope":"local","status":"Success","code":0,"mes
sage":"Executing /opt/nvOS/bin/pn-scripts/testscript.sh:
\nExecuting Test PN script!\n"}]}}
```

Understanding VirtualWire™ Technology in Netvisor ONE OS

Pluribus VirtualWire™ is an integrated physical layer feature set for the Netvisor® ONE Operating System (OS) that enables native layer 1 switching capabilities on Open Networking hardware switches. VirtualWire transforms a traditional electrical Ethernet connection to emulate a physical wired connection so that interconnections are mapped between two or more physical ports in single switch, or across a multi-switch topology. This feature enables you to interconnect devices into any topology without moving the cables around, which is a powerful capability in a network lab.

VirtualWire™ technology uses the software approach to configure cable topologies to interconnect network devices together. Network devices are physically connected to the VirtualWire switch once using Ethernet cables and transceivers that match the device port media and speed characteristics. The desired cable topology is then obtained by a remote software configuration of the Virtual Wire switch and consists of a set of Virtual Wire links. VirtualWire topology configurations can be dynamically created, saved and re-applied without any manual intervention on the physical infrastructure.

Enabling VirtualWire mode on a switch disables all the possible error-checks such as Cyclic Redundancy Check (CRC) and Runts (very small Ethernet packets with a minimum length of 50 bytes caused by excessive packet collisions). This feature also disables STP, LLDP, forwarding, and learning on all switch ports.

VirtualWire is implemented using transparent low-latency Ethernet forwarding between physical ports over a non-blocking any-port to any-port switching architecture. VirtualWire transparently cross bridges any standard or proprietary Ethernet protocol of any size, including these types of traffic:

- IPv6, Q-in-Q, VN-TAG
- Ethernet control plane traffic such as BPDU, LACP and LLDP protocol packets
- Proprietary or experimental Ethernet fabric
- Undersized or invalid frames

Network devices interconnected through a VirtualWire link behave as if the devices are directly connected with a single physical cable. For example, as shown in ***Figure 1-1***, if the port of Device A goes down, the VirtualWire switch automatically shuts down the port facing Device B.

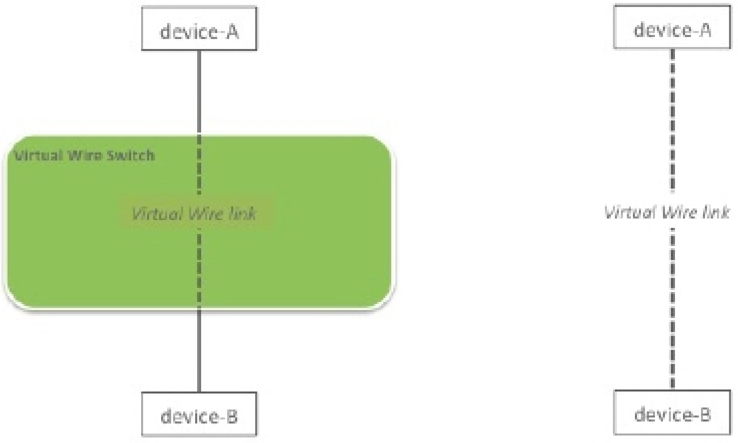


Figure 1-1 - Virtual Wire Topology

In addition, a VirtualWire switch can act as an intelligent media converter, enabling Ethernet communication between devices with different port speed and media type. That is, to provide transparent switching, you can use the port association functionality of Netvisor ONE to create a pseudo-wire between the master and slave ports.

In the example shown in ***Figure 1-2***, a VirtualWire link is created between an optical cable connecting device A and a copper cable on device B.

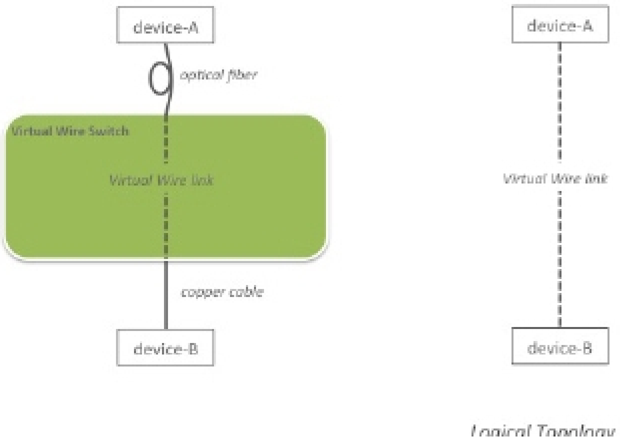


Figure 1-2 - Virtual Wire Topology with Optical and Copper Cables

Enabling VirtualWire Mode on a Switch

To setup a switch in VirtualWire mode, you must install the required license key by using the command:

```
CLI (network-admin@netvisor) > software-license-install  key
key-string
```

For example, to install the license key, ONVL-10G-VW+-LIC, use the command:

```
CLI (network-admin@netvisor) > software-license-install key
ONVL-10G-VW+-LIC
```

To view the license details on the VirtualWire switch, use the `software-license-show` command. For example,

```
CLI (network-admin@netvisor) > software-license-show

  license-id        description                  expires-on status
---------------- ---------------------------- ---------- ------
ONVL-10G-VW+-LIC 10G switches license for VW+   never      VALID
```

The following command instructs the switch to operate in VirtualWire mode and is used to enable global VirtualWire functionality on a switch:

First configure the switch as a virtual-wire bridge by using the `switch-mode-modify` command. For example,

```
CLI(network-admin@netvisor) > switch-mode-modify switch-mode
virtual-wire
```

`switch-mode-modify`	Use this command to modify the mode of a server switch.
`switch-mode store-and-forward|virtual-wire`	Specify the mode of the switch. Specifying the `virtual-wire` keyword modifies the switch as a VirtualWire switch.

Note: Enabling VirtualWire mode on a switch disables all the possible error-checks such as Cyclic Redundancy Check (CRC) and Runts (Runts are very small ethernet packets, upto 50 bytes and is caused by excessive packet collisions). This feature also disables STP, LLDP, Layer 2 learning on all switch ports as well as processing and forwarding of BPDUs .
Also, Jumbo frames are enabled by default on all ports in VirtualWire mode.
Additionally, the regular switches function like the VLANs and vRouters are not supported anymore.

To display the switch mode, use the `switch-mode-show` command:

```
CLI (network-admin@netvisor) > switch-mode-show

switch:      pn-spine1
switch-mode: virtual-wire
```

Configuring Ports for VirtualWire™ Mode

By default, all 10Gbs switch ports are configured for 10Gbs Ethernet speed.

In 10Gbs Ethernet mode, SFP+ or QSFP+ transceivers are required to connect to hosts or other switches. If you change the port speed to 1 Gigabit Ethernet, you need SFP transceivers to plug into the ports. You must also enable jumbo frames for the port.

To modify the ports to 1Gbs speed and enable auto-negotiation, use the following syntax:

```
CLI (network-admin@pn-spine1) > port-config-modify port 1-8
speed 1g autoneg
```

REST API Command : `PUT http://<switch-ip>/vRestport/port/port-configs/1-8/`

```
{
 "speed": "1g",
 "autoneg": "autoneg",
}
```

> **Note:** Jumbo frames are enabled by default and VirtualWire links support any frame size.

To display port configuration information, use the `port-config-show` command.

To see all output, add the parameters format `all layout vertical`.

Using the vertical layout displays the information in a more readable format:

```
CLI (network-admin@Leaf1) > port-config-show port 1 format all
layout vertical

switch:                 pn-spine1
intf:                   1
port:                   1
speed:                  1g
egress-rate-limit:      unlimited
autoneg:                on
jumbo:                  on
enable:                 on
lacp-priority:          32768
lacp-individual:        none
stp-port-cost:          2000
stp-port-priority:      128
reflect:                off
edge-switch:            no
```

```
pause:                  no
description:
loopback:               default
mirror-only:            off
lport:                  1
rem-rswitch-port-mac:   00:00:00:00:00:00
rswitch-default-vlan:   0
port-mac-address:       06:a0:00:02:40:1e
send-port:              0
routing:                yes
host-enable:            yes
```

REST API Command: `GET http://<switch-ip>/vRest/port-configs/1`

To display the port status, use the `port-show` command. For example, a sample output looks similar to the following:

```
CLI (network-admin@pn-spine1) > port-show format all layout
vertical

switch:            pn-spine1
port:              47
ip:                192.168.42.30
mac:               64:0e:94:28:03:56
hostname:          pubdev03
status:            up,PN-fabric,LLDP
lport:             47
rport:             47
config:            fd,10g
trunk:             trunk2
switch:            pn-spine1
port:              48
ip:                192.168.42.30
mac:               64:0e:94:28:03:56
hostname:          pubdev03
status:            up,PN-fabric,LLDP
lport:             48
rport:             48
config:            fd,10g
trunk:             trunk2
```

REST API Command: `GET http://<switch-ip>/vRest/ports`

To display all details about ports, use the `port-phy-show` command:

```
CLI (network-admin@pn-spine1) > port-phy-show format all
layout vertical

switch:        pn-spine1
port:          25
state:         up
autoneg:       none
```

```
speed:        10000
eth-mode:     10Gbase-cr
max-frame:    1540
link-quality: great (59/41)
learning:     off
def-vlan:     1
dfe-mode:     continuous
dfe-coarse:   complete
dfe-fine:     complete
switch:       pn-spine1
port:         26
state:        up
autoneg:      none
speed:        10000
eth-mode:     10Gbase-cr
max-frame:    1540
link-quality: good (57/38)
learning:     off
def-vlan:     1
dfe-mode:     continuous
dfe-coarse:   complete
dfe-fine:     complete
```

REST API Command: `GET http://<switch-ip>/vRest//port-phys`

> **Note:** The columns def-vlan, max-frame, and learning display default fixed values because regular switching is disabled on the ports.

> **Note:** Link-quality information is only available when a 10Gbps transceiver is installed in a port.

To display the transceivers connected to the ports, use the `port-xcvr-show` command:

```
CLI (network-admin@pn-spine1) > port-xcvr-show port 1-4

switch    port vendor-name      part-number  serial-number supported
-------- ---- ---------------- ------------ ------------ ---------
pn-spine1  1    PluribusNetworks SFP10-CU0P5M Y05B200393    Yes
pn-spine1  2    PluribusNetworks SFP10-CU0P5M Y05B200747    Yes
pn-spine1  3    PluribusNetworks SFP10-CU0P5M Y05B200413    Yes
pn-spine1  4    PluribusNetworks SFP10-CU0P5M Y05B200804    Yes
```

REST API Command: `GET http://<switch-ip>/vRest/port-xcvrs`

> **Note:** Each port has a LED indicator light that displays status information about the port. If the LED is solid green, the port is enabled. If the LED is green and blinking rapidly then the port is at 80% of the throughput capacity.

Implementing Unidirectional and Bidirectional VirtualWire Links

In this section you can configure a single bidirectional VirtualWire link using the `port-association-create` command with the option `virtual-wire`. Each port of the VirtualWire transmits traffic in full-duplex mode.

Use the `port-association-create` command:

```
CLI(network-admin@Spine1) > port-association-create name name-string master-ports port-list slave-ports port-list virtual-wire|no-virtual-wire
```

`port-association-create`	Creates a port association between the master and slave ports.
`name` *`name-string`*	Specify the name of the configuration
`master-ports` *`port-list`*	Specify the master port number or a list of ports that can act as master ports.
`slave-ports` *`port-list`*	Specify the slave port number or a list of ports that can act as salve ports.
`[`**`virtual-wire`**`|no-virtual-wire]`	Specify the `virtual-wire` keyword to form a virtual-wire port association. This command keyword creates two vFlows between the master and slave ports and re-directs all traffic from one port to another by creating a pseudo wire. It also creates a flow policy with Copy-to-CPU action on TCP packets (sync, ack, and rst) to provide analytics with tracking details. This keyword is available only when the switch is in VirtualWire mode.

Configuring a single bidirectional VirtualWire link using the `port-association-create` command with the option `virtual-wire` can be implemented in two ways that are functionally equivalent:

- Configuring VirtualWire direction individually - or
- Configuring a VirtualWire link using the `bidir` parameter

To configure a unidirectional VirtualWire link from device A to device B, enter the following command:

```
CLI (network-admin@Leaf1) > port-association-create name A-to-B virtual-wire master-ports 10 slave-ports 20 no-bidir
```

REST API Command: `POST http://<switch-ip>/vRes/port-associations`

```
{
“name”: “A-to-B”,
“virtual-wire”:”true”,
“master-ports”: “10”,
“slave-ports”: “20”
}
```

> **Note:** Please note that the parameter “mode(?) must be set to “true”. This is the case when the switch is running in VirtualWire mode and you are configuring VirtualWire features.

To configure a unidirectional Virtual Wire link from device B to device A, enter the following command:

```
CLI (network-admin@Leaf1) > port-association-create name B-to-A virtual-wire master-ports 20 slave-ports 10
```

> Note: If the `bidir|no-bidir` keywords are not mentioned in the above command, then, by default, a uni-directional association of VirtualWire link is configured.

REST API Command: `POST http://<switch-ip>/vRes/port-associations`

```
{
“name”: “B-to-A”,
“virtual-wire”:”true”,
“master-ports”: “20”,
“slave-ports”: “10”
}
```

To configure a bidirectional Virtual Wire link from device A to device B, enter the following command:

```
CLI (network-admin@Leaf1) > port-association-create name A-to-B bidir virtual-wire master-ports 10 slave-ports 20
```

REST API Command: `POST http://<switch-ip>/vRes/port-associations`

```
{
“name”: “A-to-B”,
“bidir”,
“virtual-wire”:”true”,
“master-ports”: “20”,
“slave-ports”: “10”
}
```

To display existing Virtual Wire links, use the `port-association-show` command:

```
CLI (network-admin@Leaf1) > port-association-show

switch     name    master-ports slave-ports  policy      virtual-wire bidir
---------  ------  -----------   ----------  ----------- ------------ -----
vw-switch  A-to-B    10            20        all-masters  true         true
```

REST API Command: `POST http://<switch-ip>/vRes/port-associations`

To delete an existing Virtual Wire link, use the `port-association-delete` command with the `name string` parameter:

```
CLI (network-admin@Leaf1) > port-association-delete name A-to-B
```

REST API Command: `POST http://<switch-ip>/vRes/port-associations/A-to-B`

Configuring CRC Checks for VirtualWire Mode

A switch running in VirtualWire Mode currently interpret the CRC header of the packets passing through. This achieves perfect transparency of the switch. However it does place limitations on the types of vFlows created on the switch, as any vFlow that modifies the packet renders the CRC on that packet invalid without updating it.

With this Netvisor One release, the CRC regeneration is a configurable option per port, so the you can decide on a per-port basis whether the switch should, or should not perform CRC regeneration.

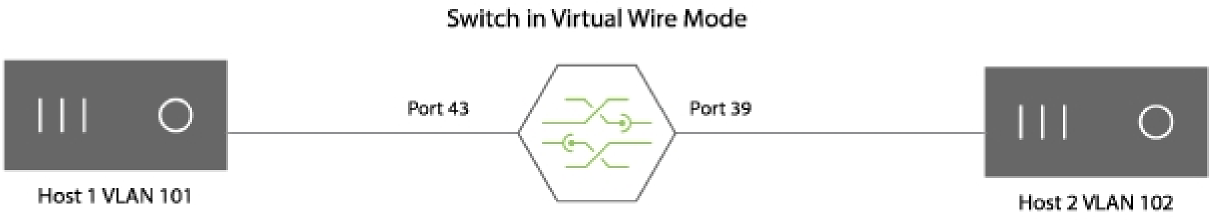


Figure 1-3 - Example Virtual Wire Mode Topology

On the virtual-wire switch, if you want to convert traffic on port 43 tagged with VLAN 101 to be tagged with VLAN 102 so Host1 and Host2 can communicate as if the two hosts are on the same VLAN, then you configure the following two vFlows:

```
CLI (network-admin@Leaf1) > vflow-create name vlan_map_101_102
scope local table L1-Virtual-Wire-1-0 vlan 101 in-port 43
precedence 15 action setvlan action-value 102 action-to-ports-
value 39

CLI (network-admin@Leaf1) > vflow-create name vlan_map_102_101
scope local table L1-Virtual-Wire-1-0 vlan 102 in-port 39
precedence 15 action setvlan action-value 101 action-to-ports-
value 43
```

However, the packets with a different VLAN now have an incorrect CRC value unless the CRC is updated when egressing the port.

For example, use the command:

```
CLI (network-admin@Leaf1) > port-config-modify port 39,43 crc-
check-enable
```

After this configuration, any packets egressing from ports 39 and 43 are updated with the CRC check.

Note: The parameter, `crc-check-enable` is only be available on switches in Virtual Wire mode. Furthermore, when the switch mode is changed to VirtualWire mode, all

> ports are configured as `crc-check-disable` by default.

Configuring Many to One Port Associations

To provide transparent switching, you can use the `port-association-create` and `port-association-modify` commands to create a pseudo-wire between the master and slave ports. The `virtual-wire` keyword enables analytics on associated ports and traffic between specified ports based on the *bidir* or *no-bidir* tag.

To create port associations between master port and slave ports and enabling link-tracking, use the command:

```
CLI(network-admin@Spine1) > port-association-create name name-
string master-ports port-list slave-ports port-list virtual-
wire|no-virtual-wire bidir|no-bidir
```

`port-association-create`	Creates a port association between the master and slave ports.
`name` *`name-string`*	Specify the name of the configuration
`master-ports` *`port-list`*	Specify the master port number or a list of ports that can act as master ports.
`slave-ports` *`port-list`*	Specify the slave port number or a list of ports that can act as salve ports.
`[`**`virtual-wire`**`|no-virtual-wire]`	Specify the `virtual-wire` keyword to form a virtual-wire port association. This enables analytics on associated ports and traffic between specified ports This keyword is available only when the switch is in VirtualWire mode.
`[bidir|no-bidir]`	Specify the `bidir` keyword to enable bidirectional port state link tracking, which sets-up virtual-wire vflows between master and slave ports. This keyword is available only when the switch is in VirtualWire mode.
`Other parameters available in the command for standard switch form are:`	
`[policy all-masters|any-master]`	Specifies the port association policy. The default is all-masters.
`[monitor-ports port-list]`	Specify the list of ports that needs to be monitored.
`[enable|no-enable]`	Specify to enable or disable port association in hardware.

> **Note**: To support analytics data, a few additional system vFlow entries (named System-vflow-x, where x can be S or F or R) are installed with a higher priority than the vFlow entry in order to copy TCP SYN/FIN/RST packets to the management CPU. This ensures that any SYN/FIN/RST packets carried by vFlow can be used for TCP flow analysis.

> **Note**: The difference between *many-to-one, one-to-many,* and *many-to-many* port associations are very important in *uni-directional* mode as the traffic goes only from the master ports to the slave ports in a *uni-directional port-association* and not the other way around.

For example,

```
CLI (network-admin@Leaf1) > port-association-create name PA_1
master-ports 1 slave-ports 2,3 virtual-wire

CLI (network-admin@Leaf1) > port-association-create name PA_2
master-ports 2 slave-ports 1,3 virtual-wire
```

The parameter, `monitor-ports`, is added to allow for ports that are not tracked by the port-association. Apart from non-tracking of the monitor port, the traffic is sent to the monitor port only and no traffic is allowed from the monitor port to the master or slave port.

This scenario can be used in cases such as sending data to a logging server (connected to a monitor port) between two network path ports (master and slave ports).

```
CLI (network-admin@Leaf1) > CLI> port-association-create name
PA_1 master-ports 1 slave-ports 2 monitor-ports 3 virtual-wire

CLI (network-admin@Leaf1) > CLI> port-association-create name
PA_2 master-ports 2 slave-ports 1 monitor-ports 3 virtual-wire
```

These commands create the same set of port-associations except that when ports 1 or 2 goes down, port 3 is not affected.

> Note: The `virtual-wire` and `bidir` keywords are available only on VirtualWire switch mode.

Configuring Packet Load Balancing over One to Many Links

When VirtualWire is deployed as legacy packet broker, moving packets from production to an analyzer tool, it requires load balancing feature because you can monitor 10Gb links with 1Gb tools.

Netvisor One load balances the traffic by distributing the traffic load to different tool ports or appliances in order to scale the monitoring. This also provides redundancy in the monitoring technology.

When a member port goes down, traffic on the port is switched to remaining member ports and evenly distributed.

To configure load balancing, use the following steps:

1) First configure a trunk on the desired ports. In this case, ports 15 and 16 are configured as a trunk:

```
CLI (network-admin@Leaf1) > trunk-create name lb_trunk ports 
15,16

Created trunk lb_trunk, id 128
```

2) Create the port association on the switch:

```
CLI (network-admin@Leaf1) > port-association-create name pa1 
master-ports 1 slave-ports 128 virtual-wire bidir
```

3) Display the configuration:

```
CLI (network-admin@Leaf1) > port-association-show

switch  name master-ports slave-ports policy      virtual-wire bidir
------- ---- ------------ ----------- ----------- ------------ -----
leaf1   pa1    1            128       all-masters   true         true

CLI (network-admin@Leaf1) > port-show port 1,16

switch  port vnet hostname status config    trunk
------- ---- ---- -------- ------ --------- --------
leaf1    1                         40g,jumbo
leaf1    16                 trunk  10g,jumbo lb_trunk

CLI (network-admin@Leaf1) > vflow-show layout vertical

name:                   Internal-Keepalive
scope:                  local
in-port:
ether-type:             ipv4
dst-ip:                 239.4.9.7
proto:                  udp
```

```
flow-class:              control
precedence:              14
action:                  to-cpu
action-to-ports-value:
enable:                  enable
table-name:              L1-Virtual-Wire-1-0

name:                    VIRT_WIRE_MAS_SLV
scope:                   local
in-port:                 1
ether-type:
dst-ip:
proto:
flow-class:
precedence:              14
action:                  to-port
action-to-ports-value:   128
enable:                  enable
table-name:              L1-Virtual-Wire-1-0
```

Configuring Topologies and Topology Links

The VirtualWire fabric enables you to segment the same fabric into multiple independent and isolated topologies. The same switch can be part of multiple topologies, with different ports configured for different topologies.

> **Note**: A single port can be part of only one topology at any point in time.

After creating a topology, you must add the physical links of the VirtualWire fabric. You need to configure the topology links only once.

The topology-* and topology-link* commands sets up a fabric-wide XML file with all details about the network topology, which you, as a network administrator can use for path computation. The port-association-* command enables the path computation of two VirtualWire switches in the topology (see *Configuring Fabric-wide Port Associations* section later). When a path is found in the topology during port association, VirtualWire reserves the identified path and local port associations are created along the path for traffic redirection. Henceforth, those reserved topology links are not considered for further path calculations.

The Figure 6-4 shows three different topologies (1,2, and 3 - color coded to differentiate) configured within the same fabric and Figure 6-5 displays the topology links between two topologies in the VirtualWire fabric.

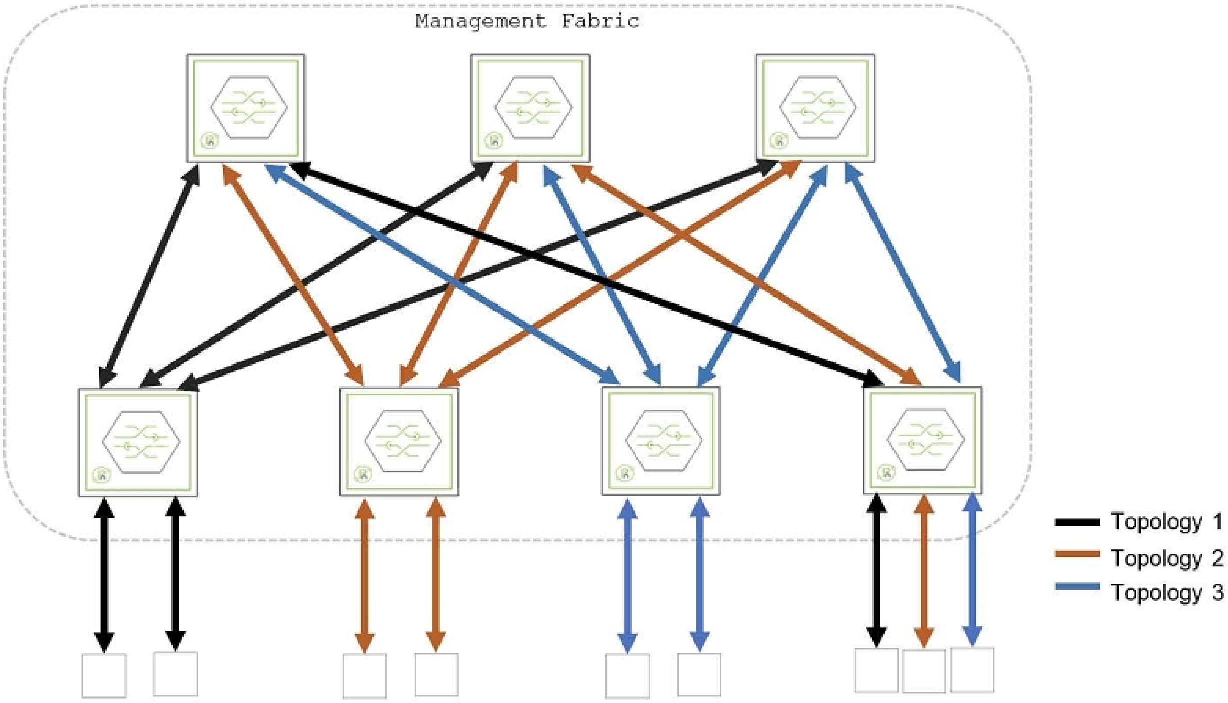


Figure 1-4: VirtualWire Fabric with Multiple Topologies for Automatic Path Creation

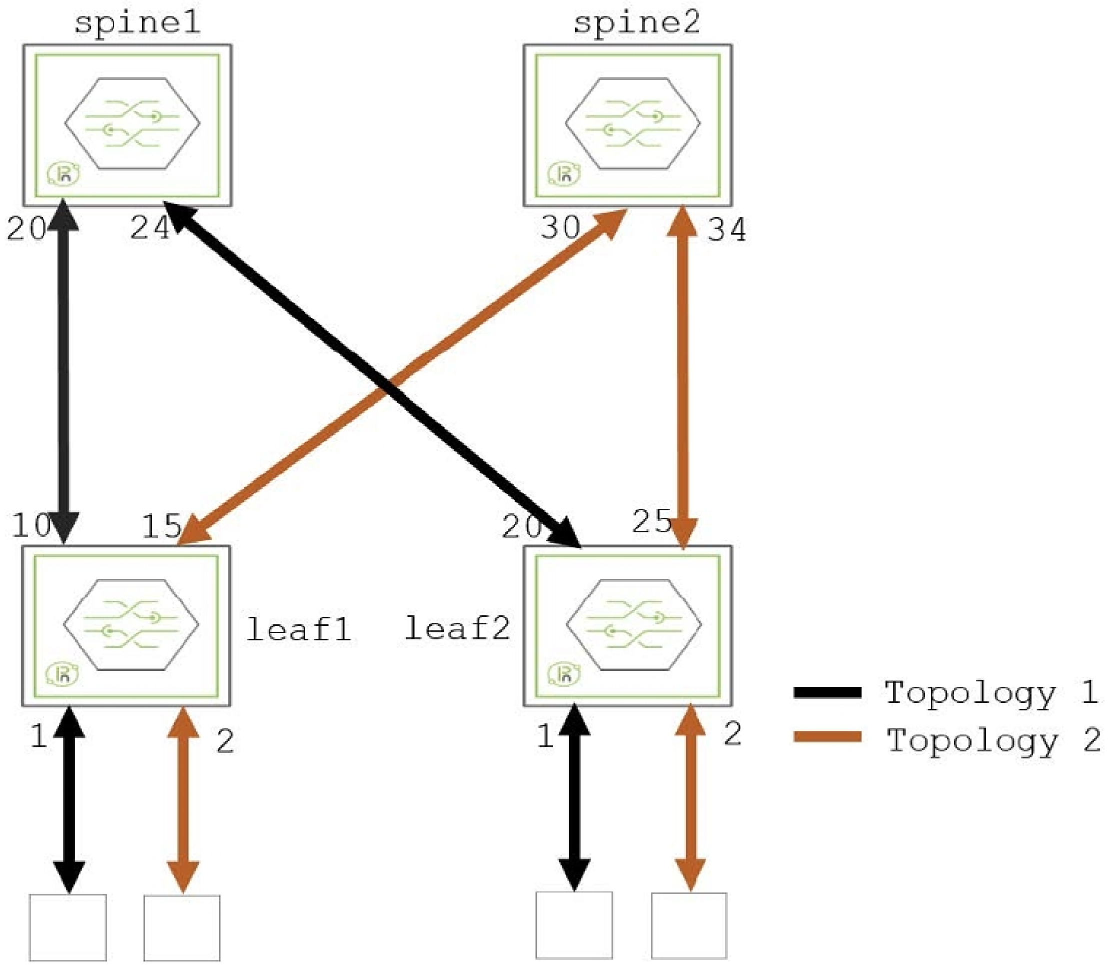


Figure 1-5: VirtualWire Fabric Linking Two Topologies

Use the commands to configure topology and topology links in VirtualWire fabric:

To create a network topology, use the command:

```
CLI (network-admin@netvisor) > topology-create name name-
string
```

`topology-create`	Use this command to create a network topology.
`name name-string`	Specify the name for the fabric topology.

To view the existing network topologies, use the command:

```
CLI (network-admin@netvisor) > topology-show name name-string
```

To delete an existing network topology, use the command:

```
CLI (network-admin@netvisor) > topology-delete name name-
string
```

To configure a link between two topologies in the VirtualWire fabric, use the command:

```
CLI (network-admin@netvisor) > topology-link-add name name-
string node1 fabric-node name node1-port node1-port-number
node2 fabric-node name node2-port node2-port-number enable|
disable
```

`name` *`name-string`*	Specify the name for the fabric topology.
`Specify the following link arguments:`	
`node1 fabric-node name`	Specify the name for link node 1
`node1-port node1-port-number`	Specify the port on node 1
`node2 fabric-node name`	Specify the name for link node 2
`node2-port node2-port-number`	Specify the port on node 2
`enable|disable`	Specify the topology link state for path calculation

To modify the link to the network topology, use the command:

```
CLI (network-admin@netvisor) > topology-link-modify name name-
string node1 fabric-node name node1-port node1-port-number
node2 fabric-node name node2-port node2-port-number enable|
disable
```

To remove the link to the network topology, use the command:

```
CLI (network-admin@netvisor) > topology-link-remove name name-
string node1 fabric-node name node1-port node1-port-number
node2 fabric-node name node2-port node2-port-number
```

To view the link details to the network topology, use the command:

```
CLI (network-admin@netvisor) > topology-link-show name name-
string
```

The following details are displayed:

`node1 fabric-node name`	Displays the name for link node 1
`node1-port node1-port-number`	Displays the port on node 1
`node2 fabric-node name`	Displays the name for link node 2
`node2-port node2-port-number`	Displays the port on node 2
`in-use yes|no`	Displays whether the topology link is in use or not
`in-path in-path-string`	Displays the topology link used by this path
`enable|disable`	Displays the topology link state for path calculation
`id id-number`	Displays the Link identifier

Configuring Fabric-wide Port Associations

VirtualWire fabric enables you to automate the path discovery process across multiple VirtualWire switches than having you to manually configure the individual port associations, which is a complex, error-prone and time-consuming process.

To configure an automated end-to-end port association on all the VirtualWire switches in a fabric, you should specify the fabric topology first (see *Configuring Topologies and Topology Links* section) and then configure the port association path. VirtualWire fabric validates and computes the path configuration thereafter.

You can configure the port associations using the CLI commands and through RESTful API to UNUM.

You can provision automatic path configuration only on a fabric that is configured with local scope. The path computation is done locally on the switch and fabric commands executed and then the hop-by-hop port associations are configured automatically and sent to respective switches.

To create a port association path, use the command:

```
CLI (network-admin@netvisor) > port-association-path-create
```

`name` *`name-string`*	Specify the name of the path
`topology topology name`	Specify the fabric topology name that was created in step 1
`node1 fabric-node name`	Specify the name for link node 1
`node1-port node1-port-number`	Specify the port on node 1
`node2 fabric-node name`	Specify the name for link node 2
`node2-port node2-port-number`	Specify the port on node 2

To delete an existing port association path, use the command:

```
CLI (network-admin@netvisor) > port-association-path-delete
name name-string
```

To view the port association path, use the command:

```
CLI (network-admin@netvisor) > port-association-path-show
```

`port-association-path-show`	Displays the port association paths
`name` *`name-string`*	Displays the path name
`topology` *`topology name`*	Displays the fabric topology name
`node1 fabric-node name`	Displays the name for link node 1
`node1-port node1-port-number`	Displays the port on node 1
`node2 fabric-node name`	Displays the name for link node 2

`node2-port node2-port-number`	Displays the port on node 2
`in-use yes|no`	Displays whether the topology link is in use or not
`in-path in-path-string`	Displays the topology link used by this path
`status down|up`	Displays the path status
`path` *`path-string`*	Displays the path string

Below is an example of a sample configuration:

Create a network topology, *VWtopo* and add topology link between node 1: *pn-vw-5,* port *125* and node2: *pn-lab-4,* port *49.* Also create another link between *pn-lab-4, port 5* and *pn-colo-1, port 5*:

```
CLI (network-admin@pn-lab-4) > topology-create name VWtopo

CLI (network-admin@pn-lab-4) > topology-link-add name VWtopo
node1 pn-vw-5 node1-port 125 node2 pn-lab-4 node2-port 49

CLI (network-admin@pn-lab-4) > topology-link-add name VWtopo
node1 pn-lab-4 node1-port 5 node2 pn-colo-1 node2-port 5
```

To view the details, use the command:

```
CLI (network-admin@tucana-colo-4) > topology-show name VWtopo

switch    name   node1      node1-port    node2       node2-port in-use in-
path enable
-------- ------ ----------  ---------- ------------ ---------- ------ -------
------
pn-lab-4  VWtopo pn-lab-4     5          pn-colo-1      5          no
    yes
pn-lab-4  VWtopo pn-vw-5      125        pn-lab-4       49         no
    yes
pn-vw-5   VWtopo pn-lab-4     5          pn-colo-1      5          no
    yes
pn-vw-5   VWtopo pn-vw-5      125        pn-lab-4       49         no
    yes
pn-colo-1 VWtopo pn-lab-4     5          pn-colo-1      5          no
    yes
pn-colo-1 VWtopo pn-vw-5      125        pn-lab-4       49         no
    yes
```

To create a port association path, use the command:

```
CLI (network-admin@pn-lab-4) > port-association-path-create
name new topology VWTOPO node1 pn-vw-5 node1-port 2 node2 pn-
colo-1 node2-port 125

Created path: pn-vw-5(2) <-> pn-vw-5(49) <-> pn-lab-4(125) <->
pn-lab-4(5) <-> pn-colo-1(5) <-> pn-colo-1(125)
```

To view the topology link details, use the command:

```
CLI (network-admin@pn-vw-5*) > topology-link-show

name   node1     node1-port   node2     node2-port in-use in-path enable
------ -------- -----------  ---------  ---------- ------ ------- ------
VWTOPO pn-vw-5      49        pn-lab-4    125        yes    new     yes
VWTOPO pn-lab-4     5         pn-colo-1   5          yes    new     yes
```

To view the port association details, use the command:

```
CLI (network-admin@pn-vw-5*) > port-association-path-show

name topology node1    node1-port node2      node2-port status
---- -------- ------   ---------- --------- ---------- ------
new  VWTOPO   pn-vw-5    2        pn-colo-1    125        up
```

Configuring Traffic Filtering Using vFlows in VirtualWire Mode

A switch in a VirtualWire fabric is capable of filtering traffic at wire speed. You can configure traffic filtering in cases such as, when multiple streams of traffic arrives into a single port and if each flow needs to be redirected to different egress ports. A vFlow classifies traffic based on various factors such as the ingress port, source-mac, destination-mac, source-ip, destination-ip, vlan, egress-port, ether-type, protocol, and so on.

All the vFlows created in VirtualWire mode must be configured under the *L1-Virtual-Wire-1-0* table.

For more details on vFlows, see the *Netvisor ONE Configuration Guide* on Pluribus Networks website.

In Figure 6-X, a VirtualWire switch is used to share a traffic generator across two DUTs. In this topology, two traffic flows come in from the traffic generator towards the VirtualWire switch on port 3 on two different subnets. Use the VirtualWire switch to filter the incoming streams based
on the source IP addresses and redirect them toward the required destination.

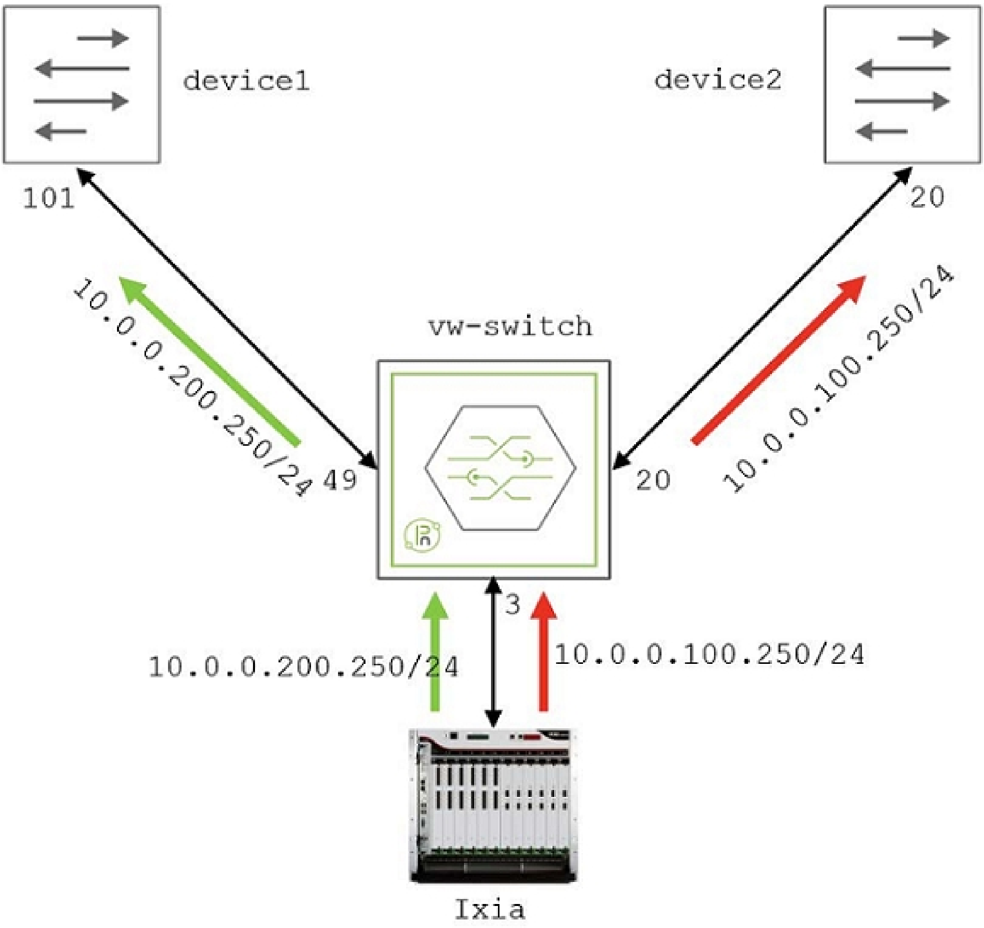


Figure 1-6: VirtualWire with vFlows for Traffic Filtering

To configure traffic filtering on the VirtualWire switch, use the following commands:

1. Configure a multi-port association with any-master policy by using the command:

```
CLI (network-admin@vw-switch) > port-association-create name
name-string master-ports port-list slave-ports port-list
virtual-wire bidir policy any-master
```

`port-association-create`	Creates a port association between different ports.
`name` *`name-string`*	Specify the name for the port association.
`master-ports` *`port-list`*	Specify the master ports.
`slave-ports` *`port-list`*	Specify the slave ports.
`virtual-wire|no-virtual-wire`	Specify the `virtual-wire` keyword for the associated ports to form a VirtuialWire.
`bidir|no-bidir`	Specify `bidir` keyword to establish a bi-directional port state tracking.
`policy all-masters|any-master`	Specify the port association policy, the default policy is `all-masters`.

Below is an example configuration named filer-traffic by specifying the master ports, 20, 49 and slave ports as 3; with any-master policy:

```
CLI (network-admin@vw-switch) > port-association-create name
filer-traffic master-ports 20,49 slave-ports 3 virtual-wire
bidir policy any-master
```

2. Create two vFlows on the VirtualWire switch to filter traffic based on source IP address:

```
CLI (network-admin@vw-switch) > vflow-create name name-string
scope local|fabric src-ip ip-address in-port port-list action
toport action-to-ports-value port-list table vflow-table-name
precedence 15
```

`vflow-create`	Creates a virtual flow definition.
`name` *`name-string`*	Specify the name for the vFlow.
`scope local|fabric`	Specify the scope for the vFlow configuration.
`src-ip ip-address`	Specify the source IP address for the vFlow.

`in-port`	Specify the incoming port for the vFlow.
`action`	Specify the forwarding action to apply to the vFlow.
`action-to-ports-value` *`port-list`*	Specify the port value for the specified action.
`table` *`vflow-table-name`*	Specify the table name as *L1-Virtual-Wire-1-0 table*.
`precedence`	Specify the traffic priority value between 2 and 15.

For example, below is an example configuration for two vflows: *filterstream1*, and *filterstream2*:

```
CLI (network-admin@vw-switch) > vflow-create name
filterstream1 scope local src-ip 10.0.100.250 in-port 3 action
toport action-to-ports-value 20 table L1-Virtual-Wire-1-0
precedence 15

CLI (network-admin@vw-switch) > vflow-create name
filterstream2 scope local src-ip 10.0.200.250 in-port 3 action
toport action-to-ports-value 49 table L1-Virtual-Wire-1-0
precedence 15
```

Use the show command to view your configuration:

```
CLI (network-admin@vw-switch) > vflow-show name name-string
```

Building a VirtualWire™ Fabric

Multiple VirtualWire switches can be interconnected to form a single VirtualWire fabric. A VirtualWire fabric is like a highly scalable and distributed patch panel that can be dynamically and remotely provisioned to implement single dedicated wire speed links between any two device ports in the network.

When all of the switches in the VirtualWire fabric are part of the same Management Fabric, they can be provisioned and controlled as a single logical VirtualWire switch.

The most efficient design for a VirtualWire fabric is based on the classic leaf-spine architecture, or Clos, a non-blocking, multistage switching topology, as in the figure below.

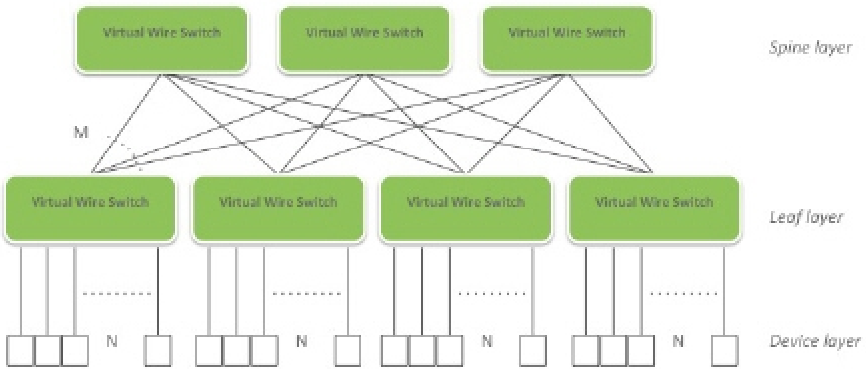


Figure 1-7 - Leaf and Spine Topology for Virtual Wire Fabric

> **Note:** In CLOS architecture, there is no limit to the number of VirtualWire links between device ports that are physically connected to the same leaf. Instead, the number of VirtualWire links between device ports that are connected to different leafs depend on the over-subscription ratio between leaf and spine.

With this approach, you can select the desired over-subscription ratio and build a modular and scalable architecture to scale up to thousands of device ports.

For example, using the Dell or Freedom series switches as building blocks, a possible leaf switch configuration uses 48 X 10 Gigabit Ethernet ports to connect to device ports and 6 x 40 Gigabit Ethernet ports to connect to the spine layer, resulting in a 1.8:1 over-subscription ratio.

Based on the desired maximum number of device ports, you can select from different scale options:

17 leaf 6 spine at 1.8:1 over-subscription ratio for a total of 748 device 10 Gbps/1Gbps ports

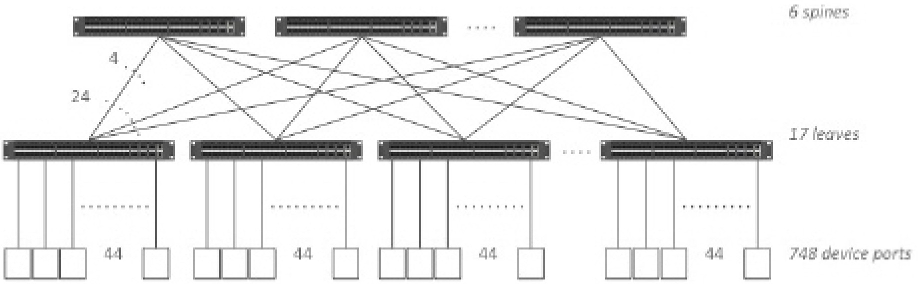


Figure 1-8: 17 Leafs and 6 Spines

34 leaf 12 spine at 1.8:1 over-subscription ratio for a total of 1496 device 10Gbps/1Gbps ports

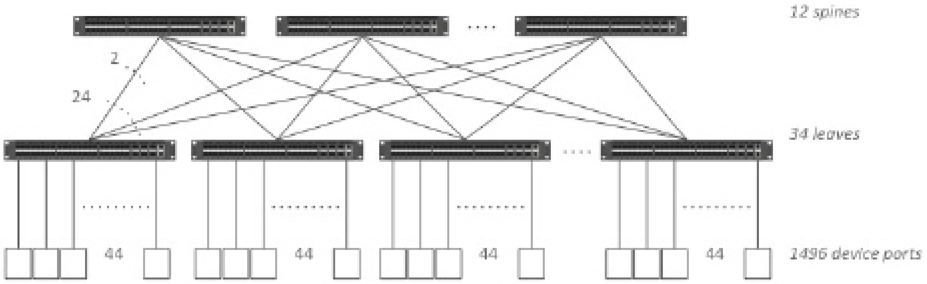


Figure 1-9: 34 Leafs and 12 Spines

68 leaf 24 spine at 1.8:1 over-subscription ratio for a total of 2992 device 10Gbps/1Gbps ports

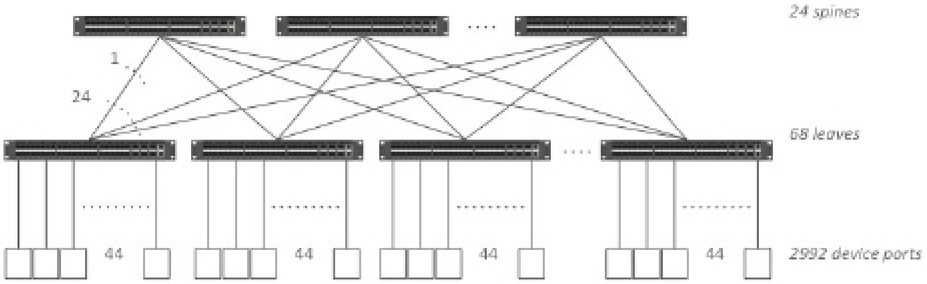


Figure 1-10 - 68 Leafs and 24 Spines

Configuring the Fabric Over the Management Interface

Fabric configuration information can be exchanged over the network through in-band communication.

However, occasional disruption to in-band traffic occurs due to factors such as network re-convergence, port flapping, power system transients, and other events.

Therefore, an alternative method is to configure fabric communication over the management interface for a dedicated communication channel.

This can be achieved while creating the fabric, for example:

```
CLI (network-admin@switch) > fabric-create name MyFabric
fabric-network mgmt
```

When you create a fabric over the management interface, any other node joining the fabric inherits this setting. In other words, all nodes within the same fabric communicate through the same network type with fabric peers. You cannot have mixed fabric configurations using both management interfaces and in-band communication.

Therefore, Netvisor does not display fabrics over an incompatible networks when you execute the `fabric-join` command. This prevents a switch from joining an incompatible fabric.

When you configure the fabric communication over the management interface, all fabric communication stays on the management network, except the following types of packets:

- Cluster synchronization-related messages and cluster keep-alive packets sent over the in-band interface.
- The fabric advertisements such as fabric keep-alive packets and global-discovery packets are controlled by fabric-advertisement-network, which is configured while creating or modifying a fabric.

While fabric-related communication such as transactions, notifications, file system replication messages, and other communications can be configured to be sent over the management network, for consistency, it is recommended to use the same management network for other purposes or communication types such as network status updates and forwarded packets, collectively referred to as control network and fabric advertisements.

The `fabric-create` command allows you to select the transmission medium for other traffic types using specific parameters named `control-network` and `fabric-advertisement-network`:

```
CLI (network-admin@switch) > fabric-create name MyFabric
[fabric-network in-band|mgmt] [control-network in-band|mgmt]
[fabric-advertisement-network inband-mgmt|inband-only|mgmt-
only]
```

Displaying Fabric Nodes

Netvisor ONE uses fabric keepalive packets to determine the state of each fabric node. To display the state, use the fabric-node-show command with the syntax:

```
CLI network-admin@switch > fabric-node-show [state offline|
online|in-band-only-online|mgmt-only-online|fabric-joined|
eula-required|setup-required|fabric-required|fresh-install]
```

Netvisor ONE supports monitoring and reporting on both management and in-band network, therefore the node state can be one of the following:

- `online` — reach-ability of node over both management and in-band interfaces
- `In-band-only-online` — reach-ability of node through in-band channel only
- `mgmt-only-online` — reach-ability of node through management network only
- `offline` — no reach-ability over either communication channel.

In this example, Netvisor ONE displays the `online` node state in the command output:

```
CLI (network-admin@switch) > fabric-node-show layout vertical

id:                       167772208
name:                     switch
fab-name:                 MyFabric
fab-id:                   a000030:5537b46c
cluster-id:               a000030:1
fab-mcast-ip:             ::
local-mac:                64:0e:94:28:00:8e
fabric-network:           in-band
mgmt-ip:                  10.9.100.100/16
mgmt-mac:                 64:0e:94:28:00:8f
mgmt-l3-port:             0
mgmt-secondary-macs:
in-band-ip:               192.168.42.10/24
in-band-mac:              64:0e:94:28:00:8e
in-band-l3-port:          0
in-band-secondary-macs:
fab-tid:                  8
cluster-tid:              1
out-port:                 0
version:                  5.0.0-5000014540
state:                    online
firmware-upgrade:         not-required
device-state:             ok
ports:                    0
```

Also check the `fab-tid` value for consistency on each node. See the *Troubleshooting the Fabric* section for details.

Displaying Fabric Information and Statistics

To display information on the configured fabrics, use the fabric-show command:

```
CLI (network-admin@switch) > fabric-show

 name      id                 vlan    fabric-network    control-network   tid
 -------   -----              -----   --------------    ---------------   ------

 Fabric1   a000030:5537b46c   3       in-band           in-band           365

 Fabric2   6000210:566621ee   100     mgmt              in-band           5055
```

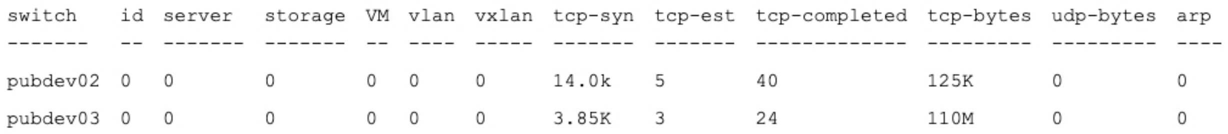
To display the information about the fabric instance of the local switch, use the `fabric-info` command:

```
CLI (network-admin@switch) > fabric-info format all layout
vertical

name:                         Fabric1
id:                           a000030:5537b46c
vlan:                         3
fabric-network:               in-band
control-network:              in-band
tid:                          365
fabric-advertisement-network: inband-only
```

To display fabric statistics use the `fabric-stats-show` command:

```
CLI (network-admin@switch) > fabric-stats-show
```



```
switch    id  server   storage  VM  vlan  vxlan  tcp-syn  tcp-est  tcp-completed  tcp-bytes  udp-bytes  arp
-------   --  -------  -------  --  ----  -----  -------  -------  -------------  ---------  ---------  ----

pubdev02  0   0        0        0   0     0      14.0k    5        40             125K       0          0

pubdev03  0   0        0        0   0     0      3.85K    3        24             110M       0          0
```

To display fabric statistics in vertical format, use the following command:

```
CLI (network-admin@switch) > fabric-stats-show format all
layout vertical
switch:        sw45
id:            0
servers:       0
storage:       0
VM:            0
vlan:          0
vxlan:         0
tcp-syn:       0
tcp-est:       0
tcp-completed: 0
tcp-bytes:     0
```

```
udp-bytes:     0
arp:           0
```

Example: Configuring a Swich for VirtualWire™ Mode

The configuration example in this section refers to a VirtualWire fabric composed by one spine and two leaf switches as in the figure below.

Two devices, **device-A** and **device-B**, have respectively two ports and one port that are physically connected to the VirtualWire switch Leaf-1. A third device, **device-C**, is physically connected to the VirtualWire switch Leaf-2.

The desired logical setup consists in a bidirectional service chain topology where device-A is inserted in-line between device-B and device-C.

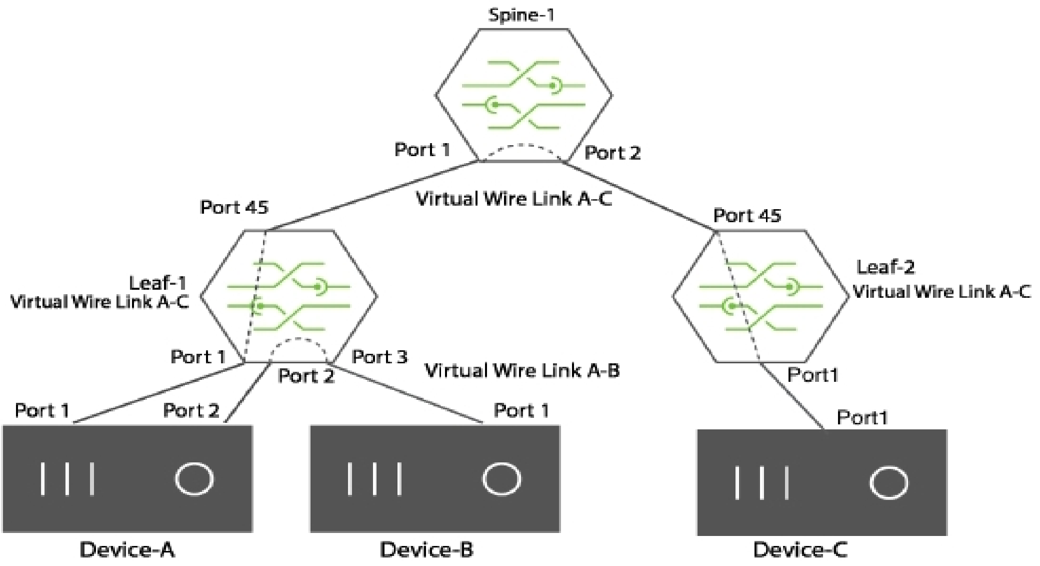


Figure 1-11 - Bidirectional Traffic over a VirtualWire Connection

To create a bidirectional virtual link from **device-A** to **device-C**, use these steps:

1) Configure a port association for **device-A** to **device-C** using port 1 and port 45 on Leaf-1.

```
CLI (network-admin@Leaf-1) > port-association-create name
link-AC virtual-wire bidir master-ports 1 slave-ports 45
```

2) Configure a port association on Spine-1 between ports 1 and 2:

```
CLI (network-admin@Spine-1) > port-association-create name
link-AC virtual-wire bidir master-ports 1 slave-ports 2
```

3) Configure a port association on Leaf-2 between ports 45 and 1:

```
CLI (network-admin@Leaf-2) > port-association-create name
link-AC virtual-wire bidir master-ports 45 slave-ports 1
```

Example: Configuring a Switch for Unidirectional VirtualWire™ Mode

Unidirectional VirtualWire links can be used for testing link fault signaling features like Cisco Unidirectional Link Detection (UDLD).

In the example below, the traffic directions are separated and individually controlled by creating unidirectional VirtualWire links in a Virtual Wire fabric composed by one spine and two leaf switches.

In the resulting logical topology, **device-B** and **device-C** are directly interconnected in one direction; in the opposite direction **device-A**, a traffic impairment tool, is inserted in-line. **Device-A** can be used to introduce errors on the wire or to emulate unidirectional fiber cut events.

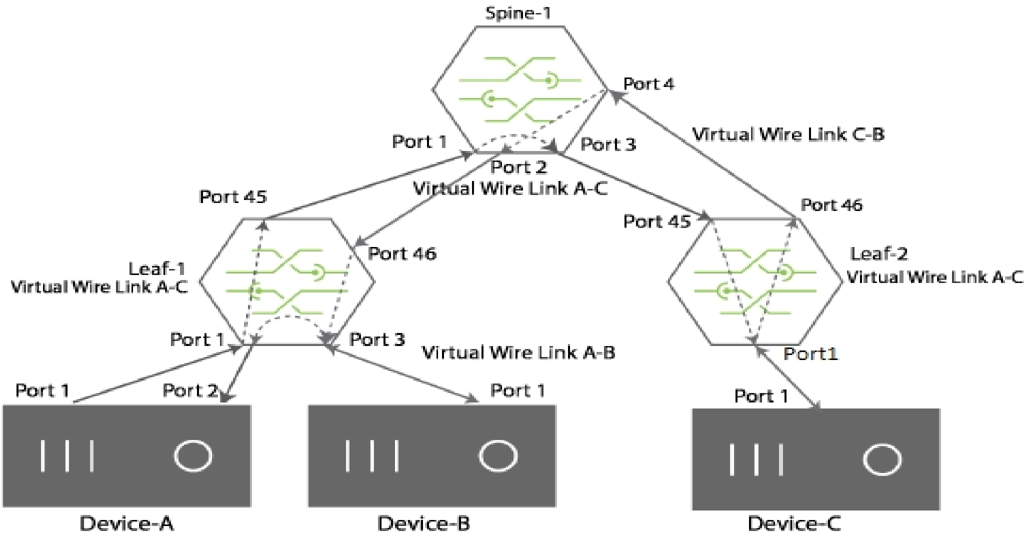


Figure 1-12 - Unidirectional Traffic over a Virtual Wire Connection

To configure the VirtualWire switch for unidirectional traffic, use the following steps:

1) Configure a port association on Leaf-1, ports 1 and 45.

```
CLI (network-admin@Leaf-1) > port-association-create name
link-AC virtual-wire master-ports 1 slave-ports 45
```

2) Configure a port association on Spine-1, ports 1 and 3:

```
CLI (network-admin@Spine-1) > port-association-create name
link-AC virtual-wire master-ports 1 slave-ports 3
```

3) Configure a port association on Leaf-2, ports 45 and 1:

```
CLI (network-admin@Leaf-2) > port-association-create name
link-AC virtual-wire master-ports 45 slave-ports 1
```

This configuration connects **device-A** to **device-C** over a unidirectional virtual wire link.

To connect **device-C** to device-B over a unidirectional virtual link, use the following steps:

1) Configure a port association on Leaf-1 for ports 3 and 46:

```
CLI (network-admin@Leaf-1) > port-association-create name
link-CB virtual-wire master-ports 3 slave-ports 46
```

2) Configure a port association on Spine-1 for ports 2 and 4:

```
CLI (network-admin@Spine-1) > port-association-create name
link-CB virtual-wire master-ports 2 slave-ports 4
```

3) Configure a port association on Leaf-2 for ports 46 and 1:

```
CLI (network-admin@Leaf-1) > port-association-create name
link-CB virtual-wire master-ports 46 slave-ports 1
```

This configuration connects over a unidirectional virtual wire link.

To configure a VirtualWire connection between **device-B** and **device-A**:

1) Configure a port association on Leaf-1 for ports 3 and 2:

```
CLI (network-admin@Leaf-1) > port-association-create name
link-BA virtual-wire master-ports 3 slave-ports 2
```

Configuring the Inline Services for VirtualWire™

The Inline Service feature manages service chains for Layer 1 VirtualWire switches. The term, Inline Services, refers to services attached to a Layer 1 VirtualWire switch such as Next-Generation Firewall (NGFW), Intrusion Detection System (IDS), Intrusion Prevention System (IPS), and Distributed Denial of Service attack (DDoS) Prevention.

When an Inline Service fails, a policy determines if traffic is allowed to bypass the Inline Services or if the traffic is blocked until the Inline Services recovers.

Security services such as NGFW, IDS, IPS, and DDoS are important for any network deployment. Inline Services provide continuous monitoring of the network for improved security. Inline security services can fail due to power failure, maintenance or other reasons. An Inline Service failure has the potential to affect the flow of traffic in the network, potentially bringing the network down. This requires continues monitoring of services on network for better security.

To safeguard against such failures, the Inline Service feature provides a way to steer traffic around the failed Inline Service so traffic is not impacted. During a failure, the network is not protected by the service provided by the Inline Service.
The Inline Service recover and failure is detected by the port link states, UP and DOWN, between the Layer 1 VirtualWire switch and the Inline Service.
However a device connected to the switch can fail without the port sending an UP or Down link state. In such cases, Netvisor One relies on a heartbeat, or a probe in a form of a pre-defined packet, sent to an attached device.

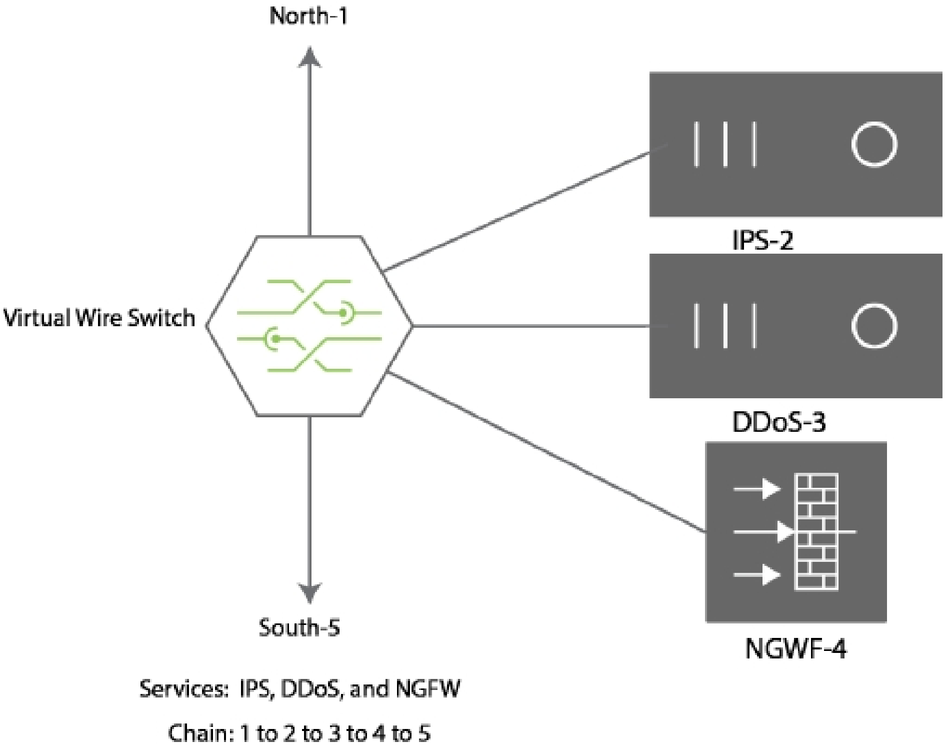


Figure 1-13 - Example of Inline Services

You configure the order of the Inline Services using the `port-association-service-*` commands.

If an inline service is configured with the parameter, `fail-open`, Netvisor One sends traffic and skips any Inline Services failing on the network.

For example, if you configure Inline Services with the chain 1->2->3->4->5, and the Inline Service 3 fails, the new chain is 1->2->4->5.

If an Inline Service is configured with the parameter, `fail-close`, and any Inline Service fails, network traffic is blocked.

For example, if you configure the chain 1->2->3->4->5, and any Inline Service such as 2, 3, or 4 fails, network traffic does not flow through the chain, and network traffic flow stops.

Configuring Heartbeat Service

Netvisor One generates a packet from the CPU to send to the receive port of an Inline Service and the Netvisor One vFlow configured for snooping is not port-specific, as Netvisor One accepts the response from either the receive port or the transmit port. You configure the heartbeat as an additional parameter for a specific Inline Service.

For example, to create a heartbeat detection service named FW-Probe, use the following syntax:

```
CLI (network-admin@Spine1) > service-heartbeat-create name
FW_probe interval 5s retry 3 vlan-id 10 src-mac
64:6e:11:1c:11:11 dst-mac 01:1b:11:01:01:01 type normal
payload 54 63 82 ff 01 46 12 ce a2 d4 00 00 00 00 00 00 00 00
```

In this example, you define the frequency of the heartbeats as well as the number of missed probes before Netvisor One detects the service with this heartbeat is down.

To add the Heartbeat Service to Inline Services, **FW-1** and **FW-2**, use the following syntax:

```
CLI (network-admin@Spine1) > inline-service-create name FW1
tx-port 11 rx-port 11 heartbeat FW_probe

CLI (network-admin@Spine1) > inline-service-create name FW2
tx-port 9 rx-port 10 heartbeat FW_probe
```

Netvisor One counts the missed heartbeats separately for **FW-1** and **FW-2**.

Configuring the Payload

Specify the payload as a packet including Ethertype of the packet, but excluding the CRC at the end. For example, an ARP packet uses this format:

```
Payload(including CRC):
```

```
0:  ffff ffff ffff 0011 0100 0001 0806 0001
   ................
16: 0800 0604 0001 0011 0100 0001 0101 0101
   ................
32: 0000 0000 0000 0101 0102 0000 0000 0000
   ................
48: 0000 0000 0000 0000 0000 0000 2160 cc6b
   ............!`.k
```

A heartbeat service, HB_4 for this ARP packet has the following syntax:

```
CLI (network-admin@Spine1) > service-heartbeat-create name
HB4_arp interval 1s retry 10 vlan 1 src-mac 00:11:01:00:00:01
dst-mac ff:ff:ff:ff:ff:ff payload "0806 0001 0800 0604 0001
0011 0100 0001 0101 0101 0000 0000 0000 0101 0102 0000 0000
0000 0000 0000 0000 0000"
```

When you create the Heartbeat Service, Netvisor One installs a specific vFlow in the vFlow table.

Netvisor One verifies the functionality of the Inline Service using two methods: 1) a normal heartbeat, and 2) a pass-through heartbeat. When you configure the parameter, type, you specify the type of heartbeat for the service as `normal`, a request-response heartbeat indicating the service responds to the heartbeat. If you specify `pass-through` as the heartbeat, Netvisor One sends the packet and returns it the switch through the service.

Configuring Inline Services with a Heartbeat Service

To configure the example topology displayed in Figure 1 - Example of Inline Services - use the following steps:

1) Configure the North-South port association, use the following syntax:

```
CLI (network-admin@Spine1) > port-association-create name
NorthToSouth master-ports 1 slave-ports 8 virtual-wire no-
bidir
```

2) Define and configure the Heartbeat Service parameters:

```
CLI (network-admin@Spine1) > service-heartbeat-create name
FW_probe interval 5s retry 3 vlan-id 10 src-mac
64:6e:11:1c:11:11 dst-mac 01:1b:11:01:01:01 type pass-through
payload 54 63 82 ff 01 46 12 ce a2 d4 00 00 00 00 00 00 00 00
```

3) Configure the Inline Services chain:

```
CLI (network-admin@Spine1) > port-association-service-add
port-association-name NorthToSouth inline-service IPS order 2
policy-action fail-open
```

```
CLI (network-admin@Spine1) > port-association-service-add
port-association-name NorthToSouth inline-service DDoS order 3
```

```
policy-action fail-open

CLI (network-admin@Spine1) > port-association-service-add
port-association-name NorthToSouth inline-service NGWF order 4
policy-action fail-closed
```

Netvisor One uses new commands to configure Heartbeat Services:

```
CLI (network-admin@Spine1) > service-heartbeat-create
```

name *name-string*	Specify a name for the Heartbeat Service.
interval duration: #d#h#m#s	Specify the interval between heartbeat packets.
retry *retry-number*	Specify the number of times to retry sending a packet.
vlan *vlan-id*5	Specify a VLAN ID.
src-mac *mac-address*	Specify the source port MAC address.
dst-mac *mac-address*	Specify the destination MAC address.
type normal|pass-through	Specify the type of heartbeat response as normal or pass-through. A normal response indicates that the Inline Service sends the response. A pass-through response indicates that Netvisor One sends the response and returns it to the Inline Service.
payload *payload-string*	Specify the payload for the heartbeat packet.

```
CLI (network-admin@Spine1) > service-heartbeat-delete
```

name *name-string*	Specify a name for the Heartbeat Service.

```
CLI (network-admin@Spine1) > service-heartbeat-modify
```

name *name-string*	Specify a name for the Heartbeat Service.
interval duration: #d#h#m#s	Specify the interval between heartbeat packets.
retry *retry-number*	Specify the number of times to retry sending a packet.

```
CLI (network-admin@Spine1) > service-heartbeat-show
```

name *name-string*	Displays the name for the Heartbeat Service.

`interval duration: #d#h#m#s`	Displays the interval between heartbeat packets.
`retry` *`retry-number`*	Displays the number of times to retry sending a packet.
`vlan` *`vlan-id`*`5`	Displays a VLAN ID.
`src-mac` *`mac-address`*	Displays the source port MAC address.
`dst-mac` *`mac-address`*	Displays the destination MAC address.
`type normal|pass-through`	Displays the type of heartbeat response as normal or pass-through. A normal response indicates that the Inline Service sends the response. A pass-through response indicates that Netvisor One sends the response and returns it to the Inline Service.
`payload` *`payload-string`*	Displays the payload for the heartbeat packet.

Configuring Service Chains

A service chain is configured using `port-association-service-*` commands. The services in the chain are managed using `inline-service-*` commands.

Inline Services are configured using the following commands:

```
CLI (network-admin@Spine1) > port-association-service-add
```

`port-association-name` *`name-string`*	Specify the name of the port association to apply the service.
`switch` *`name-string`*	Specify the switch name where the service is located.
`inline-service` *`inline-service-name`*	Specify the name of the Inline Service.
`order` *`number`*	Specify a number to designate the order of the service. This is a value between 1 and 65535
`policy-action fail-open|fail-closed`	Specify a policy action when the service fails on the network.

```
CLI (network-admin@Spine1) > port-association-service-modify
```

`port-association-name` *`name-string`*	Specify the name of the port association to apply the service.
`switch` *`name-string`*	Specify the switch name where the service is located.
`inline-service` *`inline-`*	Specify the name of the Inline Service.

service-name	
order *number*	Specify a number to designate the order of the service. This is a value between 1 and 65535
policy-action fail-open|fail-closed	Specify a policy action when the service fails on the network.

```
CLI network-admin@Spine1) > port-association-service-remove
```

port-association-name *name-string*	Specify the name of the port association to apply the service.
switch *name-string*	Specify the switch name where the service is located.
inline-service *inline-service-name*	Specify the name of the Inline Service.

```
CLI (network-admin@Spine1) > port-association-service-show
```

port-association-name *name-string*	Displays the name of the port association to apply the service.
switch *name-string*	Displays the switch name where the service is located.
inline-service *inline-service-name*	Displays the name of the Inline Service.
order *number*	Displays a number to designate the order of the service. This is a value between 1 and 65535
policy-action fail-open|fail-closed	Displays a policy action when the service fails on the network.

```
CLI (network-admin@Spine1) > inline-service-create
```

name *name-string*	Specify a name for the Inline Service.
tx-port *port-list*	Specify the transmit port for the Inline Service.
rx-port *port-list*	Specify the receive port for the Inline Service.

```
CLI (network-admin@Spine1) > inline-service-delete
```

name *name-string*	Specify a name for the Inline Service.

```
CLI (network-admin@Spine1) > inline-service-show
```

`name` *`name-string`*	Specify a name for the Inline Service.
`tx-port` *`port-list`*	Specify the transmit port for the Inline Service.
`rx-port` *`port-list`*	Specify the receive port for the Inline Service.

Configuring and Displaying Statistics

You can display standard statistics that consist of flow-based information collected and tracked continuously by the switch. To modify statistics logging, use the `stats-log-modify` command and disable or enable statistical logging as well as change the interval, in seconds, between statistical events.

To show connection-level statistics, traffic flows between a pair of hosts for an application service, including current connections and all connections since the creation of the fabric, enter the following CLI command at the prompt:

```
CLI (network-admin@Leaf1) > connection-stats-show

switch:         pubdev02
count:          0
mac:            64:0e:94:28:00:8e
vlan:           3
ip:             192.168.42.10
port:           25
iconns:         6
oconns:         0
ibytes:         224K
obytes:         10.5K
total-bytes:    235K
first-seen:     02-26,17:19:52
last-seen:      02-26,17:19:57
last-seen-ago:  17d14h6m5s
switch:         pubdev02
count:          0
mac:            64:0e:94:28:03:56
vlan:           3
ip:             192.168.42.30
port:           128
iconns:         0
oconns:         3946878
ibytes:         4.50M
obytes:         13.5M
total-bytes:    18.0M
first-seen:     01-06,09:23:07
last-seen:      08:25:20
last-seen-ago:  42s
```

REST API Command: `GET http://<switch-ip>/vRest/connection-stats`

From the information displayed in the output, you can see statistics for each switch, VLANs, client and server IP addresses, as well as the services on each connection. Latency and other information is also displayed.

The latency (us) column displays the running latency measurement for the TCP connection in microseconds. It indicates end-to-end Round-Trip-Time (RTT) between TCP/IP session client and server and includes the protocol stack processing for the connected hosts and all intermediary network hops.

To display connection latency, use the `connection-latency-show` command:

```
CLI (network-admin@Leaf1) > connection-latency-show

switch   min    max    num-conns percent avg-dur obytes ibytes
total-bytes
-------- ------ ------ --------- ------- ------- ------ ------
-----------
switch-v 0.00ns 20.0us 67.5K     76%     17.12m  32.9K  18.0K
 51.0K
switch-v 20.0us 40.0us 2.74K     3%      1.64h   8.77M  123M
  132M
switch-v 40.0us 60.0us 10.4K     11%     1.40h   22.0M  403M
  425M
switch-v 60.0us 80.0us 1.85K     2%      1.10h   8.16M  127M
  135M
switch-v 80.0us 100us  901       1%      1.02h   3.39M  53.5M
 56.9M
switch-v 100us  120us  1.35K     1%      1.23h   5.49M  126M
  132M
switch-v 120us  140us  801       0%      1.06h   5.67M  39.2M
 44.9M
switch-v 140us  160us  545       0%      1.19h   1.88M  29.4M
 31.3M
switch-v 160us  180us  1.08K     1%      1.21h   5.04M  82.8M
 87.8M
switch-v 180us  200us  583       0%      56.77m  5.15M  72.7M
 77.8M
switch-v 200us         729       0%      48.51m  2.57G  184M
  2.75G
```

REST API Command : `GET http://<switch-ip>/vRes/connection-latencies`

Adding UNUM Insight Analytics Flow for Network Visibility

UNUM Insight Analytics Flow is an application developed by Pluribus Network that enables the network administrator to extract the analytical value from the telemetry data reported by the network switches powered by Pluribus Networks Netvisor® network operating system.

UNUM connects the switches as part of the Netvisor® fabric to gain visibility into the network, and extract all the telemetry data made visible by the Pluribus Network Operating System.

Once data is collected, UNUM Insight Analytics Flow relies upon a modern search engine database infrastructure to store, aggregate, filter, correlate and visualize vast amounts of data in real-time as well as with a powerful “time machine” functionality.

As shown in the figure below UNUM Insight Analytics Flow can be deployed in a Virtual Wire fabric topology, by connecting the UNUM Insight Analytics Flow server(s) to the fabric management network. Using Netvisor® API, UNUM Insight Analytics Flow establishes a connection to any Virtual Wire switch in fabric to gain visibility into the entire fabric network and extract all the device layer telemetry data made visible by the distributed Pluribus Network Operating System.

Once data is collected, UNUM Insight Analytics Flow relies upon a modern search engine database infrastructure to store, aggregate, filter, correlate and visualize vast amount of data in real time.

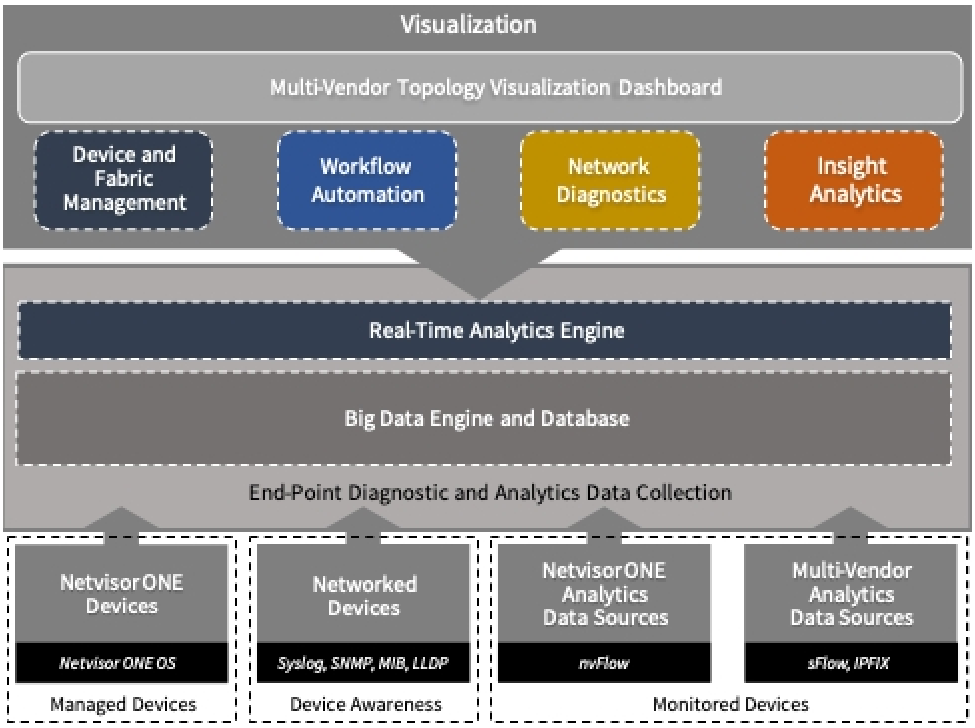

Visualization
Multi-Vendor Topology Visualization Dashboard
Device and Fabric Management
Workflow Automation
Network Diagnostics
Insight Analytics
Real-Time Analytics Engine
Big Data Engine and Database
End-Point Diagnostic and Analytics Data Collection
Netvisor ONE Devices
Netvisor ONE OS
Managed Devices
Networked Devices
Syslog, SNMP, MIB, LLDP
Device Awareness
Netvisor ONE Analytics Data Sources
nvFlow
Multi-Vendor Analytics Data Sources
sFlow, IPFIX
Monitored Devices

Figure 1-14 - UNUM Insight Analytics Flow

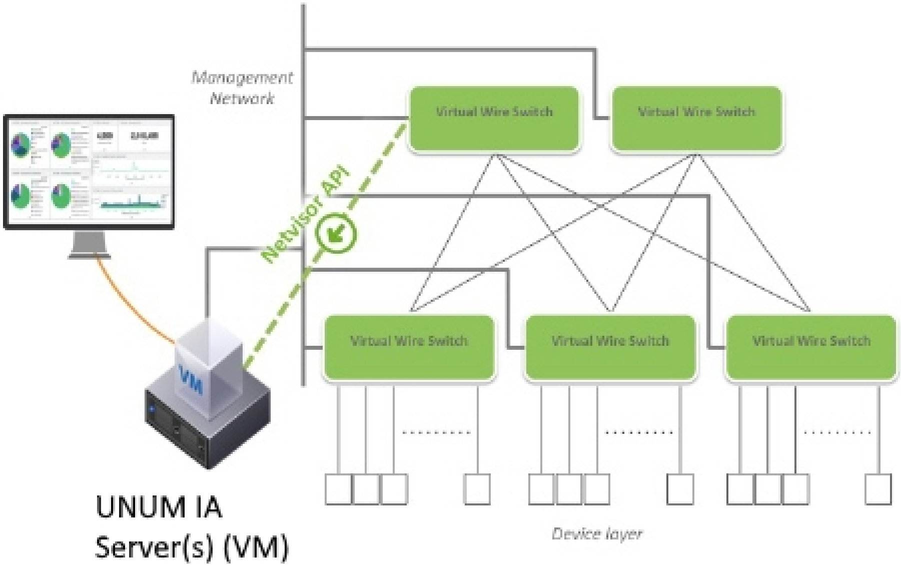


Figure 1-15 - Overview of UNUM Insight Analytics Flow Topology

To add **UNUM Insight Analytics Flow** to your network, please see the Pluribus UNUM Management Platform

Additional Information for Pluribus VirtualWire

- Installing Netvisor One and Initial Configuration

Installing Netvisor One and Initial Configuration

This section contains information about initial configuration of your switch as well as commands to manage, upgrade, and restore Netvisor One configurations.

Changes to the End User License Agreement (EULA)

Currently, the Netvisor One EULA is displayed when the switch is setup.

```
Netvisor OS Command Line Interface 5.1.0
By ANSWERING "YES" TO THIS PROMPT YOU ACKNOWLEDGE THAT YOU
HAVE READ THE TERMS OF THE PLURIBUS NETWORKS END USER LICENSE
AGREEMENT (EULA) AND AGREE TO THEM. [YES | NO | EULA]?: yes
```

If you enter the EULA option, the output displays the complete EULA text. After this action, it is not possible to confirm EULA acceptance again. In some cases, an integrator may have accepted the EULA on behalf of the actual end user.

A new command is now available to display the EULA acceptance with a timestamp of the event:

```
CLI (network-admin@pn-sw-01) > eula-show
End User License Agreement
Pluribus Networks, Inc.'s ("Pluribus", "we", or "us") software
products are designed to provide fabric networking and
analytics solutions that simplify operations, reduce operating
expenses, and introduce applications online more rapidly.
Before you download and/or use any
 of our software, whether alone or as loaded on a piece of
equipment, you will need to agree to the terms of this End
User License Agreement (this "Agreement").
...
PN EULA v. 2.1

eula-show: No fabric
eula-show: Fabric required. Please use fabric-create/join/show
CLI (network-admin@pn-sw-01) >
```

Using the Serial Console Port for Initial Configuration

This procedure assumes that you have installed the switch in the desired location and it is powered on.

> **Warning:** Do not connect any ports to the network until the switch is configured. You can accidentally create loops or cause IP address conflicts on the network.

If you are going to cable host computers to the switch, there is an option to enable or disable host ports by default.

1. Connect the console port on the rear or front (depending on the model) of the switch to your laptop or terminal concentrator using a serial cable.
2. From the terminal emulator application on your computer, log into the switch with the username **network-admin** and the default password **admin**.

> **Note:** Netvisor ONE supports both IPv4 and IPv6 addresses for the in-band interface.

> **Warning:** Be sure to type in a static IP address for the management interface during the initial configuration. Netvisor One initially uses DHCP to obtain an IP address, but DHCP is not supported after the initial configuration.

3. Begin the initial configuration using the initialization procedure displayed.
4. Enter the following details when prompted, an example is provided in the output below:
    - Accept the EULA agreement
    - Type-in the switch name. An example is provided in the output below.
    - Enter and re-enter the password
    - Enter the Management IP and netmask. An example is provided in the output below.
    - Enter the In-band IP and netmask. An example is provided in the output below.
    - Enter the IP address of the Gateway.
    - Enter the IP address for the primary and secondary DNS.
    - Enter the domain name.

```
switch console login: network-admin
Password: admin

Netvisor OS Command Line Interface 5.1.0
By ANSWERING "YES" TO THIS PROMPT YOU ACKNOWLEDGE THAT YOU
HAVE READ THE TERMS OF THE PLURIBUS NETWORKS END USER LICENSE
AGREEMENT (EULA) AND AGREE TO THEM. [YES | NO | EULA]?: yes
Switch setup required:
Switch Name (netvisor): pn-switch-01
```

```
network-admin Password: password <return>
Re-enter Password: ******** <return>
Mgmt IP/Netmask (dhcp): 10.14.2.42/23
Mgmt IPv6/Netmask:
In-band IP/Netmask: 12.1.165.21/24
In-band IPv6/Netmask:
Loopback IP:
Loopback IPv6:
Gateway IP (10.14.2.1):
Gateway IPv6:
Primary DNS IP: 10.135.2.13
Secondary DNS IP: 10.20.4.1
Domain name: pluribusnetworks.com
Automatically Upload Diagnostics (yes):
Enable host ports by default (yes):

nvOS system info:
    serial number: 1918PN8500165
    hostid:                                     b001720
    device id:                    561TG02
Switch Setup:
Switch Name          : pn-switch-01
Switch Mgmt IP       : 10.14.2.42/23
Switch Mgmt IPv6     : fe80::4e76:25ff:feef:5140/64
Switch In-band IP    : 12.1.165.21/24
Switch In-band IPv6  : fe80::640e:94ff:fe20:8787/64
Switch Loopback IP   : ::
Switch Loopback IPv6 : ::
Switch Gateway       : 10.14.2.1
Switch IPv6 Gateway  : ::
Switch DNS Server    : 10.135.2.13
Switch DNS2 Server   : 10.20.4.1
Switch Domain Name   : pluribusnetworks.com
Switch NTP Server    :
Switch Timezone      : America/Los_Angeles
Switch Date          : 2019-09-20,11:30:49
Enable host ports    : yes
Analytics Store      : default
Fabric required. Please use fabric-create/join/show
Connected to Switch pn-switch-01; nvOS Identifier:0xb001720;
Ver: 5.1.0-5010014980
```

When you setup a switch for initial configuration, the host facing ports are enabled by default. However, you can disable the host ports until you are ready to plug-in host cables to the switch. If Netvisor ONE does not detect adjacency on a port during the `quickstart` procedure, the ports remain in the disabled state.

To enable the ports after plugging in cables, use the `port-config-modify port port-list host-enable` command. Netvisor ONE enables host ports by default unless you specify NO during the `quickstart` procedure as displayed below.

```
Netvisor OS Command Line Interface 5.1.0
By ANSWERING "YES" TO THIS PROMPT YOU ACKNOWLEDGE THAT YOU
```

```
HAVE READ THE TERMS OF THE PLURIBUS NETWORKS END USER LICENSE
AGREEMENT (EULA) AND AGREE TO THEM. [YES | NO | EULA]?: yes
Switch setup required:
Switch Name (netvisor): pn-switch-01
network-admin Password: password <return>
Re-enter Password: ******** <return>
Mgmt IP/Netmask (dhcp): 10.14.2.42/23
Mgmt IPv6/Netmask:
In-band IP/Netmask: 12.1.165.21/24
In-band IPv6/Netmask:
Loopback IP:
Loopback IPv6:
Gateway IP (10.14.2.1):
Gateway IPv6:
Primary DNS IP: 10.135.2.13
Secondary DNS IP: 10.20.4.1
Domain name: pluribusnetworks.com
Automatically Upload Diagnostics (yes):
Enable host ports by default (yes): no
```

To verify, use the command:

```
CLI (network-admin@pn-switch-01) > port-show port 9,10
```

```
switch        port  bezel-port   status              config
------------  ----  ---------- -------------------   ------
pn-switch-01   9      3        phy-up,host-disabled    10g
pn-switch-01   10     3.2      phy-up,host-disabled    10g
```

To enable the port (s), use the command:

```
CLI (network-admin@pn-switch-01) > port-config-modify port
9,10 enable host-enable
```

```
CLI (network-admin@pn-switch-01) > port-show port 9,10
```

```
switch        port  bezel-port  status      config
------------  ---- ----------  ----------  ------
pn-switch-01   9     3           up,vlan-up  fd,10g
pn-switch-01   10    3.2         up,vlan-up  fd,10g
```

You cannot change (enable or disable) the host-ports by using the switch setup process after the initial configuration is done. If you try to modify the host-ports, Netvisor ONE displays an error as displayed in the example here:

```
CLI (network-admin@pn-switch-01) > switch-setup-modify
disable-host-ports
```

```
switch-setup-modify: disable/enable host ports can be set only
at initial switch-setup time
```

During the initial configuration of the switch, if the host ports are disabled, then all ports having the same port configuration will be disabled. This can be viewed using the following command:

```
<CLI (network-admin@pn-switch-01) > port-config-show port
port-list host-enable
```

In this mode, when any port comes up physically, Netvisor ONE automatically sends and receives LLDP packets to look for peer switches. If Netvisor ONE does not detect an adjacency within 5 seconds, the port is flagged as `host-disabled`. With this flag set, Netvisor ONE only accepts LLDP packets and does not initiate packet transmission.

```
CLI (network-admin@pn-switch-01) > port-config-show port 9,10

switch        port  bezel-port       status             config
------------  ----  ----------  -----------------------  ------
pn-switch-01   9     3          up,vlan-up,PN-other,LLDP  fd,10g
pn-switch-01   10    3.2        up,vlan-up                fd,10g
```

After completing switch discovery and fabric creation, use the `host-enable` option to enable host, server, or router traffic switching, and ports:

```
CLI (network-admin@pn-switch-01) > port-config-modify port 9
host-enable
```

Managing Netvisor ONE Certificates

Pluribus Networks includes the Netvisor ONE certificates along with the switches during shipment and you can access the certificates from /var/nvos/certs directory. These certificates are necessary for communication between switches in a fabric and hinders the transactions between fabric members if the certificate expires. You can view the validity (dates valid from and dates valid until) for Netvisor ONE certificate using the `switch-info-show` command.

When you configure the alarm, the certificate is checked every 24 hours and an alarm is issued if the number of days of expiry is equal to or less than 30 days . The certificate expiry alert is enabled by default for 30 days, but can configured between 7 days through 180 days on Netvisor ONE. You can disable this feature using the `cert-expiration-alert-modify no-netvisor` command.

You can view the certificate expiration alert or alarm configuration by using the `cert-expiration-alert-show` command and can schedule an alert notification before the certificate expires. You can view the alarm or alert notification in the `event.log` file and also by running the `log-alert-show command.` You can also configure a new `SNMP trap` for certificate expiry on the SNMP services.

Alarm is an event in the *event log*, an alert in `log-alert-show` command and a new SNMP trap if the trap server is configured. Frequency of alarm will be every 24 hours until the certificate has expired.

To configure the certificate expiry alert, use the command:

```
CLI (network-admin@switch01) > cert-expiration-alert-modify
```

Specify one or more of the following options:	
`netvisor|no-netvisor`	Specify whether to enable or disable Netvisor ONE certificate expiration alerts.
`days-before-expiration 7..180`	Modify the number of days before expiration to send alerts (Default 30 days). The value ranges from 7 through 180 days.

To view the alert configuration for the certificate expiry, use the command:

```
CLI (network-admin@switch01) > cert-expiration-alert-show

switch:                        switch01
days-before-expiration(d):     30
```

To enable or disable the SNMP trap for certificate expiry alert, use the command:

```
CLI (network-admin@switch01) > snmp-trap-enable-modify cert-
expiry|no-cert-expiry
```

where,

`cert-expiry|no-cert-expiry`	Specify whether to monitor certificate expiry or not.

To view the alert configuration details older than an hour, use the command:

```
CLI (network-admin@switch01) > log-alert-show older-than 1h

time      switch     code   name                count    last-message
-------- ---------- -----  ------------------   -----   ------------------------------
00:17:05  switch01   31008  smf_nvOSd_stop         1     SMF Service stopping nvOSd
00:17:08  switch01   11008  nvOSd_start            1     version 5.1.5010014665
00:35:49  switch01   31016  certificate_expiry     1     switch cert expiring in 19 days
```

The `switch-info-show` command displays the validity (dates valid from and dates valid until) for Netvisor ONE certificate. For example,

```
CLI (network-admin@nru03-sw-1*) > switch-info-show

model:                     NRU03
chassis-serial:            1937ST9100075
cpu1-type:                 Intel(R) Xeon(R) CPU D-1557 @
1.50GHz
cpu2-type:                 Intel(R) Xeon(R) CPU D-1557 @
1.50GHz
cpu3-type:                 Intel(R) Xeon(R) CPU D-1557 @
1.50GHz
cpu4-type:                 Intel(R) Xeon(R) CPU D-1557 @
1.50GHz
system-mem:                30.6G
switch-device:             OK
fan1-status:               OK
fan2-status:               OK
fan3-status:               OK
fan4-status:               OK
fan5-status:               OK
fan6-status:               OK
fan7-status:               OK
fan8-status:               OK
fan9-status:               OK
fan10-status:              OK
fan11-status:              OK
fan12-status:              OK
ps1-status:                OK
ps2-status:                OK
disk-model:                Micron_1300_MTFDDAV256TDL
disk-firmware:             M5MU000
disk-size:                 238G
disk-type:                 Solid State Disk, TRIM Supported
```

```
bios-vendor:                 American Megatrends Inc.
bios-version:                1.00.00
netvisor-cert-valid-from: Sep 13 07:00:00 2019 GMT
netvisor-cert-valid-till: Sep 14 06:59:59 2039 GMT
```

Setting the Date and Time

You can set the date and time on a switch by modifying the switch configuration using the `switch-setup-modify` command. For example, to change the date and time to September 24, 2019, 09:30:00, use the following command syntax:

```
CLI (network-admin@Leaf1) > switch-setup-modify date 2019-09-
24 T09:30:00
```

To display the configured setting, use the `switch-setup-show` command:

```
CLI (network-admin@Leaf2) > switch-setup-show
switch-name:               Leaf2
mgmt-ip:                   10.14.30.18/23
mgmt-ip-assignment:        static
mgmt-ip6:                  2721::3617:ebff:fef7:94c4/64
mgmt-ip6-assignment:       autoconf
mgmt-link-state:           up
mgmt-link-speed:           1g
in-band-ip:                192.168.101.7/24
in-band-ip6:               fe80::640e:94ff:fe83:cefa/64
in-band-ip6-assign:        autoconf
gateway-ip:                10.14.30.1
dns-ip:                    10.20.4.1
dns-secondary-ip:          172.16.1.4
domain-name:               pluribusnetworks.com
ntp-server:                0.us.pool.ntp.org
ntp-secondary-server:      0.ubuntu.pool.ntp.org
timezone:                  America/Los_Angeles
date:                      2019-09-24,09:30:00
hostid:                    184555395
location-id:               7
enable-host-ports:         yes
banner:                    * Welcome to Pluribus Networks Inc.
Netvisor(R). This is a monitored system.   *
device-id:                 1WDQX42
banner:                    *                ACCESS RESTRICTED
TO AUTHORIZED USERS ONLY                    *
banner:                    * By using the Netvisor(R) CLI,you
agree to the terms of the Pluribus Networks *
banner:                    * End User License Agreement
(EULA). The EULA can be accessed via              *
banner:                    *
http://www.pluribusnetworks.com/eula or by using the command
"eula-show"     *
```

Changing the Default Timezone

By default, Netvisor sets the default timezone to US/Pacific Standard Time (PST).

To change the timezone, use the switch-setup-modify command:

```
CLI (network-admin@Leaf1) > switch-setup-modify timezone time-
zone name
```

Viewing User Sessions on a Switch

Netvisor ONE enables you to view the user sessions on a specified switch and displays all currently logged-in users along with the IP address of the user and login time when you run the command, `mgmt-session-show`. This information is useful for troubleshooting purposes or while dealing on issues with Pluribus Customer Support teams.

```
CLI (network-admin@Leaf1) > mgmt-session-show
```

Specify any of the following:

`user` *`user-string`*	Displays the user name.
`cli-user` *`cli-user-string`*	Displays the name used to log into the switch.
`pid` *`pid-number`*	Displays the process ID.
`terminal` *`terminal-string`*	Displays the terminal ID
`from-ip` *`ip-address`*	Displays the IP address of the user.
`login-time date/time:` *`yyyy-mm-ddTHH:mm:ss`*	Displays the time and date that the user logged into the switch.
`remote-node` *`remote-node-string`*	Displays the name of the remote node.
`vnet` *`vnet-string`*	Displays the vNET assigned to the user.
`type cli|api|shell`	Displays the type of login session.

For example,

```
CLI (network-admin@Leaf1) > mgmt-session-show

user   cli-user       pid    terminal  from-ip       login-time     type
----   -------         ---    -------  --------      ---------      ----
admin  network-admin   13805  pts/3    10.60.1.216   11:20:52         cli
root   network-admin   8589   pts/2    10.14.20.109  11-15,17:16:17   cli
       network-admin                                 08:24:10         cli
root                  19139   pts/1     10.14.22.54   11-15,11:01:08  shell
```

In this example, the `root` user represents the user who has all access by default, while the admin user has only customized access privileges.

Archiving Netvisor ONE Log Files Outside the Switch

Netvisor ONE enables you to archive the nvOSd log files to an external file server periodically and these log files may be helpful for troubleshooting purposes. As a network administrator, you can configure the following parameters to enable archiving of the log files:

- Server IP address and hostname
- Username and password
- Log archival interval (minimum interval is 30 minutes and the default value is 24 hours)

On configuring this feature, the log archival configuration parameters are saved in the *log_archival_config.xml* file with an encrypted password string. A binary file deciphers the configuration parameters and also the files that are to be archived. Netvisor ONE sends an empty *time-stamped directory* to the configured remote server path and subsequently, all the log files are archived to the newly created remote directory. The new directory in the remote server is created in the *nvOS_archive.yyyymmdd_hh.mm.ss* format.

Netvisor ONE uses the Secure Copy Protocol (SCP) to archive the log files from the switch to the remote external server at specific intervals. Using the `enable` or `disable` parameter in the CLI command, you can start or stop archiving of the log files. You can archive regular log files, a set pattern of log files, or a whole directory from one of the following paths only. If you add files from other paths than the directories specified here, Netvisor ONE displays an error.

- /var/nvOS/log/*
- /nvOS/log/*
- /var/log/*

Use the below CLI commands to configure the log archival parameters and schedule the archival interval.

To modify the archival schedule parameters for the log files, use the command,

```
CLI (network-admin@switch-1) > log-archival-schedule-modify
```

<table>
<tr><td>enable|disable</td><td>Specify to enable or disable the log archival schedule.</td></tr>
<tr><td colspan="2">Specify one or many of the following options:</td></tr>
<tr><td>archive-server-username <em>&lt;string&gt;</em></td><td>Specify the SCP username of log archival server.</td></tr>
<tr><td>archive-server <em>&lt;string&gt;</em></td><td>Specify the IP address or hostname of the log archival server.</td></tr>
</table>

archive-server-path <*string*>	Specify the SCP server path to archive the log files in.
archive-interval <*30..4294967295*>	Specify the log archival interval in minutes. The range varies from 30 minutes to 4294967295 minutes with a default value of 1440 minutes (one day).
archive-server-password <*string*>	Enter the SCP server password.

For example, if you had modified the log-archival-schedule by specifying the archive-server-username as `pn-user,` archive-server as `pn-server`, and `archive-server-path` as `/home/pn-server/workspace/log_archival_tests,` use the command,

```
CLI (network-admin@switch-1) > log-archival-schedule-modify
archive-server-username pn-user archive-server pn-server
archive-server-path /home/pn-
server/workspace/log_archival_tests
```

To display the modified configuration, use the command,

```
CLI (network-admin@switch-1) > log-archival-schedule-show

switch:                   switch-1
archive-server-username: pn-user
archive-server:          pn-server
archive-server-path:     /home/pn-
server/workspace/log_archival_tests
enable:                  no
archive-interval(m):     1440
```

To add the log files to the archival list, use the command,

```
CLI (network-admin@switch-1) > log-archival-schedule-files-add
log-file log-file-string
```

log-file *log-file-string*	Specify a comma-separated list of log file names to add to the archive list.

For example, to add the `nvOSd.log` or `audit.log` or all *log* files or a whole directory, use the following commands:

```
CLI (network-admin@switch-1) > log-archival-schedule-files-add
log-file /var/nvOS/log/nvOSd.log

CLI (network-admin@switch-1) > log-archival-schedule-files-add
log-file /var/nvOS/log/*.log

CLI (network-admin@switch-1) > log-archival-schedule-files-add
log-file /var/log
```

```
CLI (network-admin@switch-1) > log-archival-schedule-files-add
log-file /nvOS/log/audit.log
```

To view the list of log files that you had scheduled to be archived, use the command,

```
CLI (network-admin@switch-1) > log-archival-schedule-files-
show

/var/nvOS/log/nvOSd.log
/var/nvOS/log/*.log
/var/log
/nvOS/log/audit.log
```

If you try to add an unsupported file or directory, an error message is displayed. For example,

```
CLI (network-admin@switch-1) > log-archival-schedule-files-add
log-file /var/nvOS

/var/nvOS, not from valid logs supported
```

To remove the log files or a list of log files from the archival list, use the command,

```
CLI (network-admin@switch-1) > log-archival-schedule-files-
remove log-file log-file-string
```

Guidelines and Tips

- When the `log-archival-schedule` is enabled and if all files are removed from the archival list, the log-archival-schedule gets disabled.
- If the systemd timer expires before the previous log-archival process is finished, then the systemd waits for the process to complete before starting the new process.

Exporting Configurations Using Secure Copy Protocol (SCP)

The SCP is a network protocol based on the BSD RCP protocol supporting file transfers between hosts on a network. SCP uses Secure Shell (SSH) for data transfer and uses the same mechanisms for authentication, and ensures the authenticity and confidentiality of the data in transit. A client can upload files to a server, optionally including basic attributes such as permissions or timestamps. Clients can also download files or directories from a server. SCP runs over TCP port 22 by default. Like RCP, there is no RFC that defines the specifics of the protocol.

In Netvisor One, you are prompted for a password when the `upload-server` option is provided in the CLI.

```
CLI (network-admin@Leaf1) > switch-config-export upload-server
10.1.1.1
server password:
```

During the software upgrade process, Netvisor One exports the switch configuration and moves it to a shared directory. The exported configuration archive is accessible from all boot environments. Netvisor One exports the configuration before the start of the software upgrade. Netvisor One stores a maximum of three configuration archives on the switch. Older configurations are deleted.

New parameters in Netvisor One support this feature:

```
CLI (network-admin@Leaf1) > switch-config-export

export-file                export a nvOS config file
upgrade-location-mappings hostid=locationid comma separated
list to upgrade old config.
upload-server              upload config file to server via scp

CLI (network-admin@Leaf1) > switch-config-export
```

Specify any of the following options:

`export-file switch-config` *`export-file`*	Exports the specified nvOS configuration file.
`upgrade-location-mappings` *`upgrade-location-mappings-string`*	Specify the upgrade location mappings by specifying the hostid=locationid comma separated list to upgrade old configuration file.
`upload-server` *`upload-server-string`*	Uploads the config file to server via SCP

For example,

```
CLI (network-admin@Leaf1) > switch-config-export upgrade-
location-mappings 0xb001a48=1 export-file switch-config-reset-
backup.2019-10-15T02.12.40.tar.gz upload-server root@ursa-
scale-leaf1:/root
server password:
Uploaded configuration to server at /root
CLI (network-admin@Leaf1) >
```

```
CLI (network-admin@aries-test-1) > switch-config-export
export-file switch-config-reset-backup.2019-10-
15T02.12.40.tar.gz upload-server root@ursa-scale-leaf1:/root
server password:
Uploaded configuration to server at /root
CLI (network-admin@aries-test-1) >
```

Displaying and Managing Boot Environment (BE) Information

Netvisor ONE provides two boot environments (BEs): the current boot environment, and the previous boot environment. Having the two BEs allows you to rollback or rollforward the software versions or configurations.

To display boot environment information, use the following command:

```
CLI (network-admin@Leaf1) > bootenv-show
```

```
name       version     current reboot space created        apply-current-config
--------- ----------- ------- ------ ----- -------------- --------------------
netvisor-1 3.1.1-13800  no      no     0     08-29,14:13:35  false
netvisor-2 5.0.0-14540  yes     yes    0     08-29,17:24:17  false
```

To reset the boot environment and reboot using the previous environment, use the following syntax:

```
CLI (network-admin@Leaf1) > bootenv-activate-and-reboot name 
netvisor-1
```

To delete a boot environment, use the following syntax:

```
CLI (network-admin@Leaf1) > bootenv-delete name netvisor-2
```

You can display information about different boot environments on the switch.

Auto-configuration of IPv6 Addresses on the Management Interface Support

IPv6 Stateless Address Auto-Configuration (SLAAC)

Like IPv4 addresses, you can configure hosts in a number of different ways for IPv6 addresses. Dynamic Host Configuration Protocol (DHCP) assigns IPv4 addresses dynamically and static addresses assign fixed IP addresses. DHCP provides a method of dynamically assigning addresses, and provides a way to assign the host devices other service information like DNS servers, domain names, and a number of different custom information.

SLAAC allows you to address a host based on a network prefix advertised from a local network router using Router Advertisements (RA). RA messages are sent by default by IPv6 router.

These messages are sent out periodically by the router and include following details:

- One or more IPv6 prefixes (Link-local scope)
- Prefix lifetime information
- Flag information
- Default device information (Default router to use and its lifetime)

Netvisor ONE enables SLAAC by default on the switch.

When you configure IPv6 address on the management interface during setup, the parameter, **assignment**, has two options:

- **none** — Disables IPv6 addresses.
- **autoconf** — Configure the interface with SLAAC.

Managing RMAs for Switches

A primary case for an RMA is a failed switch in the network. The configuration can be restored to a replacement switch using the following commands:

- fabric-join
- fabric-join repeer-to-cluster-node
- switch-config-import

For details on the RMA process, contact Pluribus Technical Support team.

Support for Local Loopback IP Addresses

Netvisor ONE uses the loopback interface as an always up and available virtual interface, and you can assign it a unique IPv4 or IPv6 address. Netvisor ONE uses a loopback interface as a termination address for some routing protocols, because of the availability of the interface. Netvisor ONE allows you to configure a loopback address for a global zone.

- Send a dedicated ping to loopback interface
- Create a BGP neighbor using the loopback Interface with OSPF so reach-ability is there for BGP and BGP next hop self
- Make sure log messages do not show any issues

Netvisor ONE deploys the loopback IP address as persistent in the configuration and not affected by a reboot or reset of Netvisor ONE.

To add a loopback IPv4 or IPv6 address or both to an existing configuration, use the following syntax:

```
CLI (network-admin@switch1) > switch-setup-modify loopback-ip
ip-address loopback-ipv6 ipv6-address
```

For example, to add the IPv4 address, 12.1.1.1, and the IPv6 address, 1212::1, use the following syntax:

```
CLI (network-admin@switch1) > switch-setup-modify loopback-ip
12.1.1.1 loopback-ip6 1212::1

CLI (network-admin@switch1) > switch-setup-show format in-
band-ip,in-band-ip6,loopback-ip,loopback-ip6, layout
horizontal

in-band-ip   in-band-ip6 loopback-ip loopback-ip6
------------ ----------- ----------- ------------
150.1.1.1/24 2001::1/96  12.1.1.1    1212::1
150.1.1.2/24 2001::2/96  12.1.1.2    1212::2
```

After configuring the loopback address, you can SSH to the switch over the management, in-band, or loopback interface using the following syntax:

```
CLI (network-admin@switch1) > ssh network-
admin@<mgmt/inband/loopback ip-address>
```

Then from CLI, execute the `shell` command to access the switch shell:

```
CLI (network-admin@switch1) > network-admin@switch:~$
```

Modifying and Upgrading Software

A switch can contact an upgrade server, either directly or through a proxy, to download and upgrade to a newer version of Netvisor ONE. You can modify the upgrade process for the switch and add a proxy host.

To complete a software upgrade:

- Obtain the required upgrade software. You can download the software manually and copy it to a switch before beginning the upgrade procedures.

> Pluribus Networks recommends upgrading the software and fabric using offline packages, that is, download the required software package to your system or copy to a USB flash drive and then use the package offline to upgrade the software and the fabric.

- Software and fabric upgrades two phases: the installation (upgrade) of the new software and a switch reboot to activate the new software.
- It is critical to understand the effects of an upgrade before beginning the process. For example, the reboot behavior after an upgrade can be completed manually or it can be controlled automatically with command options.

> **Informational Note**:This upgrade procedure applies to only one switch. To upgrade switches on the fabric, see *Implementing a Fabric Upgrade* section.

Software Tracks

Pluribus Networks manages different software releases using software tracks. By default, the *software track-release* is the standard track, but other tracks, such as Beta or Hotfixes, may be available for download.

1) To view the current version of Netvisor ONE on the switch, use the following command:

```
CLI network-admin@switch > software-show

version: 5.0.0-5000014538
```

2) If the upgrade status indicates the availability of a newer version of Netvisor ONE, request an update from the server:

```
CLI (network-admin@Leaf1)>software-upgrade

upgrade successful. rebooting...
```

Check the status while the switch is upgrading, use the `software-upgrade-status-show` command.

3) Check the status of the switch after upgrading, reconnect to the switch, and enter the following command:

```
CLI (network-admin@Leaf1)> software-show

version:                 5.1.1-5010115297
track:                   5.1-release
upgrade-status:          up-to-date
auto-upgrade:            disable
use-proxy:               no
```

> **Note:** Allow plenty of time for the switch to download and install the new version of software. Do not interrupt the operation while the upgrade is in progress. When the upgrade is complete, the switch reboots and loads the latest version of the software. If you encounter any problems with the new version of the software, a previous version can be selected as the boot software.

> **Note:** Upgrading without an Internet connection - If the switch does not have direct access to the Internet but can use a proxy server, enter the `software-modify use-proxy` command to configure the proxy and then check for software upgrade availability. If there is no access to the Internet from the switch, contact Pluribus Technical Support for instructions on upgrading a switch offline.

To upgrade the current Netvisor ONE to a later release, use the `software-upgrade` command.

```
CLI (network-admin@Leaf1) > software-upgrade package nvOS-
5.1.1-5010115280-onvl.pkg
```

The parameter `package` allows you to specify the name of the upgrade file.

> **Caution**: Do not reboot or power off any switch during the upgrade procedure. The reboot process may be completed automatically or manually as part of the upgrade procedure but do not reboot or power off a switch while the upgrade is in process.

After the upgrade, to view the software version, use the `software-show` command.

```
CLI (network-admin@Leaf1)>software-show

version:                 5.1.1-5010115297
track:                   5.1-release
```

Implementing a Fabric Upgrade

Netvisor ONE enables you to implement a fabric-wide upgrade and reboot the switches at the same time or in a sequential order.

Starting the Fabric Upgrade

Upgrading to a newer version of Netvisor ONE uses following steps:

- Establishing a serial console connection to the switch
- Performing disk checks and free disk space checks
- Copying the upgrade package to the switch
- Starting the upgrade process by using the command, fabric-upgrade-start.

The `fabric-upgrade` command establishes a controller node (primary node) to manage the fabric upgrade procedure. This node is instantiated on the switch where the `fabric-upgrade` command is run.

Copy the downloaded software to the switch (using *sftp*) that will be the controller or primary node. The controller node monitors the progress of the upgrade on each node and can view the status of the upgrade using the `fabric-upgrade-status-show` command. The controller node is identified by an "*" after its name in the status output.

The software package download is a relatively simple operation but it is important to consider the platforms that need to be upgraded in a fabric. The following package name examples reflect the platform name associated with each package.

For each platform in a fabric, a corresponding package (nvOS-X.X.X-XXX-ONVL.pkg) is needed for the fabric upgrade process. For example, if you are upgrading to 5.1.1 release on a F9372 platform, the package name is: nvOS-5.1.1-5010115297-onvl.pkg

Following are the commands that control the software or fabric upgrade process:

- `software-upgrade` – upgrade a single switch and reboot automatically

or

- `fabric-upgrade-start` – assign the controller node and begin the upgrade process with options (see Fabric Upgrade Command Variables below)
- `fabric-upgrade-status-show` – monitor the progress of the upgrade for each node in the fabric
- `fabric-upgrade-finish` – assuming auto-finish option is not used, begin the reboot process based on start options
- `fabric-upgrade-abort` – abort the upgrade process and return switches to their prior state, no reboot needed
- `fabric-upgrade-continue` – continue the upgrade if an error occurs on a particular switch

The `fabric-upgrade-start` command defines all the future behavior of the upgrade process, meaning any optional settings need to be defined with the "start" command. See optional settings in the next section. In addition, the `fabric-`

`upgrade-start` command acquires a configuration lock from all the members of the fabric. No configuration changes are permitted during the upgrade process.

The fabric upgrade feature has two phases:

- **Upgrade** — start the upgrade which creates and updates Netvisor ONE to new boot environments but does not reboot the fabric.

- **Reboot** — reboots the entire fabric after all server-switches are upgraded to new boot environments. It is also possible during this phase to abort the process and discard the new boot environments.

> **Note**: The fabric is locked during the entire process and you cannot change any configurations during the process.

Before You Begin the Fabric Upgrade

Before you begin, you may want to consider the following options for the `fabric-upgrade-start` command:

- `auto-finish` — you can specify to automatically reboot the entire fabric after the upgrade is complete. The default is `no-auto-finish.`

- `abort-on-failure` — specify if you want the upgrade to stop if there is a failure during the process.

- `manual-reboot` — specify if you want to manually reboot individual switches after the upgrade process. If you specify no-manual-reboot, all switches reboot automatically after the upgrade is complete.

- `prepare` — specify if you want to perform setup steps prior to performing the upgrade. This step copies the offline software package and then extracts and prepares it for the final upgrade process. Once you begin the prepare process, you cannot add new switches to the fabric.

You can check the status of the upgrade using the `fabric-upgrade-status-show` command:

```
CLI (network-admin@sw1) > fabric-upgrade-status-show

 log                    switch                 state
 ---                    ------                 -----
 (0:00:36)Upgrading
 software upgrade       sw3                    Running
 framework
 (0:00:08)Computing
 package update         sw2                    Running
 requirements
 (0:00:12)Agent
 needs restart          sw1*                   Agent restart wait
```

The first entry in the log is the duration of the upgrade process. It does not include waiting time. The switch with the asterisk (*) is the controller server-switch where the `fabric-upgrade-start` command was issued.

Additional commands for the fabric upgrade feature:

- `fabric-upgrade-finish` — you can issue this command at any time during the fabric upgrade to reboot all nodes in the fabric and complete the upgrade. Once the upgrade phase is complete, all server-switches display the "Upgrade complete" message in the log field. You can then safely reboot the fabric.
- `fabric-upgrade-abort` — aborts the software upgrade process. All changes to the server-switches are cleaned up and the server-switches do not reboot. The configuration lock on the fabric is also released.

If you issue the fabric-upgrade-abort command during the upgrade process, it may take some time before the process stops because the upgrade has to reach a logical completion point before the changes are rolled back on the fabric. This allows the proper cleanup of the changes.

- `fabric-upgrade-prepare-cancel` — cancels a fabric upgrade that was prepared earlier.
- `fabric-upgrade-prepare-resume` — resume a fabric upgrade that was prepared earlier.
- `fabric-upgrade-prepare-show` — displays the status of prepared upgrades on the fabric nodes.

Review bootenv

A new boot environment is built during the upgrade process. Upon reboot this new boot environment becomes active and the new software is up-and-running on the switch. Generally, it is not required to interact with the boot environments during the upgrade process. It may be necessary to review the boot environments using the command bootenv-show if there is some failure during the upgrade process.

Saving and Restoring Netvisor One Configurations

A switch contains local configuration information such as port settings as well as fabric configuration information. Fabric configurations are stored on every switch in the fabric and does not require that you save and restore before replacing a switch. When a switch is replaced, removed, or otherwise disrupted, you can save and restore the local configuration information.

The information that is saved and restored on the local switch includes the following:

- VNETs with vNET manager running on the switch
- Port VLAN associations
- Network services running on the switch

To display a full list of the current configuration details for a switch, use the `running-config-show` command.

SFTP and NFS can be used to transfer the configuration file, but you must enable the two features before using them.

> **Caution:** There is a potential for data loss when restoring a configuration. The configuration on the switch is replaced by the configuration stored in the import file. Although ISO images and disk-library images are not likely to disappear, you should only perform `switch-config-import` on a switch that doesn't have important data stored on it. As a precaution, consider using the command `switch-config-export` to save the data on the switch that you are importing the configuration file. Also, copy the ISO images and disk images from the switch using the `iso-image-library` and `disk-library-image-export` commands and copying the files from the switch.

To save the switch configuration to a file, use the following command:

```
CLI (network-admin@Leaf1) > switch-config-export export-file
pleiades24

Exported configuration to /nvOS/export/pleiades24.2013-11-
04T22.33.31.tar.gz
```

To display the files available for import and export, use the following command:

```
CLI (network-admin@Leaf1) > switch-config-show

switch       export-file
pleiades24   pleiades24.2013-11-04T22.33.31.tar.gz
```

You can now copy the configuration file to a different host using SFTP or NFS. For example, you can SFTP to the `switch-ip-address`, and login using the SFTP password. Then use cd/nvOS/import, and use get to download the configuration file.

The switch-config-export command is used to export the configuration of the local switch. The file that is created is a tar file that includes a number of configuration files for the switch. The file is created under /nvOS/export. This is the command used to export the current configuration on the local switch. Also, each time you reset the switch using the command, switch-config-reset, a backup of the configuration is made and places a file in the same location.

Once the switch configuration is exported, it becomes available to import on the same switch, by using the switch-config-copy-to-import command. Netvisor One copies the configuration tar file from the /nvOS/export to the /nvOS/import directory. Once in the /nvOS/import directory, it is possible to use the switch-config-import command to import the switch configuration.

- The switch-config-import command is used to import a configuration on the local switch. When using that command, the intention is to import a switch configuration t previously exported by the same switch.
- The switch-config-import command has a few parameters to it. The ignore-system-config and the apply-system-config parameters are two parameters that allow the imported configuration of the switch to override or not override the currently configured information found under the switch-setup-show command. When you select the ignore-system-config parameter, the local configuration is saved to an archive. If you select apply-system-config, the settings in the tar file are applied to the local switch.
- When you import a configuration using the switch-config-import command, the current configuration on the switch is overwritten by the imported configuration file.
- The skip-fabric-join option imports the fabric configuration from the tar file. However, this information may be out of date with respect to the fabric if transactions have occurred on the fabric since the file was exported which causes the imported configuration to be out-of-sync with the current fabric. The alternative is to specify do-fabric-join, which extracts the fabric name from the tar file, and attempts to join the fabric and download the current fabric configuration, so that it is in sync with the rest of the fabric. The fabric configuration in the tar file is ignored, but cluster and local configurations are imported from the tar file.

When a switch that was part of a cluster is replaced, the fabric-join repeer-to-cluster-node command is used for the new switch to receive all required switch configuration, including the local configuration.

To upload a configuration file to a switch and set the configuration for the switch using the configuration file, you must transfer the configuration file to the target switch using the following sequence of commands:

```
sftp@<switch-ip-address>
Connecting to switch-ip-address
Password: <password>
sftp> cd nvOS/import
sftp> put pleiades24.2013-11-04T22.33.31.tar.gz
```

> **Note:** The configuration file must use the `*.tar.gz` extension to be recognized by nvOS.

> **Caution:** Loading the configuration file causes nvOS to restart which results in a brief interruption to switch traffic flow.

Now load the configuration file which replaces the current configuration on the switch with the information in the file.

```
CLI (network-admin@Leaf1) > switch-config-import import-file
pleiades24.2019-11-04T22.33.31.tar.gz

New configuration imported. Restarting nvOS...
Connected to Switch pleiades24; nvOS Identifier:0xb000011;
Ver: 5.1.1.15297
```

There are many options available that allow you to control how the `switch-config-import` modifies the switch, including the following:

- `ignore-system-config` — ignore the current system configuration.The settings in the `*.tar` file are not applied to the local switch.
- `apply-system-config` — apply the system configuration in the imported file. The settings in the `*.tar` file are applied to the local switch. You typically do not want to use this option as it changes the in-band IP address and other settings.
- `skip-fabric-join` — opt out of joining the fabric. This setting imports the fabric configuration from the `*.tar` file, but this information may be out of date with respect to the fabric if additional transactions occur on the fabric since the file was exported.
- `do-fabric-join` — join the current fabric. This setting extracts the fabric name from the `*.tar` file and attempts to join the fabric. Then the switch contacts the current fabric to download the configuration so that the switch is in sync with the rest of the fabric. Cluster and local configurations are imported from the `*.tar` file.
- `no-replace-switch` — do not replace the current switch.
- `replace-switch` — replace the current switch. This setting is used to replace a faulty switch and after importing the file, has the same configuration as the replaced switch. This replaces all of the local, cluster, and fabric configuration by downloading the configurations from peer switches. No configuration is necessary or advised before running this command. However, you need to run the initial quickstart to obtain an in-band IP address.

By default, the initial switch system configuration, management IP addresses and other parameters, are not applied if there is another switch in the fabric with the same settings. To apply the initial settings, use the `apply-system-config` option. Also, by default, the imported configuration attempts to join the same fabric that the original switch was a member. If that join fails, then the import fails. You can avoid this issue by using the `skip-fabric-join` option.

Finally, if the original switch is still on the network and you want to copy the configuration to a new switch, but you want to prevent the new switch from taking ownership of any objects specific to the original switch, such as vNET services, or VLAN port settings, you must use the `no-replace-switch` option.

Copying and Importing Configuration Files

You can create a configuration file to import to another switch by using the `switch-config-copy-to-import` command. To create a configuration file with the name config-092613 to import on another switch, use the following syntax:

```
CLI (network-admin@Leaf1) > switch-config-copy-to-import
export-file config-092613
```

After you create the configuration file, you can export it to `/nvOS/export/` directory, and SFTP to it from the target switch.

To review the available files for import and export, use the following syntax:

```
CLI (network-admin@Leaf1) > switch-config-show

switch        export-file
pbg-nvos      config-092613.tar.gz
```

Depending on the available remote access services, you can now copy the configuration file to a different switch. For example, you can SFTP to another switch using the IP address of the switch, login as SFTP with the password that you previously set, `cd /nvOS/import` and `get` the configuration file.

To upload the configuration file to the target switch and set the configuration from the configuration file, transfer the configuration file to the target switch with the IP address, 192.168.3.35.

To export a configuration to a server, use the `switch-config-export` command:

```
CLI (network-admin@Leaf1) > switch-config-export
```

Rolling Back to Previous Versions of Netvisor One

After upgrading to a newer version of Netvisor One, you can rollback to an earlier version and preserve the current configuration. The new configuration is applied before booting into the previous environment so that critical ACLs and security vFlows are present when Netvisor One restarts.

A new parameter, `apply-current-config`, for the command, `bootenv-active-and-reboot`, provides support for this feature.

Before rebooting, Netvisor One copies the current boot environment transaction logs into the target boot environment.

After rebooting, Netvisor One performs the following:

- Reads the copied transaction logs, and sorts all transactions by time, then scope, fabric>cluster>local, and then the transaction ID.
- Parses the list of all transactions from oldest to newest.
- If the current transaction ID for the scope is less, Netvisor One rolls the transaction forward, and deletes the files when done.

> **Caution:** Retaining the current configuration when booting to an older version of Netvisor One is best-effort. Some transaction IDs from the newer (or current) version may not properly apply due to feature incompatibility. It is not guaranteed that all changes are applied.

> **Caution:** You must apply the parameter, `apply-current-config`, on all nodes in the fabric. There is no coordination across the fabric for this process, therefore the commitment of fabric transactions on one node but not another using this process causes the fabric to go out of sync and may result in unrecoverable errors.

About Pluribus Networks

Pluribus Networks delivers an open, controllerless software-defined network fabric for modern data centers, multi-site data centers and distributed cloud edge environments.

The Linux-based Netvisor® ONE operating system and the Adaptive Cloud Fabric™ have been purpose-built to deliver radically simplified networking and comprehensive visibility along with white box economics by leveraging hardware from our partners Celestica, Dell EMC, and Edgecore, as well as Pluribus' own Freedom™ Series of switches.

The Adaptive Cloud Fabric provides a fully automated underlay and virtualized overlay with comprehensive visibility and brownfield interoperability and is optimized to deliver rich and highly secure per-tenant services across data center sites with simple operations having no single point of failure.

Further simplifying network operations is Pluribus UNUM™, an agile, multi-functional web management portal that provides a rich graphical user interface to manage the Adaptive Cloud Fabric. UNUM has two key modules - UNUM Fabric Manager for provisioning and management of the fabric and UNUM Insight Analytics to quickly examine billions of flows traversing the fabric to ensure quality and performance.

Pluribus is deployed in more than 275 customers worldwide, including the 4G and 5G mobile cores of more than 75 Tier 1 service providers delivering mission-critical traffic across the data center for hundreds of millions of connected devices. Pluribus is networking, simplified.

For additional information contact Pluribus Networks at info@pluribusnetworks.com, or visit www.pluribusnetworks.com.

Follow us on Twitter @pluribusnet or on LinkedIn at https://www.linkedin.com/company/pluribus-networks/.

Corporate Headquarters

Pluribus Networks, Inc.
5201 Great America Parkway, Suite 422
Santa Clara, CA 95054
1-855-438-8638 / +1-650-289-4717

India Office

Pluribus Networks India Private Limited
Indiqube Brigade Square, 4th Floor
21, Cambridge Road
Bangalore 560008

Document Version - June 2020