CONFIGURATION GUIDE

Extreme Network OS Management Configuration Guide, 7.4.0

Supporting Network OS 7.4.0

9036106-00
April 2019

Legal Notice

Extreme Networks, Inc. reserves the right to make changes in specifications and other information contained in this document and its website without prior notice. The reader should in all cases consult representatives of Extreme Networks to determine whether any such changes have been made.

The hardware, firmware, software or any specifications described or referred to in this document are subject to change without notice.

Trademarks

Extreme Networks and the Extreme Networks logo are trademarks or registered trademarks of Extreme Networks, Inc. in the United States and/or other countries.

All other names (including any product names) mentioned in this document are the property of their respective owners and may be trademarks or registered trademarks of their respective companies/owners.

For additional information on Extreme Networks trademarks, please see: www.extremenetworks.com/company/legal/trademarks

Open Source Declarations

Some software files have been licensed under certain open source or third-party licenses. End-user license agreements and open source declarations can be found at: www.extremenetworks.com/support/policies/software-licensing

Contents

Preface

This section discusses the conventions used in this guide, ways to provide feedback, additional help, and other Extreme Networks® publications.

Conventions

This section discusses the conventions used in this guide.

Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

NOTE
A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

ATTENTION
An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

CAUTION
A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

DANGER
A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Text formatting conventions

Text formatting conventions such as boldface, italic, or Courier font may be used to highlight specific words or phrases.

Format	Description
bold text	Identifies command names.
	Identifies keywords and operands.
	Identifies the names of GUI elements.
	Identifies text to enter in the GUI.
italic text	Identifies emphasis.
	Identifies variables.
	Identifies document titles.

Format	Description
`Courier font`	Identifies CLI output. Identifies command syntax examples.

Command syntax conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
[ ]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{ **x** | **y** | **z** }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options.
x | **y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a "soft" line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Documentation and Training

To find Extreme Networks product guides, visit our documentation pages at:

Current Product Documentation	www.extremenetworks.com/documentation/
Archived Documentation (for earlier versions and legacy products)	www.extremenetworks.com/support/documentation-archives/
Release Notes	www.extremenetworks.com/support/release-notes
Hardware/Software Compatibility Matrices	https://www.extremenetworks.com/support/compatibility-matrices/
White papers, data sheets, case studies, and other product resources	https://www.extremenetworks.com/resources/

Training

Extreme Networks offers product training courses, both online and in person, as well as specialized certifications. For more information, visit www.extremenetworks.com/education/.

Getting Help

If you require assistance, contact Extreme Networks using one of the following methods:

Extreme Portal	Search the GTAC (Global Technical Assistance Center) knowledge base, manage support cases and service contracts, download software, and obtain product licensing, training, and certifications.
The Hub	A forum for Extreme Networks customers to connect with one another, answer questions, and share ideas and feedback. This community is monitored by Extreme Networks employees, but is not intended to replace specific guidance from GTAC.
Call GTAC	For immediate support: 1-800-998-2408 (toll-free in U.S. and Canada) or +1 408-579-2826. For the support phone number in your country, visit: www.extremenetworks.com/support/contact

Before contacting Extreme Networks for technical support, have the following information ready:

- Your Extreme Networks service contract number and/or serial numbers for all involved Extreme Networks products
- A description of the failure
- A description of any action(s) already taken to resolve the problem
- A description of your network environment (such as layout, cable type, other relevant environmental information)
- Network load at the time of trouble (if known)
- The device history (for example, if you have returned the device before, or if this is a recurring problem)
- Any related RMA (Return Material Authorization) numbers

Subscribing to Service Notifications

You can subscribe to email notifications for product and software release announcements, Vulnerability Notices, and Service Notifications.

1. Go to www.extremenetworks.com/support/service-notification-form.
2. Complete the form with your information (all fields are required).
3. Select the products for which you would like to receive notifications.

   NOTE
   You can modify your product selections or unsubscribe at any time.

4. Click **Submit**.

Providing Feedback to Us

Quality is our first concern at Extreme Networks, and we have made every effort to ensure the accuracy and completeness of this document. We are always striving to improve our documentation and help you work better, so we want to hear from you! We welcome all feedback but especially want to know about:

- Content errors or confusing or conflicting information.
- Ideas for improvements to our documentation so you can find the information you need faster.
- Broken links or usability issues.

If you would like to provide feedback to the Extreme Networks Information Development team, you can do so in two ways:

- Use our short online feedback form at https://www.extremenetworks.com/documentation-feedback/.
- Email us at documentation@extremenetworks.com.

Please provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.

About This Document

What's new in this document

On October 30, 2017, Extreme Networks, Inc. acquired the data center networking business from Brocade Communications Systems, Inc. This document has been updated to remove or replace references to Brocade Communications, Inc. with Extreme Networks, Inc., as appropriate.

NOTE

Fibre Channel (FC) is no longer supported; commands related to FC and "FCoE" (Fibre Channel over Ethernet) have been either removed or modified. However, instances of "FC" and "FCoE" and related services may still appear in CLI "show" outputs and elsewhere.

Supported hardware and software

In those instances in which procedures or parts of procedures documented here apply to some devices but not to others, this guide identifies exactly which devices are supported and which are not.

Although many different software and hardware configurations are tested and supported by Extreme Networks, Inc. for Network OS, documenting all possible configurations and scenarios is beyond the scope of this document.

The following hardware platforms are supported by this release of Network OS:

- ExtremeSwitching VDX 6740-48
- ExtremeSwitching VDX 6740T
  - ExtremeSwitching VDX 6740T-64
  - ExtremeSwitching VDX 6740T-1G
- ExtremeSwitching VDX 6940-144S
- ExtremeSwitching VDX 6940-36Q
- ExtremeSwitching VDX 8770
  - ExtremeSwitching VDX 8770-4
  - ExtremeSwitching VDX 8770-8

To obtain information about a Network OS version other than this release, refer to the documentation specific to that version.

Using the Network OS CLI

For complete instructions and support for using the Extreme Network OS command line interface (CLI), refer to the *Extreme Network OS Command Reference*.

Introduction to Data Center Fabrics

Data Center Fabric overview

Extreme Network OS is a scalable network operating system available for the Extreme data center switching portfolio products, including the VDX product line. While a Data Center Fabric can refer to many combinations of Ethernet and Fibre Channel networks working in concert, the Network OS Data Center fabrics comprise VCS Fabrics and IP Fabrics.

A Network OS VCS Fabric is a topologically flat network of Ethernet switches with shared intelligence. The VCS Fabric topology is designed for classic hierarchical Ethernet architecture with the same data center as a Extreme VCS Fabric architecture. For detailed information, refer to VCS Fabrics on page 17. A VCS Fabric is sometimes referred to as an "Ethernet Fabric" or as a "VCS cluster." In general, the concept of RBridge ID management, principal nodes, TRILL, and ISLs are concepts that pertain to a VCS Fabric.

A Network OS IP Fabric is a specialized spine-leaf topology of Ethernet switches with shared intelligence. The IP Fabric topology is designed for high-redundancy of network paths without large differences in transmission times. For detailed information, refer to the *Extreme Network OS IP Fabrics Configuration Guide*. Also, refer to the Extreme IP Fabrics Data sheet, which includes a list of features supported in a Extreme IP Fabric.

Purpose-built for mission-critical, next-generation data centers, Network OS Data Center fabrics supports the capabilities in the following table.

TABLE 1 Data Center Fabric capabilities

Feature	Capability
Simplified network management	Extreme Virtual Cluster Switching (VCS) fabrics are self-forming and self-healing, providing an operationally scalable foundation for very large or dynamic cloud deployments. Multi-node fabrics can be managed as a single logical element, and fabrics can be deployed and easily re-deployed in a variety of configurations optimized to the needs of particular workloads.
Server virtualization	Automatic Migration of Port Profile (AMPP) functionality provides fabric-wide configuration of network policies, achieves per-port profile forwarding, and enables network-level features to support Virtual Machine (VM) mobility. For more information about AMPP, refer to the "AMPP" of the *Extreme Network OS Layer 2 Switching Configuration Guide*.
Network convergence	Data Center Bridging (DCB)-based lossless Ethernet service provides isolation between IP and storage traffic over a unified network infrastructure.

In Network OS, most features are configured through a single, industry-standard command line interface (CLI). Refer to the *Extreme Network OS Command Reference* for an alphabetical listing and detailed description of all the Network OS commands.

Data Center Fabric terminology

The terms in the following table are used to describe Data Center Fabrics.

TABLE 2 Extreme Data Center Fabric terms

Term	Definition
Edge ports	In an Ethernet fabric, all switch ports used to connect external equipment, including end stations, switches, and routers.
ESI	Ethernet Segment Identifier. If a customer edge (CE) is multi-homed to two or more provider edges (PEs), the set of Ethernet links that attaches the CE to the PEs is called an "Ethernet segment". Ethernet segments must have a unique non-zero identifier.
Ethernet fabric	A topologically flat network of Ethernet switches with shared intelligence, such as the Extreme VCS Fabric.
EVI	An EVPN instance spanning across the provider edges participating in that EVPN.
EVPN	An Ethernet Virtual Private Network that connects remote sites using a Layer 2 virtual bridge.
Fabric ports	The ports on either end of an Inter-Switch Link (ISL) in an Ethernet fabric.
Inter-Switch Link (ISL)	An interface connected between switches in a VCS fabric. The ports on either end of the interface are called ISL ports or Fabric ports. The ISL can be a single link or a bundle of links forming a Extreme trunk. This trunk can either be created as a proprietary Extreme trunk, or a standard IEEE 802.3ad-based link aggregation.
IP Fabric	A spine-leaf network configuration in which each spine is physically connected to each leaf, and each leaf is physically connected to each spine. There is no physical connectivity from spine to spine or leaf to leaf, however.
RBridge	A physical switch in a VCS Fabric.
VCS Fabric	(Also: "VCS cluster" or "VCS fabric cluster") A topologically flat network of Ethernet switches with shared intelligence.
VCS ID	A unique identifier for a VCS fabric. The factory default VCS ID is 1. All switches in a VCS Fabric must have the same VCS ID.
VCS mode	From Network OS 7.1.0, the only supported VCS mode is logical chassis cluster mode, in which the entire cluster is configured from the principal node. Fabric cluster mode—in which each node kept an independent configuration database—is no longer supported.

VCS Fabrics

VCS Fabrics overview

Extreme VCS Fabrics technology is an Ethernet technology that allows you to create flatter, virtualized, and converged data center networks. Extreme VCS Fabrics technology is elastic, permitting you to start small, typically at the access layer, and expand your network at your own pace.

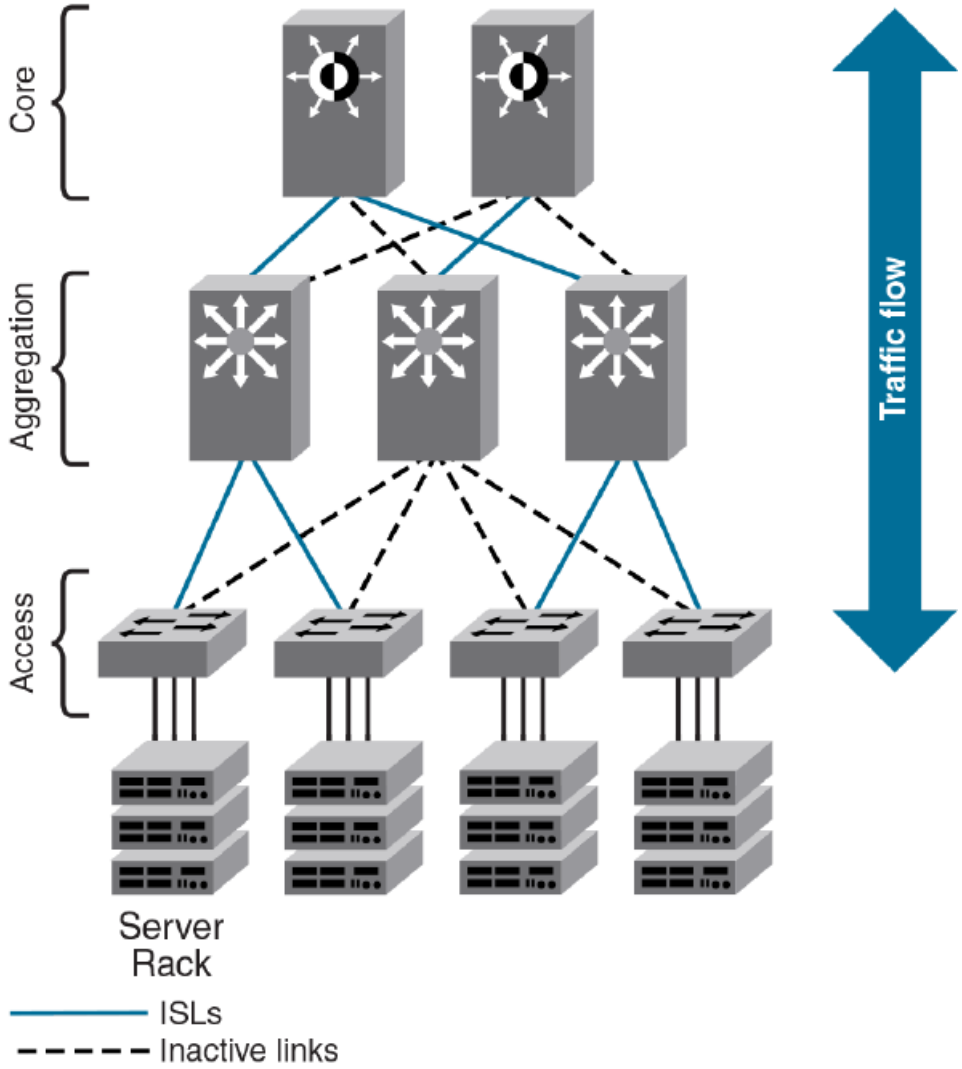
Extreme VCS Fabrics technology is built upon three core design principles:

- Automation
- Resilience
- Evolutionary design

When two or more Extreme VCS Fabrics switches are connected together, they form an Ethernet fabric and exchange information using distributed intelligence. To the rest of the network, the Ethernet fabric appears as a single logical chassis.

The following figure shows an example of a data center with a classic hierarchical Ethernet architecture and the same data center with a Extreme VCS Fabrics architecture. The Extreme VCS Fabrics architecture provides a simpler core-edge topology and is easily scalable as you add more server racks.

FIGURE 1 Classic Ethernet architecture



In addition to the standard Data Center Fabrics capabilities, VCS Fabrics provide the capabilities listed in the following table.

Feature	Capability
High resiliency	Extreme VCS fabrics use hardware-based Inter-Switch Link (ISL) Trunking to provide automatic link failover without traffic interruption.
Improved network utilization	Transparent Interconnection of Lots of Links (TRILL)-based Layer 2 routing service provides equal-cost multipaths in the network, resulting in improved network utilization. Extreme VCS Fabric technology also delivers multiple active, fully load-balanced Layer 3 gateways to remove constraints on Layer 2 domain growth, eliminate traffic tromboning, and enable inter-VLAN routing within the fabric. Virtual Router Redundancy Protocol (VRRP) eliminates a single point of failure in a static, default-route environment by dynamically assigning virtual IP routers to participating hosts. The interfaces of all routers in a virtual router must belong to the same IP subnet. There is no restriction against reusing a virtual router ID (VRID) with a different address mapping on different LANs. Refer to the "VRRP overview" section of the *Extreme Network OS Layer 3 Routing Configuration Guide* for additional information on VRRP/VRRP-E.

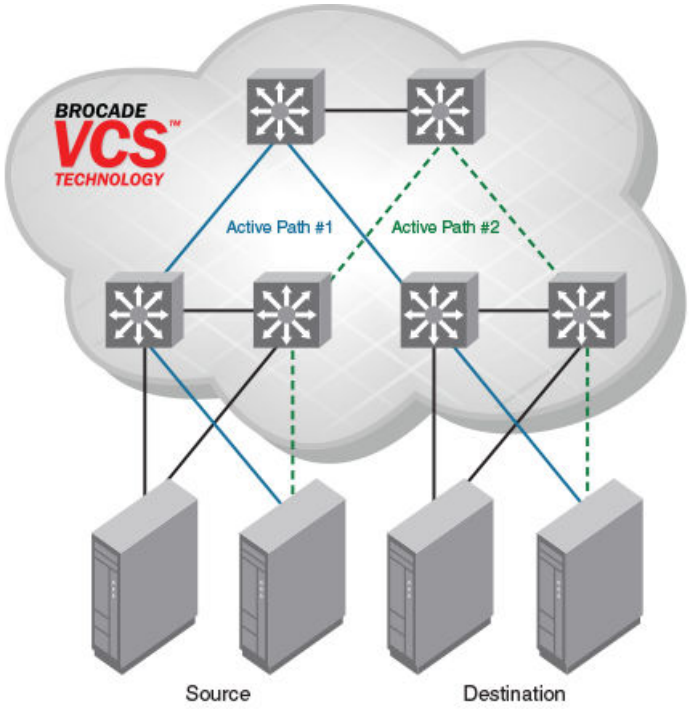
Automation

Resilience is a foundational attribute of Extreme Fibre Channel storage networks and resilience is also a requirement in modern data centers with clustered applications and demanding compute Service-Level Agreements (SLAs). In developing its VCS Fabric technology, Extreme naturally carried over this core characteristic to its Ethernet fabric design.

In traditional Ethernet networks running Spanning Tree Protocol (STP), only 50 percent of the links are active; the rest (shown as dotted lines in the previous figure) act as backups in case the primary connection fails.

When you connect two or more Extreme VCS Fabric-mode enabled switches they form an Ethernet fabric (provided the two switches have a unique RBridge ID and same VCS ID), as shown in the following figure.

FIGURE 2 Ethernet fabric with multiple paths



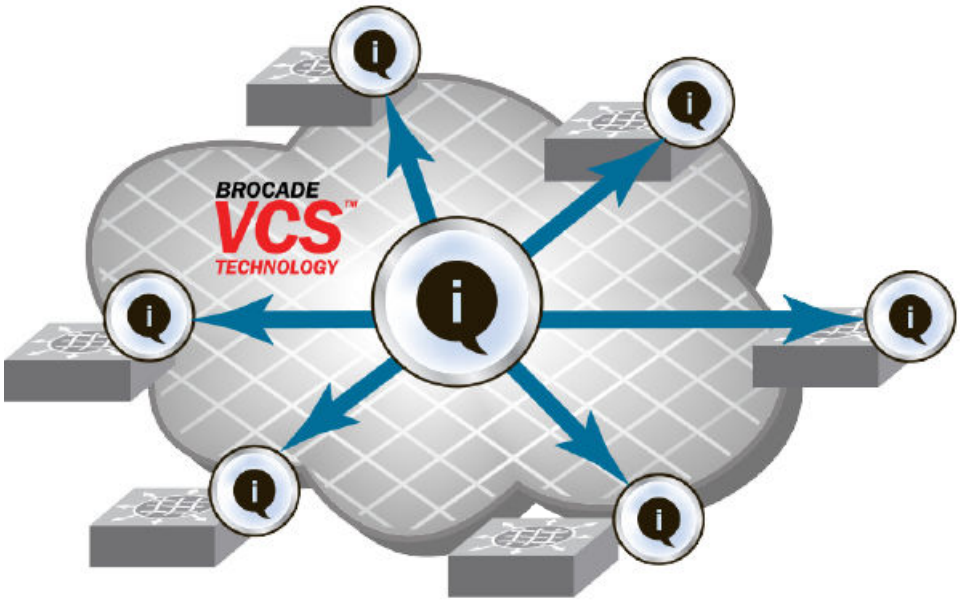
The Ethernet fabric has the following characteristics:

- It is a switched network. The Ethernet fabric utilizes an emerging standard called Transparent Interconnection of Lots of Links (TRILL) as the underlying technology.
- All switches automatically know about each other and all connected physical and logical devices.
- All paths in the fabric are available. Traffic is always distributed across equal-cost paths. As illustrated in the figure, traffic from the source to the destination can travel across two paths.
- Traffic travels across the shortest path.
- If a single link fails, traffic is automatically rerouted to other available paths. In the topology shown in the figure, if one of the links in Active Path #1 goes down, traffic is seamlessly rerouted across Active Path #2.
- STP is not necessary because the Ethernet fabric appears as a single logical switch to connected servers, devices, and the rest of the network.
- Traffic can be switched from one Ethernet fabric path to the other Ethernet fabric path.

Distributed intelligence

With Extreme VCS Fabric technology, all relevant information is automatically distributed to each member switch to provide unified fabric functionality, as illustrated in the following figure:

FIGURE 3 Distributed intelligence in a Extreme VCS Fabric



A Extreme VCS Fabric is designed to be managed as a single "logical chassis," so that each new switch inherits the configuration of the fabric, and the new ports become available immediately. The fabric then appears to the rest of the network as a single switch. This significantly reduces complexity for the management layer, which in turn improves reliability and reduces troubleshooting.

In addition, VCS Fabrics provide a NETCONF application programming interfaces (API), as well as extensions to OpenStack Quantum to orchestrate both physical and logical networking resources as part of virtual machine deployment to support multi-tiered application topologies.

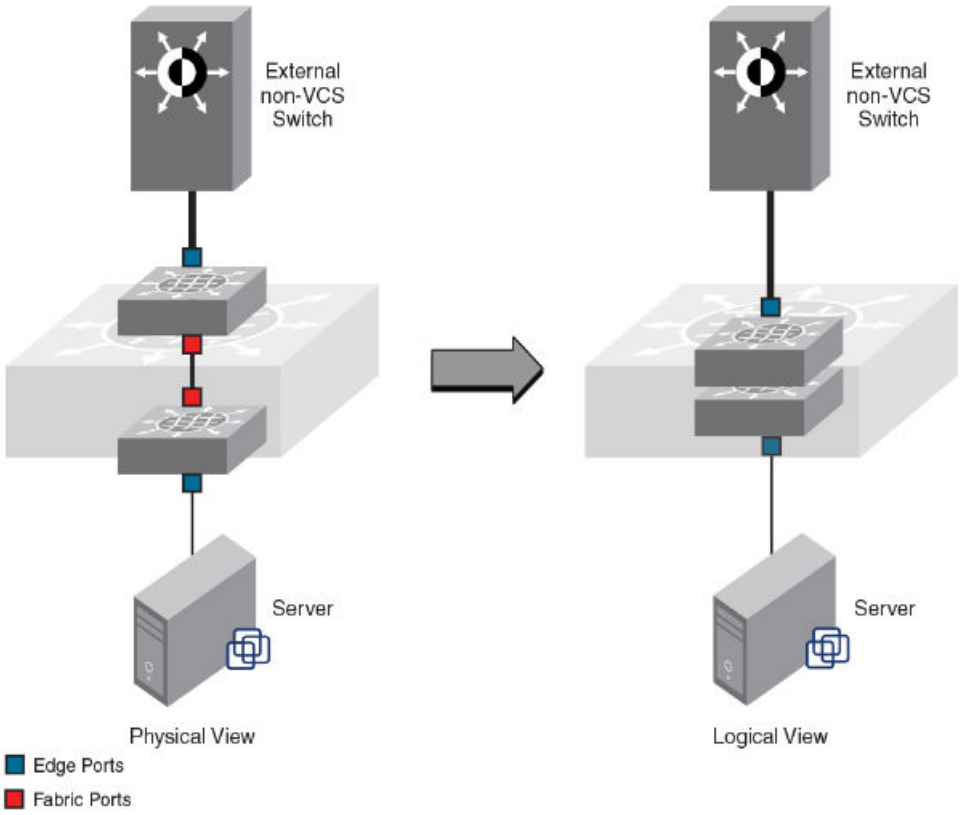
Distributed intelligence has the following characteristics:

- The fabric is self-forming. When two Extreme VCS Fabric mode-enabled switches are connected, the fabric is automatically created and the switches discover the common fabric configuration.
- The fabric is masterless. No single switch stores configuration information or controls fabric operations. Any switch can fail or be removed without causing disruptive fabric downtime or delayed traffic.
- The fabric is aware of all members, devices, and virtual machines (VMs). If the VM moves from one Extreme VCS Fabric port to another Extreme VCS Fabric port in the same fabric, the port-profile is automatically moved to the new port, leveraging Extreme's Automatic Migration of Port Profiles (AMPP) feature.

Logical chassis

All switches in an Ethernet fabric are managed as if they were a single logical chassis. To the rest of the network, the fabric looks no different from any other Layer 2 switch. The following figure illustrates an Ethernet fabric with two switches. The rest of the network is aware of only the edge ports in the fabric, and is unaware of the connections within the fabric.

FIGURE 4 Logical chassis in Ethernet fabric



Each physical switch in the fabric is managed as if it were a blade in a chassis. When a Extreme VCS Fabric-mode enabled switch is connected to the fabric, it inherits the configuration of the fabric and the new ports become available immediately.

Extreme trunks

Network OS supports Extreme trunks (hardware-based link aggregation groups, or LAGs).

These LAGs are dynamically formed between two adjacent switches. The trunk formation is controlled by the same Fibre Channel Trunking protocol that controls the trunk formation on FC switches. As such, it does not require user intervention or configuration except enabling or disabling, which instructs the switch software to form a trunk at the global level or not. All ISL ports connected to the same neighbor Extreme switch will attempt to form a trunk. Refer to Enabling a fabric trunk on page 31 for instructions.

Port groups have been established on supported standalone switches and on line cards in chassis systems for trunking. For a successful trunk formation, all ports on the local switch must be part of the same port group and must be configured at the same speed. Following are the number of ports allowed per trunk from port groups in supported platforms. For details on how port groups are arranged on these platforms, refer to the switch or chassis system Hardware Reference Manual.

- VDX 8770 switches - up to six port groups of eight ports each per blade (1 , 10, or 40 GbE)
- VDX 6740 switches - up to 16 ports per trunk

> **NOTE**
> If connections are made to 16 different switches, only eight ports will be trunk ports while the other eight ports will be normal ISL ports.

- VDX 6940-36Q switches - up to three ports per trunk
- VDX 2741 and VDX 2746 - up to 16 ports per trunk
- 48x10 GbE line card - up to eight ports per trunk
- 48x10G-T line card - up to 16 ports per trunk
- 12x40 GbE line card - up to two 40-GbE ports are allowed per trunk, and these ports must be configured in breakout mode
- 27x40 GbE line card - up to two 40-GbE ports are allowed per trunk, and these ports must be configured in breakout mode. Note that breakout mode is only allowed on the first two ports in port groups that are configured in Performance operating mode. Refer to the *Extreme 8770 Hardware Reference Manual* for more information on line card operating modes.

The following additional rules apply to Extreme trunks:

- On Extreme VDX 6740 switches, low-volume traffic below certain thresholds may not be evenly distributed on all links. This threshold can be as low as 64 K.
- The trunk is turned on by default.

Ethernet fabric formation

Extreme VCS Fabric protocols are designed to aid the formation of an Ethernet fabric with minimal user configuration.

When you connect a switch to a Extreme VCS Fabric mode-enabled switch, the Extreme VCS Fabric-mode enabled switch determines whether the neighbor also has Extreme VCS Fabric mode enabled. If the switch has Extreme VCS Fabric mode enabled and the VCS IDs match, the switch joins the Ethernet fabric.

Automatic ISL formation and hardware-based trunking

When a switch joins an Ethernet fabric, ISLs automatically form between directly connected switches within the fabric.

If more than one ISL exists between two switches, then Extreme ISL trunks can form automatically. All ISLs connected to the same neighboring Extreme switch attempt to form a trunk. The trunks are formed only when the ports belong to the same port group. No user intervention is necessary to form these trunks.

Refer to Configuring fabric interfaces on page 30 for information about enabling and disabling ISLs and trunks.

Principal RBridge election

The role of the principal RBridge is to decide whether a new RBridge joining the fabric conflicts with any of the RBridge IDs already present in the fabric. If a conflict arises, the principal RBridge keeps the joining RBridge segmented.

Extreme VCS Fabrics technology use cases

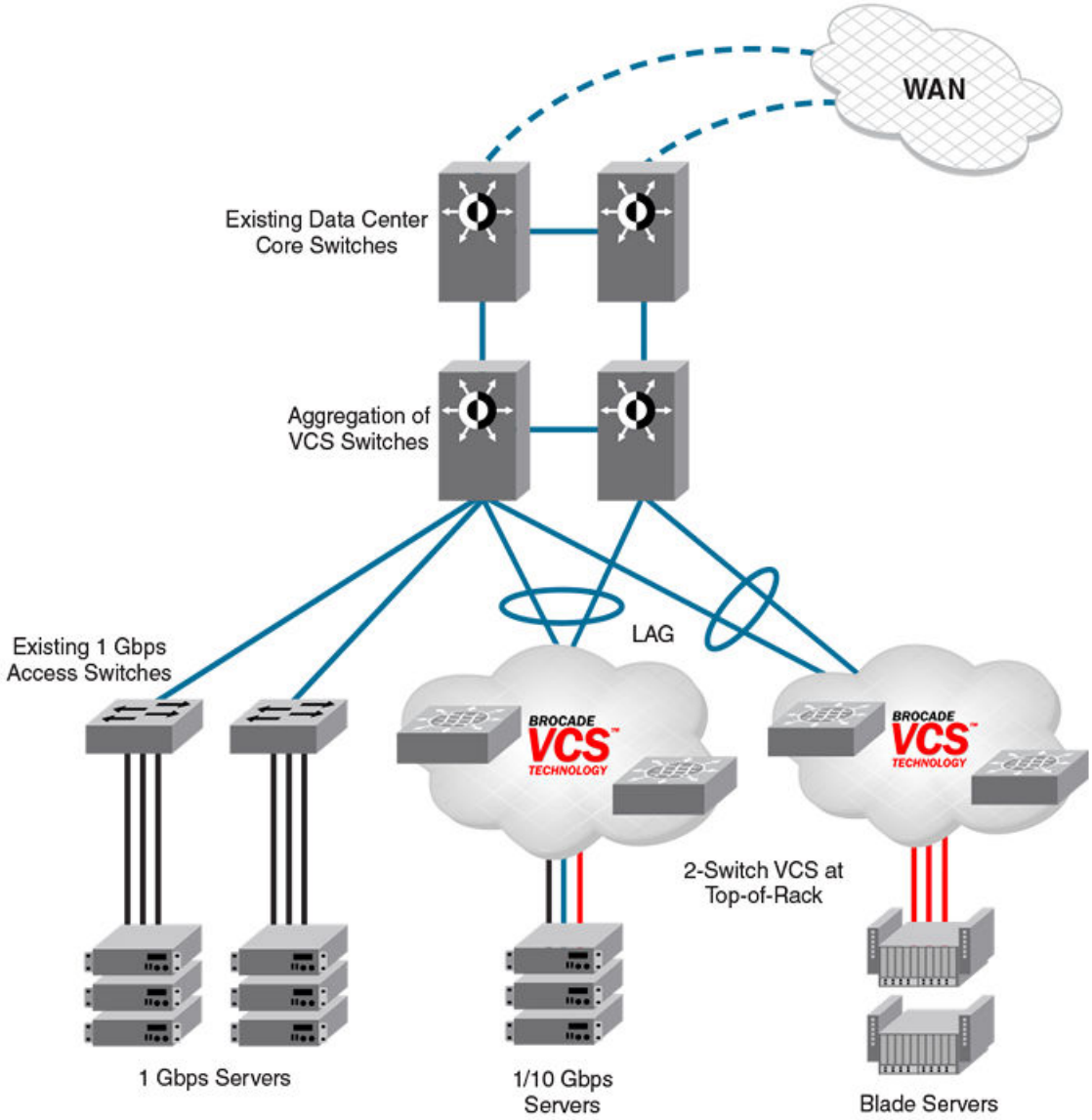
This section describes the following use cases for Extreme VCS Fabrics technology:

- Classic Ethernet
- Large-scale server virtualization
- VCS with FC SAN

Classic Ethernet access and aggregation use case

Extreme VCS Fabric can be deployed in the same fashion as existing top-of-rack switches, as shown in the following figure. In the right-most two server racks, a two-switch Ethernet fabric replaces the Ethernet switch at the top of each rack.

FIGURE 5 Pair of Extreme VDX switches at the top of each server rack



The servers perceive a single top-of-rack switch, allowing for active/active connections running end-to-end.

Extreme VCS Fabric technology in this use case provides the following advantages:

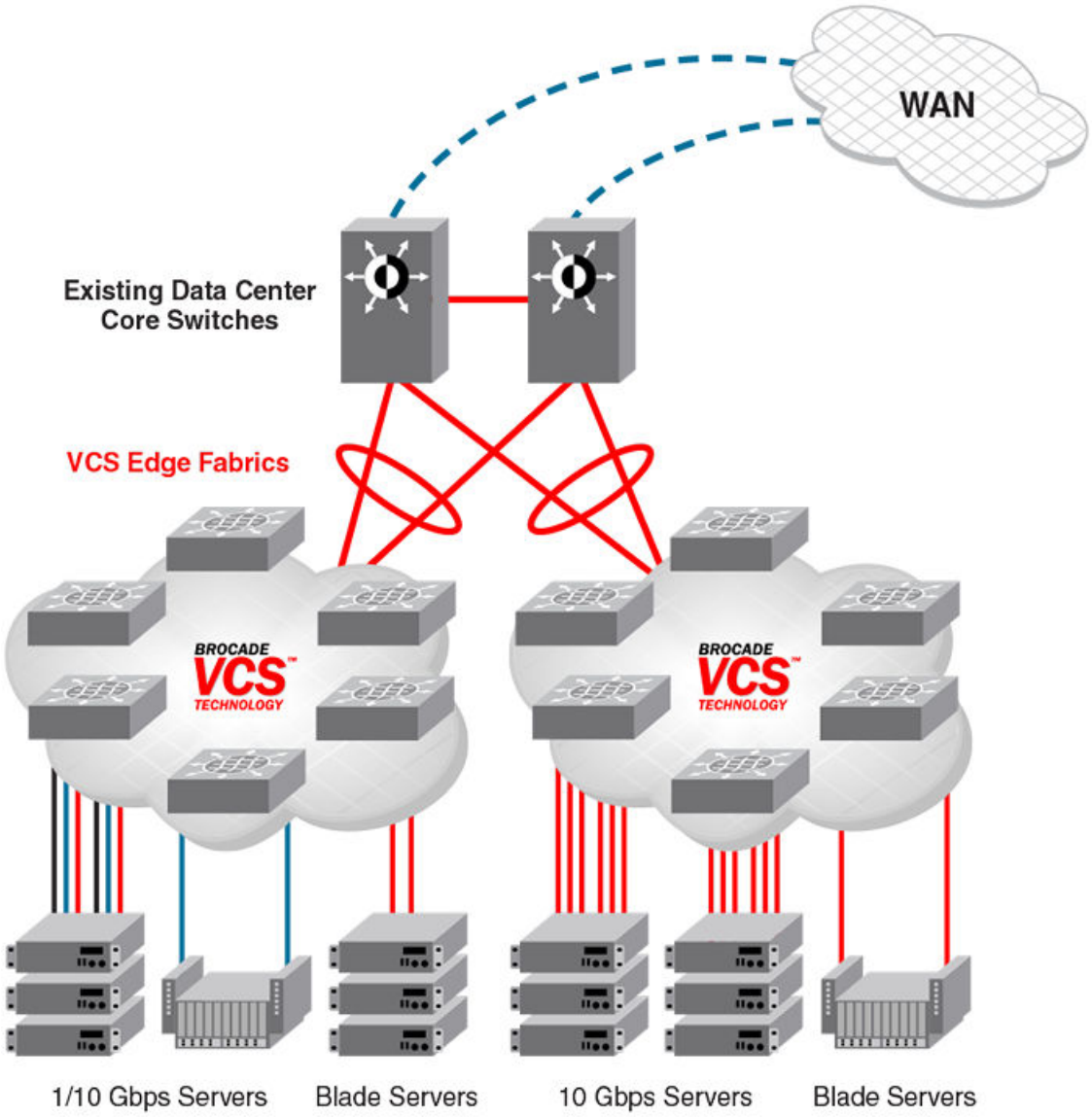
- Multiple active-active connections, with increased effective bandwidth
- Preserves existing architecture
- Works with existing core and aggregation networking products
- Co-exists with existing access switches

- Supports 1- and 10-Gbps server connectivity
- Works with server racks or blade servers

Large-scale server virtualization use case

The following figure shows an example of a logical two-tier architecture with Extreme VCS Fabrics at the edge. Each Extreme VCS Fabric appears as a single virtual switch to the switches outside the fabric, which results in flattening the network.

FIGURE 6 Collapsed, flat Layer 3 networks enabling virtual machine mobility



Extreme VCS Fabric technology in this use case provides the following advantages:

- Optimizes the multipath network (all paths and Layer 3 gateways are active, no single point of failure, and STP is not necessary)
- Increases sphere of virtual machine (VM) mobility

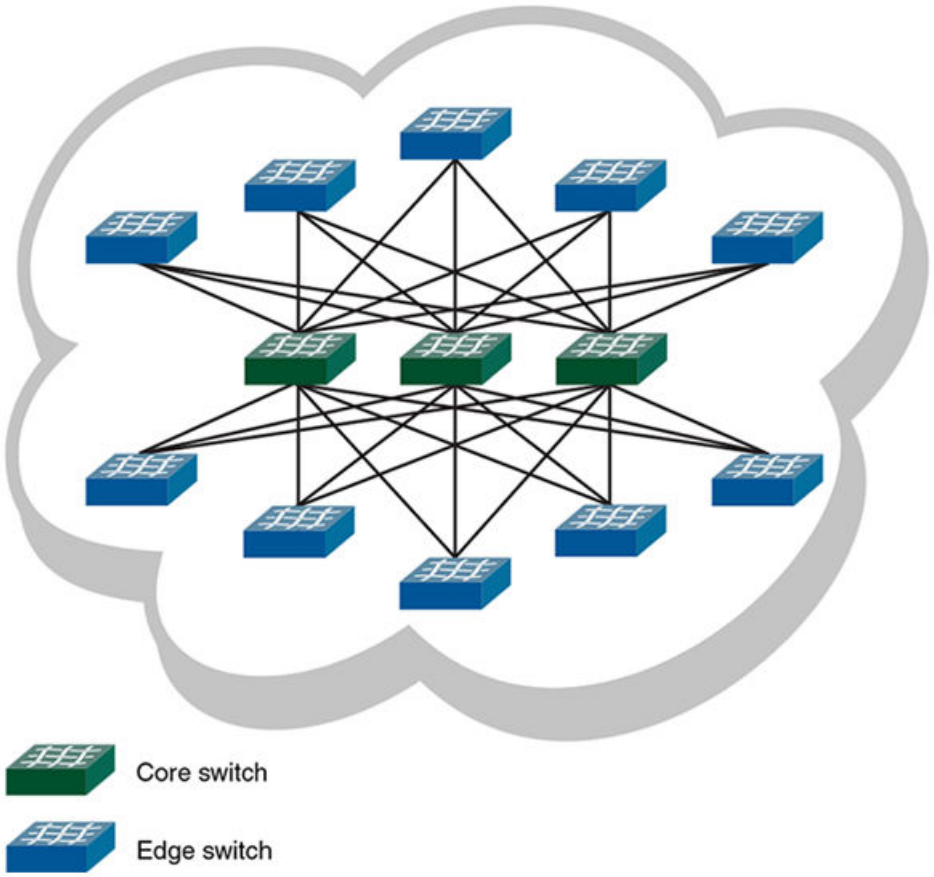
Topology and scaling

Although you can use any network topology to build a Extreme VCS Fabric, the following topics discuss the scaling, performance, and availability considerations of topologies more commonly found in data centers:

Core-edge topology

Core-edge topology devices connect to edge switches, which are connected to each other through core switches. The example shown in the following figure uses three core switches. You could use more or fewer switches in the core, depending on whether you need higher availably and greater throughput, or a more efficient use of links and ports.

FIGURE 7 Core-edge topology



This topology is reliable, fast, and scales well. It is reliable because it has multiple core switches. If a core switch or a link to a core switch fails, an alternate path is available. As you increase the number of core switches, you also increase the number of link or core switch failures your cluster can tolerate.

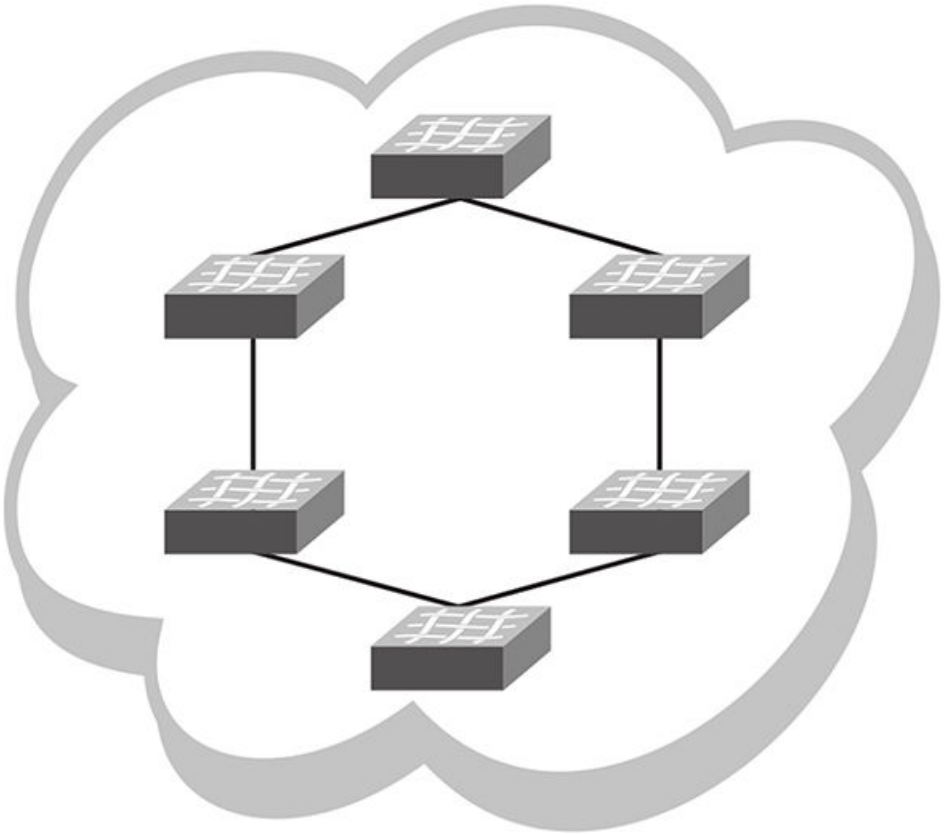
High performance and low latency are ensured because throughput is high and the hop count is low. Throughput is high because multiple core switches share the load. Two hops get you from any edge switch to any other edge switch. If you need greater throughput, simply add another core switch.

Scaling the topology also requires additional core switches and links. However, the number of additional links you need is typically not as great as with, for example, a full mesh topology.

Ring topology

Ring topology connects each node to exactly two other nodes, forming a single continuous pathway. Data travels from node to node, with each node along the path handling every packet of the data. The following figure shows a ring topology.

FIGURE 8 Ring topology

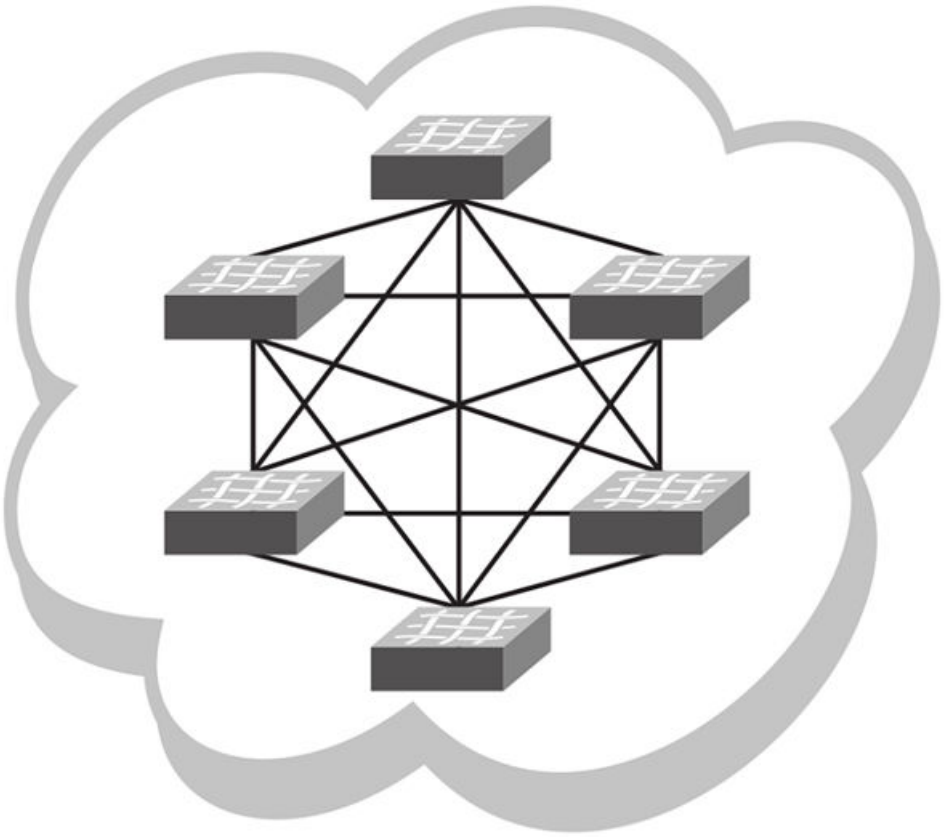


This topology is highly scalable, yet susceptible to failures and traffic congestion. It is highly scalable because of its efficient use of interswitch links and ports; an additional node requires only two ports to connect to the ring. It is susceptible to failures because it provides only one path between any two nodes. Throughput of the fabric is limited by the slowest link or node. Latency can be high because of the potentially high number of hops it takes to communicates between two given switches. This topology is useful where economy of port use is critical, but availability and throughput are less critical.

Full mesh topology

Full mesh topology connects each node to all other cluster nodes, as shown in the following figure.

FIGURE 9 Full mesh topology



This topology is highly reliable and fast, but it does not scale well. It is reliable because it provides many paths through the fabric in case of cable or node failure. It is fast with low latency because you can get to any node in the fabric in just one hop. It does not scale well because each additional node increases the number of fabric links and switch ports exponentially. This topology is suitable for smaller fabrics only.

Operational mode (logical chassis cluster)

In logical chassis cluster mode, both the data and configuration paths are distributed throughout the VCS Fabric. The entire cluster is configured from the principal node.

From version 7.1.0, Network OS supports only logical chassis cluster mode for Extreme VDX switches. Fabric cluster mode—in which each node kept an independent configuration database—is no longer supported.

VCS operational characteristics

The following are operational characteristics of a VCS Fabric:

- The maximum number of nodes supported is 48.
- In-band management (through eth0 on management modules) is supported over virtual Ethernet (VE) interfaces.

- If you use in-band management only, and wish to disable the management interface (which is considered *out of band*), refer to the **shutdown** (interface) command in the *Extreme Network OS Command Reference*.
- A single global configuration exists across all nodes, while each node can contain its unique local configuration. However, each node contains the local configuration information for all other nodes in the cluster.
- Any change that you make to the running configuration is automatically saved, with no involvement of a startup configuration file.
- When an RBridge is rejoining a VCS Fabric, the interface-level configuration is reset to the default values.
- Global and local configurations for the entire VCS Fabric are performed from the principal node only.
- If you configure a VRF on a VCS RBridge, that configuration affects the entire VCS Fabric.
- Cluster-wide firmware upgrades can be performed.
- Cluster-wide supportSave can be performed.

Temporary command blocking

Some commands cannot be run while other commands or events are processing.

If one of the following CLI command types or events is in progress in the cluster, then any one of the CLI command types in the following list will be rejected until the current command or event has finished.

- **copy file running-config** commands
- HA failover commands
- VCS ID/RbridgeID change commands
- **copy default-config startup-config**
- Configuration updates by individual commands
- Cluster formation events, such as initial cluster formation or secondary joining or rejoining of a cluster

These commands and events are considered to be blocked from occurring simultaneously. However, if the principal node changes during one of these operations or HA failover occurs on principal switch, the new principal will not retain the information that the commands or events not in progress are in a blocked state.

In the case of commands being blocked, the following messages are some of the error messages that could result:

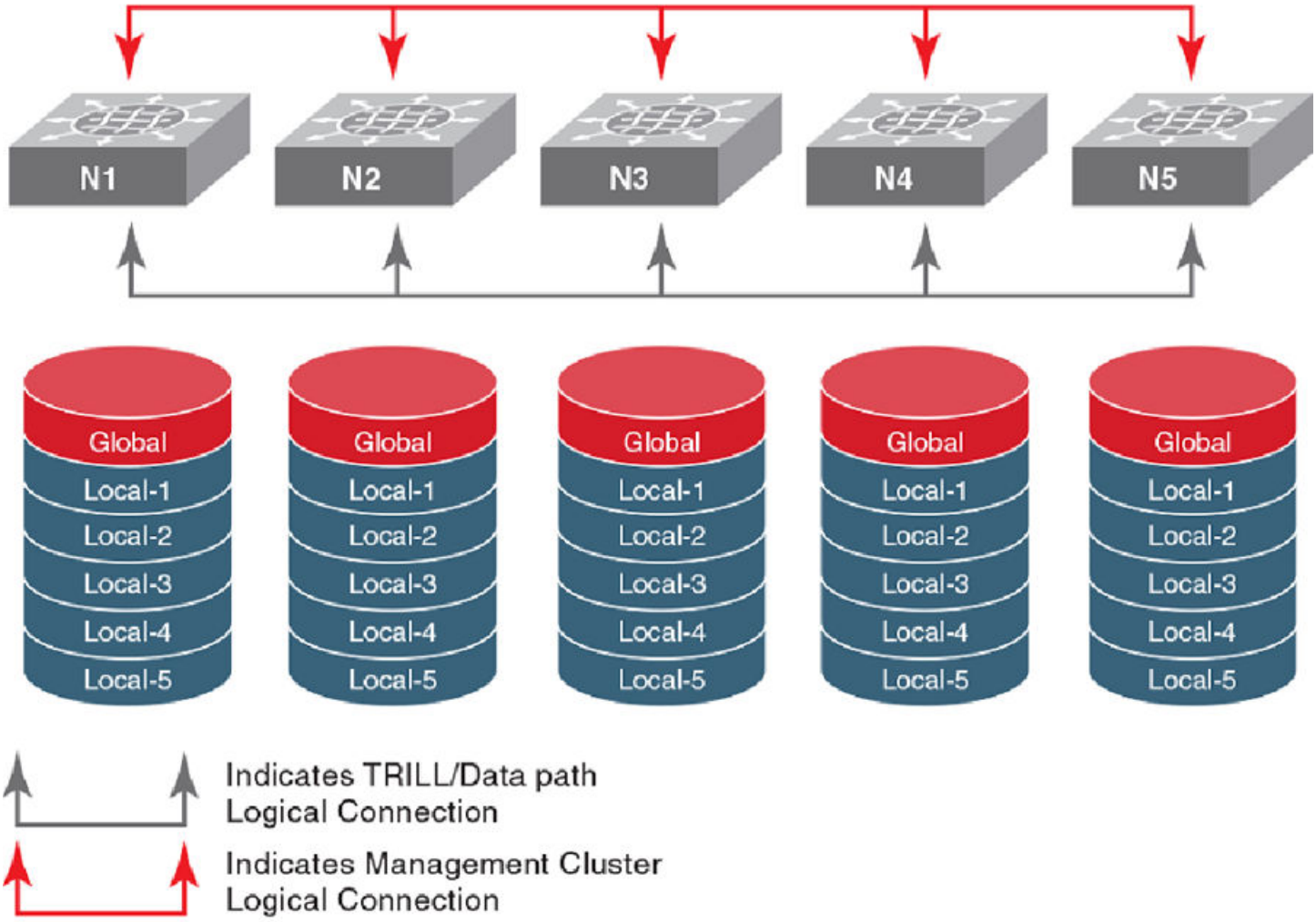
- Cluster formation is in progress. Please try again later.
- User Configuration update is in progress. Please try again later.
- Configuration file replay is in progress. Please try again later.
- HA failover is in progress in the cluster. Please try again later.
- VCS Config change is in progress in the cluster. Please try again later.
- Copy default-config startup-config is in progress. Please try again later.

Configuration database diagram

Any operation that results in writing to the configuration database gets automatically distributed. There are no exceptions.

Each node in the VCS Fabric maintains an individual copy of the configuration to enable high availability of the cluster. The following figure illustrates nodes in a VCS Fabric. Each node has its own databases, and the databases kept by each node are identical at all times.

FIGURE 10 Configuration database in a VCS Fabric



Network OS switches contain both global and local configuration. A single global configuration exists across all cluster members, while each individual member has its own local configuration.

Global configuration is required for cluster-wide operations, whereas local configuration is specific to the operation of an individual node. For more information and examples of each type of configuration, refer to Examples of global and local configurations on page 81.

Basic VCS fabric configuration

The topics in this section enable you to get a VCS up and running.

Adding a new switch into a VCS fabric

There are two methods to add a new switch into a fabric—the auto-fabric method and the manual method. This topic describes the manual method.

NOTE
For the auto-fabric method, refer to Auto Fabric on page 96.

1. Connect to the switch and log in using an account assigned to the admin role.
2. Enter the **vcs set-rbridge-id** command.

   The switch remembers its RBridge ID once it has been assigned. The **vcs set-rbridge-id** command also sets the insistent RBridge ID property on the switch.
3. Reboot the system.

   After the required reboot the switch participates in the RBridge ID allocation protocol, which insists that the same value that was manually configured prior to reboot be allocated after reboot.

   The switch is not allowed into the fabric if there is a conflict; for example, if another switch with the same ID exists and is operational in the fabric. You have the opportunity to select a new RBridge ID by using the same CLI.

   Once an ID has been assigned by the fabric protocol, these IDs are then numerically equated to RBridge IDs and are treated as such after that.

   Use the **vcs** command to configure the Extremex VCS Fabric parameters, VCS ID, and the switch RBridge ID.

   After configuring the Extreme VCS Fabric parameters, the switch applies the changes and reboots.

   The switch disable is not saved across a reboot, so if the switch was disabled prior to the reboot, the switch returns to the enabled state when it finishes the boot cycle.

Configuring fabric interfaces

A physical interface in a virtual switch cluster can either be an edge port or a fabric port, but not both. Similar to a switch-port configuration on a physical interface, you can also change a fabric-port configuration on its physical interface by using the **fabric ISL enable** and **fabric trunk enable** commands.

Enabling a fabric ISL

The **fabric isl enable** command controls whether an ISL should be formed between two cluster members. With the default setting of ISL discovery to **auto** and the ISL formation mode to **enable** , an ISL automatically forms between two cluster switches.

Performing a **fabric isl enable** command on an operational ISL has no effect. However, performing a **no fabric isl enable** command on an interface toggles its link status and subsequently disables ISL formation. In addition, the **no fabric isl enable** command triggers the switch to inform its neighbor that the local interface is ISL disabled. Upon receiving such information, a neighbor switch stops its ISL formation activity regardless of its current interface state.

NOTE
After you repair any segmented or disabled ISL ports, toggle the fabric ISL in order to propagate the changes.

NOTE
A **shutdown** command on an operating ISL interface not only brings down the physical link but also its FSPF adjacency. The main difference between a **shutdown** command and a **no fabric isl enable** command is that the link stays up after **no fabric isl enable**, while the link stays down after a shutdown.

NOTE
Upon a fabric reconvergence that due to a topology change involving the ECMP fabric-isl path, there may be sub-second flooding of known unicast traffic.

Disabling a fabric ISL

The **no fabric isl enable** command takes this interface out of the trunk group if this interface happens to be currently part of the trunk. If you know and would like to fix the edge and fabric port assignments on a switch, then this command allows you to completely turn off ISL formation logic and shorten any link bring-up delays on edge ports.

1. Connect to the switch and log in using an account assigned to the admin role.
2. Enter the **no fabric isl enable** command.

Enabling a fabric trunk

1. Connect to the switch and log in using an account assigned to the admin role.
2. Enter the **fabric trunk enable** command.

Disabling a fabric trunk

Fabric trunking is enabled by default. Enter the **no fabric trunk enable** command to revert the ISL back to a standalone adjacency between two Extreme VCS Fabric switches.

Configuring broadcast, unknown unicast, and multicast forwarding

All switches in a Extreme VCS Fabric share a single multicast tree rooted at the RBridge with the lowest RBridge ID (domain ID). All broadcast, unknown unicast, and multicast traffic between two edge RBridges is forwarded on this multicast tree inside the Extreme VCS Fabric. The multicast tree includes all RBridges in the Extreme VCS Fabric.

Multicast distribution tree-root selection

Network OS supports the following distribution tree behaviors.

- The root of the distribution tree is the switch with the lowest RBridge ID. The automated selection process does not require any user intervention.
- Each switch in the cluster optionally carries a multicast root priority. This priority setting overrides the automatically-selected multicast root. In deployments where a multicast root is required to be a specific switch that does not have the lowest RBridge ID, then the priority setting on that switch can override the root selection. If there are two switches with the same priority, then the switch with the lower RBridge ID prevails.
- A back-up multicast root is pre-selected, which is the switch with the next lowest RBridge ID. The back-up multicast root is automatically selected by all switches should the current multicast root fail.

Configuring priorities

As stated, the root of the tree is auto-selected as the switch with the lowest RBridge ID.

For example, if you had a cluster with RBridge IDs 5, 6, 7, and 8, then 5 would be the root. If you then added RBridge ID 1 to this fabric, the tree would be re-calculated with 1 as the root.

In order to avoid this behavior, you can set a priority (default is 1). The highest priority overrides the lowest RBridge ID and becomes the root.

For example, to build a fabric with RBridge ID 7 or 8 as the root, set their priority to something higher than 1 (priority values are 1 through 255). If there is a tie in priority, the lower RBridge is still chosen. If RBridge ID 7 and 8 are both set to priority 1, 7 becomes the root.

Changing the priority

1. Connect to the switch and log in using an account assigned to the admin role.
2. Enter RBridge ID configuration mode.

```
device# configure terminal
device(config)# rbridge-id 1
device(config-rbridge-id-1)#
```

3. Enter the **fabric route mcast priority** command and specify a priority, as in the following example.

```
device(config-rbridge-id-1)# fabric route mcast priority 10
```

Displaying the multicast running configuration

You can display fabric route multicast configuration information for an RBridge.

1. Connect to the switch and log in using an account assigned to the admin role.
2. In privileged EXEC configuration mode, enter the **show running-config rbridge-id** *rbridge-id* **fabric route mcast priority** command, as in the following example.

```
device# show running-config rbridge-id 12 fabric route mcast priority
fabric route mcast rbridge-id 12 priority 10
```

Configuring VCS virtual IP addresses

The virtual IP address of the principal switch automatically becomes the virtual IP address of the VCS fabric.

The management interface of the principal switch can be accessed by means of this virtual IP address. Because the virtual IP address is a property of the fabric, in the event that the principal switch goes down, the next principal switch is assigned this address.

Virtual IP address can be configured by means of the **vcs virtual ip address** command:

```
device(config)# vcs virtual ip address 10.0.0.23/24
```

When the virtual IP address is configured for the first time, the current principal switch in the cluster is assigned this IP address.

Virtual IP configuration is global in nature. All the nodes in the cluster are configured with the same virtual IP address, but address is bound to the current principal switch only. Make sure that the assigned virtual IP address is not a duplicate of an address assigned to any other management port in the cluster or network.

Extreme recommends that you use a /32 address in the same subnet as the IP address of the management interface. For example, if you are using inband management via rbridge interface ve 100 with the ip address 192.168.100.10/24, set your vcs virtual ip address as a /32 address in this subnet by using the command **vcs virtual ip address 192.168.100.1/32**. To display the currently configured virtual IP address, use the **show vcs virtual ip** command:

```
device# show vcs virtual ip
```

```
Virtual IP             : 10.21.87.2/20
Associated rbridge-id       : 2
```

To remove the currently configured virtual IP address, use the **no vcs virtual ip address** command.

```
device(config)# no vcs virtual ip address
device# show running-config vcs virtual ip address
% No entries found.
```

NOTE

You should not use the **no vcs virtual ip address** command when logged in to the switch through the virtual IP address. Use the management port IP address of the principal switch, or the serial console connection of the principal switch.

If you wish to rebind this virtual IP address to this management interface, remove the currently configured virtual IP address and reconfigure it. This situation can arise when the virtual IP address is not bound to management interface of the principal switch as a result of duplicate address detection.

A separate gateway cannot be configured for virtual IP address. The default gateway is the same as the gateway address for the management port of the same switch.

Virtual IP address configuration scenarios

Virtual IP address may be assigned to a switch whenever it is the principal switch in the cluster. The configuration scenarios that may occur are described in the following table.

TABLE 3 Virtual IP address configuration scenarios

Scenario	Description
First time cluster formation	When the cluster is first being formed, and if the virtual IP address is already configured, the principal switch is assigned the Virtual IP address. If no Virtual IP configuration exists, then the principal switch can be access using the management port IP address.
Virtual IP configuration	When you configure the virtual IP address for a cluster the first time, the address is bound to the management interface of the principal switch.
Principal switch failover	If the principal switch becomes a secondary switch while the virtual IP address is assigned to its management interface, then the virtual IP address is reassigned to the new principal switch.
Principal switch goes down	When the principal switch in the cluster goes down, the virtual IP address is released from its management interface. The virtual IP address will be assigned to the next switch that becomes the principal switch.
Principal switch chassis is disabled	When the **chassis disable** command is executed on the principal switch, the virtual IP address is released from its management interface. The virtual IP address will be assigned to the next switch that becomes the principal switch.
Virtual IP removal	If you remove the virtual IP address from the configuration, then the address is unbound from management interface of the principal switch. In this case, the principal switch can still be accessed by using the management port's IP address.
Trivial merge	In the event that two clusters merge together, the global configuration of the smaller cluster (Cluster A) is overwritten by the larger cluster (Cluster B). During this time, the virtual IP address is unbound from the principal switch of Cluster A. The virtual IP address of Cluster B can now be used to access the principal of new merged cluster. If the virtual IP address of Cluster B is not configured, there will not be a virtual IP address in the merged cluster.
Cluster reboot	When the cluster reboots, the virtual IP address is persistent and is bound to the new principal switch.

TABLE 3 Virtual IP address configuration scenarios (continued)

Scenario	Description
Cluster Islanding	If the ISL link goes down between two or more clusters that are forming, the principal switch in the original cluster retains the virtual IP address. The new principal switch in the second cluster will perform a check to confirm that the virtual IP address is not in use. If it is in use, then the address is not assigned to the switch and an error is logged in RASLog.
Virtual MAC address	Virtual MAC address are not supported by virtual IP addresses.
Management port primary IPv4 address	For a virtual IP address to work correctly, the management port's IPv4 address should be assigned and functional.

Configuring fabric ECMP load balancing

Traffic towards ECMP paths are load-balanced using the following eight fields as the Key; VlanID, MAC DA/SA, L3_ULP, L3 DA/SA, L4 Dst/Src.

For some pattern of streams, most of the traffic falls into one ECMP path, and rest of the ECMP paths are underutilized. This results in loss of data traffic, even though more ECMP paths are available to offload the traffic. You can configure the ECMP path selection method within the fabric by using the **fabric ecmp load-balance** command in RBridge ID configuration mode. The operands for this command are listed and described in the following table.

TABLE 4 ECMP load-balancing operands

Operand	Description
dst-mac-vid	Destination MAC address and VID-based load balancing
src-dst-ip	Source and Destination IP address-based load balancing
src-dst-ip-mac-vid	Source and Destination IP and MAC address and VID-based load balancing
src-dst-ip-mac-vid-port	Source and Destination IP, MAC address, VID and TCP/UDP port based load balancing
src-dst-ip-port	Source and Destination IP and TCP/UDP port-based load balancing
src-dst-mac-vid	Source and Destination MAC address and VID-based load balancing
src-mac-vid	Source MAC address and VID-based load balancing

Additionally, you can choose to swap adjacent bits of the hash key using the **fabric ecmp load-balance-hash-swap** command. This is useful in cases where a choice of any of the hash key combinations causes the distribution of traffic to not be uniform.

The **fabric ecmp load-balance-hash-swap** command is used to configure the swapping of the input fields before feeding them to the hash function. The integer is interpreted as a bitwise control of the 212-bit key. Each bit controls whether the two adjacent bits of the key are to be swapped. This 32-bit control value is written to all four hash swap control registers. This value is replicated in 32-bit block to form a 106-bit value. A value of 0x0 does not swap any input fields while a value of 0xffffffff swaps all 106 input bit-pairs.

NOTE
The **fabric ecmp load-balance-hash-swap** command is supported on the VDX 8770.

To configure the ECMP load-balancing feature, perform the following steps in global configuration mode.

1. Enter RBridge ID configuration mode.

```
device(config)# rbridge-id 2
device(config-rbridge-id-2)#
```

2. Enter the **fabric ecmp load-balance** command for the stream you want to favor.

   This example uses the Destination MAC address and VID-based load balancing flavor.

   ```
   device(config-rbridge-id-2)# fabric ecmp load-balance dst-mac-vid
   ```

3. (Optional) Use the **fabric ecmp load-balance-hash-swap** command to swap the input fields before feeding them to the hash function.

   ```
   device(config-rbridge-id-2)# fabric ecmp load-balance-hash-swap 0x4
   ```

4. Use the **show fabric ecmp load-balance** command to display the current configuration of hash field selection and hash swap.

   ```
   device# show fabric ecmp load-balance
   Fabric Ecmp Load Balance Information
   ------------------------------------
   Rbridge-Id         : 2
   Ecmp-Load-Balance Flavor  : Destination MAC address and VID based load balancing
   Ecmp-Load-Balance HashSwap : 0x4
   ```

Configuration Fundamentals

Default and running configurations

Under Network OS, when you boot up a device for the first time, the running configuration is identical to the default configuration. As you configure the device, the running configuration is modified and automatically saved.

When the device reboots, the current running configuration is reloaded.

Configuration file types

Extreme Network OS supports default and running configuration files.

TABLE 5 Device configuration files

Configuration file	Description
Default configuration • defaultconfig.novcs • defaultconfig.vcs	Part of the Network OS firmware package. The default configuration is applied, if no customized configuration is available.
Running configuration • running-config	Current configuration active on the switch. Whenever you make a configuration change, it is automatically written to the running configuration.

Default configuration files

Default configuration files are part of the Network OS firmware package and are automatically applied to the running configuration under the following conditions:

- When the switch boots up for the first time and no customized configuration is available.

- When you restore the default configuration.

You cannot remove, rename, or change the default configuration.

Running configuration files

The current device configuration (historically, the modified default configuration), is called the running configuration. The running configuration is persistent, and automatically reloads with the device.

Displaying configurations

The following examples illustrate how to display the default, startup, and running configurations, respectively.

Displaying the default configuration

To display the default configuration, enter the **show file** command with the default configuration filenames in privileged EXEC mode.

```
device# show file defaultconfig.novcs
device# show file defaultconfig.vcs
```

Displaying the startup configuration

To display the contents of the startup configuration, enter the **show startup-config** command in privileged EXEC mode.

```
device# show startup-config
```

Displaying the running configuration

To display the contents of the running configuration, enter the **show running-config** command in the privileged EXEC mode.

```
device# show running-config
```

Manual configuration restoration

If needed, you can replace the current running configuration with the default running configuration. You can also modify the current running configuration by copying from a manual backup.

NOTE
See also Automated running-config backup and restoration on page 40.

The following parameters are not affected by manual or automated configuration restoration:

- Interface management IP address
- Software feature licenses installed on the switch
- Virtual IP address

Restoring a saved configuration as the running-config

This restoration procedure merges a saved running configuration into the current running configuration:

- Settings not included in the current running configuration are copied in.

- Conflicting settings are resolved in favor of the saved running configuration.

1. In privileged EXEC mode, enter the **copy** *filename* **running-config** command, using the appropriate option:
   - To copy a backup from the device flash, specify the **flash://** keyword and the name of the backup file.

     ```
     device# copy flash://backup-config_vdx_10-08_2016_1600.txt running-config
     ```

   - To copy a backup from the USB device, specify the **usb://** keyword and the name of the backup file.

     ```
     device# copy usb://backup-config_vdx_24-Aug_2016_1000.txt running-config
     ```

   - To copy a backup from an external host, specify the protocol, the ID and password, the **@** character, the URL, and the file name.

     ```
     device# copy ftp://admin:******@122.34.98.133//archive/backup-config_vdx30-Aug_2016_1100.txt
     running-config
     ```

2. If errors are reported, act accordingly.

Restoring the default configuration

You can reset the VCS running configuration to the factory defaults.

CAUTION
This procedure causes a cluster-wide reboot and returns the entire VCS cluster to the default configuration.

1. Enter the **copy default-config startup-config** command to overwrite the running configuration with the default configuration.

   ```
   device# copy default-config startup-config
   ```

2. Enter **Y** when prompted.

Using default configuration to avoid segmentation issues

Using the **default-config enable** command allows a node to rejoin a VCS by obtaining the configuration of the principal node in the fabric.

Enabling this feature can eliminate the following segmentation issues:

- A node rebooting while a global configuration change could not be applied.
- ISL connectivity to the primary node going down while configuration is in progress.

Extreme recommends not enabling this feature on *every* node (especially on core nodes) in the fabric.

The following restrictions apply:

- This command cannot be used on the principal node of the VCS fabric.
- Principal priority cannot be set on a node which has this feature enabled.

For information on how to use this command, refer to the **default-config enable** command in the *Network OS Command Reference*.

Backing up a running configuration

Although configuration changes that you make are immediately included and saved in the running configuration, you should also save a backup of the running configuration.

NOTE
Before upgrading or downgrading the firmware, use one of the following tasks to backup the running configuration.

Automated running-config backup and restoration

You can schedule or run a VCS running configuration backup by entering the **vcs auto-config-backup timer** command.

This feature automatically restores the last saved auto-config-backup under the following conditions:

- All cluster nodes reboot simultaneously, for example, upon power restoration after an unplanned outage.
- The configuration database is corrupted on all nodes, reverting to the default configuration.

To save a backup of the running configuration every 7 days:

```
device# configure terminal
device(config)# vcs auto-config-backup timer 7
```

To cancel automatic backup:

```
device# configure terminal
device(config)# no vcs auto-config-backup timer
```

To launch a manual backup:

```
device# configure terminal
device(config)# auto-config-backup
```

To display the time and trigger of the most recent backup (timer):

```
device# show vcs auto-config-backup
Last Backup Time: 2016-08-03 20:51:11
Initiated By: timer
```

To display the time and trigger of the most recent backup (triggered by user):

```
device# show vcs auto-config-backup
Last Backup Time: 2016-08-03 21:32:02
Initiated By: admin (user)
```

Backing up running-config to flash

You can save a running configuration backup file to the device flash memory.

1. In privileged EXEC mode, enter the **copy running-config flash** command, specifying a name for the backup file.

```
device# copy running-config flash://backup-config_vdx_10-08_2016_1600.txt
```

NOTE
You do not need to specify a file extension for the backup file.

2. To verify the transaction, enter the **dir** command.

```
device# dir
total 32
drwxr-xr-x   2 root     sys          4096 Feb 17 17:50 .
drwxr-xr-x   3 root     root         4096 Jan  1  1970 ..
-rwxr-xr-x   1 root     sys           417 Oct 12  2010 defaultconfig.novcs
-rwxr-xr-x   1 root     sys           697 Oct 12  2010 defaultconfig.vcs
-rw-r--r--   1 root     root         6777 Aug 10 17:50 backup-config_vdx_10-Aug_2016_1600.txt
```

Backing up running-config to USB

You can save a running configuration backup file a USB device.

ATTENTION
The only supported USB stick for this task is the one shipped with the Extreme device.

1. Insert the USB stick into the device USB port.
2. In privileged EXEC mode, enter the **usb on** command.

```
device# usb on
USB storage enabled
```

3. Enter the **copy running-config usb** *destination_file* command, specifying the file name.

```
device# copy running-config usb://backup-config_vdx_24-Aug_2016_1000.txt
```

Backing up running-config to an external host

You can upload a running configuration backup file to a network host.

In privileged EXEC mode, enter the **copy running-config** *destination_file* command, specifying the protocol, the ID and password, the **@** character, the URL, and the file name.

```
device# copy running-config ftp://admin:******@122.34.98.133//archive/backup-config_vdx30-Aug_2016_1100.txt
```

Managing configurations on a modular chassis

The configuration data on a modular chassis are managed in a distributed fashion.

The Extreme VDX 8770-4 and VDX 8770-8 chassis maintain two types of configuration data, global configuration parameters and slot configuration parameters. The global configuration, such as the VLAN configuration, applies to the entire chassis. The slot configuration includes specific parameters that apply only to the line cards.

Managing configurations on line cards

When an line card (interface module) boots up in a slot which was never occupied previously or is not configured, the module type is automatically saved in the configuration database. The type configuration associated with a given slot persists in the database even after the line card is physically removed, powered off, or faulted. This mechanism ensures that all configuration data associated with a given slot is automatically preserved across reboots or hot swaps with the same type of line card.

If you insert an line card in a slot that was previously occupied by a module of a different type, the line card will be faulted with a "type mismatch" error. Before you replace an line card with a different type, you must clear the existing type configuration from the database. Refer to Replacing a line card on page 83 for more information.

Managing configurations across redundant management modules

In modular switches with redundant management modules, configurations are synchronized and shared between the two management modules.

The initial configuration synchronization occurs when the system boots up. After the initial synchronization has been completed successfully, synchronization can be triggered during the following events:

- When a failover occurs from the active management module to the standby management module.
- When you insert a standby management module into a chassis after the active management module is already fully initialized.
- When you restore the default configuration by issuing the **copy default-config startup-config** command on the active management module.
- When you change the VCS configuration (RBridge ID or VCS ID), the configuration change is synchronized with the standby management module and saved persistently. This event triggers a chassis reboot after the synchronization is complete.
- When you initiate a firmware download. Refer to Configuration Fundamentals on page 37 for more information.

Managing flash files

Network OS provides a set of tools for removing, renaming, and displaying files you create in the switch flash memory.

You can use the display commands with any file, including the system configuration files. The **rename** and **delete** commands only apply to copies of configuration files you create in the flash memory. You cannot rename or delete any of the system configuration files.

Listing the contents of the flash memory

To list the contents of the flash memory, enter the **dir** command in privileged EXEC mode.

```
device# dir
drwxr-xr-x   2 root     sys          4096 Feb 13 00:39 .
drwxr-xr-x   3 root     root         4096 Jan  1  1970 ..
-rwxr-xr-x   1 root     sys           417 Oct 12  2010 defaultconfig.novcs
-rwxr-xr-x   1 root     sys           697 Oct 12  2010 defaultconfig.vcs
```

Viewing the contents of a file in the flash memory

To investigate the contents of a file in the flash memory, enter the **show file** command in privileged EXEC mode.

```
device# show file defaultconfig.novcs
!
no protocol spanning-tree
!
vlan dot1q tag native
!
cee-map default
remap fabric-priority priority 0
remap lossless-priority priority 0
priority-group-table 1 weight 40 pfc on
priority-group-table 2 weight 60 pfc off
priority-group-table 15.0 pfc off
priority-table 2 2 2 1 2 2 2 15.0
```

```
!
interface Vlan 1
shutdown
!
port-profile default
vlan-profile
 switchport
 switchport mode trunk
 switchport trunk allowed vlan all
!
protocol lldp
!
end
!
```

Deleting a file from the flash memory

To delete a file from the flash memory, enter the **delete** *file* command in privileged EXEC mode.

```
device# delete myconfig
```

Renaming a flash memory file

To rename a file in the flash memory, enter the **rename** command in privileged EXEC mode.

```
device# rename myconfig myconfig_20101010
```

DHCP management

The Dynamic Host Configuration Protocol (DHCP) enables DHCP servers to pass configuration parameters such as IPv4 addresses to IPv4 hosts.

For further details on DHCP implementation, refer to the *Extreme Network OS Layer 3 Routing Configuration Guide*.

VDX 6740T as a DHCP server

The Extreme VDX 6740T device can be used as a DHCP server.

You can connect the Extreme VDX 6740T to VDX 6740 switches and use it as a DHCP server to support features such as the DHCP Automatic Deployment (DAD), DHCP zero touch provisioning (ZTP), obtaining switch IP addresses, and other services on Extreme VDX 6740 DHCP clients. Any Extreme device can be used as a DHCP client. Refer to the *Network OS VDX 6740 Hardware Installation Guide* for information on hardware connectivity for this functionality.

Configuration considerations

The following configuration considerations apply to setting up the VDX 6740T as a DHCP server:

- DHCP relay must be disabled, because the DHCP server uses the same port number.
- When the DHCP server is enabled, the CLI to change the management interface IP address is blocked. You must first disable the DHCP server to change the management interface IP address.
- Enabling or disabling the DHCP server will automatically toggle the DHCP relay and force a switch reboot.
- A valid dhcpd.conf file must be copied onto the device before enabling the DHCP server. The dhcpd.conf file can be edited with the command line interface, however, you cannot edit the dhcpd.conf file on the VDX 6740T device. Any required changes should be made outside the VDX 6740T device and then copied back.

- If the dhcpd.conf file is invalid, the DHCP server will get disabled.
- The device configuration upload or download will not include the dhcpd.conf file.
- If DHCP server has been enabled and running, you can update the DHCP configuration file and restart the DHCP server from the command line interface.
- The management interface must be configured with a valid static IP address for the DHCP server to function. If you change the management interface to an incompatible IP address or DHCP IP address, the DHCP server stops.
- The DHCP server only runs on the active partition. HA failover is supported.
- For the DHCP server to function, the looping management interface to the inband port must be present.
- Because the configuration upload does not include the dhcpd.conf file, when you perform a configuration download to the new device, the DHCP server will be disabled automatically when you perform a configuration download during device replacement. You will need to again download the dhcpd.conf file to the new device manually.
- DHCPv6 is not supported.
- DHCP server supports up to 20 clients.
- DHCP configuration with multiple subnets is not supported.
- DHCP server must run on a standalone device. You cannot enable DHCP server when the device is part of a VCS cluster.
- You must have root access to the device to debug DHCP server by checking /var/log/dhcpd.leases and /var/log/dhcplog.
- To enable DHCP server debug log, dhcpd.conf must use the “log-facility user”.
- A looping management interface to inband port support is required for the DHCP server functionality.
- DHCP server 4.1.1-P1 is installed as part of the Network OS 7.1.0 firmware.
- DHCP server support on inband ports without IP loopback and DHCP server on management port is not supported.

Management interface IP address change

When the DHCP server is enabled, the CLI to change the management interface will become blocked. You will be prompted to disable the DHCP server first. Static IPv4 address change on the management interface will also be blocked.

HA support for DHCP leases

DHCP leases manage the IP addresses assigned to the DHCP clients. The dhcpd.conf file must include an event handler “on commit”, “on release” and “on expiry”. Whenever the DHCP server assigns an IP address to the client, it calls /fabos/cliexec/dhcpevent to trigger the synchronization of the lease database. Otherwise, when HA failover occurs on the DHCP server, the DHCP client renew may not obtain the same IP address.

dhcpevent is a shell script. It prints all the command line parameters into the /var/log/dhcplog and triggers the synchronization of the lease database.

DHCP server debug log

The DHCP server debug log can be enabled by adding the “log-facility user”. This maps “user” to the /var/log/dhcplog.

Setting up the VDX 6740T as a DHCP server

The following steps describe setting up the VDX 6740T as a DHCP server.

1. Connect the switch management interface to an inband port with a CAT5 ethernet cable.

2. Set up the DHCP pool address and other parameters in the dhcp.conf file before copying the configuration file.

```
option routers 10.1.1.2;
        option subnet-mask 255.255.255.0;
        range dynamic-bootp 10.1.1.50 10.1.1.100;
        default-lease-time 1200;
        max-lease-time 7200;
        option broadcast-address 10.1.1.255;
```

3. Set up the static IP address on the management interface. Note that the IP address, netmask and routers should be compatible with dhcp.conf file.

```
device# configure
Entering configuration mode terminal
(config)# rbridge-id 1
(config-rbridge-id-1)# vrf mgmt-vrf
(config-vrf-mgmt-vrf)# address-family ipv4 unicast
(vrf-mgmt-vrf-ipv4-unicast)# ip route 0.0.0.0/0 192.166.0.1
(vrf-mgmt-vrf-ipv4-unicast)#
(config)# interface Management 1/0
(config-Management-1/0)# ip address 192.166.0.2/24
(config-Management-1/0)# end
```

4. Configure the required inband ports to join the VLAN.

```
(config)# interface vlan 100 (vlan number can be any value)
(config)# interface TenGigabitEthernet 1/0/1
(conf-if-te-1/0/1)# speed 1000
(conf-if-te-1/0/1)# switchport
(conf-if-te-1/0/1)# switchport mode access
(conf-if-te-1/0/1)# switchport access vlan 100
```

5. Download the user dhcp.conf file onto the device. The destination file name must be dhcp.conf.

```
copy scp://<username>:<password>@hostname/<filepath> flash://dhcp.conf
```

   If the DHCP server is enabled and running, you can restart the DHCP server using the command **dhcpd restart** to make the new configuration take effect.

6. Use the **dhcpd enable** command

   The **dhcpd enable** command is persistent between reboot and HA failover.

Example dhcpd.conf file

The following is a sample DHCP configuration file defining a subnet. The debug log is located at /var/log/dhcplog.

```
default-lease-time 60000;
max-lease-time 72000;
authoritative;
# enable dhcpd debug log
log-facility user;
# single subnet address pool
subnet 192.169.0.0 netmask 255.255.255.0 {
range 192.169.0.210 192.169.0.220;
option routers 192.169.0.3;
option subnet-mask 255.255.255.0;
option broadcast-address 192.169.0.255;
# commit, release and expiry are mandatory to support HA failover for/var/log/dhcpd.leases
on commit {
set ipaddr = binary-to-ascii(10, 8, ".", leased-address);
set macaddr = binary-to-ascii(16, 8, ":", substring(hardware, 1, 6));
# mandatory, it's ok to execute dhcpevent without any parameters
execute("/fabos/cliexec/dhcpevent", "commit", ipaddr, macaddr);
}
on release {
set ipaddr = binary-to-ascii(10, 8, ".", leased-address);
set macaddr = binary-to-ascii(16, 8, ":", substring(hardware, 1, 6));
# mandatory, it's ok to execute dhcpevent without any parameters
execute("/fabos/cliexec/dhcpevent", "release", ipaddr, macaddr);
}
on expiry {
set ipaddr = binary-to-ascii(10, 8, ".", leased-address);
set macaddr = binary-to-ascii(16, 8, ":", substring(hardware, 1, 6));
# mandatory, it's ok to execute dhcpevent without any parameters
execute("/fabos/cliexec/dhcpevent", "expiry", ipaddr, macaddr);
```

Using DHCP Automatic Deployment

DHCP Automatic Deployment (DAD) is a method used to bring up the switch with new firmware or a preset configuration automatically.

You can automatically bring up a switch with new firmware, with a preset or default configuration, omitting the need for logging in to the switch console to configure the switch. You must be using DHCP to use DAD. The DHCP process helps retrieve certain parameters (for example, the firmware path, VCS ID, VCS mode, RBridge ID, and preset configuration file) needed by the DAD process to perform the firmware and configuration downloads. For node replacement, the switch is set to the default configuration.

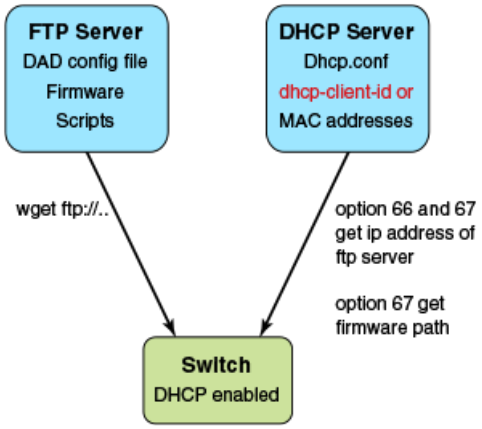
You must enable DAD from the CLI, after which the switch is rebooted automatically. After the DAD process is triggered and completed, DAD is automatically disabled. If you attempt to download new firmware that is already installed on the switch, the DAD process skips the firmware download step and continues configuring the switch.

> NOTE
> If DAD is enabled, you are warned when initiating a firmware download from the CLI that the firmware download will be unsuccessful. After a firmware download begins, DAD reports the firmware download success or failure status.

You can bring up a cluster of switches using a Python script (dad.py) for DAD. Zero Touch Provisioning (ZTP) deployment is also supported beginning with this release. When the switch is in a factory default configuration and the DHCP server for auto deployment is reachable, auto deployment is enabled automatically when the switch powers up. You can also enable DAD from the CLI, after which the switch is rebooted automatically.

> NOTE
> For more information on using Python with Network OS, refer to Python Event-Management and Scripting on page 231.

FIGURE 11 DHCP and FTP server environment for DAD



Supported DAD use cases

DAD supports the following typical use cases:

- Replacing a switch in a cluster by upgrading the firmware and setting up the switch to a preset configuration. In this instance, DAD must be completed on the new switch hardware (to update the firmware) before the new switch can be incorporated into the cluster.
- Configuring the whole cluster using a Python script.
- Zero Touch Provisioning (ZTP) support for IP fabrics with a single DAD configuration file for all nodes.

Example DHCP and DAD configuration files

The following sample configuration files provide a reference for DHCP and DAD deployment environments.

The switches of a cluster are grouped together and each cluster should have its own group of switches. Each switch is added into dhcp.conf file with control parameters, such as the FTP server IP address and path to the DAD configuration file of the switch. Chassis systems support only dhcp-client-identifier, not MAC address. The following is an example file.

> NOTE
> Starting with Network OS release 7.0.0, the DHCP client identifier format in the DHCP server configuration file has changed.

In order to support firmware upgrade or downgrade between Network OS 6.0.x to 7.0.0 and later, you must update the /etc/dhcd.conf file as follows.

```
group {
  option bootfile-name "/config/unified-cfg"; # DAD config file contains settings for all switches
  option tftp-server-name "192.168.0.2";


  host cp0_new {
    option dhcp-client-identifier = "BROCADE##VDX6740##CPL2517K0FZ"; # serial number
    hardware ethernet 50:EB:1A:12:61:F3;    # mac address for cp0
    fixed-address 192.168.0.88;         # fixed ip address

  }
  host cp1_new {
    option dhcp-client-identifier = " BROCADE##VDX6740##CPL2517K0FZ "; # serial number
    hardware ethernet 50:EB:1A:12:61:F4;    # mac address for cp1
    fixed-address 192.168.0.88;         # fixed ip address
```

```
    }


    host device_chassis_new {
      option dhcp-client-identifier = " BROCADE##VDX6740##CPL2517K0FZ "; # serial number
      fixed-address 192.168.0.87;         # fixed ip address
    }
  }
```

DHCP server configuration file example

The following is an example of the dhcp.conf file.

```
#ddns-update-style standard;
ddns-update-style interim;
ddns-ttl 600;
ignore client-updates; # Overwrite client configured FQHNs
ddns-domainname "infralab.com.";
ddns-rev-domainname "in-addr.arpa.";

option ntp-servers 192.168.0.2;
option domain-name-servers 192.168.0.2;
option domain-name "infralab.com";
option domain-search "infralab.com";

default-lease-time 600;
max-lease-time 7200;

authoritative;

log-facility local7;

key "rndc-key" {
    algorithm hmac-md5;
    secret "dtBgNTAoqZmwV5c4SueybjOvhe6OIqgac1uQrzGBv5O4X4nIEBEEGWRf0lCnbFhuIJXGExNBjDdNSqgBMeNI8w==";
};

subnet 192.168.0.0 netmask 255.255.255.0 {
  range 192.168.0.100 192.168.0.200;
  option subnet-mask 255.255.255.0;
  option broadcast-address 192.168.0.255;
  zone 0.168.192.in-addr.arpa. {
        primary 192.168.0.2;
        key "rndc-key";
  }
  zone infralab.com. {
        primary 192.168.0.2;
        key "rndc-key";
  }
}
# cluster switches
group{
  option bootfile-name "/config/unified-cfg.min";
  option tftp-server-name "192.168.0.2";
  option routers 192.168.0.2;
  # device 82
  host host10 {
    option dhcp-client-identifier = "BROCADE##VDX6740##CGS0333H00R";
    hardware ethernet 00:05:33:E5:85:37;
     fixed-address 192.168.0.82;         # fixed ip address
 }}
  host host11 {
    option dhcp-client-identifier = "BROCADE##VDX6740##CGS0333H00R";
    hardware ethernet 00:05:33:E5:85:38;
    fixed-address 192.168.0.82;         # fixed ip address
 }

  # device 90
  host host12 {
    option dhcp-client-identifier = "BROCADE##VDX6740##CPL2549J0EG";
```

```
    hardware ethernet 00:27:F8:D2:CE:5F;
    fixed-address 192.168.0.90;         # fixed ip address
  }
   host host13 {
    option dhcp-client-identifier = "BROCADE##VDX6740##CPL2549J0EG";
    hardware ethernet 00:27:F8:D2:CE:5E;
    fixed-address 192.168.0.90;         # fixed ip address
  }

}
```

DAD configuration file example

The MAC address for Extreme VDX devices can be found on the pullout tab on the front-left side of the device.

FIGURE 12 MAC address pullout tab

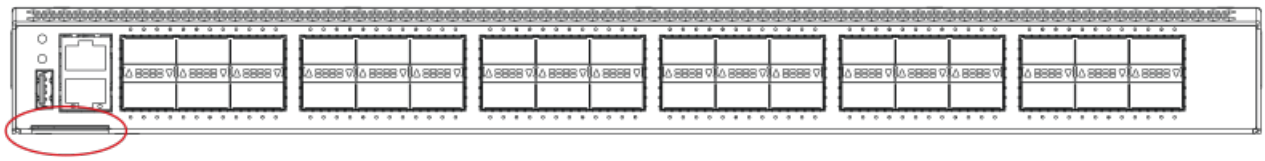
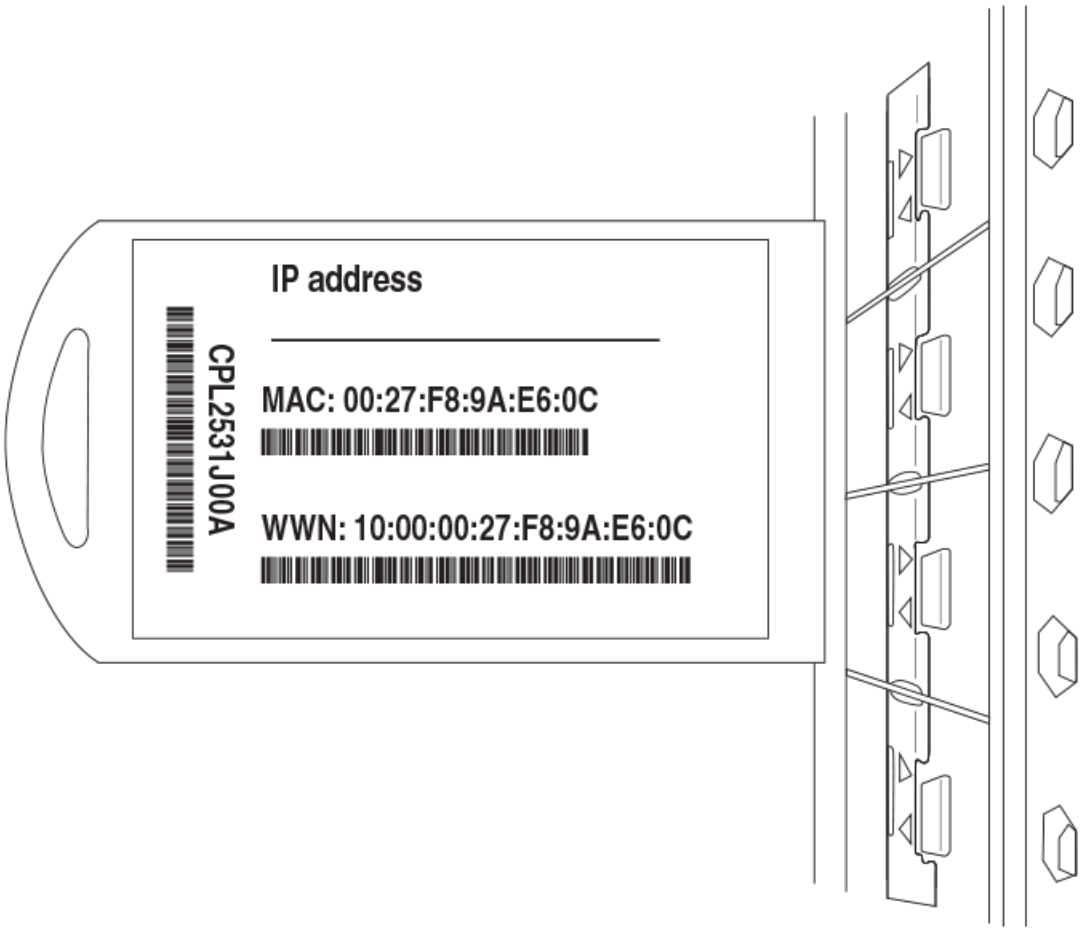


FIGURE 13 MAC address pullout tab detail



The following is an example of a DAD configuration file.

```
version=3
date=10/22/2014
supported_nos=7.0.1

common_begin
vcsmode=LC
vcsid=2
rbridgeid=1
fwdir=/firmware/nos7.0.1
scriptcfgflag=1
principlerbridgeid=1
startup=/config/rbridge.cfg (shared by all switches for dhcp enable setting)
script=/script/dad.py
# 50 minute
vcstimeout=50
# 3 days = 4320 min
dadtimeout=4320
common_end

  #host(device 82)
  host_mac=00:05:33:E5:85:37
  host_mac=00:05:33:E5:85:38
  host_sn=CGS0333H00R
  rbridgeid=82
  defaultconfig=no
  startup=/config/mc/rbridge-82.cfg
  globalcfg=/config/mc/global.cfg
  host_end

  #host(mm 145)
  host_mac=00:27:F8:46:E7:D9
  host_mac=00:27:F8:38:6C:2C
  host_sn=CDU2539H00L

  fwdir=/min/bld/bld_dadm
  rbridgeid=145
  defaultconfig=no
  startup=/config/mc/rbridge-145.cfg
  globalcfg=/config/mc/global.cfg
  host_end

  #host(device 90)
  host_mac=00:27:F8:D2:CE:5E
  host_mac=00:27:F8:D2:CE:5F
  host_sn=CPL2549J0EG

  rbridgeid=90
  startup=/config/mc/rbridge-90.cfg
  globalcfg=/config/mc/global.cfg
  defaultconfig=no
  host_end

  #host(device 91)
  host_mac=00:27:F8:DC:1E:6B
  host_mac=00:27:F8:DC:1E:6D
  host_sn=CPL2501K01G

  rbridgeid=91
  startup=/config/mc/rbridge-91.cfg
  globalcfg=/config/mc/global.cfg
  defaultconfig=no
  host_end
```

TABLE 6 DAD configuration file definitions

Variable	Description
version	Version 2 supports from Network OS 5.0.0 to Network OS 6.0.0. Version 3 supports Network OS 6.0.1+.
date	The last modification date.
supported_nos	The firmware version.
host_mac, host_sn, host_end	The MAC addresses or serial number of the switch. "host_mac" or "host_sn" mark the beginning of the configuration section for a switch. The section ends with "host_end". NOTE You must confirm the serial number on the switch with Brocade before adding the MAC address or serial number.
vcsmod	VCS mode: From Network OS 7.1.0, the only option is "LC" (logical chassis cluster mode). Fabric cluster mode is no longer supported.
vcsid	The cluster VCS ID. The range is 1 to 8192.
rbridgeid	The unique ID for each node. When commented out with # sign, for example, "#rbridgeid=5", DAD is disabled for the node. If the variable *startup* is set, the value should match the configuration file set at "startup". If not, the DAD sanity check will fail. The range is from 1 to 239. You should change the default settings of vcsid=1 and rbridgeid=1.
principalrbridgeid	Principal node rbridige id of the cluster. The designated node will become the principal node of the cluster. Depending on the topology of the cluster, ideally the principal node should have direct connection to most of secondary nodes. This is ignored in ZTP mode.
defaultconfig	Using the **defaultconfig** setting resets all the user configured settings to the factory defaults. If set to *yes*, the script, globalconfig, and number of nodes will be ignored and the DAD process is for single node only, not for the cluster.
startup	The local configuration file. The value can be *default* or *configuration file*. If the variable *defaultconfig* is set to *yes*, *startup* should be commented out or set empty. If not, the DAD sanity check will fail. • For Network OS 6.0.0 and prior releases, this means the switch running configuration. It must be uploaded from each switch with the **copy running-config ftp** command. • For Network OS 6.0.1 and later releases, this means the local configuration only.
globalcfg	The cluster global configuration file. This is only set for the DAD principal node and will be ignored by secondary nodes. This should be either set empty or a valid configuration file path must be specified. This is created using the **copy global-running-config ftp** command.
scriptcfgflag	**defaultconfig** must be set to *no*. This is not a mandatory variable. The default is 0, when not specified. The meaning of the value is: 0 - only use startup and globalcfg, script is ignored 1 - only use script, startup and globalcfg are ignored
script	The set up script file.
ztp	ZTP mode: 1 - enabled (default), 0 - disabled. When enabled, it changes the behavior of DAD. Only the settings in the "common" section are used. The "host" section is ignored. vcsmode=LC will be forced. These settings in the configuration file will be ignored.
fwdir	Firmware path in FTP server. If set to empty, DAD will skip firmware download.
common_begin, common_end	The setting in the section will be shared by all switches.

Configuring the DAD process for replacing data center fabric switches

Use the following procedure to configure DHCP Automatic Deployment (DAD) when replacing data center fabric switches.

The DAD process applies to a new switch and the principal switch.

1. Disconnect the existing switch from the cluster.
2. Connect the new switch to the cluster. The new switch must be the same model and use the same cable connection as the old switch.
   The new switch must successfully load Network OS. Note, however, that the new switch cannot join the cluster just yet.
3. From the principal switch, manually run the node replacement with the RBridge ID.
4. Establish a DAD environment for the new switch. (Make sure DHCP is enabled on the management interface.)
   - The management interface of the switch must be set up as DHCP.
   - The DHCP server must have the FTP server IP address and configuration file path.
   - The configuration file is on the FTP server and it contains the firmware path, new configuration file path, VCS ID, VCS mode, and RBridge ID.
   - The DHCP server and FTP server must be up and running.
   - DAD must be enabled on the switch by means of the CLI.
5. Enable DAD by using the **dhcp auto-deployment enable** command, and enter **yes** when prompted to reboot the system. After the new switch is rebooted, the DAD process downloads the DAD configuration file to get the VCS mode, VCS ID, and RBridge ID. The RBridge ID must be configured the same as the previous node in the cluster. The DAD process sets the VCS ID and RBridge ID. No reboot is triggered. The DAD process invokes a firmware download if new firmware is detected. Firmware download completes successfully and the switch comes up with the new firmware and configuration settings. Note that the DAD process aborts if any error is detected. When the new switch comes up, it joins the cluster with the same configuration as the previous switch.
6. Use the **show dadstatus** command to view the current DAD configuration.

Considerations for enabling DHCP auto-deployment

The **dhcp auto-deployment enable** command causes a cold/disruptive reboot and requires that Telnet, secure Telnet, or SSH sessions be restarted.

1. Scenario 1: When you enable DAD and the system starts to reboot, the DAD process is triggered after configuration replay is complete.

   For dual management module (MM) chassis (VDX 8770 chassis), the DAD process waits for the dual MM to be in sync before starting the requested firmware download. However, if you manually issue **firmwaredownload - sb** during this period (after DAD is triggered and before MM is in sync), DAD fails because the previous firmware download takes precedence. If you manually issue **firmwaredownload -sb** before DAD is triggered, DAD will fail for the same reason.
2. Scenario 2: You issue the command to enable DAD (answer "Yes" when prompted), but before the system reboot, there is an HA failover. DAD will be cancelled. You must enable DAD from the new active switch.
3. Scenario 3: You issue the command to enable DAD, but after the system reboot is invoked, takeover occurs (the previous standby switch becomes the new active switch), DAD will proceed.
4. Scenario 4: You manually issue the **firmwaredownload** command, but before the firmware download is completed, you enable DAD from the CLI and answer "Yes" when prompted to reboot the switch. When the switch boots up, even if the DAD process detects that the firmware download is needed, it will fail during the sanity check because the previous incomplete firmware download takes precedence. DAD fails.
5. Scenario 5: If you do not power down the secondary node when running DAD on the principal node, the following outcomes are observed:
   - The DAD principal node and secondary node form the cluster, but without the principal role. DAD fails on the principal node.
   - The DAD principal node and secondary node form the cluster with the principal role. DAD proceeds on the principal node because the secondary node does not affect DAD.
   - The DAD principal node and secondary node do not form the cluster. DAD proceeds on the principal node because the secondary node does not affect DAD.

DAD configuration considerations

Note the following general considerations when using DAD:

- The DAD process is disruptive to traffic.
- Currently, only DHCPv4 is supported.
- Configuration download is not supported during a firmware downgrade.
- Configurations are not downloaded if the DAD process is aborted due to a sanity check failure, or if you are downloading the same firmware version.
- If an existing firmware download session is either occurring or paused, such as during a firmware commit, DAD is not triggered. Instead, the last firmware download session continues. If a firmware download is in progress and you attempt to enable DAD, you are prompted to try again later.
- In-Service Software Upgrade (ISSU) is not supported at this time.
- For dual management module (MM) chassis (VDX 8770 chassis), the dual MM must be in sync from the chassis bootup (not from HA failover). In a chassis system, both MMs are rebooted at the same time.
- DAD is not supported in FIPS mode.
- You can choose to apply either the default configuration or the preset configuration.
- DAD must be enabled on the principal node first. All the other nodes must be enabled in parallel.

- The preset configuration can be either a configuration file or a running setup script.
- Cluster principal node failover is not supported.
- Adding a node into an existing cluster is not supported.
- DAD over in-band is not supported.
- Virtual Fabrics is not supported with DAD. You must disable Virtual Fabrics before starting the DAD process in the global configuration file or in the script.
- If switches are already configured with virtual fabric, this configuration will not be changed by the DAD process.

DAD using the Python script

The Python script dad.py allows you to configure a VCS fabric with preset configuration files.

This Python script configuration method reduces the time taken for firmware download, switch configuration, and fabric formation. Because all switches download the same firmware and the cluster parameters remain consistent across all switches, using the Python script is a more scalable method for deployment and reduces potential human error.

Once you enable DAD on the switch, the DAD process calls the dad.py script from the FTP server. The dad.py script starts running whenever the number of nodes detected in the fabric equals the number of nodes that have joined the cluster. The DAD process monitors the nodes.

Python script support considerations for DAD

- The **reboot** command is not supported in the Python script.
- If the switch has a preset local configuration, the script will overwrite it.
- The start and completion times of the script will be reported, although script internal errors will not be reported. You must check the internal errors manually.
- If you change the DAD configuration script on any node in the cluster, the switch cluster must run the DAD process again.
- The cluster DAD process has a time limit (three days) to run the script. If the number of nodes in the cluster is not equal to what is specified in the DAD configuration file, the principal node declares DAD partially complete after timeout. If the deployment fails on the secondary node, you must restart the deployment again before the principal node deployment times out.
- The global configuration replay takes place in parallel with firmware download. The global configuration file may take a long time to execute if it is big. However, the DAD status will display as complete when firmware download completes. You must manually check if the configuration file is executed completely. If the configuration file is not fully executed, the DAD status still remains complete and not failed.
- If the secondary node triggers a firmware download, the principal node will trigger the Python script once the secondary node joins the cluster. The firmware download on the secondary node continues in parallel and may complete after the principal node declares DAD complete.

DAD on the principal node

The following steps describe the procedure for DAD enabled by the dad.py Python script. The dad.py Python script runs once you enable DAD on the switch.

NOTE

Refer to Script for upgrading and reconfiguring switches (dad.py) on page 240 for more information on the dad.py script.

The following prerequisites apply to the DAD process.

- Set up the FTP server directory structure containing the firmware, Python script or the switch configuration file.
- Create the DAD configuration file for the cluster, such as the VCS mode, VCS ID, RBridge ID etc. Then upload the file to the FTP server.
- In the DAD configuration file, for the RBridge ID, you should change the values from the default settings of VCS ID=1 and RBridge ID=1. The globalcfg should be set only for the DAD principal node, and left empty for the secondary nodes.
- Set up the DHCP server configuration to assign fixed IP addresses to each switch and DAD configuration file for each switch.

The principal node completes the DAD as a standalone node first.

1. Power down all the secondary nodes in the cluster.
2. Enable DHCP.
3. Enable DAD using the **dhcp auto-deployment enable** command.

Once you enable DAD on the switch, the following process begins. The run order of the script is 3, 1, 2 on the principal node and on the secondary node it is 3 only.

1. global - Run cluster global configuration commands from the principal node.

2. rbridge-id 2 - Configuration commands for a specific node run from the principal node.

3. rbridge-id 2 - local - Configuration commands allowed to be run from the local node.

Check on the deployment progress until the process is complete. If any of the switches fail the process, you will need to run the process again before the principal node deployment times out.

DAD on the secondary node

Once the principal node deployment completes successfully, the DAD process starts on the secondary node.

1. Enable DHCP. Make sure the local time of the switch is in sync with the principal node.
2. Enable DAD using the **dhcp auto-deployment enable** command.

Once you enable DAD on the switch the DAD process begins. The script runs on the local configuration (--rbridgeid # --local).

> NOTE
> Use the **show dadstatus** command to view the status of the DAD process as it runs.

DHCP zero touch provisioning

Zero touch provisioning (ZTP) allows you to automatically start up the device with the correct firmware and configuration as soon as you plug into the network.

The following configuration considerations apply to DAD in a ZTP environment.

- Single DAD configuration file per IP fabric.
- ZTP is supported only on out-of-band ports.
- In an IP fabric, all nodes will have the same VCS ID and RBridge ID.
- A VCS cluster is not formed among any nodes.
- You must set the IP address for each switch in the DCMD configuration file with "startup" setting in the DAD configuration file or with the script.
- All nodes can either be powered up at the same time or enabled from the CLI.

- DAD executes only if the switch configuration is the default configuration. If the configuration on the switch is not the default configuration, DAD exits.
- If the switch is in the default configuration before DAD is triggered, DHCP will remain enabled after the deployment completes. However, this setting can be overwritten by the switch-specific configuration file or the dad.py script.
- The factory default DAD runs only once in a DHCP-enabled environment. Irrespective of whether this process is a success or failure, DAD will not be triggered again after a reboot or power off. You can run DAD manually using the **dhcp auto-deployment enable** command if required.
- DAD is enabled automatically upon switch reboot when you use the **write erase** command.

Using zero touch provisioning

The following steps describe the DHCP zero touch provisioning deployment process.

You must first adhere to these pre-requisites for the zero touch provisioning deployment process.

- Do not use vcsid=1. This is preserved for the factory default.
- DHCP is enabled from the startup configuration by setting the "startup" value.
- The host specific section will be ignored.
- Set the value "ztp=1" in the DAD configuration file.
- Set the value scriptcfgflag=1 when the script is configured; set the value scriptcfgflag=0 when script is not configured.
- In scenarios where zero touch provisioning is not used, you must mandatorily set "ztp=0", because ZTP is enabled by default if you do not specify this value.
- When using the script, the script should contain the IP address (static IP address with DHCP gateway) set up on the management interface.

1. Power up the switch, or run the **write erase** command from the CLI prompt.
2. When the switch boots up, check if the DHCP server is available. If the DHCP server is unavailable, the ZTP process aborts without reporting an error.
3. Disable the bare metal mode on the switch.
4. Enable DHCP once the switch reboots.
   Once you enable DHCP, the zero touch provisioning process starts. The dad.py script runs on the global and local configuration for zero touch provisioning on the node.

Session connections

You can connect to the switch through a console session on the serial port, by SSH, or by Telnet.

The switch must be physically connected to the network. If the switch network interface is not configured or the switch has been disconnected from the network, use a console session on the serial port.

You can connect to your switch through a console session on the serial port. Also, you can use SSH or Telnet to connect to the management port. You can also use SSH and Telnet to connect to an IP inband interface configured on an Ethernet port, a VE interface, or loopback in the mgmt-vrf (management VRF) or default-vrf. You can use any account login present in the local switch database or on a configured authentication, authorization, and accounting (AAA) server for authentication. For initial setup procedures, use the preconfigured administrative account that is part of the default switch configuration.

- Refer to the Extreme VDX hardware reference manuals for information on connecting through the serial port.
- Refer to Configuring Ethernet management interfaces on page 69 for details on configuring the management interface.

Telnet and SSH

Telnet and Secure Shell (SSH) are mechanisms for allowing secure access to management functions on a remote networking device. SSH provides a function similar to Telnet, but unlike Telnet, which offers no security, SSH provides a secure, encrypted connection to the device.

SSH and Telnet support is available in privileged EXEC mode on all Extreme VDX platforms. Both IPv4 and IPv6 addresses are supported.

Telnet and SSH services are enabled by default on the switch. When the Telnet server or SSH server is disabled, access to the switch is not allowed for inbound Telnet or SSH connections, thereby restricting remote access to the switch.

Network OS supports up to 32 CLI sessions on a switch.

In configuration mode, the CLI can be used to disable Telnet or SSH service on the switch. Doing so will terminate existing inbound Telnet or SSH connections and block any new inbound Telnet or SSH connections to the switch. Additional inbound Telnet or SSH connections will not be allowed until the Telnet server or SSH server is re-enabled. If you have admin privileges, you can re-enable inbound Telnet or SSH connections from configuration mode.

The command for enabling or disabling Telnet or SSH services is not distributed across the fabric. The RBridge ID of the node should be used to configure the service on individual nodes.

In operational mode, you can use the **show** command to display whether Telnet or SSH is enabled or disabled on the switch.

NOTE
Telnet alone is not supported on the Extreme VDX 2746. Telnet port 23 is denied by default.

SSH server key exchange and authentication

The Secure Sockets Layer (SSL) protocol allows users to authenticate using public and private key pairs instead of passwords. In password-based authentication, the user must enter a password for authentication purposes. In public-key authentication, the user should have a private key in the local machine and a public key in the remote machine. The user should be logged in to the local machine to be authenticated. If a passphrase is provided while generating the public and private key pair, it must be entered to decrypt the private key while getting authenticated.

Secure Shell (SSH) key-exchange specifies the method used for generating the one-time session keys for encryption and authentication with the SSH server. A user is allowed to configure the SSH server key-exchange method to DH Group 14. When the SSH server key-exchange method is configured to DH Group 14, the SSH connection from a remote SSH client is allowed only if the key-exchange method at the client is also configured to DH Group 14.

NOTE
For complete information on SSH configuration, refer to the *Extreme Network OS Security Configuration Guide*.

The following steps briefly describe public-key authentication:

1. The user generates a pair of encryption keys in a local machine using the **ssh-keygen** command, along with the public and private key. Messages encrypted with the private key can only be decrypted by the public key, and vice-versa.

```
device# ssh-keygen -t rsa
 generates RSA public and private keypair
device# ssh-keygen -t dsa
 generates DSA public and private keypair
```

2. The user keeps the private key on the local machine, and uploads the public key to the switch.

3. When attempting to log in to the remote host, the user receives an encrypted message from the remote host containing the public key. After the message is decrypted in the local host by means of the private key, the user is authenticated and granted access.

The **ssh-keygen** command is not distributed across the cluster. The RBridge ID of the node should be used to configure service on individual nodes.

Feature support for Telnet

The following features are not supported with Telnet:

- Displaying Telnet sessions
- Terminating hung Telnet sessions

Feature support for SSH

SSHv2 is the supported version of SSH, but not all features typically available with SSHv2 are supported on the Extreme VDX family of switches.

The following encryption algorithms are supported for management:

- **3des**—Triple-DES (default)
- **aes256-cbc**—AES in CBC mode with 256-bit key
- **aes192-cbc**—AES in CBC mode with 192-bit key
- **aes128-cbc**—AES in CBC mode with 128-bit key
- **aes128-ctr**—AES in CTR mode with 128-bit key
- **aes192-ctr**—AES in CTR mode with 192-bit key
- **aes256-ctr**—AES in CTR mode with 256-bit key

The following Hash-based Message Authentication Code (HMAC) message authentication algorithms are supported for management:

- **hmac-md5**—MD5 encryption algorithm with 128-bit key (default)
- **hmac-md5-96**—MD5 encryption algorithm with 96-bit key
- **hmac-sha1**—SHA-1 encryption algorithm with 160-bit key
- **hmac-sha1-96:**—SHA-1 encryption algorithm with 96-bit key
- **hmac-sha2-256:**—SHA-2 encryption algorithm with 256-bit key
- **hmac-sha2-512:**—SHA-2 encryption algorithm with 512-bit key

SSH user authentication is performed with passwords stored on the device or on an external authentication, authorization, and accounting (AAA) server.

Support is provided for multiplexed SSH sessions. Refer to "Configuring the maximum number of SSH sessions" in the "Configuration Fundamentals" chapter.

The following features are not supported with SSH:

- Displaying SSH sessions
- Deleting stale SSH keys

For more information, refer to the "SSH - Secure Shell" chapter in the *Network OS Security Configuration Guide*.

NOTE
SHA-2 algorithms are supported only for NSX Controller deployments, by means of the **nsx-controller client-cert** command. Refer to the "VXLAN Overlay Gateways for NSX Controller Deployments" chapter in the *Network OS Layer 2 Switching Configuration Guide*.

Telnet and SSH considerations and limitations

- Outgoing Telnet or SSH connections from the switch to any remote device is not affected by disabling or enabling the Telnet or SSH server in the switch.
- No RASLog or auditlog messages are reported when the Telnet or SSH server is disabled or enabled.

Establishing a physical connection for a Telnet or SSH session

1. Connect through a serial port to the switch.
2. Verify that the switch's network interface is configured and that it is connected to the IP network through the RJ-45 Ethernet port.
3. Log off the switch's serial port.
4. From a management station, open a Telnet or SSH connection using the management IP address of the switch to which you want to connect.
5. Enter the password.

   Extreme recommends that you change the default account password when you log in for the first time. For more information on changing the default password, refer to the *Extreme VDX Hardware Reference* manuals.
6. Verify that the login was successful.

   The prompt displays the host name followed by a pound sign (#).

```
device# login as: admin

admin@10.20.49.112's password:******

-----------------------------------------------------
SECURITY WARNING: The default password for at least
one default account (root, admin and user) have not been changed.
Welcome to the Brocade Network Operating System Software
admin connected from 10.110.100.92 using ssh on VDX 6740-48
```

Establishing a Telnet connection

A Telnet session allows you to access a switch remotely using port 23. However, it is not secure. If you need a secure connection, use SSH.

1. To establish a Telnet session connection, enter **telnet** followed by the switch IP address.

```
device# telnet 10.17.37.157
```

   If the switch is active and the Telnet service is enabled on it, a display similar to the following will appear.

```
Trying 10.17.37.157...
Connected to 10.17.37.157.
Escape character is '^]'.
Network OS (device)
device# login:
```

2. Once you have established the Telnet connection, you can log in normally.

NOTE
You can override the default port by using the **telnet** *ip_address* command with the optional **port** operand (range 0-65535). However, the device must be listening on that port for the connection to succeed.

The following example overrides the default port.

```
device# telnet 10.17.37.157 87
Trying 10.17.37.157...
Connected to 10.17.37.157.
Escape character is '^]'.
Network OS (device)
device# login:
```

Shutting down the Telnet service

Shutting down the Telnet service will forcibly disconnect all Telnet sessions running on a switch.

You must be in global configuration mode to shut down the Telnet service on a switch.

The Telnet service runs by default.

To shut down the Telnet service on a switch, enter **telnet server shutdown**. All Telnet sessions are immediately terminated, and cannot be re-established until the service is re-enabled.

```
device# telnet server shutdown
```

In a VCS Fabric, you must enter RBridge ID configuration mode before issuing the command.

```
device# rbridge-id 3
device(config-rbridge-id-3)# telnet server shutdown
```

Re-enabling the Telnet service

Re-enabling the Telnet service permits Telnet access to a switch.

You must be in global configuration mode to shut down the Telnet service on a switch.

To re-enable the Telnet service on a switch enter **no telnet server shutdown**.

```
device# no telnet server shutdown
```

NOTE
In a VCS Fabric, you must enter RBridge ID configuration mode before issuing the command.

```
device(config)# rbridge-id 3
device(config-rbridge-id-3)# no telnet server shutdown
```

Connecting with SSH

Connecting to a device using the SSH (Secure Socket Handling) protocol permits a secure (encrypted) connection.

For complete information on configuring SSH on the device, refer to the "SSH" chapter of the *Extreme Network OS Security Configuration Guide*.

Establishing an SSH connection

An SSH (Secure Socket Handling) connection allows you to securely access a switch remotely.

You must be in privileged EXEC mode to make an SSH connection to a switch.

1. To establish an SSH connection with default parameters, enter **ssh -l** followed by the *username* you want to use and the *ip_address* of the switch.

```
device# ssh -l admin 10.20.51.68
```

2. Enter **yes** if prompted.

```
The authenticity of host '10.20.51.68 (10.20.51.68)' can't be established.
RSA key fingerprint is ea:32:38:f7:76:b7:7d:23:dd:a7:25:99:e7:50:87:d0.
Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '10.20.51.68' (RSA) to the list of known hosts.
admin@10.20.51.68's password: ********

SECURITY WARNING: The default password for at least
one default account (root, admin and user) have not been changed.
Welcome to the Brocade Network Operating System Software

admin connected from 10.20.51.66 using ssh on C60_68F
```

NOTE

You can use the **-m** and **-c** options to override the default encryption and hash algorithms.

```
device# ssh -l admin -m hmac-md5 -c aes128-cbc 10.20.51.68
```

Configuring the maximum number of SSH sessions

An SSH server can reuse an already established TCP connection for multiple sessions (also known as multiplexing). This reduces the time required to open a new connection for each SSH session, as well as the overhead of allocating separate resources for each connection.

SSH clients must also be configured to support multiplexing, in accordance with local best practices.

Note the following additional usage guidelines.

After executing this command, in order to use the new number of sessions, you must first shut down the SSH server, by means of the **ssh server use-vrf shutdown** command, and then restart it, by means of the **no ssh server use-vrf shutdown** command.

The maximum number of sessions specified by this command is synchronized to the standby management module (MM). However, to make the change effective on the standby MM, you must first disable service on that module by means of the **no ssh server standby enable** command, and then reenable service by means of the **ssh server standby enable** command.

Use the **show running-config rbridge-id ssh server** command or the **show ssh server status** command to confirm the configuration.

A downgrade to a previous release is blocked if this command has been executed in the running configuration.

Use the **no ssh server max-sessions** command to revert to the default of 1 session. You must also stop and restart service as in the usage guidelines above.

1. From global configuration mode, enter RBridge ID configuration mode for a specified RBridge.

```
device# configure terminal
device(config)# rbridge-id 176
device(config-rbridge-id-176)#
```

2. Enter the **ssh server max-sessions** command and specify the maximum number of sessions to be supported. (Range is from 1 through 10.)
3. Use the **show running-config rbridge-id ssh server** command in this mode to confirm the running configuration, which includes key types as well as the maximum number of SSH sessions configured.

```
device(config-rbridge-id-176)# do show running-config rbridge-id ssh server
rbridge-id 176
ssh server max-sessions 7
ssh server key rsa 2048
ssh server key ecdsa 256
ssh server key dsa
```

4. You can also use the **show running-config rbridge-id ssh server** command in this mode to view the maximum number of SSH sessions configured, as well as VRF status.

```
device(config-rbridge-id-176)# do show ssh server status rbridge-id 176
rbridge-id 176:
VRF-name: mgmt-vrf       Status: Enabled
VRF-name: default-vrf    Status: Enabled
rbridge-id 176: SSH Server Max sessions: 7
```

Importing an SSH public key

Importing an SSH public key allows you to establish an authenticated login for a switch.

You must be in privileged EXEC mode to import an SSH public key to a switch.

1. To import an SSH public key, enter **certutil import sshkey**, followed by **user** *Username* **host** *IP_Address* **directory** *File_Path* **file** *Key_filename* **login** *Login_ID*.

```
device# certutil import sshkey user admin host 10.70.4.106 directory /users/home40/bmeenaks/.ssh
file id_rsa.pub login fvt
```

   This enables you to import the SSH public key for the user "admin" from a remote host.

2. Enter the password for the user.

```
Password: ***********

device# 2012/11/14-10:28:58, [SEC-3050], 75,, INFO, VDX6740-48, Event: sshutil, Status: success,
Info: Imported SSH public key from 10.70.4.106 for user 'admin'.
```

   NOTE
   In a VCS Fabric, you must enter RBridge ID configuration mode before issuing the command.

```
device# certutil import sshkey user admin host 10.70.4.106 directory /users/home40/bmeenaks/.ssh file
id_rsa.pub login fvt rbridge-id 3
```

Deleting an SSH public key

Deleting an SSH public key from a switch prevents it from being used for an authenticated login.

You must be in privileged EXEC mode to delete an SSH public key from a switch.

To delete an SSH public key, enter **no certutil sshkey user** *Username* followed by either **rbridge-id** *rbridge-id* or **rbridge-id all**.

```
device# no certutil sshkey user admin rbridge-id all
```

Specifying a specific RBridge ID removes the key from that RBridge ID; specifying all removes it from all RBridge IDs on the switch.

Shutting down the SSH service

Shutting down the SSH (Secure Socket Handling) service will forcibly disconnect all SSH sessions running on a switch.

You must be in global configuration mode to shut down the SSH service on a switch.

The SSH service runs by default.

To shut down the SSH service on a switch, enter **ssh server shutdown**.

```
device# ssh server shutdown
```

All SSH sessions are immediately terminated, and cannot be re-established until the service is re-enabled.

> NOTE
> In a VCS Fabric, you must enter RBridge ID configuration mode before issuing the command.

```
device(config)# rbridge-id 3
device(config-rbridge-id-3)# ssh server shutdown
device(config-rbridge-id-3)#
```

Re-enabling the SSH service

Re-enabling the SSH (Secure Socket Handling) service permits SSH access to a switch.

You must be in global configuration mode to shut down the SSH service on a switch.

To re-enable the SSH service on a switch enter **no ssh server shutdown**.

```
device# no ssh server shutdown
```

> NOTE
> In a VCS Fabric, you must enter RBridge ID configuration mode before issuing the command.

```
device(config)# rbridge-id 3
device(config-rbridge-id-3)# no ssh server shutdown
```

Ethernet management interfaces

The Ethernet network interface provides management access, including direct access to the Network OS CLI. You must configure at least one IP address using a serial connection to the CLI before you can manage the system with other management interfaces. You can either configure static IP addresses, or you can use a Dynamic Host Configuration Protocol (DHCP) client to acquire IP addresses automatically. For IPv6 addresses, both static IPv6 and stateless IPv6 autoconfiguration are supported.

> ATTENTION
> Setting static IPv4 addresses and using DHCP are mutually exclusive. If DHCP is enabled, remove the DHCP client before you configure a static IPv4 address. However, this does not apply to IPv6 addresses.

Extreme VDX Ethernet interfaces

The Extreme VDX Top-of-Rack (ToR) switches have a single configurable Ethernet interface, Eth0, which can be configured as a management interface.

The modular chassis, the Extreme VDX 8770-8 and VDX 8770-4, have two redundant management modules, MM1 and MM2. Each management module can communicate with each of the line cards (interface modules) through an Ethernet connection. Each

management module has two Ethernet interfaces, Eth0 and Eth2. These interfaces are also known as out-of-band (OOB) management interfaces.

Eth0 is the management interface and can be configured with an IP address. Eth2 provides connectivity to the other management module and the line cards in the chassis. The Eth2 IP addressing scheme uses default IP addresses to communicate between the modules; these addresses are not user-configurable.

To set a virtual IP or IPv6 address for the chassis, use the **chassis virtual-ip** or **chassis virtual-ipv6** command in RBridge ID configuration mode.

NOTE

The IP address of the management platform must be in the same subnet as the devices it manages. It can be configured only in the same subnet that supports the out-of-band management module interfaces, and it cannot be used in the same subnet that supports a virtual Ethernet (VE) interface.

Lights-out management

Lights-out management (LOM) is the ability for a system administrator to monitor and manage servers by a LOM remote control program.

A complete LOM system consists of a hardware component called the LOM module and a program that facilitates the continuous monitoring of variables such as microprocessor temperature and utilization. The program also allows for such remote operations as rebooting, shutdown, troubleshooting, alarm setting, fan-speed control, and operating system reinstallation.

The modular chassis, the Extreme VDX 8770-8 and VDX 8770-4, have two redundant management modules, MM1 and MM2. Each management module can communicate with each of the line cards (interface modules) through an Ethernet connection. Each management module has two Ethernet interfaces, Eth0 and Eth2. These interfaces are also known as Out of Band (OoB) management interfaces and support LOM programs.

Stateless IPv6 autoconfiguration

IPv6 allows the assignment of multiple IP addresses to each network interface. Each interface is configured with a link local address in almost all cases, but this address is only accessible from other hosts on the same network. To provide for wider accessibility, interfaces are typically configured with at least one additional global scope IPv6 address. IPv6 autoconfiguration allows more IPv6 addresses, the number of which is dependent on the number of routers serving the local network and the number of prefixes they advertise.

When IPv6 autoconfiguration is enabled, the platform will engage in stateless IPv6 autoconfiguration. When IPv6 autoconfiguration is disabled, the platform will relinquish usage of any autoconfigured IPv6 addresses that it may have acquired while IPv6 autoconfiguration was enabled. This same enabled and disabled state also enables or disables the usage of a link local address for each managed entity (though a link local address will continue to be generated for each switch) because those link local addresses are required for router discovery.

The enabled or disabled state of autoconfiguration does not affect any static IPv6 addresses that may have been configured. Stateless IPv6 autoconfiguration and static IPv6 addresses can coexist.

Switch attributes

A switch can be identified by its IP address, switch ID or RBridge ID, or by its host name and chassis name. You can customize the host name and chassis name with the **switch-attributes** command.

- A host name can be from 1 through 30 characters long. It must begin with a letter, and can contain letters, numbers, and underscore characters. The default host name is "sw0." The host name is displayed at the system prompt.
- Extreme recommends that you customize the chassis name for each platform. Some system logs identify the switch by its chassis name; if you assign a meaningful chassis name, logs are more useful. A chassis name can be from 1 through 30 characters long, must begin with a letter, and can contain letters, numbers, and underscore characters. The default chassis names are based on the switch models, such as is VDX 8770-4 or VDX 8770-8.

Switch types

The *switchType* attribute is a unique device model identifier that is displayed when you issue the **show chassis** or the **show rbridge-id** command.

When you are gathering information for your switch support provider, you may be asked for the Extreme product name. Use the information in the following table to convert the switchType identifier to a Extreme product name.

TABLE 7 Mapping switchType to Extreme product names

switchType	Extreme product name	Description
112	Management Module	Internal component on the switch
113	Switch Fabric Module	Internal component on the switch
155	VDX 2741	10Gb scalable switch for the BR-VDX2741 chassis, based on the VDX 6740 ToR switch. 42 x 10GbE internal, 14 x 10GbE/16G FC + 2 x 40GbE/64G FC external FlexPorts
138	VDX 2746	10Gb scalable switch for the Hitachi BS2500 chassis, based on the VDX 6740 ToR switch. 42x10GbE internal, 14x10GbE/16G FC + 2x40GbE/64G FC external FlexPorts.
131	VDX 6740	48 10-GbE SFP+ ports and 4 40-GbE QSFP+ ports
137	VDX 6740T	48 10-GbE 10BASE-T ports and 4 40-GbE QSFP+ ports
138	VDX 6746	10Gb scalable switch for Hitachi CB2500 chassis based on the VDX 6740 ToR switch. 42 x 10GbE internal, 14 x 10GbE/16G FC + 2 x 40GbE/64G FC external FlexPorts
151	VDX 6740T-1G	Same as VDX 6740T, but ships with ports set to 1 GbE
153	VDX 6940-36Q	36 40GbE QSFP+ ports
1000.x	VDX 8770-4	4 I/O slot chassis supporting 48x1 GbE, 48x10 GbE, 48x10G-T, 12x40 GbE, 27x40 GbE, or 6x100 GbE line cards
1001.x	VDX 8770-8	8 I/O slot chassis supporting 48x1 GbE, 48x10 GbE, 48x10G-T, 12x40 GbE, 27x40 GbE, or 6x100 GbE line cards

Modular platform basics

The Extreme VDX 8770 platform features two redundant management modules, three or six switch fabric modules, and four or eight line cards, depending on the switch model.

The Extreme VDX 8770-4 supports four line cards and the Extreme VDX 8770-8 supports eight line cards. The following table lists the modules supported on each platform.

TABLE 8 Modules supported on the Extreme VDX 8770 platform

Type MM	Module ID	Slot numbers VDX 8770-4	Slot numbers VDX 8770-8	Description
	0x70 = 112	M1, M2	M1, M2	Management module (an 8-core 1.5-GHz Control Processor)
SFM	0x71 = 113	S1 - S3	S1 - S6	Switch fabric module (core blade)
LC48X10G	0x72 = 114	L1 - L4	L1 - L8	48-port 10-GbE line card
LC12X40G	0x7F = 127	L1 - L4	L1 - L8	12-port 40-GbE line card
LC48X1G	0x83 = 131	L1 - L4	L1 - L8	48-port 1-GbE line card
LC48X10G-T	0x97 = 151	L1 - L4	L1 - L8	48-port 10 Gbps Base-T line card
LC27X40G	0x96 = 150	L1 - L4	L1 - L8	27-port 40-GbE line card
LC6X100G	0x95 = 149	L1 - L4	L1 - L8	6-port 100-GbE line card

Fabric ISLs and edge ports

All interfaces in a VDX 8770 platform come online as Fabric Inter-Switch Links (Fabric ISLs) by default and attempt to form a Extreme VCS fabric. If the ISL formation fails, the interfaces come up as "Edge ports".

Management modules

Two management modules (MMs) provide redundancy and act as the main controller on the Extreme VDX 8770-4 and VDX 8770-8 chassis.

The management modules host the distributed Network OS that provides the overall control plane management for the chassis. You can install a redundant management module in slot M1 or M2 in any of the Extreme VDX 8770 chassis. By default, the system considers the module in slot M1 the active management module and the module in slot M2 the redundant, or standby, management module. If the active module becomes unavailable, the standby module automatically takes over management of the system.

Each management module maintains its own copy of the configuration database. The startup configuration is automatically synchronized with the other management module.

Extreme recommends that each management module (primary and secondary partition) should maintain the same firmware version. For more information on maintaining firmware, refer to the “Installing and Maintaining Firmware” section of the *Extreme Network OS Software Upgrade Guide*.

Each management module has two Ethernet interfaces, Eth0 and Eth2. Eth0 is the management interface and can be configured with an IP address.

HA failover

Warm-recovery High Availability (HA) failover is supported.

Warm recovery includes the following behaviors:

- No data path disruption results for Layer 2 and Layer 3.
- All Layer 2 and Layer 3 control protocol states are retained.
- The topology state and interface state are retained.
- All running configuration is retained (including the last accepted user configuration just before HA failover).
- During a warm recovery, the principal switch remains the principal switch. After warm recovery, the principal switch reestablishes cluster management layer connection with other switches and reforms the cluster.
- A secondary switch reestablishes cluster management layer connection with the principal switch and rejoins the cluster after warm recovery.
- If you run a **reload** command on an active MM, the principal switch goes into cold recovery and comes back up as a secondary switch.
- HA behavior during In-service software upgrades is the same as for warm-recovery failover.

  NOTE
  The **ha failover** command is supported only on a dual-management-module chassis system.

Support for in-service software upgrades

Refer to the Network OS release notes for ISSU and upgrade-path information. An ISSU allows a dual management module system or Top of Rack switches to be upgraded non-disruptively and is invoked by entering the **firmware download** command from the active management module. ISSU is not supported in Network OS 6.0.0 from previous versions. Use the coldboot or manual options of the **firmware download** command.

High Availability behavior during ISSUs is the same as that of warm recovery described in HA failover on page 67. For more information, refer to the “Upgrading firmware on a modular chassis” topic in the *Extreme Network OS Software Upgrade Guide*.

Enabling and disabling standby MM console input

Use the key sequences shown in this section to either allow or prevent user console input.

By default, the standby MM console does not accept any key inputs. Using a special key sequence, you can enable /disable key-access input on the standby MM after HA synchronization is established.

Use the following keyboard sequences to disable and enable console user input:

- To enable input: ctrl _ 1 (ctrl underscore and 1)
- To disable input: ctrl _ 0 (ctrl underscore and 0)

Situations in which console user input becomes enabled are:

- Whenever an MM becomes active due to cold start or warm start.
- If the HA is out of sync for any reason.

Switch fabric modules

The switch fabric modules play a dual role in the fabric connectivity between line cards, providing both the data-plane connectivity and the control-plane connectivity needed for end-to-end credit management in each of the line cards.

In each chassis model, two slots are designated for supporting the control-plane connectivity. In the Brocade VDX 8770-4, the slots S1 and S2 are the designated control-plane slots. In the Brocade VDX 8770-8, the slots S3 and S4 are the designated control-plane slots. At least one of the control-plane slots must be populated to maintain operation. If you remove the switch fabric modules from both the control-plane slots, all line cards will be faulted and the chassis is no longer operational.

Line cards

The following line cards provide I/O ports for network Ethernet protocols:

- LC48x1G—forty eight 1-GbE/10-GbE SFP+ front ports.
- LC48x10G—forty eight 1-GbE/10-GbE SFP+ front ports.
- LC12x40G—twelve 40-GbE QSFP front ports.
- LC48x10G-T—forty eight 10 Gbps Base-T front ports.
- LC27x40G—twenty seven 40-GbE QSFP front ports.
- LC6x100G—six 100-GbE front ports.

Slot numbering and configuration

The slot number specifies the physical location of a module in a device, and the number of available slots of each type (interface, management, or switch fabric) depends on the device. Slot configuration is done on a slot-by-slot basis, and the configurations are stored in a persistent database on the device.

Slot numbering

The slot numbering on the Extreme VDX 8770 chassis is based on the module type. The slot numbers for the line card are numbered L1 through L4 on the Extreme VDX 8770-4, and L1 through L8 on the Extreme VDX 8770-8. The slots for the management modules are numbered M1 and M2. The slots for the switch fabric modules are numbered S1 through S3 on the Extreme VDX 8770-4, and S1 through S6 on the Extreme VDX 8770-8.

Slot configuration

Line cards are registered with the system by type, and the slot must be configured with the correct type before you can install an line card in that slot. When you install a new line card, the system checks whether or not a previous configuration is associated with the slot. The following rules apply when you install or replace an line card:

- When you install an line card and boot it up to an online state in a slot that was never occupied or configured, the module type information is automatically detected and saved to the database. No special configuration is required.
- If you install an line card in a slot that was previously occupied by an line card of the same type and the slot is configured for that same type, you can hot-swap the modules without powering off the line cards. No slot configuration changes are required.
- If the slot was previously configured for a different type of line card, the installation fails and the module is faulted with a "Type mismatch" error. A RASLog error message is generated. You must power off the line card and clear the slot configuration with the **no linecard** command before you can configure the slot for a new line card.

The slot configuration persists in the database even after the line card is physically removed, powered off, or faulted since it first came online. All configuration data associated with the slot is automatically preserved across reboot or hot-swap of the line card with the same type.

Using the management VRF

Virtual Routing and Forwarding (VRF) is a technology that controls information flow within a network, isolating the traffic by partitioning the network into different logical VRF domains.

For details about the management and default VRFs, refer to "Understanding and using management services in default-vrf and mgmt-vrf" in the "VRF" chapter in the *Extreme Network OS Layer 3 Routing Configuration Guide*.

Configuring and managing switches

The following sections describe how to configure and manage Extreme switches.

Configuring Ethernet management interfaces

The Ethernet network interface provides management access, including direct access to the Network OS CLI. You must configure at least one IP address using a serial connection to the CLI before you can manage the system with other management interfaces. You can either configure static IP addresses, or you can use a Dynamic Host Configuration Protocol (DHCP) client to acquire IP addresses automatically. For IPv6 addresses, both static IPv6 and stateless IPv6 autoconfiguration are supported.

ATTENTION
Setting static IPv4 addresses and using DHCP are mutually exclusive. If DHCP is enabled, remove the DHCP client before you configure a static IPv4 address. However, this does not apply to IPv6 addresses.

NOTE
You must connect through the serial port to set the IP address if the network interface is not configured already. Refer to the *Extreme VDX Hardware Reference* manual for your specific product for information on connecting through the serial port.

Configuring static IP addresses

Use static Ethernet network interface addresses in environments where the DHCP service is not available. To configure a static IPv4 or IPv6 address, you must first disable DHCP. Refer to Configuring IPv4 and IPv6 addresses with DHCP on page 71 for more information.

Configuring a static IPv4 Ethernet address

1. Connect to the switch through the serial console.
2. In privileged EXEC mode, issue the **configure terminal** command to enter global configuration mode.
3. Enter the **interface management** *rbridge-id/port* command to configure the management port.

   This command enters a management interface configuration mode where you can choose configuration parameters for IPv4 and IPv6 addresses.
   - A Top-of-Rack (ToR) switch has a single management port, and the port number for the management port is always 0.
   - On a modular switch with two redundant management modules, you can configure two management ports. The port numbers are 1 and 2.
4. Enter the **no ip address dhcp** command to disable DHCP.
5. Enter the **ip address** *IPv4_address/prefix_length* command.

6. Configure an IPv4 management VRF with a default gateway address.

```
device(config-rbridge-id-1)# vrf mgmt-vrf
device(config-vrf-mgmt-vrf)# address-family ipv4 unicast
device(vrf-ipv4-unicast)# ip route 0.0.0.0/0 <default-gateway-ipv4-addr>
device(vrf-ipv4-unicast)# exit
device(config-vrf-mgmt-vrf)# exit
```

7. Verify the configuration with the **do show running-config interface management** command.

> NOTE
> Specifying an IPv4 address with a subnet mask is not supported. Instead, enter a prefix number in Classless Inter-Domain Routing (CIDR) notation. To enter a prefix number for a network mask, type a forward slash (/) and the number of bits in the mask immediately after the IP address. For example, enter, "209.157.22.99/24" for an IP address that has a network mask with 24 leading 1s in the network mask, representing 255.255.255.0.

```
device(config-Management-1/0)# do show running-config interface management
interface Management 1/0
no ip address dhcp
ip address 10.24.85.81/20
vrf forwarding mgmt-vrf
no ipv6 address autoconfig
```

8. Apart from the two IP addresses on the management modules, modular switches also supports a chassis virtual IP address. Using this virtual IP address, you can log in to the switch. The VCS virtual IP address binds to the active MM automatically.

```
device(config)# rbridge-id 1
device(config-rbridge-id-1)# chassis virtual-ip 10.24.85.90/20
```

> NOTE
> In DHCP mode, the chassis IP address is obtained by means of DHCP.

Configuring a static IPv6 Ethernet address

1. In privileged EXEC mode, enter the **configure terminal** command to enter global configuration mode.
2. Enter the **interface management** *rbridge-id/port* command.

   This command enters a management interface configuration mode where you can choose configuration parameters for IPv4 and IPv6 addresses.

   - A Top-of-Rack (ToR) switch has a single management port, and the port number for the management port is always 0.
   - On a modular switches with two redundant management modules, you can configure two management ports. The port numbers are 1 and 2.
3. Enter the **ipv6 address** *IPv6_addresss/prefix_length* command.

```
device# configure terminal
Entering configuration mode terminal
device(config)# interface management 1/0
device(config-Management-1/0)# ipv6 address fd00:60:69bc:832:e61f:13ff:fe67:4b94/64
```

4. Configure an IPv6 management VRF with a default gateway address.

```
device(config-rbridge-id-1)# vrf mgmt-vrf
device(config-vrf-mgmt-vrf)# address-family ipv6 unicast
device(vrf-ipv4-unicast)# ip route 0:0:0:0:0:0:0:0/0 <default-gateway-ipv6-addr>
device(vrf-ipv4-unicast)# exit
device(config-vrf-mgmt-vrf)# exit
```

5. Apart from the two IP addresses on the management modules, modular switches also support a chassis virtual IP address. Using this virtual IP address, you can log in to the switch. The VCS virtual IP address binds to the active MM automatically.

```
device(config)# rbridge-id 1
device(config-rbridge-id-1)# chassis virtual-ipv6 2001:db8:8086:6502/64
```

Configuring IPv4 and IPv6 addresses with DHCP

By default, DHCP is disabled. You must explicitly enable the service. Use the **ip address dhcp** command to enable DHCP for IPv4 addresses, and the **ipv6 address dhcp** command to enable DHCP for IPv6 addresses. The Network OS DHCP clients support the following parameters:

- External Ethernet port IP addresses and prefix length
- Default gateway IP address

NOTE
When you connect the DHCP-enabled switch to the network and power on the switch, the switch automatically obtains the Ethernet IP address, prefix length, and default gateway address from the DHCP server. The DHCP client can only connect to a DHCP server on the same subnet as the switch. Do not enable DHCP if the DHCP server is not on the same subnet as the switch.

The following example enables DHCP for IPv4 addresses.

```
device(config)# interface management 1/1
device(config-Management-1/1)# ip address dhcp
```

The following example enables DHCP for IPv6 addresses.

```
device(config)# interface management 1/1
device(config-Management-1/1)# ipv6 address dhcp
```

The **show running-config interface management** command indicates whether DHCP is enabled. The following example shows a switch with DHCP enabled for IPv4 addresses.

```
device# show running-config interface management
interface Management 2/0
ip address dhcp
ip route 0.0.0.0/0 10.24.80.1
ip address 10.24.73.170/20
no ipv6 address autoconfig
```

NOTE
Enabling DHCP removes all configured static IP addresses.

NOTE
For information on configuring IP DHCP relay, refer to the *Extreme Network OS Layer 3 Routing Configuration Guide*.

Configuring IPv6 autoconfiguration

Refer also to Stateless IPv6 autoconfiguration on page 64.

1. In privileged EXEC mode, issue the **configure terminal** command to enter global configuration mode.

2. Take the appropriate action based on whether you want to enable or disable IPv6 autoconfiguration.
   - Enter the **ipv6 address autoconfig** command to enable IPv6 autoconfiguration for all managed entities on the target platform.
   - Enter the **no ipv6 address autoconfig** command to disable IPv6 autoconfiguration for all managed entities on the target platform.

     NOTE
     On the Extreme VDX 8770, the **autoconfig** command can be issued only on the interface *rbridge-id* /1. However, this operation enables auto-configuration for the entire chassis.

Displaying the network interface

If an IP address has not been assigned to the network interface, you must connect to the Network OS CLI using a console session on the serial port. Otherwise, connect to the switch through Telnet or SSH. Enter the **show interface management** command to display the management interface.

The following example shows the management interface on a Extreme VDX Top-of-Rack (ToR) switch.

```
device# show interface management
interface Management 9/0
 ip address 10.24.81.65/20
 ip gateway-address 10.24.80.1
 ipv6 ipv6-address [ ]
 ipv6 ipv6-gateways [ fe80::21b:edff:fe0f:bc00 fe80::21b:edff:fe0c:c200 ]
 line-speed actual "1000baseT, Duplex: Full"
 line-speed configured Auto
```

The following example shows the management interfaces on a Extreme VDX 8770-4. IPv6 autoconfiguration is enabled for the entire chassis, and, as a result, a stateless IPv6 address is assigned to both management interfaces.

```
device# show interface management
interface Management 110/1
 ip address 10.20.238.108/21
 ip gateway-address 10.24.80.1
 ipv6 ipv6-address [ "stateless fd00:60:69bc:85:205:33ff:fe78:7d88/64 preferred" ]
 ipv6 ipv6-gateways [ fe80::21b:edff:fe0b:7800 fe80::21b:edff:fe0b:2400 ]
 line-speed actual "1000baseT, Duplex: Full"
 line-speed configured Auto
interface Management 110/2
 ip address 10.20.238.109/21
ip gateway-address 10.24.80.1
 ipv6 ipv6-address [ "stateless fd00:60:69bc:85:205:33ff:fe78:be14/64 preferred" ]
 ipv6 ipv6-gateways [ fe80::21b:edff:fe0b:7800 fe80::21b:edff:fe0b:2400 ]
 line-speed actual "1000baseT, Duplex: Full"
 line-speed configured Auto
```

Configuring the management interface speed

By default, the speed of the interface is set to autoconfiguration, which means the interface speed is optimized dynamically depending on load and other factors. You can override the default with a fixed speed value of 10 Mbps full duplex or 100 Mbps full duplex.

1. In privileged EXEC mode, issue the **configure terminal** command to enter global configuration mode.

```
device# configure terminal
Entering configuration mode terminal
```

2. Enter the **interface management** command followed by *rbridge-id***/0** .

   This command places you in the management interface subconfiguration mode.

```
device(config)# interface management 1/0
device(config-Management-1/0)#
```

3. Enter the **speed** command with the selected speed parameter. The valid values are **10**, **100**, and **auto.**

```
device(config-Management-1/0)# speed auto
```

4. Enter the **do show interface management** command followed by *rbridge-id***/0** to display the new settings.

```
device(config-Management-1/0)# do show interface management 1/0
interface Management 1/0
ip address 10.24.81.65/20
ip route 0.0.0.0/0 10.24.80.1
ipv6 ipv6-address [ ]
ipv6 ipv6-gateways [fe80::21b:edff:fe0f:bc00 fe80::21b:edff:fe0c:c200]
line-speed actual "1000baseT, Duplex: Full"
line-speed configured Auto
```

Configuring a switch banner

A banner is a text message that displays on the switch console. It can contain information about the switch that an administrator may want users to know when accessing the switch.

The banner can be up to 2048 characters long. To create a multi-line banner, enter the **banner login** command followed by the **Esc-m** keys. Enter **Ctrl-D** to terminate the input.

The configuration is applied to all nodes in the cluster.

Complete the following steps to set and display a banner.

1. In privileged EXEC mode, issue the **configure terminal** command to enter global configuration mode.
2. Enter the **banner login** command and a text message enclosed in double quotation marks (" ").
3. Enter the **do show running-config banner** command to display the configured banner.

```
device# configure terminal

Entering configuration mode terminal
device(config)# banner login "Please do not disturb the setup on this device"

device(config)# do show running-config banner

banner login "Please do not disturb the setup on this switch"
```

   Use the **no banner login** command to remove the banner.

Configuring switch attributes

Refer also to:

- Switch attributes on page 65.
- Switch types on page 65.

Setting and displaying the host name

1. In privileged EXEC mode, enter the **configure terminal** command to enter global configuration mode.

2. If Telnet is not activated on the switch, enter the **no telnet server disable** command to activate Telnet.
3. Enter the **switch-attributes** command, followed by a question mark (?) to determine the local RBridge ID.
4. Enter the **switch-attributes** command, followed by the RBridge ID.
5. Enter the **host-name** operand, followed by the host name.
6. Verify the configuration with the **do show running-config switch-attributes** *rbridge-id* command.

```
device# configure terminal
Entering configuration mode terminal
device(config)# no telnet server disable
device(config)# switch-attributes ?
Possible completions: <NUMBER:1-239>  Specify the rbridge-id 1
device(config)# switch-attributes 1
device(config-switch-attributes-1)# host-name lab1_vdx0023
device(config-switch-attributes-1)# exit
device(config)# do show running-config switch-attributes 1
switch-attributes 1
 chassis-name VDX 6740-48
 host-name lab1_vdx0023
```

Setting and displaying the chassis name

1. In privileged EXEC mode, issue the **configure terminal** command to enter global configuration mode.
2. Enter the **switch-attributes** command, followed by a question mark (?) to determine the local RBridge ID.
3. Enter the **switch-attributes** command, followed by the RBridge ID.
4. Enter the *chassis-name* operand, followed by the chassis name.

```
device# configure terminal
Entering configuration mode terminal
device(config)# switch-attributes ?
Possible completions: <NUMBER:1-239> Specify the rbridge-id 1
device(config)# switch-attributes 1
device(config-switch-attributes-1# chassis-name lab1_vdx0023
device(config)# do show running-config switch-attributes 1
switch-attributes 1
 chassis-name lab1_vdx0023
 host-name lab1_vdx0023
```

Viewing switch types

The switchType attribute is a unique device model identifier that allows you to identify the model of a switch from the command line.

In this example, the number 1000 is the value of the switchType attribute. An optional number (.x) indicates the revision of the motherboard.

Refer also to Switch types on page 65.

Enter **show chassis**.

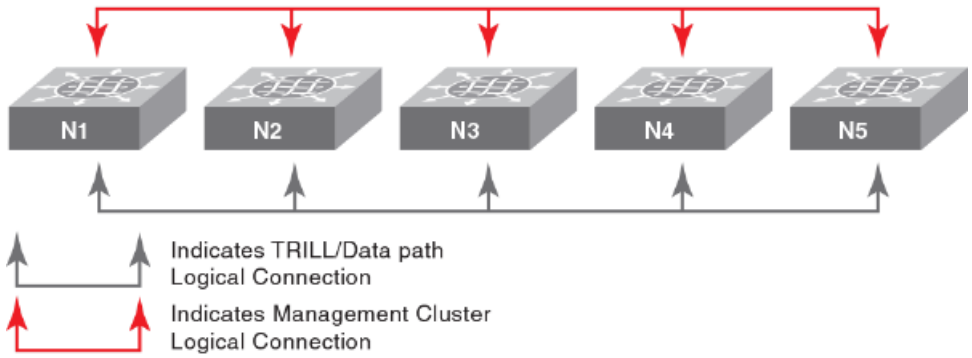
```
device# show chassis
Chassis Family: VDX 87xx
Chassis Backplane Revision: 1
switchType: 1000 <== Use table to convert this parameter
     (output truncated)
```

Creating a VCS cluster

This topic covers the basic steps to create a VCS cluster, with the assumption that all physical connectivity requirements have been met.

The following figure is a representation of a five-node VCS cluster.

FIGURE 14 Five-node VCS cluster



1. Log in to one switch that will be a member of the VCS cluster you are creating.
2. In privileged EXEC mode, enter the **vcs** command, specifying the VCD ID and the RBridge ID.

```
device# vcs vcsid 22 rbridge-id 15 logical-chassis enable
```

3. The switch reboots after you run the **vcs** command. You are asked if you want to apply the default configuration; answer **yes**.
4. Repeat the previous steps for each node in the cluster, changing only the RBridge ID each time. You must, however, set the VCS ID to the same value on each node that belongs to the cluster.
5. When you have enabled VCS on each node in the cluster, run the **show vcs** command to determine which node has been assigned as the cluster principal node. The arrow (>) denotes the principal node. The asterisk (*) denotes the current logged-in node.

```
device# show vcs
Config Mode    : Distributed
VCS Mode       : Logical Chassis
VCS ID         : 44
VCS GUID       : bcab366e-6431-42fe-9af1-c69eb67eaa28
Total Number of Nodes           : 3
Rbridge-Id       WWN                             Management IP   VCS Status       Fabric Status        HostName
-----------------------------------------------------------------------------------------------------------------
144              10:00:00:27:F8:1E:3C:8C         10.18.245.143   Offline          Unknown              sw0
152             >10:00:00:05:33:E5:D1:93*        10.18.245.152   Online           Online               cz41-h06-
m-r2
158              10:00:00:27:F8:F9:63:41         10.18.245.158   Offline          Unknown
sw0
```

In this example, RBridge ID 152 is the principal.

6. Set the clock and time zone for the principal node. Time should be consistent across all the nodes. Refer to Network Time Protocol overview on page 151.

7. Log in to the principal cluster and make any desired global and local configuration changes. These changes then are distributed automatically to all nodes in the cluster.

   NOTE
   You can enter the RBridge ID configuration mode for any RBridge in the cluster from the cluster principal node. You can change the principal node by using the **logical-chassis principal priority** and **logical chassis principal switchover** commands. For more information about cluster principal nodes, refer to Selecting a principal node for the cluster on page 76.

Selecting a principal node for the cluster

VCS principal node behavior includes:

- All configuration must be performed on the principal node.
- You can run the **show vcs** command to determine which node is the principal node. An arrow in the display from this command indicates the principal node.
- You can select any node to become the principal by running the **logical chassis principal priority** command, followed by the **logical-chassis principal switchover** command, as shown in the following example (in this example, RBridge ID 5 is being assigned with the highest priority):

```
device# configure terminal
device(config)# rbridge-id 5
device(config-rbridge-id-5)# logical-chassis principal-priority 1
device(config-rbridge-id-5)# end
device# logical-chassis principal-switchover
```

A lower number means a higher priority. Values range from 1 to 128.

Until you run the **logical-chassis principal switchover** command, the election of the new principal node does not take effect.

Adding a node to a VCS cluster

Nodes can be dynamically added to an existing VCS cluster. If the proper physical connections exist between the cluster and the new node, the process is automatic.

Log into the new node and run the **vcs set-rbridge-id** command with the desired options. You must assign the new node the VCS ID of the existing cluster.

You can run the **show vcs** command to verify that the status of the added node is "online."

Removing a node from a VCS cluster

Removing a device from a VCS cluster applies default configuration to that device.

To remove a device from a VCS cluster, use the **no vcs logical-chassis enable default-config** command in privileged EXEC mode, as in the following example:

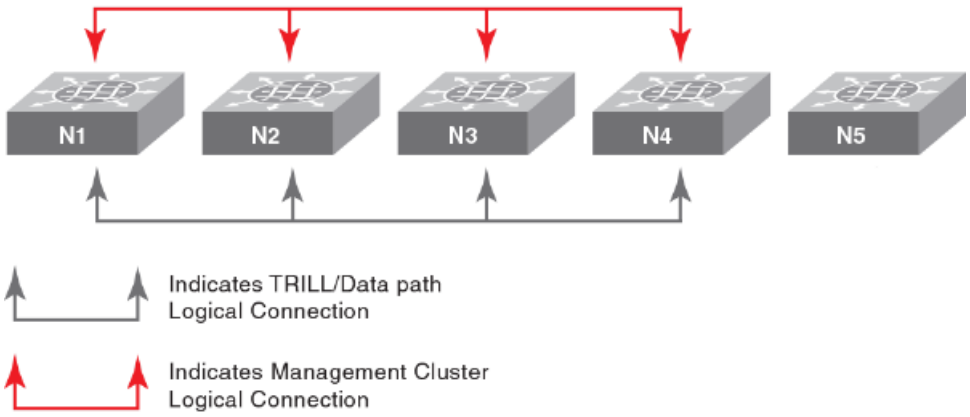
```
device# no vcs logical-chassis enable rbridge-id 239 default-config
```

The command removes the device from the VCS cluster and automatically changes the RBridge ID to 1 and the VCS ID to 8193, applying a default configuration. Configurations corresponding to that node are removed from the cluster configuration database.

To return the device to the original VCS cluster, the RBridge ID and VCS ID must be set to appropriate values for that network.

The following figure shows a cluster after node N5 has been removed. Nodes N1 through N4 remain in the cluster, and N5 is an island. There is no data path or management path connectivity between the two islands.

FIGURE 15 Removal of Node N5 from a VCS cluster



Rejoining an offline node to a VCS cluster

If a node goes into offline state, and configuration changes are made to online nodes while the VCS is in a degraded state, database mismatches can occur when the offline node tries to rejoin the VCS.

Nodes that are temporarily isolated from a VCS cluster can rejoin the cluster as long as no configuration or cluster membership changes have taken place on either the deleted node or the cluster. However, if configuration changes have occurred on either the node or cluster since the node was removed, you must reboot the node with its default configuration by issuing **copy default-config startup-config** on the segmented node.

Situations that can cause a node to go offline include:

- ISLs are shutdown, isolating a node from the VCS.
- A node reboots.

The following list describes possible scenarios that can occur when an offline node tries to rejoin its VCS, and what actions, if any, you must take. For more information about commands that are referenced, refer to the *Network OS Command Reference*.

- If local-only configurations have been added, updated, or deleted for online nodes after another node has temporarily left the VCS, the offline node automatically rejoins the VCS. Any non-default configurations specific to the rejoining node are then pushed back to the VCS configuration. If the rejoining node has the default configuration, and if no local-only configuration changes were made while this node was offline, then the VCS configuration for the rejoining node is pushed onto the rejoining node.
- If the offline node has local-only configuration changes and its global configuration is non-default and matches the global configuration of the VCS, then the offline node is allowed to rejoin the VCS. The local-only configuration changes that were made while the node was offline are preserved.
- If global configurations have been added, updated, or deleted after another node has temporarily left the VCS, the global configuration differences between the VCS and the rejoining node result in a configuration database mismatch and node segmentation. To rejoin the VCS, issue the **copy default-config startup-config** on the segmented node.
- If the rejoining node's global configuration is the default configuration, and both of the following are true: 1) local-only configuration changes have been made to the rejoining node while it was offline, and 2) the VCS contains a different set of local configurations for the rejoining node, then a configuration database mismatch occurs. This situation can occur if the user issues

a **copy default-config startup-config** command on a segmented node and issues local-only configurations on the segmented node before it rejoins the VCS. To rejoin the VCS, issue the **copy default-config startup-config** command on the segmented node or cluster island.

Replacing a node in a VCS cluster

If a node in a VCS cluster becomes damaged and can no longer be used, a similar node with identical capabilities can be used in its place.

The new node must use the same RBridge ID of the node that is being replaced. When the new node is detected, it joins the cluster as a previously known node instead of being considered a new node.

If the **vcs virtual-fabric enable** command is configured on the VCS, you must configure the **vcs virtual-fabric enable** command on the new device.

In the following VDX fabric example of four switches, RBridge-ID 2 is considered to be faulty and must be replaced:

```
device# show vcs
 Config Mode    : Distributed
 VCS Mode       : Logical Chassis
 VCS ID         : 10
 VCS GUID       : bd9870ed-3309-4839-bda5-ac179603662c
 Total Number of Nodes           : 4
 Rbridge-Id       WWN                      Management IP     VCS Status     Fabric Status     HostName
 -----------------------------------------------------------------------------------------------------
 1         >10:00:00:05:33:40:32:93*       10.17.76.74      Online          Online       VDX1-6720
 2          10:00:00:05:33:7D:6C:E6        10.17.76.76      Online          Online       VDX2-6730
 3          10:00:00:27:F8:17:C6:A8        10.17.76.121     Online          Online       VDX3-8770
 4          10:00:00:05:33:AC:AF:DB        10.17.76.72      Online          Online       VDX4-6710
```

1. Upgrade the new device to the same firmware version as the rest of the VCS cluster. The new device should be the same model and configuration as the device being replaced.
2. Configure the same VCS ID on the new device as the rest of the cluster. If properly done, the device automatically reboots. For this example, the command would be as follows.

```
device# vcs vcsid 10 set-rbridge-id 2
```

3. Once the new device reboots, run the **show vcs** command on the new device.

```
device# show vcs
 Config Mode    : Distributed
 VCS ID         : 10
 VCS GUID       : 00000000000000000000000000000000
 Total Number of Nodes           : 1
 Rbridge-Id       WWN                      Management IP   VCS Status    Fabric Status     HostName
 -----------------------------------------------------------------------------------------------
 2          >10:00:00:05:33:7D:76:B0*       10.17.76.91     Online        Online            sw0
```

4. Disconnect all data cables from the faulty device to be replaced. Make notes of the cable connections.

5. The **show vcs** command displays the RBridge ID as being offline.

```
device# show vcs
 Config Mode    : Distributed
 VCS Mode       : Logical Chassis
 VCS ID            : 10
 VCS GUID       : bd9870ed-3309-4839-bda5-ac179603662c
 Total Number of Nodes           : 4
 Rbridge-Id       WWN                   Management IP   VCS Status    Fabric Status      HostName
 -----------------------------------------------------------------------------------------------
 1          >10:00:00:05:33:40:32:93*   10.17.76.74     Online          Online          VDX1-6720
 2           10:00:00:05:33:7D:6C:E6    10.17.76.76     Offline         Unknown         sw0
 3           10:00:00:27:F8:17:C6:A8    10.17.76.121    Online          Online          VDX3-8770
 4           10:00:00:05:33:AC:AF:DB    10.17.76.72     Online          Online          VDX4-6710
```

6. On the principal device of the cluster, enter the **vcs replace** command for the RBridge-ID of the replaced device.

```
device# vcs replace rbridge-id 2
This operation will remove and replace the switch from the fabric. Do you want to continue? [Y/N]  y
```

7. The **show vcs** command displays the RBridge ID as being replaced.

```
device# show vcs
 Config Mode    : Distributed
 VCS Mode       : Logical Chassis
 VCS ID         : 10
 VCS GUID       : bd9870ed-3309-4839-bda5-ac179603662c
 Total Number of Nodes           : 4
 Rbridge-Id       WWN                   Management IP   VCS Status       Fabric Status        HostName
 ---------------------------------------------------------------------------------------------------
 1          >10:00:00:05:33:40:32:93*   10.17.76.74     Online          Online              VDX1-6720
 2           10:00:00:05:33:7D:76:B0    10.17.76.76     Replacing       Online              VDX2-6730
 3           10:00:00:27:F8:17:C6:A8    10.17.76.121    Online          Online              VDX3-8770
 4           10:00:00:05:33:AC:AF:DB    10.17.76.72     Online          Online              VDX4-6710
```

8. Add the new switch hardware to the network and connect all data cables.
9. Once the ISL interfaces boot, the **show vcs** command status changes from Rejoining to Online.

```
device# show vcs
 Config Mode    : Distributed
 VCS Mode       : Logical Chassis
 VCS ID         : 10
 VCS GUID       : bd9870ed-3309-4839-bda5-ac179603662c
 Total Number of Nodes           : 4
 Rbridge-Id       WWN                   Management IP   VCS Status     Fabric Status     HostName
 ----------------------------------------------------------------------------------------------
 1          >10:00:00:05:33:40:32:93*   10.17.76.74     Online         Online            VDX1-6720
 2           10:00:00:05:33:7D:76:B0    10.17.76.76     Rejoining      Online            VDX2-6730
 3           10:00:00:27:F8:17:C6:A8    10.17.76.121    Online         Online            VDX3-8770
 4           10:00:00:05:33:AC:AF:DB    10.17.76.72     Online         Online            VDX4-6710
```

Merging two VCS clusters

You can merge two VCS clusters that have the same VCS ID.

1. Make all required physical connections between the two independent clusters.
2. Decide which cluster should retain the configuration after the merge. Only one configuration can be retained.
3. On the cluster whose configuration will not be retained, issue the **copy default-config startup-config** command. The nodes in this cluster will reboot with the default configuration.

4. Reboot all nodes in each cluster.
   The cluster whose configuration is being retained recognizes the nodes from the other cluster as new nodes and adds them accordingly.
5. Re-apply the configuration to the cluster whose configuration was not retained.

Changing an RBridge ID on a switch within a fabric

It may become necessary to change the RBridge ID number on a switch that rebooted and has become orphaned from the cluster.

1. Backup the running configuration before changing the RBridge ID, because the local configuration will be reset to default values. Refer to Backing up a running configuration on page 40.
2. From the fabric principal switch, execute the **no vcs enable rbridge-id** *rbridge-id* command, where *rbridge-id* is the switch that was orphaned.

   ```
   device# no vcs enable rbridge-id 3
   ```

3. On the rebooted switch, execute the **vcs set-rbridge-id** command.
4. The VCSID should already be set. If it is not, set it with the **vcs vcs-id** *rbridge-id*.
5. Reboot the orphaned switch.

   The following behavior will take effect after the switch reboots:

   - All interfaces will be in *shutdown* state. You must run the **no shutdown** command on ISL interfaces before the switch will rejoin the cluster.
   - The original configuration will be lost and the switch will have a default configuration when it rejoins the cluster with the new RBridge ID.
6. Use the **show vcs detail** command to verify that the switch is in the fabric.

   ```
   device# show vcs detail
   Config Mode : Local-Only
   VCS ID : 1
   Total Number of Nodes : 6
   Node :1
   Serial Number : BKN2501G00R
   Condition : Good
   Status : Connected to Cluster
   VCS Id : 1
   Rbridge-Id : 38
   Co-ordinator : NO
   Switch MAC : 00:05:33:52:2A:82
   FCF MAC : 0B:20:B0:64:10:27
   Switch Type : BR-VDX6720-24-C-24
   Internal IP : 127.1.0.38
   Management IP : 10.17.10.38
   Node :2
   Serial Number : BZA0330G00P
   ```

Examples of global and local configurations

The following table provides examples of global and local configuration commands that are available under the respective configuration modes. These settings can be viewed respectively by means of the **show global-running-config** command and the **show local-running-config** command.

TABLE 9 Global and local configuration commands

Global	Local
Interface vlan	switch-attributes
interface port-channel	interface management
port-profile	interface ve
mac access-list	diag post
ip access-list	dpod
sflow	switch-attributes
snmp-server	fabric route mcast
protocol lldp	rbridge-id
cee-map	linecard
username	router ospf
	ipv6 router ospf
	router bgp
	protocol vrrp
	vrrp-group
	interface management
	interface gigabitethernet
	interface tengigabitethernet
	interface fortygigabitethernet

Use the **copy snapshot** commands if you need to upload or download configuration snapshot files to and from an ftp or scp server. You may need to use these commands if you took a snapshot of a configuration on a node that was disconnected from the cluster.

Displaying switch interfaces

Interfaces on the VDX 8770 platform are identified by the RBridge ID, slot number, and port number, separated by forward slashes (/). For example, the notation 9/2/8 indicates port 8 located in slot 2 on a chassis with the RBridge ID of 9.

Enter the **show running-config interface** command to display the interfaces and their status.

```
device# show running-config interface tengigabitethernet
interface tengigabitethernet 1/1/1
fabric isl enable
fabric trunk enable
no shutdown
!
interface tengigabitethernet 1/1/2
fabric isl enable
fabric trunk enable
no shutdown
!
interface tengigabitethernet 1/1/3
fabric isl enable
fabric trunk enable
no shutdown
```

```
!
interface tengigabitethernet 1/1/4
fabric isl enable
fabric trunk enable
no shutdown
!
interface tengigabitethernet 1/1/5
fabric isl enable
fabric trunk enable
no shutdown
!
interface tengigabitethernet 1/1/6
fabric isl enable
fabric trunk enable
no shutdown
```

Enter the **show interface** *interface_type rbridge_id/slot/port* command to display the configuration details for the specified interface.

```
device# show interface tengigabitethernet 1/1/9
tengigabitethernet 1/1/9 is up, line protocol is up (connected)
Hardware is Ethernet, address is 0005.3315.df5a
Current address is 0005.3315.df5a
Pluggable media present
Interface index (ifindex) is 4702109825
MTU 9216 bytes
LineSpeed Actual   : 10000 Mbit
LineSpeed Configured : Auto, Duplex: Full
Flowcontrol rx: off, tx: off
Priority Tag disable
Last clearing of show interface counters: 04:12:03
Queueing strategy: fifo
Receive Statistics:
1580 packets, 140248 bytes
Unicasts: 0, Multicasts: 1580, Broadcasts: 0
64-byte pkts: 0, Over 64-byte pkts: 1561, Over 127-byte pkts: 17
Over 255-byte pkts: 2, Over 511-byte pkts: 0, Over 1023-byte pkts: 0
Over 1518-byte pkts(Jumbo): 0
Runts: 0, Jabbers: 0, CRC: 0, Overruns: 0
Errors: 0, Discards: 0, TrillportCtrlFrames: 1564
Transmit Statistics:
1583 packets, 140120 bytes
Unicasts: 0, Multicasts: 1583, Broadcasts: 0
Underruns: 0
Errors: 0, Discards: 0, TrillportCtrlFrames: 1583
Rate info:
Input 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
Output 0.000000 Mbits/sec, 0 packets/sec, 0.00% of line-rate
Time since last interface status change: 00:15:53
```

Refer also to Slot numbering and configuration on page 68.

Displaying slots and module status information

Use the **show slots** command to display information for all slots in the chassis.

The following example shows slot information for the Extreme VDX 8770-8.

```
device# show slots
Slot  Type         Description                ID      Status
-------------------------------------------------------------------------
M1    MM           Management Module          112     ENABLED
M2    MM           Management Module          112     ENABLED
S1    SFM          Switch Fabric Module       113     ENABLED
S2                                                    VACANT@
S3    SFM          Switch Fabric Module       113     ENABLED#
S4    SFM          Switch Fabric Module       113     ENABLED#
S5    SFM          Switch Fabric Module       113     ENABLED
S6    SFM          Switch Fabric Module       113     ENABLED
L1                                                    VACANT
```

```
L2                                                      VACANT
L3    LC48X10G     48-port 10GE card          114     DIAG RUNNING POST1
L4    LC48X10G     48-port 10GE card          114     ENABLED
L5                                                      VACANT
L6                                                      VACANT
L7    LC48X1G      48-port 1GE card           114     ENABLED
L7                                                      VACANT
L8                                                      VACANT
# = At least one enabled SFM in these slots is required.
@ = The SFM Optical Switch is open.
```

Alternatively, you can use the following commands to display slots per module type:

- Use the **show mm** command to display information for the management modules.
- Use the **show sfm** command to display information for the switch fabric modules.
- Use the **show linecard** command to display information for the line cards.

Replacing a line card

You can remove a line card without powering it off. However, doing so will not remove the configuration. When you replace a card with a different type, you must first remove the configuration and then reconfigure the slot for the new line card type.

Install a new line card only if it is supported by the firmware running in the chassis. Inserting a line card into a chassis running firmware that does not support the line card may result in unexpected behavior.

Complete the following steps to replace a line card.

CAUTION
Removing the configuration requires the card to be powered off.

1. Power off the line card by issuing the **power-off linecard** command followed by the slot number.
2. Enter the **configure terminal** command to enter global configuration mode.
3. Enter the **rbridge-id** *rbridge-id* command to enter RBridge ID configuration mode.
4. Enter the **no linecard** *slot_number* command to clear the slot configuration.
5. Remove the line card.
6. Enter the **linecard** *slot_number* command followed by a question mark (?) to display the line card menu.
7. Select a line card type and enter the **linecard** *slot_number linecard_type* command.
8. Enter the **exit** command twice to return to privileged EXEC mode.
9. Insert the new line card into the configured slot.
10. Enter the **power-on linecard** command to power on the line card.

11. Verify the configuration with the **show running-config linecard** *linecard* command.

```
device# power-off linecard 4
device# configure terminal
Entering configuration mode terminal
device(config)# rbridge-id 1
device(config-rbridge-id-1)# no linecard 4
device(config-rbridge-id-1)# linecard 4 ?
Possible completions:
LC12x40G  12X40G linecard
LC48x1G        48X1G linecard
LC48x10G  48X10G linecard
LC72x1G  72X1G linecard
LC48x10GT    48X10G Base-T linecard
LC27X40G        27X40G linecard
LC6X100G        6X100G linecard
device(config-rbridge-id-1)# linecard 4 LC48x10G
Creating new linecard configuration was successful.
device(config-rbridge-id-1)# exit
device(config)# exit
device# show running-config rbridge-id 4 linecard
rbridge-id 1
linecard 1 LC48x10G
linecard 4 LC48x10G
```

Configuring High Availability

The following sections provide you with information on configuring High Availability (HA) support on Extreme switches.

Using HA commands

A variety of High Availability (HA) commands are available on the switch in privileged EXEC mode.

- The **show ha** command displays the management module status.

```
device# show ha
Local (M2): Active, Cold Recovered
Remote (M1): Standby, Healthy
HA enabled, Heartbeat Up, HA State synchronized
```

- The **ha failover** command forces the active management module to fail over. The standby management module will take over as the active management module. This command is only available in a modular chassis system.
- The **reload system** command reboots the entire chassis. This command is supported only on the active management module. This command is not supported on the standby management module. Both management modules must be in sync for the HA reboot operation to succeed. This command can be issued from the principal node to reset one remote node or all of the remote nodes by specifying either the individual *rbridge-id* or **all**.
- The **ha sync start** command enables HA state synchronization after an **ha sync stop** command has been invoked.
- The **show ha all-partitions** command displays details for all line cards and the MM HA state.

  NOTE
  For additional HA commands and related commands, refer to the *Extreme Network OS Command Reference*.

Understanding expected behaviors for reload and failover

The following tables identify expected behaviors that result from controlled and uncontrolled reload and failover conditions.

NOTE
When MMs are out of sync, the **reload** command does not work. Use the **reload system** command to reboot the switch in this case.

TABLE 10 Expected behaviors for controlled reload and failover

Command syntax	Behavior in VCS clusters	Behavior in compact switches
reload	Cold failover to standby management module (MM).	Reloads the switch
reload standby	Reboot the standby MM.	Not available
reload system	Reboot both MMs. MMs will retain the HA roles.	Reloads the switch
ha failover	Causes a reset on the active MM and a warm recovery on the standby MM. All existing telnet sessions are required to be restarted.	Not available

TABLE 11 Expected behaviors for uncontrolled failover

Command syntax	Behavior in VCS clusters	Behavior in compact switches
Panic	Warm failover to standby MM.	Reloads the switch
MM removal	Warm failover to standby MM.	Not available
Power cycle	MMs will retain the HA roles upon booting up.	Resets the switch

Disabling and enabling a chassis

The chassis is enabled after power is turned on, and diagnostics and switch initialization routines have finished. All interfaces are online. You can disable and re-enable the chassis as necessary.

- Use the **chassis disable** command if you want to take all interfaces offline. If the switch was part of an Ethernet fabric, the fabric reconfigures.
- Use the **chassis enable** command to bring the interfaces back online. All interfaces that were enabled before the chassis was disabled are expected to come back online. If the switch was part of an Ethernet fabric, it rejoins the fabric.

NOTE
Disabling the chassis is a disruptive operation. Use the **shutdown** command to disable or enable a few selected interfaces only. Refer to the *Extreme Network OS Command Reference* for more information on this command.

Rebooting a switch

Network OS provides several commands to reboot your system: **reload**, **fastboot**, and **reload system**.

CAUTION
All reboot operations are disruptive, and the commands prompt for confirmation before executing. When you reboot a switch connected to a fabric, all traffic to and from that switch stops. All ports on that switch remain inactive until the switch comes back online.

Disabling the chassis configuration database

When devices encounter abrupt power cycles, there have been rare cases of device configuration database corruption. This database corruption causes the device to reboot and reverts the device to the default configuration if it is not part of management cluster.

In the case of scheduled power-cycles, it is recommended to use the **chassis power-cycle-db-shutdown** command in privileged EXEC mode before actually rebooting or power-cycling the device.

This command shuts down the chassis configuration database, without rebooting the device, and disconnects the node from rest of the management cluster.

> ATTENTION
> This command should not be executed for planned upgrades. As switches are rebooted gracefully during upgrades, there is no need to shut down the database. In fact, shutting down the database will cause an upgrade to fail.

The rest of the cluster is informed that the node is to be power-cycled and removed from the cluster. All commands (except for the **reload** command) are blocked on this node until the node is rebooted or power-cycled. Cluster formation requests to this node are ignored. The node is not fully functional until it reboots or is power-cycled.

1. Execute the **chassis power-cycle-db-shutdown** command on all nodes to be rebooted or power-cycled. Extreme recommends that this command to be executed on secondary nodes first, and finally on the logical cluster primary node.
2. Once the **chassis power-cycle-db-shutdown** command has completed processing, use the **reload** command, or power-cycle the device, to reboot it.

Rebooting a Top-of-Rack switch

- The **reload** command performs a "cold reboot" (power off and restart) of the control processor (CP). If the power-on self-test (POST) is enabled, POST is executed when the system comes back up.
- The **fastboot** command performs a "cold reboot" (power off and restart) of the control processor (CP), bypassing POST when the system comes back up. Bypassing POST can reduce boot time significantly.

CAUTION
Do not perform a reload command between a chassis disable command and a chassis enable command. Your ports will be closed.

Rebooting a modular chassis

A chassis reboot brings up the system in sequential phases. First, software services are launched on the management modules and brought up to the active state. Then, the line cards are powered on and initialized. Software services are launched on the line cards and brought up to the active state. When the line card initialization reaches the final state, the chassis is ready to accept user commands from the CLI interface.

During the boot process system initialization, configuration data (default or user-defined) are applied to the switch through configuration replay. For more information, refer to Managing configurations across redundant management modules on page 42.

- On a modular chassis, the **reboot** and the **fastboot** commands only reboot the management module on which the command is executed. If you log in to the switch IP address and execute one of these commands, only the active management module reboots and POST is bypassed.
- The **reload system** command performs a "cold reboot" (power off and restart) of the entire chassis. If the power-on self-test (POST) is enabled, POST is executed when the system comes back up.

Troubleshooting switches

This section presents an overview of a variety of techniques for capturing data and system messages, which can be helpful in interactions with technical support.

Capturing and managing supportSave data

If you are troubleshooting a production system, you will have to capture data for further analysis or send the data to your switch service provider. The **copy support** command provides a mechanism for capturing critical system data and uploading the data to an external host or saving the data to an attached USB device.

Uploading supportSave data to an external host

To upload supportSave data interactively, enter the **copy support-interactive** command and provide input as prompted. For a non-interactive version of the command, refer to the *Extreme Network OS Command Reference*.

```
device# copy support-interactive

Server Name or IP Address: 10.38.33.131
Protocol (ftp, scp): ftp
User: admin
Password: ********
Directory: /home/admin/support
VCS support [y/n]? (y): n
Module timeout multiplier [Range: 1 to 5. Default: 1]: 1

copy support start
Saving support information for chassis:sw0, module:RAS...(output truncated)
```

Saving supportSave data to an attached USB device

You can use a Extreme-branded USB device to save the support data. This device comes with factory-configured default directories and interacts with the Network OS CLI.

1. Enter the **usb on** command to enable the USB device.
2. Enter the **usb dir** command to display the default directories.

3. Enter the **copy support usb** *directory* command.

```
device# usb on
USB storage enabled
device# usb dir

firmwarekey\ 0B 2010 Aug 15 15:13
support\ 106MB 2010 Aug 24 05:36
support1034\ 105MB 2010 Aug 23 06:11
config\ 0B 2010 Aug 15 15:13
firmware\ 380MB 2010 Aug 15 15:13
Available space on usbstorage 74%
device# copy support usb directory support
```

You can use the **rbridge-id all** option to invoke supportSave on all nodes at the same time. The **copy support rbridge-id all** command is a blocking command. The Telnet session from which the command is issued will be blocked until supportSave is completed on all nodes in the cluster; however, users can again Telnet into the same node or any other nodes in the cluster. When the command is in progress, output messages from all nodes are shown that include the respective node RBridge IDs. The **copy support** command, when executed with USB as the protocol option, will collect support files to the USB device that is connected to the respective nodes. All USB devices connected to each of the nodes should be enabled before the **copy support usb** command is executed.

The following example shows the **copy support** command with the **rbridge-id all** option.

```
device# copy support ftp host 10.1.2.30 user fvt password pray4green directory /support rbridge-id all
switch 100: copy support start
switch 117: Saving support information for chassis:sw0, module:RAS...
switch 100: Saving support information for chassis:sw, module:RAS...
switch 117: Saving support information for chassis:sw0, module:CTRACE_OLD...
......
switch 100: copy support completed
switch 117: copy support completed
2011/04/07-18:03:07, [SS-1000], 2752,, INFO, VDX6720-24, copy support has uploaded support information to the
host with IP address 10.70.4.101.
```

Displaying the status of a supportSave operation

Enter the **show copy-support status** command.

```
device# show copy-support status
Slot Name       SS type         Completion Percentage
# # # # # # # # # # # # # # # # # # # # # # # # # # #
M1              NORMAL          [100%]
L1/0            NORMAL          [100%]
L1/1            NORMAL          [100%]
L2/0            NORMAL          [100%]
L2/1            NORMAL          [100%]
L4/0            NORMAL          [100%]
L4/1            NORMAL          [100%]
```

Configuring automatic uploading of supportSave data

You can configure a switch to upload first-fault data capture (FFDC) and trace data files automatically to a remote server that is specifically set up for collecting information that results from the **supportSave** command.

To enable this feature, you must configure a dedicated server, then invoke the **autoupload-param** command to set the parameters, followed by the **support autoupload enable** command to enable the configurations.

```
device(config)# support autoupload-param hostip 10.31.2.27 username supportadmin directory /users/support/
ffdc_autoupload protocol ftp password (<string>): ******
```

Displaying the autoupload configuration

Enter the **show running-config support autoupload-param** command to display the autoupload configuration on the local switch.

```
device(config)# do show running-config support autoupload-param

support autoupload-param hostip 10.31.2.27 username supportadmin directory /users/support/ffdc_autoupload
protocol ftp password "3iTYxJWEUHp9axZQt2tbvw==\n"
```

Using additional supportSave commands

Use the following commands to configure additional supportSave data collection parameters:

- Use the **show support** command to display a list of core files on the switch.
- Use the **clear support** command to erase support data on the switch.

Logging error messages

Network OS provides several mechanisms for logging error messages including syslog, RASLog, and audit log. The types of message logging available and the setup procedures are documented in the "Introduction to Extreme Error Message Logging" chapter of the *Network OS Message Reference Manual*.

Configuring policy-based resource management and hardware profiles

The policy-based resource management feature allows users to make better use of hardware resources.

In particular, pre-made profiles are provided that optimize ASIC resources for Keep-Alive Protocol (KAP) profiles, route-table profiles, and ternary content-addressable memory (TCAM) profiles. The profiles are enabled by keywords available under the **hardware-profile** command in RBridge ID configuration mode. The switched-based hardware profile configuration is local to an RBridge within a VCS Fabric.

> ATTENTION
> This is a disruptive command. In order for the last update of the hardware profile configuration to take effect on a switch, the switch has to be rebooted, using the **reload system** command.

Configuring KAP profiles

The Keep-Alive Protocol (KAP) profile feature allows users to allocate hardware resources to various keep-alive protocols.

KAP profile overview

As a result, during a failover, those protocols can continue to send keep-alive packets to network neighbors and maintain connectivity, achieving a hitless failover. Users now have the flexibility of defining custom KAP profiles for the following protocols: LACP, XSTP, RPVST, UDLD, BFD-VXLAN, and BFD-L3. In addition, default KAP profiles are predefined for the supported platforms.

There are two types of entries for the resources allocated to KAP applications: hardware and software. With hardware entries, the switch continues to send keep-alive packets even during an HA failover; with software entries, the keep-alive packets are resent only after the failover process is completed.

Default KAP profile

By default, the switch comes up with a default KAP profile. With this profile, any protocol that requests a KAP entry gets it first, with requests processed on a first-come, first-served basis. The switch processes the hardware entries first. When those are consumed, the switch begins allocating new KAP entries from the software entry pool.

Custom KAP profiles

A custom KAP profile is defined globally within a cluster and applied to individual switches. Normally, a user would define a global KAP profile for each type of switch, based on physical capability and deployment requirements. The user can define as many custom KAP profiles as needed, each with a unique name and parameter settings.

The **custom-profile** option is applied in hardware configuration mode, by means of the global **hardware** command. Then, by means of the **kap** keyword, the user can create custom named instance and select from among the available protocols. Profiles include the keep-alive interval and the number of keep-alive entries per slot. The following example illustrates this process, with the **show running-config hardware custom-profile** command used to verify the configuration.

```
device(config)# hardware
device(config-hardware)# ?
Possible completions:
  connector                Configure a connector
  connector-group     Configure a connector group
  custom-profile         Configure customized hardware profiles
  describe                 Display transparent command information
  do                       Run an operational-mode command
  exit                     Exit from current mode
  flexport                 Option to change the Ethernet port to FibreChannel port
  help                     Provide help information
  no                       Negate a command or set its defaults
  port-group          Configure a port-group
  pwd                      Display current mode path
  top                      Exit to top level and optionally run command
device(config-hardware)# custom-profile kap myprofile
device(config-kap-myprofile)# ?
Possible completions:
  bfd-l3          Configure BFD-L3 protocol KAP parameters
  bfd-vxlan       Configure BFD-VXLAN protocol KAP parameters
  describe        Display transparent command information
  do              Run an operational-mode command
  exit            Exit from current mode
  help            Provide help information
  lacp            Configure LACP protocol KAP parameters
  no              Negate a command or set its defaults
  pwd             Display current mode path
  rpvst           Configure RPVST protocol KAP parameters
  top             Exit to top level and optionally run command
  udld            Configure UDLD protocol KAP parameters
  xstp            Configure xSTP protocol KAP parameters
  device(config-kap-myprofile)#
  device(config-kap-myprofile)# udld num-entry 64
  device(config-kap-myprofile)# lacp hello-interval 30000 num-entry 50
  device(config-kap-myprofile)# bfd-vxlan hello-interval 300 num-entry 20
  device(config-kap-myprofile)# xstp num-entry 64
  device(config-kap-myprofile)# bfd-l3 num-entry 100
  device(config-kap-myprofile)# fcoe num-entry 64
  device(config-kap-myprofile)# rpvst num-entry 128
  device# show running-config hardware custom-profile
      hardware
         custom-profile kap myprofile
            lacp hello-interval 30000 num-entry 50
            xstp num-entry 64
            rpvst num-entry 128
            udld num-entry 64
            bfd-vxlan hello-interval 300 num-entry 20
            bfd-l3 num-entry 100
```

```
    !
  !
```

NOTE

The "hello-interval" and "num-entry" parameters are optional. If they are not specified, default values are used.

The profile "myprofile" can now be applied to multiple switches in the cluster. The settings are dependent on the platform and the user application. For a custom profile to take effect, it has to be applied to a switch in RBridge ID configuration mode, as illustrated in the following section. When a global custom profile is defined, only a generic validation process is performed, not a platform-specific validation.

Once a custom KAP profiie is activated on one or more switches, the profile cannot be modified or deleted. To change custom profile settings on a switch, the user must first define a new custom KAP profile and apply it to the switch. Only when a custom profile is no longer applied to any switch in the cluster can that profile be modified or deleted.

```
device(config-hardware)# no custom-profile kap myprofile
```

NOTE

All of the protocol types listed in the menu example support the **no** keyword. This allows the user to delete all or part of the settings for the protocol. If a custom setting for a protocol is removed from a custom KAP profile, the default KAP setting for that protocol is applied.

Switch-based KAP profile

This following example illustrates how to apply a custom KAP profile to a specific switch. This is done in RBridge ID configuration mode, by means of the **hardware-profile** command with the **kap** keyword. The configuration is verified by means of the **show running config rbridge-id** command with the **hardware-profile** keyword.

```
device(config-rbridge-id-85)# hardware-profile ?
 Possible completions:
  kap                   Select KAP profile type
  route-table           Select route table profile type
  tcam                  Select TCAM profile type
device(config-rbridge-id-85)# hardware-profile kap ?
Possible completions:
  custom-profile       Customized profile
  default           Basic support for all applications
device(config-rbridge-id-85)# hardware-profile kap custom-profile myprofile
%Warning: To activate the new profile config, please run 'reload system' on the target switch.

device# show running-config rbridge-id 85 hardware-profile
rbridge-id 85
 hardware-profile tcam default
 hardware-profile route-table ipv4-max-arp maximum_paths 32
 hardware-profile kap custom-profile myprofile
```

When a custom KAP profile is applied, a validation process checks whether the specified custom profile exists. Then a platform-dependent validation of the custom profile is triggered. Both the parameter settings for individual protocols and the total entries are checked against platform capabilities. As with other hardware profiles, the switch must be rebooted for changes to take effect.

Limitations

If the user executes the **copy default-config startup-config** command, the KAP profile settings in the hardware are set to the default following a reboot. This occurs even if there is a mapping of a KAP profile setting to a custom KAP profile that has been removed as a result of the global configuration reverting to the default state. Other hardware profile settings are maintained.

Configuring route-table profiles

Route-table profiles can be optimized in hardware.

The following table describes the available command options (keywords) to optimize route profiles, available under the **hardware-profile route-table** keyword.

TABLE 12 Options for optimizing route-table profiles

Keyword	Optimizes resources for . . .
default	Basic IPv4/IPv6 dual-stack operations
ipv4-max-arp	Maximum number of IPv4 ARP entries
ipv4-max-route	Maximum number of IPv4 routes
ipv4-min-v6	IPv4 routes in dual-stack configurations
ipv6-max-nd	Maximum number of IPv6 Neighbor Discovery entries
ipv6-max-route	Maximum number of IPv6 routes
kap	Keep-alive profiles
maximum_paths	Maximum number of ECMP paths (8, 16, 32)
openflow	OpenFlow support

The **maximum-paths** keyword allows the user to configure the maximum number of ECMP paths. This keyword is optional. If not selected, the maximum-path configuration remains unchanged. The valid values are 8, 16, and 32, with 8 the initial default. (Once the latter value is changed, it must be reentered explicitly, which is different from previous releases.) As with other changes to the hardware profile, the switch must be rebooted for changes to take effect.

The **openflow** keyword allows the user to enable or disable support for the OpenFlow feature in the route-table profile. This keyword is optional. If it is not used, the OpenFlow profile setting is unchanged. This keyword is also available under the TCAM profile.

The following example specifies maximum paths and disables OpenFlow in the TCAM profile, and then confirms the configuration.

```
device(config-rbridge-id-67)# hardware-profile route-table ipv4-max-arp ?
Possible completions:
  maximum_paths   Maximum number of load-sharing paths
  openflow        Enable/disable openflow support
  <cr>
device(config-rbridge-id-67)# hardware-profile route-table ipv4-max-arp maximum-path 16
%Warning: To activate the new profile config, please run 'reload system' on the target switch.

device# show running-config rbridge-id hardware-profile
rbridge-id 67
 hardware-profile tcam default
 hardware-profile route-table ipv4-max-arp  maximum-path 16 openflow off
 hardware-profile kap default
```

The following example selects a route-table profile to optimize resources for the maximum number of IPv6 Neighbor Discovery entries:

```
device(config)# rbridge-id 10
device(config-rbridge-id-10)# hardware-profile route-table ipv6-max-nd
%Warning: To activate the new profile config, please run 'reload system' on the target switch.
```

Configuring TCAM profiles

TCAM profiles can be optimized in hardware.

The following table describes the available command options (keywords) to optimize TCAM profiles, available under the **tcam** keyword.

TABLE 13 Options for optimizing TCAM profiles

Keyword	Optimizes resources for . . .
acl-max	Optimizes resources for maximum ACL rules. (Refer to the following Note.)
default	Basic support for all applications
dyn-arp-insp	Dynamic ARP inspection (DAI)
ipv4-acl	IPv4 ACLs (Refer to the following Note.)
ipv4-v6-mcast	Multicast
ipv4-v6-pbr	IPv4 and IPv6 ACLs and policy-based routing tables
ipv4-v6-qos	IPv4 and IPv6 ACLs and QoS
l2-ipv4-acl	Layer 2 and IPv4 ACLs
l2-acl-qos	Layer 2 ACLs and QoS
openflow	OpenFlow support

NOTE

The **acl-max** keyword is supported only on the Extreme VDX 6940-144S. The **ipv4-acl** keyword is supported only on the Extreme VDX 8770 series. For ACL scaling numbers, refer to the "ACLs" chapter in the *Network OS Security Configuration Guide*.

The OpenFlow feature requires consistency between the route-table and TCAM profile configurations. This consistency is enforced automatically. When the user selects **openflow** for a TCAM profile, the route-table profile is set to OpenFlow support, which is enabled on top of the current setting. Similarly, when the user selects any other type of TCAM profile that does not support OpenFlow, the route-table profile is automatically set to disable OpenFlow support. If the user first disables OpenFlow support in the route-table profile, then the TCAM profile is set to the default subtype. To specify a nondefault TCAM subtype, the user must specify that choice in the TCAM menu. If the user first specifies OpenFlow support in the route-table profile, then the TCAM profile is set automatically to support OpenFlow. Status messages report consistency between the two types of profiles, as in the following configuration examples.

```
device(config-rbridge-id-2)# hardware-profile tcam ?
Possible completions:
  [openflow]
  default                basic support for all applications
  dyn-arp-insp         optimized for dynamic Arp Inspection
  ipv4-acl           optimized for IPv4 ACL only
  ipv4-v6-mcast        optimized for multicast
  ipv4-v6-pbr          optimized for IPv4 and IPv6 ACL, PBR
  ipv4-v6-qos          optimized for IPv4 and IPv6 ACL, QOS
  l2-acl-qos             optimized for L2 ACL, QOS
  l2-ipv4-acl            optimized for L2 and IPv4 ACL
  openflow               optimized for openflow support
device(config-rbridge-id-2)# hardware-profile tcam openflow
%INFO: Openflow is also enabled in route-table profile.
%Warning: To activate the new profile config, please run 'reload system' on the target switch.

device# show running-config rbridge-id hardware-profile
rbridge-id 2
 hardware-profile tcam openflow
 hardware-profile route-table ipv4-max-arp maximum_paths 8 openflow on
 hardware-profile vlan-classification default
 hardware-profile kap default

device(config-rbridge-id-2)# hardware-profile tcam default
%INFO: Openflow is also disabled in route-table profile.
%Warning: To activate the new profile config, please run 'reload system' on the target switch
sw0# show running-config rbridge-id hardware-profile
rbridge-id 2
 hardware-profile tcam default
 hardware-profile route-table ipv4-max-arp maximum_paths 8 openflow off
 hardware-profile vlan-classification default
 hardware-profile kap default
```

```
device(config-rbridge-id-2)# hardware-profile route-table ipv4-max-arp ?
Possible completions:
  maximum_paths   Maximum number of load-sharing paths
  openflow              Enable/disable openflow support
  <cr>
device(config-rbridge-id-2)# hardware-profile route-table ipv4-max-arp openflow ?
Possible completions:
  [on]
  off   Disable openflow
  on    Enable openflow
device(config-rbridge-id-2)# hardware-profile route-table ipv4-max-arp openflow on
%INFO: The TCAM profile has been set to openflow.
%Warning: To activate the new profile config, please run 'reload system' on the target switch

device# show running-config rbridge-id hardware-profile
rbridge-id 2
 hardware-profile tcam openflow
 hardware-profile route-table ipv4-max-arp maximum_paths 8 openflow on
 hardware-profile vlan-classification default
 hardware-profile kap default

device(config-rbridge-id-2)# hardware-profile route-table ipv4-max-arp openflow off
%INFO: The TCAM profile has been set to default. For more granular TCAM selections, please use the TCAM
menu.
%Warning: To activate the new profile config, please run 'reload system' on the target switch.

device# show running-config rbridge-id hardware-profile
rbridge-id 2
 hardware-profile tcam default
 hardware-profile route-table ipv4-max-arp maximum_paths 8 openflow off
 hardware-profile vlan-classification default
 hardware-profile kap default
```

The following example illustrates the configuration of optimization for DAI.

```
device(config-rbridge-id-2)# hardware-profile ?
Possible completions:
  kap                       Select KAP profile type
  route-table               Select route table profile type
  tcam                      Select TCAM profile type
device(config-rbridge-id-2)# hardware-profile tcam ?
Possible completions:
  [openflow]
  default                 basic support for all applications
  dyn-arp-insp         optimized for dynamic Arp Inspection
  ipv4-v6-mcast        optimized for multicast
  ipv4-v6-pbr          optimized for IPv4 and IPv6 ACL, PBR
  ipv4-v6-qos          optimized for IPv4 and IPv6 ACL, QOS
  l2-acl-qos              optimized for L2 ACL, QOS
  l2-ipv4-acl             optimized for L2 and IPv4 ACL
  openflow                optimized for openflow support
device(config-rbridge-id-2)# hardware-profile tcam dyn-arp-insp
%Warning: To activate the new profile config, please run 'reload system' on the target switch
```

Note the following additional conditions for TCAM profiles:

- TCAM profiles affect only ACLs, policy-based routing (PBR), flow-based QoS, and multicast entries, without affecting other features, protocols, or hardware resources.
- The TCAM profile options (listed in the table) are not customizable or configurable, and they may not be appropriate to all network designs.
- The following QoS features are optimized by TCAM profiles:
  - Flow-based QoS and flow-based policing for Layer2/Layer 3 ingress and egress
  - System Qos (VLAN-based) for Layer2/Layer 3 ingress and egress
  - Auto NAS
  - Storm control

  - Flow-based SPAN and RSPAN, including VXLAN based
  - Flow-based sFlow, including VXLAN based
- The following QoS features are not affected by TCAM profiles:
  - All port-based QoS features (RED; PFC and legacy flow control; CoS mutation, DSCP CoS, DSCP traffic class, DSCP mutation; scheduling, shaping, and port-based policing)
  - Port-based SPAN and RSPAN
  - Port-based sFlow

Guidelines for changing hardware profiles

Note the following guidelines for changing hardware profiles.

- You must reload the target switch after changing a switch-based profile for the new profile to take effect. When a secondary switch rejoins the cluster with a default profile configuration while the profile configuration for the secondary switch is "nondefault" on the principle switch, you must reload the secondary switch again after it has rejoined the cluster for the nondefault profile to take effect.
- After a Netinstall or a firmware upgrade, the default profiles are automatically set for all hardware types, with OpenFLow disabled and maximum-paths set to 8. Also, the profile configuration defaults after changing a switch VCS ID or RBridge ID.
- There is no **no** option for the switch-based **hardware-profile** command, because switch-based hardware profiles always exist, with either the default or one of the nondefault configurations.
- When you change a hardware profile, the supported scale numbers remain the same with respect to the configuration even if hardware may not be able to fulfill them. This ensures that the same protocol and interface information remain valid with all hardware profile settings.

Using hardware profile show commands

The following **show** commands can be used to verify the status of hardware profiles.

TABLE 14 Network OS show commands

Command	Description
show hardware-profile	Displays the current active profile information and subtype details for each profile type and RBridge ID on local switch or specified RBridge ID or all switches in the fabric. For complete details on the **show hardware-profile** command, refer to the *Network OS Command Reference*.
show running-config rbridge-id hardware-profile	Displays the enabled route table and TCAM profiles in the running configuration for all RBridge IDs, or a specific enabled RBridge ID.
show running-config hardware custom-profile	Displays user-defined custom profiles. (With the initial release, only custom KAP profiles are supported.)

The **show hardware-profile** command supports fabric-wide hardware profile information. To view the current hardware-profile information on other switches in the cluster, specify an RBridge ID as in the following example.

```
device# show hardware-profile rbridge-id 89 current
```

This also applies to the predefined hardware-profile information of the selected type. The predefined hardware-allocation details for the target platform are retrieved and displayed as for the following example command.

```
device# show hardware-profile rbridge-id 91 tcam ipv4-v6-qos
```

To display hardware profile information for all switches in the fabric at one time, use the **rbridge-id all** keywords as in the following examples:

```
device# show hardware-profile current rbridge-id all

device# show hardware-profile rbridge-id all route-table ipv6-max-nd openflow
```

Hardware-profile upgrade/downgrade considerations

During an upgrade from Network OS 5.x.x or 6.0.0 to Network OS 6.0.1 or later, the database conversion of hardware profile configurations is fully supported. After the upgrade, the hardware profile settings are as follows:

- TCAM settings are unchanged.
- Route-table settings are unchanged.
- KAP settings are set to the default.
- ECMP maximum-path settings are set to 8.
- OpenFlow settings are set to disabled.

Towards a downgrade from Network OS 6.0.1 or later to 6.0.0 or 5.x.x, the user must first:

- Set the KAP profile to the default.
- Disable OpenFLow.
- Set ECMP maximum-paths to 8.
- Disable TCAM DIA and OpenFlow optimization.

NOTE
Because of the extension of the route-table profile parameters, during a downgrade the replaying of the ASCII configuration file collected from Network OS 6.0.1 or later to any prior releases is not supported for the **hardware-profile route-table** command.

Auto Fabric

Auto Fabric is a feature that allows plug-and-play for Extreme VDX switches.

A type of configuration known as *bare-metal* is required for a switch to join an existing VCS cluster by means of plug and play. You must pre-configure this mapping in the existing VCS cluster to allow the bare-metal switch to join the cluster. A switch with its bare-metal flag set to true (which is the factory default) can request the following required configuration settings from a physically connected neighbor that already belongs to the cluster:

- VCS ID
- RBridge ID
- Global VLAN state

From this information, the bare-metal switch auto-configures itself and reboots with the bare-metal flag now disabled. (Any configuration performed on the switch disables the bare-metal flag, but you may need to toggle the ISL. However the bare-metal configuration on the remaining nodes in the cluster remains intact.) Bare-metal state must be disabled for cluster formation to occur.

NOTE
If VCS parameters are not set, the switch returns to the bare-metal-enabled state if you issue a **write erase** command, or if you do a Netinstall. In addition, if VCS parameters are not set, the **firmwaredownload default-config** command also returns the switch to the bare-metal-enabled state.

Using maintenance mode for graceful traffic diversion (VCS Fabrics only)

Maintenance mode diverts all possible traffic through a switch under maintenance to other switches in the cluster, while leaving the switch under maintenance as part of the cluster.

The following sections provide an overview and configuration examples, respectively.

Overview of maintenance mode

Prior to Network OS 7.0.0, during maintenance activity the user had to shut down all interfaces one by one to minimize traffic disruption, resulting in the switch leaving the VCS Fabric.

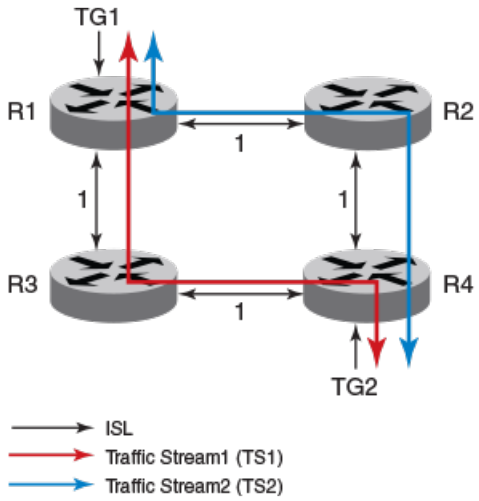
During a firmware upgrade or debugging activity, the user had to disable all edge and ISL ports. This caused data traffic to be lost and the node to leave the cluster, becoming inaccessible from the principle node. Maintenance mode keeps the ISL ports enabled, and existing traffic is gracefully diverted to other nodes in the cluster. The edge ports become administratively down, and the path cost of the node's ISL ports are set to such a high value that traffic through a node in maintenance mode is not preferred when an alternate path is available. The node remains part of the fabric, minimizing traffic disruption in case of an unexpected reload during maintenance.

In the case of vLAGs, if a vLAG member port is shut down abruptly, there can be a black-holing of traffic that lasts a few second. When maintenance mode is enabled, other vLAG member nodes in the cluster are notified to take over the traffic, avoiding loss of traffic.

The following figures illustrate the flow of two traffic streams (TSs) from two traffic generators (TGs) before and after maintenance mode is applied.

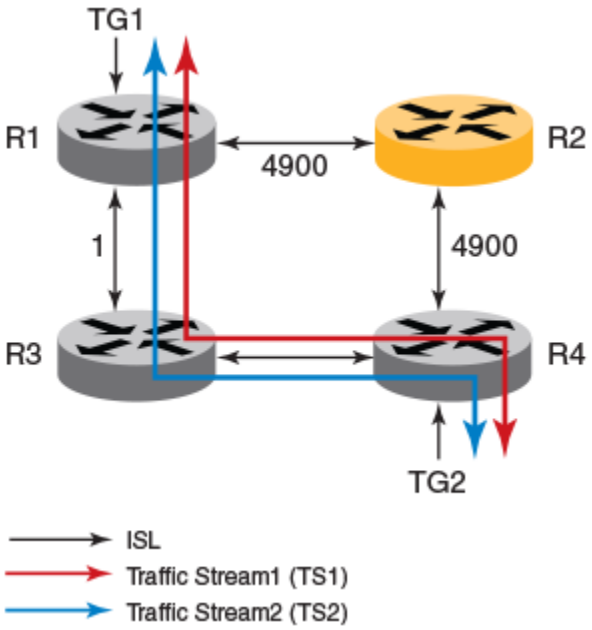
> NOTE
> Traffic is diverted only if an alternative path is available. The user is responsible for ensuring that traffic has a path other than through the node under maintenance.

FIGURE 16 Traffic flow: All RBridges online (before maintenance mode)



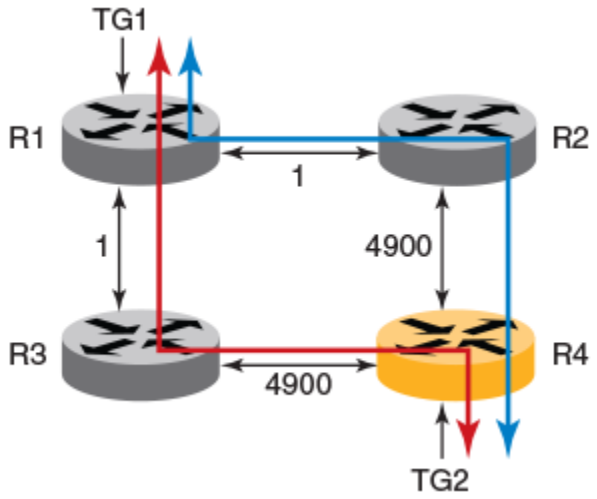
- All nodes in VCS Fabric are in online state
- Path cost of all ISLs is 1
- TS1 flow: TG1 > R1 > R3 > R4 > TG2
- TS2 flow: TG1 > R1 > R2 > R4 > TG2

FIGURE 17 Traffic flow: Intermediate RBridges in maintenance mode



- R2 set to maintenance mode
- Path cost of all ISLs connecting to R2 increased to 4900 (infinity)
- Traffic takes available low-cost alternate paths
- TS1 flow: TG1 > R1 > R3 > R4 > TG2
- TS2 flow: TG1 > R1 > R3 > R4 > TG2

FIGURE 18 Traffic flow: No alternate path available

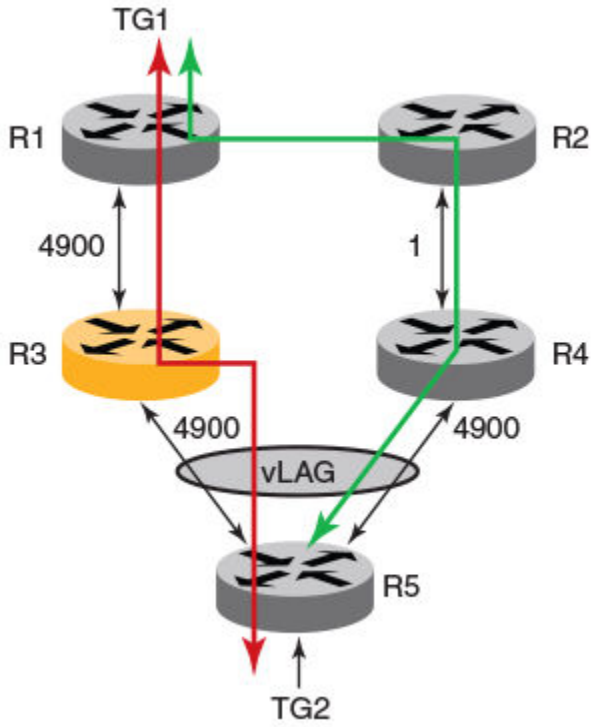


- R4 set to maintenance mode
- Path cost of all ISLs connecting to R4 increased to 4900 (infinity)
- No alternate path for TS1 and TS2 (traffic takes same path)

  NOTE
  Because R4 is in maintenance mode, the TG2 edge interface is in the administratively down state, resulting in traffic disruption. The user must enable the TG2 edge interface manually for traffic to resume while R4 is in maintenance mode.

- TS1 flow: TG1 > R1 > R3 > R4 > TG2
- TS2 flow: TG1 > R1 > R2 > R4 > TG2

FIGURE 19 Traffic flow: vLAG member



- Initially R3 carries vLAG traffic between TG1 and TG2
- TS1 flow: TG1 > R1 > R3 > R5 > TG2
- R3 set to maintenance mode, notifies R4 to take over vLAG traffic. The R3 link is shut down because the vLAG link is considered to be an edge port.
- TS2 flow: TG1 > R1 > R2 > R4 > R5 > TG2

Additional considerations and limitations for maintenance mode

Note the following:

- The maintenance mode configuration and the administrative state of the ports persists across reboots and ISSU/firmware updates.
- It is possible to change the administrative state of any interface while in maintenance mode, but enabling or modifying port-channel and port-channel member ports are not supported.
- If a port-channel or port-channel member configuration is modified while in maintenance-mode, the user should remove and reconfigure the port-channel and port-channel member after exiting maintenance mode.
- ISL ports that are shut down while in maintenance mode will be preserved after maintenance mode is disabled.
- When maintenance mode is disabled, the original user-configured administrative state of the edge ports is restored.
- ISSU and HA failover are supported during maintenance mode.
- Maintenance mode does not restrict any file-copy operations.
- If the switch is the root of a multicast tree, it remains part of a TRILL multicast tree.

Maintenance mode behavior under various scenarios

The following table summarizes the behavior of maintenance mode under various scenarios.

TABLE 15 Maintenance mode behavior under various scenarios

Operation with maintenance mode ENABLED	Interface administrative state with maintenance mode ENABLED	Final interface administrative state with maintenance mode DISABLED
Chassis enable/disable	Error message: "Not allowed"	N/A
Interface shut/no shut on edge ports	Current state is changed accordingly	State prior to maintenance mode is restored
ISL shut	Current state is changed accordingly	Current administrative state is preserved
Slot power off/on	Admin state in running-config is applied	State prior to maintenance mode is restored
ISSU/HA failover	Current state is maintained	State prior to maintenance mode is restored
Breakout/unbreakout/FlexPort	Default admin state (no shut) is applied to new interfaces	Default admin state is maintained for new interfaces
Insert new line card	Default admin state (no shut) is applied to new interfaces	Default admin state is maintained for new interfaces
Reboot	Admin state in running-config is applied	State prior to maintenance mode is restored
Copying default-config to startup-config	Default admin state is applied, maintenance mode is disabled	N/A

Configuring maintenance mode

This task configures maintenance mode and verifies the configuration.

1. To enable maintenance mode, enter the **system-mode maintenance** command in RBridge ID configuration mode.

```
device# config terminal
device(config)# rbridge-id 1
device(config-rbridge-id-1)# system-mode maintenance
```

2. To confirm the configuration, you can use the **show vcs**, **show vcs detail**, and **show system** commands, as in the following respective examples.

```
device# show vcs
Config Mode    : Distributed
VCS Mode       : Logical Chassis
VCS ID         : 10
VCS GUID       : 6990c885-efa2-4b27-bb46-669d2d4d79f1
Total Number of Nodes           : 3
Rbridge-Id       WWN                          Management IP   VCS Status       Fabric Status        HostName
-----------------------------------------------------------------------------------------------------------
2                10:00:00:05:33:E6:BD:00      10.24.84.203    Online           Online               sw0
3               >10:00:00:05:33:65:63:64*     10.24.83.99     Maintenance      Online               sw0
4                10:00:00:27:F8:86:C2:45      10.24.83.97     Offline          Unknown              sw0
```

```
device# show vcs detail
Config Mode    : Distributed
VCS Mode       : Logical Chassis
VCS ID         : 149
VCS GUID       : 00000000000000000000000000000000
Virtual IP                      : 20.0.0.1/24
Virtual IPV6                    : 203::607/64
Associated rbridge-id           : 2
Total Number of Nodes           : 1
Nodes Disconnected from Cluster : 0
Cluster Condition               : Good
Cluster Status                  : All Nodes Present in the Cluster
Node :1
    Serial Number : CPL2548J00Y
    Condition     : Good
    VCS Status    : Co-ordinator
    VCS Id        : 149
    Rbridge-Id    : 2*
    Co-ordinator  : YES
    Switch MAC    : 00:27:F8:C3:C4:B2
    FCF MAC       : DE:AD:BE:EF:DE:AD
    Switch Type   : BR-VDX6740
    Internal IP   : 127.1.0.2
    Management IP : 10.37.18.150
    Fabric Status : Online
    Maintenance Mode   : ON
```

```
device# show system
Stack MAC                     : 00:05:33:e4:7f:2c

  -- UNIT 0 --
Unit Name                     : sw0
Switch Status                 : Maintenance
Hardware Rev                  : 1000.0
TengigabitEthernet Port(s)    : 96
FortyGigabitEthernet Port(s)  : 27
Up Time                       : up 3 days 20:55
Current Time                  : 06:50:54 GMT
NOS Version                   : swrel_nos5.0.x_maint_150420_1403
Jumbo Capable                 : yes
```

3. Alternatively, use the **no** form of the command to disable maintenance mode and revert to the previous configuration status.

```
device(config-rbridge-id-1)# no system-mode maintenance
```

Resolving repeated MAC-moves

Repeated MAC-moves, often caused by loops, overload control-plane resources. Resolving MAC-moves usually solves the problem.

The term *repeated MAC-move* (or *MAC move*) refers to the following sequence:

1. A frame is received on an interface and its MAC address is recorded in the MAC address table.
2. A frame with the identical MAC is received on a different interface on the same switch or VCS.

Following each MAC move, the control plane is interrupted to update the MAC address table with the most recent interface on which such a MAC address was seen. Resolution of repeated MAC-moves is necessary to avoid excessive demand on control plane resources.

Repeated MAC-moves can also lead to MAC table inconsistency across nodes in the cluster. Because the MACs learned on each RBridge are synchronized with the other RBridges in the cluster, the table inconsistences may lead to traffic flooding or black-holing. There is also the danger that even on a given RBridge, there may be inconsistencies among MAC addresses on the management module (MM), line cards (LCs), and the driver.

The following issues can result in repeated MAC-moves:

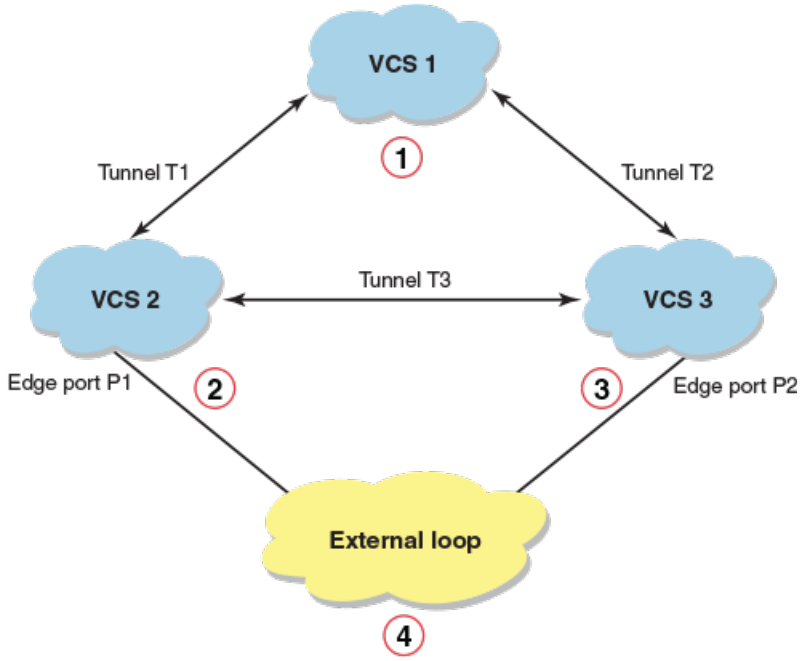
- Network loops
- Server-side flapping
- Other hardware or software issues

MAC-move detection

MAC-move detection provides a mechanism to detect too-frequent MAC moves across interfaces. This feature supports frequent MAC move detection globally at the VCS level, and also provides a mechanism to detect frequent MAC moves between interfaces across the VCS cluster in an overlay environment.

Consider the VCS overlay topology in the following figure.

FIGURE 20 MAC-move detection in a VCS overlay topology



1. MAC-move will be detected between tunnel T1 and tunnel T2.
2. MAC-move will be detected between edge port P1 and tunnel T3.
3. MAC-move will be detected between edge port P2 and tunnel T3.
4. MAC address will keep moving between edge port P1 in VCS 2 and edge port P2 in VCS 3.

The edge ports P1 in VCS 2 and P2 in VCS 3 will be shut down. No action will be taken on VCS 1. The traffic for these ports is black-holed. Consequently, the user must unshut one of the ports so that traffic resumes and a loop is prevented.

MAC-move detection is disabled by default.

If you enable this feature, by means of the **mac-address-table mac-move detect** command, you have the option of specifying the MAC-move threshold (the maximum number of moves allowed within any 10-second window). This is done by means of the **mac-address-table mac-move limit** command. An interface that exceeds the threshold is automatically shut down, and a "Repeated MAC-move" RASLog message is generated. To restore such an interface, you must enter the **no shutdown** command on that interface.

NOTE
The interface must not be a tunnel interface. It must be an edge port.

In addition, support is provided for the following:

- You can specify an action to be taken when the MAC-move detection algorithm identifies a port of interest, by using the **mac-address-table mac-move action** command. The action can be either (1) to shut down the interface (automatically generating a RASLog), or (2) to generate a RASLog only, without shutting down the interface.
- You can enable auto-recovery and specify an interval for auto-recovery following a MAC-move detection and response, by using the **mac-address-table mac-move auto-recovery** command.

NOTE
For details of MAC-move resolution that does not require entering a restoration command, refer to Edge-Loop Detection on page 223.

MAC consistency-check

MAC consistency check provides a mechanism to detect MAC-address inconsistencies across learning modules and the driver in an RBridge or across RBridges in a VCS.

MAC consistency-check is enabled by default. You have the option of specifying the consistency-check interval (the number of seconds between consistency checks). If this feature detects MAC-address inconsistency, it corrects the inconsistent MAC addresses.

Configuring MAC-move detection for an entire VCS cluster

MAC-move detection is disabled by default. You can enable MAC-move detection by means of the **mac-address-table mac-move detect** command, and specify a threshold for repeated MAC-move detection by means of the **mac-address-table mac-move limit** command. The default threshold for repeated-MAC-move detection is 20 moves within any 10-second window.

This use of the **mac-address-table mac-move detect** command detects moves on the entire VCS cluster.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

2. To enable MAC-move detection, enter the **mac-address-table mac-move detect** command.

```
device(config)# mac-address-table mac-move detect
```

3. To modify the default MAC-move threshold, enter the **mac-address-table mac-move limit** command.

```
device(config)# mac-address-table mac-move limit 10
```

4. To restore the default MAC-move threshold of 20 moves, enter the **no mac-address-table mac-move limit** command.

```
device(config)# no mac-address-table mac-move limit
```

5. To disable MAC-move detection, enter the **no mac-address-table mac-move detect** command.

```
device(config)# no mac-address-table mac-move detect
```

6. You can use the **show mac-address-table mac-move** command or the **show ip interface brief** command to confirm the configuration, as in the following examples, respectively.

```
device# show mac-address-table mac-move
Mac Move detect :      Disabled
Threshold:             10

device# show ip interface brief
Interface                  IP-Address     Status    Protocol
============================================================
Port-channel 1             unassigned        up       up
TenGigabitEthernet 1/0/1   unassigned        up       up
TenGigabitEthernet 1/0/2   unassigned  admin-down     down  <-- Shut down as a result of exceeded
threshold
```

Configuring MAC consistency check

MAC consistency check examines MAC-address consistency across VCS RBridges. If needed, this feature resynchronizes the MAC addresses.

NOTE
By default, MAC consistency check is enabled.

1. Enter **configure** to access global configuration mode.

```
device# configure
```

2. To modify the consistency-check interval (default: 300 seconds), enter the **mac-address-table consistency-check interval** command.

```
device(config)# mac-address-table consistency-check interval 500
```

3. To restore the default consistency-check interval of 300 seconds, enter the **no mac-address-table consistency-check interval** command.

```
device(config)# no mac-address-table consistency-check interval
```

4. To disable MAC consistency check, enter the **mac-address-table consistency-check suppress** command.

```
device(config)# mac-address-table consistency-check suppress
```

5. To restore enablement of MAC consistency check, enter the **no mac-address-table consistency-check suppress** command.

```
device(config)# no mac-address-table consistency-check suppress
```

Configuring a MAC-move action

Do the following to specify an action to be taken when the MAC-move detection algorithm identifies a port of interest.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter the **mac-address-table mac-move action** command and specify one of the following options:
   - Use the **shutdown** keyword to cause the port to be shutdown and a RASLog to be generated.

```
device(config)# mac-address-table mac-move action shutdown
```

   - Use the **raslog** keyword to cause a RASLog to be generated without shutting down the port.

```
device(config)# mac-address-table mac-move action raslog
```

3. Enter the **no mac-address-table mac-move action** command to disable this feature.

```
device(config)# no mac-address-table mac-move action
```

4. Enter the **show mac-address-table mac-move action** command to view the results of the configuration.

Configuring MAC-move auto-recovery

Do the following to enable auto-recovery and specify an interval for auto-recovery following a MAC-move detection and response.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter the **mac-address-table mac-move auto-recovery enable** command to enable auto-recovery.

```
device(config)# mac-address-table mac-move auto-recovery enable
```

3. Enter the **mac-address-table mac-move auto-recovery time** command to specify an interval (10 minutes in this example).

```
device(config)# mac-address-table mac-move auto-recovery time 10
```

4. Enter the **no mac-address-table mac-move auto-recovery enable** command to disable auto-recovery. (The **enable** keyword appears only if auto-recovery has been enabled.)

```
device(config)# no mac-address-table mac-move auto-recovery enable
```

5. Enter the **no mac-address-table mac-move auto-recovery time** command to revert to the default interval.

```
device(config)# mac-address-table mac-move auto-recovery time
```

MAC-move show commands

There are several **show** commands that display repeated-MAC-move information. They are documented in the *Network OS Command Reference*, and listed here with descriptions.

TABLE 16 MAC-move show commands

Command	Description
show interface *interface_id*	Displays MAC-move information on a per-interface basis.
show ip/ipv6 interface brief	Indicates whether an interface was shut down due to repeated MAC moves.
show mac-address-table consistency-check	Displays the operational data for MAC address-table consistency check.
show mac-address-table mac-move	Displays MAC-move threshold and enable/disable information.

SFP breakout

Breakout interfaces are those interfaces created on the breakout SFP. The number of interfaces created is dependent on the SFP type.

For example, when a Quad SFP (QSFP) is not in breakout mode, only one 40-Gbps interface exists; however, when that QSFP has breakout mode enabled, four 10-Gbps interfaces are created. These interfaces, whether breakout mode is enabled or disabled, are administered and operate exactly the same as any other interface created on a regular SFP with no breakout capability. As a result, existing DCE module operations are not affected.

Fabric Inter-Switch Links (ISLs) are supported in breakout mode, and the default admin state for breakout interfaces is enabled.

QSFP dynamic breakout is supported, which enables the user to configure breakout mode without having to reboot the switch.

Breakout mode support

Under Network OS, top of rack (ToR) platforms support dynamic breakout and chassis platforms support static breakout.

The following table lists the chassis platforms, which support static breakout mode only.

TABLE 17 Chassis platforms, supporting static breakout only

Platform	Port configuration	QSFP ports
VDX 8770-4 VDX 8770-8	12x40G and 27x40G	12 (12x40G) and 27 (27x40G)

The following table lists the ToR platforms, which support dynamic breakout only; breakout-capable ports and the configuration.

TABLE 18 ToR platforms, supporting dynamic breakout only

Supported platforms	Breakout-capable ports	Supported breakout configuration
VDX 2741	45-48 (4)	40G <-> 4x10G
VDX 2746	57-58 (2)	40G <-> 4x10G
VDX 6740	49-52 (4)	40G <-> 4x10G
VDX 6740T	49-52 (4)	40G <-> 4x10G
VDX 6740T-1G	49-52 (4)	40G <-> 4x10G
VDX 6940-36Q	1-36 (36)	40G <-> 4x10G
VDX 6940-144S	97-108 (12)	40G <-> 4x10G

Breakout mode properties

A breakout interface basically supports all operations or configurations that a regular interface supports (with few exceptions, which are noted in Breakout mode limitations on page 110). As such, it has the following properties:

- Has its own admin and operational state.
- Has its own ASIC resources interface statistics.
- Supports any configuration applicable to any regular SFP interface.
- Can be a port-channel or vLAG member.

Breakout mode can be static or dynamic, depending on the targeted platform. The default state for an SFP is "no breakout."

Ranging and the use of a comma delimiter are allowed in connector interface assignments.

The **no connector** *rbridge-id/slot/port* command is not supported. Consequently, a connector assignment cannot be removed once it has been configured.

Breakout (or unbreakout) is not allowed under the following conditions:

- If any involved physical interfaces on a ToR switch is not in an administratively shut-down state.
- If an involved chassis line card is not in a powered-down state.
- If involved chassis ports on a 27x40G line card are not in Performance mode.
- If any involved physical interfaces belong to a port-channel.
- If any involved breakout port is configured as a FlexPort of Fibre Channel type.

Breakout mode interfaces

The SFP connector ID identifies a physical front-end SFP and has the same meaning as the port ID used in the interface name.

The connector ID states which interfaces are created or deleted as a result of the breakout mode change. All interfaces attached to this connector must be disabled before the command is accepted. An interface created on an SFP in breakout-enabled mode adds a colon followed by a numeric suffix to the existing interface name.

This nomenclature identifies that a port is in breakout mode. For example, line card 2 port 1 on a node with RBridge ID 3 that has a Quad SFP (QSFP) in breakout disabled mode; if breakout is then set to enabled, the existing interface Fo 3/2/1 is deleted and four new interfaces —Te 3/2/1:1, Te 3/2/1:2, Te 3/2/1:3 and Te 3/2/1:4— are created. The new interfaces have the default port configuration. If breakout is set to disabled, the four Te interfaces are deleted and a single Fo interface is created, along with any configuration.

The following table shows an example of an SFP in breakout mode and its respective interface names.

TABLE 19 SFP breakout values

SFP # (rbridge-id/slot/port)	SFP type	Interface name	
		Breakout disabled	Breakout enabled
3/2/1	QSFP (4 x10G)	Fo 3/2/1	Te 3/2/1:1
			Te 3/2/1:2
			Te 3/2/1:3
			Te 3/2/1:4

Breakout mode limitations

In most circumstances, breakout interfaces behave the same as nonbreakout (normal) interfaces with regard to port attributes and states. Each breakout interface maintains its administrative state, operational state, and statistics. The exception is at the physical layer, whereby the hardware platform does not have per-breakout interface information.

- SFP media

  In breakout mode, there is only SFP and no per-breakout media information. The **show media** command displays the same media information for all breakout interfaces.

  NOTE
  The TX Power Field in the **show media** command is not supported by the 40Gbps optics.

- LED state

  For Extreme VDX 6740 series platforms, the LED state for a breakout interface is deterministic. For all other supported platforms, the LED state for a breakout interface *is not* deterministic.

In addition, the 27x40G line card supports nine port groups of three ports each that you can configure for Performance or Density operating modes. If a port group is configured for Performance mode, the first two ports in a group are enabled in Performance mode, but the third port is disabled. If a port group is set for Density mode, all three ports operate in Density mode. Breakout mode can be configured only on ports enabled for Performance mode.

For more information on these modes, line card port groups, and instructions for configuring Performance mode on port groups, refer to the following sections in the *Extreme VDX 8770-4 Hardware Reference Manual* or *Extreme VDX 8770-8 Hardware Reference Manual*:

- 27x40 GbE operating modes
- Configuring operating modes on 27x40 GbE line cards

QSFP dynamic breakout

This feature allows the user to configure breakout mode without having to reboot the switch.

Prior to Network OS 6.0.0, the port breakout of a 40-Gbps QSFP port, typically used as an uplink port to aggregation switches, was available only in a static manner, requiring a reboot of the device and the line card to which the port belonged. Now such a port can be configured as four individual 10-Gbps ports without the need for a reboot, which also prevents other ports from being disturbed.

Configuration updates with dynamic breakout

When dynamic breakout (or unbreakout) is configured, the following are deleted from the running configuration (and backend processes).

- For breakout, the original 40-Gbps physical interface (for example, FortyGigabitEthernet 1/0/50).
- For unbreakout, the four associated breakout 10-Gbps physical interfaces (for example, FortyGigabitEthernet 1/0/50:1-4).
- For unbreakout, the associated connector group and FlexPort configurations.
- All configurations under the physical interface context that is being removed.
- Other protocols (such as VLAN membership) that are associated with the physical interface context that is being removed.

When dynamic breakout (or unbreakout) is configured, the following are created in the running configuration.

- For breakout, a four-breakout 10-Gbps interface (for example, FortyGigabitEthernet 1/0/50:1-4) with a default configuration.
- For breakout, a default connector-group configuration for the platforms that support it.
- For unbreakout, the original 40-Gbps physical interface (for example, FortyGigabitEthernet 1/0/50) with a default configuration.

Additional considerations

Note the following conditions for the preservation of a breakout configuration:

- The breakout configuration is preserved across a **copy default-config startup-config** operation.
- The breakout configuration is not preserved across the operations of changes in RBridge ID, VCS ID, and a write erase.

Note the following conditions when a secondary node joins or rejoins a cluster:

- The breakout configuration in "enabled" state from the secondary node is always preserved.
- Following a rejoin with a nondefault local configuration, the breakout configuration in "enabled" state from the principal node is applied.
- For both ToR and chassis platforms, the breakout configuration from the principal switch is applied to the secondary switch. For dynamic-only (ToR) platforms, no reboot is needed. For static-only (chassis) platforms, the user is notified that a reboot is required for the configuration to take effect.

Configuring static breakout mode for a chassis system

To configure static breakout mode on a blade in a chassis, complete the following procedure.

1. Use the **linecard power-off** command to power off the appropriate line card.

```
device# power-off linecard 2
```

NOTE

The interface and its current configuration will still exist in the configuration, as revealed by a **show running-config** command, but operational commands will not show interfaces on this line card.

2. Enter hardware configuration mode.

```
device# configure terminal
device(config)# hardware
```

3. Enter connector configuration mode for the port you wish to break out.

```
device(config-hardware)# connector 2/2/1
```

4. Execute the **sfp breakout** command. For example, to break out a 40G port into four 10G ports you would use the following example.

```
device(config-connector-2/2/1)# sfp breakout
```

NOTE
Existing interfaces under the previous mode are removed, along with their associated configurations. If the target port is a QSFP, one FortyGigabitEthernet interface is deleted. The configurations of nontarget ports are not affected.

5. (Optional) You can also apply the command to a range of ports, or apply a comma delimiter, as in the following example.

```
device(config-hardware)# connector 2/2/1-3,5
device(config-connector-2/2/1-3,5)# sfp breakout
```

NOTE
The interfaces under the new mode are not created yet; they are created only when the line card is powered back on.

6. Use the **power-on linecard** command to power the line card back on.

```
device# power-on linecard 2
```

NOTE
The SFP interfaces come up under the new mode, with default configurations. Unaffected interfaces retain the configurations they had before the line card was powered off.

7. Execute the **do show running-config hardware connector** command to confirm the breakout.

```
device(config-connector-2/2/1)# do show running-config hardware connector
hardware

 connector 2/2/1
   sfp breakout
 !
```

8. To unbreakout a connector and revert to the previous configuration, use the **no sfp breakout** command as in the following example. You must power off and power on the line card as in the previous steps.

```
device(config-connector-2/2/1)# no sfp breakout
```

NOTE
Four TenGigabitEthernet interfaces are deleted.

Configuring dynamic breakout mode for a ToR system

To configure dynamic breakout mode on a ToR system, complete the following procedure.

1. Ensure that the target physical port is administratively down.

```
device# configure terminal
device(config)# interface fortygigabitethernet 2/0/49
device(config-if-fo-2/0/49)# shut
```

2. Enter hardware configuration mode.

```
device# configure terminal
device(config)# hardware
```

3. Enter connector configuration mode for the port you wish to break out.

```
device(config-hardware)# connector 2/0/49
```

4. Execute the **sfp breakout** command. For example, to break out a 40G port into four 10G ports you would use the following example.

```
device(config-connector-2/0/49)# sfp breakout
```

NOTE
The interface comes up in the "no shut" state. You do not need to reboot the system.

5. (Optional) You can also apply the command to a range of ports, or apply a comma delimiter.

```
device(config-hardware)# connector 2/0/49-50,52
device(config-connector-2/0/49-50,52)# sfp breakout
```

6. Execute the **do show running-config hardware connector** command to confirm the breakout.

```
device(config-connector-2/0/49)# do show running-config hardware connector
hardware

 connector 2/0/49
   sfp breakout
 !
```

7. You can also confirm the entire configuration by using the **show ip interface brief** command.

```
device# show ip int br
Interface                      IP-Address     Vrf                Status         Protocol
==========================     ==========     ================   =========      =========
FortyGigabitEthernet 2/0/50    unassigned     default-vrf        up down
FortyGigabitEthernet 2/0/51    unassigned     default-vrf        up down
FortyGigabitEthernet 2/0/52    unassigned     default-vrf        up down
TenGigabitEthernet 2/0/1       unassigned     default-vrf        up up (ISL)
TenGigabitEthernet 2/0/2       unassigned     default-vrf        up down
TenGigabitEthernet 2/0/3       unassigned     default-vrf        up down
TenGigabitEthernet 2/0/47      unassigned     default-vrf        up down
TenGigabitEthernet 2/0/48      unassigned     default-vrf        up down
TenGigabitEthernet 2/0/49:1    unassigned     default-vrf        up down(ISL)
TenGigabitEthernet 2/0/49:2    unassigned     default-vrf        up down(ISL)
TenGigabitEthernet 2/0/49:3    unassigned     default-vrf        up down(ISL)
TenGigabitEthernet 2/0/49:4    unassigned     default-vrf        up down(ISL)
Vlan 1                         unassigned     administratively   down down
Vlan 4093                      unassigned                        up down
Vlan 4095                      unassigned     administratively   down down
```

8. To unbreakout a connector and revert to the previous configuration, first ensure that the breakout ports are administratively down, and then use the **no sfp breakout** command on the connector, as in the following example.
   a) Shut down all of the breakout ports administratively.

```
device# configure terminal
device(config)# interface tengigabitethernet 2/0/49:1-4
device(config-if-te-2/0/49:1-4)# shut
```

   b) Disable breakout for the connector.

```
device(config)# hardware
device(config-hardware)# connector 2/0/49
device(config-connector-2/0/49)# no sfp breakout
```

NOTE
Four TenGigabitEthernet interfaces are deleted. The 40G interface is created automatically with a default configuration in the "no shut" state.

Reserving and releasing breakout ports

You can use the **dpod** command to enable the Dynamic Ports on Demand feature for breakout ports, and reserve or release them while they are in breakout mode.

1. In global configuration mode, execute the **dpod** command with the **reserve** keyword for the interface you want to reserve.

```
device(config)# dpod 2/2/49 reserve
```

NOTE
You cannot reserve a subinterface.

2. To release the interface, use the **release** keyword.

```
device(config)# dpod 2/2/49 release
```

Dual personality ports

The dual personality port feature for the VDX 6940-144S allows ports 97, 98, 103, and 104 on this device to be configured as 40 GbE QSFP+ or 100 GbE QSFP28 ports provided appropriate transceivers are installed.

Four rows of 40 GbE ports are located on the right side of the VDX 6940-144S front panel with three ports per row. Each row is assigned a dual personality group number of 1 through 4. The following table shows port group mapping to the dual personality port and other ports in the row. Refer to the *Extreme VDX 6940 Hardware Installation Guide* for the specific location of dual personality ports and port groups on the device.

TABLE 20 Dual personality port groups

Port Group	40/100 GbE port # (dual personality port)	40 GbE port #	40 GbE port #
1	97	99	101
2	98	100	102
3	103	105	107
4	104	106	108

The group number is used in Network OS commands for configuration. When you configure 100 GbE mode for a port group, only the dual personality port operates at 100 GbE, if the appropriate 100 GbE QSFP28 transceiver is installed in that port. The remaining two 40 GbE ports in the group are disabled. Conversely, if you reconfigure the port group from 100 GbE to 40 GbE operation, the dual personality port will transition to 40 GbE mode. All three ports in the group will come back online at 40 GbE operation if appropriate 40 GbE optics transceivers installed.

The following port configurations are possible for the twelve 40 GbE ports on the device:

- 12 40 GbE QSFP ports and No 100 GbE QSFP28 ports
- 9 40 GbE QSFP ports and 1 100 GbE QSFP28 port
- 6 40 GbE QSFP ports and 2 100 GbE QSFP28 ports
- 3 40 GbE QSFP ports and 3 100 GbE QSFP28 ports
- No 40 GbE QSFP ports and 4 100 GbE QSFP28 ports

NOTE
100 GbE QSFP28 transceivers do not support breakout mode.

Configure 100 GbE operation on specific dual personality port groups using the **port-group** *rbridge-id/slot/port-group-id* and **mode** *100g* commands. You can transition to 40 GbE operation for specific port groups using the **port-group** *rbridge-id/slot/port-group-id* and **mode** *40g* commands.

NOTE
After configuring 100 GbE or 40 GbE operation for a port group, you must reboot the VDX 6940-144S to enable the new operation mode.

The **show hardware port-group** command displays the port numbers mapped to port groups and modes currently configured for each group. Following is an example showing port group 3 enabled for 100 GbE mode and groups 1, 2, and 4 enabled for 40 GbE mode.

```
device# show hardware port-group

Port-Group          Ports             Mode
=============================================
237/0/1      237/0/97 ,99 ,101        40G
237/0/2      237/0/98 ,100,102        40G
237/0/3      237/0/103,105,107        100G
237/0/4      237/0/104,106,108        40G
```

Before enabling 100 GbE mode for a port group, perform the following tasks:

- Disable 4x10 GbE SFP breakout mode for all ports in that port group, if configured. Ports must be enabled for 40 GbE mode.
- Disable all interfaces in the port group.
- Install the appropriate 100 GbE QSFP28 transceiver in the dual-personality port cage for that port group (port 97, 98, 103, or 104).
- Reserve the DPOD for the ports in the port group.

Limitations and considerations

Consider the following when configuring and using dual-personality ports and port groups:

- When you enable 100 GbE mode for a port group, the leftmost port in the group (dual personality port 97, 98, 103, or 104) will be enabled for 100 GbE operation and the other ports in the port group will be disabled.

- When a port group is enabled in 40 GbE mode, all three ports can be configured to breakout mode. However, you must disable breakout mode on all three ports before configuring 100 GbE mode for the port group. When a port group is enabled in 100 GbE mode, the dual personality port will enabled for 100 GbE operation and the other two ports will be disabled.
- Ports with supported 40 GbE QSFP transceivers that are not in a dual personality port group configured for 100 GbE operation can function in 40 GbE mode and can be configured 4x10 GbE breakout mode.
- To transition dual personality ports 97, 98, 103, or 104 from 100 GbE to 40 GbE operation, you must install a supported 40 GbE QSFP+ transceiver in that port. Note that you can still configure 40 GbE mode for the port group, but without a 40 GbE QSFP+ transceiver installed, the dual personality port will be disabled. The other two ports in the group will operate at 40 GbE. This is same case when configure from 40 GbE mode to 100 GbE mode.
- After configuring 100 GbE or 40 GbE operation for a port group, you must reboot the system to enable the new operation mode.
- Any configuration applied to ports in a port group will be removed when you change to 100 GbE mode or 40 GbE mode.
- A specific dual personality port license is not required, but to enable a dual-personality port in a port group for 100 GbE operation, all three ports in the port group must have ports on demand (POD) reservations allocated from a 40 GbE Port Upgrade license.
- Trunking is not supported on 100 GbE ports.

Configuring 100-GbE operation

Use this topic to configure a dual personality port for 100-GbE operation.

Perform the following tasks before enabling 100-GbE operation:

- Disable 4x10-GbE breakout mode for all 40-GbE ports in that port group.
- Disable all interfaces in the port group using the **shutdown** command.
- Install a qualified 100-GbE transceiver in the dual personality port (the leftmost port in the port group row).

  NOTE
  In the following example, port 97 in dual personality port group 1 is configured for 100-GbE operation.

1. Enter hardware configuration mode.

```
device# configure terminal
Entering configuration mode terminal
device(config)# hardware
```

2. Disable all ports in port group 1 using the **shutdown** command.
3. Enter the port group configuration mode for dual personality port group 1.

```
device(config-hardware)# port-group 1/0/1
device(config-port-group-1/0/1)#
```

   NOTE
   The **port-group** command uses the *rbridge-id/slot/port-group-id* variable to identify the port. To configure a dual personality port group, use slot=0 and port-group-id=1-4. Slot 0 is always used for fixed switches, such as the VDX 6940-144S.

4. Configure the port group for 100 GbE operation using the **mode 100g** command.

```
device(config-port-group-1/0/1)# mode 100g
```

5. Reboot the VDX 6940-144S to enable the new operating mode.
6. Execute the **show running-config hardware port-group** command to confirm the operating mode for the port group.

```
device# show running-config hardware port-group
hardware
port-group 1/0/1
  mode 100g
```

7. Execute the **show hardware port-group** command to display the port numbers mapped to port groups and modes defined for each group.

```
device# show hardware port-group

Port-Group          Ports             Mode
================================================
237/0/1      237/0/97,99,101          100G
237/0/2      237/0/98,100,102         40G
237/0/3      237/0/103,105,107        40G
237/0/4      237/0/104,106,108        40G
```

Configuring 40-GbE operation

Use this topic to transition a dual-personality port from 100-GbE to 40-GbE operation.

Perform the following tasks before enabling 40-GbE operation:

- Disable all interfaces in the port group using the **shutdown** command.
- Install a qualified 40-GbE transceiver in the dual personality port (the leftmost port in the port group row).

NOTE
Steps in the following example configure port 97 in dual personality port group 1 for 40-GbE operation.

1. Enter hardware configuration mode.

```
device# configure terminal
Entering configuration mode terminal
device(config)# hardware
```

2. Disable all ports in port group 1 using the **shutdown** command.
3. Enter the port group configuration mode for dual personality port group 1.

```
device(config-hardware)# port-group 1/0/1
device(config-port-group-1/0/1)#
```

NOTE
The **port-group** command uses the *rbridge-id/slot/port-group-id* variable to identify the port. To configure a dual personality port group, use slot=0 and port-group-id=1-4. Slot 0 is always used for fixed switches, such as the VDX 6940-144S.

4. Configure the port group for 40 GbE operation using the **mode 40g** command.

```
device(config-port-group-1/0/1)# mode 40g
```

5. Reboot the VDX 6940-144S to enable the new operating mode.

6. Execute the **show running-config hardware port-group** command to confirm the operating mode for the port group.

```
device# show running-config hardware port-group
hardware
port-group 1/0/1
  mode 40g
```

7. Execute the **show hardware port-group** command to display the port numbers mapped to port groups and modes defined for each group.

```
device# show hardware port-group

Group No.      Port Number         Current Mode
=========================================
0/1            1/0/97,99,101        40G
0/2            1/0/98,100,102       40G
0/3            1/0/103,105,107      40G
0/4            1/0/104,106,108      40G
```

Tunable SFP+ optics

Tunable SFP+ optics can be installed in 6740-10GbE ports, 6940-144S and 8770 with 48X10G line cards.

The T-SFP+ interface defaults to a "no wavelength" state. When a Extreme device boots, the firmware sets the desired wavelength only if the wavelength is previously configured by the user. Else the device boots with no-wavelength (default).

If you are installing a T-SFP+ in a 144S port, the T-SFP+ optic needs to be installed in both ends of the cable. The T-SFP+ at each end of the cable link must be configured at the same wavelength by setting them to the same channel on each device.

Failure to duplicate the channel setting may allow the link to come online, but the link behavior may be erratic.

When a T-SFP+ interface is installed it is very important that the interface is configured to the same channel (wavelength) at both ends. Use the **show media tunable-optic-sfpp** command to determine the currently configured wavelength.

Configuring tunable SFP+ optics

Perform the following task to configure a tunable SFP+ optic (T-SFP+) to a specific channel.

Repeat these steps for both devices on which the T-SFP+ is installed. You must set both devices to the same channel.

1. Select a channel for the T-SFP+ interface using the **show media tunable-optic-sfpp** command. There are 102 channels available.

```
device# show media tunable-optic-sfpp
command is show-media-tunable-optic-sfpp.
  Channel   Wavelength
  ========= ==============
  1         1568.77
  2         1568.36
  3         1567.95
  4         1567.54
  5         1567.13
  6         1566.72
  7         1566.31
  8         1565.90
  9         1565.50
  10        1565.09
(Output truncated for brevity.)
```

2. Enter global configuration mode.

```
device# configure terminal
```

3. Enter configuration mode for the interface for which the T-SFP+ is installed.

```
device(config)# interface tengigabitethernet 2/0/1
```

4. Use the **tunable-optics** command to set the channel for the interface.

```
device(conf-if-te-2/0/1)# tunable-optics sfpp channel 5
```

5. Confirm the channel configuration is online with the **show media optical-monitoring** command.

```
device(conf-if-te-2/0/1)# do show media optical-monitoring interface tengigabitethernet 2/0/1
N/A - Not Available.
N/S - Optical-monitoring Not Supported.
Port            Module      Supply       Channel      Frequency Wavelength Bias
Channel               Temperature     Temp        Voltage      TX Power
Current      RX Power
               ( C )        ( mVolts )    ( uWatts )    ( GHz )    ( nm )      ( mAmps )     ( uWatts )
=============== =========== ============ ============ ========= ========== ============ ============
Te 2/0/1      35          3314.1       0.1          0         0          0.000        0.0
```

6. Confirm that the correct wavelength is configured by issuing the **show media interface** command.

```
device# show media interface tengigabitethernet 88/0/47
  Interface      TenGigabitEthernet 88/0/47
  Identifier     3    SFP
  Connector      7    LC
  Transceiver           0000000000000000
  Name           id
  Encoding       6
  Baud Rate      99  (units 100 megabaud)
  Length 9u      80  (units km)
  Length 9u      255 (units 100 meters)
  Length 50u     0   (units 10 meters)
  Length 62.5u   0   (units 10 meters)
  Length Cu      0   (units 1 meter)
  Vendor Name    BROCADE
  Vendor OUI     00:05:1e
  Vendor PN      57-1000266-01
  Vendor Rev     A
  Wavelength     1528(units nm)
  Options        065a Loss_of_Sig,Loss_of_Sig_Inverted
  BR Max         10
  BR Min         4
  Serial No      AJJ21407557700C
  Date Code      140223
  Optical Monitor yes
  Temperature    38 Centigrade
  Voltage        3307.6  (mVolts)
  Current        0.000  (mAmps)
  TX Power       0.1  (uWatts)
  RX Power       0.0  (uWatts)
```

Tunable SFP+ optics channels

Tunable SFP+ optics are optional hardware that can be tuned to specific wavelengths and frequencies using pre-defined channels.

The following tables lists the frequency and wavelength assigned to channels for tunable SFP+ optic interfaces.

TABLE 21 Supported wavelengths and channel numbers

Channel	Frequency (THz)	Wavelength (nm)
1	191.10	1568.77
2	191.15	1568.36
3	191.20	1567.95
4	191.25	1567.54
5	191.30	1567.13
6	191.35	1566.72
7	191.40	1566.31
8	191.45	1565.90
9	191.50	1565.50
10	191.55	1565.09
11	191.60	1564.68
12	191.65	1564.27
13	191.70	1563.86
14	191.75	1563.45
15	191.80	1563.05
16	191.85	1562.64
17	191.90	1562.23
18	191.95	1561.83
19	192.00	1561.42
20	192.05	1561.01
21	192.10	1560.61
22	192.15	1560.20
23	192.20	1559.79
24	192.25	1559.39
25	192.30	1558.98
26	192.35	1558.58
27	192.40	1558.17
28	192.45	1557.77
29	192.50	1557.36
30	192.55	1556.96
31	192.60	1556.55
32	192.65	1556.15
33	192.70	1555.75
34	192.75	1555.34
35	192.80	1554.94
36	192.85	1554.54
37	192.90	1554.13
38	192.95	1553.73
39	193.00	1553.33
40	193.05	1552.93
41	193.10	1552.52

TABLE 21 Supported wavelengths and channel numbers (continued)

Channel	Frequency (THz)	Wavelength (nm)
42	193.15	1552.12
43	193.20	1551.71
44	193.25	1551.32
45	193.30	1550.92
46	193.35	1550.52
47	193.40	1550.12
48	193.45	1549.72
49	193.50	1549.32
50	193.55	1548.91
51	193.60	1548.51
52	193.65	1548.11
53	193.70	1547.72
54	193.75	1547.32
55	193.80	1546.92
56	193.85	1546.52
57	193.90	1546.12
58	193.95	1545.72
59	194.00	1545.32
60	194.05	1544.92
61	194.10	1544.53
62	194.15	1544.13
63	194.20	1543.73
64	194.25	1543.33
65	194.30	1542.94
66	194.35	1542.54
67	194.40	1542.14
68	194.45	1541.75
69	194.50	1541.35
70	194.55	1540.95
71	194.60	1540.56
72	194.65	1540.16
73	194.70	1539.77
74	194.75	1539.37
75	194.80	1538.98
76	194.85	1538.58
77	194.90	1538.19
78	194.95	1537.79
79	195.00	1537.40
80	195.05	1537.00
81	195.10	1536.61
82	195.15	1536.22

TABLE 21 Supported wavelengths and channel numbers (continued)

Channel	Frequency (THz)	Wavelength (nm)
83	195.20	1535.82
84	195.25	1535.43
85	195.30	1535.04
86	195.35	1534.64
87	195.40	1534.25
88	195.45	1533.86
89	195.50	1533.47
90	195.55	1533.07
91	195.60	1532.68
92	195.65	1532.29
93	195.70	1531.90
94	195.75	1531.51
95	195.80	1531.12
96	195.85	1530.72
97	195.90	1530.33
98	195.95	1529.94
99	196.00	1529.55
100	196.05	1529.16
101	196.10	1528.77
102	196.15	1528.38

FlexPort

The FlexPort feature allows up to 32 ports to transmit data as either 10G Ethernet or Fibre Channel, and to be changed from one type to the other without requiring a reboot. These ports are grouped together as connector groups.

Connector groups share common speed and protocol type properties. The settings allow any port within each connector group to operate as either an Ethernet or a Fibre Channel port, and support the appropriate optic transceivers. The Fibre Channel ports must be running any supported 8-Gbps or 16-Gbps Extreme FC transceivers.

The default setting is for Ethernet, and ports that do not support the Fibre Channel protocol are not allowed to have their connector group setting changed from the default setting.

The following speed combinations are allowed per connector group:

- **LowMixed** - 2/4/8G Fibre Channel, and Ethernet speeds (default)
- **HighMixed** - 16G Fibre Channel, and Ethernet speeds
- **FibreChannel** - 2/4/8/16G Fibre Channel (only if all eight ports are already set as Fibre Channel type)

For the currently supported platforms, the connector-group numbers range from 1 through 6. They are related directly to the ports as numbered on each platform. The connector-group numbers that are allowed to be changed and their associated port numbers are shown in the following table. For example, on a Extreme VDX 2741, ports 43 through 50 belong to connector group 1. Not every connector group is supported on a switch.

ATTENTION
Changing connector-group speed is disruptive to ports within the same connector-group.

TABLE 22 FlexPort supported hardware

Platform	Port number range	Connector group
Extreme VDX 2741	57-58	3
Extreme VDX 6740	1-8	1
	17-24	3
	33-40	5
	41-48	6

NOTE
For the Extreme VDX 6740T and VDX 6740T-1G, Ethernet operation is supported on 40-GbE QSFP ports configured in 40-GbE mode that use qualified 40-GbE transceivers and on 40-GbE ports in 4x10 GbE breakout mode that use qualified 4x10-GbE QSFP transceivers. Fibre Channel operation is supported on 40-GbE ports configured in 40-GbE breakout mode that use qualified 4x16 QSFP+ transceivers. FlexPort is not supported on 40-GbE QSFP ports on the Extreme VDX 6740. Long-distance configurations are not supported.

Only Extreme-branded SFPs are supported. While all 2/4/8/16G Extreme-branded SFPs are allowed for Extreme engineering purposes, only the following SFPs are qualified and supported:

- 8G SWL 57-1000117-01 (Finisar, Avago)
- 8G LW-10KM 57-1000027-01 (Finisar, Aago)
- 8G ELWL-25KM 57-0000080-01 (Finisar, Avago)
- 16G-SWL 57-0000088-01 (Finisar, Avago)

The 8G SFPs also support 2G and 4G speeds. The 16G SFP also supports 4G and 8G speeds.

The following table lists speeds, buffer-to-buffer (BB) credits, and optimal maximum distances for E and F port types.

TABLE 23 Speeds, BB credits, and optimal maximum distances for E and F port types

Port type	Speed, Gbps	BB credits	Optimal maximum distance, km
E	2	11	11
E	4	11	5.5
E	8	11	2.75
E	16	3	0.375
F	2	50	50
F	4	50	25
F	8	50	12.5
F	16	18	2.25

Configuring FlexPort

The FlexPort feature allows up to 32 ports to transmit either Ethernet or Fibre Channel.

The FlexPort feature is set to Ethernet by default. You should only need to perform this task to switch to Fibre Channel, or back to Ethernet.

1. Enter hardware configuration mode.

```
device(config)# hardware
device(config-hw)#
```

2. Enter FlexPort configuration mode for the switch. This command configures FlexPort 5 on RBridge ID 1. FlexPort 5 is part of connector group 1 on the Extreme VDX 6740.

```
device(config-hw)# flexport 1/0/5
device(conf-hw-flex-1/0/5)#
```

3. Set the FlexPort type to Fibre Channel.

```
device(conf-hw-flex-1/0/5)# type FibreChannel
```

4. Optionally, you may adjust the speed for the connector group. For example, if you want the connector group 1 to function at 16Gbps speed:

```
device(conf-hw-flex-1/0/5)# connector-group 1/0/1
device(config-connector-group-1/0/1)# speed HighMixed
```

5. Repeat these steps for additional switches as needed.
6. Confirm the FlexPort configuration with the **show running-config hardware flexport** command.

```
device# show running-config hardware flexport
Hardware
 flexport 1/0/1
    type FibreChannel
!
Hardware
flexport 1/0/5
    type FibreChannel
!
connector-group 1/0/1
  speed HighMixed
!
```

FlexPorts and breakout mode

Breakout mode is supported on FlexPorts. This allows you to breakout certain QSFP and SFP ports and switch them between Fibre Channel and Ethernet without the need to reboot the device.

NOTE
A 40G license is not required to enable breakout Fibre Channel FlexPorts.

FlexPort requires the QSFP port to be configured in breakout mode.

Restrictions on FlexPorts in breakout mode:

- Fibre Channel SFP+ ports can operate at 2G, 4G, 8G and 16G speeds.

- Fibre Channel QSFP+ ports can operate at 4G, 8G and 16G.

TABLE 24 Valid breakout ports for FlexPort

Platform	Port number range	Connector group
Extreme VDX 2741	43-50 51-56 57-58 (Ports 43 through 56 are SFP and 57 through 58 are QSFP.)	1 2 3
Extreme VDX 6740T Extreme VDX 6740T-1G	49-50 51-52	7 8

Configuring FlexPorts for breakout mode

Configures FlexPorts to function in breakout mode.

The FlexPort feature is set to Ethernet by default.

1. Enter hardware configuration mode.

```
device(config)#hardware
```

2. Enter FlexPort configuration mode for the switch. This command configures FlexPort 49:1 on RBridge ID 1. Flexport 49:1 is part of connector group 7 on the Extreme VDX 6740.

```
device(config-hw)# flexport 1/0/49:1
device(conf-hw-flex-1/0/49:1)#
```

3. Set the FlexPort type to Fibre Channel.

```
device(conf-hw-flex-1/0/49:1)# type FibreChannel
```

4. Optionally, you may adjust the speed for the connector group. For example, if you want the connector group 7 to function at 16 Gbps:

```
device(conf-hw-flex-1/0/49:1)# connector-group 1/0/7
device(config-connector-group-1/0/49:1)# speed HighMixed
```

5. Repeat these steps for additional switches as needed.
6. Confirm the FlexPort configuration with the **show running-config hardware flexport** command.

```
device# show running-config hardware flexport
Hardware
 flexport 1/0/1
    type FibreChannel
!
Hardware
flexport 1/0/49:1
    type FibreChannel
!
connector-group 1/0/7
  speed HighMixed
!
```

Metro VCS

Metro VCS overview

Metro VCS allows you to interconnect different locations and form clusters of data centers over long distance in order to provide disaster protection/recovery and load sharing.

NOTE
Metro VCS is not supported by IP Fabrics.

In cases where distances are moderate, within 80 km, and where either dedicated fiber or transparent wavelength services are available, the Metro VCS approach is a good and cost-effective Layer 2 interconnect solution. Because of the multi-pathing capabilities of the TRILL-based Metro VCS solution, there is no issue with topology loops between multiple DC locations.

For longer distances alternative solutions are available.

- **Interconnecting separate fabrics through Layer 2 point-to-point connectivity** -- Layer 2 point-to-point connectivity is used to interconnect VCS Fabrics using their edge ports. If more than one Layer 2 link is needed for capacity or for redundancy reasons, link aggregation (LAG/vLAG) can be used in order to avoid loops between the VCS Fabric edge ports and allow for active/active protection.
- **Interconnecting separate fabrics through Layer 2-VPN connectivity** -- VCS Fabrics are interconnected through their edge ports using Layer 2-VPN connectivity. This can be implemented with Layer 2-VPN services from Connectivity Service Providers or using VPLS functionality implemented on Extreme MLX routers.

FIGURE 21 Metro VCS configuration example

NOTE
The topologies included here are high-level examples only. For detailed topologies and deployment information, refer to "Pre-Deployment Guide: Metro VCS and Distributed Ethernet Fabrics" for this release series, in conjunction with Extreme technical support.

Both options are distance independent in relation to the speed of the protocols used (for example, LAG/vLAG is a slow protocol) and provide flat Layer 2 interconnection between multiple locations. In the case of more than two DC locations, care needs to be taken in order to avoid any topology loops.

Metro VCS details and configuration

Metro VCS allows for interconnection of up to five locations and allows to form clusters of data centers over metro distances (up to 80 km with appropriate optics), in order to provide disaster protection/recovery and application load sharing.

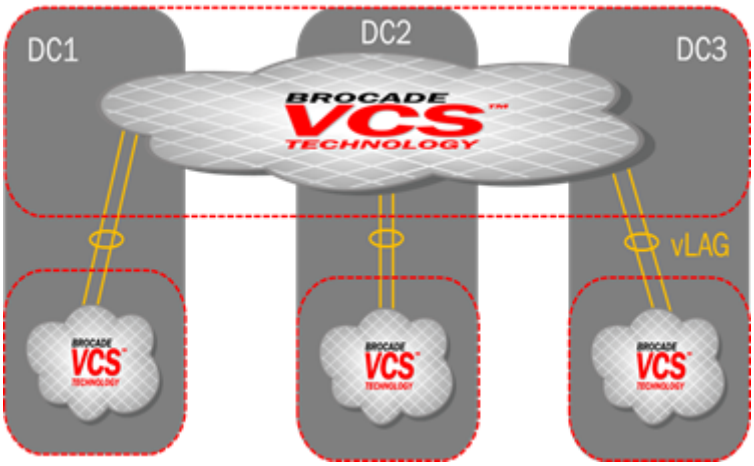
By using Inter-Switch Links (ISLs) over longer distances (more than the standard 1000 m), Ethernet fabrics can be distributed across data centers located in geographically different locations.

NOTE
For simplicity, only three data centers are shown below. However, up to five data centers are supported.

FIGURE 22 Basic Metro VCS configuration

If more complex setups are needed within different locations, then local VCS Fabrics can be used interconnected over a stretched interconnect fabric.

FIGURE 23 Complex Metro VCS configuration



Standard VCS Fabrics scaling limitations apply if Inter-Switch Links (ISLs) are used over distances of up to 1000 m. Using Inter-Switch Links (ISLs) over longer distances (more than the standard 1000 m) is currently available for 10G, 40G, and 100G ISLs and can be done in two ways:

- **Using standard ISLs over longer distances**—This only works for restricted topologies (a maximum of ten nodes and five locations) and for standard Ethernet (lossless Ethernet capabilities, such as lossless iSCSI, cannot be used in this case). The

standard 10G ISL can be used over distances up to 80 km (with appropriate optics), and standard 40G and 100G ISLs can support distances up to 40 km.

- **Configuring long-distance ISLs (LD-ISLs)** —This is supported only on 10G ISLs and does not restrict fabric topologies (such as the numbers of nodes and number of locations) beyond the standard VCS Fabric scalability. Long-distance ISLs can be used over distances up to 30 km. Links up to 10 km are lossless for DCB services.

Metro VCS using standard-distance ISLs

In order to deploy Metro VCS using standard-distance ISLs, no configuration is required on the ISL.

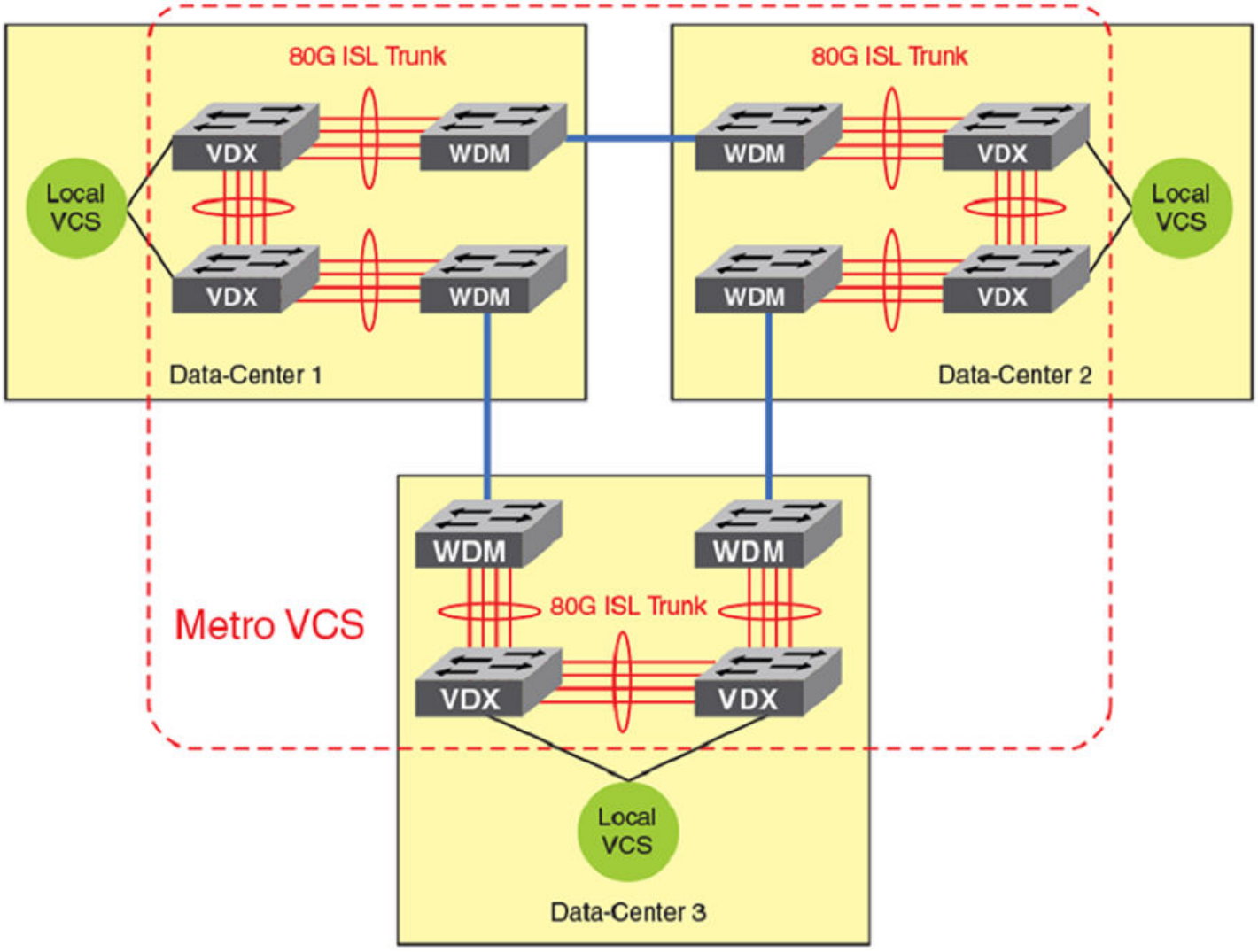
The default configuration on the 10-, 40-, and 100-Gbps interface by means of the **fabric isl enable** and **fabric trunk enable** commands allows ISL formation with other Extreme VDX switches in the same VCS cluster automatically. BLDP negotiation takes place to form ISLs for distances up to 80 km for 10G and 40 km for 40G and 100G interfaces.

Metro VCS using standard ISLs is supported on the following platforms:

- Extreme VDX 2741
- Extreme VDX 2746
- Extreme VDX 6740, VDX 6740T, and VDX 6740T-1G
- Extreme VDX 8770 with the LC48X10G line card
- Extreme VDX 8770 with the LC27X40G line card
- Extreme VDX 8770 with the LC12X40G line card
- Extreme VDX 8770 with the LC6X100G line card
- Extreme VDX 6940-36Q
- Extreme VDX 6940-144S

The following figure is an example deployment topology supported for interconnecting data centers by extending Extreme VCS Ethernet fabrics using standard-distance ISLs. The local VCS clusters are connected to the Metro VCS clusters by Extreme vLAGs. In this case, local data-center Ethernet fabrics from both site are not merged while providing seamless Layer 2 extension. For Metro VCS, Extreme standard-distance ISL supports trunks up to a maximum of 80 Gbps for the Extreme VDX 8770 and Extreme VDX 6740 series platforms, or 120 Gbps for x`Extreme VDX 6940 series platforms.

FIGURE 24 Typical deployment topology for Metro VCS using standard-distance ISLs



The following table lists the port groups and number of port groups available on each platform for Metro VCS using standard-distance ISLs.

TABLE 25 Standard Metro VCS port-group schema

Platform	Port groups	Number of port groups on platform
Extreme VDX 6740	1-16, 17-32, 33-40, 41-48, 49-50, 51-52	6
Extreme VDX 6740T	49-50, 51-52 (40G interfaces only)	2
Extreme VDX 6740T-1G	49-50, 51-52 (40G interfaces only)	2
Extreme VDX 8770 (LC6X100G)	1-2, 3-4, 5-6	3
Extreme VDX 8770 (LC27X40G)	1-3, 4-6, 7-9, 10-12, 13-15, 16-18, 19-21, 22-24, 25-27	9
Extreme VDX 8770 (LC12X40G)	1-2, 3-4, 5-6, 7-8, 9-10, 11-12	6
Extreme VDX 8770 (LC48x10G line card)	1–8, 9–16, 17–24, 25–32, 33–40, 41–48	6

TABLE 25 Standard Metro VCS port-group schema (continued)

Platform	Port groups	Number of port groups on platform
Extreme VDX 6940-36Q	1-6, 7-12, 13-18, 19-24, 25-30, 31-36	6
Extreme VDX 6940-144S	1-24+61-72, 25-48+73-84, 49-60+97-102, 85-96+1-3-108	4

Guidelines and restrictions for standard-distance Metro VCS

Consider the following guidelines and restrictions when configuring Metro VCS with standard-distance ISLs:

- Metro VCS with the standard-distance ISL solution supports up to 5 data center sites.
- A maximum of two nodes are supported for each site, which provides node redundancy. If more-complex local designs are required, a local VCS sub-fabric design must be used.
- Only standard Ethernet services are supported, lossless Ethernet capabilities, such as lossless iSCSI, cannot be used in this case.
- Extreme trunking up to 80G (8x10G or 2x40G) or 120G (3x40G or 12x10G):
  - VDX 8770: 8x10G
  - VDX 6740: 8x10G and 2x40G
  - VDX 6740T: 2x40G
  - VDX 6740T-1G: 2x40G
  - VDX 6940-36Q: 3x40G or 12x10G
  - VDX 6940-144S: 3x40G and 12x10G

Metro VCS using long-distance ISLs

Extending Ethernet Fabrics over distance is accomplished by using long-distance ISLs. The buffer allocation within a single port group is optimized, which extends the supported ISL distance.

Metro VCS supports long-distance ISL ports up to 30 km on the Extreme VDX platforms listed in the following table. Links up to 10 km are lossless.

TABLE 26 Limitations for long-distance Metro VCS

Supported hardware	Extended ISL up to 2 km	Extended ISL up to 5 km	Extended ISL up to 10 km	Extended ISL up to 30 km
Extreme VDX 6740	yes	yes	yes	yes
Extreme VDX 6940-144S - 10GbE interface	yes	yes	yes	yes
Extreme VDX 8770 - LC48x10G linecard	yes	yes	yes	yes

The following table lists the conditions on extended ISLs for Network OS hardware.

TABLE 27 Conditions for long-distance Metro VCS

Condition	Extended ISL up to 2 km	Extended ISL up to 5 km	Extended ISL up to 10 km	Extended ISL up to 30 km
Support for lossless iSCSI traffic on the Metro VCS port group	yes	yes	yes	no
Layer 2/IP lossy traffic support	yes	yes	yes	yes

TABLE 27 Conditions for long-distance Metro VCS (continued)

Condition	Extended ISL up to 2 km	Extended ISL up to 5 km	Extended ISL up to 10 km	Extended ISL up to 30 km
Number of Metro VCS long-distance ports supported per port group	1	1	1	1
Number of regular ISLs supported on a port group configured for long distance	1	1	0	0
Trunking support between multiple long-distance ISLs	no	no	no	no
Long-distance ISLs on copper interfaces	no	no	no	no
Support 40G to 4x10G breakout long-distance ISL	no	no	no	no
CEE map or port allowed in same port group	no	no	no	no
eNS Sync (MAC address table sync)	yes	yes	yes	yes
HA failover	yes	yes	yes	yes
Node redundancy check	yes	yes	yes	yes
vMotion	yes	yes	yes	yes
Maximum PFCs supported	3 (2 on the Extreme VDX 6740)	3(2 on the Extreme VDX 6740)	3 (2 on the Extreme VDX 6740)	3 (2 on the Extreme VDX 6740)
Long-distance ISL on 40G to 4x10G breakout interfaces	no	no	no	no
Long-distance ISL on 1G and 10G copper interfaces	no	no	no	no

The following table lists the port groups and number of port groups available on each platform for long-distance Metro VCS.

TABLE 28 Long-distance Metro VCS port-group schema

Platform	Port groups	Number of port groups on platform
Extreme VDX 6740	1-32, 33-48 (49-52 are 40G ports and do not support long distance)	2 *
Extreme VDX 6940-144S - 10GbE interface	1-24+61-72, 25-48+73-84, 49-60, 85-96	4
Extreme VDX 8770 (LC48x10G linecard)	1-8, 9-16, 17-24, 25-32, 33-40, 41-48	6 per 10GbE blade

* Not a valid deployment scenario at distances longer than 5 km, as no normal ISLs are allowed if both port groups are configured with long-distance ISLs for 10 km and 30 km.

Guidelines and restrictions for long-distance Metro VCS

Consider the following guidelines and restrictions when configuring long-distance Metro VCS:

- Long-distance-ISLs are only supported on 10G interface links.
- Only one long-distance-ISL is supported within a single port group.
- Long-distance-ISL is not supported on 10G copper RJ-45 interfaces
- Long-distance-ISL is not supported on 40G-to-10G breakout interfaces

- Extreme trunking is not supported with long-distance ISLs, but up to eight 8-link ECMP trunks can be used.
- Edge ports within the same port group where a long-distance ISL is configured cannot be configured with DCB maps.
- A maximum of three PFCs can be supported on a Metro VCS configured platform. By default, PFC is enabled by class 3 and 7.
- The Extreme VDX 6740 switches support only two PFCs.
- All the ports in the port group are rebooted when a port is configured for long distance.
- For 2-, 5-, 10-km long distance, use Extreme-supported long-range (LR) optics for direct connectivity.
- For 30-km long distance, use Extreme-supported extended-range (ER) optics for direct connectivity.
- A port group containing a long-distance port cannot have a CEE map configuration on any edge port.
- For 2-km and 5-km long-distance ISLs, one additional standard ISL connection is supported on the same long-distance port group.

Metro VCS combined with vLAGs

Outside of Metro distances, and whenever bit transparency may be a problem, edge-to-edge interconnected fabrics using 10G, 40G, or 100G vLAG over multiple standard Ethernet links can be used. This allows you to connect separate Ethernet fabrics that can be located in different data centers, even if the distance between those locations up to 80 km for 10G interfaces and up to 40 km for 40G and 100G interfaces.

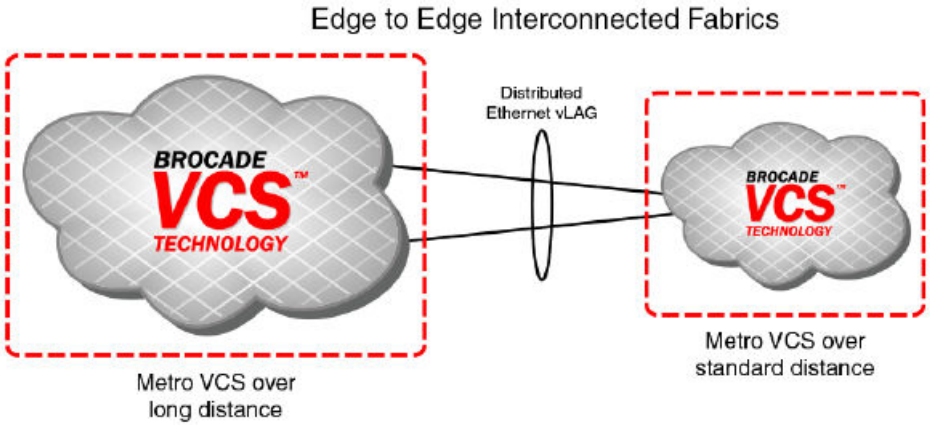
Metro VCS over a long-distance fabric

Whenever standard VCS Fabrics are interconnected through LAG/vLAG, they are not limited beyond LAG/vLAG protocol capabilities.

Metro VCS Fabrics (stretched fabrics) that are interconnected with standard fabrics are supported for distances up to 100 km.

As shown in the following figure, one side can be a Metro VCS over a long-distance fabric.

FIGURE 25 Metro VCS and distributed Ethernet fabrics



In all deployment with interconnects using edge ports, lossless Ethernet traffic (DCB) is not supported.

In order to connect two distinct VCS Ethernet fabrics between data centers, a third Metro VCS fabric can be formed, and the distinct local VCS Ethernet fabrics can connect to the Metro VCS fabric by means of Virtual Link Aggregation (vLAG).

Alternatively, the distinct VCS Ethernet fabrics in the respective data centers can be directly connected to each other by means of vLAG over xWDM up to a distance of 200 km. Wherever bit-transparency is not achievable in xWDM equipment, this solution can be successfully deployed for edge-to-edge interconnectivity (using 10G, 40G, or 100G vLAGs over multiple standard Ethernet links). This deployment is referred to as "Distributed Ethernet Fabrics using vLAG."

This implementation eliminates the need for the creation of a separate Metro VCS fabric to achieve local VCS cluster isolation while providing Layer 2 connectivity. In such a deployment, DCB lossless Ethernet traffic is not supported.

> NOTE
> When a port-channel from a node in one VCS spans across multiple RBridges in other VCS cluster, a vLAG is formed on the RBridges in the VCS cluster that are part of the same port-channel. For Distributed Ethernet Fabrics using vLAG over long distances, only LACP-based standard port-channels are supported. For details on how to create port-channels and vLAGs, refer to the “Configuring Link Aggregation” chapter of the *Extreme Network OS Layer 2 Switching Configuration Guide*.

Supported platforms for Distributed Ethernet Fabrics using vLAG

The following VDX platforms are supported for Distributed Ethernet Fabrics using vLAG:

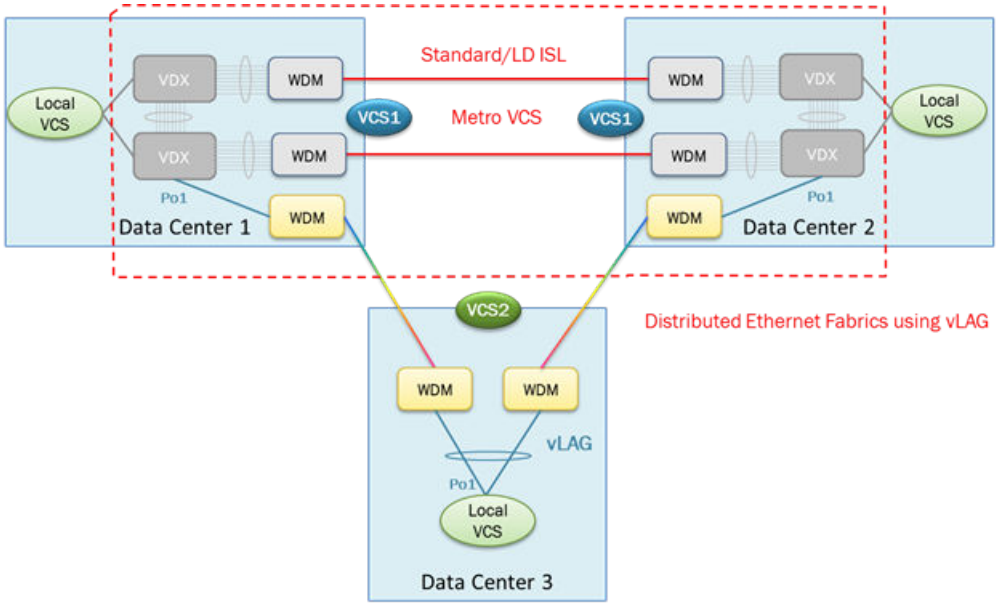
- Extreme VDX 6740, 6740T, and 6740T-1G
- Extreme VDX 8770 with the LC48X10G line card
- Extreme VDX 8770 with the LC27X40G line card
- Extreme VDX 8770 with the LC12X40G line card
- Extreme VDX 8770 with the LC6X100G line card
- Extreme VDX 6940-36Q
- Extreme VDX 6940-144S

Topology for Distributed Ethernet Fabrics using vLAG

When a port-channel from a node in one VCS spans across multiple RBridges in another VCS cluster, a vLAG is formed on the RBridges in the VCS cluster that are part of the same port-channel. For Distributed Ethernet Fabrics using vLAG over long distances, only LACP-based standard port-channels are supported. For details on how to create port-channels and vLAGs, refer to *Configuring Link Aggregation* in the *Layer 2 Switching Configuration Guide*.

The following figure is a typical deployment topology that uses distributed Ethernet Fabrics using vLAG to interconnect data centers. Nodes from the local VCS cluster are connected by means of xWDM to the nodes in a distant VCS clusters to form a vLAG in between. The distant VCS cluster can be a standard VCS cluster or could be spanned across two data centers over standard-distance or long-distance ISLs, as shown in the following figure. In this case, the vLAG between the two data centers provides VCS fabric isolation while providing seamless Layer 2 connectivity.

FIGURE 26 Connecting local VCS clusters over long distance using vLAG



Guidelines and restrictions for Distributed Ethernet Fabrics using vLAG

Note the following guidelines and restrictions for Distributed Ethernet Fabrics using vLAG.

- Only dynamic vLAG is supported.
- DCB lossless Ethernet traffic is not supported.
- The maximum distance between standard VCS Fabrics is not limited beyond the capabilities of the LAG/vLAG protocol.
- The maximum supported distance between a stretched Fabric (Metro VCS) and an additional standard VCS Fabric connected through a vLAG is limited to 200 km.

Configuring a long-distance ISL

To configure a long-distance ISL, perform the following steps in privileged EXEC mode. Each long-distance ISL port of a VCS must be connected to a long-distance ISL port on the remote VCS.

1. Verify that the default standard-distance ISL configuration is correct by using the **show running-config** command.

```
device# show running-config interface tengigabitethernet 51/0/1
interface TenGigabitEthernet 51/0/1
fabric isl enable
fabric trunk enable
no shutdown
```

2. Set the port to support Metro VCS up to 30 km by using the **long-distance-isl** command.

```
device# interface tengigabit 51/0/1
device(conf-if-te-51/0/1)# long-distance-isl 30000
```

3. Perform the same long-distance ISL configuration on the interface of the peer RBridge on the remote sites of the Metro VCS.

4. Verify that the long-distance ISL is correctly formed by using the **show fabric isl** and **show fabric islports** command.

```
device(conf-if-te-51/0/1)# do show fabric isl
Rbridge-id: 51   #ISLs: 1
 Src       Src           Nbr       Nbr
Index   Interface       Index   Interface              Nbr               BW   Trunk  Nbr-Name
------------------------------------------------------------------------------------------
4       Te 51/0/1        4      Te 53/0/1       10:00:00:05:33:65:3B:50   10G   Yes   "VCS3-53"
device(conf-if-te-51/0/1)# do show fabric islports
Name:       VCS3-51
Type:       131.4
State:      Online
Role:       Fabric Principal
VCS Id:     3
Config Mode:Local-Only
Rbridge-id: 51
FCF MAC:    00:05:33:e5:d0:cf
 Index   Interface      State   Operational State
=================================================================
  0     Fo 51/0/49        Down
  1     Fo 51/0/50        Down
  2     Fo 51/0/51        Down
  3     Fo 51/0/52        Down
  4     Te 51/0/1         Up    ISL 10:00:00:05:33:65:3B:50 "VCS3-53" (Trunk Primary)
<Truncated>
```

5. Use the **show ip interface brief** command to confirm the configuration, making sure that Status is "up" and Protocol is "LD ISL," as in the following example output.

```
device# show ip interface brief

Interface                    IP_Address     VRF             Status         Protocol
==========================   ===========    =============   ===========    =============
TenGigabitEthernet 51/0/1    unassigned     default-vrf     up             up (LD ISL)
```

Configuring interconnected Ethernet Fabrics

To deploy interconnected Ethernet Fabrics using vLAG, create a port-channel interface on the RBridges that are to be connected. Then add the member interfaces to the port-channel and bring them online. Configure switchport and add the VLANs that are to be allowed over the port-channel. After the port-channels on all the RBridges are online, the vLAG forms automatically on the RBridge that connects to multiple nodes on the other VCS cluster. In the deployment topology shown in Topology for Distributed Ethernet Fabrics using vLAG on page 134, the vLAG forms on the RBridges that are part of port-channel Po1 in Data-Centers 1 and 2 that forms VCS 1.

This configuration must be applied on RBridges that connect the two VCS instances. The port-channel is created only from the principal node and is applied globally.

1. Create a port-channel interface on all RBridges that are directly connected to RBridges in other VCS instances.

```
device(config)# interface port-channel 300
```

2. Verify that the port-channel is created correctly by using the **show running-config** command.

```
device(config-Port-channel-300)# do show running-config interface port-channel 300

interface Port-channel 300
 vlag ignore-split
 shutdown
```

3. Configure the port-channel interface for the switchport trunk and add the VLANs to be allowed on the trunk interface by using the **switchport** command.

```
device(config-Port-channel-300)# switchport
device(config-Port-channel-300)# switchport mode trunk
device(config-Port-channel-300)# switchport trunk allowed vlan all
```

4. Verify the port-channel interface configuration by using the **show running-config** command.

```
device(config-Port-channel-300)# do show running-config interface Port-channel 300

interface Port-channel 300
 vlag ignore-split
 switchport
 switchport mode trunk
 switchport trunk allowed vlan all
 switchport trunk tag native-vlan
 spanning-tree shutdown
 shutdown
```

5. Add member interfaces to the port-channel interface by using the **channel-group** command. Do this for all interfaces that must be part of the port-channel.

```
device(conf-if-te-53/0/31)# channel-group 300 mode active type standard
device(conf-if-te-53/0/31)# do show running-config interface

TenGigabitEthernet 53/0/31
interface TenGigabitEthernet 53/0/31
 fabric isl enable
 fabric trunk enable
 channel-group 300 mode active type standard
 lacp timeout long
 no shutdown
```

6. Bring the port-channel online in both VCS instances by executing **no shutdown** on the port-channel interface.

```
device(config-Port-channel-300)# no shutdown
```

7. Verify the port-channel interface configuration by using the **show running-config** command.

```
device(config-Port-channel-300)# do show running-config interface Port-channel 300

interface Port-channel 300
 vlag ignore-split
 switchport
 switchport mode trunk
 switchport trunk allowed vlan all
 switchport trunk tag native-vlan
 spanning-tree shutdown
 no shutdown
```

8. Verify that console RASLogs indicate the formation of the vLAG by using the **no shutdown** command.

```
device(config-Port-channel-300)# no shutdown

2013/06/17-16:40:53, [NSM-1023], 224126, DCE, INFO, VCS1-51, RBridge ID 51 has joined Port-channel
300. Port-channel is a vLAG with RBridge IDs 52 51.
```

9. Verify the formation of the port-channel vLAG by using the **show port-channel** command.

```
device# show port-channel 300
LACP Aggregator: Po 300 (vLAG)
Aggregator type: Standard
Ignore-split is enabled
  Member rbridges:
    rbridge-id: 51 (2)
    rbridge-id: 52 (2)
  Admin Key: 0010 - Oper Key 0010
  Partner System ID - 0x8000,01-e0-52-00-00-02
  Partner Oper Key 0010
Member ports on rbridge-id 51:
   Link: Te 51/0/31 (0x19180E801C) sync: 1
   Link: Te 51/0/32 (0x19180F001D) sync: 1
```

Using Diagnostic Port for link traffic tests

The Diagnostic Port (D_Port) feature allows you to convert a TRILL ISL port into a diagnostic port for testing link traffic between a pair of switches. The test results can be very useful in diagnosing a variety of port and link problems.

For details, see the "Diagnostic Port" chapter in the *Network OS Monitoring Configuration Guide*.

LLDP

LLDP overview

The IEEE 802.1AB Link Layer Discovery Protocol (LLDP) enhances the ability of network management tools to discover and maintain accurate network topologies and simplify LAN troubleshooting in multi-vendor environments. To efficiently and effectively operate the various devices in a LAN you must ensure the correct and valid configuration of the protocols and applications that are enabled on these devices. With Layer 2 networks expanding dramatically, it is difficult for a network administrator to statically monitor and configure each device in the network.

Using LLDP, network devices such as routers and switches advertise information about themselves to other network devices and store the information they discover. Details such as device configuration, device capabilities, and device identification are advertised. LLDP defines the following:

- A common set of advertisement messages.
- A protocol for transmitting the advertisements.
- A method for storing the information contained in received advertisements.

NOTE
LLDP runs over the data-link layer which allows two devices running different network layer protocols to learn about each other.

LLDP information is transmitted periodically and stored for a finite period. Every time a device receives an LLDP advertisement frame, it stores the information and initializes a timer. If the timer reaches the time to live (TTL) value, the LLDP device deletes the stored information ensuring that only valid and current LLDP information is stored in network devices and is available to network management systems.

Layer 2 topology mapping

The LLDP protocol lets network management systems accurately discover and model Layer 2 network topologies.

As LLDP devices transmit and receive advertisements, the devices store information they discover about their neighbors. Advertisement data such as a neighbor's management address, device type, and port identification is useful in determining what neighboring devices are in the network.

NOTE
The Extreme LLDP implementation supports up to two neighbors.

The higher level management tools, such as the Extreme Network Advisor, can query the LLDP information to draw Layer 2 physical topologies. The management tools can continue to query a neighboring device through the device's management address provided in the LLDP information exchange. As this process is repeated, the complete Layer 2 topology is mapped.

In LLDP the link discovery is achieved through the exchange of link-level information between two link partners. The link-level information is refreshed periodically to reflect any dynamic changes in link-level parameters. The basic format for exchanging information in LLDP is in the form of a type, length, value (TLV) field.

LLDP keeps a database for both local and remote configurations. The LLDP standard currently supports three categories of TLVs. The Extreme LLDP implementation adds a proprietary Extreme extension TLV set. The four TLV sets are described as follows:

- Basic management TLV set — This set provides information to map the Layer 2 topology and includes the following TLVs:
  - Chassis ID TLV — Provides the ID for the switch or router where the port resides. This is a mandatory TLV.
  - Port description TLV — Provides a description of the port in an alphanumeric format. If the LAN device supports RFC-2863, the port description TLV value equals the "ifDescr" object. This is a mandatory TLV.
  - System name TLV — Provides the system-assigned name in an alphanumeric format. If the LAN device supports RFC-3418, the system name TLV value equals the "sysName" object. This is an optional TLV. If the LLDP advertisement is split into multiple PDUs, the system name will be available in each LLDP PDU.
  - System description TLV — Provides a description of the network entity in an alphanumeric format. This includes system name, hardware version, operating system, and supported networking software. If the LAN device supports RFC-3418, the value equals the "sysDescr" object. This is an optional TLV.
  - System capabilities TLV — Indicates the primary functions of the device and whether these functions are enabled in the device. The capabilities are indicated by two octets. The first octet indicates Other, Repeater, Bridge, WLAN AP, Router, Telephone, DOCSIS cable device, and Station, respectively. The second octet is reserved. This is an optional TLV.
  - Management address TLV — Indicates the addresses of the local switch. Remote switches can use this address to obtain information related to the local switch. This is an optional TLV.
- IEEE 802.1 organizational TLV set — This set provides information to detect mismatched settings between local and remote devices. A trap or event can be reported once a mismatch is detected. This is an optional TLV. This set includes the following TLVs:
  - Port VLANID TLV — Indicates the port VLAN ID (PVID) that is associated with an untagged or priority tagged data frame received on the VLAN port.
  - PPVLAN ID TLV — Indicates the port- and protocol-based VLAN ID (PPVID) that is associated with an untagged or priority tagged data frame received on the VLAN port. The TLV supports a "flags" field that indicates whether the port is capable of supporting port- and protocol-based VLANs (PPVLANs) and whether one or more PPVLANs are enabled. The number of PPVLAN ID TLVs in a Link Layer Discovery Protocol Data Unit (LLDPDU) corresponds to the number of the PPVLANs enabled on the port.
  - VLAN name TLV — Indicates the assigned name of any VLAN on the device. If the LAN device supports RFC-2674, the value equals the "dot1QVLANStaticName" object. The number of VLAN name TLVs in an LLDPDU corresponds to the number of VLANs enabled on the port.
  - Protocol identity TLV — Indicates the set of protocols that are accessible at the device's port. The protocol identity field in the TLV contains a number of octets after the Layer 2 address that can enable the receiving device to recognize the protocol. For example, a device that wishes to advertise the spanning tree protocol includes at least eight octets: 802.3 length (two octets), LLC addresses (two octets), 802.3 control (one octet), protocol ID (two octets), and the protocol version (one octet).
- IEEE 802.3 organizational TLV set — This is an optional TLV set. This set includes the following TLVs:
  - MAC/PHY configuration/status TLV — Indicates duplex and bit rate capabilities and the current duplex and bit rate settings of the local interface. It also indicates whether the current settings were configured through auto-negotiation or through manual configuration.
  - Power through media dependent interface (MDI) TLV — Indicates the power capabilities of the LAN device.
  - Link aggregation TLV — Indicates whether the link (associated with the port on which the LLDPDU is transmitted) can be aggregated. It also indicates whether the link is currently aggregated and provides the aggregated port identifier if the link is aggregated.
  - Maximum Ethernet frame size TLV — Indicates the maximum frame size capability of the device's MAC and PHY implementation.

DCBX

Storage traffic requires a lossless communication which is provided by DCB. The Data Center Bridging (DCB) Capability Exchange Protocol (DCBX) is used to exchange DCB-related parameters with neighbors to achieve more efficient scheduling and a priority-based flow control for link traffic.

DCBX uses LLDP to exchange parameters between two link peers; DCBX is built on the LLDP infrastructure for the exchange of information. DCBX-exchanged parameters are packaged into organizationally specific TLVs. The DCBX protocol requires an acknowledgment from the other side of the link, therefore LLDP is turned on in both transmit and receive directions. DCBX requires version number checking for both control TLVs and feature TLVs.

DCBX interacts with other protocols and features as follows:

- *LLDP* — LLDP is run in parallel with other Layer 2 protocols such as RSTP and LACP. DCBX is built on the LLDP infrastructure to communicate capabilities supported between link partners. The DCBX protocol and feature TLVs are treated as a superset of the LLDP standard.
- *QoS management* — DCBX capabilities exchanged with a link partner are passed down to the QoS management entity to set up the Extreme VDX hardware to control the scheduling and priority-based flow control in the hardware.

The DCBX QoS standard is subdivided into two features sets:

- Enhanced Transmission Selection on page 141
- Priority Flow Control on page 142

Enhanced Transmission Selection

In a converged network, different traffic types affect the network bandwidth differently. The purpose of Enhanced Transmission Selection (ETS) is to allocate bandwidth based on the different priority settings of the converged traffic.

For example, Inter-process communications (IPC) traffic can use as much bandwidth as needed and there is no bandwidth check; LAN and SAN traffic share the remaining bandwidth. The following table displays three traffic groups: IPC, LAN, and SAN. ETS allocates the bandwidth based on traffic type and also assigns a priority to the three traffic types as follows: Priority 7 traffic is mapped to priority group 0 which does not get a bandwidth check, priority 2 and priority 3 are mapped to priority group 1, priorities 6, 5, 4, 1 and 0 are mapped to priority group 2.

The priority settings shown in the following table are translated to priority groups in the Extreme VDX hardware.

TABLE 29 ETS priority grouping of IPC, LAN, and SAN traffic

Priority	Priority group	Bandwidth check
7	0	No
6	2	Yes
5	2	Yes
4	2	Yes
3	1	Yes
2	1	Yes
1	2	Yes
0	2	Yes

Priority Flow Control

With Priority Flow Control (PFC), it is important to provide lossless frame delivery for certain traffic classes while maintaining existing LAN behavior for other traffic classes on the converged link. This differs from the traditional 802.3 PAUSE type of flow control where the pause affects all traffic on an interface.

PFC is defined by a one-byte bitmap. Each bit position stands for a user priority. If a bit is set, the flow control is enabled in both directions (Rx and Tx).

NOTE
When PFC is enabled, the Extreme VDX 6740 series platforms support up to three PGIDs with the execution of **cee-map default** . By default, PGID 1 (with TC3) and PGID 15.0 (for network control traffic) are enabled when PFC is enabled.

LLDP configuration guidelines and restrictions

Follow these LLDP configuration guidelines and restrictions when configuring LLDP:

- The Extreme implementation of LLDP supports Extreme-specific TLV exchange in addition to the standard LLDP information.
- Mandatory TLVs are always advertised.
- The exchange of LLDP link-level parameters is transparent to the other Layer 2 protocols. The LLDP link-level parameters are reported by LLDP to other interested protocols.

NOTE
DCBX configuration involves configuring DCBX-related TLVs to be advertised.

Configuring and managing LLDP

The following sections discuss working with the Link Layer Discovery Protocol (LLDP) on Extreme devices.

Understanding the default LLDP

The following table lists the default LLDP configuration. Consider this when making changes to the defaults.

TABLE 30 Default LLDP configuration

Parameter	Default setting
LLDP global state	Enabled
LLDP receive	Enabled
LLDP transmit	Enabled
Transmission frequency of LLDP updates	30 seconds
Hold time for receiving devices before discarding	120 seconds
DCBX-related TLVs to be advertised	dcbx-tlv

Enabling LLDP globally

The **protocol lldp** command enables LLDP globally on all interfaces unless it has been specifically disabled on an interface, or the global LLDP disable command has been executed. LLDP is globally enabled by default.

To enable LLDP globally, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to change to global configuration mode.

```
device# configure terminal
```

2. Enter LLDP configuration mode.

```
device(config)# protocol lldp
```

Disabling LLDP globally

LLDP is enabled globally by default. You can disable LLDP globally without changing any other aspect of the LLDP configuration.

To globally disable LLDP, perform the following steps:

1. From privileged EXEC mode, access global configuration mode.

```
device# configure terminal
```

2. Access LLDP configuration mode.

```
device(config)# protocol lldp
```

3. Disable LLDP globally.

```
device(config-lldp)# disable
```

The following configuration is an example of the previous steps to disable LLDP.

```
device# configure terminal
device(config)# protocol lldp
device(config-lldp)# disable
```

If required, re-enable LLDP.

```
device(config-lldp)# no disable
```

Resetting LLDP globally

The **no protocol lldp** command returns all configuration settings made using the protocol LLDP commands to their default settings.

To reset LLDP globally, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Reset LLDP globally.

```
device(config)# no protocol lldp
```

Configuring LLDP global command options

After entering the **protocol lldp** command from global configuration mode, you are in LLDP configuration mode. Using the keywords in this mode, you can set nondefault parameter values that apply globally to all interfaces.

Specifying a system name for the Extreme VDX hardware

The global system name for LLDP is useful for differentiating between switches. By default, the "host-name" from the chassis/entity management information base is used. By specifying a descriptive system name, you will find it easier to configure the switch for LLDP.

To specify a global system name for the Extreme VDX hardware, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
device(config)# protocol lldp
```

3. Specify an LLDP system name for the DCB switch.

```
device(conf-lldp)# system-name Brocade_Alpha
Brocade_Alpha(conf-lldp)#
```

Specifying an LLDP system description for the Extreme VDX hardware

NOTE
Extreme recommends you use the operating system version for the description or use the description from the chassis/entity management information base (MIB). Do not use special characters, such as #$!@, as part of the system name and description.

To specify an LLDP system description for the Extreme VDX hardware, perform the following steps from privileged EXEC mode. The system description is seen by neighboring switches.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
device(config)# protocol lldp
```

3. Specify a system description for the Extreme VDX hardware.

```
device(conf-lldp)# system-description IT_1.6.2_LLDP_01
```

Specifying a user description for LLDP

To specify a user description for LLDP, perform the following steps from privileged EXEC mode. This description is for network administrative purposes and is not seen by neighboring switches.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
device(config)# protocol lldp
```

3. Specify a user description for LLDP.

```
device(conf-lldp)# description Brocade-LLDP-installed-jan-25
```

Enabling and disabling the receiving and transmitting of LLDP frames

By default both transmit and receive for LLDP frames are enabled. To enable or disable the receiving (rx) and transmitting (tx) of LLDP frames, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
device(config)# protocol lldp
```

3. Enter the **mode** command to do one of the following:
   - Enable only receiving of LLDP frames:

```
device(conf-lldp)# mode rx
```

   - Enable only transmitting of LLDP frames:

```
device(conf-lldp)# mode tx
```

   - Enable both transmit and receive modes.

```
device(conf-lldp)# no mode
```

Configuring the transmit frequency of LLDP frames

To configure the transmit frequency of LLDP frames, perform the following steps from privileged EXEC mode. The default is 30 seconds.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
device(config)# protocol lldp
```

3. Configure the transmit frequency of LLDP frames.

```
device(conf-lldp)# hello 45
```

Configuring the hold time for receiving devices

To configure the hold time for receiving devices, perform the following steps from privileged EXEC mode. This configures the number of consecutive LLDP hello packets that can be missed before removing the neighbor information. The default is 4.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
device(config)# protocol lldp
```

3. Configure the hold time for receiving devices.

```
device(conf-lldp)# multiplier 6
```

Advertising the optional LLDP TLVs

To advertise the optional LLDP TLVs, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
device(config)# protocol lldp
```

3. Advertise the optional LLDP TLVs.

```
device(conf-lldp)# advertise optional-tlv management-address port-description system-capabilities
system-name system-description
```

Configuring the advertisement of LLDP DCBX-TLV

For a device in Extreme VCS Fabric mode the dcbx-tlv is advertised by default.

To configure the LLDP DCBX-TLV to be advertised, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
device(config)# protocol lldp
```

3. Advertise the LLDP DCBX-TLV using the following command:

```
device(conf-lldp)# advertise dcbx-tlv
```

Configuring LLDP profiles

The Extreme device supports up to 64 profiles. When you configure a profile, its default parameters are from the global LLDP configuration.

1. In privileged EXEC mode, access global configuration mode.

```
device# configure terminal
```

2. Enter LLDP configuration mode.

```
device(config)# protocol lldp
```

3. Configure the profile name.

```
device(conf-lldp)# profile UK_LLDP_IT
```

4. Specify a description for the profile.

```
device(conf-lldp-profile-UK_LLDP_IT)#description standard_profile_by_Jane
```

5. Configure the transmission frequency of LLDP updates.

```
device(conf-lldp-profile-UK_LLDP_IT)# hello 10
```

6. Configure the hold time for receiving devices.

```
device(conf-lldp-profile-UK_LLDP_IT)# multiplier 2
```

7. Advertise the optional LLDP TLVs.

```
device(conf-lldp)# advertise optional-tlv system-name
```

8. Advertise the LLDP organizationally-specific TLVs.

```
device(conf-lldp-profile-UK_LLDP_IT)# advertise advertise dot1-tlv
```

NOTE
Extreme does not recommend advertising dot1.tlv and dot3.tlv LLDPs if your network contains CNAs from non-Extreme vendors. Functionality problems can occur.

9. Return to privileged EXEC mode.

```
device(conf-lldp-profile-UK_LLDP_IT)# end
```

10. Verify the configuration.

```
device# show running-config protocol lldp profile
 profile UK_LLDP_IT
  hello 10
  multiplier 2
  advertise dot1-tlv
  advertise option-tlv system-name
  description standard_profile_by_Jane
```

The following configuration is an example of the previous steps.

```
device# configure terminal
device(config)# protocol lldp
device(conf-lldp)# profile UK_LLDP_IT
device(conf-lldp-profile-UK_LLDP_IT)# description standard_profile_by_Jane
device(conf-lldp-profile-UK_LLDP_IT)# hello 10
device(conf-lldp-profile-UK_LLDP_IT)# multiplier 2
device(conf-lldp-profile-UK_LLDP_IT)# advertise option-tlv system-name
device(conf-lldp-profile-UK_LLDP_IT)# advertise advertise dot1-tlv
```

Configuring iSCSI priority

The iSCSI TLV is used only to advertise the iSCSI traffic configuration parameters to the attached CEE enabled servers and targets. No verification or enforcement of the usage of the advertised parameters by the iSCSI server or target is done by the switch. The iSCSI priority setting is used to configure the priority to be advertised in the DCBx iSCSI TLV.

To configure the iSCSI priority, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
device(config)# protocol lldp
```

3. Configure the iSCSI priority.

```
device(conf-lldp)# iscsi-priority 4
```

NOTE
The default iscsi-priority is 4 and does not display unless you change the iscsi-priority to a different value.

4. Advertise the TLV.

```
device(conf-lldp)# advertise dcbx-iscsi-app-tlv
```

Configuring the iSCSI profile

You can configure an iSCSI profile to be applied to individual interfaces. However, the priority bit must be set manually for each interface.

To configure iSCSI profiles, perform the following steps from privileged EXEC mode.

1. Configure the CEE map, if it has not already been created.

```
device(config)# cee-map default
device(conf-ceemap)# priority-group-table 1 weight 50 pfc
device(conf-ceemap)# priority-group-table 2 weight 30 pfc on
device(conf-ceemap)# priority-group-table 3 weight 20 pfc on
device(conf-ceemap)# priority-table 1 1 1 1 2 3 1 1
```

   The **priority-table** command syntax is as follows:

```
priority-table PGID0 PGID1 PGID2 PGID3 PGID4 PGID5 PGID6 PGID7
```

   For all PGID values, the PGID value range is 0 through 7 for the DWRR Priority Group, and 15.0 through 15.7 for the Strict Priority Group. The PGID value and the CoS value are equivalent, so that specifying PGID0 sets the Priority Group ID for all packets with CoS = 0, specifying PGID1 sets the Priority Group ID for all packets with CoS = 1, all the way through specifying PGID7, which sets the Priority Group ID for all packets with CoS = 7.

   Priority-Table in CEE map configuration requires that PGID 15.0 is dedicated for CoS7. Because of this restriction, make sure that PGID15.0 is configured only as the last parameter for Priority-Table configuration.

   An explanation of syntax "priority-table 1 2 2 2 2 2 2 15.0" is as follows:

   This shows the definition of a CEE Map with Priority to Priority Group mapping of CoS=1, CoS=2, CoS=3, CoS=4, CoS=5, and CoS=6 to a DWRR Priority Group ID of 2, and CoS=0 to a Priority Group ID of 1, and CoS=7 to a Strict Priority Group.

   This is one way to provision the CEE Priority to Priority Group Table, which maps each of the eight ingress CoS into a Priority Group.

2. Enter LLDP configuration mode.

```
device(conf-ceemap)# protocol lldp
```

3. Create an LLDP profile for iSCSI.

```
device(conf-lldp)# profile iscsi_config
```

4. Advertise the iSCSI TLV.

```
device(conf-lldp-profile-iscsi_config)# advertise dcbx-iscsi-app-tlv
```

5. Enter configuration mode for the specific interface.

```
device (conf-lldp-profile-iscsi_config)# interface te 5/0/1
```

6. Apply the CEE provisioning map to the interface.

```
device(conf-if-te-5/0/1)# cee default
```

7. Apply the LLDP profile you created for iSCSI.

```
device(conf-if-te-5/0/1)# lldp profile iscsi_config
```

8. Set the iSCSI priority bits for the interface.

```
device(conf-if-te-5/0/1)# lldp iscsi-priority 4
```

9. Repeat steps 5 through 8 for additional interfaces.

Configuring LLDP interface-level command options

Only one LLDP profile can be assigned to an interface. If you do not use the **lldp profile** option at the interface level, the global configuration is used on the interface. If there are no global configuration values defined, the global default values are used.

To configure LLDP interface-level command options, perform the following steps from privileged EXEC mode.

1. Enter the **interface** command, specifying the DCB interface type and RBridge/slot/port.

```
device(config)# interface tengigabitethernet 5/0/10
```

2. Apply an LLDP profile to the interface.

```
device(conf-if-te-5/0/10)# lldp profile network_standard
```

3. Return to privileged EXEC mode.

```
device(conf-if-te-5/0/10)# end
```

Displaying LLDP-related information

To display LLDP-related information, perform one or more of the following steps from privileged EXEC mode.

1. To display LLDP general information, use the **show lldp** command.

```
device# show lldp
```

2. To display LLDP interface-related information, use the **show lldp interface** command.

```
device# show lldp interface tengigabitethernet 22/0/1
LLDP information for gi 22/0/1
  State:                   Enabled
  Mode:                    Receive/Transmit
  Advertise Transmitted:   30 seconds
  Hold time for advertise: 120 seconds
  Re-init Delay Timer:     2 seconds
  Tx Delay Timer:          1 seconds tengigabitethernet
  DCBX Version :           CEE
  Auto-Sense :             Yes
  Transmit TLVs:           Chassis ID          Port ID
                           TTL                 IEEE DCBX
                           Link Prim           Extreme Link
```

3. To display LLDP neighbor-related information, use the **show lldp neighbors** command.

```
device# show lldp neighbors interface tengigabitethernet 22/0/1 detail
Neighbors for Interface Te 22/0/1

MANDATORY TLVs
===============
Local Interface: Te 22/0/1  (Local Interface MAC: 0027.f854.501e)
Remote Interface: TenGigabitEthernet 3/0/1 (Remote Interface MAC: 0005.334b.7198)
Dead Interval: 120 secs
Remaining Life : 117 secs
Chassis ID: 0005.334b.7173
LLDP PDU Transmitted: 1165  Received: 1164

OPTIONAL TLVs
==============
DCBX TLVs
==============
Version : CEE
DCBX Ctrl OperVersion: 0  MaxVersion: 0  SeqNo: 2 AckNo: 2
DCBX ETS OperVersion: 0  MaxVersion: 0  Enabled: 1  Willing: 0 Error: 0
Enhanced Transmission Selection (ETS)
  Priority-Group ID Map:
    Priority : 0  1  2  3  4  5  6  7
    Group ID : 0  0  0  0  0  0  0  0
  Group ID Bandwidth Map:
    Group ID  :   0   1   2   3   4   5   6   7
    Percentage:   0   0   0   0   0   0   0   0
    Number of Traffic Classes supported: 8
DCBX PFC OperVersion: 0  MaxVersion: 0  Enabled: 1  Willing: 0 Error: 0
Priority-based Flow Control (PFC)
     Enabled Priorities: none
     Number of Traffic Class PFC supported: 8
LAN LLS OperVersion: 0  MaxVersion: 0  Enabled: 1  Willing: 0 Error: 0
LAN  Logic Link Status: Up
```

Clearing LLDP-related information

To clear LLDP-related information, perform the following steps from privileged EXEC mode.

1. Use the **clear** command to clear LLDP neighbor information.

```
device# clear lldp neighbors interface tengigabitethernet 0/1
```

2. Use the **clear** command to clear LLDP statistics.

```
device# clear lldp statistics interface tengigabitethernet 0/1
```

Network Time Protocol

Network Time Protocol overview

Network Time Protocol (NTP) maintains uniform time across all devices in a network. The NTP commands support the configuration of an external time server to maintain synchronization between all local clocks in a network.

To keep the time in your network current, it is recommended that each device have its time synchronized with at least one external NTP server. External NTP servers should be synchronized among themselves in order to maintain fabric-wide time synchronization.

All devices in the fabric maintain the current clock server value in nonvolatile memory. By default, this value is the local clock server of the switch.

Date and time settings

Extreme devices maintain the current date and time inside a battery-backed real-time clock (RTC) circuit. Date and time are used for logging events. Device operation does not depend on the date and time; a device with incorrect date and time settings can function correctly. However, because the date and time are used for logging, error detection, and troubleshooting, you should set them correctly.

Time zone settings

The time zone setting has the following characteristics:

- The setting automatically adjusts for Daylight Savings Time.
- Changing the time zone on a device updates the local time zone setup and is reflected in local time calculations.
- By default, all switches are in the Greenwich Mean Time (GMT) time zone (0,0). If all switches in a fabric are in one time zone, it is possible for you to keep the time zone setup at the default setting.
- System services that have already started will reflect the time zone changes only after the next reboot.
- Time zone settings persist across failover for high availability.
- Time zone settings are not affected by NTP server synchronization.

Configuring NTP

The following sections discuss how to correctly configure the Network Time Protocol for Extreme switches.

Configuration considerations for NTP

Network time synchronization is guaranteed only when a common external time server is used by all switches. When the **ntp server** command is invoked on one switch in a cluster, the configuration is applied to all switches in the cluster.

The **ntp server** command accepts up to five server addresses in IPv4 or IPv6 format. When you configure multiple NTP server addresses, the **ntp server** command sets the first obtainable address as the active NTP server. If there are no reachable time servers, then the local switch time is the default time until a new active time server is configured.

Setting the date and time

The **clock set** command sets the local clock date and time.

The **clock set** command sets the local clock date and time. Valid date and time values must be in the range between January 1, 1970 and January 19, 2038. If a time zone is not configured, the time zone defaults to Greenwich Mean Time (GMT). If an active NTP server is configured for the switch, it overrides the local time settings.

Enter the **clock set** *CCYY-MM-DDTHH:MM:SS* command.

The variables represent the following values:

- *CCYY* specifies the year; the valid range is 1970 through 2038.
- *MM* specifies the month; the valid range is 01 through 12.
- *DD* specifies the day; the valid range is 01 through 31.
- *HH* specifies the hour; the valid range is 00 through 23.
- *MM* specifies the minutes; the valid range is 00 through 59.
- *SS* specifies the seconds; the valid range is 00 through 59.

If you are in VCS mode, setting the time and date is done using the RBridge ID of the node.

Here is an example of setting and displaying the date and time in VCS mode:

```
device# clock set 2013-06-06T12:15:00 rbridge-id all
device# show clock
rbridge-id all: 2013-06-06 12:15:05 Etc/GMT+0
```

Setting the time zone

Use the **clock timezone** command to set the time zone for a switch. You must use the command for all switches for which a time zone must be set. However, you only need to set the time zone once on each switch, because the value is written to nonvolatile memory.

Setting the time and date can be done in Privileged EXEC mode by using the RBridge ID of the node. (Setting the date and time can also be done in RBridge ID configuration mode, but must be done on a per-node basis in this mode.) Refer to the **clock timezone** command in each mode in the *Network OS Command Reference*.

Refer to the *Network OS Troubleshooting Guide* for a complete list of configurable regions and cities.

Enter the **clock timezone** *region/city* command.

```
device# clock timezone America/Los_Angeles rbridge-id all
```

NOTE

Upgrade considerations: The existing timezone of the system is retained after a firmware upgrade, and it will be updated in configuration settings.

General downgrade considerations: The existing timezone of the system is retained after firmware downgrade and the respective entry will be removed from configuration settings.

Russian time zone downgrade considerations: Because of changes to the Russian time zone in Network OS 6.0.1, if you want to downgrade to a release prior to 6.0.1 and you have Russian time zones configured, be aware that the downgrade is blocked if the running configuration includes the new Russian time zone configuration.

Displaying the current local clock and time zone

The **show clock** command returns the local time, date, and time zone.

> NOTE
> This command is currently supported on the local switch.

This example shows the local switch clock time:

```
device# show clock
rbridge-id 1: 2012-05-04 16:01:51 America/Los Angeles
```

This example shows the clock time for all switches in the cluster:

```
device# show clock rbridge-id all
rbridge-id 1: 2013-06-06 12:15:05 Etc/GMT+0
rbridge-id 5: 2013-06-06 12:15:05 Etc/GMT+0
rbridge-id 10: 2013-06-06 12:15:05 Etc/GMT+0
```

This example shows the clock time for the switch with rbridge-id 16:

```
device# show clock rbridge-id 16
rbridge-id 16: 2012-05-04 18:18:51 America/Los Angeles
```

Removing the time zone setting

Use the **no clock timezone** command to remove the time zone setting for the local clock. This operation returns the local time zone to the default value (GMT). When using the **no** operand, you do not need to reference a timezone setting.

Enter the **no clock timezone** command.

```
device# no clock timezone rbridge-id 5
```

> NOTE
> The **clock timezone** command can be run in privileged EXEC mode, as shown in the previous example, or in RBridge ID configuration mode. Refer to the *Network OS Command Reference* for descriptions of this command in each of these modes.

Synchronizing the local time with an external source

Use the **ntp server** command to synchronize the local switch time with an NTP server. You can configure up to five IP address. At least one IP address in the list must be a reachable, configured NTP server or the request will fail.

The following example synchronizes the time on the local switch with the ntp server at 192.168.10.1.

Enter the **ntp server** *ip_address* command.

```
device(config)# ntp server 192.168.10.1
```

Displaying the active NTP server

Information about the currently active NTP server can be displayed. When an NTP server has been configured, the server IP address is displayed. If an NTP server is not configured or the server is unreachable, the output displays LOCL (for local device time).

Only the local NTP server information is displayed.

If the RBridge ID parameter is not provided, status results default to the local switch. If **rbridge-id all** is specified, the command displays the status for all switches in the cluster. If the RBridge ID is specified, the command displays that node's NTP status.

NTP server status when an NTP server is not configured

The following example shows the local device NTP status when an NTP server is not configured:

```
device# show ntp status
rbridge-id 1: active ntp server is LOCL
```

NTP server status when an NTP server is configured

The following example shows the status of a configured NTP server:

```
device# show ntp status
rbridge-id 1: active ntp server is 10.31.2.81
Clock is synchronized, stratum 5, Version 3, Precision 2**24
reference time:    dd6fa3a3.2f11920f  Fri, Sep 22 2017 14:58:43.183
offset 11.392637 sec, delay 0.02592
```

NTP server status for all devices in a cluster

This example shows the NTP status for all switches in a cluster.

```
device# show ntp status rbridge-id all
rbridge-id 7: active ntp server is LOCL
```

Removing an NTP server IP address

To remove an NTP server IP address from the list of server IP addresses on a switch, enter **no ntp server** followed by the server IP address.

The following example removes the NTP server at 192.168.10.1 from the local server IP address database.

```
device(config)# no ntp server 192.168.10.1
device# show ntp status
rbridge-id 1: active ntp server is LOCL
```

ATTENTION
At least one IP address in the remaining list must be for a reachable and configured NTP server; if there is not one, the remove request will fail.

Network Time Protocol Authentication

The NTP can be configured to provide cryptographic authentication of messages with the clients/peers, and with its upstream time server.

NTP supports symmetric key scheme for authentication. The scheme uses MD5 key hash algorithm. The key-id and the calculated digest form the Message Authentication Code (MAC). When authentication is enabled on the server, it is expected that the client's request message should have valid MAC. If authentication of the client message fails, NTP replies with crypto-NAK packet.

Enabling NTP authentication

To enable Network Time Protocol (NTP) strict authentication, use the authenticate command. To disable the function, use the no form of this command.

```
device(config)# ntp authenticate
```

Syntax: [no] ntp authenticate

Defining an authentication key

To define an authentication key for Network Time Protocol (NTP), use the authentication-key command. To remove the authentication key for NTP, use the no form of this command.

```
device(config)# ntp authentication-key key-id 1 md5 moof
```

Syntax: [no] ntp authentication-key key-id [md5] keystring

The valid key-id parameter is 1 to 65535.

The md5 keyword specifies the message authentication support that is provided using the Message Digest 5 Algorithm.

SNMP

Simple Network Management Protocol overview

Simple Network Management Protocol (SNMP) is a set of protocols for managing complex networks. SNMP protocols are application layer protocols. Using SNMP, devices within a network send messages, called protocol data units (PDUs), to different parts of a network. Network management using SNMP requires three components:

- SNMP Manager
- SNMP Agent
- Management Information Base (MIB)

SNMP Manager

The SNMP Manager can communicate to the devices within a network using the SNMP protocol. Typically, SNMP Managers are network management systems (NMS) that manage networks by monitoring the network parameters, and optionally, setting parameters in managed devices. Normally, the SNMP Manager sends read requests to the devices that host the SNMP Agent, to which the SNMP Agent responds with the requested data. In some cases, the managed devices can initiate the communication, and send data to the SNMP Manager using asynchronous events called traps.

SNMP Agent

The SNMP agent is a software that resides in the managed devices in the network, and collects data from these devices. Each device hosts an SNMP Agent. The SNMP Agent stores the data, and sends these when requested by an SNMP Manager. In addition, the Agent can asynchronously alert the SNMP Manager about events, by using special PDUs called traps.

Management Information Base (MIB)

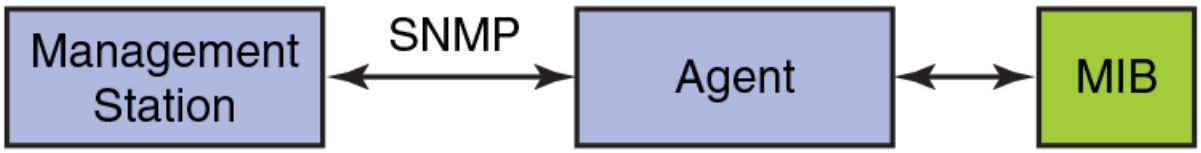
SNMP Agents in the managed devices store the data about these devices in a database called Management Information Base (MIB). The MIB is a hierarchical database, which is structured on the standard specified in the RFC 2578 [Structure of Management Information Version 2 (SMIv2)].

The MIB is a database of objects that can be used by a network management system to manage and monitor devices on the network. The MIB can be retrieved by a network management system that uses SNMP. The MIB structure determines the scope of management access allowed by a device. By using SNMP, a manager application can issue read or write operations within the scope of the MIB.

Basic SNMP operation

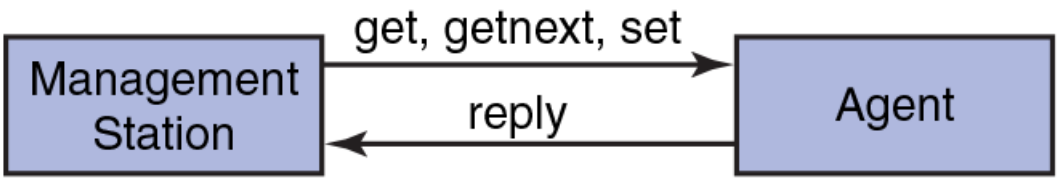
Every Extreme device carries an *agent* and management information base (MIB), as shown in the next figure. The agent accesses information about a device and makes it available to an SNMP network management station.

FIGURE 27 SNMP structure



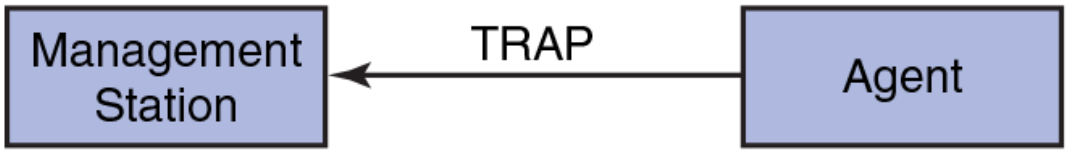
When active, the management station can "get" information or "set" information when it queries an agent. SNMP commands, such as **get**, **set**, **getnext**, and **getresponse**, are sent from the management station, and the agent replies once the value is obtained or modified as shown in the next figure. Agents use variables to report such data as the number of bytes and packets in and out of the device, or the number of broadcast messages sent and received. These variables are also known as managed objects. All managed objects are contained in the MIB.

FIGURE 28 SNMP query



The management station can also receive *traps*, unsolicited messages from the device agent if an unusual event occurs as shown in the next figure.

FIGURE 29 SNMP trap



The agent can receive queries from one or more management stations and can send traps to up to six management stations.

SNMP configuration

The following sections discuss configuring the Simple Network Management Protocol on Extreme devices. This includes configuring SNMP community strings, SNMP server hosts, SNMP server contexts, password encryption for SNMPv3 users, and displaying SNMP configurations.

> ATTENTION
> The SNMP default configurations, such as community string, user name, community, group, view configuration, and so on, have been removed, because of a security risk. The **show running-config snmp-server** command displays only system parameters during startup. You can configure the appropriate SNMP attributes by using a variety of **snmp-server** commands. (This does not apply to the Extreme VDX 2746, which displays both system and default user SNMP configurations.)

There is no change in the behavior of the upgrade from the running configuration.

Configuring SNMP community strings

SNMP versions 1 and 2c use community strings to restrict access to the switch. There is support for a total of 256 SNMP communities.

Adding an SNMP community string

The **snmp-server community** command sets the community string and associates it with the user-defined group to restrict the access of MIBs for SNMPv1 and SNMPv2c requests. You execute this command in global configuration mode.

1. Enter the **configure terminal** command.

```
device# configure terminal
```

2. Enter the **snmp-server community** command.

```
device(config)# snmp-server community comm1 groupname accGroup1
```

When creating a new community string without specifying a group name, there is no group name associated with the community string. You must associate the community string with any nonexisting or existing group name to be able to contact the switch using SNMPv1/v2c.

The following example also applies an IPv4 ACL and an IPv6 ACL.

```
device(config)# snmp-server community comm1 groupname accGroup1 ipv4-acl standV4ACL1 ipv6-acl
standV6ACL1
```

NOTE
For the entire flow of implementing SNMP ACLs, refer to Implementation flow of ACLs under SNMP on page 166.

NOTE
The Extreme VDX 2746 supports only the default group names (admin and user).

Removing an SNMP community string

The following example removes the community string "public" and its associated group "user".

1. Enter the **configure terminal** command.
2. Enter the **no snmp-server community** command.

```
device(config)# no snmp-server community public
```

The following example removes the associated IPv4 and IPv6 ACLs from the SNMP community.

```
device(config)# no snmp-server community public ipv4-acl
device(config)# no snmp-server community public ipv6-acl
```

Configuring SNMP server hosts

The **snmp-server host** command sets the trap destination IP addresses, SNMP version, community string for SNMPv1and SNMPv2c, the destination port for the SNMP server host, the severity level, and the SNMP communication through default VRF (default-vrf) or management VRF (mgmt-vrf) or user-defined VRF.

To configure SNMP trap hosts associated with community strings, you must create the community string using the **snmp-server community** command before configuring the host.

The SNMP agent supports six trap-recipient severity levels. The default value for each attribute is as follows: host = 0.0.0.0; udp-port = 162; severity-level = none. The length of the community string must be from 2 through 64 characters.

Setting the SNMP server host

The **snmp-server host** command configures a SNMP host by associating with the SNMP users. You execute this command in global configuration mode.

1. Execute the **configure terminal** command to enter global configuration mode.
2. Execute the **snmp-server host** command to configure the SNMP server host.

   The following example sets up "commaccess" as a community string and sets 10.32.147.6 as a trap recipient with SNMPv2c on target port 162, with default-vrf.

   ```
   device(config)# snmp-server host 10.32.147.6 commaccess version 2c udp-port 162 severity warning use-
   vrf default-vrf
   ```

Removing the SNMP server host

The **no snmp-server host** *host* **community-string** *string* command removes the SNMP server host from the switch configuration.

Configuring the SNMP system group

The following tasks allow you to configure the system contact and system location objects for the SNMP system group.

Setting the SNMP server contact

Use the **snmp-server contact** command to set the SNMP server contact string.

The default contact string is "Field Support." The number of characters allowed is from 4 through 255.

1. Enter the **configure** command.
2. Enter the **snmp-server contact** *string* command.

   ```
   device(config)# snmp-server contact "Operator 12345"
   ```

   The example changes the default contact string to "Operator 12345." You must enclose the text in double quotes if the text contains spaces.

Removing the SNMP server contact

The **no snmp-server contact** *string* command restores the default contact information (Field Support).

Setting the SNMP server location

Use the **snmp-server location** command to set the SNMP server location string.

The default SNMP server location string is "End User Premise." The number of characters allowed is from 4 through 255.

1. Enter the **configure terminal** command.

2. Enter the **snmp-server location** *string* command.

```
device(config)# snmp-server location "Building 3 Room 214"
```

3. Enter the **no snmp-server location** command to remove the location.

   The example changes the default location string to "Building 3 Room 214." You must enclose the text in double quotes if the text contains spaces.

Setting the SNMP server description

Use the **snmp-server sys-descr** command to set the SNMP server description string.

The default SNMP server description string is "Extreme-VDX-VCS <vcsid>." The number of characters allowed is from 4 through 255.

1. Enter the **configure** command.
2. Enter the **snmp-server sys-descr** command.

```
device(config)# snmp-server sys-descr "Brocade-VDX Test Bed"
```

3. Enter the **no snmp-server sys-descr** command to remove the location.

   The example changes the default location string to "Extreme-VDX Test Bed." You must enclose the text in double quotes if the text contains spaces.

Configuring multiple SNMP server context-to-VRF mappings

A single SNMP agent can be supported by multiple instances of the same MIB module if the context name is mapped to a virtual routing and forwarding (VRF) instance created within the switch.

Each VRF is mapped with a specific key called *context_name*. The context name is used to identify the VRF and fetch the MIB details of the mapped VRF from the underlying modules. For example, the OSPF-MIB by default returns the queried OSPF-MIB object values pertaining to the default VRF (**default-vrf**). The user input will vary for different SNMP versions. In case of SNMPv1 and SNMPv2c, the context with the community is mapped by means of the **snmp-server mib community-map** command. The SNMP agent supports 256 contexts for context-to-VRF mapping.

ATTENTION
SNMP SET requests work only on the default VRF.

The mapping of the community to a context and the context to a VRF is illustrated in the following examples.

1. In global configuration mode, enter the **snmp-server community** command to create a community “public” with Groupname “admin”.

```
device# configure terminal
device(config)# snmp-server community public groupname admin
```

2. Enter the **snmp-server context** command to configure a context and map with a VRF. The VRF can be the default VRF or a user-defined VRF.

```
device(config)# snmp-server context mycontext vrf myvrf
```

3. Enter the **snmp-server mib community-map** command to map the community to a context.

   In this example, the community "public" is mapped with the context "mycontext" and the context is in turn mapped with the "myvrf". To fetch the OSPF MIB-objects values belonging to the VRF "myvrf", use community "public" for SNMPv1 and SNMPv2c queries.

   ```
   device(config)# snmp-server mib community-map public context mycontext
   ```

   Perform the following steps to set the SNMP server context, using the **snmp-server context** command, to map the context name in a SNMPv3 packet's protocol data unit (PDU) to a VRF instance. The context-to-VRF mapping is one-to-one and is applicable to all SNMP versions. Only one context is allowed per VRF instance. For SNMPv3, it is sufficient to map the context with a VRF. (The SNMPv3 request PDU takes the context name as input.)

4. Enter the **snmp-server user** command to create user “snmpuser1” with group name “admin”.

   An SNMPv3 query uses the user and context name with the corresponding credentials. For group name information, refer to "Configuring SNMP server groups."

   ```
   device(config)# snmp-server user snmpuser1 auth md5 auth-password 123456789 priv DES priv-password
   123456789 groupname admin
   ```

Configuring SNMP server views

SNMP views are named groups of MIB objects that can be associated with user accounts to allow limited access for viewing and modifying the SNMP statistics and system configuration. With SNMP views user can create or remove the access to MIB object that need to be included or excluded from viewing.

SNMP views reference MIB objects using object names. It represents the hierarchical location of the object in the MIB tree. Use the following procedure to create or remove a view entry with MIB objects to be included or excluded from user access. The views are associated with each group to restrict or allow access to the OIDs.

NOTE
View creation is not supported on the Extreme VDX 2746.

1. Enter the **configure terminal** command.
2. Enter the **snmp-server view** *view-name mib_tree* {**included** | **excluded**} command.

   ```
   device(config)# snmp-server view view1 1.3.6.1.2.1.1.3 excluded
   ```

   Enter the **no** form of the command to remove the configured SNMP server view entry.

   The following is an example to create a SNMP view entry "view1" with excluded permission for the MIB object ID "1.3.6.1.2.1.1.3" create a SNMP view entry "view1" with excluded permission for the MIB object ID "1.3.6.1.2.1.1.3":

   ```
   device(config)# snmp-server view view1 1.3.6.1.2.1.1.3 excluded
   ```

   The following is an example to create SNMP view entry "view2" with included permission for the MIB object ID "1.3.6.1":

   ```
   device(config)# snmp-server view view2 1.3.6.1 included
   ```

   The following is an example to remove the SNMP view entry "view1" from the configuration list:

   ```
   device(config)# no snmp-server view view1 1.3.6.1.2.1.1.3 excluded
   ```

   NOTE
   The maximum number of views supported with MIB tree entries is 10.

Configuring SNMP server groups

SNMP groups map the SNMP user for the version v3 and the community for the versions v1 and v2c to SNMP views.

Each SNMP group can be configured with a read view with read-only access, a write view with read-write access, and a notify view to trap notifications to be encrypted and sent to target hosts, or all of the options. Users who are mapped to a group with SNMP view can use its views for access control.

The authoring parameter determines the authentication and privacy required by the users to access the view. The auth **auth** | **noauth** | **priv** parameter is available when you select v3, not v1 or v2.
Use the following procedure to configure or remove a specified SNMP group.

NOTE
Group creation is not supported for the Extreme VDX 2746. Only the default groups snmpadmin and snmpuser are supported.

1. Enter the **configure terminal** command.
2. Enter the **snmp-server group** command.

```
device(config)# snmp-server group group1 v3 auth read myview write view1 notify view1
```

Enter the **no** form of the command to remove the configured SNMP server group.

NOTE
The maximum number of SNMP groups supported is 10.

The following is an example to create SNMP server group entries for SNMPv3 user group with auth/noauth permission:

```
device(config)# snmp-server group admin v3 auth read view2 write view2 notify view2
device(config)# snmp-server group admin1 v3 noauth read view1 write view1 notify view1
```

The following is an example to remove the configured SNMP server groups:

```
device(config)# no snmp-server group admin v3 auth read view2 write view2
device(config)# no snmp-server group admin v3 noauth read view1 write view1 notify view1
```

Configuring SNMP server users

The **snmp-server user** command configures a SNMPv3 user and allows the configured user to be associated with user-defined SNMP groups. You can execute this command in global configuration mode and RBridge ID configuration mode.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

2. Enter the **snmp-server user** command.

   The following example configures the SNMPv3 users "user1" and "user2" associated with user-defined group "group1" under global configuration mode.

   ```
   device(config)# snmp-server user user1 groupname group1
   device(config)# snmp-server user user2 groupname group1 auth md5 auth-password password priv DES
   priv-password
   ```

   The following example configures the SNMPv3 users "user1" and "user2" associated with user-defined group "group1" under global configuration mode. It also applies an IPv4 ACL and an IPv6 ACL to "user1."

   ```
   device(config)# snmp-server user user1 groupname group1 ipv4-acl standV4ACL1 ipv6-acl standV6ACL1
   device(config)# snmp-server user user2 groupname group1 auth md5 auth-password password priv DES
   priv-password
   ```

   NOTE
   You can associate only global SNMPv3 users with ACLs. For the entire flow of implementing SNMP ACLs, refer to Implementation flow of ACLs under SNMP on page 166.

   The following example configures the SNMPv3 user "snmpadmin1" under RBridge ID configuration mode.

   ```
   device(config-rbridge-id-1)# snmp-server user snmpadmin1 groupname snmpadmin auth sha auth-password
   private123 priv DES priv-password public123
   ```

   NOTE
   When creating a new SNMPv3 user without the **groupname** option, by default there is no group name mapped with the SNMPv3 user. You must map the configured SNMPv3 user with a group name available in the group CLI configuration to contact the switch through SNMPv3.

   NOTE
   The behavior of this command in RBridge ID configuration mode is same as in global configuration mode. If the user name is configured to be the same in both global and RBridge ID configurations, the RBridge ID configuration takes precedence. The encrypted password generated in global configuration mode can be used for another global user to modify the passwords. The encrypted passwords generated in global configurations cannot be used in RBridge ID configurations, and vice versa.

Configuring password encryption for SNMPv3 users

For SNMPv3 users, the passwords for **auth-password** and **priv-password** are encrypted. You can configure either with a plain-text password or an encrypted password. In both cases, the passwords are shown in the **show running-config** command as encrypted.

The following example shows a plain-text password configuration:

```
device(config)# snmp-server user snmpadmin1 auth md5 auth-password private123 priv DES priv-password
public123
```

The following example shows an encrypted password configuration:

```
device(config)# snmp-server user snmpadmin2 groupname snmpadmin auth md5 authpassword "MVb
+360X3kcfBzug5Vo6dQ==\n" priv DES priv-password "ckJFoHbzVvhR0xFRPjsMTA== \n" encrypted
```

NOTE
This command may not be successful where encrypted passwords are generated by third-party or open-source tools.

Configuring SNMP server v3hosts

The **snmp-server v3host** command configures a SNMPv3 host by associating with the SNMP users. You execute this command in global configuration mode and RBridge ID configuration mode.

Refer to the *Extreme Network OS Command Reference* for complete descriptions of all the SNMPv3 server host options.

1. Enter the **configure terminal** command.
2. Enter the **snmp-server v3host** command to associate the users to the host.

```
device(config)# snmp-server v3host 1050::5:600:300c:326b snmpadmin2 severity-level Info use-vrf
default-vrf
```

This example configures the SNMPv3 trap IPv6 host 1050:0:0:0:5:600:300c:326b associated with SNMP user "snmpadmin2" under global configuration mode with default vrf.

```
device(config-rbridge-id-1)# snmp-server v3host 10.26.3.166 snmpuser2 severity-level Info udp-port
4425
```

This example configures the SNMPv3 trap host 10.26.3.166 associated with SNMP user "snmpuser2" under RBridge ID configuration mode.

The global SNMPv3 host can be configured by associating with only global SNMPv3 users and the local SNMPv3 host can be configured by associating with only local SNMPv3 users. You cannot create a SNMPv3 host in global configuration by associating with the local SNMPv3 users and vice versa.
The **no snmp-server v3host** command removes the SNMP server v3host from the switch configuration altogether.

Managing SNMP access rights using ACLs

Access lists (ACLs) enable you to permit or deny SNMP access by IP address.

SNMP server groups enable you to specify read, write, and notify permissions for the following entities:

- Community, under SNMPv1 and SNMPv2c
- User, under SNMPv3

For SNMP packets that pass community/user validation, access lists (ACLs) offer an additional permit/deny level, filtered by IP addresses that you specify.

If SNMP ACLs are applied, the validation order is as follows:

1. SNMP-server validation (community/user string). If not validated, the SNMP packet is dropped.
2. Server-ACL validation
   - If there is a **deny** match—including an explicit or implicit `deny any` rule—the packet is dropped.

     NOTE
     Unless you include an explicit `permit any` rule, an implicit `deny any` rule is automatically applied for IP addresses not explicitly permitted.

   - If there is a **permit** match—including a `permit any` rule—validation continues.
3. Server-group validation, the concluding step of the validation flow

Implementation flow of ACLs under SNMP

The implementation flow for ACLs under SNMP is as follows:

1. Create access lists (ACLs) that permit or deny specified IP addresses. For details, refer to Creating standard ACLs for SNMP on page 166.
2. Define server groups with the needed combination of Read or Write; and Notify permissions. For details, refer to Adding an SNMP community string on page 159.
3. For SNMPv1 or SNMPv2c, implement Adding an SNMP community string on page 159.
4. For SNMPv3, implement Configuring SNMP server users on page 163.

Creating standard ACLs for SNMP

Use these procedures to create access lists (ACLs) that contain rules permitting or denying access from specified IP addresses.

Creating an IPv4 ACL for SNMP

A standard ACL permits or denies traffic according to source address only. SNMP supports only standard ACLs.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter the **ip access-list standard** command to create the access list.

```
device(config)# ip access-list standard stdACL3
```

3. For each ACL rule, enter a **seq** command, specifying the needed parameters.

```
device(config-ipacl-std)# seq 5 permit host 10.20.33.4
device(config-ipacl-std)# seq 15 deny any
```

The following example does the following, under SNMPv3:

1. Creates a IPv4 standard ACL named "test".
2. Defines rules that permits packets from a specified host and denies packets from any other host.
3. Configures the SNMP server user "user1", including application of the "test" IPv4 ACL.

```
device(config)# ip access-list standard test
device(conf-ipacl-std)# permit host 10.1.1.1
device(conf-ipacl-std)# deny any
device(conf-ipacl-std)# exit
device(config)# snmp-server user user1 groupname snmpadmin auth sha auth-password private123 priv DES priv-
password public123 ipv4-acl test
```

Creating an IPv6 ACL for SNMP

A standard ACL permits or denies traffic according to source address only. SNMP supports only standard ACLs.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter the **ipv6 access-list standard** command to create the access list.

```
device(config)# ipv6 access-list standard std_V6_ACL4
```

3. For each ACL rule, enter a **[seq] {permit | deny | hard-drop}** command, specifying the needed parameters.

```
device(config-ip6acl-std)# seq 5 permit host 2001:db8::1:2
device(config-ip6acl-std)# seq 15 deny any
```

The following example does the following, under SNMPv1 or SNMPv2c:

1. Creates an IPv6 standard ACL named "stdv6acl".
2. Defines rules that permits packets from a specified host and denies packets from any other host.
3. Configures the SNMP server community "c1", including application of the "stdv6acl" IPv6 ACL.

```
device(config)# ipv6 access-list standard stdv6acl
device(conf-ip6acl-std)# permit fe::/24
device(conf-ip6acl-std)# deny any
device(conf-ip6acl-std)# exit
device(config)# snmp-server community c1 groupname admin ipv6-acl stdv6acl
```

Displaying SNMP configurations

Use the **show running-config snmp-server** command to display the current SNMP configurations for the SNMP host, community string, contact, and location, as well as other SNMP configuration options such as SNMPv3 host address, context, VRF mapping, and applied ACLs.

Enter the **show running-config snmp-server** command.

```
device# show running-config snmp-server

snmp-server contact "Field Support."
snmp-server location "End User Premise."
snmp-server sys-descr "Brocade VDX Switch."
snmp-server enable trap
snmp-server community ConvergedNetwork groupname user
snmp-server community OrigEquipMfr groupname admin
snmp-server community "Secret C0de" groupname admin
snmp-server community c1 groupname admin ipv4-acl test ipv6-acl stdv6acl
snmp-server community c2 groupname g1 ipv4-acl 1 ipv6-acl 2
snmp-server community com1
snmp-server community common groupname user
snmp-server community private groupname admin
snmp-server community public groupname user
snmp-server host 10.20.53.161 private
 severity-level Info
 use-vrf mgmt-vrf
!
snmp-server user snmpadmin1 groupname user
snmp-server user snmpadmin2 groupname snmpadmin
snmp-server user snmpadmin3 groupname snmpadmin
snmp-server user snmpuser1 groupname snmpuser
snmp-server user snmpuser2 groupname snmpuser
snmp-server user snmpuser3 groupname snmpuser
snmp-server user user1 groupname admin auth sha auth-password "ggc+pJR30+ORXd6OULEU6GaUFN4=\n" priv DES
priv-password "4qrmMZ5wR1W9MQMKxrQWFpaAgl8=\n" encrypted ipv4-acl test ipv6-acl stdv6acl
snmp-server view All 1 included
snmp-server group admin v1 read All write All notify All
snmp-server group admin v2c read All write All notify All
snmp-server group snmpadmin v3 notify All
snmp-server group snmpuser v3 notify All
snmp-server group user v1 read All notify All
snmp-server group user v2c read All notify All
```

Enabling and disabling SNMP server trap link-status

The **snmp-server trap-link status** command is available under the following interface sub-modes.

- Physical interface
- Virtual Ethernet (VE) interface
- Port-Channel interface
- Loopback interface

Under all the listed interfaces, the traps are enabled by default. Enter the **no snmp-server trap-link status** command to disable the linkUp trap or linkDown trap under an interface. As the traps are enabled by default, the **no** form of the command is not displayed under the **show running-config** command.

Enabling and disabling the SNMP server offline-if enable

The **no snmp-server offline-if enable** command is used to enable the command behavior to display the interfaces belonging to an offline slot. By default, this behavior is disabled. The new behavior is enabled via SNMP. This command will only change the behavior of SNMP with respect to interfaces on an offline slot. There is no change in the behavior of any existing CLI commands.

When the **no linecard** command is executed on a particular slot, the interfaces related to that slot will not be displayed via SNMP as well, irrespective of whether the offline slot interface display behavior is enabled or disabled.

> NOTE
> The **no snmp-server offline-if enable** command only applies to chassis systems and not to pizza box switches.

Configuring tuple representation for IF-MIB objects

You can configure whether the ifDescr and ifName objects that belong to the Interfaces Group MIB (IF-MIB) are represented in 2-tuple or 3-tuple format.

By default, 2-tuple representation is implemented.

1. From global configuration mode, enter RBridge ID configuration mode.

```
device# configure terminal
device(config)# rbridge-id 1
```

2. Enter the **snmp-server three-tuple-if enable** command.

```
device(config-rbridge-id-1)# snmp-server three-tuple-if enable
```

> NOTE
> The results of this command will appear in the running configuration only when it is enabled. This option is disabled by default, and the ifDescr and ifName objects are represented as 2-tuple.

3. Use the **no** form of this command to return to the default behavior (2-tuple representation).

```
device(config-rbridge-id-1)# no snmp-server three-tuple-if enable
```

VMware vCenter

vCenter and Network OS integration overview

The VMware vCenter Server allows for the management of multiple ESX /ESXi servers and virtual machines (VMs) from different ESX servers through a single graphical user interface (GUI). It provides unified management of all the hosts and VMs in the data center, from a single console with an aggregate performance monitoring of clusters, hosts and VMs.

The VMware vCenter and Extreme Network OS integration supported in Extreme VCS Fabric mode enables you to discover VMware ESX servers managed by a vCenter server. The VMware server hosts (ESX servers) are connected directly to the physical switches through the switch ports (edge ports in Extreme VCS Fabric mode). The server hosts implement a virtual switch (vSwitch), which is used to provide connections to the VMs. The fundamental requirement for the vCenter and Network OS integration is the IP-level management connectivity of the vCenter Server 4.0 version and later with the Extreme VDX switches.

NOTE
The Network OS integration with vCenter requires vCenter versions 4.0, 4.1, 5.1 or 5.5.

You can view virtual switches and virtual machines, their associated MAC addresses, and network policies using the Network OS command line interface (CLI). Refer to the *Extreme Network OS Command Reference* for details about the **vcenter** and **vnetwork** commands.

vCenter properties

The vCenter manages the VMware ESX/ESXi hosts. The vCenter user interface is provided through a vSphere client on the same management network as the vCenter, and virtual machines (VMs) are created using the vSphere client user interface. In addition to creating the VMs, the server administrator associates the VMs with distributed virtual switches, distributed virtual port groups, standard virtual switches (vSwitches) and standard port groups.

The vCenter automatically generates some of the VM properties (such as the MAC address), and some properties must be configured (such as the VLAN properties). Most of the VM configuration, including network policies, is done using the vCenter's vSphere user interface and is beyond the scope of this document.

For VMWare configuration information, visit the VMware documentation site.

vCenter guidelines and restrictions

Follow these guidelines and restrictions when configuring vCenter:

- Special characters in the port group names are replaced with the URL-encoded values.
- Standard port groups with the same name that reside in different ESX/ESXi hosts must have identical VLAN settings across all hosts.
- For all vCenter port groups, Network OS automatically creates a port profile with the following format: auto-*vcenter_name-datacenter_ID-port-group-name*. User editing of these auto port groups is not supported.

- Network OS supports vCenter discovery that is based on events.
- Network OS supports LLDP and QoS (IEEE 8021.p) for distributed virtual switches (dvSwitches).
- Using port-profile names fewer than 63 characters has been shown to conserve CPU resources.
- CDP/LLDP-receiving interface ports must not have any conflicting configurations on the interface that prevent them from being in a port-profiled mode.
- An interface is prevented from becoming a port-profile-port only when conflicting switchport, QoS, and security configurations reside on the interface.
- Before configuring a vCenter in the fabric, remove all the manually created port profiles that have vCenter inventory MAC associations.
- Up to four vCenter configurations are supported per fabric, with support for multiple data centers.
- Duplicate vCenter asset values are not supported, such as duplicate MAC addresses and duplicate Host names.

NOTE
Refer to the *Release Notes* for the number of port groups supported per platform, as well as other related information.

vCenter discovery

A Extreme VDX switch connected to VMware ESX/ESXi hosts and virtual machines must be aware of network policies in order to allow or disallow traffic; this requires a discovery process by the VDX switch. During VDX switch configuration, relevant vCenters that exist in its environment and the discovery of virtual assets from the vCenter occurs in the following circumstances:

- When a switch boots up
- When a new vCenter is configured on the VDX switch and activated (activation turns on the timer processing, set to 30-minute intervals)
- When the discovery is explicitly initiated with the CLI

The following assets are discovered from the vCenter:

- Hosts and data centers associated with the vCenter
- Virtual machines (VMs) that have been created on the hosts
- VMware distributed virtual port groups (dvPortGroups)
- Standard port groups, with QoS priority associated with a dvPortGroup
- Standard virtual switches
- Distributed virtual switches

vCenter configuration

Configuring vCenter consists of three basic steps performed in this order:

1. Enabling VMware vSphere QoS.
2. Enabling CDP/LLDP on switches.
3. Adding and activating the vCenter.

These steps and postconfiguration steps are discussed in this section.

Step 1: Enabling QoS

You must edit the network resource pool settings and set QoS priorities. Refer to the latest VMware vSphere Networking documentation.

Step 2: Enabling CDP/LLDP

In order for an Ethernet Fabric to detect the ESX/ESXi hosts, you must first enable Cisco Discovery Protocol (CDP) and Link Layer Discovery Protocol (LLDP) on all the virtual switches (vSwitches) and distributed vSwitches (dvSwitches) in the vCenter Inventory.

For more information, refer to the VMware KB article 1003885.

Enabling CDP/LLDP on vSwitches

Complete the following steps to enable CDP/LLDP on virtual switches (vSwitches).

1. Log in as root to the ESX/ESXi Host.
2. Use the following command to verify the current CDP/LLDP settings.

```
[root@server root]# esxcfg-vswitch -b vSwitch1
```

3. Use the following command to enable CDP/LLDP for a given virtual switch. Possible values here are **advertise** or **both**.

```
[root@server root]# esxcfg-vswitch -B both vSwitch1
```

Enabling CDP/LLDP on dvSwitches

Complete the following steps to enable CDP on distributed virtual switches (dvSwitches).

1. Connect to the vCenter server by using the vSphere Client.
2. On the vCenter Server home page, click **Networking**.
3. Right-click the distributed virtual switches (dvSwitches) and click **Edit Settings**.
4. Select **Advanced** under **Properties**.
5. Use the check box and the drop-down list to change the CDP/LLDP settings.

Step 3: Adding and activating the vCenter

After CDP is enabled on all the vSwitches and dvSwitches in the vCenter, configuration on the Network OS side is a two-step process, consisting of adding the vCenter and activating the vCenter.

Adding the vCenter

You must add the vCenter before initiating any discovery transactions. To authenticate with a specific vCenter, you must first configure the URL, login, and password properties on the VDX switch.

NOTE
By default, the vCenter server accepts only HTTPS connection requests.

1. Enter the **vcenter** command with the name, URL, user name, and password of the vCenter.

```
device(config)# vcenter myvcenter url https://10.2.2.2 username user password pass
```

2. An invalid state or condition of a vCenter can cause the deletion of all auto-port-profiles in a system. To prevent this from happening, configure the **ignore-delete-all-response** operand of the **vcenter** command to ignore the "delete-all" responses from the vCenter.

```
device# vcenter MYVC discover ignore-delete-all-response 5
```

Activating the vCenter

After adding the vCenter, you must activate the configured vCenter instance.

NOTE
In a VCS Fabric, you can configure the vCenter by using any node. Discovery is initiated by the primary node.

1. Enter the **configure terminal** command.
2. Enter the **vcenter** command to activate the vCenter.

```
device(config)# vcenter myvcenter activate
```

Immediately following first-time vCenter activation, the Network OS starts the virtual asset discovery process. Use the **show vnetwork vcenter status** command to display the vnetwork status, as in the following example.

```
device# show vnetwork vcenter status
vCenter          Start                Elapsed (sec)    Status
================ ==================== ================ =============
myvcenter        2011-09-07 14:08:42  10               In progress
```

When the discovery process completes, the status displays as "Success." Network OS has performed all the necessary configurations needed for the vCenter Server, and is now ready for CDP transmissions from the virtual switches to identify which ESX/ESXi host is connected to which physical interface in the Ethernet Fabric.

Discovery timer interval

By default, Network OS queries the vCenter updates every thirty minutes. If any virtual assets are modified (for example, adding or deleting virtual machines (VMs), or changing VLANs), Network OS detects those changes and automatically reconfigures the Ethernet Fabric during the next periodic rediscovery attempt.

Use the **vcenter interval** command to manually change the default timer interval value to suit the individual environment needs.

```
device(config)# vcenter myvcenter interval ?
Possible completions:
 <NUMBER:0-1440>  Timer Interval in Minutes (default = 30)
```

NOTE
Best practice is to keep the discovery timer interval value at the default (30). A value of 0 disables the periodic vCenter discovery.

User-triggered vCenter discovery

The discovery of virtual assets from the vCenter occurs during one of the following circumstances:

- When a switch boots up.
- When a new vCenter is configured on the VDX switch and activated (activation turns on the timer processing, set to 180-second intervals.)

- When the discovery is explicitly initiated with the CLI.

To explicitly initate vCenter discovery, perform the following task in global configuration mode.

1. An invalid state or condition of a vCenter can cause the deletion of all auto-port-profiles in a system. To prevent this from happening, configure the **ignore-delete-all-response** operand of the **vcenter** command to ignore the “delete-all” responses from the vCenter.

```
device(config)# vcenter MYVC discover ignore-delete-all-response 5
```

2. Return to privileged EXEC mode with the **exit** command.

```
device(config)# exit
```

3. Use the **vnetwork vcenter** command to trigger a vCenter discovery manually.

```
device# vnetwork vcenter myvcenter discover
```

Viewing the discovered virtual assets

Enter one of the following **show vnetwork** commands:

- **show vnetwork datacenter vcenter** *vcenter_name*

  NOTE
  The **datacenter** keyword is optional in the following commands and need not be used unless required.

- **show vnetwork dvpgs datacenter** *datacenter_id* **vcenter** *vcenter_name*
- **show vnetwork dvs datacenter** *datacenter_id* **vcenter** *vcenter_name*
- **show vnetwork hosts datacenter** *datacenter_id* **vcenter** *vcenter_name*
- **show vnetwork pgs datacenter** *datacenter_id* **vcenter** *vcenter_name*
- **show vnetwork vcenter status**
- **show vnetwork vmpolicy macaddr datacenter***datacenter_id* **vcenter** *vcenter_name*
- **show vnetwork vms datacenter** *datacenter_id* **vcenter** *vcenter_name*
- **show vnetwork vss datacenter** *datacenter_id* **vcenter** *vcenter_name*

where:

- **dvpgs** — Displays discovered distributed virtual port groups.
- **dvs** — Displays discovered distributed virtual switches.
- **hosts** — Displays discovered hosts.
- **pgs** — Displays discovered standard port groups.
- **vcenter status** — Displays configured vCenter status.
- **vmpolicy** — Displays the following network policies on the Extreme VDX switch: associated media access control (MAC) address, virtual machine, (dv) port group, and the associated port profile.
- **vms** — Displays discovered virtual machines (VMs).
- **vss** — Displays discovered standard virtual switches.

Refer to the *Extreme Network OS Command Reference* for detailed information about the **show vnetwork** commands.

Blocking VLAN creation when importing port profiles from vCenter

Extreme VDX devices discover different types of port groups from vCenter and create corresponding Extreme Automatic Migration of Port Profiles (AMPP) port profiles on the switch. This default behavior may not be desirable and can be managed.

During the discovery process, VLANs are also created when they are not already present on the switch. With the ability to block the automatic creation of VLANs, port groups can be filtered out when the VLANs for them are not present on the switch, and corresponding RASlogs are created.

To override the default behavior, enter the **vcenter** *vcenter-name* **vlan-create switch-admin** command in global configuration mode, specifying a vCenter.

```
device# configure terminal
device(config)# vcenter my_vcenter vlan-create switch-admin
```

The discovery process ignores all port groups for which VLANs are not already created on the switch. In this case, the VLANs must be configured manually by the switch administrator.

To revert to the default behavior, enter the **vcenter** *vcenter-name* **vlan-create auto** command.

```
device# configure terminal
device(config)# vcenter my_vcenter vlan-create auto
```

Consider the following scenarios for discovery, manual VLAN creation, and manual VLAN deletion.

Discovery

The following table summarizes the behavior of port group discovery and port profile creation where some VLANs are not established. In this case, the switch has VLANs 1, 2, 4, and 5 established. VLANs 3 and 6 are not established on the switch.

TABLE 31 Port group discovery and port profile creation with VLANs not established

Port groups received	Action	Reason
PG_1 (VLAN 1)	Port group and port profile created	
PG_2 (VLAN 2)	Port group and port profile created	
PG_3 (VLAN 3)	Skipped	VLAN 3 not established
DVPG_1 (VLAN 1) **NOTE** VMware distributed virtual port group (DVPG)	Port group and port profile created	
DVPG_6 (VLAN 6)	Skipped	VLAN 6 not established
PG_10 (VLANs 3, 5)	Skipped	VLAN 3 not established

Manual creation of VLANs

If VLAN 3 is now established manually on the switch, the port groups and port profiles for PG_3 and PG_10 are created automatically during the next discovery cycle. The following table summarizes this behavior. The user can also invoke manual discovery for the creation of these port groups and port profiles after VLAN 3 is created.

TABLE 32 Port group discovery and port profile creation with VLANs established manually

Port groups received	Action	Reason
PG_1 (VLAN 1)	Port group already on switch	
PG_2 (VLAN 2)	Port group already on switch	
PG_3 (VLAN 3)	Port group and port profile created	VLAN 3 established before discovery
DVPG_1 (VLAN 1)	Port group already on switch	
DVPG_6 (VLAN 6)	Skipped	
PG_10 (VLANs 3, 5)	Port group and profile created	VLAN 3 established before discovery

Deleting port profile VLANs

If a port profile is not activated, the associated VLAN must be deleted manually. Following the deletion of VLANs, you must use the **vnetwork reconcile vcenter** command on the switch and specify the name of the vCenter.

```
device# vcenter reconcile vcenter my_vcenter
```

NOTE

If a profile is activated, the deletion of VLANs in the profile is not allowed. You must delete the profile, and then delete the VLAN.

RASlog message components

The RASlog message ID, text, cause, and action are listed in the following table.

TABLE 33 RASlog message components

Message ID	1105
Text	vCenter Port Group *pg_name* is ignored as vlan- create mode is switch-admin and VLANs are not present on the switch.
Cause	Indicates that the Port Group fetched during discovery or event handling is being ignored because the requisite VLAN is not already present on the switch.
Action	Either change the VLAN creation mode to auto, or create the requisite VLANs on the switch.

Upgrade and downgrade considerations

During an upgrade from a previous release, all vCenter entries are populated with the default "vlan-create auto" mode. During a downgrade, if any vCenter is in the "vlan-create switch-admin" mode, the downgrade is blocked.

Virtual Fabrics

Virtual Fabrics overview

The Virtual Fabrics feature delivers Layer 2 Multitenancy solutions that provide support for overlapping VLANs, VLAN scaling, and transparent VLAN services by providing both traditional VLAN service and a transport service.

These services are offered by provisioning a Virtual Fabric (VF) in the data center. A VF operates like a regular 802.1Q VLAN, but has a 24-bit address space that allows the number of networks to scale beyond the standard 4K (4096) limit. The transport service is provided by configuring a transport VF, whereas traditional Layer 2/Layer 3 VLAN service is provided by configuring a service VF.

The Virtual Fabrics feature is deployed in data centers where logical switch partitioning and server virtualization require a large number of customer VLAN domains that must be isolated from each other in the data plane. The VLAN ID range is extended from the standard 802.1Q limit of 4095, to extend through 8191 on both a local RBridge and in a single VCS Fabric.

Data center virtualization, such as that provided by VMware vCenters, challenges network design in a variety of areas. Large numbers of networks can be required to support the virtualization of server hosts and multiple-tenant virtual machines (VMs), with Ports on Demand (POD) for the virtual data center (vDC) leading to POD configurations replicated at different ports. Such virtualization topologies are inherently largely independent of physical networks, with VM network configurations decoupled from the addressing and configurations of physical networks. The underlying Layer 2/Layer 3 infrastructure is an extension of a VMware virtual switch, or vSwitch, requiring address mapping at the boundary between the physical and virtual networks. In addition, the requirement for VM mobility makes it necessary to support the migration of VMs across the Virtual Cluster Switching (VCS) data center or across geographically separated sites, independently of the connectivity of the underlying infrastructure.

When a fabric is VF-enabled, existing VLAN configurations can apply to any VLAN. When a fabric is not VF-enabled, VLAN classification is not supported, and VLAN configurations are 802.1Q VLANs with VLAN IDs 1 through 4095.

A Virtual Fabric is just like a regular 802.1Q VLAN, but with a 24-bit address space that has the potential to support up to approximately16 million VLANs to be provisioned in the fabric. This VF VLAN address space is common to regular 802.1Q VLANs and classified VLANs. VLAN IDs from 1 through 4095 identify a conventional 802.1Q VLAN. VLAN IDs greater than or equal to 4096, up through 8191, identify VFs that need frame classification. A VF VLAN ID is unique within a local VCS Fabric, but may not be unique across multiple VCS Fabrics.

> NOTE
> A service VF is defined on the basis of the encapsulation classification of the ingress frame, with frames classified at the edge port according to the 802.1Q VLAN ID or MAC address. For the same service VF, the 802.1Q classification rule at each interface is a link-local configuration; the rule may be different at each interface.

A service VF thus represents a virtualized, normalized VLAN domain, where different link-protocol VLAN identifiers (port number, MAC address, and customer VLAN ID, or C-VID) are mapped to the same VLAN. In other words, VMs on the same service VF belong to the same forwarding domain, even though the attachment interfaces use different classification rules. When a VM moves among these interfaces, the Layer 2 forwarding domain does not change.

Extending a service VF among VCS data centers makes it possible to migrate VMs across those data centers.

The Extreme VDX 6940 supports distributed VXLAN gateways. "Distributed" means that the gateway can be deployed anywhere in the VCS Fabric and coexist with other non-gateway RBridges, subject to the topology constraints and other limitations as noted in the distributed VXLAN gateways section. Only the VDX 6940 supports this capability, and no special configuration is required. Using Fabric-Virtual-Gateway is one of many ways to configure the gateway IP address.

NOTE
Only Layer 2 extension gateways are supported. NSX Controller gateways are not supported.

Virtual Fabrics features

The following VLAN switch-port configurations are supported:

- Regular 802.1Q configuration (VLAN IDs 1 through 4095, with the exception of reserved VLANs)
- VLAN classification by means of a 802.1Q tag at the trunk port
- VLAN classification by means of a source MAC address at the access port

The following standard VLAN features remain supported by the service VF feature:

- Private VLANs (PVLANs)
- Layer 3 virtual Ethernet (VE) interfaces
- VLAN classifiers
- IGMP snooping
- VLAN ACLs
- Automatic Migration of Port Profiles (AMPP)
- RSPAN
- xSTP

In addition, support is now provided for transport service VFs, enabling a provisioning model in which a specific group of 802.1Q VLANs at an interface is classified into a common forwarding domain.

The maximum number of VLANs supported in this release, 802.1Q and classified, is as follows:

- There is fabric-wide support for 8K VF instances. The number of VFs supported on a local switch is platform-dependent.
- In a pure transport service deployment, port-based transport VFs are supported on every edge port, with up to 2048 VLANs (both 802.1Q and classified) across all ports.

For additional scalability details, refer to Virtual Fabrics performance considerations on page 185.

For example topologies and detailed discussion, refer to Virtual Fabrics configuration overview on page 185.

NOTE
Network OS supports conversational MAC learning (CML) on classified VLANs as well as on 802.1Q VLANs. For details on conversational MAC learning, refer to the *Extreme Network OS Layer 2 Switching Configuration Guide*.

Virtual Fabrics considerations and limitations

FGL limitations

Support is provided for pre-IETF standard Fine-Grained Labeling (FGLs) on ISLs. IETF defines the Ethertype for inner and outer labels as 0x893B. The outer and inner Ethertypes on ISLs are set to 0x893B and 0x8100, respectively.

STP support

The correct configuration of xSTP is the responsibility of the user. Much as the user must ensure that VLAN configurations and VLAN instance mappings are consistent on all switch ports, so also the user must understand whether a specific protocol, whether RSTP, MSTP, or PVST, is applicable to the underlying physical topology when 802.1Q VLANs and VFs coexist in the fabric.

Extreme VDX 6740 series limitations

The Extreme VDX 6740 series platforms do not support full port-based C-TAG translation. Whenever there is a translation conflict among the ports because the same C-TAG is used for different VLANs, the conflict cannot be resolved without incurring severe internal VLAN ID (IVID) scalability constraints. Optimal scalability is possible only when overlapping C-TAG classification occurs across port groups.

In a pure service VF deployment, these platforms support up to 4096 802.1Q VLANs and 2048 classified VLANs, assuming that each VLAN configured on a single port constitutes a single conflict.

In a pure transport VF deployment, these platforms support port-based transport VFs on every edge port, and a total of 2048 combined 802.1Q and classified VLANs, assuming that each VLAN is configured on a single port.

VCS extension through VXLAN

Network OS supports the use of VFs—by means of VXLAN Layer 2 extension—to extend a VCS Fabric to another VCS Fabric.

Distributed VXLAN gateways overview

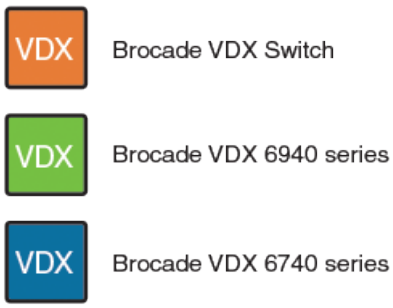
The distributed VXLAN gateways feature eliminates the need for an external gateway device.

Prior to Network OS 6.0.1, VXLAN gateways had to be connected to the VCS Fabric (for example, a four-node gateway fabric) as an external device in order to bridge the overlay network and the physical networks that are interconnected by the VCS Fabric. Beginning with the current release, the gateway function can be hosted by the VCS RBridges. This eliminates the need for an external gateway device and optimizes network resources, improving network performance in the data center.

> NOTE
> Existing VRRP-E implementations cannot support more than four RBridges in a session. As a result, VRRP-E-based VXLAN gateways are also limited to a maximum of four RBridges.

The following sections present various use cases, both supported and unsupported, and their corresponding topologies. Refer to the following legend for those topologies.

FIGURE 30 Distributed VXLAN gateways legend



NOTE
Distributed VXLAN gateways support both Virtual Fabrics Extension deployments and NSX Controller deployments.

Distributed VXLAN gateways supported topologies

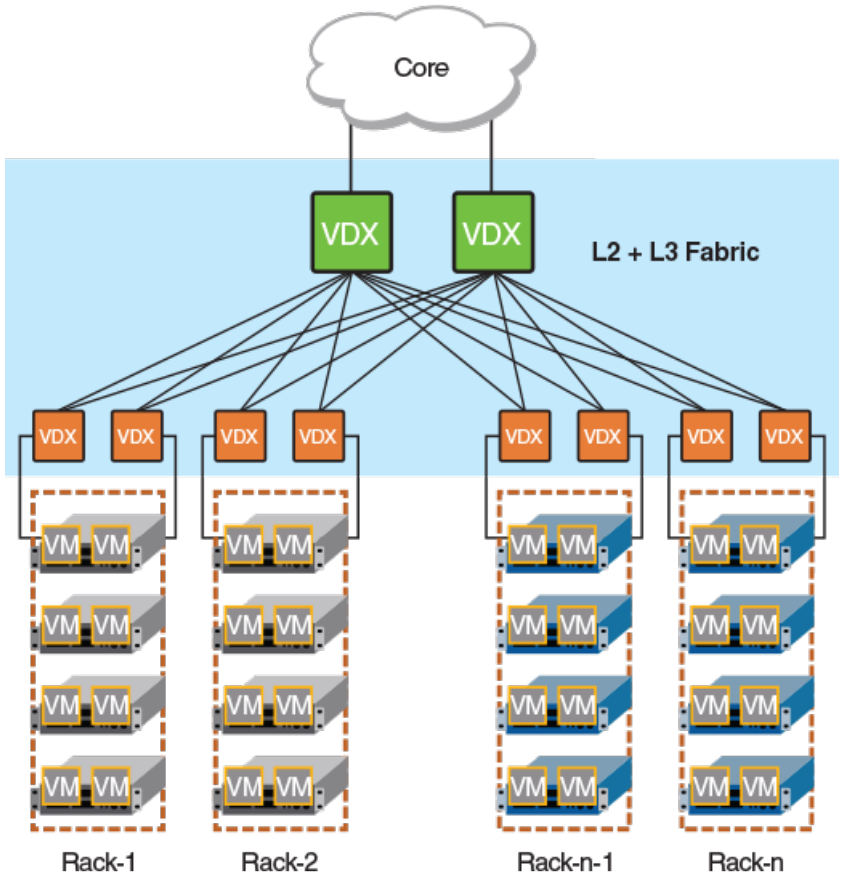
The following sections present the topologies that are supported in this release.

VDX 6940 at the aggregation layer

A Extreme VDX 6940 at the aggregation layer provides gateway functions for a VXLAN hypervisor at top of rack (ToR) or in the core.

A distributed VXLAN gateway makes it possible to connect to a hypervisor through a TRILL fabric, as the Extreme VDX 6940 supports TRILL-plus-VXLAN encapsulation. Refer to the following figure.

FIGURE 31 VDX 6940 in a Layer 2/Layer 3 fabric



In this topology, bridging a physical server and VXLAN servers that are located in the same rack requires interaction with an aggregation gateway, introducing an extra hop. The gateway supports VXLAN Network Identifier (VNI) classification for both east-west and north-south traffic. Note the following considerations:

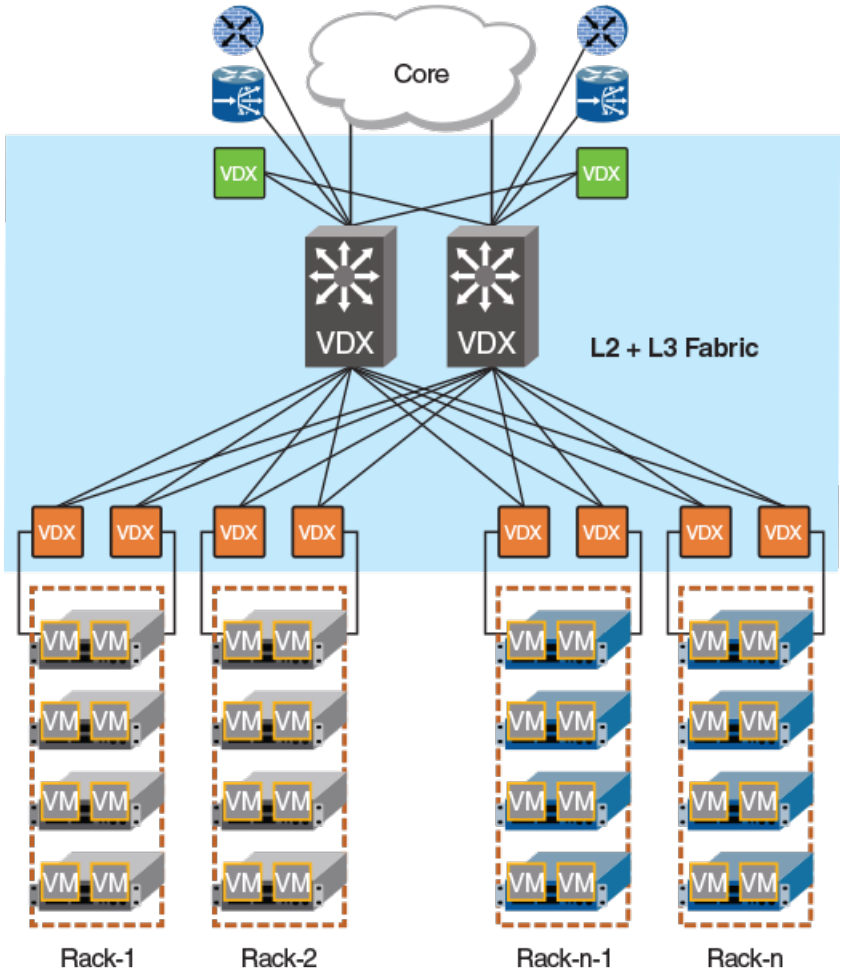
- The number of overlay networks that are supported in the fabric is limited by ASIC resources for the VNI classifications.
- The Extreme VDX 6940 is not supported at top of rack. This prevents VLAN-to-VXLAN traffic from "tromboning" in case the ToR gateway that does the VXLAN encapsulation is not in the same rack that holds the VXLAN server.

VDX 6940 as an appliance

A Extreme 6940 gateway can be inserted into the fabric as an appliance.

The following figure illustrates how a data center can use a VXLAN-incapable aggregation switch (such as a Extreme VDX 8770) to provide connectivity to the core while the switch is attached to a VXLAN gateway functioning as an appliance. Traffic between the physical server and a VXLAN-enabled server must always take an extra hop through the VDX to reach the appliance gateway.

FIGURE 32 VDX 6940 as a gateway appliance



Distributed VXLAN gateways unsupported topologies

The following topologies, although they can provide a certain degree of functionality, are not supported by Extreme for this feature.

VDX 6740-based fabric

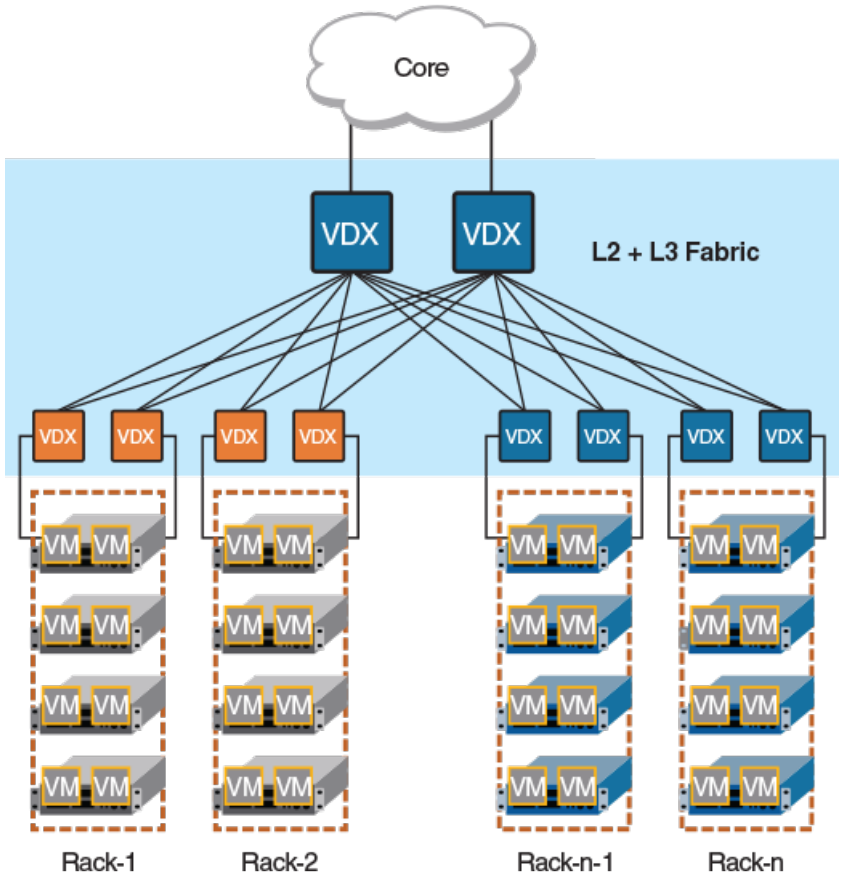
In a Extreme VDX 6740-based fabric, the gateway functionality is distributed across every VDX 6740 RBridge.

This topology is not recommended because of the following limitations:

- It requires a direct attachment between the gateway and a VXLAN-enabled rack, because the VDX-6740 cannot support TRILL-plus-VXLAN encapsulation.

- The number of server racks is limited to eight. This is constrained by the maximum number of RBridges that are allowed in a gateway.
- The VDX 6740 supports a maximum of 2000 VLANs.

FIGURE 33 VDX 6740-based fabric



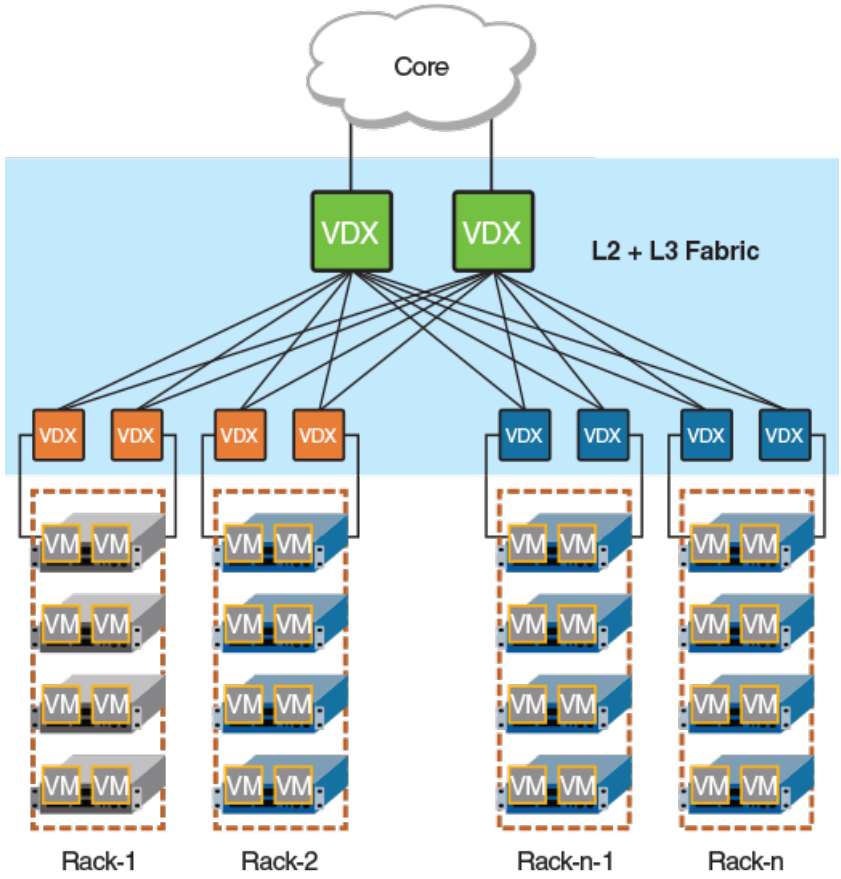
VDX 6740 and VDX 6940-based fabric

In this topology, a gateway comprises a mix of Extreme VDX 6740 and VDX 6940 RBridges.

The VDX 6740 is placed at top of rack to serve directly connected VXLAN servers. The VDX 6740 gateway handles VNI classifications for east-west traffic, thereby offloading traffic to the VDX 6940 gateway at the aggregation layer to serve other north-south traffic classifications.

Although this topology helps scale out the number of overlay networks supported in the fabric, it has the same limitations as in the previous topology, where the VDX 6740 is placed at top of rack.

FIGURE 34 VDX 6740 and VDX 6940-based fabric



Distributed VXLAN gateways RBridge scalability

Every RBridge in the fabric can act as a gateway. However, because dynamic virtual RBridge IDs (VRBs) support a maximum of eight RBridges, a tunnel VRB-ID can represent a maximum of eight RBridges out of all the gateway RBridges that are specified in the overlay-gateway configuration.

NOTE

The maximum number of gateway RBridges deployed at the aggregation layer is four. If this number is exceeded, the RBridges that are reachable by means of VRBs is nondeterministic.

Distributed VXLAN gateways upgrade and downgrade considerations

For the Extreme VDX 6740, after an upgrade or downgrade, the existing four-node VDX 6740 fabric can continue to serve as a four-node gateway fabric, but it should not join another fabric. In addition, the Virtual Fabrics extension gateway at the aggregation layer can continue to act as an extension gateway, but it cannot be configured as a VXLAN gateway.

ATTENTION

In a downgrade from the current release to a release earlier than Network OS 6.0.1, any new commands introduced in this release must be removed from devices before the downgrade is honored. Also, TRILL+VXLAN functionality is lost during a downgrade, and there is no warning to the user.

Distributed VXLAN gateways limitations

The following functions are not supported for distributed VXLAN gateways:

- BUM optimization
- Loop detection that involves both a tunnel and a nontunnel path
- Flow-based load balancing for tunnels over router ports
- Routing protocols over tunnels
- More than one VTEP per fabric
- QoS that is limited to DiffServ tunneling pipe mode
- A SPAN destination that is not a tunnel

Virtual Fabrics operations

This section summarizes the fundamental behavior of Virtual Fabrics.

Enabling and disabling a Virtual Fabric

In a VF-incapable fabric, ISL encapsulation is based on C-TAGs. In a VF-enabled fabric, ISL encapsulation is based on Fine-Grain Labels (FGLs), using both C-TAGs and S-TAGs. The VF is enabled only when the user issues the **vcs virtual-fabric enable** command. This enables the transition from one encapsulation type to another without disrupting existing traffic.

Enabling VFs

To ensure that there is no data disruption, the ISL encapsulation transition must be coordinated at the fabric level without toggling the link state. When the fabric is in a VF-capable state, the user executes the fabric-wide **vcs virtual-fabric enable** command to enable this transition. The command is distributed to all RBridges in the fabric and is executed in multiple stages across nodes.

> NOTE
> If the fabric state is VF-incapable, the **vcs virtual-fabric enable** command will not succeed.

Disabling VFs

To disable VFs in the fabric, the user must first remove all VF configurations in the fabric before issuing the **no vcs virtual-fabric enable** command. This command is distributed to all RBridges in the fabric. Each RBridge reverses the stage execution from what was done to enable the VF. When ISL encapsulation returns to being C-TAG based, the fabric is in a VF-capable state.

> NOTE
> Prior to issuing the **no vcs virtual-fabric enable** command, the user must ensure that all new commands and enhanced commands that reference VFs (VFs with IDs greater than 4095) in the fabric are removed from the running configuration. Otherwise, the command will be rejected.

Joining a switch to the fabric

If VFs are disabled in the running configuration, the RBridge will not able to join a VF-enabled fabric upon reboot. Although the RBridge may form ISLs with a neighbor RBridge at the link level, the RBridge will be segmented from the VF-enabled cluster because of a capability mismatch.

In order for the RBridge to join a VF-enabled cluster, the procedures described in the "Upgrading and downgrading firmware with Virtual Fabrics" task must apply to the segmented RBridge. At the end of the procedures, the segmented RBridge will also be in the VF-enabled state. All segmented ISLs on the RBridge are automatically toggled so that it can rejoin the fabric.

Default Virtual Fabrics state

A switch after a **netinstall** is VF-incapable in the default configuration. It will fail to join a VF-enabled cluster when it reboots.

The default VF state in the running configuration is changed when the **copy default-config startup-config** command is executed. The resulting state is the same as the current VF state of the fabric.

Virtual Fabrics configuration overview

This section addresses a variety of Virtual Fabrics architecture scenarios and configuration issues related to service VFs.

Virtual Fabrics performance considerations

VLAN configurations, whether for 802.1Q or classified VLANs, are VCS Fabric global configurations that are distributed to all RBridges in the fabric.

If there are 100K VLANs in the data center, this implies that an RBridge will receive all those configuration, even though the RBridge could support a maximum of 8K (8192) VLANs. The processing of VLAN configurations for classified VLANs that are not actually provisioned in the RBridge introduces performance overhead. The impact of this will be manifest in configuration playback during system boot up and will result in a longer time for the fabric to reach a ready state to forward data.

Feature scalability

The following table lists VF resource numbers for the Extreme VDX 8770, VDX 6740, and VDX 6940 series platforms.

TABLE 34 VF resource numbers for Extreme VDX 8770, VDX 6740, and VDX 6940 series

Resources	VDX 8770 series	VDX 6740 series	VDX 6940 series
802.1Q VLANs per switch	4096	4096	4096
VFs per switch	4096	2000	4096
VLANs per fabric	8192	8192	8192
MAC-based classifications per switch	2048	256	1000
Service VF with overlapping C-TAG (best case)	4096	2000	4096
Service VF with overlapping C-TAG (worst case)	4096	64	4096
AMPP port profiles	1000	1000	1000
Transport VFs (best case)	1000	1000	1000
Transport VFs (worst case)	1000	178	144

Maximum number of VLANs

The target number of VLANs supported is 8K at a local RBridge. These may be traditional 802.1Q VLANs or service VFs. The following are some factors that determine the actual number of VLAN IDs that are configurable on a local RBridge constrained by the availability of ASIC resources to support internal VLAN IDs (IVIDs).

AMPP port-profile provisioning

When a port becomes a profile-port, IVIDs are allocated internally for VLANs that are defined in a port-profile domain associated with the interface or VLANs that are defined in the port-profiles associated with the interface. These resources are consumed before the VLANs are provisioned, underutilizing the IVID resources if a VM does not appear on these VLANs. The overbooked IVIDs will not be available for VLANs configured by the **switchport** command.

Number of VE interfaces

The **interface ve** command creates a VE interface on a specified VLAN on a specified RBridge. Because the VLAN flood domain must extend within the VCS Fabric by means of ISLs for routing purposes, an IVID must be allocated for the VLAN even though the VLAN is not configured on any local RBridge switch ports, imposing a scalability issue. Each VE interface consumes an IVID from the resource pool. The total number of interfaces (without switch ports) and VLANs that are provisioned cannot exceed 8192 on the VDX 8770 series and VDX 6740 series.

Number of VLAN ACLs

A VLAN ACL requires an IVID allocation for the target VLAN. If the target VLAN is configured on the local switch port, the ACL can be applied on the IVID for this VLAN. However, on the switch where the VLAN is transiting (that is, it is not configured on any switch port), an IVID must still be allocated for the ACL entry. The maximum number of VLANs that can be configured on the switch is determined by the maximum number of IVIDs minus the number of transit VLAN ACLs that are configured on the switch.

Extreme VDX 6740 series limitations

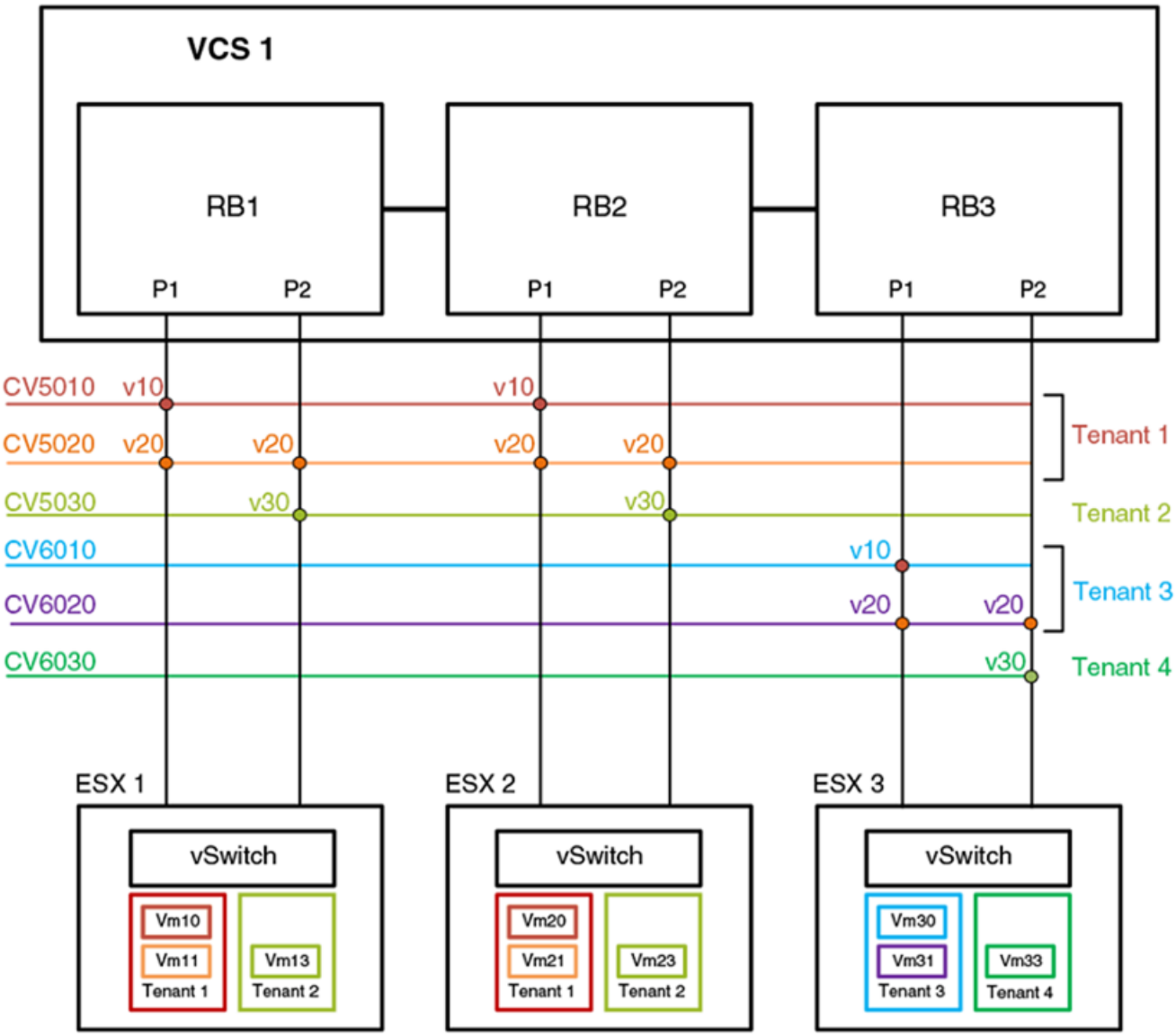
The Extreme VDX 6740 series RBridge supports a total of 6096 (4096 + 2000) VLANs, 4096 of which must be 802.1Q VLANs. The other 2000 VLANs can be configured as service VFs, assuming that the VLAN is configured on a single port. This limitation comes from the VPN table size on this platform.

VLAN virtualization

When a cloud computing provider provisions a virtual datacenter by replicating server-rack ports on demand (PODs) across server ports, different tenant domains exist but with overlapping 802.1Q VLANs at the server ports. The tenant domains are isolated by mapping the 802.1Q VLAN at each interface into a different VLAN forwarding domain. This capability allows the switch to support more than the 4K VLANs permitted by the 802.1Q address space.

In the example VMware topology shown in the following figure, the data center has three PODs, provided by RBridges RB1, RB2, and RB3. All three PODs (VMware ESXi hypervisors 1 through 3) have an identical pre-installed configuration. Each POD supports two tenants. The first tenant can have two applications running on VFs 10 and 20. The other tenant has only one application, running on VF 30. Here, four tenant applications are provisioned. Tenant 1 and 2 applications run on ESX1 and ESX2. Tenant 3 and 4 applications run on ESX3.

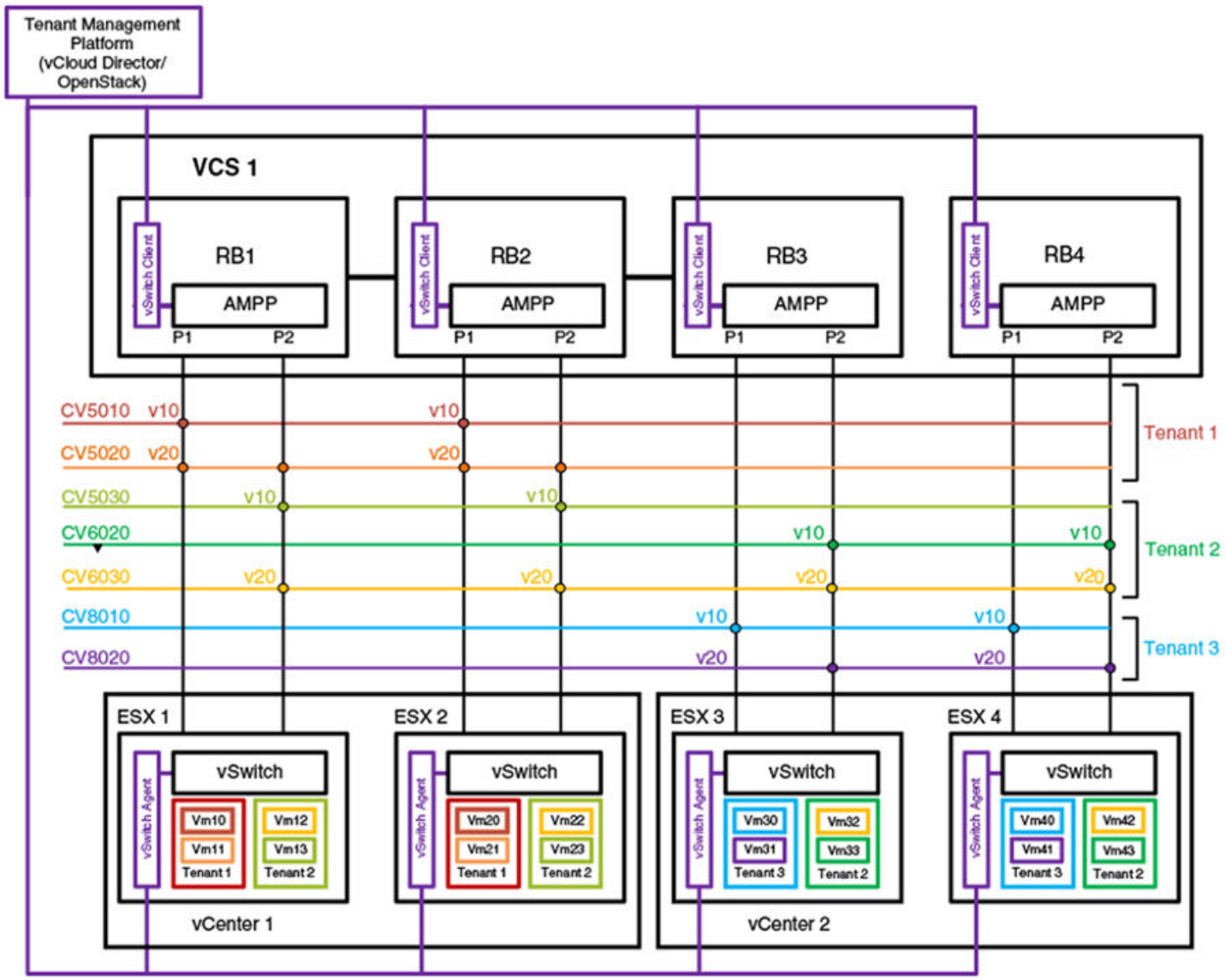
FIGURE 35 VLAN virtualization



Virtual data center deployment

The following figure illustrates an example VDC infrastructure that supports a VMware deployment.

FIGURE 36 VDC infrastructure



In a VMware-based cloud provider network, a VCS Fabric is connected to multiple vCenters, where each data center manages its own set of tenant networks. VMware vCloud/OpenStack is responsible for orchestrating tenant VLAN configuration through the vCenter agent integrated into a VCS RBridge and its ESXi servers. Each data center connects to the VCS Fabric by means of dedicated edge ports. The ability of the VCS Fabric to support 802.1Q VLAN virtualization allows each data center to support more than 4000 tenant VLANs. ESXi servers may use the same 802.1Q VLAN to represent different tenant VLANs at the edge port, or have them belong to a single VLAN domain. A vCenter agent running at an RBridge achieves VLAN virtualization by collating information obtained from the ESXi servers and the vCenter database. AMPP port profiles with service VF classifications are configured on the respective server ports.

The topology in the figure shows two vCenters. vCenter1 is connected to VCS RBridges1 and2. Because the VCS Fabric supports VLAN virtualization, the vCenter can assign two tenant networks, CV5010 and CV5030, that use the same 802.1Q VLAN (VLAN 10) on the ESXi server. Similarly, in vCenter2, three tenant VLANs —CV5030, CV6020, and CV8010— are configured, each representing a unique VLAN domain, but all using the same customer classification, C-TAG 10. If a VM application needs to run across applications,

then the same service VF can be configured on both vCenters; this is illustrated by CV6030, which is configured at all RBridges and uses the same C-TAG (C-TAG 20).

The service VF configuration at each edge port can be done as part of an AMPP configuration or automatically through vCenter orchestration. (Refer to VMware documentation for details of the vCenter Orchestrator.)

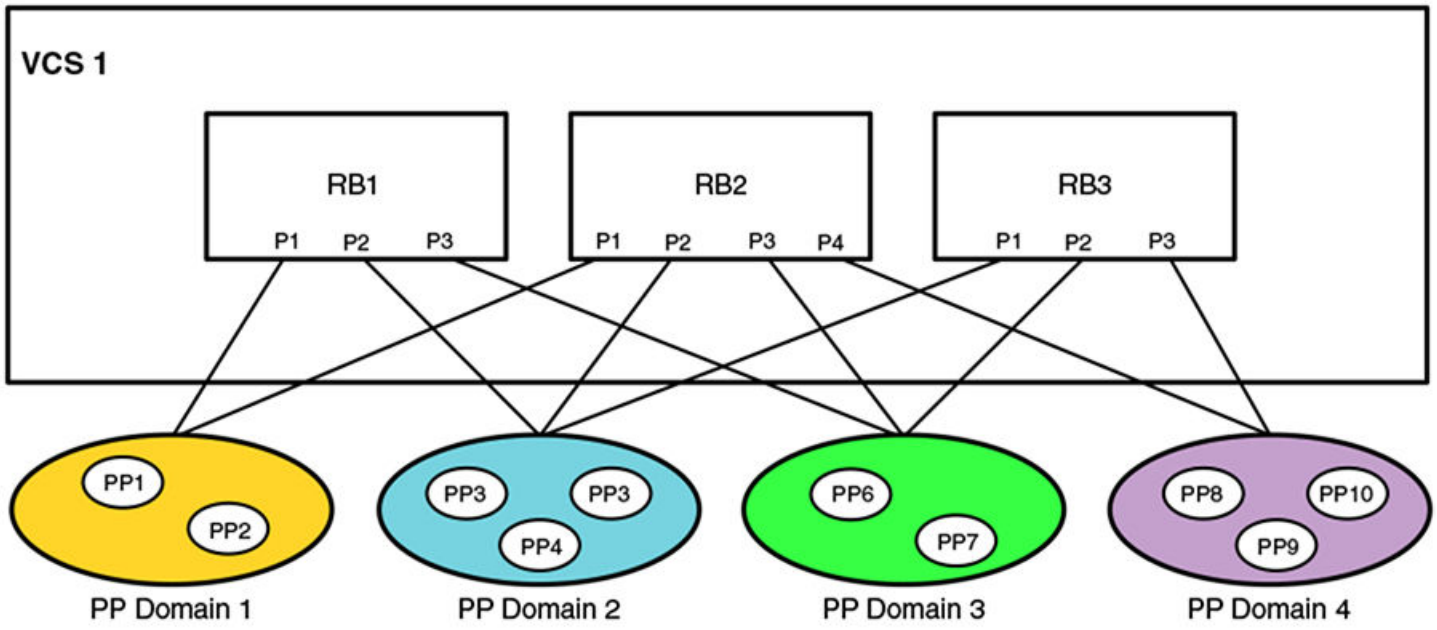
AMPP provisioning with service VFs

When the Automatic Migration of Port Profiles (AMPP) feature is used , a VCS Fabric is partitioned into port-profile (PP) domains. A PP domain is a set of port-profiles whose service VF ID cannot have conflicting C-TAG or MAC classifications. A port-profile domain supports a maximum of 4096 service VFs. The scope of VM mobility is defined by the set of interfaces onto which the port-profile domain is applied.

Port-profile domain topology

The following illustrates an example VCS Fabric that serves four port-profile domains across three RBridges.

FIGURE 37 Port-profile domains



Port-profile configuration rules

The VLAN classifications in a port-profile domain follow these rules:

1. Each profiled-port is associated with a port-profile domain. Different profiled-ports (on the same RBridge) may belong to different domains.
2. The same port-profile may be associated with multiple domains.
    a. This allows a service VF that is defined in one domain to extend into another domain.
    b. The VM MAC address must be unique within a domain.

3. The service VF classification rules within a domain are as follows:
    a. MAC-based classification is allowed only on an access port.
    b. C-TAG classification is allowed only on a trunk port
    c. Service VF classification cannot overlap across port-profiles in the same port-profile domain.
    d. Classification can overlap across PP domains.
4. The following example configurations are allowed for classification across all port-profiles in the same domain:
    a. **switchport trunk allow vlan add 5000 ctag 10**
    b. **switchport trunk allow vlan add 6000 ctag 20**
    c. **switchport trunk allow vlan add 7000 ctag 30**
5. The following example configurations are allowed for MAC-based classification on an access port:
    a. **switchport access vlan 8001**
    b. **switchport access vlan 8002 mac 2.2.2**
    c. **switchport access vlan 8002 mac 3.3.3**
6. The following example configurations are disallowed across all profiles in the same port-profile domain.
    a. Overlapping VF VLANs
        - **switchport trunk allow vlan add 8000 ctag 80**
        - **switchport trunk allow vlan add 8000 ctag 800**
    b. Overlapping C-TAG:
        - **switchport trunk allow vlan add 5000 ctag 10**
        - **switchport trunk allow vlan add 6000 ctag 10**
    c. Overlapping MAC address across all profiles in the same domain:
        - **switchport access vlan 8002 mac 2.2.2**
        - **switchport access vlan 8003 mac 2.2.2**
    d. Conflicting access and trunk service VF configurations:
        - **switchport access vlan 8000 mac 2.2.2**
        - **switchport trunk vlan add 8000 ctag 20**
7. The following configuration can be present on only one port-profile domain; it cannot coexist with other C-TAG-based classifications in that domain:
    a. **switchport trunk allow vlan all**
    b. When a port becomes a profiled-port, all service VFs in that domain are provisioned on this port.

Port-profile backward compatibility

The current AMPP port-profile-domain-based provisioning model is different from that in releases prior to Network OS 4.1.1 in two respects:

- The default port-profile configuration is not the same. The **switchport trunk allow vlan all** command that was present in prior releases has been removed. Other related configurations remain the same.
- A user-defined port-profile-domain has been introduced to support VM mobility. A port-profile must be explicitly associated with a profile domain.

In order to maintain legacy AMPP behavior when service VFs are disabled as a result of an upgrade, the following occurs:

- After upgrade, a new port-profile named UpgradedVlanProfile is auto-created. This profile has the single VLAN profile that contains the statement "switch port trunk allow vlan all". This is the configuration that is present in the pre-Network OS 4.1.1 default port-profile to resolve the provisioning differences before the service VFs are enabled.
- After upgrade, a default port-profile domain is created. This default domain contains all the existing user-created port-profiles and vCenter-created auto-profiles prior to the upgrade, in addition to the UpgradedVlanProfile.

The following rules apply to the UpgradedVlanProfile and the default port-profile domain while the switch is in the VF-disabled state after the upgrade.

- The user cannot edit the UpgradedVlanProfile.
- The newly created user port-profile or vCenter auto-profile is automatically added to the default port-profile domain.
- The deleted user or auto port-profile is automatically deleted from the default port-profile domain.
- The **show running-config** command or the **show port-profile domain** command shows the port-profiles in the default-profile-domain.
- The user is not allowed to edit the default port-profile domain.

The following rules apply after service VFs are enabled in the fabric.

- The user can edit the UpgradedVlanProfile just like any other port-profile.
- The newly created port-profile is not automatically added to the default domain. It can only be explicitly added to or removed from the default profile-domain.
- vCenter-managed auto-profiles continue to be added to or deleted automatically from the default port-profile domain.

  NOTE
  In Network OS 4.1.1 and later, the vCenter auto-profile does not support service VF classification.

- The **show running-config** command or the **show port-profile domain** command shows the port-profiles in the port-profile domain.
- The user is allowed to edit the default port-profile domain.
- The user is not allowed to delete the default port-profile-domain.

The following table compares the results of a **show running-config** command before and after an upgrade from Network OS 4.0.0 to Network OS 4.1.1 and later.

TABLE 35 Configuration status before and after upgrade

Network OS 4.0.0	Network OS 4.1.1 and later
port-profile default allow non-profiled-macs vlan-profile switchport switchport mode trunk switchport trunk allowed vlan all switchport trunk native-vlan 1 ! restrict-flooding ! port-profile pp1 vlan-profile switchport switchport mode trunk switchport trunk allowed vlan add 11 !	port-profile default allow non-profiled-macs vlan-profile switchport switchport mode trunk switchport trunk native-vlan 1 ! restrict-flooding ! ! port-profile pp1 vlan-profile switchport switchport mode trunk switchport trunk allowed vlan add 11 !

TABLE 35 Configuration status before and after upgrade (continued)

Network OS 4.0.0	Network OS 4.1.1 and later
! port-profile pp2 vlan-profile switchport switchport mode trunk switchport trunk allowed vlan add 12 ! !	! port-profile pp2 vlan-profile switchport switchport mode trunk ! ! port-profile UpgradedVlanProfile vlan-profile switchport switchport mode trunk switchport trunk allowed vlan all ! port-profile-domain default port-profile pp1 port-profile pp2 port-profile UpgradedVlanProfile !

Association of port-profile-domains with an interface

The **port-profile-port** command allows a user to associate a port-profile-domain (default or nondefault) with an interface. The result is that all VLANs specified therein are configured onto the interface.

When the **domain** keyword is not used, the default port-profile-domain is associated with an interface, as in the following example.

```
device(config)# int te 2/0/1
device(config-int-te-2/0/1)# port-profile-port
```

STP with service VFs

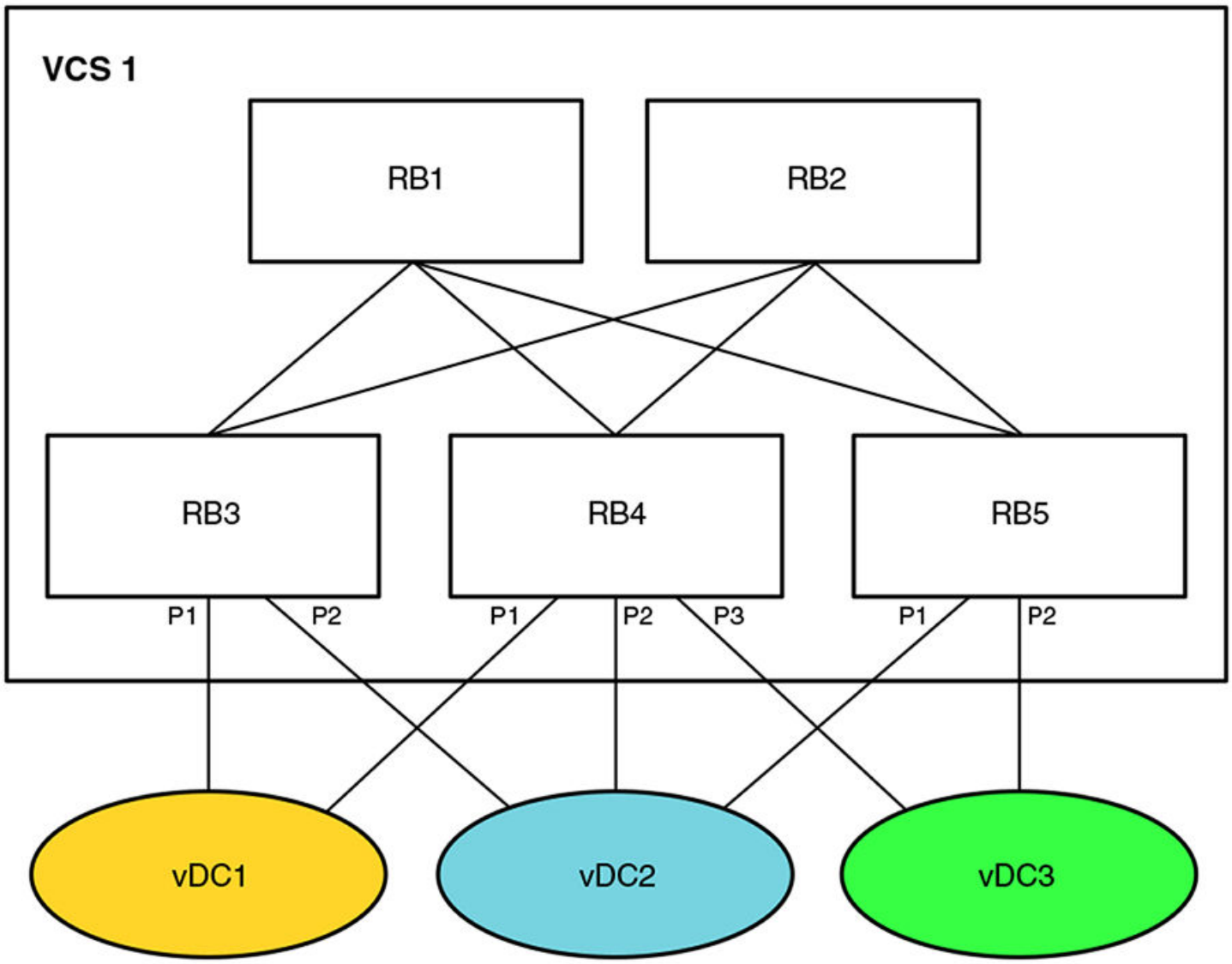
Spanning Tree Protocol (xSTP) is configured on a switch port as in previous releases. However, the user is responsible for configuring xSTP correctly. The user must ensure that VLAN configurations and VLAN-to-instance mappings are consistent across switch ports, and must understand whether RSTP or MSTP are applicable when 802.1Q VLANs and overlapping service VF classifications coexist in the fabric.

A VCS Fabric supports only a single STP domain of any protocol flavor that sends untagged BPDUs (for example, RSTP/MSTP). This domain should contain only switch ports that have identical VLAN configurations, whether 802.1Q or service VF. This is necessary for STP to operate correctly across the fabric. All other switch ports that do not participate in this domain must have STP disabled.

STP-with-service-VFs topology

The following illustrates an example VCS Fabric that is connected to three vDCs. There must not be any external physical connectivity among the vDCs.

FIGURE 38 STP-with-service-VFs topology



STP-with-service-VFs configuration rules

It is the user's responsibility to determine which vDC should participate in RSTP. The following behavior and rules apply:

- RSTP
  - RSTP is VLAN-unaware. Service VF configurations are allowed in each vDC.
  - A single RSTP topology is formed by the VCS Fabric (which appears as a single logical switch) and all attached vDCs.
  - A topology change in one vDC affects the topology of other vDCs.
  - Data loops between the VCS Fabric and individual vDCs are detected by this RSTP topology.
- MSTP
  - The VCS Fabric and the attached vDCs belong to the same MSTP region.
  - VLAN-to-instance mapping must be the same in the VCS Fabric and for each vDC.
  - An MSTP instance topology is formed by the VCS Fabric (which appears as a single logical switch) and all attached vDCs.

- Service VF configuration is allowed. Service VFs (VLAN IDs greater than 4095) are not assigned to any instance and are always in a forwarding state.
- A topology change in one MSTP instance for a vDC affects the topology of the same instance in other vDCs.
- Data loops between the VCS Fabric and individual vDCs are detected by each MSTP topology.

STP participation

The default state for all VLANs (802.1Q or service VFs) is "no spanning-tree shutdown."

For RSTP, there is one RSTP instance in the VCS fabric; the protocol is still VLAN-unaware. This RSTP instance consists of switch ports that have identical VLAN configurations. The VLAN configuration may consist of 802.1Q VLANs or classified VLANs. The STP port state applies to all VLANs (802.1Q and classified) on a port. For switch ports that cannot participate in the same RSTP instance, it is the user's responsibility to shut down spanning tree on these ports. This is the case where ports have overlapping C-TAG classifications and these C-TAGs represent different service VFs.

For MSTP, the VCS Fabric is a single MSTP region. The VLAN-to-instance mapping is applicable only to 802.1Q VLANs. The MSTP VLAN digest calculation is based on 802.1Q VLANs alone. The MSTP state is applied on a port-instance basis (as in previous releases). For switch ports that cannot participate in the same MSTP instance, it is the user's responsibility to shut down spanning tree on these ports. This is the case where ports have overlapping C-TAG classifications.

Service VFs can participate only in MST instance 0 and cannot be assigned to another MST instance. When a service VF is shut down, it is assigned to an internal instance (instance 255) that is always in the forwarding state. The default state for all 802.1Q VLAN and service VFs is "no spanning-tree shutdown."

For PVST, the STP instance is on a per-service-VF basis. PVST can be enabled only on a service VF that has a classification tag. The classification tag identifies the default 802.1Q VLAN in the attached network and is carried in the PVST BPDU; it is also used to form the root RBridge ID. The service VF must have the same C-TAG classification and a nonconflicting classification with other VFs on all RBridge interfaces. This is to ensure the uniqueness of the root RBridge ID for each PVST instance. Consequently, note the following conditions:

- PVST cannot be enabled on a service VF with a VLAN ID greater than 4095 on an access port.
- PVST cannot be enabled on a trunk-mode native VLAN that has no C-TAG classification.

For edge-loop detection (ELD), the protocol instance is on a per-service-VF basis. The user can use the CLI to enable ELD for any service VF on a switch port. Because an ELD configuration applies on a port, the classification for that service VF must exist before the ELD configuration can be accepted.

STP tunneling

BPDU tunneling can be controlled on a per-port basis by means of the existing **bpdu-drop** command. In a multitenancy environment, where a VCS Fabric could be connected to multiple STP-enabled networks, the fabric should tunnel only one instance of the STP BPDU. It is the user's responsibility to enable tunneling on ports that belong to the same STP instance. Other switch ports should have tunneling disabled.

Currently, the **bpdu drop** command controls BPDU forwarding only on the ingress port; the forwarding decision must be applied on the egress port as well. Nontagged BPDUs are tunneled on VCS control VLAN 4095. At the egress RBridge, switch ports that have BPDU tunnels disabled should be removed from the flood membership of the VLAN. For tagged BPDUs (as in PVST), a BPDU is tunneled on its own service-VF flood domain.

PVLANs with service VFs

Private VLAN (PVLAN) configurations apply to service VFs. A service VF can be a primary or a secondary VLAN. However, before an association between the primary and secondary VLAN can be made at the trunk port, the classification of a PVLAN that is a service VF must have been configured. Based on the PVLAN type, the classification is done at the respective promiscuous or host port. That is, the classification of a primary VLAN is done at a promiscuous port, that of a community VLAN at a community port, and that of an isolated VLAN at an isolated port.

The service VF classification is required only if a port operates in trunk mode. An access port does not require classification rules. The validation that is done for the service VF corresponds to the port type.

IP over service VFs

Layer 3 configurations are applicable to service VFs as well. The **interface ve** command for virtual Ethernet (VE) interfaces also applies to service VFs, and all commands under the **interface ve** submode are supported.

Each VE interface is mapped to a service VF, and all such interfaces share the router's MAC address.

A VE interface can be assigned to a VRF instance. Layer 3 runs per-VRF OSPF instances to exchange routes between RBridges in a given VRF instance.

Transport VFs

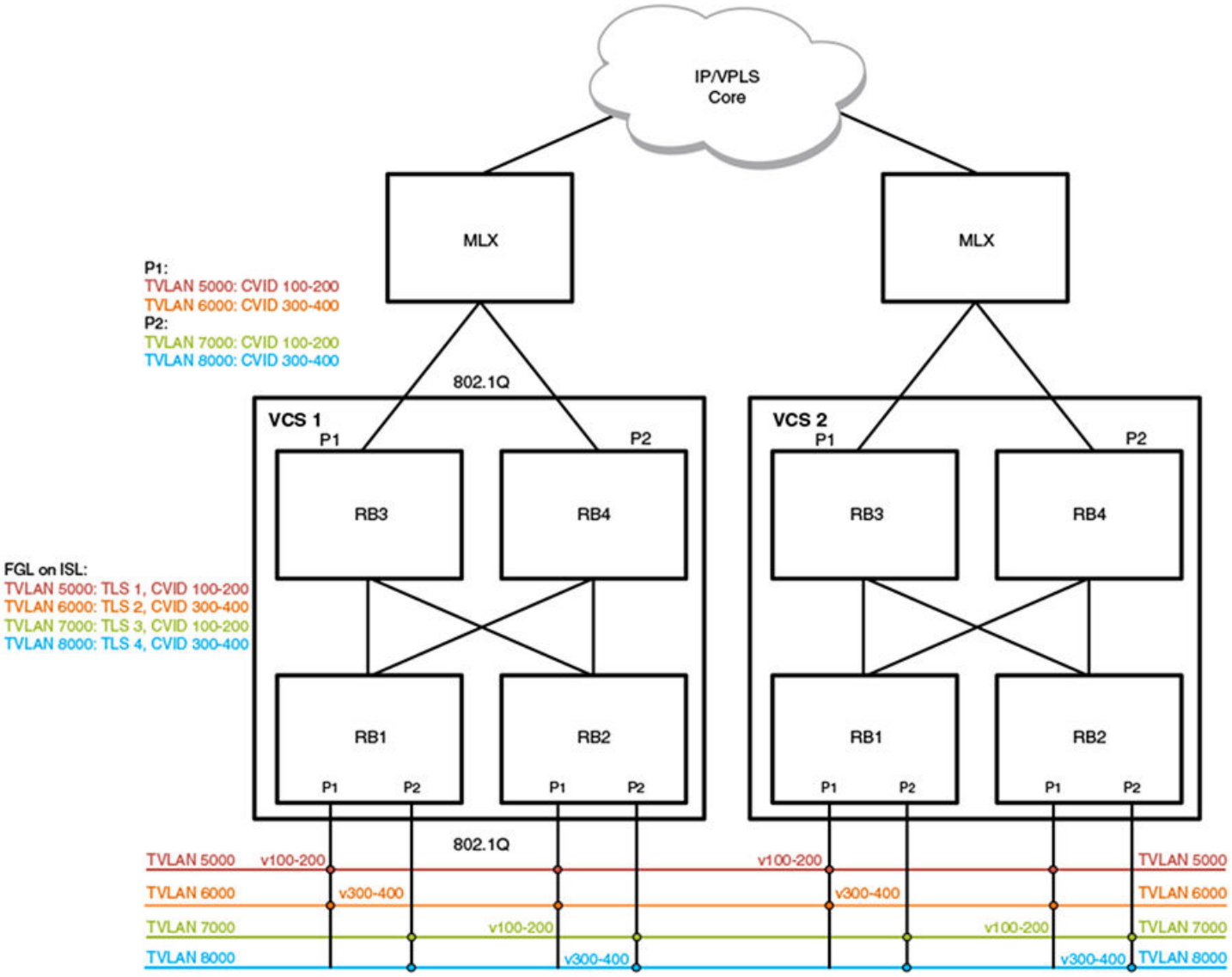
Transport service enables a cloud provider to offer service applicability on a VLAN group level, rather than at a per-individual tenant VLAN level.

The cloud-based service provider needs to provide different service-level agreements (SLAs) to support tenant needs and changes. Transport VFs enable the provider to offer services applicability on a VLAN group level, rather than at a per-individual tenant VLAN level, where the VLAN group represents a specific tenant application. Accordingly, the service offered can be associated only with a single transparent VLAN that collectively represents all the VLANs in the group that participate in supporting a given application. With transport VFs, all VLANs that support the application share the same Layer 2 forwarding domain (individual VLAN isolation is not maintained), and all end stations that participate in a given application or transport VF instance must use unique MAC addresses.

Transport VF topology

The following figure shows four transport VF instances on the respective transparent VLANs (TVLANs) 5000, 6000, 7000 and 8000. Each transport VF carries 101 VLANs in that application. Across the transport VFs, overlapping service VFs among tenants can be supported. The transport VFs can also be extended to another VCS Fabric over a VPLS network.

FIGURE 39 Transport service



The transport VFs that can extend outside of the VCS Fabric are numbered up through 4095, bound by the 802.1Q interface. Because the extension port cannot support QinQ encapsulation, transport VFs that have overlapping C-TAGs cannot be configured on the same port. In the initial release of this feature, no Layer 2 or Layer 3 configuration is supported.

Transport VFs are created by configuring a transparent VLAN that is classified by a set of VLANs at a trunk interface. The operational model is similar to the implementation of SVL (shared VLAN learning). The forwarding behavior is that all service VFs in the transport VF instance are mapped to the transparent LAN forwarding domain (individual VLAN isolation is not maintained), and all end stations participating in the transport VF must have unique MAC addresses.

Transport VF classification rules

The following Transport Layer Security classification rules apply:

- Transport VFs can be configured only on a trunk port, by means of the **transport-service** *tlsid* command in VLAN configuration mode.
- The maximum number of transport VFs that can be configured in this release is 1000.
- Both service and transport VFs can coexist on the same port. A C-TAG is assigned exclusively to a service or transport VF.
- Multiple transport VFs can be configured on the same port.
- Control traffic is classified and handled as follows:
  - Untagged control traffic is not subject to transport VF classification rules. It is handled according to the respective protocol configuration (that is, trapped, dropped, forwarded).
  - Tagged control traffic received on a transport VF is forwarded on the transport VF domain as is data traffic. (PVST cannot be established on a transport VF and is always in the shutdown state.)
  - Tagged control traffic received on a service VF is governed by its respective protocol configuration.
- Transport VF classification can be based on any of the following:
  - A C-TAG range
  - The native VLAN
  - Default traffic (any nonmatching data traffic)
- A vXLAN VNI cannot be mapped to a transport VF.
- Layer 2/Layer 3 configurations are not supported on a transport VF. This means that AMPP/xSTP/PVLAN/RSPAN/ACL/VE configurations are not allowed on a transparent VLAN.

Additional transport VF classification issues

The following table summarizes the classification rules that are applicable to some special VLANs and native VLANs.

TABLE 36 Additional transport VF classification rules

C-TAG	VF classification	Transport VF C-TAG range	Transport VF default VLAN	Notes
Default VLAN 1	No	No	Yes	VLAN 1 cannot be used as a VF classification C-TAG in regular trunk mode. However, it can be used as a VF classification C-TAG in no-default-native-VLAN trunk mode.
Native VLAN	Yes	Yes	Yes	The transport VF default VLAN excludes matching any existing VLAN classification on the port. Because a native VLAN exists as the implicit classification on a trunk port, it is not classified into the default transport VF.
Reserved VLAN	Yes	Yes	Yes	Each platform has a certain VLAN range that is reserved for internal operations. A VLAN in this range can be used as a service or transport C-TAG if the VLAN ID is not internally configured on an edge port. (VLAN 4095 is an internal VLAN and cannot be used.)

Service and transport VF classification with native VLANs

This section addresses two ways to classify service and transport VFs with native VLANs: a default native VLAN mode, and a nondefault native VLAN mode.

Default-native-VLAN trunk mode

When a port is configured in normal trunk mode, a default native VLAN exists. Consequently, the native VLAN complies with the existing native VLAN configuration and forwarding behavior in this mode. The normal behavior for the existing native VLAN is as follows:

- Default native VLAN 1 exists when the port first enters this mode.
- The default native VLAN always exists (either as VLAN 1 or any 802.1Q VLAN ID) and cannot be deleted.
- A tagged native VLAN is always forwarded and cannot be discarded, unless it is blocked by STP.
- Egress tagging behavior depends on acceptable ingress frame types:
  - Tagged egress is enabled only if the acceptable ingress frame type is tagged only.
  - Untagged egress is enabled only if the acceptable ingress frame type includes untagged frames.
  - Egress tagging cannot preserve ingress frame encapsulation.

The following commands are applicable to native VLANs (802.1Q VLANs or service VFs):

- [**no**] **switchport trunk native vlan** *vid* [**ctag** *ctag*]
- [**no**] **dot1q tag native** (global configuration mode)
- [**no**] **switchport trunk tag native-vlan** (interface subtype configuration mode)

The first command is used to define a native 802.1Q VLAN or a native service or transport VF. The last two commands are used to control the ingress acceptable frame and egress tagged behavior. The 802.1Q native VLAN classifications and the role of the respective commands are summarized in the following table.

TABLE 37 802.1Q native VLAN classifications

Ingress allowed frame type	Egress tagging: Untagged only	Egress tagging: Tagged only	CLI commands
None	N/A	N/A	None (tagged 802.1Q native VLAN is always forwarded)
Untagged only	No	N/A	None (tagged 802.1Q native VLAN is always forwarded)
Tagged only	N/A	Yes	**switchport trunk native vlan** *vid*, **switchport trunk tag native-vlan**, AND **vlan dot1q tag native**
Untagged and tagged	Yes	No	**switchport trunk native vlan** *vid*, **no switchport trunk tag native-vlan**, OR **no vlan dot1q tag native**

The service VF classification rules are similar to those for native VLAN classification, but with the following exceptions:

- VLAN 1 cannot be used as a classification CTAG.
- Ingress and egress tagging behavior is controlled by the interface-level configuration, not by the global configuration.

The following summarizes the native service VF (VLAN ID > 4095) classifications that can or cannot be supported with the respective commands.

TABLE 38 Service VF native VLAN classifications

Ingress allowed frame type	Egress tagging: Untagged only	Egress tagging: Tagged only	CLI
None	N/A	N/A	None (tagged native VLAN is always forwarded)
Untagged only	No	N/A	**switchport trunk native vlan** *vid*, **no switchport trunk tag native-vlan**
Tagged only	N/A	Yes	**switchport trunk native vlan** *vid* **ctag** *cvid*,

TABLE 38 Service VF native VLAN classifications (continued)

Ingress allowed frame type	Egress tagging: Untagged only	Egress tagging: Tagged only	CLI
			switchport trunk tag native-vlan
Untagged and tagged	Yes	No	**switchport trunk native vlan** *vid* **ctag** *cvid*, **no switchport trunk tag native-vlan**

The following illustrates configurations that are valid or invalid in regular trunk mode.

```
(config)# int te 5/0/1
(config-if-te-5/0/1)# switchport mode trunk
(config-if-te-5/0/1)# switchport trunk native-vlan-untagged 6000
ERROR: invalid command
# VLAN 1 cannot be used as a classification tag
(config-if-te-5/01)# switchport trunk native-vlan 6000 ctag 1
ERROR: default vlan 1 cannot be classified to a virtual fabric
# CTAG not used elsewhere can be used
(config-if-te-5/01)# switchport trunk native-vlan 6000 ctag 2
# Service VF classification without C-TAG is permitted if ingress frame type allows untagged packet
# at that interface. Global mode tagging control does not apply to a service VF.
(config-if-te-5/01)# switchport trunk native-vlan 7000
ERROR: interface is not configured to accept untagged packet
(config-if-te-5/01)# no switchport trunk tag native-vlan
(config-if-te-5/01)# switchport trunk native-vlan 7000
```

No-default-native-VLAN trunk mode

Another set of native VLAN classifications for service and transport VFs is available in trunk mode without the implicit creation of a default VLAN. The purpose of this trunk mode is to provide flexibility that is not available in default VLAN trunk mode and do the following:

- Provide a raw Layer 2 switchport with no VLAN configuration.
- Allow native VLAN configuration when it is needed.
- Allow the mapping of a native VLAN to a service or transport VF.
- Allow the independent specification of ingress acceptable frame type and egress tagging options.

This trunk mode differs from the default-VLAN trunk mode in the following ways:

- Default VLAN 1 is not implicitly created in this mode.
- Native VLAN commands that are applicable in default-VLAN trunk mode are not supported in this mode.
- Native VLAN commands that are applicable in this mode are not supported in default-VLAN trunk mode.

Because of the different mode behaviors, the user must be aware of the following:

- A port in default-VLAN trunk mode cannot use the new classifications. For example, an AMPP profile port operates in default-VLAN trunk mode and therefore the new classifications are not supported in this case.
- When a port is configured in this mode there is no assumption that VLAN 1 exists. The default VLAN 1 must be configured explicitly. For example, PVST VLAN 1 is not enabled on an interface unless that VLAN is configured explicitly.

The following commands are used to support different native VLAN configurations:

- **switchport mode trunk-no-default-native**
- [**no**] **switchport trunk native-vlan-untagged** *vid*
- [**no**] **switchport native-vlan-xtagged** *vid* [**ctag** *cvid*] **egress** [**tagged** | **untagged** | **any**]

The service and transport VF native VLAN classifications and the role of the respective commands are summarized in the following table.

TABLE 39 Service and transport VF native VLAN classifications

Ingress allowed frame type	Egress tagging: Untagged only	Egress tagging: Tagged only	CLI commands
None	N/A	N/A	**switchport mode trunk-no-default-native**
Untagged only	Yes	N/A	**switchport trunk native-vlan-untagged** *vid*
Tagged only	N/A	Yes	Use regular VLAN classification: **switchport trunk allow vlan add** *vid* [**ctag** *cvid*]
Untagged and tagged	Yes	Yes	**switchport trunk native-vlan-xtagged** *vid* [**ctag** *cvid*] [**ctag** *cvid*] **egress** [**untagged** | **tagged** | **any**]

The following configuration rules apply in no-default-native-VLAN trunk mode.

Rules for the **switchport trunk-no-default-native** command:

- There is no automatic native VLAN configuration.
  - VLAN 1 is not created automatically when the port is configured in this mode.
  - VLAN 1 is not a default VLAN. It is just another C-TAG that is available for classification.
  - The user can explicitly create VLAN 1 to achieve a default-VLAN trunk mode configuration, by using the **switchport trunk allow vlan add** *vid* command.
- An untagged or unclassified frame is discarded by default.
- All switchport configurations except the following native VLAN commands in default-VLAN trunk mode continue to be supported:
  - **switchport trunk tag native-vlan**
  - **switchport trunk native vlan** *vlan_id*
  - **dot1q tag native-vlan** (a global command that does not apply to a port)
- All service and transport VF configurations that are available in default-VLAN trunk mode continue to be supported. An 802.1Q native VLAN can be classified to a service or transport VF with the new commands.

Rules for the [**no**] **switchport native-vlan-untagged** *vid* command are as follows:

- This command accepts untagged packets only and allows egress untagged packets. The VLAN ID can be a regular 802.1Q VLAN ID or a service or transport VF.
- The **no** form of this command removes the native VLAN classification, and untagged frames are dropped.

Rules for the [**no**] **switchport native-vlan-xtgagged** *vid* [**ctag** *cvid*] **egress** [**tagged** | **untagged** | **any**] command are as follows:

- This command accepts untagged or tagged 802.1Q VLANs and service and transport VFs and specifies egress tagging behavior. If a VLAN is an 802.1Q VLAN or a service VF, the supported egress tagging behavior is untagged or tagged. If a VLAN is a transport VF, the **egress** option must be **any**, to indicate that the ingress frame encapsulation must be preserved.
- The **no** form of this command removes the native VLAN classification. Untagged frames and the associated tagged 802.1Q VLAN are dropped.

The following illustrates configuration in no-default-native-VLAN trunk mode.

```
device(config)# int vlan 5000
device(config)# int vlan 6000
device(config-Vlan-6000)# transport-service 60
device(config)# int vlan 7000
device(config-Vlan-7000)# transport-service 70
```

In the new mode, the default behavior is to drop all packets.

```
(config)# int te 1/0/1
(config-if-te-1/0/1)# switchport mode trunk-no-default-native
```

VLAN configuration similar to that of regular trunk mode can be achieved explicitly after VLAN 1 is configured as a tagged VLAN.

```
device(config-if-te-1/0/1)# switchport trunk allow vlan add 1
```

The following creates native VLAN 10. All service and transport VFs can continue to coexist on the same port.

```
device(config-if-te-1/0/1)# switchport trunk native-vlan-untagged 10
device(config-if-te-1/0/1)# switchport trunk allow vlan 6000 ctag 100-200
device(config-if-te-1/0/1)# switchport trunk default-vlan 7000
```

The following accepts ingress tagged or untagged frames, but egress frames must be tagged only. This is not allowed in default-VLAN trunk mode.

```
device(config-if-te-1/0/1)# switchport trunk native-vlan-xtagged 10 egress tagged
```

The following classifies a tagged native VLAN to service VF 5000.

```
device(config-if-te-1/0/1)# switchport trunk native-vlan-xtagged 5000 ctag 10 egress tagged
```

The following classifies VLAN 10 to transport VF 6000. Because this native VLAN is a transport VF, the option for **egress** is **any.**

```
device(config-if-te-1/0/1)# switchport trunk native-vlan-xtagged 6000 ctag 10 egress any
device(config-if-te-1/0/1)# switchport trunk allow vlan 6000 ctag 100-200
```

The following configurations are valid or invalid in no-default-native-VLAN mode.

```
device(config)# int te 5/0/1
device(config-if-te-5/0/1)# switchport mode trunk-no-default-native
device(config-it-te-5/0/1)# switchport trunk tag-native
    ERROR: Port mode not set to trunk
device(config-it-te-5/0/1)# switchport trunk native vlan 2
    ERROR: Port mode not set to trunk
    # C-TAG classification to service or transport VF mapping is still 1 to 1.
    # Rejected as duplicate classification even when same CTAG-to-VF mapping is given.
device(config-if-te-5/0/1)# switchport trunk native-vlan-xtagged 6000 ctag 10 egress any
device(config-if-te-5/0/1)# switchport trunk allow vlan add 6000 ctag 10-20
    ERROR: ctag already used in other classification
device(config-if-te-5/0/1)# switchport trunk allow vlan 7000 ctag 10
    ERROR: ctag already used in other classification
```

Configuring and managing Virtual Fabrics

Whenever Virtual Fabrics configuration changes are made, they are saved automatically.

Refer also to Virtual Fabrics overview on page 177.

Configuring a service VF instance

Configuring a service VF instance consists of enabling VF configuration in the fabric, and then configuring a service VF instance that is greater than 4095.

The current release of this feature supports up through 8192 VLANs, with 8191 being the largest number that can be assigned.

The **vcs virtual-fabric enable** command, issued in global configuration mode, expands the VLAN ID address space beyond the 802.1Q limit in the fabric, allowing VLANs with IDs greater than 4095 to be supported, up through 8191. The command, which is accepted only if the state of the fabric is capable of supporting VLAN virtualization, does not disrupt existing 802.1Q data traffic in the fabric.

Enabling service VF configuration

In global configuration mode, issue the **vcs virtual-fabric enable** command:

```
device(config)# vcs virtual-fabric enable
```

NOTE
Use the **no** form of this command to disable service VF configuration.

Creating a service VF instance

In global configuration mode, use the **interface vlan** *vlan_id* command, where *vlan_id* is a number greater than 4095, through 8191, and is not a reserved VLAN.

```
device(config)# interface vlan 5000
```

NOTE
Use **no interface vlan** *vlan_id* to delete a service VF.

Configuring a transport VF instance

The following example command sequence illustrates the configuration of three transport VF instances.

```
device(config)# interface vlan 6001
device(config-Vlan-6001)# transport-service 1
device(config)# interface vlan 6002
device(config-Vlan-6002)# transport-service 2
device(config)# interface vlan 6003
device(config-Vlan-6003)# transport-service 3
```

Configuring VF classification to a trunk interface

The following command sequence illustrates the configuration of VF classification to a trunk interface.

```
device(config)# interface TenGigabitEthernet 1/0/1
device(conf-if-te-1/0/1)# switchport
device(conf-if-te-1/0/1)# switchport mode trunk
device(conf-if-te-1/0/1)# switchport trunk allowed vlan add 5000 ctag 100
device(conf-if-te-1/0/1)# switchport trunk allowed vlan add 5001 ctag 101
```

Configuring transport VF classification to a trunk interface

The following example command sequence illustrates the configuration of VF classification to a trunk interface.

```
device(config)# interface TenGigabitEthernet 1/0/1
device(conf-if-te-1/0/1)# switchport
device(conf-if-te-1/0/1)# switchport mode trunk
device(conf-if-te-1/0/1)# switchport trunk allowed vlan add 6000 ctag 600-610
```

Creating a default VLAN with a transport VF to a trunk interface

The following example command sequence illustrates the configuration of a default VLAN with a transport VF to a trunk interface.

```
device(config)# interface TenGigabitEthernet 1/0/1
device(conf-if-te-1/0/1)# switchport
device(conf-if-te-1/0/1)# switchport mode trunk
device(conf-if-te-1/0/1)# switchport trunk default-vlan 7000
```

Configuring a native VLAN in regular VLAN trunk mode

The following examples illustrate the configuration of a native VLAN in regular VLAN trunk mode, where the default native VLAN 1 exists.

The following native VLAN commands are supported in this trunk mode:

- **switchport trunk tag native-vlan**
- **switchport trunk native-vlan**

1. Create native VLAN 10.

   NOTE

   VLAN 1 is the default VLAN in this mode.

   ```
   device(config)# interface te 2/1/1
   device(config-if-te-2/1/1)# switchport mode trunk
   device(config-if-te-2/1/1)# switchport trunk native-vlan 10
   ```

2. Configure untagged native VLAN (service VF) 5000 and transport VLAN (transport VF) 6000, and make VLAN 7000 the default VLAN.

   ```
   device(config)# interface te 2/1/1
   device(config-if-te-2/1/1)# switchport mode trunk
   device(config-if-te-2/1/1)# no switchport trunk tag native
   device(config-if-te-2/1/1)# switchport trunk native-vlan 5000
   device(config-if-te-2/1/1)# switchport trunk allow vlan add 6000 ctag 100-200
   device(config-if-te-2/1/1)# switchport trunk default-vlan 7000
   ```

3. Configure transport VLAN 6000 to accept the C-TAG range 100 through 200, as well as tagged or untagged native VLAN traffic.

   ```
   device(config)# interface te 2/1/1
   device(config-if-te-2/1/1)# switchport mode trunk
   device(config-if-te-2/1/1)# no switchport trunk tag native
   device(config-if-te-2/1/1)# switchport trunk native-vlan 6000 ctag 10
   device(config-if-te-2/1/1)# switchport trunk allow vlan add 6000 ctag 100-200
   ```

Configuring a native VLAN in no-default-native-VLAN trunk mode

The following examples illustrate the configuration of a native VLAN in a trunk mode where the native VLAN does not exist. The native VLAN must be explicitly created in this mode.

The following native VLAN commands are supported in this mode:

- **switchport trunk native-vlan-untagged**
- **switchport trunk native-vlan-xtagged**

1. Configure a trunk port without a default native VLAN, then explicitly configure the native VLAN.

```
device(config)# interface te 2/1/1
device(config-if-te-2/1/1)# switchport mode trunk-no-default-native
device(config-if-te-2/1/1)# switchport trunk native-vlan-xtagged 10 egress tagged
```

2. In no-default-native-VLAN trunk mode, configure untagged native VLAN 5000 and transport VLAN 6000, and make VLAN 7000 the default VLAN.

```
device(config)# interface te 2/1/1
device(config-if-te-2/1/1)# switchport mode trunk-no-default-native
device(config-if-te-2/1/1)# switchport trunk native-vlan-untagged 5000
device(config-if-te-2/1/1)# switchport trunk allow vlan add 6000 ctag 100-200
device(config-if-te-2/1/1)# switchport trunk default-vlan 7000
```

3. In no-default-native-VLAN trunk mode, configure transport VLAN 6000 to accept C-TAG range 100 through 200 as well as tagged or untagged native VLAN traffic..

```
device(config)# interface te 2/1/1
device(config-if-te-2/1/1)# switchport mode trunk-no-default-native
device(config-if-te-2/1/1)# switchport trunk native-vlan-xtagged 6000 ctag 10 egress any
device(config-if-te-2/1/1)# switchport trunk allow vlan add 6000 ctag 100-200
```

Configuring additional Layer 2 service VF features

This section addresses additional features that are available on trunk ports once a Virtual Fabric is established.

Configuring private service VFs

The following examples illustrate the configuration of a variety of private VLANs (PVLANs) as service VFs and their association on access and trunk ports.

Refer also to PVLANs with service VFs on page 195.

Configuring service VFs and defining and associating PVLANs

1. In global configuration mode, use the **interface vlan** command to create VLAN instances that are greater than 4095, through 8191.

```
device(config)# interface vlan 5000
device(config)# interface vlan 6000
device(config)# interface vlan 7000
```

2. In VLAN configuration mode, use the **private-vlan** command to create the three types of PVLAN.

```
device(config)# interface vlan 5000
device(conf-vlan-5000)# private-vlan primary
device(config)# interface vlan 6000
device(conf-vlan-6000)# private-vlan isolated
device(config)# interface vlan 7000
device(conf-vlan-7000)# private-vlan community
```

3. Using the **private-vlan association** command, associate the secondary PVLANs with the primary PVLAN.

```
device(config)# interface vlan 5000
device(conf-vlan-5000)# private-vlan association add 6000
device(conf-vlan-5000)# private-vlan association add 7000
```

Configuring physical interfaces

1. Create classification rules for the primary and secondary VLAN at the respective primary and host ports.

   The classification must be done before the primary-to-secondary VLAN associations are specified. In following example, the same C-TAG is used to classify the primary and secondary VLANs.

   Interface te 1/0/1 is a primary trunk port.

```
device(config)# interface te 1/0/1
device(conf-if-te-1/0/1)# switchport mode private-vlan trunk promiscuous
device(conf-if-te-1/0/1)# switchport trunk allow vlan add 5000 ctag 10
```

   Interface te 1/0/2 is an isolated trunk port.

```
device(config)# interface te 1/0/2
device(conf-if-te-1/0/2)# switchport mode private-vlan trunk host
device(conf-if-te-1/0/2)# switchport trunk allow vlan add 6000 ctag 10
```

   Interface te 0/3 is a community trunk port.

```
device(config)# interface te 1/0/3
device(conf-if-te-1/0/3)# switchport mode private-vlan trunk host
device(conf-if-te-1/0/3)# switchport trunk allow vlan add 7000 ctag 10
```

2. Configure the PVLAN association on the promiscuous trunk port.

```
device(config)# interface te 1/0/1
device(conf-if-te-1/0/1)# switchport mode private-vlan trunk promiscuous
device(conf-if-te-1/0/1)# switchport trunk allowed vlan add 400
device(conf-if-te-1/0/1)# switchport trunk allowed vlan add 5000 ctag 10
device(conf-if-te-1/0/1)# switchport private-vlan mapping 5000 add 6000
device(conf-if-te-1/0/1)# switchport private-vlan mapping 5000 add 7000
```

3. Configure the PVLAN association on the promiscuous access port.

```
device(config)# interface te 1/1/1
device(conf-if-te-1/1/1)# switchport mode private-vlan promiscuous
device(conf-if-te-1/1/1)# switchport private-vlan mapping 5000 add 6000
device(conf-if-te-1/1/1)# switchport private-vlan mapping 5000 add 7000
```

4. Configure the isolated PVLAN on the trunk port.

```
device(config)# interface te 1/0/2
device(conf-if-te-1/0/2)# switchport mode private-vlan trunk host
device(conf-if-te-1/0/2)# switchport private-vlan host-association 5000 6000
```

5. Configure the isolated PVLAN on the access port.

```
device(config)# interface te 1/1/2
device(conf-if-te-1/1/2)# switchport mode private-vlan host
device(conf-if-te-1/1/2)# switchport private-vlan host-association 5000 6000
```

6. Configure the community PVLAN on the trunk port.

```
device(config)# interface te 1/0/3
device(conf-if-te-1/0/3)# switchport mode private-vlan trunk host
device(conf-if-te-1/0/3)# switchport trunk allowed vlan add 7000 ctag 10
device(conf-if-te-1/0/3)# switchport private-vlan host-association 5000 7000
```

7. Configure the community PVLAN on the access port.

```
device(config)# interface te 1/1/3
device(config-if-te-1/1/3)# switchport mode private-vlan host
device(config-if-te-1/1/3)# switchport mode private-vlan host 5000 7000
```

8. Configure the PVLAN trunk port.

```
device(config)# interface te 1/4/1
device(conf-if-te-1/4/1)# switchport private-vlan trunk allowed vlan add 5000 ctag 10
device(conf-if-te-1/4/1)# switchport private-vlan trunk allowed vlan add 6000 ctag 20
device(conf-if-te-1/4/1)# switchport private-vlan trunk allowed vlan add 7000 ctag 30
device(conf-if-te-1/4/1)# switchport mode private-vlan trunk
```

The trunk port can have both PVLAN and regular VF configurations. The following configures PVLAN VFs.

```
device(conf-if-te-1/4/1)# switchport trunk allowed vlan add 5000 ctag 10
device(conf-if-te-1/4/1)# switchport trunk allowed vlan add 6000 ctag 10
device(conf-if-te-1/4/1)# switchport trunk allowed vlan add 6000 ctag 10
device(conf-if-te-1/4/1)# switchport private-vlan association trunk 5000 6000
device(conf-if-te-1/4/1)# switchport private-vlan association trunk 5000 7000
```

The following configures non-PVLAN VFs.

```
device(conf-if-te-1/4/1)# switchport private-vlan trunk allowed vlan add 400
device(conf-if-te-1/4/1)# switchport private-vlan trunk allowed vlan add 5000 ctag 100
```

Understanding PVLAN configuration failures

The following examples illustrate success or failure of PVLAN configurations.

- If a PVLAN is a service VF and the classification does not exist for this port, the PVLAN association executed on this interface will fail. This following commands fail because there is no classification rule for primary service VF 5000.

```
device(config)# interface te 1/0/1
device(conf-if-te-1/0/1)# switchport mode private-vlan trunk promiscuous
device(conf-if-te-1/0/1)# switchport private-vlan mapping 5000 add 200
%%ERROR: Primary Vlan should have a ctag associated with it on this port.
device(config)# interface te 1/0/2
device(conf-if-te-1/0/2)# switchport mode private-vlan trunk host
device(conf-if-te-1/0/2)# switchport private-vlan host-association 100 add 6000
%%ERROR: Secondary Vlan should have a ctag associated with it on this port.
```

- The following command succeeds because the primary port is an access port.

```
device(conf-if-te-1/1/1)# switchport mode private-vlan promiscuous
device(conf-if-te-1/1/1)# switchport private-vlan mapping 5000 add 6000
```

- The following command succeeds because the secondary port is an access port.

```
device(conf-if-te-1/1/2)# switchport mode private-vlan host
device(conf-if-te-1/1/2)# switchport private-vlan host-association 5000 add 6000
```

Configuring MAC groups

You can assign individual MAC addresses or a group of MAC addresses to a service VF at an access port.

Creating a MAC group instance and assigning MAC addresses

1. In global configuration mode, create a MAC group instance to define the MAC addresses of end stations, by using the **mac-group** *mac-group-id* command.

   The value of *mac-group-id* ranges from 1 through 500.

```
device(config)# mac-group 1
```

2. In MAC group configuration mode, use the **mac** *mac_address* command to add a MAC address in hexadecimal notation.

   NOTE
   Ranging is not allowed. Leading zeros can be omitted.

```
device(config-mac-group 1)# mac 0002.0002.0002
```

Deleting a MAC group or MAC address

1. This command deletes a MAC group.

```
device(config)# no mac-group 1
```

2. This command deletes a MAC address from a MAC group.

   NOTE
   Only one MAC address can be deleted at a time.

```
device(config)# mac-group 1
device(config-mac-group 1)# no mac 0004.0004.0004
```

Configuring an interface for service VF MAC address access

The following illustrates various options and errors that can occur in configuring an interface for service VF MAC address access.

1. In interface configuration mode, set switchport mode to access and change the default VLAN to a service VF.

   NOTE
   The default VLAN must be unique. It must not be the same as that used for another MAC classification.

```
device(config)# int te 2/0/1
device(config-if-te-2/0/1)# switchport access vlan 5000
```

2. Classify the access VLAN by means of a MAC address.

```
device(config-if-te-2/0/1)# switchport access vlan 6000 mac 0002.0002.0002
```

3. Classify another access VLAN on the same interface by means of a MAC address.

```
device(config-if-te-2/0/1)# switchport access vlan 7000 mac 0004.0004.0004
```

NOTE
Frames that do not match 0002.0002.0002 or 0004.0004.0004 are classified into service VF 5000.

4. Create a MAC group to be used to classify a VLAN.

```
device(config)# mac-group 1
device(config-mac-group 1)# mac 0002.0002.0002
device(config-mac-group 1)# mac 0005.0005.0005
device(config-mac-group 1)# mac 0008.0008.0008
```

5. Configure another service VF on the same interface by means of a MAC group.

```
device(config-if-te-2/0/1)# switchport access vlan 7000 mac-group 1
Error: mac address is already used in another classification
```

NOTE
The MAC address cannot be used in another service VF classification.

6. Configure another interface with a third service VF and classify the VLANs by MAC group and MAC address, respectively.

```
device(config-if-te-3/0/1)# switchport mode access
device(config-if-te-3/0/1)# switchport access vlan 7000 mac-group 1
device(config-if-te-3/0/1)# switchport access vlan 8000 0008.0008.0008
device(config-if-te-3/0/1)# %Error: Mac-address/Mac-group is overlapping with another Mac-address/
Mac-group configuration on the same port.
```

Configuring Layer 3 service VF features

Layer 3 configurations are applicable to service VFs, by means of existing **interface ve** commands.

Each virtual Ethernet (VE) interface is mapped to a service VF, and all VE interfaces share the router's MAC address. A VE interface can be assigned to a VRF instance, with per-VRF OSPF instances exchanging routes between RBridges in that VRF instance. Virtual Router Redundancy Protocol (VRRP) provides high availability.

Refer to IP over service VFs on page 195.
Do the following to configure Layer 3 service VF features.

1. Create a service VF and add it to the trunk port with a C-TAG.

```
device(config)# interface vlan 5000
device(config)# interface te 3/1/1
device(conf-if-te-3/1/1)# switchport
device(conf-if-te-3/1/1)# switchport mode trunk
device(conf-if-te-3/1/1)# switchport trunk allowed vlan add 5000 ctag 50
```

2. Enable VRF and VRRP on an RBridge.

```
device(config)# rbridge-id 3
device(config-rbridge-id-3)# protocol vrrp
```

3. Create a VRF instance and set OSPF routing parameters.

   NOTE
   BGP is also supported in the appropriate context.

```
device(config)# ip vrf VRF_CUST1
```

   a) Use the **rd** (Route Distinguisher) command to assign an administrative number.

```
device(config-vrf-CUST1)# rd 10.1.1.1:1
```

   b) Set the OSPF address family to IPv4.

```
device(config-vrf-CUST1)# address-family ipv4
device(config-vrf-CUST1)# exit
```

   c) Return to config mode.

```
device(config-vrf-CUST1)# exit
```

4. Create the VE interface on the service VF and configure VRF forwarding and VRRP.

   a) Configure forwarding for a VRF instance.

```
device(config-ve-5000)# ip vrf forwarding VRF_CUST1
```

   b) In RBridge ID configuration mode, create a virtual Ethernet interface, and assign the service VF and IP address. Be sure to enable the interface.

```
device(config)# rbridge-id 3
device(config-rbridge-id-3)# interface ve 5000
device(config-ve-5000)# ip address 10.1.1.1/24
device(config-ve-5000)# no shutdown
```

   c) Create a VRRP group and assign a virtual IP address.

```
device(config-ve-5000)# vrrp-group 22
device(config-vrrp-group-22)# virtual-ip 10.1.1.1
```

Configuring Layer 2 extension over Layer 3 with Virtual Fabrics

The VLAN-based broadcast domain is extended over a Layer 3 network without the need for an orchestrator.

The extension function commonly resides at the aggregation layer, either embodied in the aggregation node itself, or hanging as a service off the aggregation node in a one-arm topology. This feature resides in the network spine, with up to four RBridges participating. Tunnels are provided through router ports and switch ports.

In addition, the user can collect statistics on tunnel and gateway traffic, through the range of attached VLANs. Both transmit and receive traffic can be monitored for a single tunnel and range of VLANs, or all tunnels and a range of VLANs. Support is also provided for similar sFlow monitoring.

Supported and unsupported features

The following new features are supported:

- Layer 3 forwarding for extended Layer 2 segments, for connections local to the fabric or through a tunnel
- Extension as part of the main fabric or through a "1-arm" topology in which VLAN and VXLAN traffic occupies the same path through connected subnets

- Ability to use VLANs, service VFs, and transport VFs for extension, in a flexible deployment model
- Nondefault VRFs
- Fully meshed tunnels between fabrics, with BUM through head-end replication
- Multicast (without multicast traffic optimization)
- Split horizon in the data plane for loop avoidance
- MAC address learning on tunnels
- Layer 2 fault domain isolation across fabrics
- Layer 3 protocol isolation controls (off by default)
- MAC, IPv4, and IPv6 ingress ACLs
- sFlow configurations

The following features are not supported:

- VXLAN-to-VLAN routing, and vice versa
- VNI translation (because of a single VNI domain)
- Exposing of ARPs behind the physical network as a unicast map
- VXLAN transit
- VLAG as undelay
- Loopback IP as VTEP IP
- BUM optimization (including ARP-related optimization)
- Loop-detection protocols over tunnels. There is no support for STP or tunneling STP BPDUs. (The user is responsible for avoiding loops over any interfaces.)
- Keepalives on tunnels
- TRILL with VXLAN
- Tunnels as a SPAN destination
- vLAG of tunnels to the same remote VTEP
- PACL on tunnels (VACL is supported)
- Tunnels as profile-ports. (However, a VLAN that spans a tunnel can have a profiled physical port, as in the previous release.)
- Router port IP addresses or VE IP addresses used as VTEP addresses

Note the following considerations with respect to ACLs, SPAN, sFlow, and statistics, as these features share the ternary content-addressable memory (TCAM) tables for the corresponding features of nontunnel traffic:

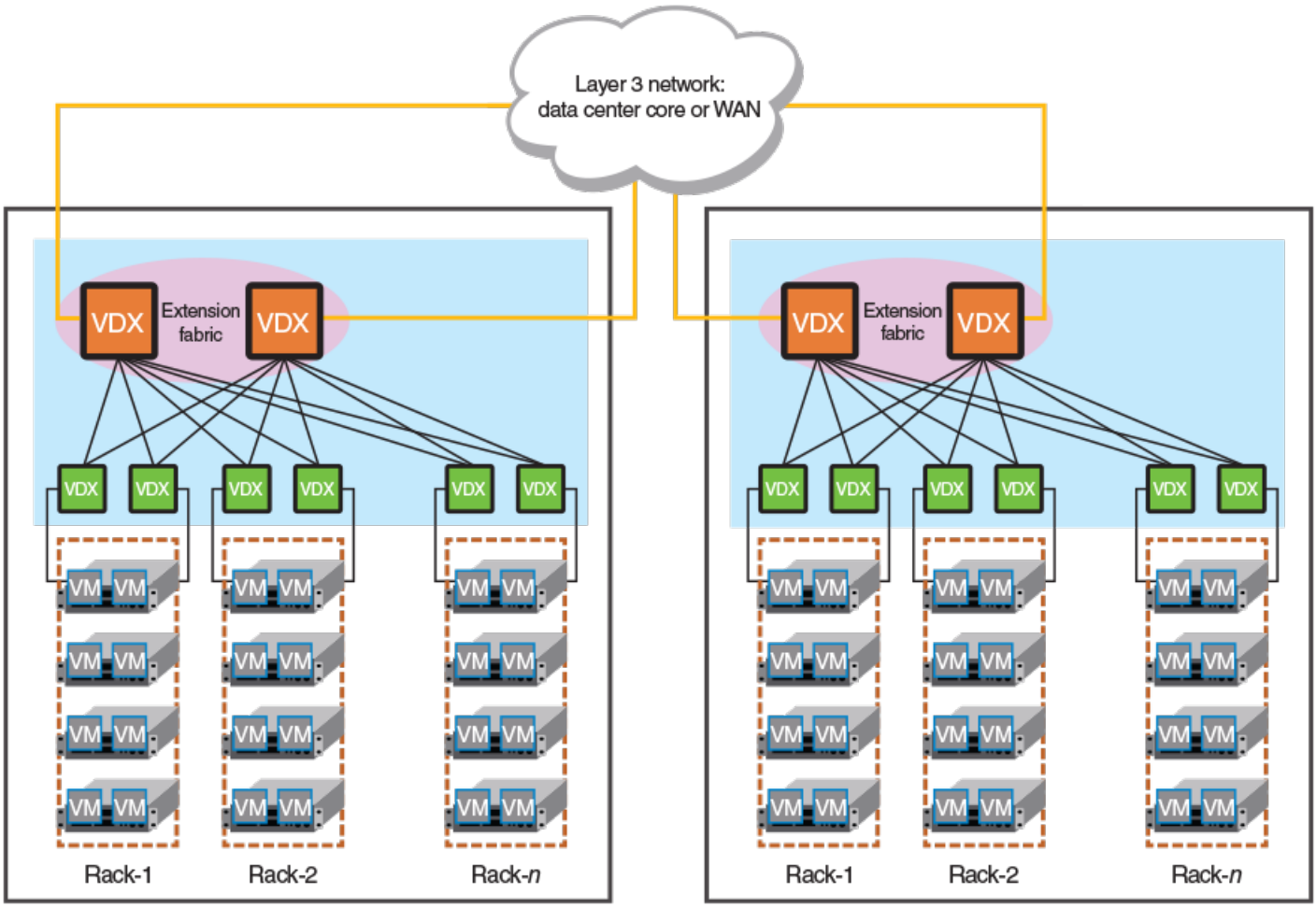
- Ingress statistics and sFLow both need a counting ability. Consequently, if both functions are enabled and a particular traffic type satisfies both classifiers, then statistics alone takes effect and sFlow does not.
- Ingress SPAN and sFlow both use the same resources. Consequently, if both functions are enabled and a particular traffic type satisfies both classifiers, then sFlow takes effect and SPAN does not.
- Ingress statistics and ACLs both use the same resources. Consequently, if both functions are enabled and a particular traffic type satisfies both classifiers, then ACLs take effect and statistics does not.
- The scaling limits of the features are constrained by the profiles applied on the system.

High-level topologies

The following illustrates an example high-level topology for Layer 2 extension at the aggregation layer only. Note the following restrictions for this topology:

- If a vLAG is deployed, the number of aggregation VDX devices supporting extension is limited to the size of the vLAG with respect to RBridges and the number of RBridges in a VRRP group. (This restriction does not apply if router ports are used.)
- A static route must be configured on the upstream device, or VRRP must use the First Hop Redundancy Protocol (FHRP) gateway.

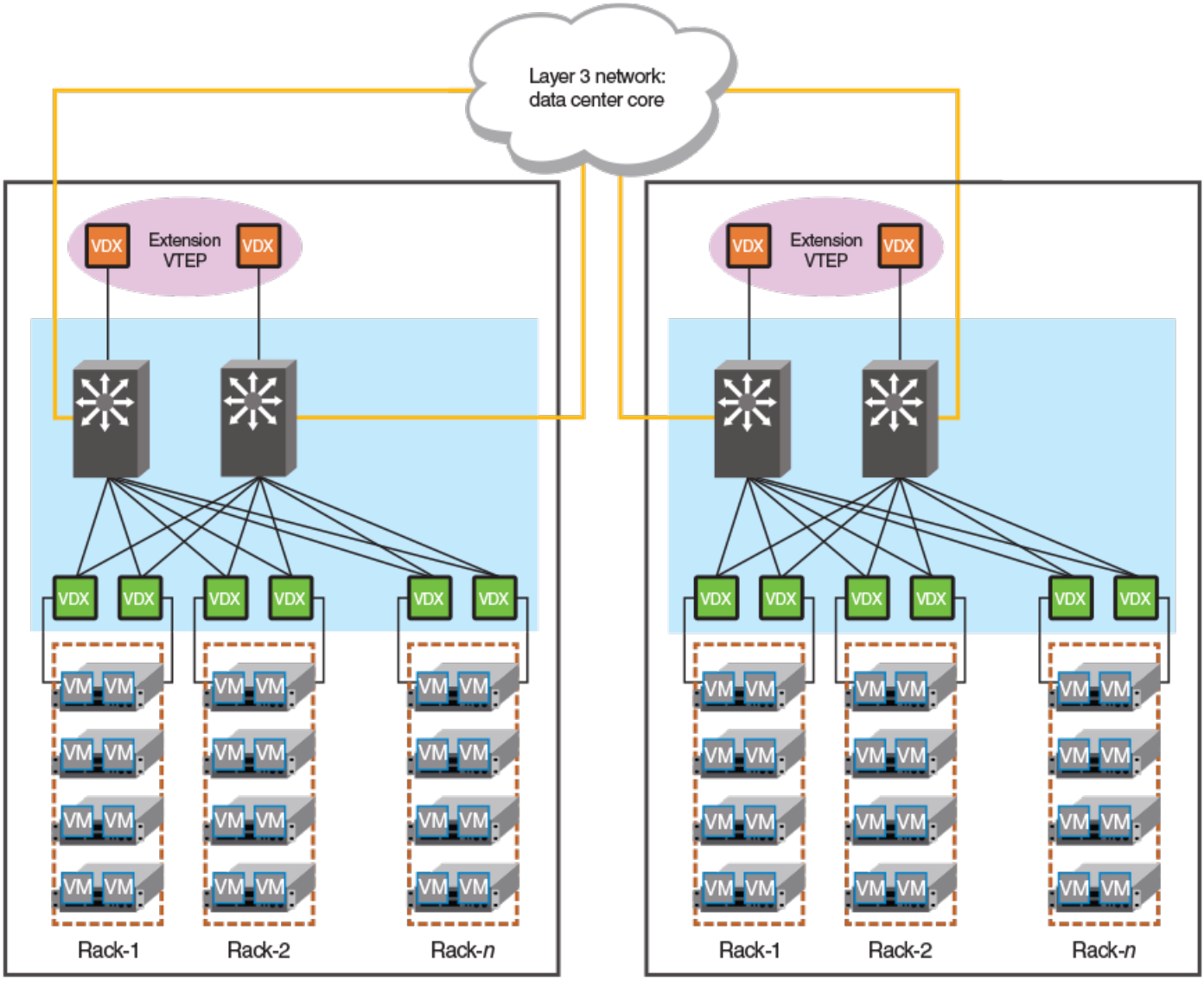
FIGURE 40 Layer 2 extension at the aggregation layer only



The following illustrates an example high-level topology that uses VXLAN virtual network interface (VNI) tunnel endpoints (VTEPs) in separate fabrics with a one-arm VDX deployment. Note the following restrictions for this topology:

- The fabric must be extended through an 802.1Q port.
- Only vLAG connectivity is supported. Router ports are not supported to the aggregation device.

FIGURE 41 Layer 2 extension with VTEPs in separate fabrics (one arm)



VXLAN overlay gateway for Layer 2 extension: summary of steps

This overview summarizes the basic steps that are required to extend a Layer 2 VLAN-based broadcast domain over a Layer 3 network, without the use of an orchestrator such as the NSX Controller.

The basic provisioning and configuration steps can be summarized as follows:

1. Ensure that the network administrator has created a broadcast domain on the physical side. This domain consists of 802.1Q VLANs, as well as the service and transport VFs available in a Virtual Fabrics context.
2. Create a VXLAN tunnel endpoint (VTEP) container with a "type" qualifier that specifies the use of the VTEP for Layer 2 extension.

NOTE
If you are configuring a loopback interface to serve as a VTEP, you must manually configure distinct router IDs, by means of the **ip router-id** command, for use by routing protocols.

3. Create containers for the remote sites. These containers will have the IP addresses of the site. This leads to the automatic creation of VTEP tunnels to the remote sites. The tunnels refer to VTEP as the source endpoint, and to the IP addresses of the remote site as the destination endpoint. (Multiple addresses, corresponding to a LAG of tunnels, are not supported in this release.)

   NOTE
   A full mesh of tunnels must be created.

4. A VLAN-to-VNI map is created for the VTEP. This provides a global mapping across all sites for the VTEP. The map can be created automatically, where VLAN-to-VNI mapping is autogenerated, or the user can specify each map explicitly.
5. In the site container, extend the required VLANs (including service and transport VFs) that must be extended to a given remote site.

Configuring the VXLAN overlay gateway for Layer 2 extension

This task configures the nondefault mode of configuration for a VXLAN overlay gateway.

The network administrator must first configure a broadcast domain.

1. In VXLAN overlay gateway configuration mode, set the type of gateway to support Layer 2 extension.

```
device(config)# overlay-gateway gateway1
device(config-overlay-gw-gateway1)# type layer2-extension
```

2. **Using automatic VLAN-to-VNI mapping**: You can optionally use the **map vlan vni auto** command to configure VLAN-to-VNI mappings automatically if they have not been configured manually.

```
device(config-overlay-gw-gateway1)# map vlan vni auto
```

3. **Using explicit VLAN-to-VNI mapping**: The following configures VLAN-to-VNI mappings manually.

```
device(config-overlay-gw-gateway1)# map 10,20-22 vni 5000-5002,6000
```

   NOTE
   Automatic and explicit mappings are mutually exclusive. Also, the VLANs used in the **extend vlan** configuration must be a subset of the VLANs used in the **map vlan** configuration. In other words, a VLAN must have a VNI mapping for it to be extended across sites.

4. Configure the site, which includes providing a site name and entering VXLAN overlay gateway site configuration mode, providing an IPv4 address for the VNI tunnel endpoint (VTEP) in that mode, extending VLANs, and enabling the tunnels administratively.
   a) Enter a site name and enter VXLAN overlay gateway site configuration mode.

```
device(config-overlay-gw-gateway1)# site mysite1
device(config-overlay-gw-gateway1-mysite1)#
```

   b) Configure an IPv4 address for the tunnel endpoints.

```
device(config-overlay-gw-gateway1-mysite1)# ip address 10.10.10.1
```

      NOTE
      Only one IPv4 address is allowed.

   c) Extend the VLANs.

```
device(config-overlay-gw-gateway1-mysite1)# extend vlan add 10,20-22
```

   d) Enable the tunnels administratively.

```
device(config-overlay-gw-gateway1-mysite1)# no shutdown
```

5. The following substeps attach the RBridge IDs to the VXLAN gateway instance, set the IP address of the gateway, and activate the instance.
   a) Enter the **attach rbridge-id** command to attach existing RBridge IDs to this VXLAN gateway instance. You can specify a range of RBridge IDs up to a maximum of four.

```
device(config-overlay-gw-gateway1)# attach rbridge-id add 1-2
```

   b) Enter the **ip interface ve** *veid* **vrrp-extended-group** *group-ID* command to set the IP address of the VXLAN overlay gateway, as shown in the following example.

```
device(config-overlay-gw-gateway1)# ip interface ve 10 vrrp-extended-group
100
```

      The command accepts the VE interface ID and VRPP-E group ID, then sets the VXLAN overlay gateway's IP address as identical to the already configured VRRP-E virtual IP address. Tunnels that form use this IP address as the source IP address for outgoing packets.
   c) Enter the **activate** command to activate this gateway instance.

```
device(config-overlay-gw-gateway1)# activate
```

      This enables all tunnels associated with this gateway. VXLAN tunnels are not user configurable.
   d) Return to privileged EXEC mode.

```
device(config-overlay-gw-gateway1)# end
device(config)# end
```

Configuring MAC learning of Layer 2 extension site through BGP

The user can choose the way MAC learning is achieved for a Layer 2 extension tunnel.

BGP routing must be configured.

Data plane (Layer 2) MAC address learning is enabled by default at the remote site. However, this leads to scalability issues. Where BGP routing is used, do the following to enable MAC learning by means of BGP instead. This delegates the responsibility for MAC learning on a tunnel to the Layer 3 control-plane protocol, such as BGP EVPN.

1. Enter VXLAN overlay-gateway site configuration mode.

```
device# configure terminal
device(config)# overlay-gateway gateway1
device(config-overlay-gw-gateway1)# site mysite
device(config-overlay-gw-gateway1-site-mysite)#
```

2. Enter the **mac-learning protocol bgp** command.

```
device(config-overlay-gw-gateway1-site-mysite)# mac-learning protocol bgp
```

3. To disable BGP MAC learning and return to the default of Layer 2 learning, use the **no mac-learning protocol bgp** command.

```
device(config-overlay-gw-gateway1-site-mysite)# no mac-learning protocol bgp
```

Additional commands for VXLAN configuration

Additional commands that support VXLAN configuration are listed in the following table.

NOTE
For complete information on the those commands as well as other VXLAN overlay-gateway commands and commands related to the NSX Controller, refer to the *Network OS Command Reference.*

TABLE 40 Additional commands for VXLAN configuration

Command	Description
clear overlay-gateway	Clears counters for the specified gateway.
enable statistics	Enables statistics for tunnels.
sflow remote-endpoint	Applies an sFlow profile for a VXLAN overlay gateway and sets the remote endpoints for tunnel interfaces.
show nsx controller	Displays connection status of the NSX Controller. Includes an option to display the gateway certificate that is needed for NSX "transport node" configuration.
show overlay-gateway	Displays status and statistics for the VXLAN overlay-gateway instance.
show running-config overlay-gateway	Displays the running configuration of the overlay gateway configuration, including the connection type.
show tunnel	Displays tunnel statistics, including those for the NSX Service Node.

Troubleshooting VXLAN Layer 2 extension configurations

This section provides examples of a variety of **show** commands that can be used to troubleshoot VXLAN Layer 2 extension configurations. In general, do the following to confirm the proper operation of the configuration:

- Confirm that the overlay gateway is up, with a valid VTEP IP address on all RBridges.
- Confirm that the tunnel interface is a member of the VLAN.
- Confirm that the administrative and operational states of the tunnels are up on all RBridges.
- Confirm that there is one multicast-replicator enabled RBridge for every extension tunnel.
- Confirm that only one BUM forwarder is enabled for the next hop for every extension tunnel.

- Confirm that all sites are configured so that the tunnels form a full mesh, and that there are no VLAN extension inconsistencies across sites.
- Confirm the MAC addresses.

Confirming the status of the VXLAN Layer 2 extension overlay gateway

NOTE
In the following examples, items to look for and confirm are highlighted in **bold.**

Confirm the IP address, loopback ID, and administrative state, by using the **show overlay-gateway** command:

```
device(config-overlay-gw-gw121)# do show overlay-gateway

Overlay Gateway "gw121", ID 1, rbridge-ids 1
Type Layer2-Extension, Tunnel mode VXLAN
IP address 10.2.2.1 ( Loopback 100 ), Vrf default-vrf
Admin state up
Number of tunnels 0
Packet count: RX 0                TX 0
Byte count  : RX (NA)             TX 0
```

NOTE
Ensure that there are no "Unknown" entries.

Confirming that the tunnel interface is a member of the VLAN

Confirm the tunnel and VNIs, by using the **show vlan brief** command:

```
device# show vlan brief

Total Number of VLANs configured    : 14
Total Number of VLANs provisioned   : 14
Total Number of VLANs unprovisioned : 0
VLAN    Name     State                       Ports           Classification
(R)-RSPAN                                (u)-Untagged
(T)-TRANSPARENT                             (c)-Converged
                                         (t)-Tagged
======= ======== ========================== =============== ==================
1       default  ACTIVE                     Po 333(t)
                                            Po 444(t)
1002(F) VLAN1002 INACTIVE(no member port)
5000    VLAN5000 INACTIVE(member port down) Tu 61441(t)     vni 15000
6000    VLAN6000 INACTIVE(member port down) Tu 61441(t)     vni 16000
                                            Tu 61442(t)     vni 16000
6001    VLAN6001 INACTIVE(member port down) Tu 61441(t)     vni 16001
                                            Tu 61442(t)     vni 16001
6002    VLAN6002 INACTIVE(member port down) Tu 61441(t)     vni 16002
                                            Tu 61442(t)     vni 16002
```

Confirming the status of the VTEP and tunnel

Confirm the administrative and operational state, by using the **show tunnel brief rbridge-id** command for the relevant RBridges:

```
device# show tunnel brief rbridge-id 10

Tunnel 61441, mode VXLAN, rbridge-ids 10
Admin state UP, Oper state UP
Source IP 10.2.2.1, Vrf default-vrf
Destination IP 10.1.1.1


device# show tunnel brief rbridge-id 40

Tunnel 61441, mode VXLAN, rbridge-ids 40
Admin state UP, Oper state UP
Source IP 10.2.2.1, Vrf default-vrf
Destination IP 10.1.1.1
```

NOTE

Ensure that there are no discrepancies in the operational states and source IP addresses across RBridges.

Confirming source and destination IP and MAC addresses

Confirm the source and destination IP and MAC addresses, by using the **show tunnel** command:

```
device# show tunnel 61441

Tunnel 61441, mode VXLAN, rbridge-ids 10,40
Ifindex 2080374796, Admin state up, Oper state up
Overlay gateway "test", ID 1
Source IP 10.2.2.1 ( Loopback 100 ), Vrf default-vrf
Destination IP 10.1.1.1, Site VCS_2
Active next hops on rbridge 10:
    IP: 10.2.2.2, Vrf: default-vrf
    Egress L3 port: Ve 20, Outer SMAC: 0027.f86f.cc18
    Outer DMAC: 0011.bbbb.dddd
    Egress L2 Port: Po 10, Outer ctag: 20
    BUM forwarder: no

    IP: 10.3.3.3, Vrf: default-vrf
    Egress L3 port: Ve 30, Outer SMAC: 0027.f86f.2233
    Outer DMAC: 0011.bbbb.1111
    Egress L2 Port: Po 30, Outer ctag: 30
    BUM forwarder: yes

Active next hops on rbridge 40:
    IP: 10.1.1.1, Vrf: default-vrf
    Egress L3 port: Te 20/0/1, Outer SMAC: 0027.f86f.3344
    Outer DMAC: 0000.0000.dddd
    Egress L2 Port: Po 10, Outer ctag: 20
    BUM forwarder: no

Packet count: RX 0              TX 0
Byte count  : RX (NA)           TX 00
```

NOTE

You can also use the **rbridge-id** keyword. Watch out for discrepancies in the operational states and source and destination IP addresses across RBridges. The "Active next hops . . . " lines can vary across participating RBridges. Ensure that BUM forwarding is enabled for the next hop, which carries egress BUM traffic. There must be only one such next hop for the extension tunnel.

To view tunnel details for the site, you can use the **show tunnel site** command with the **brief** keyword:

```
device# show tunnel site VCS_2 brief

Tunnel 61441, mode VXLAN, rbridge-ids 10
Admin state UP, Oper state UP
Source IP 10.2.2.1, Vrf default-vrf
Destination IP 10.1.1.1
```

Viewing tunnel statistics of the VXLAN Layer 2 extension overlay gateway

Confirm the transmission of packets on tunnels, by using the **show tunnel statistics** command:

NOTE

Receive bytes are not available.

```
device# show tunnel statistics
Tnl ID   RX packets      TX packets      RX bytes        TX bytes
======== =============== =============== =============== ================
61441    123             456             (NA)            4560
61442    567             890             (NA)            8900
```

Confirming the MAC addresses for VLANs and tunnels

Confirm the MAC addresses and the state of the tunnels for the appropriate VLANs:

```
device# show mac-address-table

VlanId  Mac-address    Type    State  Ports

30      0000.0000.0001 Dynamic Active Tu 61441
30      0000.0000.0002 Dynamic Active Tu 61441
30      0000.0000.0003 Dynamic Active Tu 61441
30      0000.0000.0004 Dynamic Active Tu 61441
30      0000.0000.0005 Dynamic Active Tu 61442
30      0000.0000.0006 Dynamic Active Tu 61442
20      0027.f880.1890 System  Remote XX 40/X/X
20      0027.f880.328f Dynamic Active Po 1
```

Active Tu 61441

NOTE

Only an Active state confirms that the MAC address is getting forwarded locally unto the tunnel.

Viewing VXLAN overlay gateway statistics

Confirm transmit and receive packets for the VLAN, by using the **show overlay-gateway** command:

```
device# show overlay-gateway name test vlan statistics

VLAN ID   RX packets           TX packets
========= ==================== ====================
30        1234                 5678
```

Confirming the tunnel SPAN status

Confirm the session status and the source and destination addresses by using the **show monitor** command:

```
device# show monitor

Session               : 10
Description           : [None]
State                 : Enabled
Source Interface      : Tu 61442 (Down)
Destination Interface : Te 1/0/48 (Up)
Direction             : Both
```

Confirm all sFlow details by using the **show sflow all** command:

```
device# show sflow all

sFlow services are:                     enabled
Global default sampling rate:           32768 pkts
Global default counter polling interval: 20 secs
Rbridge-Id                          Collector server address                    Number of samples sent
-------------------------------------------------------------------------------------------------------
         2                                  1.1.1.1:6343                                          1212
sflow info for tunnel ifindex: 0x7c00f001
------------------------------
sFlow services are:          enabled
Samples received from hardware:      1212
Effective sample-rate:               512
```

Troubleshooting Virtual Fabrics

Use the following **show** commands to view the status of Virtual Fabric configurations. For details, refer to the *Extreme Network OS Command Reference*.

TABLE 41 Virtual Fabrics show commands

Command	Description
show vlan brief	Displays all classified VLANs that are configured, provisioned (active) or unprovisioned (inactive).
show port profile	Displays the AMPP port-profile configuration information.
show overlapping-vlan-resource usage	For VDX 6740 and VDX 6940 series only. Displays the utilization of the hardware table entries that support classified or transport VLAN classifications that use overlapping C-TAGs in a Virtual Fabric context.
show virtual-fabric status	Displays the status of the Virtual Fabric: VF-capable, VF-incapable, or VF-enabled.

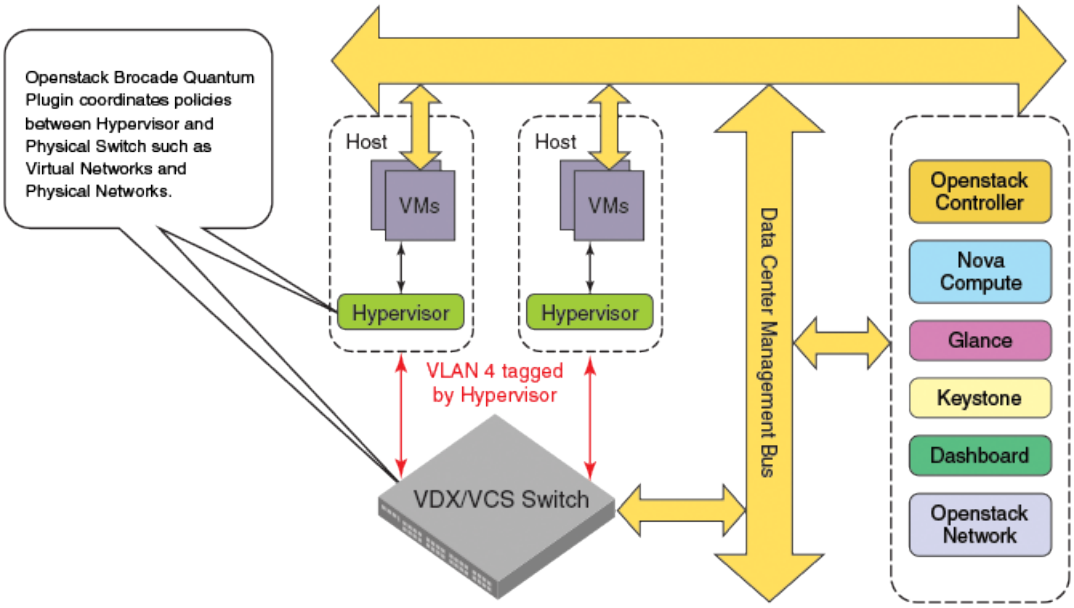
Extreme Support for OpenStack

Overview of OpenStack

OpenStack is an open-source Infrastructure as a Service (IaaS) initiative for creating and managing large groups of virtual private servers in a cloud computing environment. The Extreme Neutron ML2 Plugin for VDX/VCS provides a means to interface with OpenStack networking to orchestrate the Extreme physical switches.

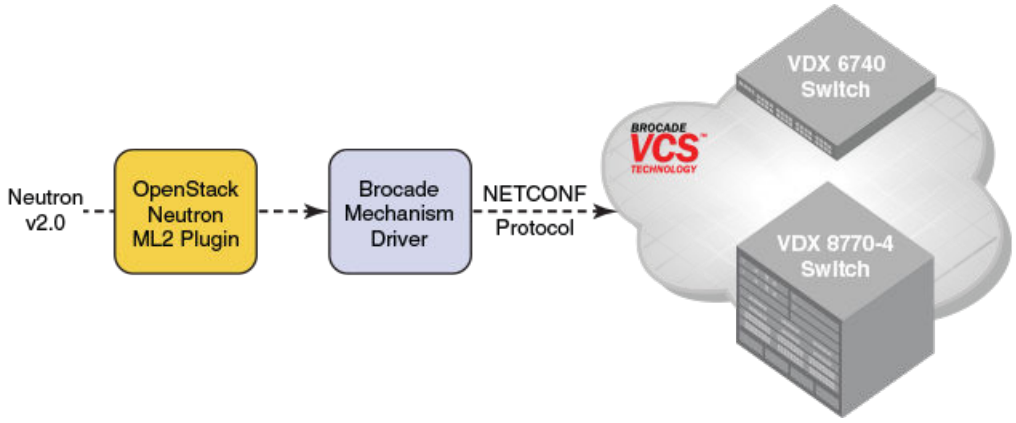
In cloud environments where virtual machines (VMs) are hosted by physical servers, the VMs see a new virtual access layer provided by the host machine. This new access layer can be created using many mechanisms, such as Linux bridges or virtual switches. The policies of the virtual access layer (virtual network) must be coordinated with the policies set in the hardware switches. The Extreme Neutron plugin helps coordinate this behavior automatically without any need for intervention by the administrator.

FIGURE 42 Virtual and physical network orchestration



The Extreme Neutron ML2 Plugin communicates with the Extreme Mechanism Driver, which uses NETCONF on the backend to configure the Extreme switches.

FIGURE 43 OpenStack configuration path



For the latest version of the OpenStack Neutron plugin, go to the https://wiki.openstack.org/wiki/Neutron site and search under plugins for the Extreme Networks plugin. The OpenStack community can be contacted at http://www.openstack.org/.

The plugins supported by the current version of Network OS are:

- OpenStack L2 Gateway plugin -- Network OS supports the OpenStack "Kilo" release. OpenStack L2 Gateway is a plugin that bridges Layer 2 traffic between the virtual world (VMs in the OpenStack Compute node) and the physical world (such as a Extreme VDX). OpenStack uses Open vSwitch Database (OVSDB) to distribute configurations between itself and the VDX, acting as an OVSDB client while the VDX acts as an OVSDB server.
- Modular Layer 2 plugin -- This plugin allows you to simultaneously utilize a variety of Layer 2 networking features, replacing the many plugins associated with those Layer 2 agents.
- Layer 3 SVI plugin -- This plugin provides ASIC level routing and gateway functionality in the switch for configured VLANs. This plugin enables line rate routing and gateway functionality.
- IP Fabric plugin -- In this plugin, leaf switches are connected only with the spine switches and are never connected to each other. Similarly, the spine switches are connected only with the leaf switches and are never connected to each other.
- VCS (Ethernet) Fabric plugin -- The two fabrics are connected through an IP network (such as an IP address on the underlay that is configured outside of the Openstack orchestration).

For complete information on configuring OpenStack and the supporting plugins, refer to the Extreme deployment guides for OpenStack.

Edge-Loop Detection

Edge-loop detection overview

Edge-loop detection (ELD) detects and disables Layer 2 loops that would cause broadcast storms. Typically, these loops are caused by misconfigurations.

ELD is configured and enabled on Extreme VCS Fabric clusters. Any topology that includes one or more Extreme VCS Fabric clusters use ELD to detect Layer 2 loops and prevent broadcast storms. Standalone switches can be included in such a cluster, but loop detection takes place on the Extreme VCS Fabric cluster, and not on the standalone switch. You cannot use ELD in a network consisting of standalone switches only.

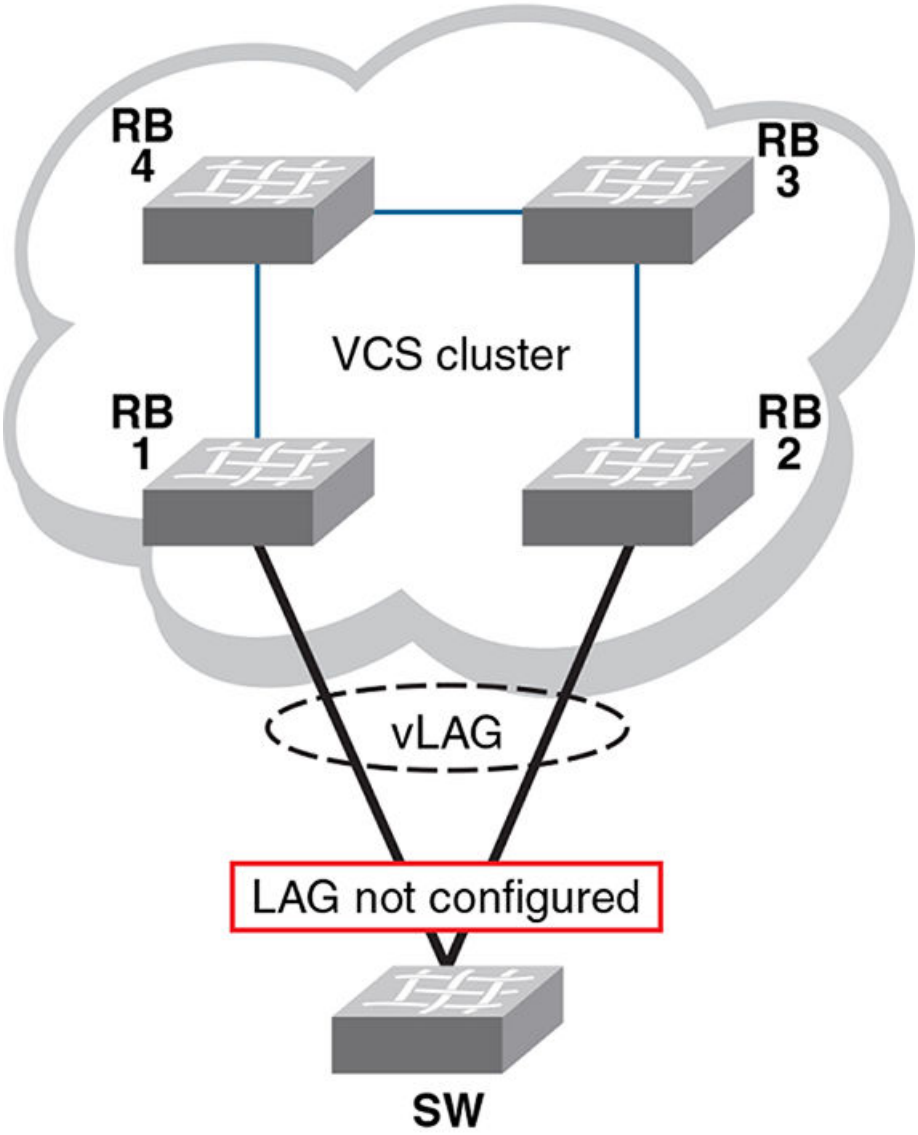
Specifically, ELD can be used to prevent broadcast storms caused by Layer 2 loops in the following topologies:

- A Extreme VCS Fabric cluster connects to a standalone switch.
- A Extreme VCS Fabric cluster connects to a multiple node network.
- A Extreme VCS Fabric cluster connects to other Extreme VCS Fabric clusters.

For an additional method of detecting and resolving L2 loops, refer to Resolving repeated MAC-moves on page 104.

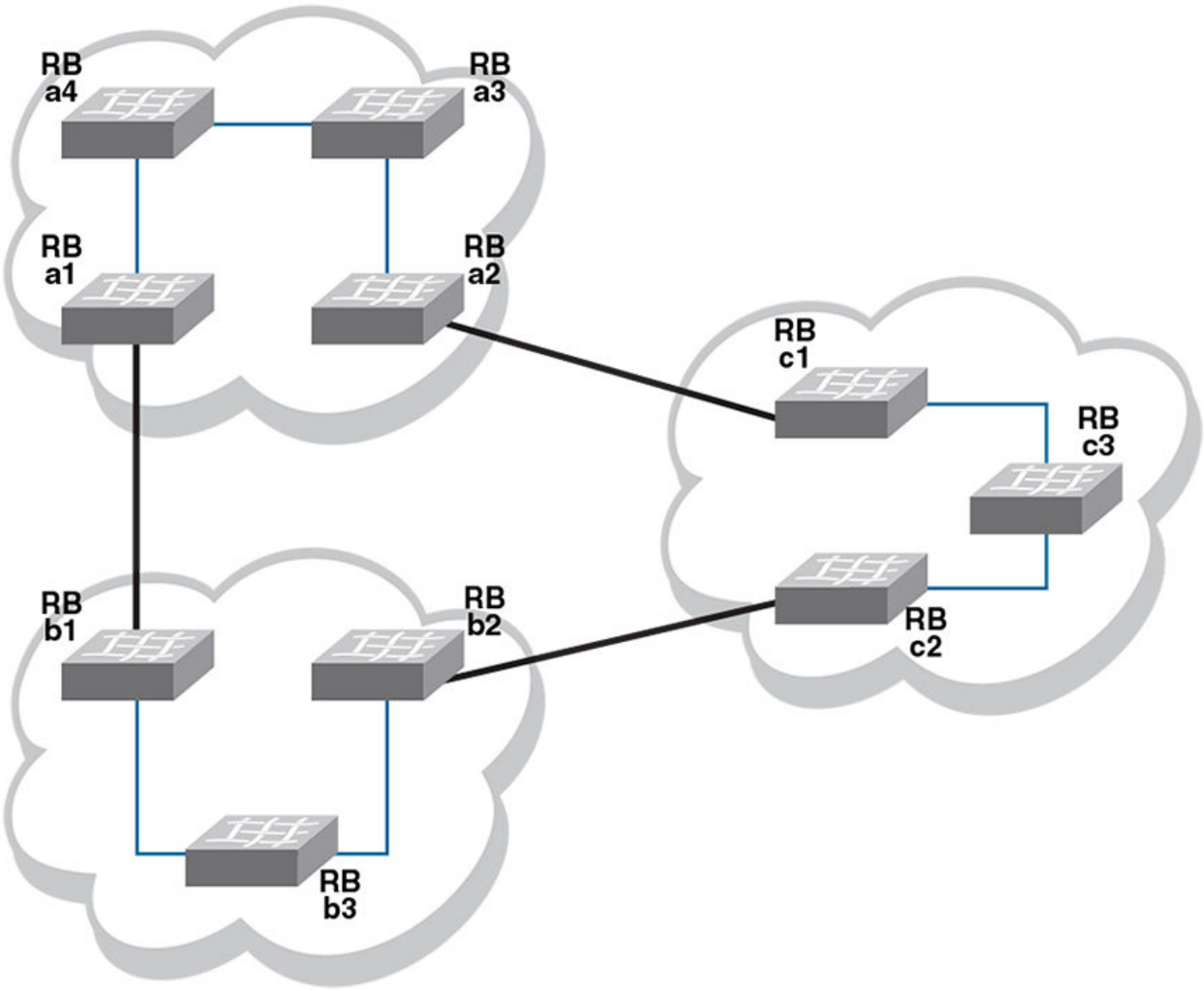
The following figure shows an example of a misconfiguration between a Extreme VCS Fabric cluster and a standalone switch that could cause a Layer 2 loop. In this case, a VLAG is configured on the edge devices of the Extreme VCS Fabric cluster for the two ISLs that connect the Extreme VCS Fabric cluster to the standalone switch. In this case, a LAG has not been created on the standalone switch at the other end of the ISLs. ELD detects and breaks this potential Layer 2 loop.

FIGURE 44 Missing LAG causes loop



The following figure shows another example for which ELD could be used to detect and break a Layer 2 loop. In this case, multiple Extreme VCS Fabric clusters are interconnected in a manner that creates a Layer 2 loop.

FIGURE 45 Interconnected Extreme VCS Fabric clusters cause loop



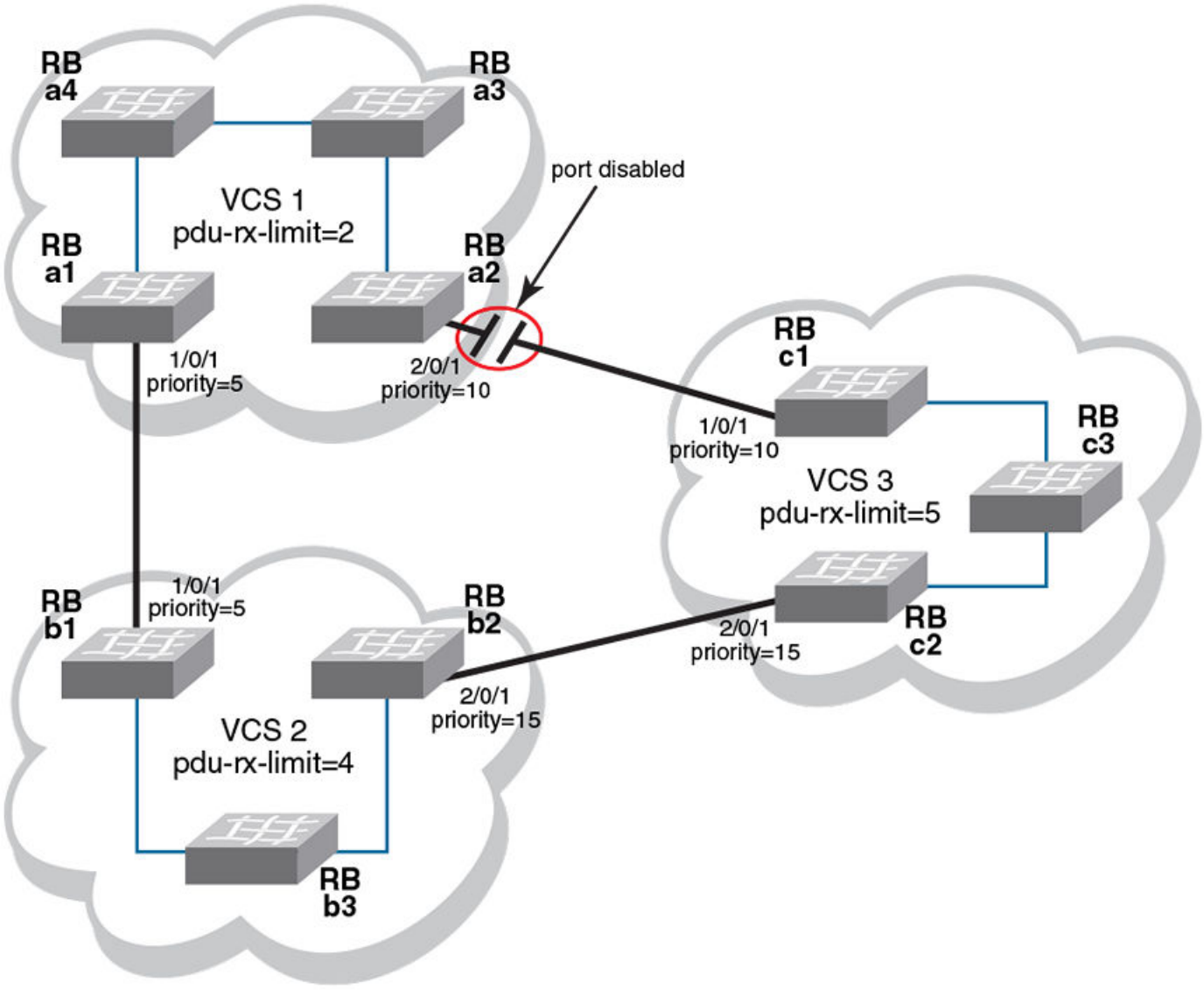
How ELD detects loops

ELD works by multicasting Protocol Data Unit (PDU) packets on edge ports. A device recognizes a loop when it receives a PDU that it initiated. Once the device recognizes that a Layer 2 loop exists, it can take action to disable a port and break the Layer 2 loop.

To minimize the number of disabled ports, ELD assigns a priority to each port and a unique receive limit (pdu-rx-limit) to each Extreme VCS Fabric cluster. The port priority determines whether the sending or receiving edge port of the cluster is disabled. The pdu-rx-limit determines on which Extreme VCS Fabric the action takes place. Without these configured values, it is possible that a Layer 2 loop could be detected in multiple clusters at the same time. As a result, multiple ports would be disabled, stopping traffic among the Extreme VCS Fabric clusters.

The following figure shows the same interconnections as the previous figure under Edge-loop detection overview on page 223, but with ELD enabled on each edge port, and with port priorities and receive limits assigned.

FIGURE 46 Interconnected Extreme VCS Fabric clusters with ELD enabled



With all ELD enabled edge ports sending PDUs at the same rate, VCS1 reaches its pdu-rx-limit first. Port 2/0/1 has a lower priority (higher priority number) than port 1/0/1, and is therefore selected to be disabled. If both ports have the same priority, the port with the higher port-ID is disabled.

If the port being shut down by ELD is part of a LAG, all member ports of the LAG are also shut down. If the port being shut down is part of a vLAG, all member ports of the vLAG on that RBridge are also shut down.

Once ELD disables a port, normal operation is for the port to remain disabled until any misconfiguration is repaired. Once the repair is finished, the port can be re-enabled manually.

NOTE

When ELD disables a port, the port is operationally down, but administratively still up. If a port is disabled by STP or some other Layer 2 protocol, ELD does not process PDUs for that port.

Configuring edge-loop detection

Edge-loop detection requires configuration at the global level and at the interface level. For global level configuration, you need to set the number of PDUs that the Extreme VCS Fabric cluster receives on any port before determining that a loop exists. This value is the **pdu-rx-limit** . You must also set the interval between sending PDUs by using the **hello-interval** command The combination of **pdu-rx-limit** and **hello-interval** determines the time it takes for ELD to detect and break a Layer 2 loop.

At the interface level, you must enable ELD on each port you want it to run on and set the port priority. You should also specify a VLAN on which ELD is enabled

Enter the **pdu-rx-limit** command to set the limit to a different number on each Extreme VCS Fabric cluster so that only one Extreme VCS Fabric cluster disables a port. Set this value in the increment of two to prevent race conditions which might disable ports on two Extreme VCS Fabric clusters that are incrementally only one apart.

Enter the **hello-interval** command to set the interval between PDUs. This interval must be set to the same value on all Extreme VCS Fabric clusters for which ELD is configured, otherwise the results of edge-loop detection become unpredictable.

Optionally, enter the **mac refresh** command to flush MAC addresses at a specified interval (in seconds) on either the entire cluster or on the partner port to remove any MAC inconsistencies in your system. If two interfaces are present in a Layer 2 loop, each interface learns the same set of MAC addresses. When ELD detects the Layer 2 loop, it puts the participating interface into an operationally down state. Consequently, MAC addresses learned on that interface get flushed. However, the same MAC addresses are present at the interface at the other end of the already detected loop, thereby creating this MAC inconsistency.

Optionally, enter the **shutdown-time** command to configure ports to be re-enabled after a specified period of time (range 10 minutes to 24 hours). A typical use for this feature is in environments in which reconfiguration is common, such as in a typical lab environment. Typical use is to allow the default value of zero, which does not allow ports to be re-enabled automatically.

> NOTE
> Any change to **shutdown-time** only takes effect for the ports that are disabled by ELD after the configuration change. Any ports that were already disabled by ELD before the **shutdown-time** change continues to follow the old **shutdown-time** value. These ports start to follow the new shutdown time after the currently running timer expires and ELD still detects the loop and shuts down the port again.

For each interface on which ELD runs, enter the **edge-loop detection** command to enable ELD. You must also enter the **edge-loop-detection port-priority** command to specify the ELD port priority.

Setting global ELD parameters for a Extreme VCS Fabric cluster

The values in this example configure the Extreme VCS Fabric cluster to detect and break loops on receipt of 5 PDUs. Because the PDU interval is set to 2000 ms (2 seconds), any loop breaks after 10 seconds. The selected port will remain disabled for 24 hours, after which it is automatically re-enabled.

Perform this procedure on every Extreme VCS Fabric cluster where you configure ELD.

1. Log in to any switch in a Extreme VCS Fabric cluster.
2. In global configuration mode, enter the **protocol edge-loop-detection** command to enter edge-loop detection configuration mode.

```
device(config)# protocol edge-loop-detection
```

3. Enter the **pdu-rx-limit** *number* command to set the number of PDUs that will be received before breaking the Layer 2 loop.

   The *number* variable must be a value in the range1 through 5. The default value is 1.

   ```
   device(config-eld)# pdu-rx-limit 5
   ```

4. Enter the **hello-interval** *number* command to set the interval between PDUs.

   The *number* variable has a unit of 1 millisecond (ms). It must be in the range from 100 ms to 5000 ms . The default value is 1000 ms.

   ```
   device(config-eld)# hello-interval 2000
   ```

5. Enter the **mac-refresh** *number* command to flush MAC addresses at a specified interval (in seconds) on either the entire cluster or on the partner port to remove any MAC inconsistencies in your system.

   The *number* variable must be in the range 60 through 300 (seconds). You must also specify either **all** or *port*, depending on whether you want to flush the entire cluster or only the partner port.

   ```
   device(config-eld)# mac refresh 100 all
   ```

6. Enter the **shutdown-time** *number* command to set the number of minutes after which the shutdown port is re-enabled.

   The *number* variable must be in the range 10 through 1440 (10 minutes through 24 hours). The default value is 0, indicating that the port is not automatically re-enabled.

   ```
   device(config-eld)# shutdown-time 1440
   ```

Setting interface parameters on a port

Perform this procedure for every port you want to be monitored by ELD.

1. Log in to any switch in a Extreme VCS Fabric cluster.
2. In global configuration mode, enter the **interface** command to select the *rbridge-id/slot/port* for which you want to enable edge-loop detection.
3. In interface configuration mode, enter the **edge-loop-detection vlan** command to specify the VLAN you want ELD to monitor on this port.

   If you do not specify a VLAN, the command fails.
4. Enter the **edge-loop-detection port-priority** command to specify the ELD port priority of the specified port for the selected VLAN. However, enabling switching is not mandatory for assigning a port-priority.

   NOTE
   The priority range of values is from 0 through 255. A port with priority 0 means that shutdown for this port is disabled. The default value port priority is 128

Setting the ELD port priority on two port/VLAN pairs

This example sets the ELD port priority on two port/VLAN pairs: port 1/0/7 VLAN 10 and port 4/0/6 VLAN 10.

If both these ports are detected in the same loop, ELD shuts down port 4/0/6 when the pdu-rx-limit for the Extreme VCS Fabric cluster is reached. Port 4/0/6 is chosen for shut down because it has been assigned the lower priority (higher number) then port 1/0/7.

```
device(config)# interface TenGigabitEthernet 1/0/7
device(conf-if-te-1/0/7)# edge-loop-detection vlan 10
device(conf-if-te-1/0/7)# edge-loop-detection port-priority 5
device(conf-if-te-1/0/7)# top
device(config)# interface TenGigabitEthernet 4/0/6
device(conf-if-te-1/0/7)# edge-loop-detection vlan 10
device(conf-if-te-1/0/7)# edge-loop-detection port-priority 7
```

Troubleshooting edge-loop detection

Use the edge-loop detection commands to view and correct misconfigurations

1. Log in to any switch in a Extreme VCS Fabric cluster.
2. In the global configuration mode, enter the **show edge-loop-detection** command to display edge-loop detection statistics for the Extreme VCS Fabric cluster.
   The command output shows ports disabled by ELD.
3. Correct any misconfigurations detected in step 2.
4. Perform one of the following operations in global configuration mode:
   - To re-enable a single port disabled by ELD:
     1. Enter the **shutdown** command on the port.
     2. Enter the **no shutdown** command on the port.

        NOTE
        If an edge-port becomes an ISL port because a remote port's VCS ID was changed, a port that was already shut down by ELD must be cycled with the **shutdown** and **no shutdown** commands to be detected as an ISL port.

   - To re-enable all ports disabled by ELD, enter **clear edge-loop-detection**.

Python Event-Management and Scripting

Python under Extreme operating systems

The Python interpreter installed with supported Extreme operating systems enables you to access a Python shell or to launch Python scripts. You can also define event handlers that run such scripts automatically upon specified conditions.

NOTE
Network OS is among the Extreme operating systems that support Python.

Python overview

Python is a high-level scripting language that also supports object-oriented programming. If you have previous programming experience, you can quickly learn how to write useful, simple Python scripts.

NOTE
For Python resources, refer to http://python.org.

Working interactively in the Python shell

Use this procedure to access a Python shell, within which you can use Python commands that call and manipulate Extreme operating system commands.

NOTE
The Python shell is accessible only to admin-role users.
Python syntax is case-sensitive.

1. In privileged EXEC mode, enter **python** to access the Python shell.

   ```
   device# python
   ```

   The `device#` prompt changes to a Python prompt:

   ```
   device# python
   Python 3.3.2 (default, Apr 11 2014, 13:05:18)
   [GCC 4.8.2] on linux
   Type "help", "copyright", "credits" or "license" for more information.
   ```

2. To exit the Python shell and return to the Extreme operating system prompt, enter either:
   - `exit()`
   - `Ctrl-D`

3. To run a Extreme operating system command from within the Python shell, enter the `CLI( )` command.

```
>>> cmd_show_running_vlan = CLI('show running-config interface vlan')
!Command: show running-config interface vlan
!Time: Tue Jan  6 00:12:37 2015
interface Vlan 1
>>>
```

   The statement entered above does two things:

   - Runs the **show running-config interface vlan** command and displays the result.
   - Assigns that command to a Python variable named `cmd_show_running_vlan`

4. To run a series of Extreme operating system commands from within the Python shell, separate the commands with `\n`.

```
>>> cmd_config_vlans = CLI('configure \n interface vlan 101-103 \n exit')
!Command: configure \n interface vlan 101-103 \n exit
!Time: Tue Jan  6 00:24:17 2015
>>>
```

   NOTE

   There is a difference between running a sequence of Extreme operating system CLI commands in the Python shell rather than in the standard Extreme operating system interface. Whereas in the standard interface the result of a command is persistent, in the Python shell each `CLI( )` statement is independent of any preceding ones.

In the following example, the lines beginning with **#** are added for explanation.

```
device# python
Python 3.3.2 (default, Apr 11 2014, 13:05:18)
[GCC 4.8.2] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> cmd_show_running_vlan = CLI('show running-config interface vlan')
# The Network OS show running-config int vlan command is run,
# and that command is assigned to the Python variable cmd_show_running_vlan.
!Command: show running-config interface vlan
!Time: Tue Jan  6 00:12:37 2015
interface Vlan 1
!
>>> cmd_config_vlans = CLI('configure \n interface vlan 101-103 \n exit')
# A series of three commands are run and assigned to the Python variable cmd_config_vlans.
!Command: configure \n interface vlan 101-103 \n exit
!Time: Tue Jan  6 00:24:17 2015

>>> cmd_show_running_vlan.rerun()
# The rerun() function appended to cmd_show_running_vlan gives the following output:
!Command: show running-config interface vlan
!Time: Tue Jan  6 00:24:34 2015
interface Vlan 1
!
interface Vlan 101
!
interface Vlan 102
!
interface Vlan 103
!
```

Python scripts

Python scripts enable you to manipulate and launch Extreme operating system commands, taking advantage of the power and flexibility of Python. Such scripts also support event handling.

The topics in this section guide you through the process of writing and testing Python scripts, copying them to supported devices, and running them with the **python** command from the command line.

NOTE

There are also Python scripts—installed along with the firmware—that you launch with the **execute-script** command, rather than with the **python** command. For details, refer to the following topics:

- The *Extreme Network OS Command Reference* **execute-script** topic
- The *Extreme Network OS Troubleshooting Guide* "Tracing the data path with a script" topic

Guidelines for writing Python scripts

The general guidelines for writing Python scripts to run under Network OS are as follows:

- Although previous experience programming in Python is helpful, experience in other high-level languages is enough to get you started with simple Python scripts.
- The Python developer either needs experience with Network OS CLI or access to a resource with such experience.
- To help decide which editor or integrated development environment (IDE) to use, refer to http://www.python.org.
- Make sure that the appropriate version of the Python interpreter is installed on your development computer. For Network OS 7.1.0, install Python 3.3.2.
- Make sure that in the *Extreme Network OS Command Reference* you are familiar with the **python** and the **CLI( )** topics.
- The script must include `from CLI import CLI`. This enables the **CLI( )** command, by which Python can interact with SLX-OS.
- Write the Python script and save it, with a `.py` suffix. Valid filenames range from 4 through 32 characters (including the suffix). The first character must be alphabetic.

Guidelines for RASLOG event-handler scripts

The additional guidelines for writing RASLOG event-handler scripts are as follows:

- The script must include `import json`.
- If an event-handler detects the log messages specified in the triggers and the trigger-function conditions are met, the script is executed. As part of this execution, the script is passed a --raslog-triggers argument with JSON-formatted dictionary containing the relevant message identifiers (MSGID) as keys and the message text as values.
- If the trigger-function is OR, only a single MSGID and message text are present.
- If the trigger function is AND, all relevant MSGIDs and messages are passed inside the json data structure.

NOTE

For sample scripts, refer to Python scripts and run-logs on page 235.

Testing Python-script statements

While developing a Python script, you can test Extreme operating system calls by entering them in the device Python command shell. After the script is stable, you copy it to the device and then test it further by running it from the command line.

1. In privileged EXEC mode, enter the **python** command to access the Python shell.

```
device# python
Python 3.3.2 (default, Apr 11 2014, 13:05:18)
[GCC 4.8.2] on linux
Type "help", "copyright", "credits" or "license" for more information.
```

Note that the `device#` prompt changed to a >>> Python prompt:

2. Enter the script statements one at a time, verifying that they run as expected.

```
>>> cmd_config_vlans = CLI('configure \n interface vlan 101-103 \n exit')
!Command: configure \n interface vlan 101-103 \n exit
!Time: Tue Jan  6 00:24:17 2015
>>>
```

3. Make corrections as needed.

Copying Python files to the device

After writing and testing a Python script file, use one of these topics to copy it to device flash memory.

Copying a file from a USB device

Use this topic to copy a file from a USB stick to an Extreme device.

ATTENTION
The only supported USB device for this task is the Extreme USB stick shipped with the device.

1. Copy the Python script file to the USB stick.
2. Insert the USB stick into the device USB port and enter the **usb on** command.
3. In privileged EXEC mode, enter the **copy** command to copy the Python file from the USB stick to the device flash memory.

```
device# copy usb://pythscript1.py flash://pythscript1.py
```

4. To display a list of files in the device flash memory, enter the **dir** command.

```
device# dir
total 24
drwxr-xr-x   2 root     sys          4096 Mar 26 15:22 .
drwxr-xr-x   3 root     root         4096 Jan  1  1970 ..
-rw-r--r--   1 root     root         1051 Mar 24 16:09 create_po.py
-rw-r--r--   1 root     root          207 Mar 24 16:09 create_vlans.py
-rw-r--r--   1 root     sys           557 Mar 26 10:37 defaultconfig.novcs
-rw-r--r--   1 root     sys           778 Mar 26 10:37 defaultconfig.vcs

1922789376 bytes total (828317696  bytes free)
```

5. To display the contents of a Python file copied into the device flash memory, enter the **show file** command.

```
device# show file pythscript1.py
```

Downloading a file from a network

Use this topic to download a file from a network location to the Extreme device.

1. Make sure that the Python script file is uploaded to an accessible network location.
2. In privileged EXEC mode, enter the **copy** command to copy the Python file from the network location to the device flash memory.

```
device# copy ftp://MyUserID:MyPassword@10.10.10.10//pythscript1.py flash://pythscript1.py
```

For other file-transfer options, refer to the *Extreme Network OS Command Reference* **copy** topic.

3. To display a list of files in the device flash memory, enter the **dir** command.

```
device# dir
total 24
drwxr-xr-x   2 root     sys          4096 Mar 26 15:22 .
drwxr-xr-x   3 root     root         4096 Jan  1  1970 ..
-rw-r--r--   1 root     root         1051 Mar 24 16:09 create_po.py
-rw-r--r--   1 root     root          207 Mar 24 16:09 create_vlans.py
-rw-r--r--   1 root     sys           557 Mar 26 10:37 defaultconfig.novcs
-rw-r--r--   1 root     sys           778 Mar 26 10:37 defaultconfig.vcs

1922789376 bytes total (828317696  bytes free)
```

4. To display the contents of a Python file copied into the device flash memory, enter the **show file** command.

```
device# show file pythscript1.py
```

Running Python scripts from the command line

The facility to run Python scripts from the Extreme operating system command line enables you to execute complex and repetitious tasks with accuracy and efficiency. This facility also enables you to validate scripts intended for event management.

CAUTION
Make sure that you test Python scripts according to standard quality assurance practices before deploying them.

In privileged EXEC mode, enter the **python** command, specifying the Python script file that you want to run.

```
device# python create_po.py
```

After the script runs, the Network OS prompt displays.

Python scripts and run-logs

This section contains sample Network OS Python scripts and run-logs.

NOTE
To access sample scripts and related resources, refer to https://github.com/extremenetworks/ExtremeScripting.

Script for assigning interfaces to port channels (create_po.py)

The `create_po.py` script is an example for automating common configuration tasks.

This script runs the relevant **show running-config** commands before and after the following actions:

- VLAN configuration
- Port-channel configuration
- Adding interfaces to port channels

NOTE
Lines beginning with # are annotations.

```
#Required in all scripts for NOS:
from CLI import CLI

rbridges = [3, 4]
interfaces = [40, 41, 42, 43]
port_channel = 10
```

```
vlan_range = "101-105"
slot_number = 0

# Runs show running-config int vlan before the configuration, and assigns this NOS
# command to a Python variable named cmd_show_running_vlans.
cmd_show_running_vlans = CLI("show running-config int vlan")

# Configures VLANs.  {} is a placeholder for the format (arg1, arg2, ... argN) variables:
cmd_configure_vlans = CLI("config \n int vlan {}".format(vlan_range))

# Reruns show running-config int vlan, following definition of new VLANs:
cmd_show_running_vlans.rerun()

# Configures port channel (vLAG):
cmd_show_running_port_channels = CLI("show running-config int po")
cmd_configure_port_channel = CLI("config \n int po {} \n switchport \n switchport mode trunk \n switchport
trunk allowed vlan add {} \n switchport trunk tag native-vlan ; no shut".format(port_channel, vlan_range))

# Reruns show running-config int po, following configuration changes:
cmd_show_running_port_channels.rerun()

# Add interfaces to port channel (vLAG)
for rbridge in rbridges:
    for interface in interfaces:
        cmd_configure_interface = CLI("config \n int te {}/{}/{} \n fabric isl enable \n fabric trunk
enable \n channel-group {} mode active type standard \n no shut".format(rbridge, slot_number, interface,
port_channel))
        cmd_show_running_int_tengig = CLI("show running-config int te {}/{}/{}".format (rbridge,
slot_number, interface))

# Runs show running-config:
cmd_show_running = CLI("show running-config")
```

Run-log (create_po.py)

A log upon running `create_po.py` was as follows:

```
device# python create_po.py
!Command: show running-config int vlan
!Time: Mon May 18 15:08:52 2015

interface Vlan 1
!
!Command: config
 int vlan 101-105
!Time: Mon May 18 15:08:52 2015

!Command: show running-config int vlan
!Time: Mon May 18 15:08:53 2015
interface Vlan 1
!
interface Vlan 101
!
interface Vlan 102
!
interface Vlan 103
!
interface Vlan 104
!
interface Vlan 105
!

!Command: show running-config int po
!Time: Mon May 18 15:08:54 2015

% No entries found.

!Command: config
 int po 10
 switchport
```

```
 switchport mode trunk
 switchport trunk allowed vlan add 101-105
 switchport trunk tag native-vlan ; no shut
!Time: Mon May 18 15:08:54 2015

!Command: show running-config int po
!Time: Mon May 18 15:08:56 2015

interface Port-channel 10
 vlag ignore-split
 switchport
 switchport mode trunk
 switchport trunk allowed vlan add 101-105
 switchport trunk tag native-vlan
 spanning-tree shutdown
 no shutdown
!
!Command: config
 int te 3/0/40
 fabric isl enable
 fabric trunk enable
 channel-group 10 mode active type standard
 no shut
!Time: Mon May 18 15:08:57 2015

!Command: show running-config int te 3/0/40
!Time: Mon May 18 15:08:59 2015

interface TenGigabitEthernet 3/0/40
 fabric isl enable
 fabric trunk enable
 channel-group 10 mode active type standard
 lacp timeout long
 no shutdown
!
!Command: config
 int te 3/0/41
 fabric isl enable
 fabric trunk enable
 channel-group 10 mode active type standard
 no shut
!Time: Mon May 18 15:09:00 2015

!Command: show running-config int te 3/0/41
!Time: Mon May 18 15:09:01 2015

interface TenGigabitEthernet 3/0/41
 fabric isl enable
 fabric trunk enable
 channel-group 10 mode active type standard
 lacp timeout long
 no shutdown
!
!Command: config
 int te 3/0/42
 fabric isl enable
 fabric trunk enable
 channel-group 10 mode active type standard
 no shut
!Time: Mon May 18 15:09:03 2015

!Command: show running-config int te 3/0/42
!Time: Mon May 18 15:09:05 2015

interface TenGigabitEthernet 3/0/42
 fabric isl enable
 fabric trunk enable
 channel-group 10 mode active type standard
 lacp timeout long
 no shutdown
!
!Command: config
```

```
 int te 3/0/43
 fabric isl enable
 fabric trunk enable
 channel-group 10 mode active type standard
 no shut
!Time: Mon May 18 15:09:06 2015

!Command: show running-config int te 3/0/43
!Time: Mon May 18 15:09:07 2015

interface TenGigabitEthernet 3/0/43
 fabric isl enable
 fabric trunk enable
 channel-group 10 mode active type standard
 lacp timeout long
 no shutdown
!
!Command: config
 int te 4/0/40
 fabric isl enable
 fabric trunk enable
 channel-group 10 mode active type standard
 no shut
!Time: Mon May 18 15:09:08 2015

!Command: show running-config int te 4/0/40
!Time: Mon May 18 15:09:10 2015

interface TenGigabitEthernet 4/0/40
 fabric isl enable
 fabric trunk enable
 channel-group 10 mode active type standard
 lacp timeout long
 no shutdown
!
!Command: config
 int te 4/0/41
 fabric isl enable
 fabric trunk enable
 channel-group 10 mode active type standard
 no shut
!Time: Mon May 18 15:09:11 2015

!Command: show running-config int te 4/0/41
!Time: Mon May 18 15:09:12 2015

interface TenGigabitEthernet 4/0/41
 fabric isl enable
 fabric trunk enable
 channel-group 10 mode active type standard
 lacp timeout long
 no shutdown
!
!Command: config
 int te 4/0/42
 fabric isl enable
 fabric trunk enable
 channel-group 10 mode active type standard
 no shut
!Time: Mon May 18 15:09:13 2015

!Command: show running-config int te 4/0/42
!Time: Mon May 18 15:09:15 2015

<output truncated>
```

Script illustrating the .get_output function (get_output.py)

The .get_output function returns—as a list—the output of the Network OS CLI commands assigned to a Python object.

This script displays a list of the following information:

- Number of RBridges
- IP addresses of the management interfaces
- The VCS status
- The fabric status
- Device hostnames

NOTE

Lines beginning with # are annotations.

```
#Required in all scripts for NOS:
from CLI import CLI

# Import the Python Regular Expressions (re) module:
import re

# Create Python objects:
rbridges = []
mgmt_ip_addresses = []
vcs_status = []
fabric_status = []
hostnames = []

# Runs show vcs, and assigns this NOS command to a Python object named cmd_show_vcs.
# The optional "False" keyword suppresses stdout output:
cmd_show_vcs = CLI("show vcs", False)

# As a list, assign the result of show vcs to a Python object named output:
output = cmd_show_vcs.get_output()

for line in output:
    found = re.search(r'^(\d+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)$', line, re.M)

    if found:
        rbridges.append(found.group(1))
        mgmt_ip_addresses.append(found.group(3))
        vcs_status.append(found.group(4))
        fabric_status.append(found.group(5))
        hostnames.append(found.group(6))

print('rbridges: ', rbridges)
print('mgmt_ip_addresses: ', mgmt_ip_addresses)
print('vcs_status: ', vcs_status)
print('fabric_status: ', fabric_status)
print('hostnames: ', hostnames)
```

Run-log (get_output.py)

A log upon running `get_output.py` was as follows:

```
device# python get_output.py
rbridges:  ['2']
mgmt_ip_addresses:  ['10.24.81.101']
vcs_status:  ['Online']
fabric_status:  ['Online']
hostnames:  ['sw0']
```

Script for upgrading and reconfiguring switches (dad.py)

The `dad.py` script configures an RBridge or all VCS RBridges using DHCP automatic deployment (DAD).

This example configures an RBridge or all of the RBridges in a VCS using DHCP automatic deployment (DAD). The **python dad.py** command calls this script. Lines beginning with `#` are annotations.

> **ATTENTION**
> This script is run automatically by the DAD process. Refer to the "Using DHCP Automatic Deployment" section of the *Network OS Management Configuration Guide*.

```
##
# Usage:
# dad.py -h | --help     help message
#        -g | --global   option for LC mode to set cluster global config
#        -r | --rbridgeid <#> option for LC mode and FC mode to set switch-specific config
#        -l | --local otp option for LC mode to set switch-specific config
#
import os
import sys, getopt

# Import Network OS Python CLI module
from CLI import CLI

# debug log
log=open("/tmp/dadscript.log", "a")

def main(argv):
   mode = -1 # 0 - global, 1 - local
   rbridgeid = 0
   localcmd = 0 # local command
   try:
      opts, args = getopt.getopt(argv,"ghr:l",["global", "help", "rbridgeid=", "local"])
   except getopt.GetoptError:
      log.write('dad2.py [-h|--help] [-g|--global] [-r|--rbridgeid <#>] [-l|--local]\n')
      sys.exit(2)

# Parsing options
   for opt, arg in opts:
      if opt in ("-g", "--global"):
         if len(opts) > 1:
            log.write("only one option is allowed\n")
            sys.exit(3)
         mode = 0
      elif opt in ("-r", "--rbridgeid"):
         mode = 1
         rbridgeid = int(arg)
      elif opt in ("-l", "--local"):
         if len(opts) == 1:
            log.write("-r is required\n")
            sys.exit(4)
         mode = 1
         localcmd = 1
      else:
         log.write('dad2.py [-h|--help] [-g|--global] [-r|--rbridgeid] [-l|--local]\n')
         sys.exit(1)
   log.write("mode = " + str(mode) + '\n')

# Execute command
   if mode == 0:
      log.write("apply global config\n")
      CLI("conf ; banner login DAD ; end")
   elif localcmd == 0:
      log.write("apply local config\n")
      log.write("rbridgeid = " + str(rbridgeid) + '\n')
      CLI("conf ; rbridge-id " + str(rbridgeid) " ;  switch-attributes host-name sw " + str(rbridgeid) +
" ; end")
   else: # localcmd == 1
      log.write("apply local command\n")
```

```
        log.write("rbridgeid = " + str(rbridgeid) + '\n')
        os.system("date > /tmp/localcmd.date")
if __name__ == "__main__":
   main(sys.argv[1:])
```

Run-log (dad.py)

The following are logs of running `dad.py` with various parameters.

> NOTE
> Because `dad.py` is run automatically by the DAD process, you need to enter **show dadstatus** in order to see the results.

The following command runs the `dad.py` script and sets the switch login banner to "DAD".

```
device# python dad.py -g
!Command: conf ; banner incoming DAD0 ; end
!Time: Thu Apr 16 00:06:35 2015

device#
```

The following command runs the `dad.py` script on RBridge 90, and sets the switch hostname to sw90.0:

```
device# python dad.py -r 90
!Command: conf ; rbridge-id 90 ; root en ; switch-attributes host-name sw90.0 ; end
!Time: Thu Apr 16 00:06:39 2015

device#
```

The following command runs the `dad.py` script on RBridge 90, and runs a Linux command to log the current time into `/tmp/localcmd.date`, with no monitor output:

```
device# python dad.py -r 90 -l

device#
```

Event-handler script

This is a typical script launched when a specific RASLOG message is generated.

> NOTE
> Although this script is crafted for Network OS, it illustrates the basic principles of a script triggered by a RASLOG message.

The `all_ports_down.py` script below was deployed—as a temporary workaround—to ensure that if a port-channel is shut down the members are also shut down.

> NOTE
> To access this script and related resources, refer to https://github.com/extremenetworks/ExtremeScripting.

```
#!/usr/local/python/3.3.2/bin/python3

import getopt
import json
import sys
import io
import re
import pdb
from CLI import CLI
int_down_raslog = 'NSM-1020'
int_up_raslog = 'NSM-1019'
shutdown_string = ''
match = None
raslog_triggers = {}
output = ''
```

```
# For argument processing:
options, remainder = getopt.gnu_getopt(sys.argv[1:], '', ['raslog-triggers='])

for opt, arg in options:
    if opt in ('--raslog-triggers'):
        print('--raslog-triggers: ', arg)
        output = io.StringIO(arg)
        raslog_triggers = json.load(output)

# Opening a file for logging:
f = open("poout.txt", "a")
f.write("testing:\n")

def po_members(po_num, shutdown_str):
# Determines the physical members of the port-channel and executes
# a shutdown/no shutdown command as requested for each physical interface:
    print("inside po_members", po_num, shutdown_str)
    f.write("inside po_members" + str(po_num) + str(shutdown_str))
    members = []
# Executing the show port-channel CLI and matching the output:
    poCLI = "show port-channel " + str(po_num) + " | begin Link:"
    output = CLI(poCLI, do_print=False).get_output()
    print("output =\n" + str(output))
    f.write("output = \n" + str(output))

# Iterating over each interface found in the output, and executing the
# appropriate configuration commands:
    for entry in output:
        print(entry)
        str1 = "".join(entry)
        str1 = str1.lstrip()
        print(str1.startswith("Link"))
        if str1.startswith("Link"):
            phy = str1.split("Link: ")[1].split("(0x")[0]
            members.append(phy)
            command = "config term\nint " + phy + "\n" + str(shutdown_str)
            output = CLI(command, do_print=False).get_output()
            print("cli output = " + command + "\n" + str(output))
            f.write("cli output = " + command + "\n" + str(output))
            str1 = ""
    return members

# The following section uses the passed information from the event-handler
# and calls po_members with the correct information:
print('raslog_triggers:\n', str(raslog_triggers))
f.write("raslog_triggers:\n" + str(raslog_triggers) + '\n')
if int_down_raslog in raslog_triggers:
    shutdown_string = 'shutdown'
    match = re.search(r'interface Port-channel (\d+)',
                      raslog_triggers[int_down_raslog], re.IGNORECASE)
elif int_up_raslog in raslog_triggers:
    shutdown_string = 'no shutdown'
    match = re.search(r'interface Port-channel (\d+)',
                      raslog_triggers[int_up_raslog], re.IGNORECASE)

if match:
    po = match.group(1)
    print('Performing operation "' + shutdown_string
          + '" on members for Port-channel ' + po + '\n')
    f.write('Performing operation "' + shutdown_string
            + '" on members for Port-channel ' + po + '\n')
    members = po_members(po, shutdown_string)
    print('\tMembers on Port-channel ' + po + ': ' + str(members) + '\n')
    f.write('\tMembers on Port-channel ' + po + ': ' + str(members) + '\n')

f.close()
```

Test-run (Event-handler script)

The following command tests the `all_ports_down.py` script for an NSM-1020 RASLOG:

```
device# python all_ports_down.py --raslog-triggers {"NSM-1020":"interface Port-channel 10 is
administratively down."}
```

The following command tests the `all_ports_down.py` script for an NSM-1019 RASLOG:

```
device# python all_ports_down.py --raslog-triggers {"NSM-1019":"interface Port-channel 10 is
administratively up."}
```

To verify that the script with "NSM-1019" works correctly, enter the following commands:

```
device# show port-channel detail
 LACP Aggregator: Po 10
 Aggregator type: Standard
  Actor System ID - 0x8000,78-a6-e1-45-95-14
  Admin Key: 0010 - Oper Key 0010
  Receive link count: 4 - Transmit link count: 4
  Individual: 0 - Ready: 1
  Partner System ID - 0x0001,00-24-38-8b-f1-00
  Partner Oper Key 0102
 Flag * indicates: Primary link in port-channel
 Number of Ports: 4
 Member ports:
   Link: Eth 0/3 (0xC006000) sync: 1
   Link: Eth 0/4 (0xC008000) sync: 1   *
   Link: Eth 0/5 (0xC00A000) sync: 1
   Link: Eth 0/6 (0xC00C000) sync: 1

device# flex-cli show interface brief
Port    Link      Port-State    Dupl    Speed       Tag    MAC             Name
po10    Up        N/A           N/A     40000 Mbit  Yes    78a6.e145.9562
po11    Up        N/A           N/A     30000 Mbit  Yes    78a6.e145.9563
lb1     Down      N/A           N/A     N/A         N/A    N/A             N/A
0/1     Disabled  None          None    None        No     78a6.e145.9519  Ifml
0/2     Disabled  None          None    None        No     78a6.e145.951a
0/3     Up        Forward       Full    10000 Mbit  No     78a6.e145.951b
0/4     Up        Forward       Full    10000 Mbit  No     78a6.e145.951c
0/5     Up        Forward       Full    10000 Mbit  No     78a6.e145.951d
0/6     Up        Forward       Full    10000 Mbit  No     78a6.e145.951e
0/7     Up        Forward       Full    10000 Mbit  No     78a6.e145.951f
```

Python event-management

Python event management enables you to specify a Python script that runs automatically upon specified conditions.

The conditions available for triggering a Python script are as follows:

- A specified RASlog message. You can use POSIX extended regular expressions further to limit the trigger.
- One of the following switch events:
  - Switch bootup
  - Completion of cluster formation and completion of configuration-file replay

Configuring an event-handler profile

Use this procedure to create an event-handler profile, define one or more triggers for it, and specify a Python script that runs upon one or more trigger events.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter the **event-handler** command.

```
device(config)# event-handler eventHandler1
```

3. For each trigger that you need for a profile, enter the **trigger** command.

```
device(config-event-handler-eventHandler1)# trigger 1 raslog LOG-1001
```

   The trigger event is RASlog message #LOG-1001.

4. Enter the **action python-script** command to specify a Python script that runs when the event-handler is triggered.

```
device(config-event-handler-eventHandler1)# action python-script example.py
```

5. To add an event-handler-profile description, enter **description**, followed by the description text.

```
device(config-event-handler-eventHandler1)# description This is a sample description.
```

CAUTION
Make sure that you test Python scripts according to standard quality assurance practices before deploying them.

The following example includes all of the steps.

```
device# configure terminal
device(config)# event-handler eventHandler1
device(config-event-handler-eventHandler1)# trigger 1 raslog LOG-1001
device(config-event-handler-eventHandler1)# action python-script example.py
device(config-event-handler-eventHandler1)# description This is a sample description.
```

The following example defines a trigger that uses POSIX extended REGEX to search for a match within a specified RASlog message ID.

```
device# configure terminal
device(config-event-handler-eventHandler1)# event-handler eventHandler2
device(config-event-handler-eventHandler2)# trigger 1 raslog NSM-1003 pattern Interface (One|Ten|Forty|
Hundred)GigabitEthernet 89/0/[1-9] is link down
```

The specified RASlog message NSM-1003 includes "**interface** *interface-name* is link down", indicating that an interface is offline because the link is down. The REGEX searches within such a message for an interface from 89/0/1 through 89/0/9.

Activating an event-handler on an RBridge

Use this procedure to activate one or more event-handlers on an RBridge. If a trigger specified in the event-handler profile occurs, a designated Python script runs.

1. Enter **configure terminal** to access global configuration mode.

```
device# configure terminal
```

2. Enter the **rbridge-id** command to access RBridge configuration mode.

```
device(config)# rbridge-id 1
```

3. Enter the **event-handler activate** command, specifying the event handler that you are activating on the RBridge.

```
device(config-rbridge-id-1)# event-handler activate eventHandler1
```

4. To activate an additional event handler on the RBridge, enter the **event-handler activate** command, specifying the additional event handler.

```
device(config-activate-eventHandler1)# event-handler activate eventHandler2
```

5. To de-activate an event handler on that Rbridge, enter the **no event-handler activate** command

```
device(config-activate-eventHandler2)# no event-handler activate eventHandler2
```

The following example includes all of the activation steps.

```
device# configure terminal
device(config)# rbridge-id 1
device(config-rbridge-id-1)# event-handler activate eventHandler1
device(config-activate-eventHandler1)# event-handler activate eventHandler2
```

Configuring event-handler options

After you activate an event handler on a RBridge, you can configure various options. For example, you can specify if the event-handler script runs more than once and how multiple triggers are handled.

1. Activate an event handler on an RBridge, as described in Configuring an event-handler profile on page 244:

```
device# configure terminal
device(config)# rbridge-id 1
device(config-rbridge-id-1)# event-handler activate eventHandler1
```

2. To specify a delay from when a trigger is received until execution of the event-handler action, enter the **delay** command.

```
device(config-activate-eventHandler1)# delay 60
```

   The example specifies a delay of 60 seconds.

3. To specify multiple iterations of the action when a trigger is received:
   a) Enter the **iterations** command.
   b) To specify an interval between iterations, enter the **interval** command.

```
device(config-event-handler-eventHandler1)# iterations 3
device(config-activate-eventHandler1)# interval 30
```

   The example sets the number of iterations to 3 and specifies an interval of 30 seconds between each iteration.

4. To specify that a triggered action not be interrupted by cluster formation, enter the **run-mode exclusive** command.

```
device(config-activate-eventHandler1)# run-mode exclusive
```

5. To specify a maximum number of minutes to wait for an action script to complete execution, enter the **action-timeout** command.

```
device(config-activate-eventHandler1)# action-timeout 30
```

   The example sets the timeout to 30 minutes.

6. To limit action-recurrence upon multiple trigger-events, enter one of the following commands:
   - **trigger-mode only-once**—for the duration of a switch's configuration, the event-handler action is launched only once.

     ```
     device(config-activate-eventHandler1)# trigger-mode only-once
     ```

   - **trigger-mode on-first-instance**—as long as the switch is running, the event-handler action is launched only once. Following a switch restart, the event-handler action can be triggered again.

     ```
     device(config-activate-eventHandler1)# trigger-mode on-first-instance
     ```

7. If multiple triggers are defined, to specify that the action run only if all of the triggers occur, enter the **trigger-function AND time-window** command.

   ```
   device(config-activate-eventHandler1)# trigger-function AND time-window 120
   ```

   The example specifies that the action run only if all triggers occur within 120 seconds.

Troubleshooting event-management

Use these topics to troubleshoot issues that arise during implementation of Python event-management.

Aborting an event-handler action

If needed, abort a Python script launched by an event-handler action.

1. In privileged EXEC mode, enter the **event-handler abort action** command.

   ```
   device# event-handler abort action eh1
   This operation will abort an event handler action that is currently running and may leave the device
   in an inconsistent state.  Do you want to continue? [y/n]:y
   ```

2. To confirm aborting the action, type `y`.

   ```
   Operation completed successfully.
   ```

   The Python script launched by the event-handler action is aborted.

Event-management show commands

There are several **show** commands that display event-management information. They are documented in the *Network OS Command Reference*, and listed here with descriptions.

TABLE 42 Event-management show commands in the *Network OS Command Reference*

Command	Description
show event-handler activations	Displays operational data of activated event-handlers.
show running-config event-handler	Displays details of event-handler profiles defined on the device. You can display the results by description, Python-script action, or trigger ID.
show running-config rbridge-id event-handler	Displays one or all of the event-handlers in the running configuration. You can limit the display by RBridge ID, event-handler activation, and additional parameters.