

IBM Distributed Virtual Switch 5000V Quickstart Guide

Managed Layer 2 distributed virtual switch for VMware

Advanced networking and management features

Implementation and troubleshooting



David Gain
Per Ljungström
Jason G. Neatherway
Sangam Racherla
Lutfi Rachman

Redbooks



International Technical Support Organization

IBM Distributed Virtual Switch 5000V Quickstart Guide

June 2014

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (June 2014)

This edition applies to Version 1, Release 1, Modification 1 of the IBM Distributed Virtual Switch 5000V (DVS 5000V).

© Copyright International Business Machines Corporation 2014. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	v
Trademarks	vi
Preface	vii
Authors	vii
Now you can become a published author, too!	ix
Comments welcome	ix
Stay connected to IBM Redbooks	ix
Chapter 1. Introducing the IBM Distributed Virtual Switch 5000V	1
1.1 Introduction	2
1.2 System requirements	2
1.3 DVS 5000V solution overview	3
1.3.1 DVS 5000V components	4
1.3.2 Edge Virtual Bridging	4
1.4 VMware networking	5
1.4.1 Virtual switches	6
1.4.2 Port Groups and vNIC profiles	7
1.5 DVS 5000V enhancements since Version 1.0	8
1.5.1 Enhancements	8
1.5.2 Portless profiles feature	9
1.5.3 Recovery feature	10
1.5.4 Upgrade procedure for the enhancements	10
1.6 More information	10
Chapter 2. IBM Distributed Virtual Switch 5000V reference architecture	11
2.1 Architecture overview	12
2.2 Physical architecture	12
2.3 Logical architecture	14
2.3.1 Layer 2	14
2.3.2 Layer 3	15
Chapter 3. IBM Distributed Virtual Switch 5000V installation	17
3.1 DVS 5000V installation prerequisites	18
3.2 Installing the DVS 5000V Host module	18
3.2.1 Installation using the ESXi shell	18
3.3 Installing the DVS 5000V Controller	19
3.4 DVS 5000V initial configuration	25
3.4.1 Setting the IPv4 management address	25
3.4.2 Enabling remote access	26
3.5 Creating the Global vDS instance	27
3.6 DVS 5000V licensing	28
3.7 Adding the ESXi host to the DVS 5000V	28
Chapter 4. Systems management	31
4.1 Authentication, Authorization, and Accounting	32
4.1.1 Local user accounts	32
4.1.2 RADIUS authentication and authorization	33
4.1.3 TACACS+ authentication and authorization	34

4.2	Network management protocols	34
4.2.1	Simple Network Management Protocol	35
4.2.2	System time	38
4.2.3	Logging and syslog	39
4.3	sFlow	39
4.3.1	Enabling sFlow traffic on ESXi	40
4.3.2	Configuring sFlow	41
4.4	Port mirroring	41
4.4.1	Configuring port mirroring	42
4.4.2	Configuring ERSPAN	42
4.4.3	Viewing ERSPAN in Wireshark	43
Chapter 5. IBM Distributed Virtual Switch 5000V implementation		47
5.1	Uplink configuration	48
5.1.1	Uplink profiles	48
5.1.2	Link aggregation	48
5.1.3	Link aggregation hash	52
5.2	Connecting virtual machines	53
5.2.1	Configuring vNIC profiles	53
5.2.2	Configuring stand-alone ports and port-level configuration	57
5.2.3	Associating VMs with the DVS 5000V	58
5.3	Virtualization Aware Networking with Edge Virtual Bridging	62
5.3.1	Prerequisites and limitations	63
5.3.2	Implementation overview	63
5.3.3	Configuring the VSI Type database	63
5.3.4	Applying a VSI Type with Virtual Ethernet Port Aggregator mode connectivity	67
5.3.5	Configuring the physical network for EVB	68
5.3.6	Connecting a virtual machine and verifying the EVB VEPA implementation	70
5.3.7	VM Mobility with VEPA	71
5.3.8	Virtual Edge Bridging with VSI Types	71
5.4	Advanced switch features	72
5.4.1	Quality of service (QoS)	72
5.4.2	Access Control Lists	73
5.4.3	Private VLANs	77
Chapter 6. Maintenance and troubleshooting		79
6.1	Configuration management	80
6.1.1	Configuration file and block	80
6.1.2	Managing the configuration	80
6.2	Firmware management	81
6.2.1	Loading a new image in to the DVS 5000V Controller	82
6.2.2	Selecting an image and configuration	82
6.3	Logging and reporting	83
6.3.1	Information that is needed for a Problem Management Record	83
6.4	Contacting IBM Support	83
Related publications		85
Online resources		85
Help from IBM		85

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.


COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®	IBM Flex System®	Redbooks (logo)  ®
BladeCenter®	IBM®	System p®
Extreme Blue®	PureFlex®	System x®
Global Technology Services®	RackSwitch™	
IBM Flex System Manager™	Redbooks®	

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

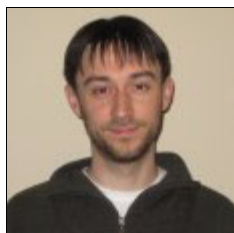
Preface

The IBM® Distributed Virtual Switch 5000V (DVS 5000V) is a software-based network switching solution that is designed for use with the virtualized network resources in a VMware enhanced data center. It works with VMware vSphere and ESXi 5.0 and beyond to provide an IBM Networking OS management plane and advanced Layer 2 features in the control and data planes. It provides a large-scale, secure, and dynamic integrated virtual and physical environment for efficient virtual machine (VM) networking that is aware of server virtualization events, such as VMotion and Distributed Resource Scheduler (DRS). The DVS 5000V interoperates with any 802.1Qbg compliant physical switch to enable switching of local VM traffic in the hypervisor or in the upstream physical switch. Network administrators who are familiar with IBM System Networking switches can manage the DVS 5000V just like IBM physical switches by using advanced networking, troubleshooting, and management features to make the virtual switch more visible and easier to manage.

This IBM Redbooks® publication helps network and system administrators install, tailor, and quickly configure the IBM Distributed Virtual Switch 5000V (DVS 5000V) for a new or existing virtualization computing environment. It provides several practical applications of the numerous features of the DVS 5000V, including a step-by-step guide to deploying, configuring, maintaining, and troubleshooting the device. Administrators who are already familiar with the CLI interface of IBM System Networking switches will be comfortable with the DVS 5000V. Regardless of whether the reader has previous experience with IBM System Networking, this publication is designed to help you get the DVS 5000V functional quickly, and provide a conceptual explanation of how the DVS 5000V works in tandem with VMware.

Authors

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.



David Cain is a Network and Systems Engineer for the IBM Software Group in Research Triangle Park, North Carolina. He has over 10 years of experience in the data center, with expertise in Ethernet switching, storage, SAN, security, virtualization, IBM System x®, and Linux system infrastructure. Dave holds a Bachelor of Science degree in Computer Science from North Carolina State University, and has co-authored two patents and invention disclosures in the networking field. He joined IBM full-time in the year 2006 after gaining valuable experience on various internships with IBM while a student, including an IBM Extreme Blue® internship in the year 2005.



Per Ljungström is a Network Systems Engineer for IBM in the Nordic countries, focusing on presales, architect designs, education, and network proof-of-concepts. He has 27 years of experience in Ethernet, WAN, and fiber communications, networking, VPN, firewall security, and data center design. He is certified and has experience with products such as Checkpoint, Cisco, BlueCoat, and F5. Per holds a Bachelor of Electronic Engineering degree from the Technical University of Denmark, and a second Bachelor of Information Technology degree from Niels Brock Copenhagen Business College. He is one of the inventors of the IBM BlueExpert community in IBM Europe, and has several inventions in the telephone and computer science field. He has also served for ten years as censor for graduating Bachelor of Information Technology students at different colleges and military schools in Denmark.



Jason G. Neatherway is an Infrastructure Architect in IBM Global Technology Services®, Strategic Outsourcing Delivery in Australia. He has 13 years experience in communications and networking. Jason has been with IBM for 10 years working in various roles in network delivery and architecture. He works as a solution architect providing network and security solutions and consulting services for IBM customers. Jason has a Bachelor of Engineering degree in Communications and a Post Graduate Certificate in E-commerce Management.



Sangam Racherla is an IT Specialist. He holds a degree in Electronics and Communication Engineering and has 11 years of experience in the IT field. His areas of expertise include Microsoft Windows, Linux, IBM AIX®, IBM System x, and IBM System p® servers, and various SAN and storage products.



Lutfi Rachman is an IT Specialist working for IBM Global Technology Services in Indonesia. He joined IBM in 2009 as one of the Apprenticeship program participants. He is responsible for service delivery in various networking solutions, managed services projects, and x86 virtualization across various industries. He is a VMware Certified Professional, and is familiar with Cisco, Juniper, Procurve, and IBM System Networking devices. He holds a Bachelor of Computer Science degree from Gadjah Mada University.

Thanks to the following people for their contributions to this project:

Megan Gilge, Ann Lund, Jon Tate
International Technical Support Organization, San Jose Center

Venkatas Amulothu, Stephan Benny, Jay Kidambi, Muhammad Khan, Pushkar Patil, Tim Shaughnessy
IBM

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at: ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:

ibm.com/redbooks

- ▶ Send your comments in an email to:

redbooks@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- ▶ Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- ▶ Look for us on LinkedIn:

<http://www.linkedin.com/groups?home=&gid=2130806>

- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>

- ▶ Stay current on recent Redbooks publications with RSS Feeds:

<http://www.redbooks.ibm.com/rss.html>



Introducing the IBM Distributed Virtual Switch 5000V

The IBM Distributed Virtual Switch 5000V (DVS 5000V) is a software-based network switching solution that is designed for use with the virtualized network resources in a VMware enhanced data center.

This chapter covers the following topics:

- ▶ Introduction
- ▶ Solution overview
- ▶ VMware networking

1.1 Introduction

The DVS 5000V is an advanced, feature rich distributed virtual switch for VMware environments with policy-based virtual machine (VM) connectivity. The DVS 5000V enables network administrators familiar with IBM System Networking switches to manage the DVS 5000V just like IBM physical switches by using advanced networking, troubleshooting, and management features so that the virtual switch is no longer hidden and difficult to manage.

The DVS 5000V includes the following advanced Layer 2 features:

- ▶ VLANs
- ▶ Private VLANs
- ▶ Port mirroring
- ▶ ERSPAN
- ▶ sFlow
- ▶ ACLs
- ▶ QoS
- ▶ LACP and Advanced Teaming
- ▶ SNMP
- ▶ RADIUS
- ▶ TACACS+
- ▶ Syslog
- ▶ Edge Virtual Bridging (EVB) (802.1QBG: VEPA, VDP, and VSI Manager)

The DVS 5000V enables a large-scale, secure, and dynamic integrated virtual and physical environment for efficient VM networking that is aware of server virtualization events, such as VMotion and Distributed Resource Scheduler (DRS). The DVS 5000V Data Path Module (DPM) is a Layer 2 virtual switch that is embedded in each ESXi hypervisor that provides each VM with a dedicated virtual switch port. The DVS 5000V Controller manages these virtual switches across multiple ESXi hypervisors, unifying them under a single, consolidated interface. Configuration is performed through the DVS 5000V Controller, and is automatically propagated so administrators can define configurations for all virtual switches from a single interface to enable simplified management of VM traffic. Private VLANs enable VM traffic separation, ACLs provide VM traffic control, and local port mirroring (SPAN) and remote port mirroring (ERSPAN) enable advanced VM traffic visibility and troubleshooting.

1.2 System requirements

The DVS 5000V solution requires the following items to function in a VMware datacenter:

- ▶ VMware vCenter: This is a VMware product that is on a server within the data center. It provides a centralized tool for installing, managing, and synchronizing vDS instances, hypervisors, and VMs on host servers throughout the data center.
- ▶ VMware vSphere Client: This is a VMware product that is on administrative client devices. It provides the server administrator or network administrator with rich, remote access to vCenter management tools.
- ▶ VMware ESXi 5.0: This is a VMware hypervisor product that is on individual host servers within the data center. It provides the software infrastructure for installing, running, and managing VMs and vDSs on the hosts. A valid vSphere Enterprise Plus license is required for operation.

1.3 DVS 5000V solution overview

The DVS 5000V works with VMware vSphere 5.0 and beyond and interoperates with any 802.1Qbg compliant physical switch to enable switching of local VM traffic in the hypervisor or in the upstream physical switch.

The DVS 5000V components and their relationship to VMware is summarized in Figure 1-1.

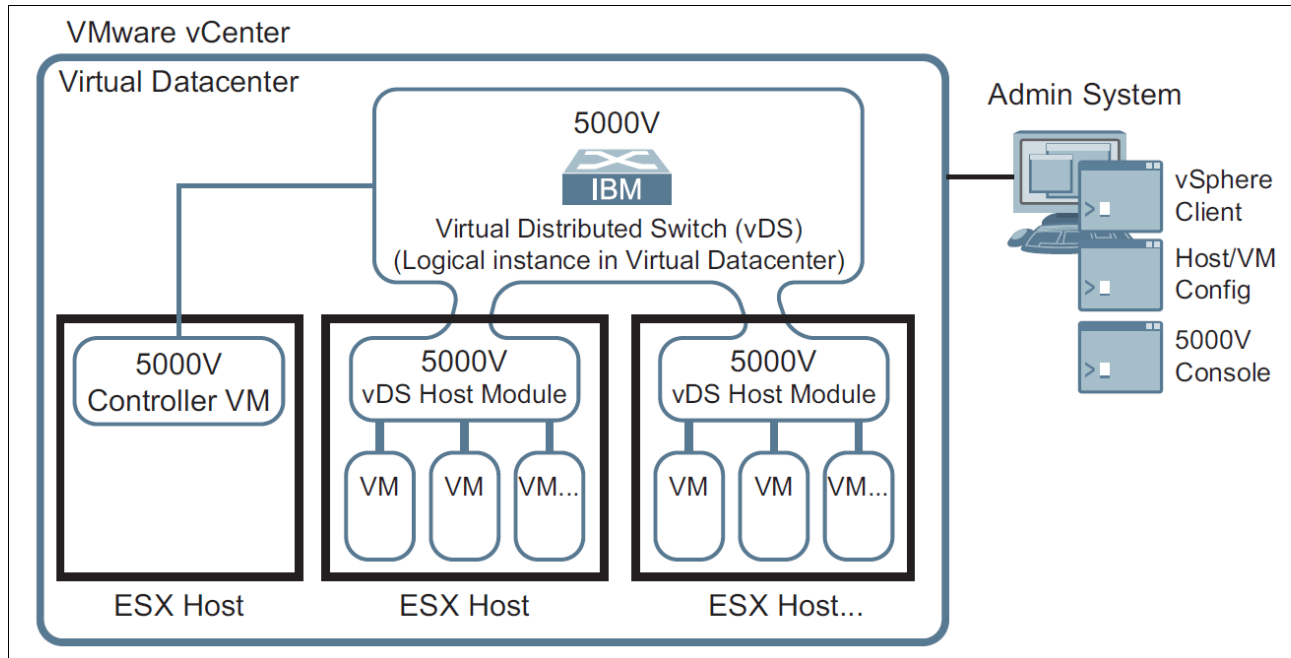


Figure 1-1 DVS 5000V in a VMware vCenter

Using the VMware vSphere Distributed Switch (vDS) model, the DVS 5000V software switch modules are “distributed” to each participating VMware ESXi host. Although each DVS 5000V host module handles traffic for the local VMs, all distributed modules also work in unison as an aggregate virtual switching device. The DVS 5000V solution can be roughly equated to an interconnected stack of independent switches, which are unified and controlled by a single management plane.

The DVS 5000V works with VMware vSphere and ESXi 5.0 and beyond to provide an IBM Networking OS management plane, and advanced Layer 2 features in the control and data planes:

- ▶ The management plane that is embedded in the DVS 5000V Controller includes an Industry-Standard Command-Line Interface (ISCLI) that runs on a VMware VM, and is packaged as an Open Virtual Appliance (OVA) file.
- ▶ The control/data plane is implemented by a software module that runs inside each participating ESXi hypervisor, which essentially is a low-level kernel module. It is packaged as a vSphere Installation Bundle (VIB) file.

Using this VMware vDS model, the network administrator can define the DVS 5000V at the data center level within the VMware vCenter. When ESXi hosts in the data center join the DVS 5000V, a virtual switch instance, or portset, is created on the host. Portsets inherit their properties from the global virtual switch. VMware vDS infrastructure synchronizes all the portsets and manages state migration during VMotion, the movement of (VMs within and among ESXi hypervisors).

Note: The term “vDS” that is used here refers to the infrastructure that is provided by VMware to achieve a distributed virtual switch implementation. The VMware vDS product has a different scope from the DVS 5000V and is expected to be managed by the *server administrator*. The DVS 5000V is managed by the *network administrator*. This way, network administrators can have control and visibility into virtualized network resources.

1.3.1 DVS 5000V components

This section introduces the DVS 5000V components.

DVS 5000V vDS Host Module

This is an IBM product that is in participating ESXi hypervisors on host servers within the data center. It implements a vDS portset as defined in the VMware vDS API and acts a virtual network switch for the given host server. At its core, it forwards frames based on destination MAC addresses, controlling Layer 2 access to and from the associated VMs. It also provides advanced switching features, such as VLANs and IGMP snooping. The settings for each feature are configured by the network administrator through the DVS 5000V Controller.

DVS 5000V Controller

This is an IBM product that is in a VM. It works with the VMware vCenter and ESXi hypervisors to unify all DVS 5000V host modules into a single management controller interface. Through the VMware vSphere client or Telnet/SSH, it provides a full ISCLI for switch configuration, operation, and the collection of switch information and statistics. All traffic to and from the controller is consolidated into a single virtual NIC.

This traffic includes the following items:

- ▶ Management traffic for applications such as Telnet, SSH, and SNMP
- ▶ vSphere API traffic between the vSphere Client and the VMware vCenter
- ▶ Traffic between controller and the virtual switch elements on the ESXi hosts

1.3.2 Edge Virtual Bridging

EVB is an IEEE standard that specifies the interaction between virtual switching environments in a hypervisor and the first layer of the physical switching infrastructure. The EVB 802.1Qbg standard addresses the conventional vSwitch’s lack of network management, monitoring, and security. Typically, a conventional vSwitch is invisible to and not configured by the network administrator. Additionally, any traffic that is handled internally by the vSwitch cannot be monitored or secured.

Support for the EVB 802.1Qbg standard enables VM traffic management in the physical/virtual network through Virtual Ethernet Port Aggregation (VEPA) and Virtual Station Interface (VSI) Discovery and Configuration Protocol (VDP). This enables live VM mobility across the network through automatic creation, migration, and deletion of VM-specific network port profiles.

1.4 VMware networking

This section provides an overview of the VMware networking environment in which the DVS 5000V operates. The concepts might be familiar for those experienced with server virtualization, but should serve as a useful introduction for a network specialist comfortable working with physical devices.

VMware networking allows the network administrator to build the server access layer of the typical LAN switching topology for the virtual server environment. VMware networking provides a Layer 2 switching engine within a ESXi host or across multiple hosts. The virtual switch provides server access ports to guest VMs and uplink ports that map to the physical host interfaces. A virtual switch operates in a similar fashion to a traditional physical switch, with MAC address forwarding tables and capabilities for VLAN separation. The virtual switch also has some key differences:

- ▶ Spanning Tree Protocol (STP) is not used or needed. Because the virtual switch is a stub node in the LAN topology, there is no way normal operation introduces loops, so STP is not needed. It is possible to introduce loops with some abnormal implementation, such as directly connecting ESXi hosts or running a virtual switch inside a VM.
- ▶ Server access ports are not bound, like fixed configuration physical switches, to a relatively small number of pre-set ports. Ports can be dynamically added or removed as required. (There are overall system maximums for the number of virtual ports.)
- ▶ Virtual switches cannot be directly interconnected (within the ESXi host). This is not required as it is in the physical LAN to increase port density. It also ensures the principles for a loop free topology is maintained.

vSwitch port limits: VMware virtual switching has port limits that are documented at the following website:

<https://www.vmware.com/pdf/vsphere5/r51/vsphere-51-configuration-maximums.pdf>

Figure 1-2 shows an overview of a VMware network with a vSphere Standard Switch.

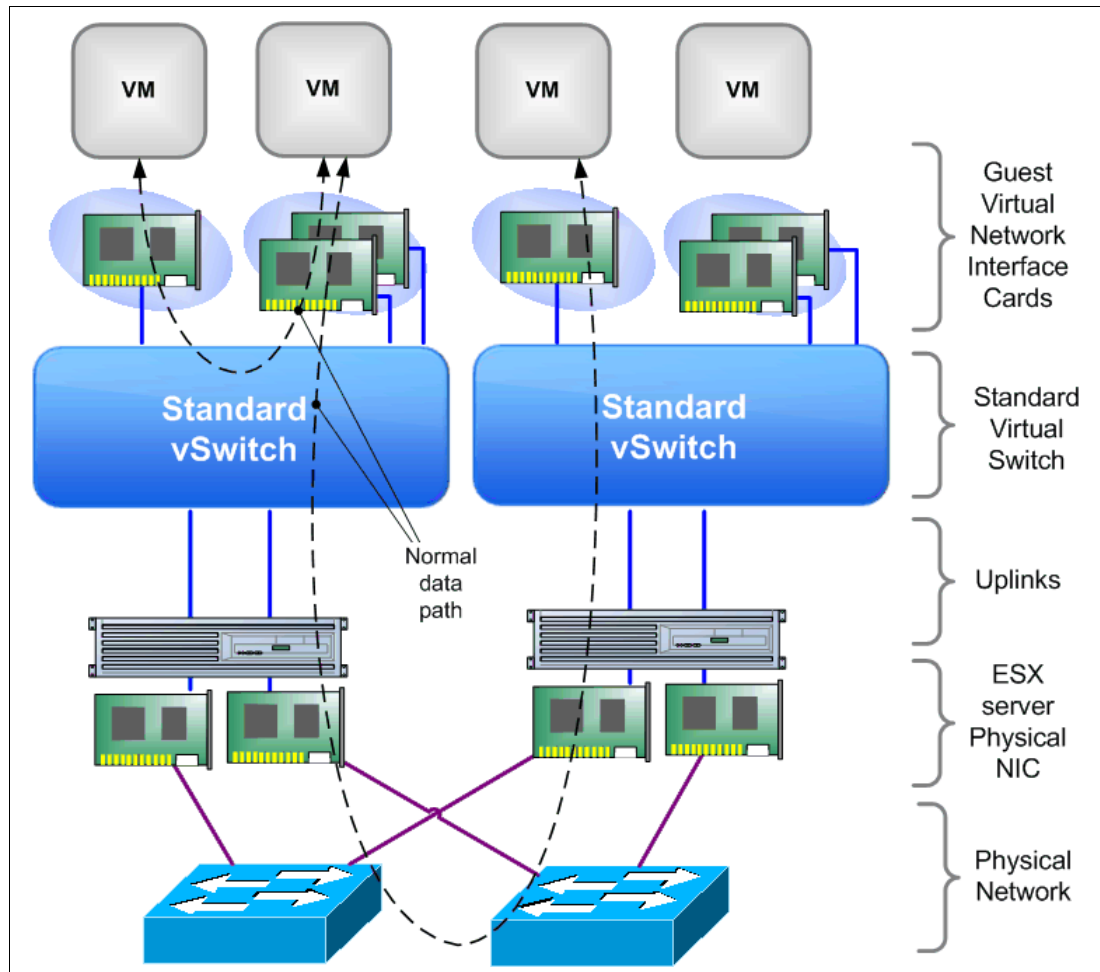


Figure 1-2 VMware network overview with a vSphere Standard Switch

1.4.1 Virtual switches

VMware offers two types of virtual switches

- ▶ vSphere Standard Switch: This switch is local to a particular ESXi host.
- ▶ vDS: This switch has a similar function to the Standard Switch, but extends across multiple ESXi hosts. The primary benefit of this implementation is that VMs can maintain a consistent network configuration as they migrate across hosts in a cluster (through VMotion).

The DVS 5000V is also a distributed switch, like the VMware version, but with enhanced functions and a common ISCLI for network management and administration.

The network overview diagram in Figure 1-3 on page 7 shows the usage of an abstracted DVS. This can either be the DVS 5000V, or the VMware vDS. The data path is the same as the Standard Switch. The principal function of the DVS is to have a common control plane that is VM-aware across multiple hosts. This means that network configuration in the vSwitch can follow the VM as it moves throughout the data center.

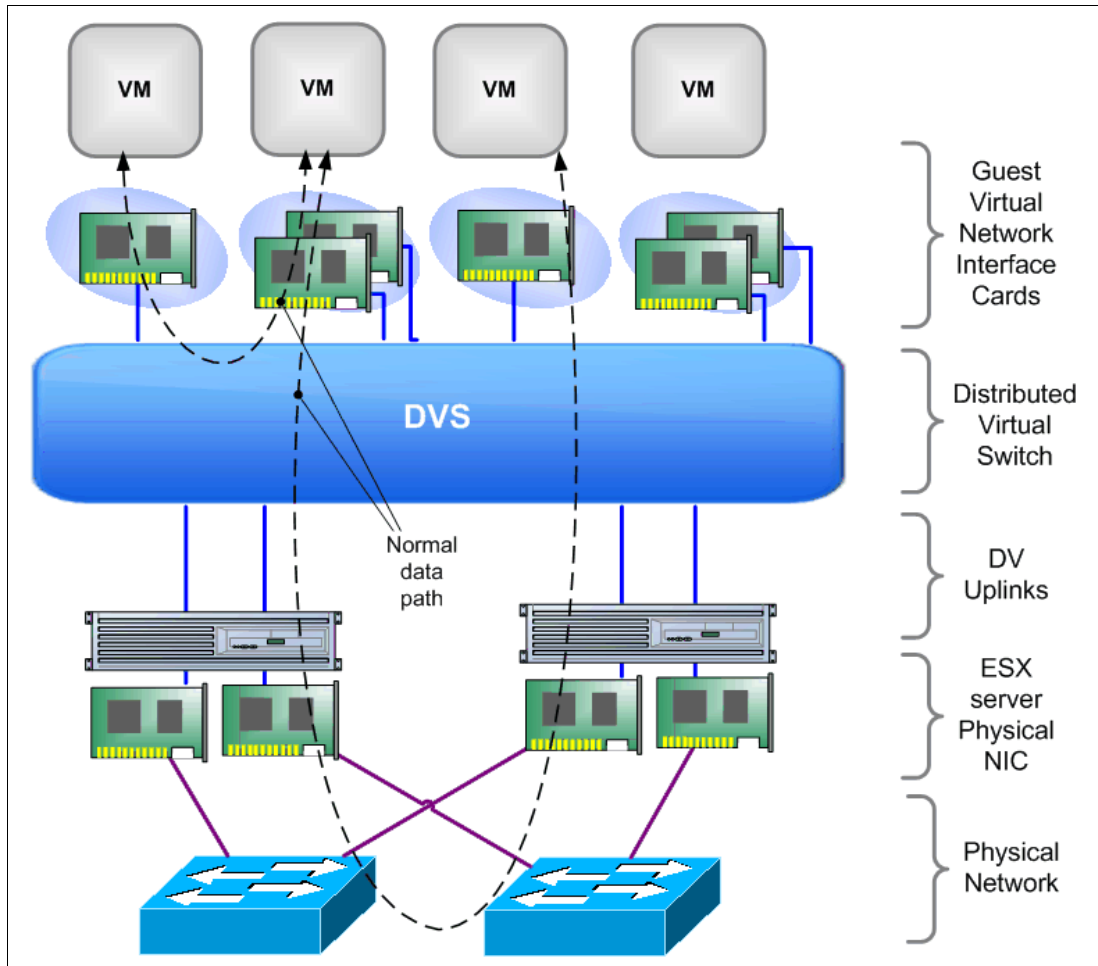


Figure 1-3 VMware network overview with a Distributed Virtual Switch

1.4.2 Port Groups and vNIC profiles

An important concept of VMware networking is Port Groups. Port Groups can be considered as templates that define a standard set of configuration that can be applied to a VM's network access. Port Groups can be used to define parameters such as VLAN, bandwidth management, and security. When defined, VMs can be assigned to a Port Group to inherit the required network configuration. This greatly simplifies and standardizes the network properties for VM provisioning. To support VM mobility, Port Groups must be manually replicated across hosts and Standard Virtual Switches. A Distributed Port Group, conversely, extends across multiple hosts as part of a DVS configuration.

The DVS 5000V uses the term vNIC profiles, which are synonymous with VMware Distributed Port Groups. vNIC profiles are configured in the DVS 5000V, and then appear as Distributed Port Groups that can be assigned to guests in VMware vSphere. vNIC profiles become a reusable building block for network administrators to provision network access and make it available to virtualization administrators.

For detailed information about configuring vNIC profiles in the DVS 5000V, see 5.2.1, "Configuring vNIC profiles" on page 53.

1.5 DVS 5000V enhancements since Version 1.0

IBM has developed enhancements for the DVS 5000V since the original release. The following section describes these modifications since Version 1.0. These enhancements to the product help incorporate a wider range of profiles, add vPorts per vDS in each Virtual Data Center (VDC), and provide a more intelligent recovery feature for the DVS 5000V Controller VM.

1.5.1 Enhancements

The DVS 5000V has the following enhancements:

- ▶ Support for 256 vNIC profiles per instance
- ▶ Support for 256 ports per host per instance
- ▶ Support for up to 60,000 VM ports per instance
- ▶ Support for recovery of the DVS 5000V Controller configuration upon appliance failure

Note: There is a 60,000-port per vCenter limit that is imposed by VMware on vSphere 5.1 and 5.5. A single DVS 5000V instance can use all 60,000 ports only if it is the only vDS in the vCenter.

Figure 1-4 shows the enhancements layout.

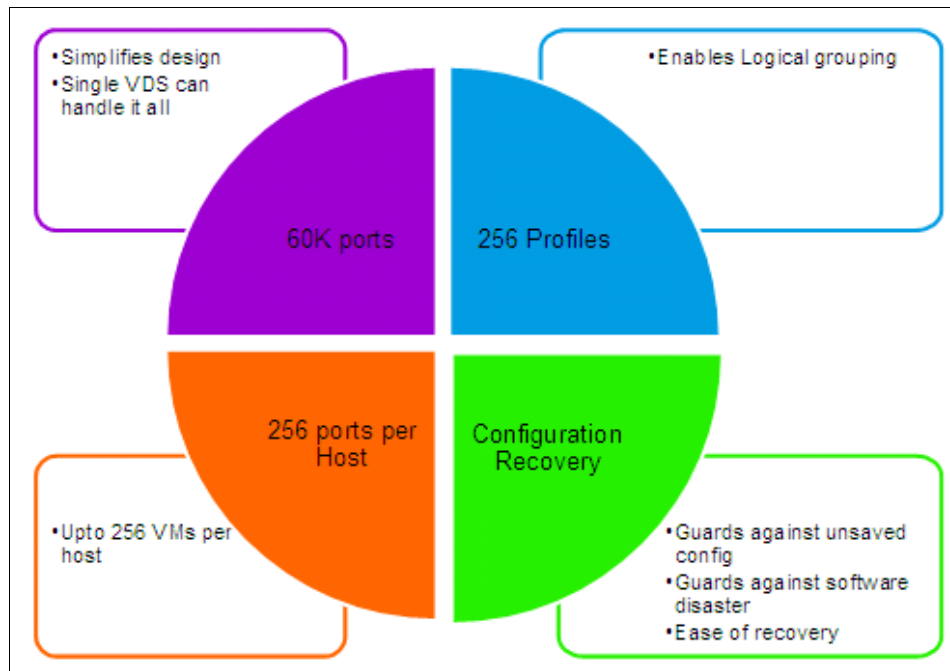


Figure 1-4 Enhancement layout of DVS 5000V

Table 1-1 on page 9 lists compatible VMware vSphere versions with the appropriate versions of code that are required in the DVS 5000V solution.

Table 1-1 Enhancement support matrix

Customer VMware version	Minimum required DVS 5000V Controller version	Minimum required DVS 5000V Host Module version
vSphere 5.1 and 5.0	1.1.1	1.1.2 ^a
vSphere 5.5	2.0	2.0 ^b

a. Host Module version 1.x supports only ESXi 5.0 and 5.1, and Host Module version 2.x supports only ESXi 5.5.

b. 5000V Host Module 2.0.0 is required for supporting IBM SDN VE, VMware Edition, on ESXi 5.5.

1.5.2 Portless profiles feature

The maximum number of ports on single DVS 5000V is increased from 4,000 to 60,000 through the concept of a *portless profile*. The ports in a portless profile have ports that are represented in VMware vCenter, but are not assigned port numbers by default at the DVS 5000V Controller, unlike the traditional vNIC profile support in previous releases. All vNIC profile settings are supported on a portless profile, but because there are no DVS 5000V Controller port numbers that are assigned to individual ports of a portless profile, the user cannot manage them individually through the interface menu.

A portless profile can be created by running the `iswitch vnicprof <profile-name> portless` command. Portless profiles are implicitly created when the `iswitch vnicprof <profile-name>` command is run (with the `portless` keyword omitted) when there are not enough ports available in the range 101 - 3900. When created, a portless profile can be manipulated through the vNIC profile menu, just like traditional vNIC profiles. Traditional vNIC profile implementation continues to be supported in this new release.

Although ports in a portless profile do not have a port number at the DVS 5000V Controller by default, it is possible to later map them to available ports from a fixed, reserved pool of 100 ports (3901 - 4000) to perform interface-level configuration (such as port mirroring) if needed. Use the `iswitch add-bos-mapping <vds-port-number> <bladeos-port-number> <profile-name>` command for this mapping. `vds-port-number` refers to the port number assigned by the vCenter, and `bladeos-port-number` is the port number that is used on the DVS 5000V Controller.

Limitations

Because there is no 802.1Qbg IBM physical switch corresponding port, port-specific operations are not possible with portless profiles. The lack of 802.1Qbg IBM physical switch corresponding ports in portless profiles means that the following features are not supported:

- ▶ ERSPAN
- ▶ sFlow
- ▶ Port mirroring
- ▶ Per-port statistics at the DVS 5000V Controller (statistics can still be viewed in VMware vCenter)
- ▶ Ability to shut down and re-enable a specific port in the profile

1.5.3 Recovery feature

The DVS 5000V Controller now supports the full recovery of the running configuration even if there is a complete failure of the DVS 5000V Controller appliance with no copy of the configuration that is saved elsewhere. In previous releases, the user had to copy the running configuration to a remote location periodically through TFTP or SCP to re-create the configuration in the unlikely event of a complete failure of the DVS 5000V Controller appliance. If such a remote copy was not made, the configuration had to be re-created from scratch, resulting in user disruption.

In the latest release, the complete running the configuration of the DVS 5000V Controller is automatically saved to vCenter every 10 seconds. In the unlikely event that the DVS 5000V Controller and its configuration are lost, a new DVS 5000V Controller Appliance can be deployed and used in place of the failed DVS 5000V Controller that previously was used to configure the DVS 5000V. The configuration recovery mechanism is triggered when the user enters the `vcenter` and `vds` commands with the same parameters as on the old DVS 5000V Controller. The new DVS 5000V Controller fetches the entire configuration from the vCenter and configures itself automatically.

1.5.4 Upgrade procedure for the enhancements

When upgrading from Version 1.1.0, if the start configuration includes traditional vNIC profiles and the `dmc` command, reloading the Controller might fail after the upgrade, depending on the order in which the commands were originally issued by the user. Although the problem might appear to have been caused by the upgrade, it is a result of the configuration commands being stored in an incorrect order in Version 1.1.0. This scenario also occurs during reload of the Controller within Version 1.1.0. This ordering issue has been fixed in Version 1.1.1, but it might be seen during an upgrade from Version 1.1.0 to 1.1.1.

To fix the issue before the upgrade, and upgrade to the latest version, perform the following steps:

1. Save and copy the running configuration through TFTP or SCP to an external server.
2. Edit the configuration file to move the `iswitch dmc` command from its original location to after the `iswitch vnicprof` commands and the `iswitch addports` commands (if any).
3. Copy the modified configuration file back to the Controller through TFTP or SCP to the start configuration-block.
4. Copy the upgraded image to one of the image blocks.
5. Reload the Controller.

Your controller should now be successfully upgraded.

1.6 More information

For more information about the DVS 5000V, see the User Guide at the following website:

<http://www.ibm.com/support/docview.wss?uid=isg3T7000628>



IBM Distributed Virtual Switch 5000V reference architecture

This chapter presents the network architecture that is used in this book to implement the IBM Distributed Virtual Switch 5000V (DVS 5000V) solution by using IBM PureFlex® System and IBM System Networking switches.

This design is based on preferred practices and the experience of the authors and is meant to provide support for the implementation chapters later in this book. This network topology does not represent a complete data center or server farm network architecture, but it is a simple topology that is used to show the capabilities and features of the DVS 5000V.

This chapter covers the following topics:

- ▶ Architecture overview
- ▶ Physical architecture
- ▶ Logical architecture

2.1 Architecture overview

The following equipment is used for the network topology that is described in this chapter:

- ▶ Two IBM System Networking RackSwitch™ G8264 switches
- ▶ Two IBM Flex System® Fabric EN4093/R 10Gb Scalable Switches
- ▶ Three IBM Flex System x240 Compute Nodes, with VMWare vSphere 5.1 installed
- ▶ DVS 5000V appliance
- ▶ VMware vCenter 5 appliance
- ▶ Linux operating system as the guest OS

2.2 Physical architecture

The physical architecture describes Layer 1 connections between the components that are described in 2.1, “Architecture overview” on page 12.

Table 2-1 lists the hardware components.

Table 2-1 Hardware components

Host name	Device type	Firmware version	Management IP address
G8264tor_1	G8264	7.4.1	172.25.101.243
G8264tor_2	G8264	7.4.1	172.25.101.244
EN4093flex_1	EN4093	7.5.1	172.25.101.238
EN4093flex_2	EN4093	7.5.1	172.25.101.239
5000V	DVS 5000V	1.0.1.1768	172.25.155.5

Two G8264 RackSwitches are connected to two EN4093 embedded switches with four aggregation links using vLAG with 10 Gb Fibre cable and SFP+ Transceiver connectors. There are ISL connections between both G8264 rack switches and also between both EN4093 switches.

The interconnection details between the Enterprise Chassis switch and Top of Rack switch are shown in Figure 2-1.

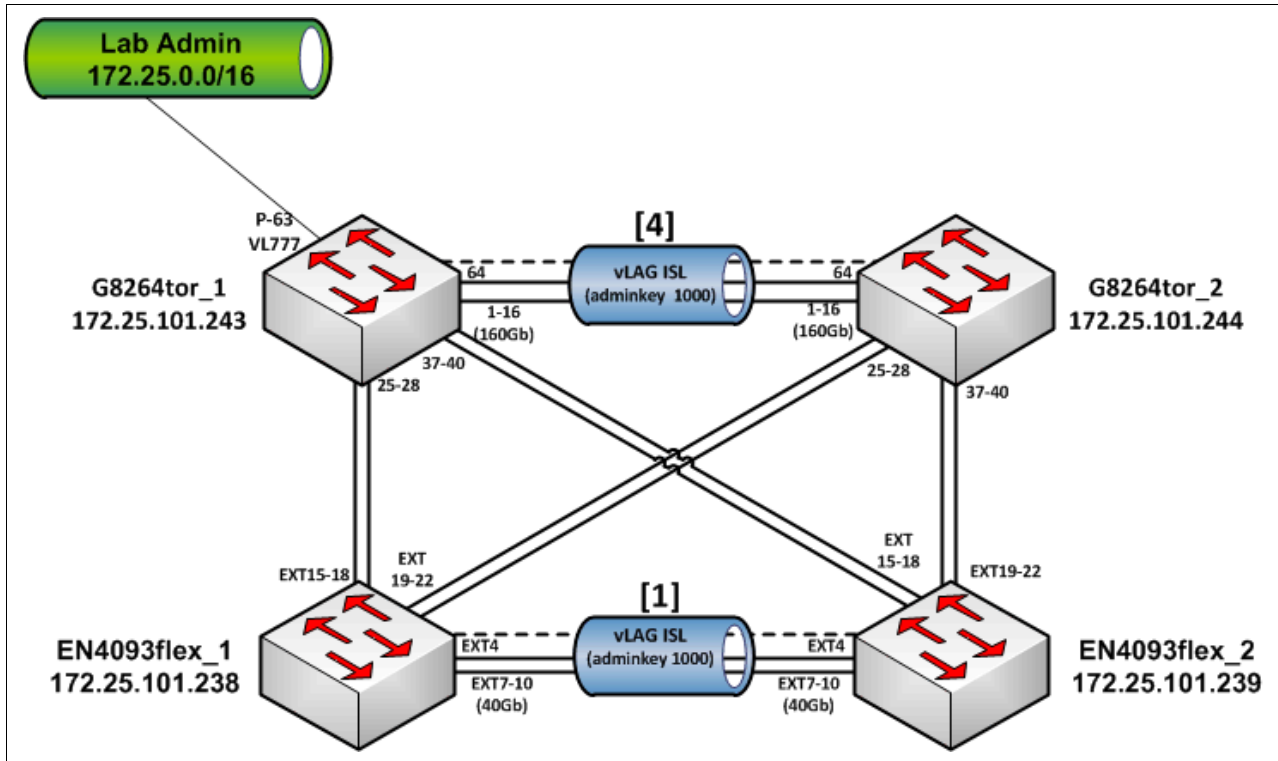


Figure 2-1 Physical architecture between IBM PureFlex System chassis switches with Top of Rack switches

From the IBM DVS 5000V perspective, the internal ports of the EN4093 embedded switches use compute node uplinks, where each compute node has two NICs, as described in Figure 2-2.

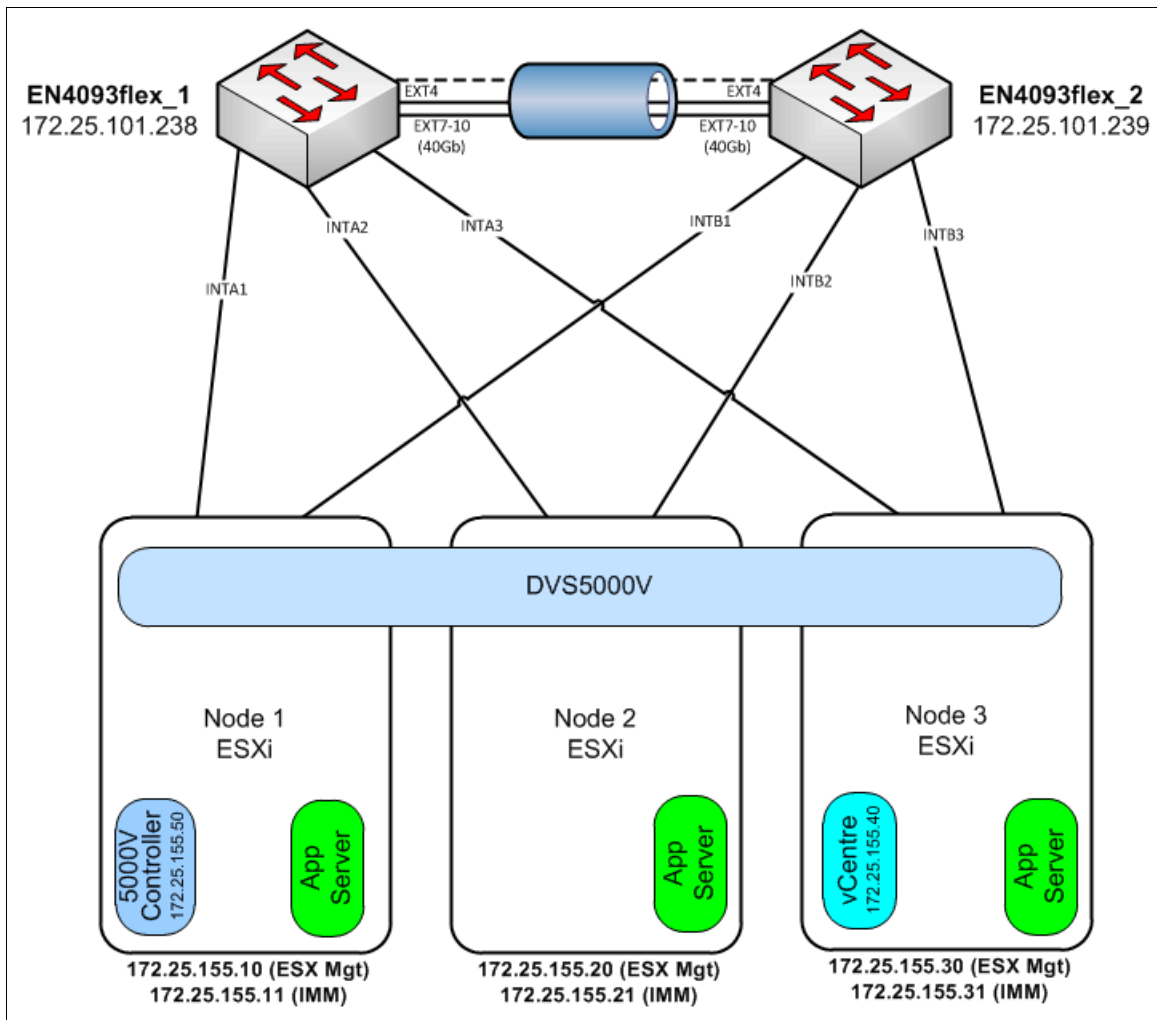


Figure 2-2 Physical architecture within a PureFlex System chassis

2.3 Logical architecture

This section covers the Layer 2 and 3 configuration that was used in the lab topology, focusing on the IBM DVS 5000V related configuration on the EN4093 and G8264 pair.

2.3.1 Layer 2

This section describes the virtual LAN (VLAN) configuration that was applied in the lab scenario. As mentioned in 1.4, “VMware networking” on page 5, Spanning Tree Protocol (STP) is not applicable within the DVS 5000V.

Virtual LAN

Table 2-2 on page 15 shows the VLAN that is used in this topology. In the EN4093 and G8264 switches, all of these VLANs are tagged in each switch interconnection port.

Table 2-2 VLAN list and port memberships in the DVS 5000V

VLAN number	Description	Port member
101	DATA-1	dvportgroup-98 Ports 1,121-140,221-240
102	DATA-1	dvportgroup-99 Ports 141-160
103	DATA-3	dvportgroup-100 Ports 161-180
192	VMotion Network	N/A (non-routed VLAN)
200	Private VLAN Primary	dvportgroup-121 Ports 201-220
203	Private VLAN Isolated	dvportgroup-122 Ports 181-200
777	Management VLAN	Ports 101-120

VLAN 192 is a non-routed VLAN that is used for VMotion traffic between machines in the cluster.

2.3.2 Layer 3

The Layer 3 architecture covers IP addressing and Virtual Router Redundancy Protocol (VRRP). VRRP was chosen for seamless failover for the RackSwitch G8264 Top-of-Rack (ToR) pair.

IP addressing

The IPv4 addressing plan that we used in this lab scenario can be found in Table 2-3. The list consists of the interface VLAN IP addresses of each VLAN used in the topology.

Table 2-3 VLAN IP addressing

VLAN number	Description	Network	Interface VLAN in ToR 1	Interface VLAN in ToR 2
101	DATA-1	192.168.1.0/24	192.168.1.2	192.168.1.3
102	DATA-1	192.168.2.0/24	192.168.2.2	192.168.2.3
103	DATA-3	192.168.3.0/24	192.168.3.2	192.168.3.3
192	VMotion Network	192.168.192.0/24	Non-routed	Non-routed
200	Private VLAN Primary	192.168.200.0/24	192.168.200.2	192.168.200.3
203	Private VLAN Isolated	192.168.203.0/24	192.168.203.2	192.168.203.3
777	Management VLAN	172.25.0.0/16	N/A	N/A

Virtual Router Redundancy Protocol

VRRP specifies an election protocol that dynamically assigns responsibility for a virtual router to one of the VRRP routers on a LAN. The VRRP router that controls the IP address that is associated with a virtual router is called the master, and it forwards packets that are sent to these IP addresses. The election process provides dynamic failover of the forwarding responsibility if the master becomes unavailable.

The priority value is given 110 in ToR 1 and 101 in ToR 2 to make ToR 1 become the VRRP master router and ToR 2 to become the backup router. Details of the VRRP configuration can be found in Table 2-4.

Table 2-4 VRRP configuration

VLAN number	Description	VRRP group	IP address
101	DATA-1	1	192.168.1.1
102	DATA-1	2	192.168.2.1
103	DATA-3	3	192.168.3.1



IBM Distributed Virtual Switch 5000V installation

This chapter covers the steps for the basic deployment, installation, and initial configuration of the IBM Distributed Virtual Switch 5000V (DVS 5000V).

This chapter covers the following topics:

- ▶ DVS 5000V installation prerequisites
- ▶ DVS 5000V Host Module installation
- ▶ DVS 5000V Controller installation
- ▶ DVS 5000V licensing procedures

3.1 DVS 5000V installation prerequisites

Before installing the DVS 5000V, ensure that the VMware infrastructure meets all the requirements that are listed in this section:

- ▶ The following software files of DVS 5000V are required:
 - The DVS 5000V Controller (an Open Virtual Appliance (OVA) file).
 - The DVS 5000V vDS host module (a compressed file (ZIP) file).
- ▶ VMware vCenter Server and ESXi ≤ 5.0 with an Enterprise Plus license must be properly installed, connected, and fully functional.
- ▶ In each ESXi host where the Host Module of DVS 5000V must be installed, the following criteria must be fulfilled:
 - There must be more than one host for vMotion.
 - There must be at least one 1 Gb or 10 Gb physical NIC. Multiple-separated physical NICs are recommended for redundancy and traffic separation between management, VMKernel, and virtual machine (VM) traffic.
 - There must be L2/L3 functional network connectivity.
- ▶ Here is the list of requirements for the deployment of the DVS 5000V Controller appliance VM:
 - There must be at least 1 GB of physical memory that is available in the ESXi host.
 - There must be at least one 1 Gb or 10 Gb physical NIC. A multiple-separated physical NIC is recommended for redundancy and traffic separation between management, VMKernel, and the VM.
 - The DVS 5000V Controller must have connectivity to the vCenter server and all hosts that handle DVS 5000V operations.
 - For high availability and protecting the controller against unscheduled downtime, activate VMware High Availability (HA) and Fault Tolerance (FT) features.

3.2 Installing the DVS 5000V Host module

The software is available in two formats: as a DVS 5000V vSphere Installation Bundle (VIB) file, or as an offline bundle (ZIP) file. The software can be installed on one or more ESXi hosts by using the vCLI or ESXi shell (ESXCLI) on each individual host, or by using the VMware Update Manager (VUM) to multiple hosts at once.

3.2.1 Installation using the ESXi shell

To install the DVS 5000V Host module by using the ESXCLI, complete the following steps:

1. SSH access must first be enabled on the ESXi host to use the ESXCLI. The option for enabling SSH can be found in the ESXi console in the Troubleshooting Mode Options menu.
2. Copy the DVS 5000V VIB file or as an offline bundle (ZIP) file by using Secure Copy (SCP).
3. Run `esxc1i` on the host for to install the host module. As shown in Example 3-1 on page 19, the ESXi host must be rebooted after the installation process.

Example 3-1 ESXCLI commands for performing host modules installation along with the output

```
~ # esxcli software vib install -d <<path to ZIP file>>
-OR-
~ # esxcli software vib install -v <<path to VIB file>>
```

Installation Result

Message: The update completed successfully, but the system needs to be rebooted for the changes to be effective.

Reboot Required: true

VIBs Installed: IBM_bootbank_ibm-esx-5000V_1.0.1-1646

VIBs Removed:

VIBs Skipped:

4. After rebooting the host, verify that the host module installation was successful by running the ESXCLI commands that are shown in Example 3-2.

Example 3-2 Host module verification by using the ESXCLI commands

```
~ # esxcli software vib list | grep 5000V
ibm-esx-5000V 1.0.1-1646 IBM VMwareAccepted 2012-11-19
```

```
~ # vmkload_mod -l | grep 5000V
ibm_5000V 3 124
```

```
~ # ps | grep 5000V
4616 5000V_helper_world
4617 5000V_lldp_world
5033 5033 ibm_5000V_agent /opt/ibm/sbin/ibm_5000V_agent
5041 5033 ibm_5000V_agent /opt/ibm/sbin/ibm_5000V_agent
5042 5033 ibm_5000V_agent /opt/ibm/sbin/ibm_5000V_agent
5043 5033 ibm_5000V_agent /opt/ibm/sbin/ibm_5000V_agent
5044 5033 ibm_5000V_agent /opt/ibm/sbin/ibm_5000V_agent
5045 5033 ibm_5000V_agent /opt/ibm/sbin/ibm_5000V_agent
5046 5033 ibm_5000V_agent /opt/ibm/sbin/ibm_5000V_agent
```

Note: The VMware vSphere Update Manager (VUM) plug-in can also be used to automate the host module installation across multiple ESXi hosts in the data center.

More details can be found in the IBM System Networking Distributed Switch 5000V User Guide, available at the following website:

<http://www.ibm.com/support/docview.wss?uid=isg3T7000628>

3.3 Installing the DVS 5000V Controller

This section covers the DVS 5000V Controller installation, which acts as the DVS across multiple ESXi hosts. The software is distributed as an OVA. Complete the following steps:

1. Obtain the OVA file from your IBM account representative and put it on a system where vSphere Client will be started to access the ESXi host or vCenter server (such as an administrative workstation or notebook).
2. Start the vSphere Client and connect to the vCenter server or ESXi host where the controller will be deployed.

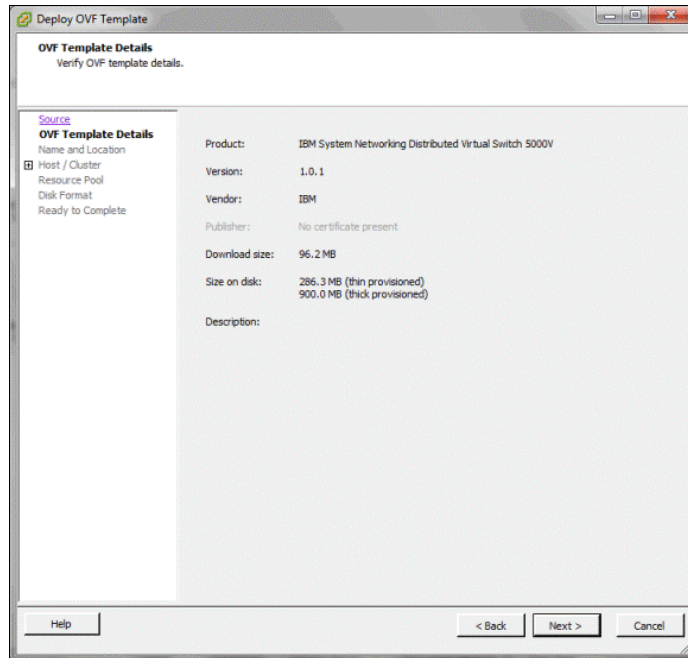


Figure 3-3 Verifying the controller OVF image

6. Provide the VM name for the DVS 5000V Controller, choose the inventory location, and click **Next**, as shown in Figure 3-4.

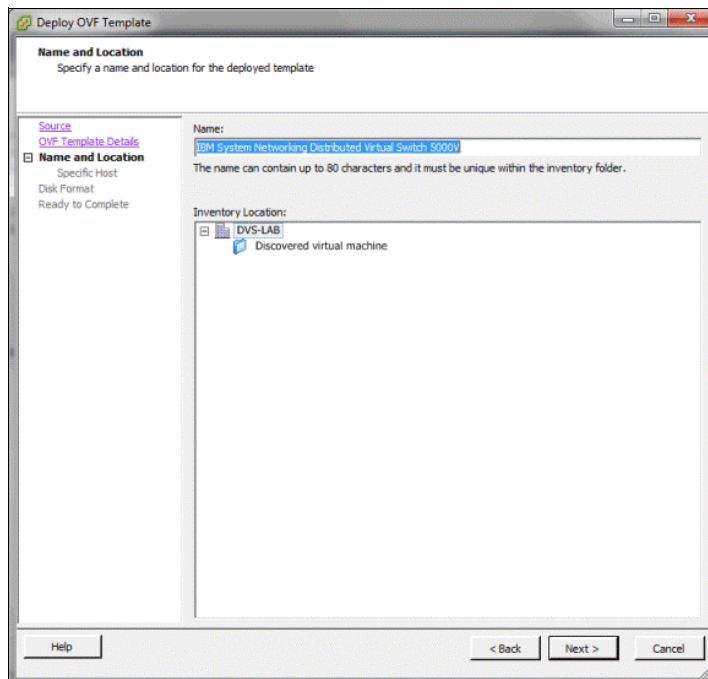


Figure 3-4 Providing a name and choosing an inventory location for the controller VM

- Specify the host on which the controller will be deployed and click **Next**, as shown in Figure 3-5.

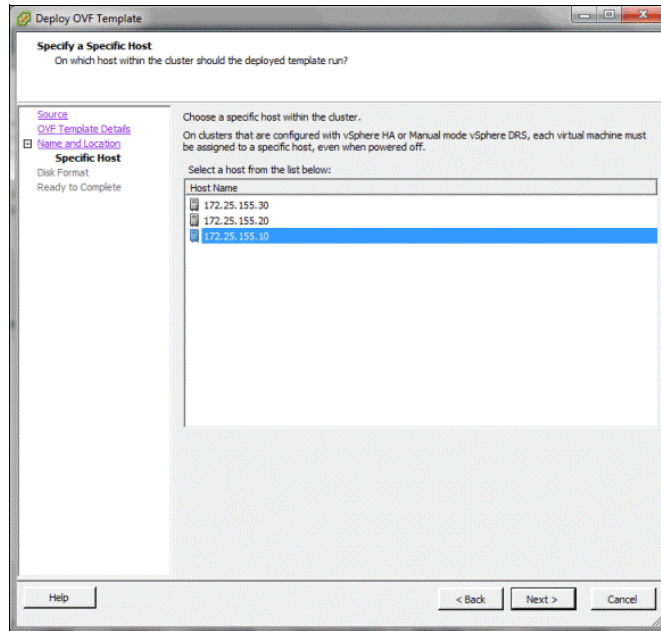


Figure 3-5 Selecting the host where the controller will be placed

- Specify the location from the available storage of the host where the Controller VM will be stored and click **Next**, as shown in Figure 3-6.

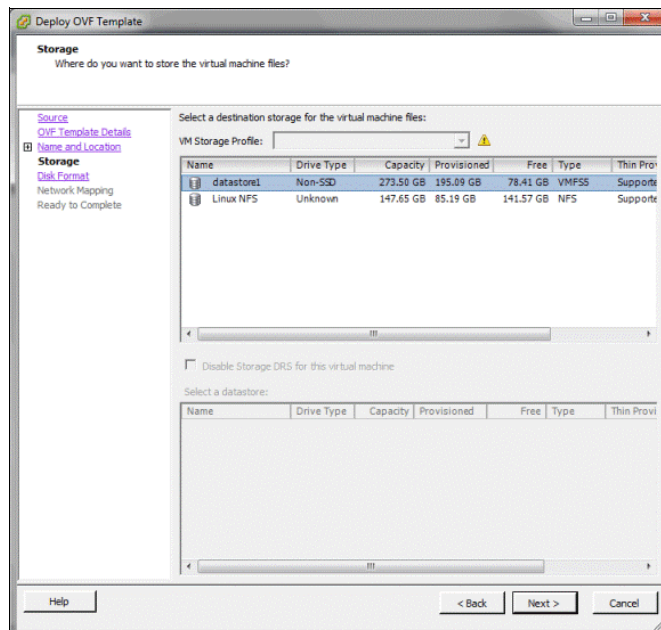


Figure 3-6 Selecting the location to store the controller VM from available storage on the host

- Select a disk format and click **Next**. As shown in Figure 3-7 on page 23, the recommendation is Thick Provisioned Lazy Zeroed.

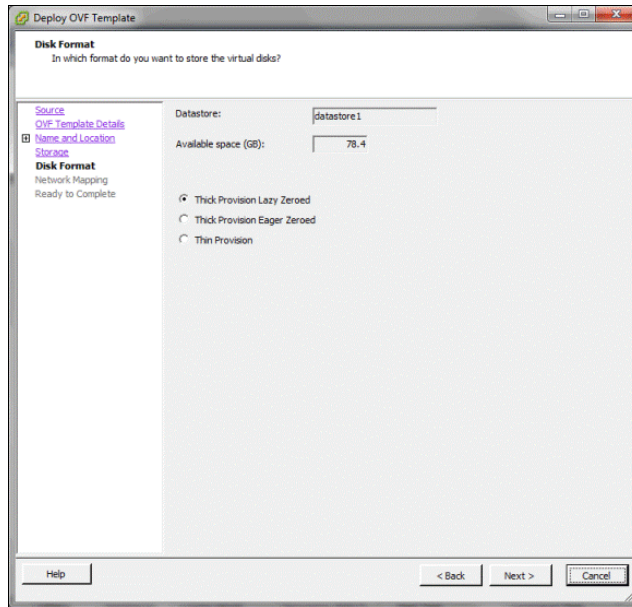


Figure 3-7 Selecting a disk format

10. Map the network for management purposes of the controller, as shown in Figure 3-8, and click **Next**.

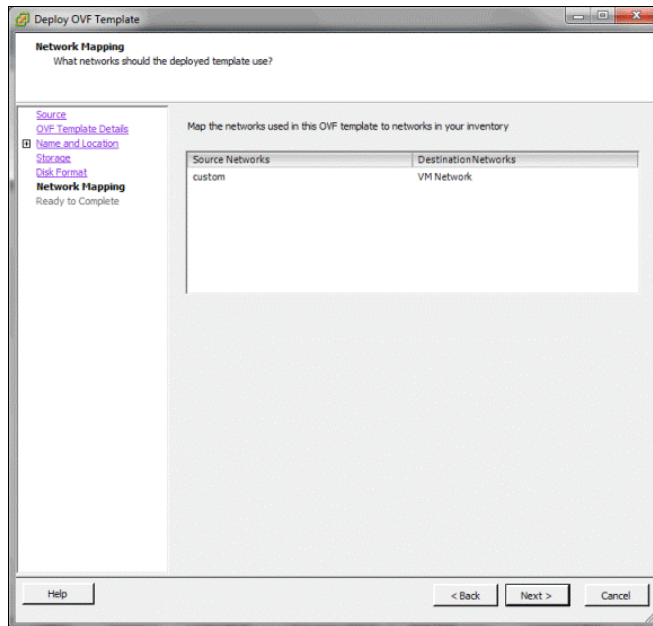


Figure 3-8 Choosing the destination network to which the management port of the controller will be attached

11. As shown in Figure 3-9, verify the specified options of deployment, select **Power on after deployment**, and click **Next**.

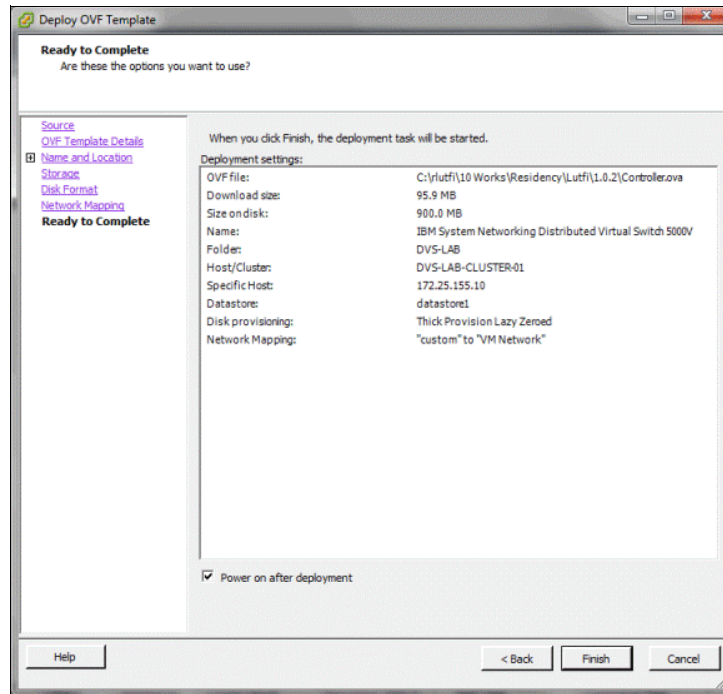


Figure 3-9 Verifying the specified option before starting the deployment

This initiates the controller deployment, as shown in Figure 3-10.

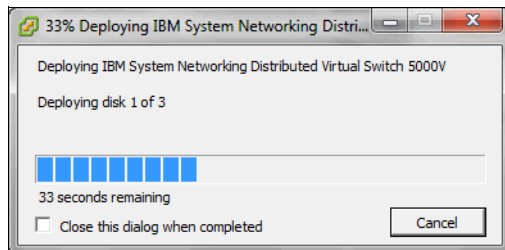


Figure 3-10 OVF deployment progress bar of the DVS 5000V Controller

12. After the deployment is complete, the controller VM is powered on and the VM console opens, as shown in Figure 3-11 on page 25.

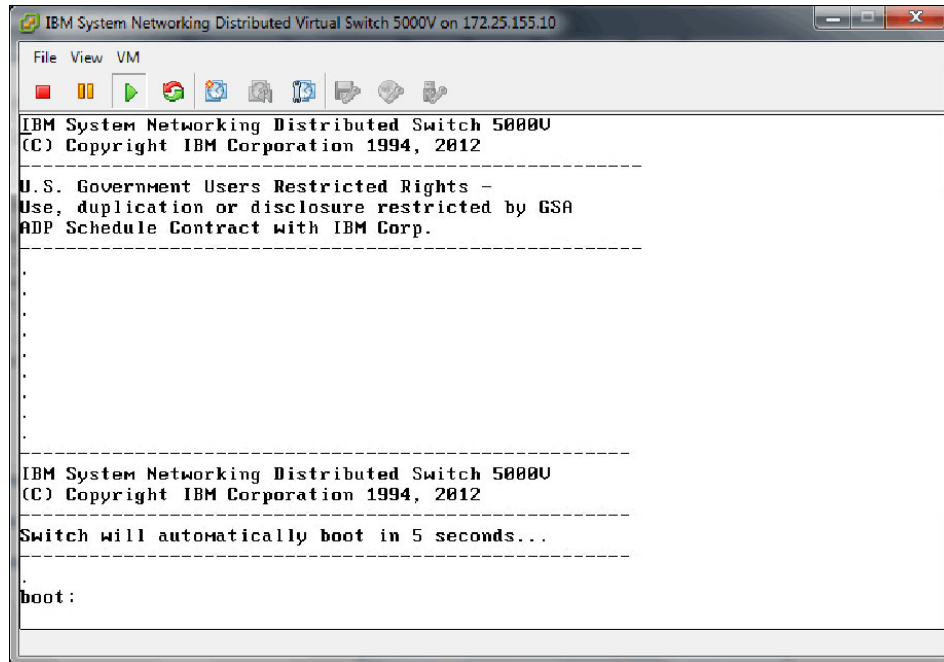


Figure 3-11 First boot window after the controller powers on

3.4 DVS 5000V initial configuration

After completing the controller installation, initial configuration must be done by entering commands through the Industry-Standard Command-Line Interface (ISCLI). The ISCLI can be accessed through the VM console from vSphere Client. More information about ISCLI can be found in Chapter 13. “CLI Basics”, of the *IBM System Networking Distributed Switch 5000V User Guide*, found at:

<http://www.ibm.com/support/docview.wss?uid=isg3T7000628>

Note: DVS 5000V configuration must be done through the ISCLI in the Controller interface (or less commonly through IBM System Networking Switch Center) instead of from the vCenter (even when the vCenter interface seems to allow the user to perform administrative actions).

Host operations, such as adding ESXi hosts and uplinks or assigning VM network interfaces to vDS ports or profiles, must be done through the vCenter interface.

3.4.1 Setting the IPv4 management address

To add the DVS 5000V Controller to the network with a static IP address, complete the following steps:

1. Log in to the controller through the vSphere Console. The default user name and password are shown in Example 3-3.

Example 3-3 Logging in to the DVS 5000V Controller Console

```
user      :admin
password  :admin
```

2. Enter the configuration mode by running the commands that are shown in Example 3-4.

Example 3-4 Entering the configuration mode

```
5000V> en
Enable privilege granted.
5000V# configure terminal
Enter configuration commands, one per line. End with Ctrl/Z.
5000V(config)#
```

3. Set the management IPv4 address. This IP address is used for the communication between DVS 5000V, vCenter, and participating hosts. By default, the DHCP option is enabled in the controller. If a DHCP server exists in the environment, the controller can acquire an IP address automatically. However, manual configuration is used in this lab scenario. The commands for manual configuration are shown in Example 3-5.

Example 3-5 Manual configuration for setting the management IP address

```
5000V(config)#interface ip-mgmt address 172.25.155.5 255.255.0.0 172.25.1.1
5000V(config)#interface ip-mgmt gateway enable
```

3.4.2 Enabling remote access

The initial configuration can be performed only through the vSphere console. However, there are two features available for the remote access of DVS 5000V: Telnet and SSH.

Telnet access

The purpose of the Telnet protocol is to provide a fairly general bidirectional, eight-bit, and byte-oriented communications facility. Its primary task is to provide a standard method of interfacing terminal devices. The protocol can also be used for terminal-terminal communication (“linking”) and process-process communication (distributed computation). By default, Telnet access is disabled. Here is the command to enable Telnet access:

```
5000V(config)# access telnet enable
```

Note: By default, Telnet traffic uses application port 23/TCP.

Secure Shell access

Although Telnet provides remote access for the network administration and configuration tasks, this protocol cannot provide a secure connection. Another option for the remote access is Secure Shell (SSH). Unlike telnet, SSH is a protocol for secure remote login and other secure network services over an insecure network. SSH ensures that all the data that is transmitted over the network is secured and encrypted. By default, SSH is disabled in the DVS 5000V. Here is the command to enable SSH feature:

```
5000V(config)# ssh enable
```

Note: By default, application port 22/TCP is used for SSH access.

3.5 Creating the Global vDS instance

From the vCenter perspective, there is a hierarchal structure of the virtual environment, as shown in Figure 3-12. In a vCenter instance, one or more data centers can be defined. Each data center is composed of logical sets of ESXi hosts, clusters, and vSphere Distributed Switches (vDSs).

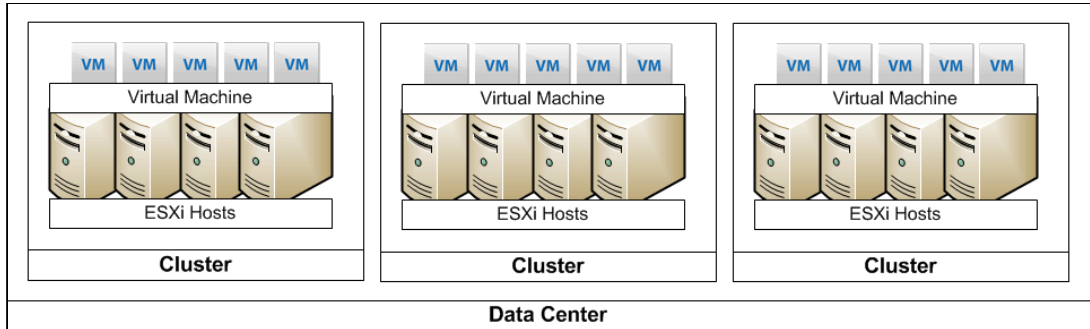


Figure 3-12 Virtual infrastructure hierarchy in VMware vSphere

To summarize, the global vDS instance commands create an association between the Virtual Data Center (VDC) and the DVS 5000V. The commands in Example 3-6 are required to set up the vDS instance.

Example 3-6 Associating with the VMware vSphere Server

```
5000V(config)# iswitch vcenter 172.25.155.40 root
Password:
vCenter-Port (default 443):
5000V(config)# iswitch vds ?
STRING vds name
5000V(config)# iswitch vds 5000V ?
STRING Datacenter name
5000V(config)# iswitch vds 5000V DVS-LAB ?
<cr> create a new vds
5000V(config)# iswitch vds 5000V DVS-LAB
```

After completing these commands, the global vDS instance of DVS 5000V displays as shown in Figure 3-13.

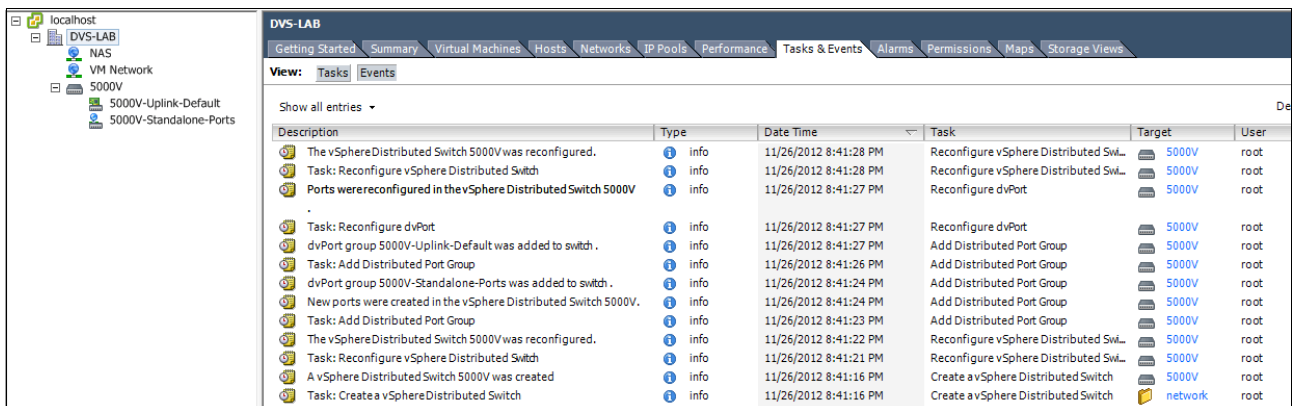


Figure 3-13 Global vDS instance for DVS 5000V in vCenter

3.6 DVS 5000V licensing

When the DVS 5000V Controller and host module installation is complete, the next step is installing the license. DVS 5000V licenses are based on the number of physical CPU sockets, regardless the number of cores in each CPU. Each host is added to the global vDS instance only if there are enough sockets remaining in the license. Otherwise, the host still can be added to the global vDS instance, but the virtual ports that are used by the individual VMs are in a disabled or blocked state, which prevents network traffic to and from the VM.

3.7 Adding the ESXi host to the DVS 5000V

Within a VDC, you must establish an association between the DVS 5000V controller with host modules in each participating ESXi host.

To add a host to the controller, complete the following steps:

1. From the vSphere Client, log in to the vCenter server.
2. From the home window, click **Inventory** → **Networking**.
3. Right-click the vDS instance and select **Add Host**, as shown in Figure 3-14.

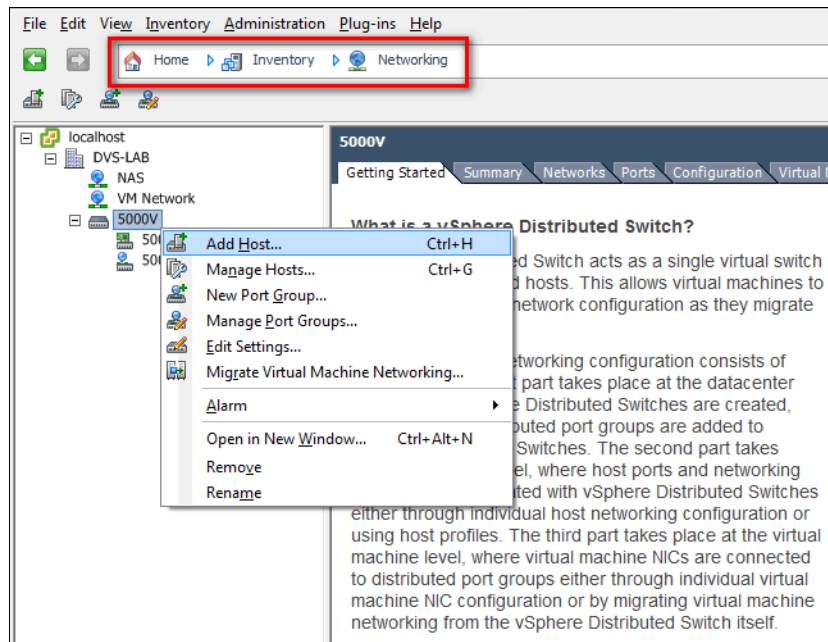


Figure 3-14 Adding a host to the DVS 5000V instance

- Select the host that you want to add. Also, specify the physical NICs from that selected host (see Figure 3-15) that should be used as the uplinks to the DVS 5000V for carrying VM traffic.

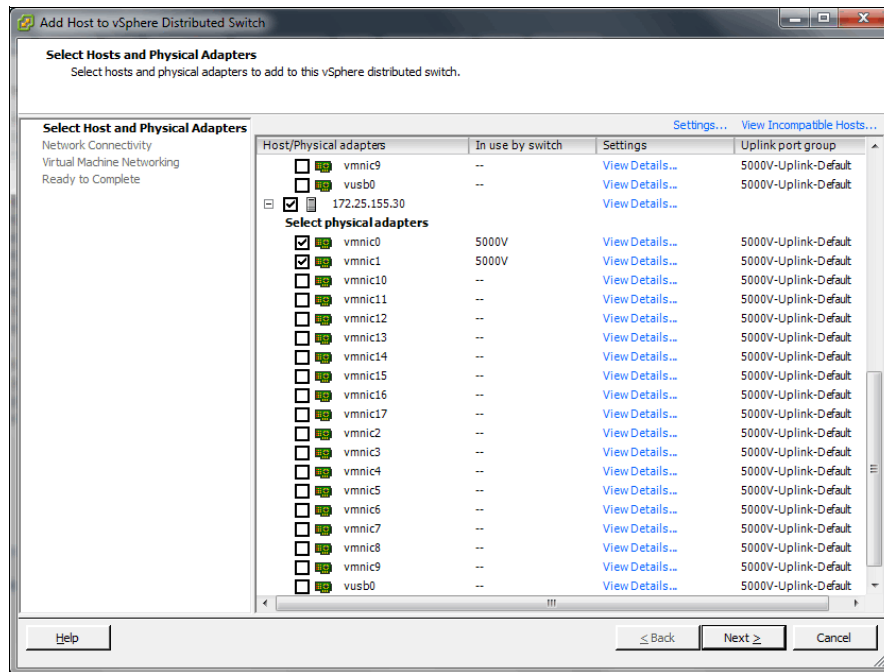


Figure 3-15 Selecting the host and physical NIC to be added to a global vDS instance

- Verify the specified options for adding hosts to the DVS 5000V controller and click **Next** to initiate the host addition process, as shown in Figure 3-16.

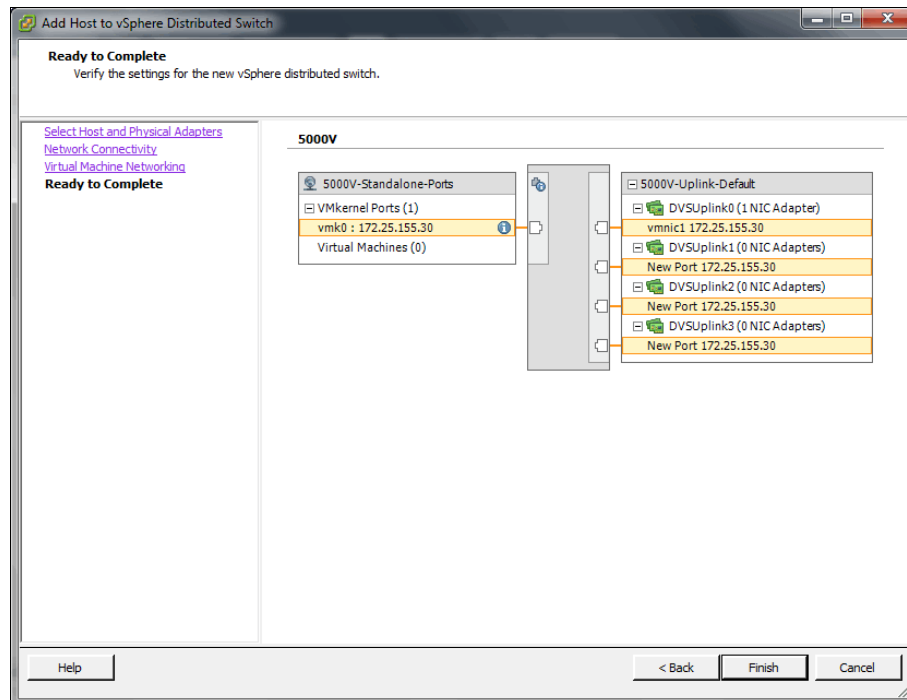


Figure 3-16 Verifying the options that are selected for adding the host to a DVS 5000V controller

After the process is completed, the selected host displays in the DVS 5000V instance, as shown in Figure 3-17.

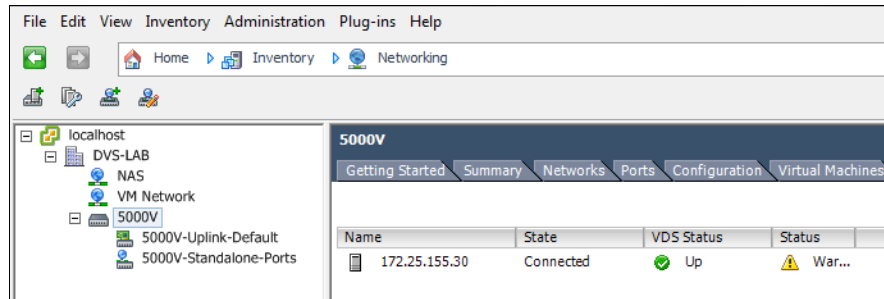


Figure 3-17 Host member in the DVS 5000V instance

Further configuration, such as VLAN, LAG, QoS, ACL, and 802.1Qbg, is described in Chapter 4, “Systems management” on page 31.



Systems management

One of the many benefits of the IBM Distributed Virtual Switch 5000V (DVS 5000V) is that the management framework aligns with standard network management protocols and interfaces. This allows the DVS 5000V to be managed by network administrators, who are familiar with the physical network, through standard network management tools and procedures.

This chapter covers the following topics:

- ▶ Authentication, Authorization, and Accounting (AAA)
- ▶ Network management protocols
- ▶ sFlow
- ▶ Port mirroring

4.1 Authentication, Authorization, and Accounting

The DVS 5000V supports administrator access by using the following items:

- ▶ Local user accounts
- ▶ RADIUS
- ▶ TACACS+

4.1.1 Local user accounts

The DVS 5000V has three built-in accounts, representing the three supported access levels. These users cannot be removed. An additional 10 local user accounts can be created. Table 4-1 lists the built-in accounts and the associated access levels.

Table 4-1 Default local user accounts

User name	Default status	Access level description	Initial password
user	Enabled	The user has no direct responsibility for switch management. The user can view all switch status information and statistics but cannot make any configuration changes to the switch.	user
oper	Disabled	The operator manages all functions of the switch. The operator can reset ports, except for the management port.	oper
admin	Always enabled	The super-user administrator has complete access to all commands, information, and configuration commands on the switch, including the ability to change both the user and administrator passwords.	admin

In Example 4-1, a new user named “nwadmin” is created with administrative privileges. The user is created at index 1 (of 10).

Example 4-1 Creating a local user

```
5000V(config)# access user 1 name nwadmin
5000V(config)# access user 1 level administrator
5000V(config)# access user 1 password
Changing nwadmin password; validation required.
Enter current admin password:
Enter new nwadmin password(max 128 characters):
Reenter new nwadmin password:
New "nwadmin" password accepted.
```

```
5000V(config)# access user 1 enable
```

The password for the built-in accounts can also be changed with subcommands of the **access user** command. The oper and user accounts can be disabled by entering an empty password.

Information about the provisioned users can be viewed with the **show** commands, as shown in Example 4-2 on page 33.

Example 4-2 Displaying user accounts on the DVS 5000V Controller

```
5000V# show access
```

```
Current System Access Settings:
```

```
Username:
```

```
admin - Always Enabled - online 1 session.  
user - enabled - offline  
oper - disabled - offline
```

```
Current User ID table:
```

```
1: name nwadmin , ena, cos admin , password valid, offline
```

```
Telnet server currently OFF
```

```
SNMP access currently read-write
```

```
TFTP occurs over port 69
```

```
SCP uses server port 22
```

```
5000V#
```

4.1.2 RADIUS authentication and authorization

A RADIUS server can be used to authenticate and authorize remote administrator access into the DVS 5000V. Example 4-3 demonstrates the configuration of RADIUS authentication and authorization.

The **secure-backdoor** setting is used to allow login when the RADIUS server is not available, using the local administrator account instead.

Example 4-3 RADIUS configuration

```
5000V(config)# radius-server primary-host 10.1.1.1 key my-shared-key  
5000V(config)# radius-server secondary-host 10.2.1.1 key my-shared-key2  
5000V(config)# radius-server retransmit 3  
5000V(config)# radius-server timeout 5  
5000V(config)# radius-server secure-backdoor  
5000V(config)# radius-server enable
```

When using RADIUS, authorization is provided by the RADIUS server returning the standard attribute, Service-Type(6). The Service-Type attribute is used to map a user into one of the three privilege levels that are described in Table 4-1 on page 32. The Service-Type value for the three levels is given in Table 4-2.

Table 4-2 Mapping of RADIUS Service-Type to privilege level

Privilege level	Service-Type value
User	255
Operator	252
Admin	6

4.1.3 TACACS+ authentication and authorization

A TACACS+ server can be used to authenticate and authorize remote administrator access into the DVS 5000V. Example 4-4 demonstrates the configuration of TACACS+ authentication and authorization. The **secure-backdoor** setting is used to allow login when the RADIUS server is not available, using the local administrator account.

Example 4-4 TACACS+ configuration

```
5000V(config)# tacacs-server primary-host 10.1.1.5 key my-tac-key
5000V(config)# tacacs-server secondary-host 10.2.1.5 key my-tac-key2
5000V(config)# tacacs-server retransmit 2
5000V(config)# tacacs-server secure-backdoor
5000V(config)# tacacs-server enable
```

When using TACACS+, access-level authorization is mapped based on the returned TACACS+ level. There are default and alternative mapping options available, as shown in Table 4-3. The alternating mapping aligns with standard TACACS+ levels for administrative privilege, so the administrator can simplify TACACS+ server configuration in a mixed-vendor network. Alternative mapping is enabled with this command:

```
5000V(config)# tacacs-server privilege-mapping
```

Table 4-3 TACACS+ authorization levels

DVS 5000V access level	Default TACACS+ level	Alternative TACACS+ mapping
user	0	0 - 1
oper	3	6 - 8
admin	6	14 - 15

The DVS 5000V also provides command-level authorization and accounting. Enabling command authorization configures the DVS 5000V to authorize each command execution with the TACACS+ server. Command logging enables logging of each command that is run.

Example 4-5 shows enabling TACACS+ command authorization and logging.

Example 4-5 TACACS+ command authorization and logging

```
5000V(config)# tacacs-server command-authorization
5000V(config)# tacacs-server command-logging
```

4.2 Network management protocols

This section describes some of the network management protocols that can be configured on the DVS 5000V.

4.2.1 Simple Network Management Protocol

The DVS 5000V supports Simple Network Management Protocol (SNMP) Versions 1, 2, and 3. SNMP is enabled by default with default read and write community strings of “public” and “private”.

SNMP defaults: It is a preferred security practice that the SNMP defaults be changed or disabled as part of the initial configuration.

The SNMP MIB file is included with the DVS 5000V software bundle. The standard MIBs listed here are also supported:

- ▶ MIB II (RFC 1213)
- ▶ Ethernet MIB (RFC 1643)
- ▶ Bridge MIB (RFC 1493)

SNMP Versions 1 and 2

The SNMP configuration requires the definition of a community string to allow SNMP polling. The SNMP configuration is shown in Example 4-6, and includes basic SNMP system information.

Example 4-6 SNMP Versions 1 and 2 configuration

```
5000V(config)# snmp-server read-community myReadCommunity
5000V(config)# snmp-server write-community myWriteCommunity
5000V(config)# snmp-server contact "Network Operations Ext.555"
5000V(config)# snmp-server location "DC1 ESX Cluster 1"
5000V(config)# snmp-server name "DVS-5000V-01"
```

SNMP trap support: Configuration of SNMP trap hosts is documented in the *IBM System Networking Distributed Switch 5000V User Guide* (<http://www.ibm.com/support/docview.wss?uid=isg3T7000628>), but is not supported. The **snmp-server host** command is not implemented. The workaround is to use the SNMPv3 engine and configuration to send SNMPv2 traps, as shown in Example 4-9 on page 37.

SNMP Version 3

SNMP Version 3 provides improved authentication and message security in addition to granular access control to the MIB tree. These SNMPv3 features are referred to as the User-based Security Model (USM) and View-Based Access Control Model (VACM).

Implementing SNMP Version 3 view-based access control on the DVS 5000V requires that you complete the following steps:

1. Define a SNMPv3 user.
2. Define a group and add users to that group.
3. Define SNMPv3 views to identify a subtree of the SNMP MIB (or use the built-in “iso” view for the whole tree).
4. Grant the group-specific (read/write/notify) access to a set of views.

Example 4-7 shows how to create a SNMPv3 user with full read/write access.

Example 4-7 SNMPv3 view-based access control configuration

```
5000V(config)# snmp-server user 10 name NMSuser
5000V(config)# snmp-server user 10 authentication-protocol md5
authentication-password
Changing authentication password; validation required:
Enter current admin password:
Enter new authentication password:
Reenter new authentication password:
New authentication password accepted.
5000V(config)# snmp-server user 10 privacy-protocol des privacy-password
Changing privacy password; validation required:
Enter current admin password:
Enter new privacy password:
Reenter new privacy password:
New privacy password accepted.
5000V(config)#
5000V(config)# snmp-server group 10 group-name myV3NMSservers
5000V(config)# snmp-server group 10 user-name NMSuser
5000V(config)# snmp-server group 10 group-name myV3NMSservers
5000V(config)# snmp-server group 10 user-name NMSuser
5000V(config)#
5000V(config)# snmp-server access 10 notify-view iso
5000V(config)# snmp-server access 10 read-view iso
5000V(config)# snmp-server access 10 write-view iso
5000V(config)# snmp-server access 10 notify-view iso
5000V(config)# snmp-server access 10 name myV3NMSservers
5000V(config)#
```

In Example 4-8 shows how to create a restricted view with the **snmp-server view** commands. The new view has access only to the interface table from the IF-MIB. Full access from the group is removed, with a restricted view with read-only access applied instead.

Example 4-8 SNMPv3 view configuration

```
5000V(config)# snmp-server view 10 name InterfaceOnly
5000V(config)# snmp-server view 10 tree 1.3.6.1.2.1.2.2
5000V(config)# snmp-server view 10 type included
5000V(config)#
5000V(config)# no snmp-server access 10
5000V(config)# snmp-server access 10 name myV3NMSservers
5000V(config)# snmp-server access 10 read-view InterfaceOnly
```

SNMP Version 3 traps

SNMPv3 trap configuration is similar to the view-based access control that is defined in “SNMP Version 3” on page 35. The SNMPv3 engine can also be used to generate Version 1 and 2 traps.

Example 4-9 on page 37 shows how to use the SNMPv3 engine to implement SNMPv2 traps. Traps are configured with an entry to the SNMP notify table with a name and tag. Traps are sent to targets that include the defined tag in their tag list.

Example 4-9 SNMPv3 configuration for SNMPv2 traps

```
5000V(config)# snmp-server target-address 10 name myNMSserver address
169.254.87.93
5000V(config)# snmp-server target-address 10 parameters-name v2param
5000V(config)# snmp-server target-address 10 taglist v2traptag
5000V(config)# snmp-server target-parameters 10 name v2param
5000V(config)# snmp-server target-parameters 10 security snmpv2
5000V(config)# snmp-server target-parameters 10 message snmpv2c
5000V(config)# snmp-server notify 11 name thisIsMyCommunityStr
5000V(config)# snmp-server notify 11 tag v2traptag
```

The next SNMPv3 configuration, which is shown in Example 4-10, shows the implementation of SNMPv3 traps with USM. The user that is created and tied to the target parameter configuration is used to identify the trap PDU.

Example 4-10 SNMPv3 trap configuration

```
5000V(config)# snmp-server user 12 name snmpv3User
5000V(config)# snmp-server user 12 authentication-protocol md5
authentication-password
Changing authentication password; validation required:
Enter current admin password:
Enter new authentication password:
Reenter new authentication password:
New authentication password accepted.
5000V(config)#
5000V(config)# snmp-server group 12 group-name snmpv3TrapGp
5000V(config)# snmp-server group 12 security usm
5000V(config)# snmp-server group 12 user-name snmpv3User
5000V(config)# snmp-server access 12 name snmpv3TrapGp
5000V(config)# snmp-server access 12 level authnopriv
5000V(config)# snmp-server access 12 notify-view iso
5000V(config)# snmp-server access 12 security usm
5000V(config)# snmp-server target-address 12 name myV3TrapServer address
169.254.87.93
5000V(config)# snmp-server target-parameters 12 name v3trap
5000V(config)# snmp-server target-parameters 12 level authnopriv
5000V(config)# snmp-server target-parameters 12 message snmpv3
5000V(config)# snmp-server target-parameters 12 security usm
5000V(config)# snmp-server target-parameters 12 user-name snmpv3User
5000V(config)# snmp-server target-address 12 taglist v3trapTag
5000V(config)# snmp-server target-address 12 parameters-name v3trap
5000V(config)# snmp-server notify 12 name notifyV3Trap
5000V(config)# snmp-server notify 12 tag v3trapTag
```

Figure 4-1 shows the SNMP trap that is generated by Example 4-10 on page 37.

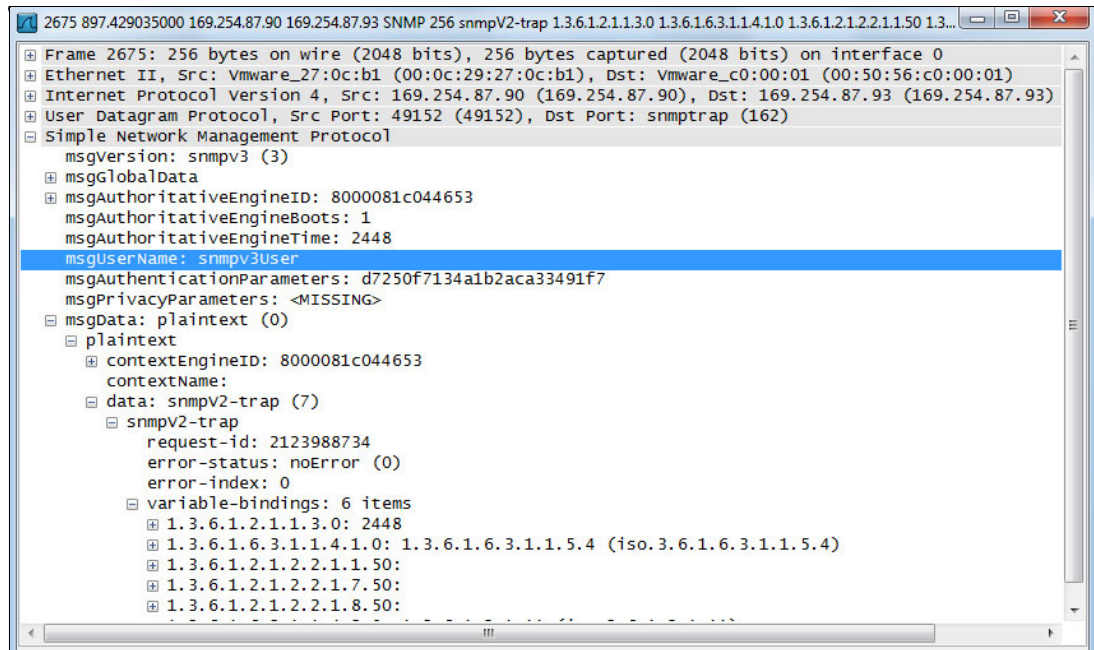


Figure 4-1 Packet capture to show an SNMPv3 trap

4.2.2 System time

System time is configured by running the **timezone**, **date**, and **time** commands, as shown in Example 4-11.

Example 4-11 Configuring the system time

```
5000V(config)# system timezone
Please identify a location so that time zone rules can be set correctly.
Please select a continent or ocean.
 1) Africa
 2) Americas
 3) Antarctica
 4) Arctic Ocean
 5) Asia
 6) Atlantic Ocean
 7) Australia
 8) Europe
 9) Indian Ocean
10) Pacific Ocean
11) None - disable timezone setting
#Enter zone: 7
Please select one of the following time zone regions.
 1) Lord Howe Island          6) Queensland - most locations
 2) Tasmania                  7) Queensland - Holiday Islands
 3) Victoria                  8) South Australia
 4) New South Wales - most locations 9) Northern Territory
 5) New South Wales - Yancowinna    10) Western Australia
#Enter zone: 4
Time zone set to 247 - Australia/NSW Most (UTC +10:00)
5000V(config)#
```

```
5000V(config)# system time 21:20:35
5000V(config)# system date 2013 1 12
```

Time can be synchronized by using NTP, as shown in Example 4-12.

Example 4-12 Configuring NTP

```
5000V(config)# ntp primary-server 10.150.1.50
5000V(config)# ntp secondary-server 10.150.2.50
5000V(config)# ntp enable
```

4.2.3 Logging and syslog

The DVS 5000V can send a syslog to two hosts. The logging can be enabled for levels 0 - 7, where level 7 is the most verbose.

Logging can also be enabled and disabled for the specific features that are listed here:

- ▶ ALL
- ▶ Command-line interface
- ▶ Internet Protocol
- ▶ Network time protocol
- ▶ System
- ▶ VLAN
- ▶ Private VLAN
- ▶ Telnet
- ▶ Notice
- ▶ Config Save and Restore
- ▶ TFTP
- ▶ SYSLOG

Example 4-13 shows syslog configuration that enables logging for all of the features.

Example 4-13 Syslog configuration example

```
5000V(config)# logging host 1 address 169.254.87.93
5000V(config)# logging host 1 facility 2
5000V(config)# logging host 1 severity 7
5000V(config)# logging log all
```

Logging in to the console is enabled and disabled by running the following command:

```
5000V(config)# [no] logging console
```

4.3 sFlow

The DVS 5000V supports sFlow Version 5. sFlow is a sampling technology and protocol set that is used to provide visibility into network traffic. The sFlow agent that is implemented on the DVS 5000V can provide valuable information about the virtual network for traffic management, problem determination, or metering and billing.

sFlow provides both flow and counter samples:

- ▶ *Flow samples* are packet samples that are taken randomly at a rate of one every N packets. The sample copies the header sample (34 bytes) into the sFlow UDP PDU and forwards it to the collector. The aggregated samples provide a statistical representation of the network traffic.
- ▶ *Counter samples* are taken at defined time periods, and provide time series data for host, port, and VLAN statistics.

Header sample size: The sFlow header sample size of 34 bytes includes 14 bytes for Ethernet headers (source and destination MAC + type), plus the 20-byte IPv4 header. However, the sFlow PDU that is sent by the DVS 5000V has the header length set to 14 (MAC header only), so collectors might not capture the IP data.

The DVS 5000V can provide sampling at a global level or up to 31 sample groups. Sample groups have a separate sampling engine, and take samples for a set of specific ports and VLANs. Sample sets that are used for a custom group are excluded from the global set to avoid duplication.

4.3.1 Enabling sFlow traffic on ESXi

Before sFlow can be used, the sFlow data flow must be allowed to go through the ESXi host firewall. This setting is configured on each ESXi host by using rule set configuration files that are in the following directory:

```
/etc/vmware/firewall/
```

To enable sFlow traffic, log in to the ESXi server CLI and create a rule set file, as shown in Example 4-14. The command example creates the rule file with content that is shown between the EOF markers.

Example 4-14 Creating an ESXi firewall rule set file

```
cat > /etc/vmware/firewall/DVS5000V.xml << EOF
<ConfigRoot>
  <service>
    <id>ibm5000VsFlow</id>
    <rule id='0000'>
      <direction>outbound</direction>
      <protocol>udp</protocol>
      <porttype>dst</porttype>
      <port>6343</port>
    </rule>
    <enabled>true</enabled>
    <required>false</required>
  </service>
</ConfigRoot>
EOF
```

To activate the rule, use the ESXi CLI firewall refresh command, as shown in Example 4-15. The example also shows how to verify the policy with `list` commands.

Example 4-15 Using the ESC CLI to enable and verify a firewall rule set for sFlow

```
~ # esxcli network firewall refresh
~ # esxcli network firewall ruleset list | grep 5000V
```

```
ibm5000VsFlow          true
~ # esxcli network firewall ruleset rule list | grep 5000V
ibm5000VsFlow          Outbound  UDP      Dst          6343      6343
~ #
```

4.3.2 Configuring sFlow

sFlow can be configured at the global level or for specific ports and VLANs that are known as *sFlow groups*. sFlow groups can be configured with unique sample rates and collection intervals. Example 4-16 shows the configuration of global sFlow and a specific group for port 1.

Example 4-16 sFlow global and group configuration

```
5000V(config)# sflow
sFlow configuration
5000V(config-sflow)# agent-ip 169.254.87.90
5000V(config-sflow)# collector 192.168.87.1
5000V(config-sflow)# counter-poll 300
5000V(config-sflow)# sample-rate 128
5000V(config-sflow)# enable
sFlow enabled...
5000V(config-sflow)# group 10
sFlow port specific instance configuration: 10
5000V(config-sflow-group-10)# counter-poll 60
5000V(config-sflow-group-10)# sample-rate 5
5000V(config-sflow-group-10)# collector 192.168.87.1
5000V(config-sflow-group-10)# add port 1
5000V(config-sflow-group-10)# end
5000V#
```

When implementing sFlow for interfaces with a high data rate and load, use a higher sample packet period and set the sample rate to 256 or higher.

4.4 Port mirroring

The DVS 5000V supports port mirroring (Switch Port Analyzer (SPAN)) and Encapsulated Remote SPAN (ERSPAN) for monitoring network traffic. Port mirroring can be configured to copy traffic from one or more ports to a monitor port. The monitor port can then be used by software such as sniffers, traffic analyzers, or security devices. The monitor port must be on the same host as the mirrored ports. However, this restriction can be overcome with ERSPAN, which encapsulates the mirror traffic into IP and forwards it to a remote monitor tool.

4.4.1 Configuring port mirroring

In Example 4-17, all traffic (both) from ports 1 and 5 (mirroring ports) is sent to port 2 (the monitor port). Port mirroring must be enabled before configuring mirror ports.

Example 4-17 Configuring port mirroring

```
5000V(config)# port-mirroring enable
5000V(config)# port-mirroring monitor-port 2 mirroring-port 1 both
5000V(config)# port-mirroring monitor-port 2 mirroring-port 5 both
```

The status of port mirroring can be reviewed by running **show**, as shown in Example 4-18.

Example 4-18 The show port mirroring command

```
5000V# show port-mirroring

Port Monitoring : Enabled

Monitoring Ports      Mirrored Ports
-----
1                     none
2                     (1,BOTH) (5,BOTH)
3                     none
4                     none
5                     none
...
```

4.4.2 Configuring ERSPAN

ERSPAN is configured by completing three steps:

1. Create an ERSPAN source definition.
2. Define the ERSPAN flow (where to send the data from the source).
3. Enable ERSPAN.

The ERSPAN source defines a set of access ports or VLANs to be mirrored. Other source configuration options allow the selection of the following items:

- ▶ Direction: Traffic direction, whether in, out, or both.
- ▶ Traffic type: Unicast, multicast, broadcast, or all.
- ▶ Capture mode: L2 or L3. Only L2 is available.
- ▶ Priority: 0-7. The 802.1p priority is for ERSPAN traffic.
- ▶ VLAN ID: The VLAN ID that is applied to ERSPAN headers.

Example 4-19 shows the configuration of ERSPAN.

Example 4-19 ERSPAN configuration

```
5000V(config)# erspan
ERSPAN configuration
5000V(config-erspan)# source 10
ERSPAN source configuration:10
5000V(erspan-source-10)# direction both
5000V(erspan-source-10)# traffic all
5000V(erspan-source-10)# mode l3
L3 mode is not supported yet, skipping..
```

```

5000V(erspan-source-10)# mode 12
5000V(erspan-source-10)# add port 1
5000V(erspan-source-10)# exit
5000V(config-erspan)# flow 1 10 192.168.87.1
5000V(config-erspan)# enable
ERSPAN enabled...
5000V(config-erspan)#

```

4.4.3 Viewing ERSPAN in Wireshark

The Wireshark network protocol analyzer supports packet dissection of Cisco ERSPAN. However, because the DVS 5000V implementation does not have ERSPAN headers that are consistent with the Cisco protocol, the packet dissection fails, as shown in Figure 4-2.

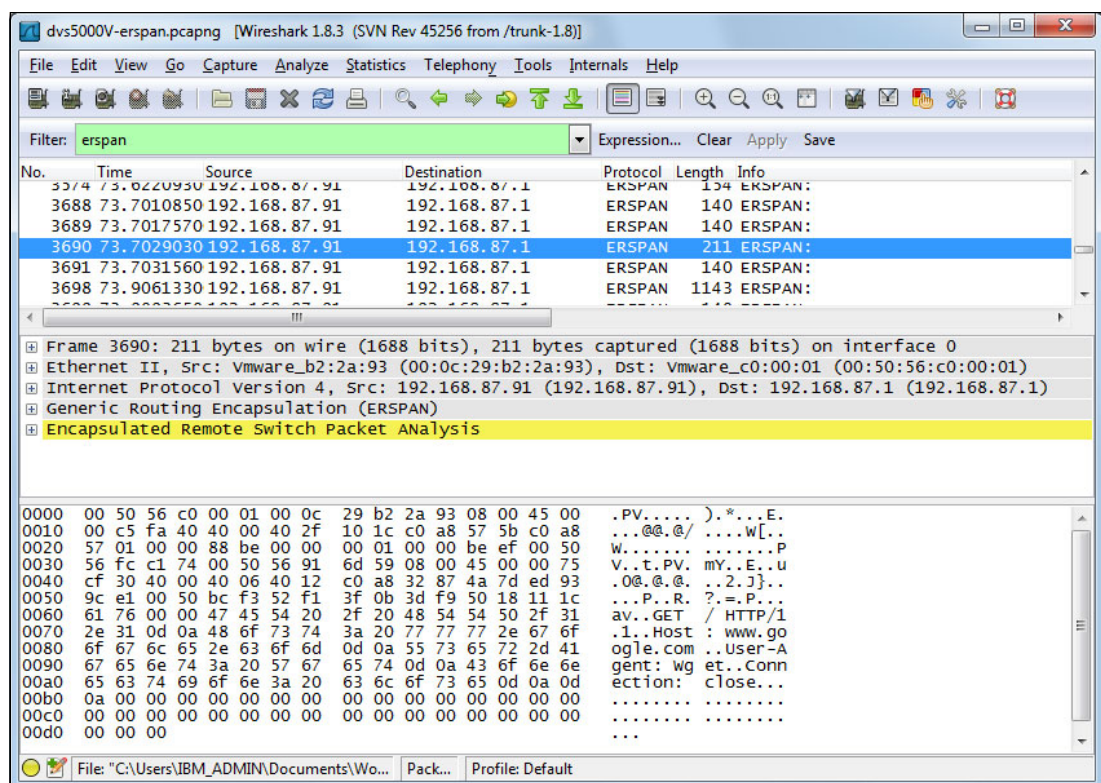


Figure 4-2 ERSPAN packet dissection in Wireshark

Note: The DVS 5000V sends ERSPAN packets with the version field set to 0, where the protocol dissector expects the known Cisco values of 1 or 2.

Changing the Wireshark ERSPAN protocol dissector to accept a version value of 0 allows full packet dissection in Wireshark. The remainder of this section explains how to patch Wireshark for this value. It also explains how to use ERSPAN on the DVS 5000V for traffic analysis.

The following steps provide the process for patching the Wireshark ERSPAN protocol dissector. The example process is on a Linux operating system. Some system knowledge is assumed for the task because users might need to install developer packages for compiling Wireshark and resolving dependencies.

1. Download the Wireshark source from SVN by running the following command:

```
svn checkout http://anonsvn.wireshark.org/wireshark/trunk wireshark
```
2. Create a patch file called ERSPAN-ver0.patch using the content from Example 4-20.

Example 4-20 Wireshark ERSPAN patch for Version 0 support

```
Index: wireshark/epan/dissectors/packet-cisco-erspan.c
=====
--- wireshark/epan/dissectors/packet-cisco-erspan.c(revision 48209)
+++ wireshark/epan/dissectors/packet-cisco-erspan.c(working copy)
@@ -150,7 +150,7 @@
     if (tree) {
         ti_ver = proto_tree_add_item(erspan_tree, hf_erspan_version, tvb, offset, 2,
                                     ENC_BIG_ENDIAN);
-       if ((version != 1) && (version != 2 )) {
+       if ((version != 0) && (version != 1) && (version != 2 )) {
             expert_add_info_format(pinfo, ti_ver, PI_UNDECODED, PI_WARN, "Unknown version,
please report or test to use fake ERSPAN preference");
             return;
         }
@@ -162,7 +162,7 @@
         ENC_BIG_ENDIAN);
         proto_tree_add_item(erspan_tree, hf_erspan_unknown2, tvb, offset, 2,
                             ENC_BIG_ENDIAN);
-       if (version == 1)
+       if ((version == 1) || (version == 0))
             proto_tree_add_item(erspan_tree, hf_erspan_direction, tvb,
                                 offset, 2, ENC_BIG_ENDIAN);
         else /* version = 2 */
```

3. Apply the patch to the Wireshark source by running the following command:

```
patch -p0 <ERSPAN-ver0.patch
```
4. Build Wireshark from the patched source by running the following commands:

```
cd wireshark
./autogen.sh
./configure
make
```

Dependencies: When you compile Wireshark, it has some dependencies. It is assumed that you can resolve these dependencies. However, for reference, the `bison`, `byacc`, `flex`, `gtk2-devel`, `libpcap-devel`, and `patch` packages were added to the Linux workstation that was used for the test environment for this book.

5. You can now run the patched Wireshark. If you have an existing installation of the Wireshark software, it might interfere with your build version. Set the environment variable `WIRESHARK_RUN_FROM_BUILD_DIRECTORY` when running Wireshark, as shown here:

```
WIRESHARK_RUN_FROM_BUILD_DIRECTORY=1 ./wireshark
```


Figure 4-3 shows the same sample ERSPAN packet that is shown in Figure 4-2 on page 43. However, this time, the dissection works all the way into the packet that is mirrored from the DVS 5000V.

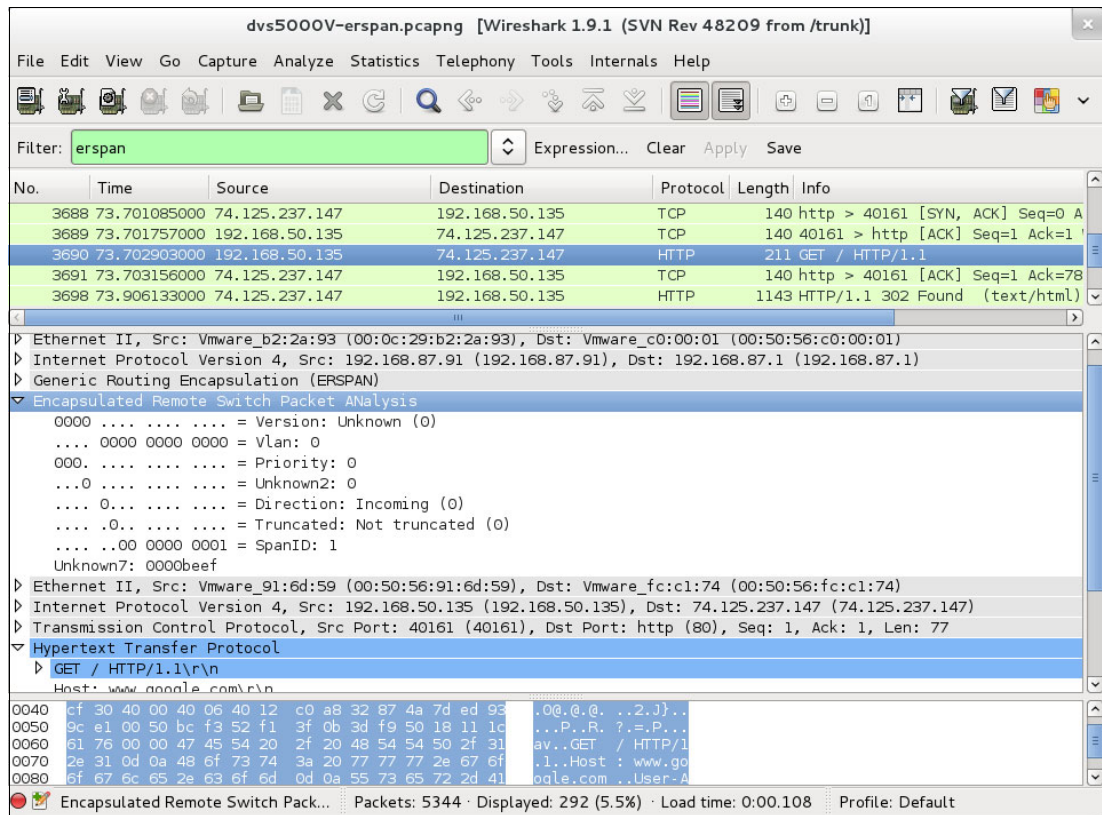


Figure 4-3 Working ERSPAN packet dissection



IBM Distributed Virtual Switch 5000V implementation

This chapter examines the implementation of the IBM Distributed Virtual Switch 5000V (DVS 5000V) in to an existing VMware environment. In this example, we use the base installation that is established in Chapter 3, “IBM Distributed Virtual Switch 5000V installation” on page 17 and look at how to configure uplinks, connect virtual machines (VMs), and use 802.1Qbg Edge Virtual Bridging (EVB). This chapter also explains how to use some more advanced features for the DVS 5000V.

This chapter covers the following topics:

- ▶ Configuring uplinks to hosts
- ▶ Connecting VMs to the DVS 5000V
- ▶ Virtualization-aware networking with EVB
- ▶ Advanced switch features

5.1 Uplink configuration

This section defines the terminology and options that are associated with uplink ports and gives an example of how to connect a host's uplink ports to the DVS 5000V.

Uplink ports are also referred to as the host's Physical NIC (or pNICs in a redundant configuration), which connect the DVS 5000V Host Module that was installed in Chapter 3, "IBM Distributed Virtual Switch 5000V installation" on page 17 to the external network through a physical Ethernet switch. Recall that during the host module installation, at least one pNIC must be associated with the DVS 5000V Controller for VM traffic to function.

To aid in multiple deployments across a series of hosts, uplink profiles can also be used for the configuration of uplinks, which includes options for specifying link aggregation modes.

5.1.1 Uplink profiles

Uplinks ports are not numbered or configured like the access (or VM-facing) ports on the DVS 5000V. Instead, all of the uplinks that are connected to the same vDS host module within an ESXi host are treated as a single link aggregation (LAG). To configure the uplinks, the administrator creates and applies uplink profiles. The uplink profiles contain settings such as the LAG type and MTU size.

A default uplink profile is defined when the global DVS 5000V vDS is created in the VMware datacenter. By default, the aggregation model is asymmetric and the MTU is 1,500. Example 5-1 shows that a profile can be defined by running `iswitch uprof` on the DVS 5000V Controller ISCLI interface.

Example 5-1 Configuring an uplink profile

```
5000V(config)# iswitch uprof <uprof name>
5000V(config-uprof)# ?
lagmode                Specify the LAG mode
laghash                Specify the LAG hash
mtu                    Configure the Maximum Transmission Unit (MTU)
lacp                   Configure the LACP
exit                   Exit the Current Mode
```

5.1.2 Link aggregation

The LAG mode defines how the uplinks are aggregated. The modes that are supported in the uplink profile on the DVS 5000V are asymmetric, static, and LACP.

Asymmetric LAG

The DVS 5000V uses asymmetric LAG mode by default in an EVB-supported configuration. This mode allows all uplinks that are connected to the same vDS host module within an ESXi host to form a logical LAG that is transparent to physical switches that are connected to the physical NICs. In this mode, the uplinks can connect to different physical switches, and their connected switch ports should not be configured to perform any type of link aggregation. This behavior emulates the default mode of operation within the VMWare standard switch, which is known as "Route based on the originating port ID" in the interface.

In Asymmetric LAG mode, the DVS 5000V ensures that packets with a source MAC address always use the same uplink port to avoid MAC address table problems on the connected switches. Duplicated and reflected packets are not delivered to VMs.

To set Asymmetric LAG mode, run the following command:

```
5000V(config-uprof)# lagmode asymmetric
```

Static LAG

In this mode, all uplinks ports that are connected to the same vDS host module within an ESXi host form a static 802.3ad-compatible link aggregation group. The physical switch ports that are connected the uplink ports' physical NICs must also be configured to form a corresponding static 802.3ad-compatible link aggregation group.

To set Static LAG mode, run the following command:

```
5000V(config-uprof)# lagmode static
```

LACP

In this mode, 802.3ad LACP runs on each of the uplinks. At most, one LACP LAG is formed for a vDS within an ESXi Host. When the uplinks are connected to different partners, only the uplinks that are connected to the partner on which the LACP negotiation is first completed are LACP-selected; others are LACP-unselected.

Important: EVB is not supported for uplinks using LACP mode.

When the aggregation is released, uplinks that are connected to another partner can be elected to use the aggregation. Example 5-2 shows how to implement LACP in the DVS 5000V.

Example 5-2 Configuring 802.3ad LACP on the DVS 5000V

```
5000V(config)# iswitch uprof Uplinks-LACP
5000V(config-uprof)# lacp ?
    mode                Configure LACP mode
    system-priority      Configure LACP system priority
    timeout              Configure LACP timeout
5000V(config-uprof)# lacp mode active
5000V(config-uprof)# lacp system-priority 32768
5000V(config-uprof)# lacp timeout long
5000V(config-uprof)# exit
```

After you create the uplink profile for an LACP configuration, then you assign a particular host's NIC to the specified uplink profile. To assign the host's NIC to an uplink profile, complete the following steps:

1. Access vCenter from vSphere Client and open the DVS 5000V by clicking **Home** → **Inventory** → **Networking**.

- Right-click the DVS 5000V instance and select **Manage Hosts**. Click **Next**, as shown in Figure 5-1.

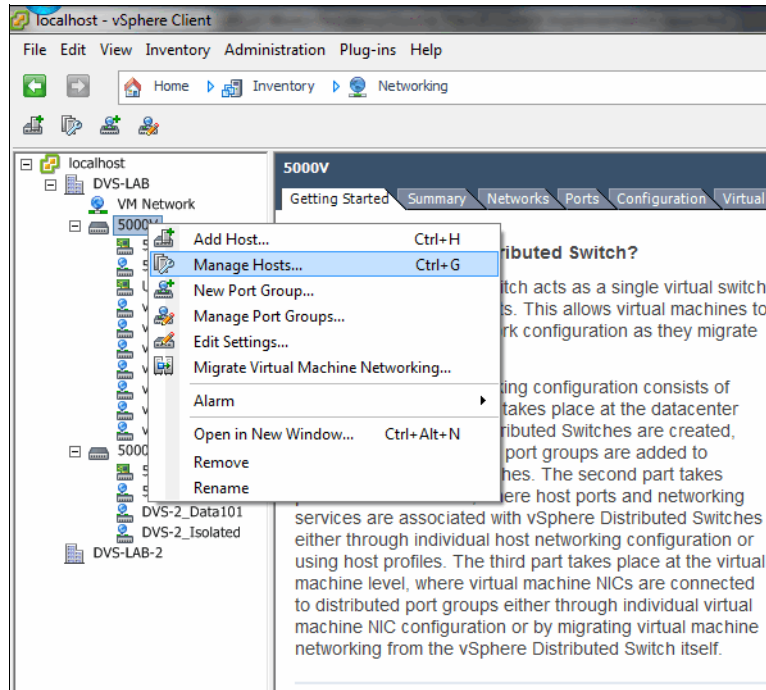


Figure 5-1 Manage Hosts menu from the DVS 5000V instance in vCenter

- Select the host whose NIC participates in the LACP configuration and click **Next**, as shown in Figure 5-2.

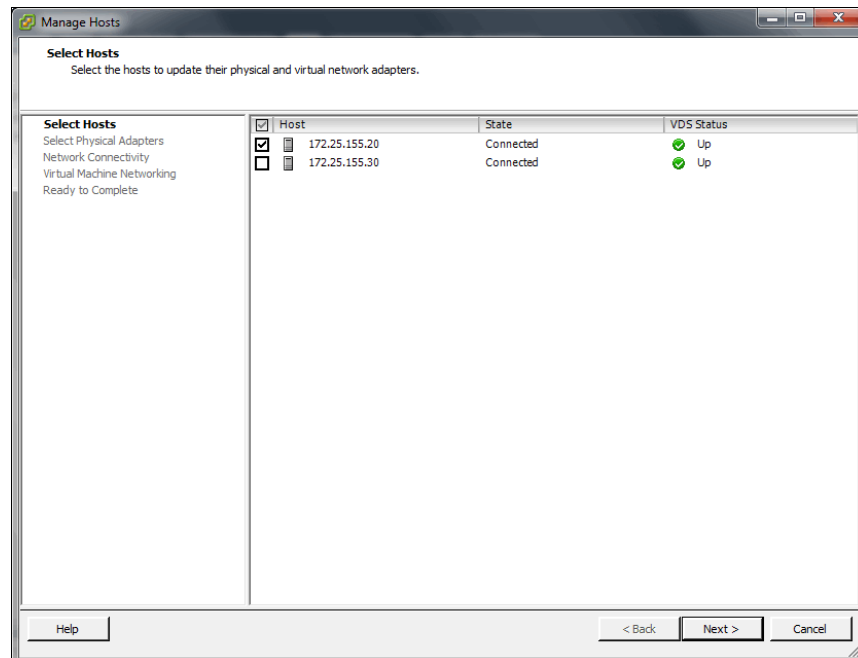


Figure 5-2 Selecting the host that participates in the LACP configuration

- Change the uplink port group of the host's NIC to **Uplink-LACP** profile and click **Next**, as shown in Figure 5-3 on page 51.

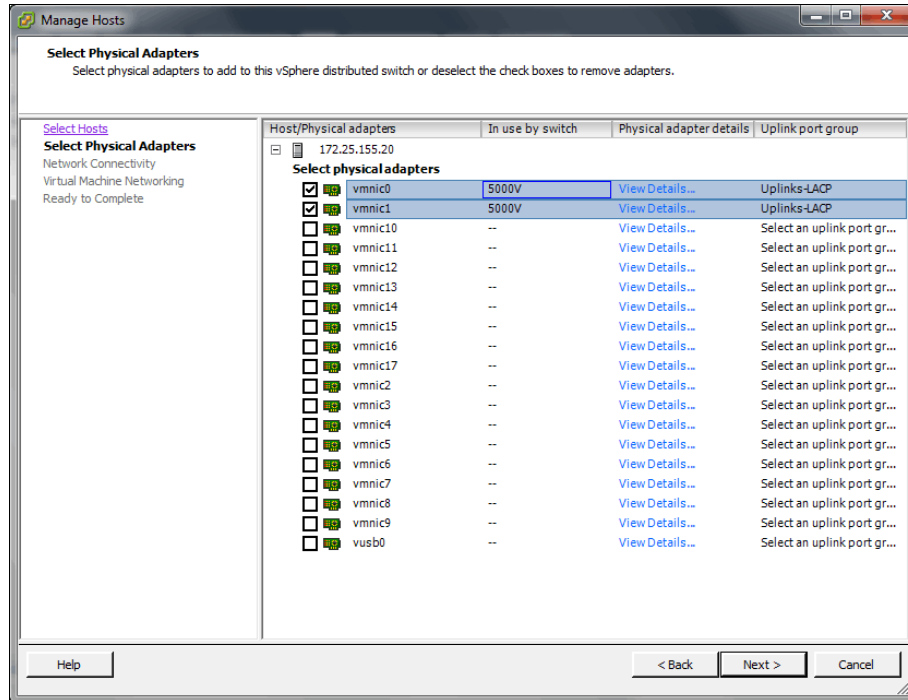


Figure 5-3 Selecting the NICs of the host and changing their uplink port group to the uplink profile that was created for the LACP configuration

5. If the management network and VM network are on the same VLAN or vNIC profile, skip setting the **Network Connectivity** and **Virtual Machine Networking** options. Review the configuration and click **Finish** to apply the uplink profile to the host's NICs, as shown in Figure 5-4.

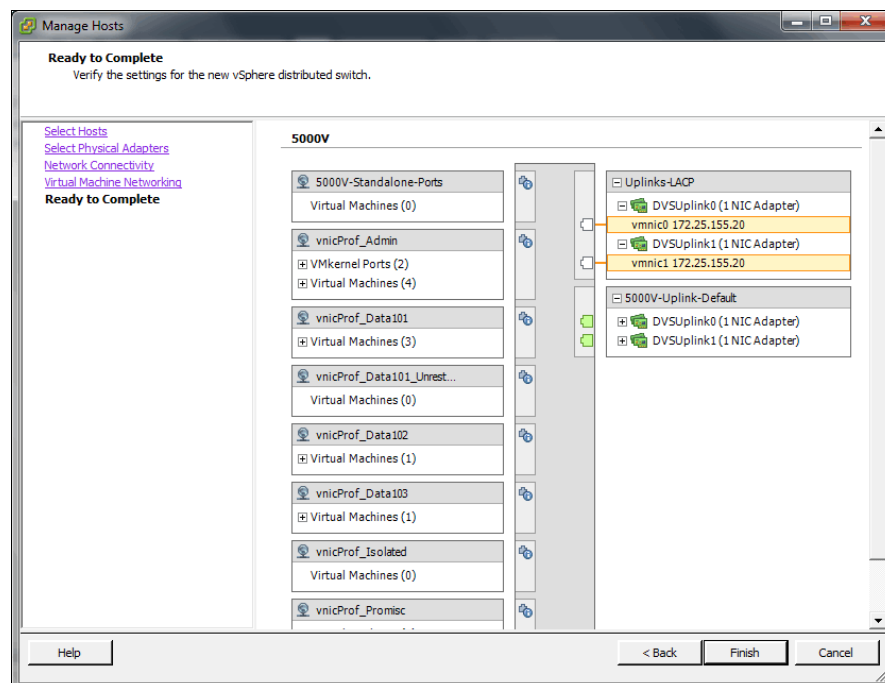


Figure 5-4 Reviewing the configuration for applying the uplink profile for LACP

The physical switch must also be configured for LACP. In this implementation scenario, we use the Flex System Fabric EN4093/R 10 Gb Scalable Switch. The implementation commands are shown in Example 5-3.

Example 5-3 EN4093/R LACP implementation

```
en4093flex_1(config)#interface port INTA2
en4093flex_1(config-if)#lacp mode active
en4093flex_1(config-if)#exit
en4093flex_1(config)#vlag adminkey 2 enable
en4093flex_1(config)#exit

en4093flex_2(config)#interface port INTA2
en4093flex_2(config-if)#lacp mode active
en4093flex_2(config-if)#exit
en4093flex_2(config)#vlag adminkey 2 enable
en4093flex_2(config)#exit
```

5.1.3 Link aggregation hash

Several different hashing algorithms can be deployed on a per-uplink port basis to determine how the physical switching infrastructure is configured upstream from the DVS 5000V Host Module.

Here are the supported hashing methods:

- ▶ VPORT: Transmit based on the hash value of vDS port ID that originates the packet. With this hash method, packets that originated from the same access port are forwarded over the same uplink port.
- ▶ SMAC: Transmit based on the hash value of the source MAC address. With this hash method, packets with the same source MAC address are forwarded over the same uplink port.
- ▶ DMAC: Transmit based on the hash value of the destination MAC address.
- ▶ DSMAC: Transmit based on the hash value of both the destination and source MAC address.
- ▶ SIP: Transmit based on the hash value of the source IP address.
- ▶ DIP: Transmit based on the hash value of the destination IP address.
- ▶ DSIP: Transmit based on the hash value of both the destination and source IP address.

Example 5-4 shows how to configure the link aggregation hash (LAG) hash on the DVS 5000V.

Example 5-4 LAG hash configuration on the DVS 5000V

```
5000V(config-uprof)# laghash ?
sip          LAG hash based on source IP address
smac        LAG hash based on source MAC address
vport       LAG hash based on source virtual port number
dip         LAG hash based on destination IP address
dsip        LAG hash base on both source and destination IP
            address
dmac        LAG hash base on destination MAC address
dsmac       LAG hash base on both source and destination
            MAC address
```

5.2 Connecting virtual machines

The DVS 5000V manages VM access ports through two configuration methods:

- ▶ vNIC profiles
- ▶ Stand-alone ports

vNIC profiles are introduced in 1.4.2, “Port Groups and vNIC profiles” on page 7. They are used as templates for common port configuration options, including VLAN, QoS, and access control. When a vNIC profile is created, an initial set of 20 ports in the DVS 5000V is assigned to the profile and inherits the profile attributes.

Stand-alone ports are configured individually like traditional physical switch ports. When the DVS 5000V is created, ports 1 - 100 are assigned as stand-alone ports. These ports are fixed and cannot be deleted.

The DVS 5000V supports up to 60,000 access ports. Ports can be added and deleted as stand-alone ports or to a vNIC profile as required.

Here is the typical workflow for managing VM network access:

1. The network administrator configures the network access profile for the new VM through the DVS 5000V ISCLI, IBM System Networking Switch Center, or through the IBM Flex System Manager™ as either a vNIC profile or a stand-alone port.
2. The profile name or port details are passed on to vCenter to be selected by the server administrator.
3. The server administrator selects the appropriate vNIC profile or stand-alone port for the VM's network connection.

5.2.1 Configuring vNIC profiles

To create vNIC profiles, run the following commands to enter vNIC profile configuration mode, where attributes can then be defined for the profile:

```
5000V(config)# iswitch vnicprof <profile name>
5000V(config-vnic-profile)#
```

Upon running the commands, the DVS 5000V Controller pushes the configuration to the vCenter server and creates a Distributed Port Group.

In Example 5-5, three vNIC profiles are created in the test environment. A new profile that is named `vnicProf_Admin` is created, and the controller automatically allocates 20 ports to it. A default VLAN ID (`pvid`) for the profile is assigned. All ports in the profile are members of the VLAN ID.

Example 5-5 Creating vNIC profiles

```
5000V# conf t
Enter configuration commands, one per line. End with Ctrl/Z.
5000V(config)# iswitch vnicprof vnicProf_Admin
Ports Allocated: 101-120
To see mapping to VDS port IDs, run show iswitch ports
5000V(config-vprof)# pvid 777
5000V(config-vprof)# exit
5000V(config)# iswitch vnicprof vnicProf_Data101
Ports Allocated: 121-140
To see mapping to VDS port IDs, run show iswitch ports
5000V(config-vprof)# pvid 101
5000V(config-vprof)# exit
5000V(config)# iswitch vnicprof vnicProf_Data102
Ports Allocated: 141-160
To see mapping to VDS port IDs, run show iswitch ports
5000V(config-vprof)# pvid 102
5000V(config-vprof)# exit
5000V(config)#
```

After creating these vNIC profiles, the corresponding Distributed Port Groups are visible in the vCenter, as shown in Figure 5-5 on page 55. The recent task history shows the configuration tasks that are initiated by the DVS 5000V Controller.

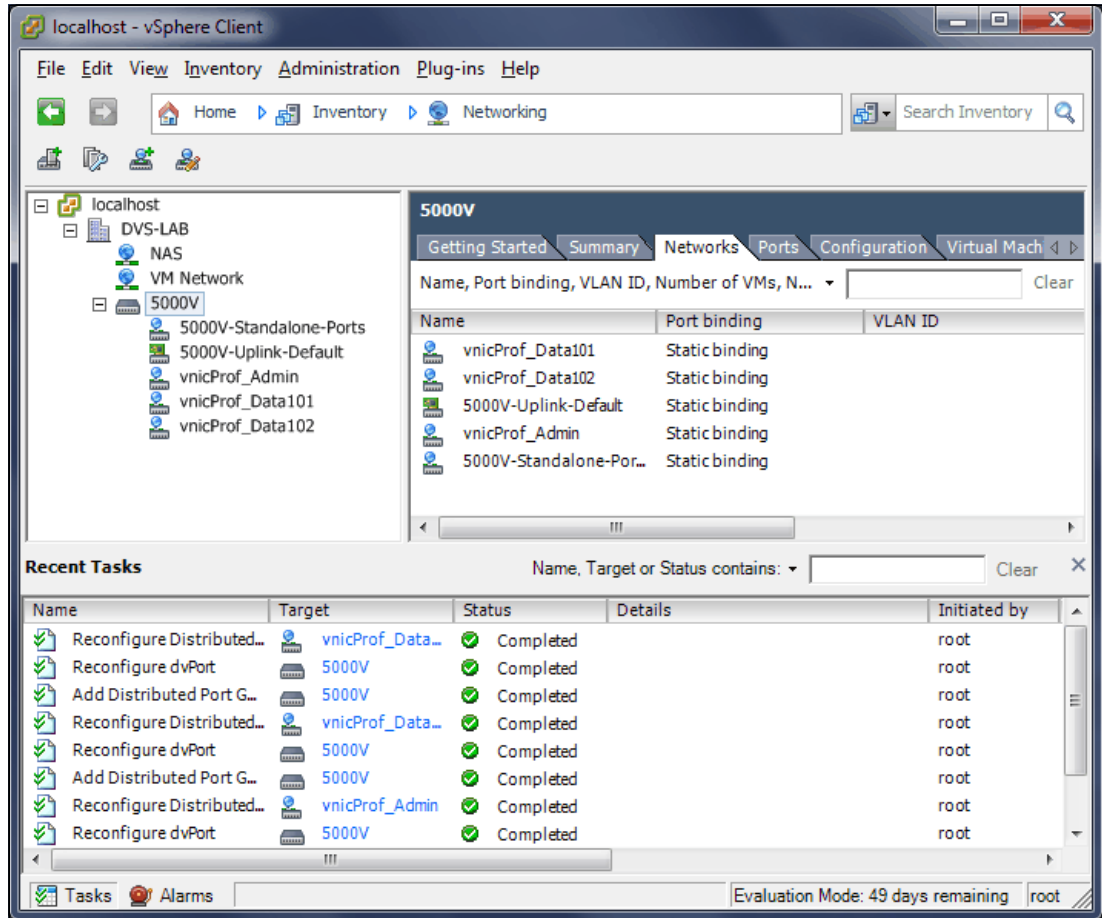


Figure 5-5 vNIC profiles in vCenter as Distributed Port Groups

After creating vNIC profiles, they can be applied to VMs, as described in 5.2.3, “Associating VMs with the DVS 5000V” on page 58.

vNIC profile properties

In Example 5-5 on page 54, a vNIC profile was created with the port VLAN ID (pvid) set to 101. Now, look at other properties that can be applied through a vNIC profile. The commands in Table 5-1 can be entered in vNIC profile configuration mode, which is indicated by the following prompt:

```
5000V(config-vprof)# ?
```

Table 5-1 vNIC profile configuration options

vNIC profile subcommand	Description	Example
addports delports	When the profile is created, 20 access ports are allocated to the profile. You can then add or remove ports in lots of 10.	addports 20 delports 101-110
pvid	The port VLAN ID for members of the profile	pvid 101
tagging	Enables (1) or disables (0) 802.1Q VLAN tagging on the port to the VM.	tagging 1

vNIC profile subcommand	Description	Example
tagpvid	Enables tagging of the pvid on a 802.1Q trunk. (Normally the pvid is untagged.)	tagpvid
vlanlist	Sets the list of allowed VLANs on the trunk connecting the VM. (The range is separated by a colon.)	vlanlist 101,102 vlanlist 101:103
addvlan remvlan	Adds or removes VLANs from the list that is initialized by the vlanlist command.	addvlan 103:110
dot1p	Sets the 802.1p default priority for untagged ingress packets. Priority for tagged packets is accepted as marked from the guest.	dot1p <0-7>
vepa	Enables Virtual Ethernet Port Aggregator (VEPA) mode for the VM port (see 5.3.4, “Applying a VSI Type with Virtual Ethernet Port Aggregator mode connectivity” on page 67).	vepa
vsitype	Sets the VSI type ID for the port, which is shared with the physical edge switch.	vsitype 1 version 1
designated-uplinks	Defines a preference to a particular set of uplink ports, or prevents traffic going to the uplink with the drop option. The uplink number is the trailing number of the uplink alias, that is, DVSUplink0 is uplink 0.	designated-uplinks 1-2 designated-uplinks drop
dps-disabled	Sets the default port state (dps) for a VM to be disabled. When a new VM is attached, its port is disabled in the DVS until the network administrator enables it by running the interface port configuration command ops-enabled .	dps-disabled
service-policy	Applies a QoS policy map to the interface (see 5.4.1, “Quality of service (QoS)” on page 72).	service-policy input 10 service-policy output 13
rate-limit	Applies a simple QoS rate limit in Kbps, up to 1000000 (1 Gbps). The direction that is given in the command is relative to the virtual switch interface.	rate-limit input 1024 rate-limit output 10240
access-list	Apply a MAC or IP access control list to the vNIC port profile. Access control is available only for ingress traffic. The direction is relative to the virtual switch interface (see 5.4.2, “Access Control Lists” on page 73).	access-list ip 129 in access-list mac 10 in

5.2.2 Configuring stand-alone ports and port-level configuration

Stand-alone ports are configured in interface configuration mode consistent with tasks on a traditional physical switch. The configuration options are similar to the options that are available in vNIC profiles, but with some additions and some variation in syntax. Additionally, a port that is allocated to a vNIC profile can be customized with a port-level configuration that is specific for a VM. The resulting configuration is the culmination of the vNIC profile and the port-level configuration, with the port level configuration taking precedence over any conflicting settings. Table 5-2 lists the port configuration options.

Table 5-2 Interface configuration commands

Interface configuration subcommands	Description	Example
pvid	The port VLAN ID.	pvid 101
tagging	Enables 802.1Q VLAN tagging on the port to the VM.	tagging
tag-pvid	Enables tagging of the pvid on a 802.1Q trunk. (Normally the pvid is untagged.)	tag-pvid
dot1p	Sets the 802.1p default priority for untagged ingress packets. Priority for tagged packets is accepted as marked from the guest.	dot1p <0-7>
vepa	Enables VEPA mode for the VM port (see 5.3.4, “Applying a VSI Type with Virtual Ethernet Port Aggregator mode connectivity” on page 67).	vepa
vsitype	Sets the VSI type ID for the port, which is shared with the physical edge switch. (For more information, see 5.3, “Virtualization Aware Networking with Edge Virtual Bridging” on page 62.)	vsitype 1 version 1
designated-uplinks	Used to define a preference to a particular set of uplink ports or prevent traffic from going to the uplink with the drop option. The uplink number is the trailing number of the uplink alias; that is, DVSuplink0 is uplink 0.	designated-uplinks 1-2 designated-uplinks drop
dps-disabled	Sets the default port state (dps) for a VM to be disabled. When a new VM is attached, the port is disabled in the DVS until the network administrator enables it through the interface port configuration command ops-enabled .	dps-disabled
service-policy	Applies a QoS policy map to the interface (see 5.4.1, “Quality of service (QoS)” on page 72).	service-policy input 10 service-policy output 13

Interface configuration subcommands	Description	Example
rate-limit	Applies a simple QoS rate limit in Kbps, up to 1000000 (1 Gbps). The direction that is given in the command is relative to the virtual switch interface.	rate-limit input 1024 rate-limit output 10240
ip access-group mac access-group	Applies a MAC or IP access control list to the port. Access control is available only for ingress traffic. The direction is relative to the virtual switch interface. (See 5.4.2, "Access Control Lists" on page 73.)	ip access-group 129 in mac access-group 10 in
auto	Sets auto negotiation.	auto
flowcontrol	Enables flow control for send, receive, or both.	flowcontrol both
link-logging	Enables logging for link up/down.	link-logging
name	Sets the interface name.	name "Web server 1"
shutdown	Administratively disables the port.	shutdown no shutdown
ops-enabled	Enables a disabled port. Used to enable a new port that is set with default port state disable (dps-disable).	ops-enabled

Example 5-6 shows configuring the VLAN ID and rate limit for a stand-alone port.

Example 5-6 VLAN ID and rate-limit configuration

```
5000V# conf t
Enter configuration commands, one per line. End with Ctrl/Z.
5000V(config)# interface port 1
5000V(config-if)# pvid 101
5000V(config-if)# rate-limit input 20480
5000V(config-if)# end
5000V#
```

5.2.3 Associating VMs with the DVS 5000V

You can now apply the DVS 5000V vNIC profile or stand-alone port to a VM network interface in the same way as any distributed or standard port group, through the Virtual Machine Properties window. Figure 5-6 on page 59 shows the configured vNIC profiles that are available in the Virtual Machine Properties dialog box.

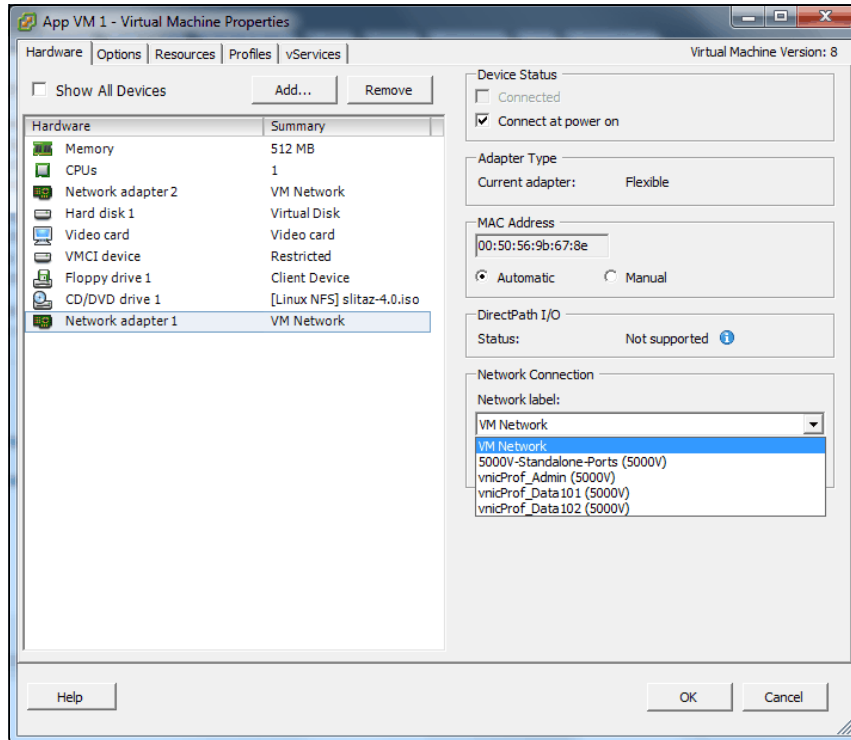


Figure 5-6 Applying a vNIC profile to a VM

After applying the vNIC profile to a VM's network adapter and clicking **OK**, the VM is dynamically assigned a port from the ports that are available for the selected profile. The port allocation can be viewed from the vCenter or from the DVS 5000V Controller VM.

Port numbering: The port numbering in the DVS 5000V and VMware vCenter are indexed differently. The DVS 5000V port numbering starts from 1, and in vCenter ports start at 0. Additionally, when profiles are added and removed, the number mapping might differ greatly. Always check the mapping between vCenter and the DVS 5000V.

To view a mapping of ports and VMs in the DVS 5000V, run the following **show** command:

```
5000V# show iswitch ports
```

Example 5-7 shows a sample output from this command. The output is trimmed to display more illustrative output.

Example 5-7 Viewing the port to VM mapping (viewed through the show iswitch ports command)

```
5000V# show iswitch ports
Port vDs-Port Profile    Connectee    Host          Mac-Address    Status
=====
   1    0 STANDALONE
   2    1 STANDALONE
   3    2 STANDALONE
[----- output cut -----]
  99   98 STANDALONE
 100   99 STANDALONE
 101  100 vnicProf_A..
 102  101 vnicProf_A.. vmk0          172.25.155.30  34:40:b5:be:7a:f8 Enabled
 103  102 vnicProf_A.. AAA & Radi.. 172.25.155.30  00:50:56:9b:2d:69 Enabled
```

104	103	vnicProf_A..					
105	104	vnicProf_A..	AP Linux V..	172.25.155.30	00:50:56:9b:97:c9	Enabled	
106	105	vnicProf_A..	AP Linux V..	172.25.155.30	00:50:56:9b:3e:06	Enabled	
107	106	vnicProf_A..	App VM 1	172.25.155.30	00:50:56:9b:67:8e	Enabled	
108	107	vnicProf_A..					

To view the port allocation information in vCenter, you can search for a specific VM through the Virtual Machine Properties dialog box that is shown in Figure 5-7, or for all VMs, as shown in Figure 5-8 on page 61.

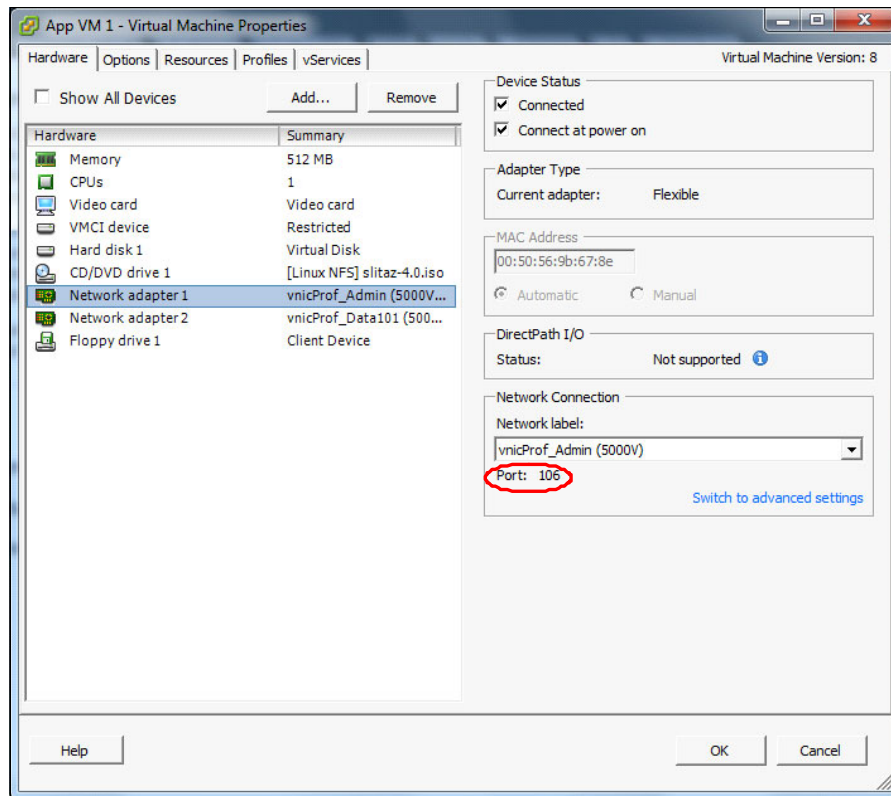


Figure 5-7 Virtual machines port allocation in the Virtual Machine Properties dialog box

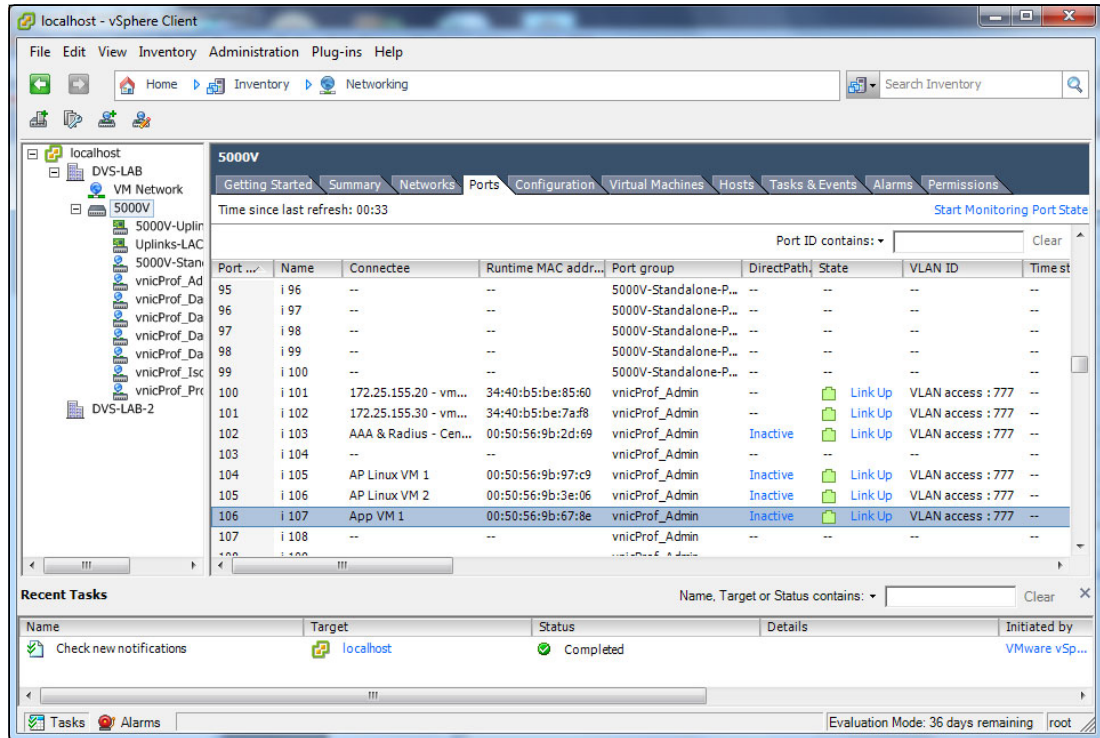


Figure 5-8 Viewing the VM port allocation for the DVS 5000V in vCenter

Connecting specific ports to virtual machines

When specific ports are customized for a VM, for example, when using stand-alone ports, the virtualization administrator can assign a specific port to a VM. This is performed in the Virtual Machine Properties window by clicking the **Switch to advanced settings toggle** link and then specifying the wanted port in the Port ID window.

Remember: The port number in vCenter is different from the DVS 5000V port number.

Figure 5-9 shows the allocation of vCenter distributed port 0 (which is port 1 on the DVS 5000V) to a VM network adapter.

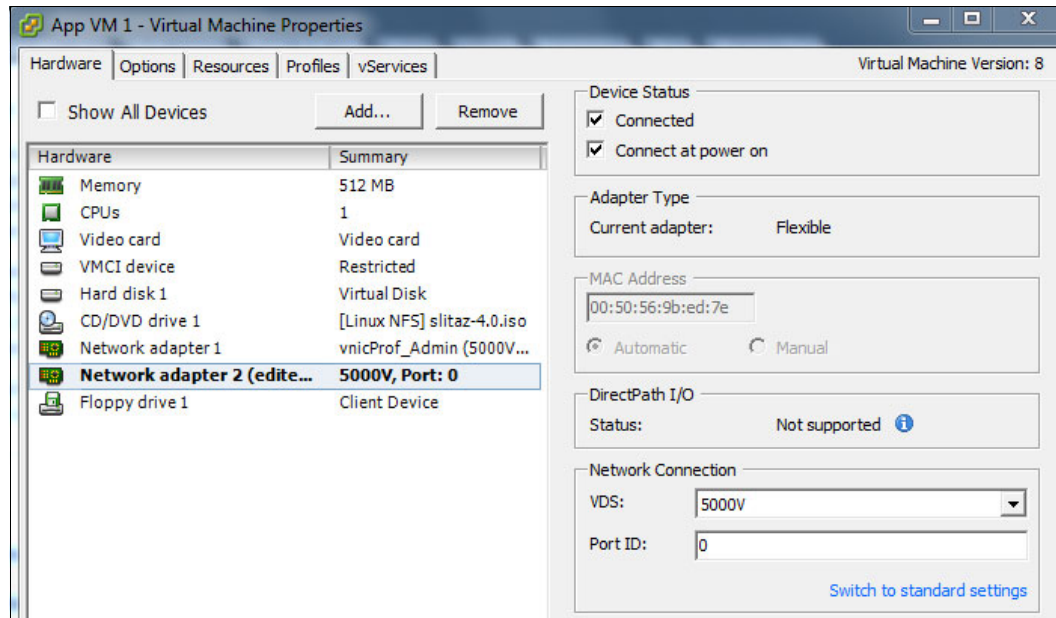


Figure 5-9 Assigning a specific virtual port to a VM

5.3 Virtualization Aware Networking with Edge Virtual Bridging

The 802.1Qbg/EVB is an emerging IEEE standard for allowing networks to become VM-aware. EVB bridges the gap between physical and virtual network resources. The IEEE 802.1Qbg specification simplifies network management by providing a standards-based protocol that defines how virtual Ethernet bridges exchange configuration information. In EVB environments, virtual NIC (vNIC) configuration information is available to EVB devices. This information is not available to an 802.1Q bridge.

The EVB features on the DVS 5000V are compliant with the IEEE 802.1Qbg Authors Group Draft 0.2. For a list of documents about this feature, see the following website:

<http://www.ieee802.org/1/pages/802.1bg.html>

The 5000V implementation of EVB supports the following protocols:

- ▶ Virtual Ethernet Bridging (VEB) and Virtual Ethernet Port Aggregator (VEPA): VEB and VEPA are mechanisms for switching between VMs on the same hypervisor. VEB enables switching with the server, either in the software (virtual switch), or in the hardware (using single root I/O virtualization capable NICs). VEPA requires the edge switch to support “Reflective Relay”, which is an operation where the switch forwards a frame back to the port on which it arrived if the destination MAC address is on the same port.
- ▶ Edge Control Protocol (ECP): ECP provides reliable delivery of service dispatcher units (SDUs) between the station and bridge, and between the port extender and bridge.
- ▶ Virtual Station Interface (VSI) Discovery and Configuration Protocol (VDP): VDP allows hypervisors to advertise VSIs to the physical network. This protocol also allows centralized configuration of network policies that persist with the VM, independent of its location.
- ▶ EVB Type-Length-Value (TLV): EVB TLV is a component of Link Layer Discovery protocol (LLDP)-based TLV that is used to discover and configure VEPA, ECP, and VDP.

For more information about EVB, see the *IBM System Networking Distributed Virtual Switch 5000V User Guide* at the following website:

<http://www.ibm.com/support/docview.wss?uid=isg3T7000628>

5.3.1 Prerequisites and limitations

Before implementing EVB with the DVS 5000V, consider the following prerequisites and limitations:

- ▶ The uplinks profile must use asymmetric LAG. The 802.3ad modes (static or LACP) are not supported.
- ▶ VMReady cannot coexist with 802.1Qbg EVB and must be disabled.

SR-IOV for S-channel, vNIC, FCoE, and Stacking are not supported.

5.3.2 Implementation overview

The proceeding sections describe and give examples for the implementation of EVB with the DVS 5000V and IBM System Networking switches.

Implementation of EVB requires completing of the following tasks:

1. Create a VSI Type database. This task can be accomplished by the DVS 5000V in-built database, IBM System Networking Switch Center (SNSC), or the IBM Flex System Manager.
2. Configure the DVS 5000V to subscribe to the VSI Type database.
3. Configure the DVS 5000V to use the VSI Types by linking them to vNIC profiles or access ports and enabling VEPA.
4. Configure the physical edged switch for EVB and subscribe to the VSI Type database.
5. Configure VM properties to use the ports or port profiles that use EVB and VSI Types.
6. Verify the configuration.

5.3.3 Configuring the VSI Type database

The VSI Type database is a key component to the EVB system, providing a central repository for VSI types that can be shared between virtual and physical infrastructure components. The DVS 5000V provides a built-in VSI Type database that is configured through the ISCLI using the vsiman (VSI Manager) configuration mode.

Figure 5-10 gives a procedural overview of the entire EVB process, and how the VSI database fits into the overall solution.

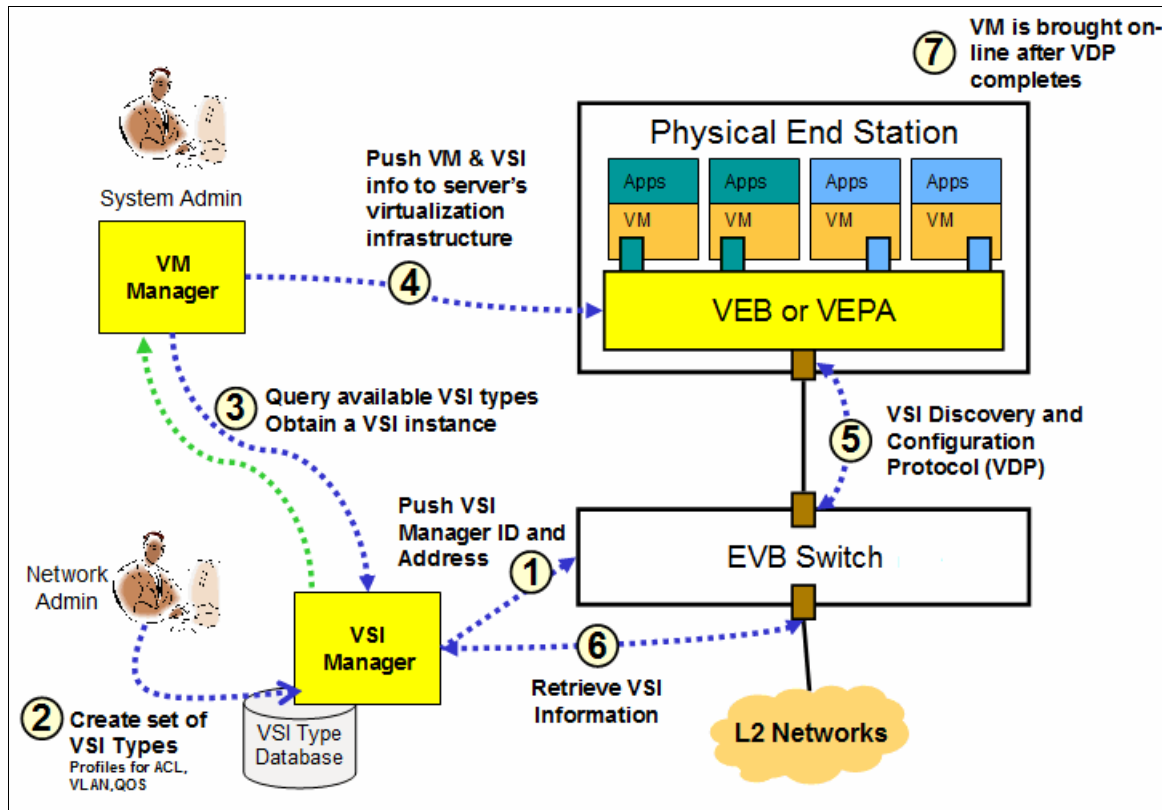


Figure 5-10 EVB overall workflow diagram

Enabling VSI Manager: The built-in VSI Type database

To enable the built-in VSI Type database, enter VSI Manager configuration mode and assign a manager ID. You can also optionally set the TCP port on which the VSI Type database listens. Example 5-8 shows how to enable the VSI type database in the DVS 5000V Controller.

Example 5-8 Enabling the VSI type database

```
5000V# conf t
Enter configuration commands, one per line. End with Ctrl/Z.
5000V(config)# vsiman
5000V(config-vsiman)#
5000V(config-vsiman)# managerid 1
5000V(config-vsiman)# port 44444
5000V(config-vsiman)# end
5000V#
```

Managing VSI Types in VSI Manager

VSI Type definitions are added to the DVS 5000V VSI Manager through the ISCLI. The IBM System Networking implementation of EVB offers a set of common network attributes that can be defined by using a VSI Type. A VSI Type is identified by the unique combination of the assigned ID and a version number.

Example 5-9 shows the creation of a VSI Type with the following attributes:

- ▶ ID: 5, Version: 1.
- ▶ Set the name to “CBR10Mbps-ACL192”.
- ▶ Allow VMs in VLANs 101 - 103.
- ▶ Apply bandwidth rate limiting at 10 Mbps (10240 Kbps) (transmit and receive).
- ▶ Apply access control list number 192.

Example 5-9 Creating a VSI Type in the DVS 5000V VSI Manager

```
5000V# conf t
Enter configuration commands, one per line. End with Ctrl/Z.
5000V(config)# vsiman
5000V(config-vsiman)# typeid 5 version 1
5000V(typeid-version)# name "CBR10Mbps-ACL192"
5000V(typeid-version)# vlans 101-103
5000V(typeid-version)# qos tx-cbr 10240
5000V(typeid-version)# qos rx-cbr 10240
5000V(typeid-version)# acls 192
5000V(typeid-version)# end
5000V#
```

The configured VSI Type can be reviewed through the running configuration or by running the following command:

```
5000V# show vsitypeid
```

Most configuration options for VSI Types are illustrated in Example 5-9; however, a complete list of VSI Type configuration commands is given in Table 5-3.

Table 5-3 VSI Type configuration commands

VSI Type command	Description	Example
acls	Access control lists that are applied to the VM's TX traffic.	acls 192-193
action-setpriority	Use an ACL to set the 802.1p priority. Any matches receive the defined priority.	action-setpriority acl 201 priority 6
name	Set the name of the VSI Type.	name "CBR10Mbps-ACL192"
qos rx-burst qos rx-cbr qos tx-burst qos tx-cbr	Set the constant bit rate (CBR) limit and burst size for the VM. TX/RX is relative from the VM. CBR limits are rates in Kbps up to 1 Gbps and burst sizes are in kilobits. Values must be multiples of 64.	qos rx-burst 2048 qos rx-cbr 10240 qos tx-cbr 10240
vlans	Set the permitted VLANs for the VSI Type. This command does not set the port VLAN ID; it defines the allowed VLANs. If a VM tries to associate with the VSI Type and its PVID is not in the permitted list, the association is rejected.	vlans 101-103

Subscribing to the VSI Type database

It is important to recognize that the VSI Type data stores that are used by VSI Manager and DVS 5000V act as different entities. So, whether the local VSI Type database is used or an external one is used, the DVS 5000V iSwitch must be configured to subscribe to the VSI Type database. Figure 5-10 on page 64 shows a pictorial representation of the VSI database, and how it fits into the overall process.

The DVS 5000V periodically downloads the database in XML format. The VSI Type database can be the local instance or one of the IBM management systems that are listed here:

- ▶ IBM System Networking Switch Center (SNSC)

SNSC provides the benefit of a more centralized configuration point for VSI Types that can be shared across multiple physical and virtual switches in a scalable way. Further information about SNSC can be found at the following website:

<http://www.ibm.com/systems/networking/software/snsc/>

- ▶ IBM Flex System Manager

The IBM Flex System Manager provides a central management system for managing multiple IBM Flex System Enterprise chassis. It includes a built-in VSI Type database, a configuration interface, and integrates EVB provisioning on switches. Further information on IBM Flex System Manager can be found at the following website:

http://publib.boulder.ibm.com/infocenter/flexsys/information/topic/com.ibm.acc.8731.doc/product_page.html

The configuration command on the DVS 5000V is consistent, regardless of the VSI Type database that is used. Only the host and port details vary. Example 5-10 shows the configuration of the DVS 5000V to use the local VSI Type database. The port number matches the port that was used when you configured the VSI manager in Example 5-9 on page 65.

Example 5-10 Configuring the DVS 5000V to subscribe to a VSI Type database

```
5000V# conf t
Enter configuration commands, one per line. End with Ctrl/Z.
5000V(config)# iswitch myvsidb 127.0.0.1 port 44444 path /vsitypes/
5000V(config)# end
5000V#
```

Here is the syntax for the **myvsidb** configuration command:

```
iswitch myvsidb <IP address> [path <VSI URI path>] [port <1 - 65535>]
```

The VSI Type database can be verified by running the **show iswitch** commands, as shown in Example 5-11.

Note: The **show iswitch myvsidb** command can be run only from configuration mode.

Example 5-11 Verifying the VSI Type database download

```
5000V(config)# show iswitch myvsidb
VSI Data Base Address: 127.0.0.1
VSI Data Base Port    : 44444
VSI Data Base Path    : /vsitypes/
INDEX : 1
-----
Name : CBR10Mbps-ACL192
```

```
Type ID: 5
Version: 1
Manager ID: 1
VLAN : 101,102,103
TxRate: 10240
TxBurst: 0
RxRate: 10240
RxBurst: 0
```

```
ACL Index: 1
```

```
-----
SRC MAC: 00:00:00:00:00:00
SRC MAC MASK: 00:00:00:00:00:00
DST MAC: 00:00:00:00:00:00
DST MAC MASK: 00:00:00:00:00:00
VLAN: 0
VLAN MASK: 0 (0x0)
Ether Type: 0x0800 (IPv4)
SRC IP: 0.0.0.0
SRC IP MASK: 0.0.0.0
DST IP: 0.0.0.0
DST IP MASK: 0.0.0.0
TOS: 0 (0x00)
IP Protocol: 6 (TCP)
TCP Flags: 0 (0x00)
SRC Port: 22, Mask (0xffff)
DST Port: 0, Mask (0xffff)
ACL Action: deny
```

```
5000V(config)#
```

5.3.4 Applying a VSI Type with Virtual Ethernet Port Aggregator mode connectivity

VEPA is a network switching solution for virtualization environments where all network traffic from VMs is forwarded to the physical network for switching, filtering, and metering.

The next step to complete the DVS 5000V configuration for EVB with VEPA is to apply the VSI Type and enable VEPA for the access ports. Like other port configurations, this can be done either with a vNIC profile or for a specific port configuration.

Example 5-12 shows configuring VEPA and the VSI Type for a vNIC profile. In this example, we use the VSI Type 5 version 1, which is created in Example 5-9 on page 65.

Example 5-12 Configuring VSI Type and VEPA for a vNIC profile

```
5000V# conf t
Enter configuration commands, one per line. End with Ctrl/Z.
5000V(config)# iswitch vnicprof d101-vepa-vsi5v1
Ports Allocated: 241-260
To see mapping to VDS port IDs, run show iswitch ports
5000V(config-vprof)# pvid 101
5000V(config-vprof)# vsitype 5 version 1
5000V(config-vprof)# veapa
```

```
5000V(config-vprof)# end
5000V#
```

Example 5-13 shows configuring VEPA and VSI Type for a stand-alone port.

Example 5-13 Configuring VSI Type and VEPA for a stand-alone port

```
5000V# conf t
Enter configuration commands, one per line. End with Ctrl/Z.
5000V(config)# interface port 2
5000V(config-if)# pvid 101
Port 2 is an UNTAGGED Port and its PVID is changed to 101
5000V(config-if)# vsitype 5 version 1
5000V(config-if)# veпа
5000V(config-if)# end
5000V#
```

This example completes the DVS 5000V configuration for EVB and VEPA. The physical network equipment must also be configured to support EVB and interact with the DVS 5000V.

5.3.5 Configuring the physical network for EVB

IBM System Networking RackSwitches, Flex System Fabric Switches, and the Virtual Fabric 10Gb Switch Module for IBM BladeCenter® all support EVB based on the 802.1Qbg standard. The lab environment uses an IBM Flex System Enterprise Chassis with EN4093/R 10Gb Scalable Switches, so the examples reflect this product. IBM RackSwitch configuration for EVB is consistent with the IBM EN4093 switch. However, you should see the relevant product documentation for more details.

The physical switch must be running a version of code that supports EVB. For the IBM EN4093, this was introduced in Version 7.5, and the lab environment for this book uses Version 7.5.1 of the IBM N/OS code.

EVB configuration on the physical switch can be summarized by the following tasks:

1. Disable VMReady. VMReady can be considered a pre-standard implementation of EVB. It must be disabled when using 802.1Qbg EVB.
2. Subscribe to the VSI Type database. As we did for the DVS 5000V, the physical switch is configured to download periodically the VSI Type database in XML format.
3. Enable Link Layer Discovery Protocol (LLDP). LLDP is used to carry EVB protocols between the DVS 5000V and the physical switch.
4. Enable EVB and related protocols for interfaces by using an EVB profile.
5. Repeat the procedure for all switches connecting uplinks from the DVS 5000V so that when a VM is migrated between hosts, its network profile (and associated configuration) moves with it.

Example 5-14 on page 69 demonstrates steps 1- 3. VMReady is disabled by running the **no virt enable** command. Then, the VSI Type database is configured to use the DVS 5000V database setup in Example 5-9 on page 65.

Note: The management IP for the DVS 5000V Controller is 172.25.155.5. This is required when configuring EVB, as shown in Example 5-14 on page 69.

The `lldp enable` command enables LLDP globally.

Example 5-14 Physical switch configuration for EVB

```
en4093flex_1#conf t
Enter configuration commands, one per line. End with Ctrl/Z.
en4093flex_1(config)#no virt enable
en4093flex_1(config)#virt evb vsidb 1
en4093flex_1(config-vsidb)#host 172.25.155.5
en4093flex_1(config-vsidb)#port 44444
en4093flex_1(config-vsidb)#filepath vsitypes
en4093flex_1(config-vsidb)#update-interval 60
en4093flex_1(config-vsidb)#exit
en4093flex_1(config)#lldp enable
en4093flex_1(config)#end
en4093flex_1#
```

To verify the download of the VSI Type database on the physical switch, run the following command:

```
5000V# show virt evb vsitypes
```

Any defined VSI Types are listed in the output.

Next, enable EVB on the interfaces connecting to the DVS 5000V uplinks. This is done by creating an EVB profile and adding it to the interface port configuration, as shown in Example 5-15.

This example enables the two required features:

- ▶ Reflective relay, which is needed for hair-pinning traffic in VEPA mode
- ▶ VSI Discovery Protocol (VDP), which is used for capability negotiation and to associate VM MAC addresses and VSI Types on the interface

Example 5-15 Physical switch interface configuration for EVB

```
en4093flex_1#conf t
Enter configuration commands, one per line. End with Ctrl/Z.
en4093flex_1(config)#virt evb profile 1
en4093flex_1(config-evbprof)#reflective-relay
en4093flex_1(config-evbprof)#vsi-discovery
en4093flex_1(config-evbprof)#exit
en4093flex_1(config)#interface port INTA2
en4093flex_1(config-if)#evb profile 1
en4093flex_1(config-if)#exit
en4093flex_1(config)#interface port INTA3
en4093flex_1(config-if)#evb profile 1
en4093flex_1(config-if)#end
en4093flex_1#
```

EVB discovery and configuration do not commence on the DVS 5000V until a VM is assigned to a profile or port with a VSI Type. Before this assignment occurs, `show` commands on the physical switch do not reveal any EVB capabilities in the LLDP neighbor.

Our lab environment includes two IBM EN4093 switches connecting to our IBM Flex System compute nodes, so we repeat the EVB configuration for the second IBM EN4093 switch.

5.3.6 Connecting a virtual machine and verifying the EVB VEPA implementation

To connect a VM with VEPA, you can use the vNIC profile or stand-alone port that is created for VEPA. In Figure 5-11, we use the vNIC profile with VEPA from Example 5-12 on page 67, and assign it to the VM.

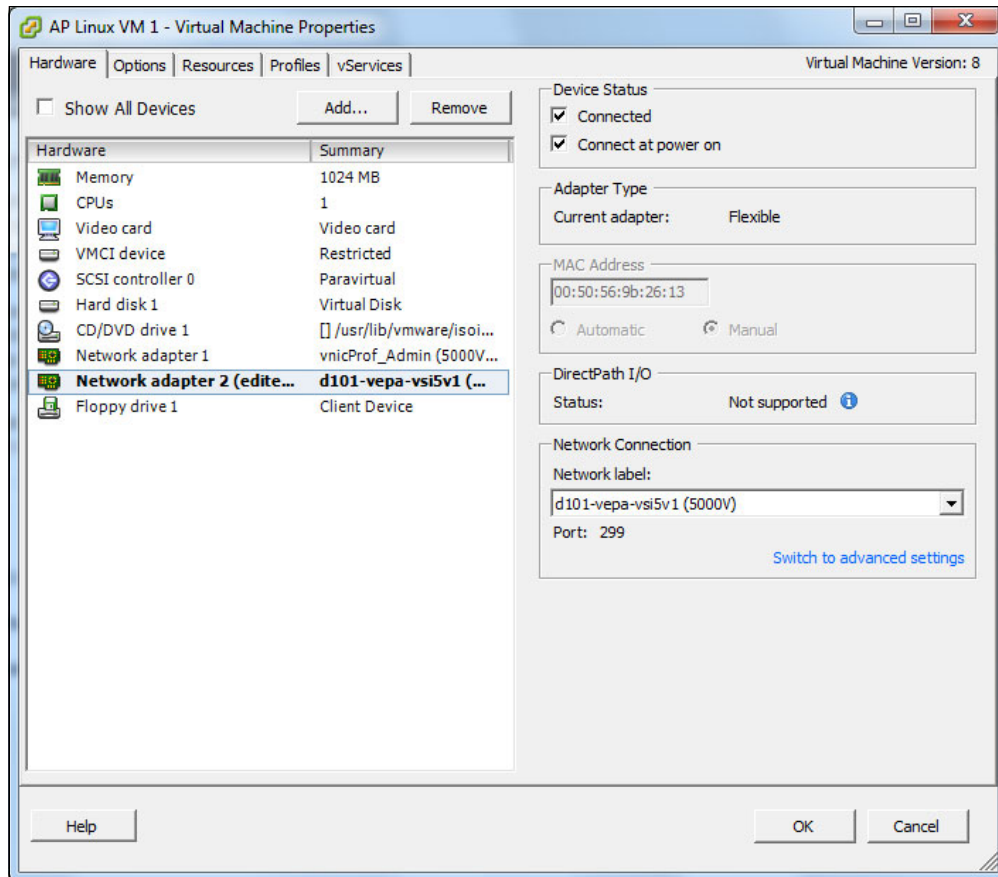


Figure 5-11 Applying VEPA to a VM network interface with a vNIC profile

When the VM is configured with the VEPA profile, the DVS 5000V performs EVB negotiation and configuration with the physical switch. The IBM EN4093 switch logs the following message when the VM is successfully associated:

```
en4093flex_1 WARNING vm: VSI Type ID 5 Associated mac 00:50:56:9b:26:13 on port INTA3
```

The VM associations can be viewed by running **show** commands, as shown in Example 5-16.

Example 5-16 Verifying the EVB VM association

```
en4093flex_1#show virt evb vdp tlv
VDP TLVs
Type Length OUI Subtype Request Resp MgrId
---- -
127 38 00:17:ef 2 ASSOCIATE 0 0
Type ID : 5
Type Version: 1
Instance ID : 0x49424d30303530353639623236313300
```

```

Mac Vlan      : 1
Num Entries  : 1
VCB Port     : INTA3
VCB Stag     : 0
VCB State    : 4
VCB timestp  : 546787691
VCB index    : 1
Entry        : 1
  MAC        : 00:50:56:9b:26:13
  Vlan       : 101

```

```

en4093flex_1#show virt evb vdp vm
Total number of VM Association entries : 1
TypeId  MAC                Vlan  Port  TxACL  RxEntry  ACLs
-----  -
5       00:50:56:9b:26:13  101  INTA3  256   33       255
en4093flex_1#

```

The ACL numbers in the output that is shown in Example 5-16 on page 70 are dynamically created ACLs that are based on the ACLs that are defined in VSI Type. The ACLs can be reviewed by running **show access-control list**. They represent ingress filtering or metering on the physical switch (TX from the VM).

There is no command that is available to review the ACL for egress bandwidth metering (applied through the rx-cbr VSI Type setting). However, out of policy discards can be viewed by running **show interface port interface-counters**. Discards from ingress access control and bandwidth metering (TX traffic from the VM) shows as Filter Discards, and egress bandwidth metering is counted as Other Discards.

5.3.7 VM Mobility with VEPA

There are no special considerations for migrating a VM with EVB and VEPA. The standard VMware migration checking occurs for storage and networking (the target host must be a member of the DVS 5000V). When migrating a VM, the DVS 5000V and EVB protocols take care of migrating the network profile to the new physical port.

It is important to remember that the EVB configuration must be on all physical ports connecting the VMware ESXi cluster. There are no consistency checks for this when initiating a migration, so a migration proceeds even when this condition is not met.

5.3.8 Virtual Edge Bridging with VSI Types

The use of VSI Types in EVB is intended for operation with VEPA. However, VSI Types and VDP can be implemented with Virtual Edge Bridging (VEB) connected VMs. The result of this type of implementation is that the policy or profile for the VSI Type is not implemented for traffic that is locally switched within the host by the virtual switch. However, VDP still associates the VM with the defined VSI Type in the physical switch. Traffic that is destined for an external station, or a VM on another host, is forwarded to the physical switch where it is governed by the policy for the associated VSI Type.

Although this is not a recommended implementation, the behavior should be understood for configuration guidelines, troubleshooting, or consideration for whether it can be used for a unique use case.

5.4 Advanced switch features

In the following sections, some of the advanced features of the DVS 5000V are presented.

5.4.1 Quality of service (QoS)

Quality of service (QoS) features allow you to allocate network resources to mission-critical applications at the expense of applications that are less sensitive to factors such as time delays or network congestion. The network can be configured to prioritize specific types of traffic, ensuring that each type receives the appropriate QoS level. By assigning QoS levels to traffic flows on your network, you can ensure that network resources are allocated where they are needed most. QoS features allow you to prioritize network traffic, providing better service for selected applications.

DVS 5000V QoS functions include packet classification and traffic conditioning. Packets are classified based on the content in the packet header. Traffic conditioning includes metering, policing, and remarking to ensure that traffic entering the DVS 5000V domain conforms to the traffic classifier rules, traffic profiles, metering, marking, and discarding rules that are applied to the traffic.

The basic workflow of QoS is classifying traffic that is based on specified parameters and processing packets that are based on a set of defined actions. Figure 5-12 shows the logical processing flow of QoS.

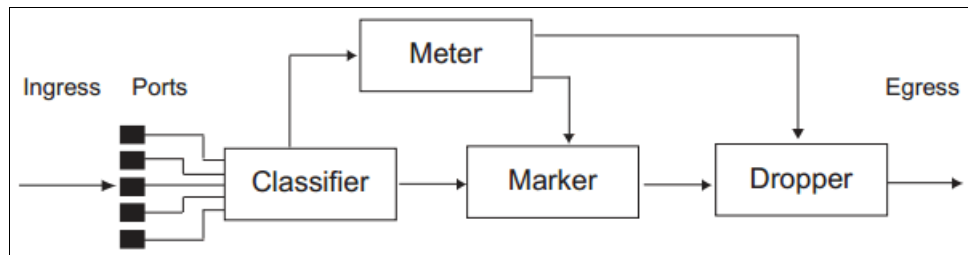


Figure 5-12 Logical flow of QoS process in DVS 5000V

Here are the traffic classifiers that can be used in the DVS 5000V QoS:

- ▶ CoS: Classifies packets that are based on the 802.1p priority value.
- ▶ DSCP: Classifies packets that are based on the DSCP value.
- ▶ ACLs: Classifies packets that are based on a combination of one or more header fields, such as source address, destination address, protocol ID, and port number.

Each filter defines the conditions that must match for inclusion in the filter. The actions are performed when a match is made. Here are the actions that are applicable in the QoS:

- ▶ Defining bandwidth and burst parameters
- ▶ Selecting actions to perform on in-profile and out-of-profile traffic
- ▶ Denying packets
- ▶ Permitting packets
- ▶ Marking DSCP or 802.1p priority

5.4.2 Access Control Lists

Access control lists (ACLs) are filters for permitting or denying traffic to provide security to the network. ACLs can be used with QoS for packet segmentation and classification of different types of traffic. ACLs work by inspecting each packet header and running specific actions to decide whether a packet can be passed or dropped, based on specified traffic parameters, such as network address, host address, and application port.

The DVS 5000V supports two types of ACLs: IPv4 ACLs and MAC ACLs.

IPv4 ACLs

DVS 5000V supports up to 127 ACLs for any network that uses IPv4. There are two types of IPv4 ACLs: standard and extended. Standard ACLs use only network and host address. In addition to Standard ACLs, Extended ACLs can be used for specifying an application port.

The following command implements IPv4 ACLs:

```
5000V(config)# access-list ip <128-254> <standard|extended>
```

ACLs allow you to classify packets according to their contents. Once classified, further processing procedures are identified. The following IPv4 classifiers are provided in the DVS 5000V:

- ▶ Source IPv4 address and subnet mask
- ▶ Destination IPv4 address and subnet mask
- ▶ IP Protocol number or name, as shown in Table 5-4

Table 5-4 Well-known protocols

Port number	Protocol name
1	ICMP
4	IPV4
6	TCP
17	UDP
89	OSPF
103	PIM

The classifiers in Table 5-4 are applicable in Standard ACLs. For a more granular configuration of ACLs, the following header content classifiers can be used in Extended ACLs (including MAC ACLs):

- ▶ TCP/UDP application ports (see Table 5-5)
- ▶ TCP flag values

Table 5-5 lists the TCP/UDP classifier in the extended mode of ACLs.

Table 5-5 TCP/UDP classifier in the extended mode of ACLs

Port number	TCP/UDP application name
20	ftp-data
21	ftp

Port number	TCP/UDP application name
22	ssh
23	telnet
25	smtp
37	time
42	name
43	whois
53	domain
69	tftp
70	gopher
79	finger
80	http
109	pop2
110	pop3
111	sunrpc
119	nntp
123	ntp
143	imap
144	news
161	snmp
162	snmptrap
179	bgp
194	irc
220	imap3
389	ldap
443	https
520	rip
554	rtsp
1645/1812	radius
1813	radius accounting
1985	hsrp

Table 5-6 on page 75 lists the TCP flags and values.

Table 5-6 TCP flags and values

TCP flag	Value
URG	0x0020
ACK	0x0010
PSH	0x0008
RST	0x0004
SYN	0x0002
FIN	0x0001

MAC ACLs

Beside packet filtering that is based on an IPv4 address, the DVS 5000V can also support up to 128 ACLs based on a MAC address.

Use the following command to implement MAC ACLs:

```
5000V(config)# access-list mac <1-127> extended
```

MAC ACLs packet classification are based on the following criteria:

- ▶ Source MAC address
- ▶ Destination MAC address
- ▶ VLAN number
- ▶ Ethernet type (ARP, IP, or RARP)
- ▶ Ethernet Priority (the IEEE 802.1p priority)

ACL implementation in the DVS 5000V

There are two main steps of ACL implementation:

1. Defining filters that consist of an action for specific type of traffic
2. Assigning filters to either a port, VLAN, or vNIC profile

Each packet that flows in an assigned area (ports, VLAN, or vNIC profile) can be processed differently when classified by ACLs.

Defining standard and extended ACLs

The ACL implementation scenarios that encompass standard and extended ACLs are described in Table 5-7.

Table 5-7 List of ACLs

ACL rules	Definition
128	Allow access from VLAN Data 101.
129	Allow HTTP access.
130	Allow HTTPS access.
131	Allow ICMP access.
132	Allow Telnet access.
133	Allow SSH access.

ACL rules	Definition
200	Deny traffic from App Linux 1 MAC address.
201	Deny any traffic.

The commands for defining ACLs based on the criteria that are defined in Table 5-7 on page 75 are shown in Example 5-17.

Example 5-17 ACL configuration

```
5000V(config)# access-list ip 128 standard
5000V(config-std-nacl)# permit 192.168.1.0 255.255.255.0 host 172.25.155.9
5000V(config-std-nacl)# exit
```

```
5000V(config)# access-list ip 129 extended
5000V(config-ext-nacl)# permit 80 any any
5000V(config-ext-nacl)# exit
```

```
5000V(config)# access-list ip 139 extended
5000V(config-ext-nacl)# permit tcp any any eq 443
5000V(config-ext-nacl)# exit
```

```
5000V(config)# access-list ip 131 extended
5000V(config-ext-nacl)# permit icmp any any
5000V(config-ext-nacl)# exit
```

```
5000V(config)# access-list ip 132 extended
5000V(config-ext-nacl)# permit 23 any any
5000V(config-ext-nacl)# exit
```

```
5000V(config)# access-list ip 133 extended
5000V(config-ext-nacl)# permit 22 any any
5000V(config-ext-nacl)# exit
```

```
5000V(config)# access-list mac extended 1
5000V(config-ext-macl)# deny host 00:50:56:9b:97:c9 host 00:50:56:9B:2D:69
5000V(config-ext-macl)# exit
```

```
5000V(config)# access-list ip 201 standard
5000V(config-ext-nacl)# deny any any
5000V(config-ext-nacl)# exit
```

Assigning ACLs

When the ACLs are defined on the switch, they must be assigned to the appropriate ports, VLAN, or vNIC profiles. Each port can accept multiple ACLs, and each ACL can be applied for multiple ports. ACLs can be assigned individually. In this scenario, ports and ACL association are described in Table 5-8.

Table 5-8 ACLs and nodes association

Node	VLAN	IP address	MAC address	Port number	ACLs rules
Web server	777	172.25.155.9	00:50:56:9b:97:c9	102	1, 128, 129

To implement a MAC address ACL that allows access from VLAN 101 for HTTP access to a web server, apply ACL 1, 128, and 129 to port 102, as shown in Example 5-18 on page 77.

Example 5-18 Assigning ACLs to ports

```
5000V(config)# interface port 102
5000V(config-if)# mac access-group 1 in
5000V(config-if)# ip access-group 128 in
5000V(config-if)# ip access-group 129 in
```

5.4.3 Private VLANs

Private VLANs provide Layer 2 isolation between ports within the same broadcast domain. They split a broadcast into multiple isolated broadcast subdomains.

Private VLAN ports

Private VLAN ports are defined as follows:

- ▶ **Promiscuous:** A promiscuous port is a port that belongs to the primary VLAN. The promiscuous port can communicate with all the interfaces, including ports in the secondary VLANs (Isolated VLAN and Community VLANs). Each promiscuous port can be a member of only one Private VLAN.
- ▶ **Isolated:** An isolated port is a host port that belongs to an isolated VLAN. Each isolated port has complete Layer 2 separation from other ports in the same private VLAN (including other isolated ports), except for the promiscuous ports.
 - All traffic that is sent to an isolated port is blocked by the Private VLAN, *except* the traffic from promiscuous ports.
 - All traffic that is received from an isolated port is forwarded *only* to promiscuous ports.
- ▶ **Community:** A community port is a host port that belongs to a community VLAN. Community ports can communicate with other ports in the same community VLAN, and with promiscuous ports. These interfaces are isolated at Layer 2 from all other interfaces in other communities and from isolated ports within the Private VLAN.

Private VLAN types

Each subdomain is composed of one primary VLAN and one or more secondary VLANs, as follows:

- ▶ **Primary VLAN:** Unidirectional traffic downstream from promiscuous ports to isolated or community ports. Each Private VLAN configuration has only one primary VLAN. All ports in the private VLAN are members of the primary VLAN.
- ▶ **Secondary VLAN”** Secondary VLANs are internal to a private VLAN domain and are defined as follows:
 - **Isolated VLAN:** Carries unidirectional traffic upstream from the host servers toward ports in the primary VLAN and the gateway. Each Private VLAN configuration can contain only one isolated VLAN.
 - **Community VLAN:** Carries upstream traffic from ports in the community VLAN to other ports in the same community, and to ports in the primary VLAN and the gateway. Each Private VLAN configuration can contain multiple community VLANs.

After you define the primary VLAN and one or more secondary VLANs, you map the secondary VLANs to the primary VLAN.

Configuring a private VLAN in DVS 5000V

To configure a private VLAN in the DVS 5000V, complete the following steps:

1. Create a VLAN and define the private VLAN type as primary, as shown in Example 5-19.

Example 5-19 Creating a VLAN and selecting the primary type of the private VLAN

```
5000V(config)# vlan 200
5000V(config-vlan)# enable
5000V(config-vlan)# private-vlan type primary
5000V(config-vlan)# private-vlan enable
5000V(config-vlan)# exit
```

2. Create the secondary VLAN, define the private VLAN type as isolated, and map it to the primary private VLAN, as shown in Example 5-20.

Example 5-20 Configuring a secondary VLAN as the isolated type of the private VLAN

```
5000V(config)# vlan 203
5000V(config-vlan)# enable
5000V(config-vlan)# private-vlan type isolated
5000V(config-vlan)# private-vlan map 200
5000V(config-vlan)# private-vlan enable
5000V(config-vlan)# exit
```

3. Assign the primary and isolated type of private VLAN to the specific VNIC profile, as described in Example 5-21.

Example 5-21 Assigning a specific VNIC profile to the private VLAN

```
5000V(config)# iswitch vnicprof vnicProf_Isolated dvportgroup-121
5000V(config-vprof)# pvid 203
5000V(config-vprof)# exit
5000V(config)# iswitch vnicprof vnicProf_Promisc dvportgroup-122
5000V(config-vprof)# pvid 200
5000V(config-vprof)# exit
```

4. After you have VNIC profiles for the private VLAN, you can assign specific ports in the DVS 5000V to the VLAN, as described in 5.2.3, “Associating VMs with the DVS 5000V” on page 58.



Maintenance and troubleshooting

This chapter presents a high-level overview of the maintenance and troubleshooting tasks that can be performed on the IBM Distributed Virtual Switch 5000V (DVS 5000V).

This chapter covers the following topics:

- ▶ Configuration management
- ▶ Firmware management
- ▶ Logging and reporting
- ▶ Escalation to IBM Support

6.1 Configuration management

This section describes configuration management.

6.1.1 Configuration file and block

The switch stores its configuration in two files:

- ▶ `startup-config` is the configuration that the switch uses when it is reloaded.
- ▶ `running-config` is the configuration that reflects all the changes that you made from the CLI. It is stored in memory and is lost after the reload of the switch.

The switch stores its configuration in one of two configuration blocks, or partitions:

- ▶ `active-config` is the active configuration file.
- ▶ `backup-config` is the alternative configuration file.

This setup has the flexibility that you need to manage the configuration of the switch and perform a configuration rollback if needed.

6.1.2 Managing the configuration

This section describes how to manage the configuration by using the following commands on the CLI:

- ▶ **show running-config**

Dumps the current configuration to a script file. You can specify additional optional parameters for more specific information.

Command mode: All

- ▶ **show config-not-restored**

Displays an unsaved configuration that was not restored when the switch was last reloaded.

Command mode: All

- ▶ **show startup-config**

Dumps the startup configuration that is stored in flash memory.

Command mode: All

- ▶ **show active-config**

Dumps the active configuration that is stored in flash memory.

Command mode: All

- ▶ **show backup-config**

Dumps the backup configuration that is stored in flash memory.

Command mode: All

- ▶ **copy running-config backup-config**
Copies the current (running) configuration from switch memory to the backup-config partition. You must save configuration settings to flash memory for the DVS 5000V to reload the settings after a reset. If you do not save the changes, they are lost the next time that the system is rebooted.
Command mode: Privileged EXEC
- ▶ **copy running-config startup-config**
Copies the current (running) configuration from the switch memory to the startup-config partition.
Command mode: Privileged EXEC
- ▶ **copy running-config active-config**
Copies the current (running) configuration from the switch memory to flash memory.
Command mode: Privileged EXEC
- ▶ **show config-not-restored**
Shows the unsaved configuration from the previous boot.
Command mode: Privileged EXEC
- ▶ **copy running-config {tftp|scp}**
Backs up the current configuration to a file on the selected TFTP/SCP server.
Command mode: Privileged EXEC
- ▶ **copy {tftp|scp} active-config**
Restores the current configuration from a TFTP/SCP server.
Command mode: Privileged EXEC
- ▶ **copy {tftp|scp} backup-config**
Restores the backup configuration from a TFTP/SCP server.
Command mode: Privileged EXEC
- ▶ **copy active-config {tftp|scp}**
Copies the current configuration to a TFTP/SCP server.
Command mode: Privileged EXEC
- ▶ **copy backup-config {tftp|scp}**
Copies the backup configuration to a TFTP/SCP server.
Command mode: Privileged EXEC

6.2 Firmware management

The switch software image is the executable code that runs on the DVS 5000V. It has two image banks for the operating system firmware. The command in Example 6-1 is used to determine the current running operating system version.

Example 6-1 Command for showing the current running operating system in DVS 5000V

```
5000V# show boot
Currently set to boot software image2, active config block
image1: version 1.0.0.1506, downloaded
```

```
image2: version 1.0.0.1506, downloaded
boot: version 1.0.0.1506
```

Here are the requirements for upgrading the DVS 5000V:

- ▶ Load the new image on to an SCP or TFTP server on your network.
- ▶ Transfer the new image from the SCP or TFTP server to your switch.
- ▶ Select the new software image to be loaded into switch memory the next time the switch is reset.

6.2.1 Loading a new image in to the DVS 5000V Controller

The DVS 5000V Controller runs an implementation of IBM N/OS. The upgrade process is similar to any other IBM System Networking product (RackSwitch, Flex System Fabric Switches, and so on). The privileged commands that are shown in Example 6-2 can be used to upload a new version of software to the 5000V Controller.

Example 6-2 Uploading an image from a TFTP server on to a DVS 5000V

```
5000V# copy {tftp|scp} {image1|image2}
Address or name of remote host: <IP address or hostname>
Source file name: <filename>
User name: {<username>|<Enter>}
```

The system then prompts the user to confirm the request.

6.2.2 Selecting an image and configuration

When the upload process is done, you can verify that it uploaded by using the command that is shown in Example 6-1 on page 81. The next step is to select the image to be booted on the next reload of the controller.

Assuming the new image is in image2, Example 6-3 shows how to select the new image for booting inside the configuration mode.

Example 6-3 Selecting the image for the next boot

```
5000V(config)#boot image image2
Next boot will use switch software image2 instead of image1.
```

In addition to the image to be used upon reload, you also can select the wanted configuration to be used after the controller boots. Run the command that is shown in Example 6-4 to load the wanted configuration.

Example 6-4 Selecting the configuration to be loaded

```
5000V(config)#boot configuration-block active
Next boot will use switch active config block instead of factory default
```

After completing the image and configuration block selection process, you can reboot the switch to complete the changes by running the command that is shown in Example 6-5 on page 83.

Example 6-5 Reloading DVS 5000V

```
5000V#reload
Confirm reload (y/n)?y
```

6.3 Logging and reporting

This section contains information about where to go for additional information about IBM and IBM products, what to do if you experience a problem with your system, and whom to call for service, if necessary.

Before calling, make sure that you have taken these steps to try to solve the problem yourself:

- ▶ Check all cables to make sure that they are connected.
- ▶ Use the troubleshooting information in your system documentation and use the diagnostic tools that come with your system.
- ▶ Go to the IBM Support website at <http://www.ibm.com/systems/support/> to check for technical information, hints, tips, and new device drivers.

6.3.1 Information that is needed for a Problem Management Record

To ensure a complete level of service from IBM Support, collect the following data from the DVS 5000V by using the following commands before opening a Problem Management Record (PMR):

- ▶ **show tech-support**
 - ▶ Dumps 5000V information, statistics, and the running configuration to the terminal screen. You can save the output to a file.
Command mode: All except User EXEC
- ▶ **show flash-dump**
 - ▶ Displays memory dump information.
Command mode: All
- ▶ **show logging messages**
 - ▶ Shows the system's log messages.
Command mode: Privileged EXEC
- ▶ **show license**
 - ▶ Shows the current licensing information.
Command mode: Privileged EXEC

6.4 Contacting IBM Support

To contact IBM Support, use the following resources:

- ▶ For information about which products are supported by the IBM Support Line in your country or region, see the following website:
<http://www.ibm.com/services/sl/products/>

- ▶ For more information about IBM Support Line and other IBM services, see the following website:
<http://www.ibm.com/services/>
- ▶ For IBM Support telephone numbers, see the following website:
<http://www.ibm.com/planetwide/>
- ▶ In the US and Canada, call 1-800-IBM-SERV (1-800-426-7378).

Related publications

The publications that are listed in this section are considered suitable for a more detailed discussion of the topics that are covered in this book.

Online resources

This website is relevant as a further information source:

- ▶ *IBM SDN VE User Guide (1.0/VMware)*, found at:

<http://www.ibm.com/support/docview.wss?uid=isg3T7000628>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



IBM Distributed Virtual Switch 5000V Quickstart Guide

(0.2"spine)
0.17"->0.473"
90->249 pages



IBM Distributed Virtual Switch 5000V Quickstart Guide



Managed Layer 2 distributed virtual switch for VMware

Advanced networking and management features

Implementation and troubleshooting

The IBM Distributed Virtual Switch 5000V (DVS 5000V) is a software-based network switching solution that is designed for use with the virtualized network resources in a VMware enhanced data center. It works with VMware vSphere and ESXi 5.0 and beyond to provide an IBM Networking OS management plane and advanced Layer 2 features in the control and data planes. It provides a large-scale, secure, and dynamic integrated virtual and physical environment for efficient virtual machine (VM) networking that is aware of server virtualization events, such as VMotion and Distributed Resource Scheduler (DRS). The DVS 5000V interoperates with any 802.1Qbg compliant physical switch to enable switching of local VM traffic in the hypervisor or in the upstream physical switch. Network administrators who are familiar with IBM System Networking switches can manage the DVS 5000V just like IBM physical switches by using advanced networking, troubleshooting, and management features to make the virtual switch more visible and easier to manage.

This IBM Redbooks publication helps the network and system administrator install, tailor, and quickly configure the IBM Distributed Virtual Switch 5000V (DVS 5000V) for a new or existing virtualization computing environment. It provides several practical applications of the numerous features of the DVS 5000V, including a step-by-step guide to deploying, configuring, maintaining, and troubleshooting the device. Administrators who are already familiar with the CLI interface of IBM System Networking switches will be comfortable with the DVS 5000V. Regardless of whether the reader has previous experience with IBM System Networking, this publication is designed to help you get the DVS 5000V functional quickly, and provide a conceptual explanation of how the DVS 5000V works in tandem with VMware.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-8115-00

ISBN 0738439886