NX-OS and Cisco Nexus Switching

Next-Generation Data Center Architectures

The complete guide to planning, configuring, managing, and troubleshooting NX-OS in enterprise environments

Kevin Corbin, CCIE® No. 11577
Ron Fuller, CCIE No. 5851
David Jansen, CCIE No. 5952

ciscopress.com

NX-OS and Cisco Nexus Switching

Next-Generation Data Center Architectures

Kevin Corbin, CCIE No. 11577
Ron Fuller, CCIE No. 5851
David Jansen, CCIE No. 5952

Cisco Press
800 East 96th Street
Indianapolis, IN 46240

NX-OS and Cisco Nexus Switching
Next-Generation Data Center Architectures

Kevin Corbin, Ron Fuller, David Jansen

Published by:

Cisco Press

800 East 96th Street

Indianapolis, IN 46240 USA

Printed in the United States of America

Second Printing: July 2010

Library of Congress Cataloging-in-Publication data is on file.

ISBN-13: 978-1-58705-892-9
ISBN-10: 1-58705-892-8

Warning and Disclaimer

This book is designed to provide information about the Nexus Operating system and Nexus family of products. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an “as is” basis. The authors, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

Corporate and Government Sales

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact: **U.S. Corporate and Government Sales** 1-800-382-3419 corpsales@pearsontechgroup.com

For sales outside the United States please contact: **International Sales** international@pearsoned.com

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers’ feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through email at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Publisher Paul Boger

Associate Publisher Dave Dusthimer

Executive Editor Brett Bartow

Managing Editor Sandra Schroeder

Project Editor Seth Kerney

Editorial Assistant Vanessa Evans

Interior and Cover Designer Louisa Adair

Composition Mark Shirar

Manager, Global Certification Erik Ullanderson

Business Operation Manager, Cisco Press Anand Sundaram

Senior Development Editor Christopher Cleveland

Copy Editor Apostrophe Editing Services

Technical Editors Phil Davis, Eric Murray

Indexer WordWise Publishing Services

Proofreader Water Crest Publishing

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

Dedications

Kevin Corbin: I would like to dedicate this book to my parents. You have loved and supported me through all my endeavors. Mom, you instilled in me a work ethic that has been at the root of everything I have done. Dad, you taught me perseverance, and that the only time something is impossible is when you think it is. Nothing that I will ever accomplish would have been possible without both of you, I love you.

Ron Fuller: This book is dedicated to my loving wife Julie and my awesome children: Max, Sydney, Veronica, and Lil Bubba. Thank you for showing me the world through your perspective and helping me appreciate the things I would have otherwise taken for granted. I can't thank you enough for believing in me when I told you I was going to write another book. Your support and encouragement has and always will be the key to any success I enjoy. Thank you for your love and support.

David Jansen: This book is dedicated to my loving wife Jenise and my three children: Kaitlyn, Joshua, and Jacob. You are the inspiration that gave me the dedication and determination to complete this project. Kaitlyn, Joshua, Jacob, you are three amazing kids, you are learning the skills to be the best at what you do and accomplish anything; keep up the great work. Thank you for all your love and support; I could not have completed this without your help, support, and understanding. I'm so grateful to God, who gives endurance, encouragement, and motivation to complete such a large project like this.

About the Authors

Kevin Corbin, CCIE No. 11577, is a technology solutions architect with Cisco. In this role for three years, Kevin works with Enterprise customers to help them develop their next-generation data center architectures. Kevin has more than 14 years of server and networking experiencing including routing, switching, security, and content networking. Kevin has also held multiple certifications from Microsoft, Citrix, HP, Novell, and VMWare. Prior to joining Cisco, Kevin worked for many large enterprises and most recently in a consulting capacity for large enterprise customers.

Ron Fuller, CCIE No. 5851 (Routing and Switching/Storage Networking), is a technical solutions architect for Cisco specializing in data center architectures. He has 19 years of experience in the industry and has held certifications from Novell, HP, Microsoft, ISC2, SNIA, and Cisco. His focus is working with Enterprise customers to address their challenges with comprehensive end-to-end data center architectures. He lives in Ohio with his wife and three wonderful children and enjoys travel and auto racing.

David Jansen, CCIE No. 5952, is a technical solutions architect for Data Center for Central Area. David has more than 20 years experience in the information technology industry. He has held multiple certifications from Microsoft, Novell, Checkpoint, and Cisco. His focus is to work with Enterprise customers to address end-to-end data center Enterprise architectures. David has been with Cisco for 12 years and working as a Technical Solutions Architect for 4 years and has provided unique experiences helping customers build architectures for Enterprise data centers. David has also been instrumental in developing data center interconnect solutions to address L2 requirements between multiple data centers to meet application clusters and virtualization requirements. David has been presenting data center interconnect at Cisco Live for 3 years. David holds a B.S.E. degree in computer science from the University of Michigan (Go Blue!) and an M.A. degree in adult education from Central Michigan University.

About the Technical Reviewers

Phil Davis, CCIE No. 2021, is a technical solutions architect with Cisco, specializing in routing and switching and data center technologies. Phil has been with Cisco for more than 10 years and has more than 17 years of experience in the industry. Phil currently uses his expertise with Enterprise customers designing their data center and multiprotocol network architectures. Phil holds multiple certifications, including VMware's VCP, and is often presenting on many of today's top technologies. Phil lives near Cincinnati, Ohio, with his wife and two children.

Eric Murray is a network engineer for a large healthcare company. He has more than 15 years experience with designing, implementing, and maintaining Cisco Enterprise networks in the fast-paced healthcare and manufacturing industries. Eric has implemented several Nexus data center network designs and migrations and is a subject matter expert in utilizing Nexus 7000, 5000, and 2000 series switches. Eric is currently involved with designing, testing, implementing, and providing technical support for a Cisco Unified Communications solution. Eric also has extensive experience in multiprotocol WAN and data center LAN environments utilizing Cisco switching and routing platforms.

Acknowledgments

Kevin Corbin: I would like to first thank my co-authors Ron Fuller and David Jansen. I truly enjoy working with you on a day-to-day basis, and I am truly honored to have the opportunity to collaborate with you, and to even be considered in the same league as you guys. You are both rock stars. I would also like to recognize Steve McQuerry for his role in getting me involved in this project and providing coaching throughout this process.

I would like to thank the Cisco Press team, specifically Brett Bartow and Chris Cleveland. Thank you for being patient with me as a I got ramped up for the project and keeping me motivated to make this project a reality. To Phil Davis and Eric Murray, thank you for keeping us honest throughout your review process.

The development of this content would not have been possible without a significant amount of access to equipment, and I'd like to thank Hongjun Ma and Jon Blunt for their commitment to ensure that gear was available and accessible to me.

Working at Cisco has opened up a world of opportunity for me and challenged me on almost a daily basis to accomplish things that I never could have imagined that I was capable of. For this I would be remiss if I didn't give my most sincere thanks to Joel Ekis for opening the door; Gary McNiel for taking a chance on me; and Scott Sprinkle and Jason Heiling for their support throughout my time at Cisco.

Ron Fuller: First I'd like to thank my co-authors Dave Jansen and Kevin Corbin. Dave, thank you for being such a good friend, a trusted co-worker, and a leader in our organization. You set the bar the rest of us try to reach. It has been great sharing a brain with you, and I look forward to more challenges and fun. Keep the goat rodeos coming! Kevin, thank you for stepping in to help complete this project. You are awesome to work with and your technical acumen is top-notch. People like you and Dave are the reason I love my job.

I'd like to thank Brett Bartow for his (almost) infinite patience with this project. It is a huge undertaking and his persistence and understanding and encouragement were greatly appreciated.

Chris Cleveland, it has been a pleasure working with you. Your guidance on the formatting and consistency makes the book something we all can be proud of. Thank you for making three propeller heads from Cisco look good.

To our technical editors, Phil Davis and Eric Murray—wow, you guys are picky! Thank you for the detail-oriented work and assistance making the book accurate and concise.

To Jeff Raymond, Marty Ma, and Charlie Lewis—thank you for allowing us access to the hardware. This book wouldn't have been possible without your help.

I'd like to thank my manager, Bill Taylor, for his support throughout this project and understanding. You are a great manager and I truly enjoy working for you. Thanks for the opportunity and the support you've provided over the last five years. (Time flies when you are having fun!)

To my family, thank you for the many times you wanted me to do something and hearing about a book on things you don't get to see. Your understanding and support through the weekends and late nights are truly appreciated.

For the extended teams at Cisco—thank you for responding to my many emails and calls no matter how inane you thought they were. There was a method to the madness—I think. Working with a world-class organization like this makes coming to work a pleasure.

Finally, I want to thank God for the gifts he has given me and the opportunity to do what I love to do with people I enjoy to support my family. I couldn't ask for more.

David Jansen: This is my second book, and it has been a tremendous honor to work with the great people at Cisco Press. There are so many people to thank, I'm not sure where to begin. I'll start with Brett Bartow: Thank you for getting me started in the writing industry; this is something I enjoy doing. I appreciate your patience and tolerance on this project. I really appreciate you keeping me on track to complete the project in a timely manner, as we have missed several completion dates.

First, I would like to thank my friend and co-authors Ron Fuller and Kevin Corbin. I can't think of two better people to work with to complete such a project. Cisco is one of the most amazing places I've ever worked, and it's people like you, who are wicked smart and a lot of fun to work with, that make it such a great place. I look forward to working on other projects in the future. I am truly blessed by having both of you as a co-worker and friend. I look forward to continue to work with you and grow the friendship into the future.

Chris Cleveland, again it was a pleasure to work with you. Your expertise, professionalism, and follow-up as a development editor is unsurpassed; thank you for your hard work and quick turn-around; this helped to meet the deadlines set forth.

To our technical editors—Phil Davis and Eric Murray—thank you for the time, sharp eyes, and excellent comments/feedback. It was a pleasure having you as part of the team.

Thank you to Jeff Raymond, Marty Ma, Lincoln Dale, and Ben Basler from Data Center Business Unit (DCBU) to provide access to hardware to complete this book. Also, thank you Charlie Lewis in RTP CPOC for scheduling hardware to complete this book as well.

Thanks to my manager at Cisco, Bill Taylor—I appreciate your guidance and your trust in my ability to juggle the many work tasks along with extra projects like working on a book.

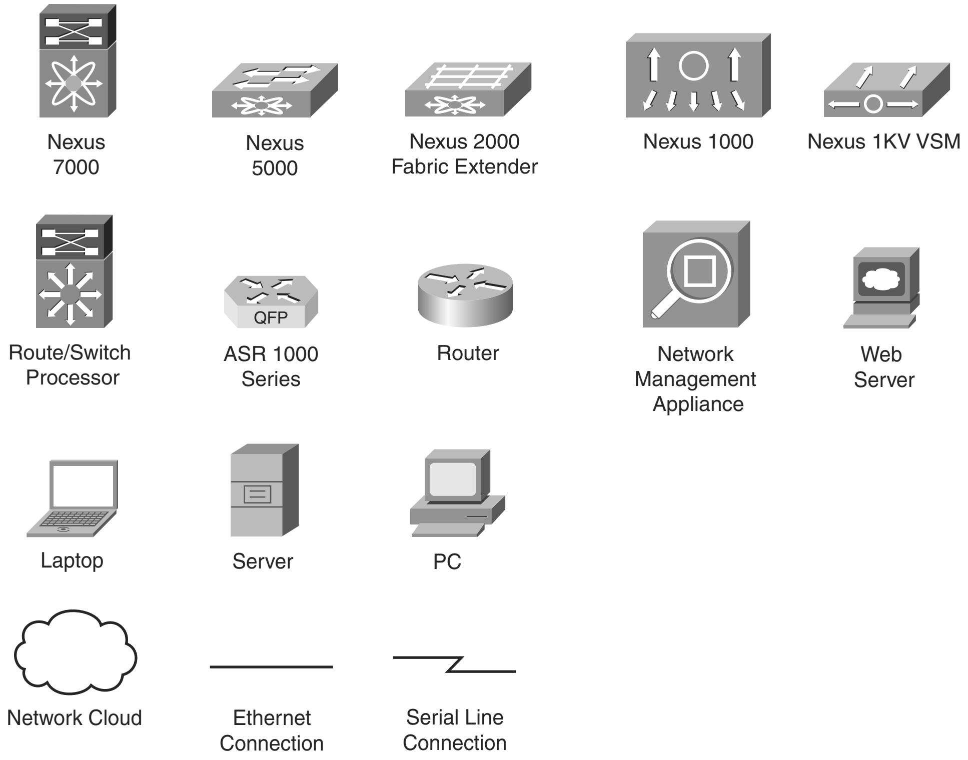
I would like to thank the heavy metal music world out there—it allowed me to stay focused when burning the midnight oil; I would not have been able to complete this without loud rock 'n roll music. Thank you.

I want to thank my family for their support and understanding while I was working on this project late at night and being patient with me when my lack of rest may have made me a little less than pleasant to be around.

Most important, I would like to thank God for giving me the ability to complete such a task with dedication and determination and for providing me the skills, knowledge, and health needed to be successful in such a demanding profession.

Contents

Icons Used in This Book



Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- **Boldface** indicates commands and keywords that are entered literally as shown. In actual configuration examples and output (not general command syntax), boldface indicates commands that are manually input by the user (such as a **show** command).
- *Italic* indicates arguments for which you supply actual values.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets ([ ]) indicate an optional element.
- Braces ({ }) indicate a required choice.
- Braces within brackets ([{}]) indicate a required choice within an optional element.

Foreword

More than five years ago, Cisco had the vision of unifying the fabrics in the data center to enable consolidation, virtualization, and automation. Cisco gathered input from customers and partners, and feedback from TAC and the sales team, to begin the design of the Nexus series of switches. With the launch of the Nexus 7000 in 2008, the years of planning, discussion, and hard work paid off as this new platform was released to our customers. The Nexus 5000, Nexus 2000, and Nexus 1000V quickly followed, providing a comprehensive end-to-end data center architecture designed to solve the emerging challenges faced in the ever-changing space that is the data center.

Supporting key innovations that make the 24×7×365 highly available data center a reality, while aligning with the increased demands of virtualization, the Nexus portfolio is truly game-changing. These innovations span the breadth of the product line and encompass both hardware and software changes. A subset includes capabilities such as In-Service Software Upgrade (ISSU), virtual device contexts (vDC), virtual Port Channels (vPC), VN-Link, and Unified Fabric for Fibre Channel over Ethernet (FCoE). This breadth of new capabilities brings increased efficiencies to how data center networks are designed, engineered, and operated.

To that end, a book like the one you are reading will hopefully become a convenient reference for best practices deployment of these new technologies. It is written by three of our Enterprise data center technology solutions architects, who work with our customers on a daily basis and help them develop next-generation data center architectures. Their breadth of experience makes them perfect candidates to drive a project such as this.

We hope that as you read this book and learn more about the Nexus series of switches, and NX-OS specifically, you'll see the years of effort that made this product the Cisco flagship data center operating system now and in the years to come. Enjoy!

Umesh Mahajan, VP/GM
Ram Velaga, VP Product Management
Data Center Switching Technology Group
Cisco, San Jose

Introduction

The modern data center is rapidly changing and evolving to support the current and future demands of technology. At the center of this change is the network—the single entity that connects everything and touches all components of the data center. With that in mind, Cisco has launched a new series of switches, Nexus, based on a revolutionary new operating system, NX-OS, to meet these changes and provide a platform with the scalability, reliability, and comprehensive feature set required in the next generation data center.

The purpose of this book is to provide a guide for the network administrator who might not be familiar with Nexus and NX-OS. It is intended to be used as a "go-to" resource for concise information on the most commonly used aspects of NX-OS across the Nexus 7000, 5000, and 1000V platforms.

Goals and Methods

The goal of this book is to provide best practice configurations to common internetworking scenarios involving Nexus products. Having been network administrators ourselves, we are conscious of the pressures and challenges with finding accurate and relevant information, especially on new technology. We intend this book to be a resource the network administrator reaches for first.

Although there might be more than one way to accomplish a networking requirement, we focused on the best way that minimizes operational complexity and maximizes supportability. We realize and respect that there might be corner-case scenarios that call for configurations not described in this book but sincerely hope we address the vast majority of common configurations.

Who Should Read This Book?

This book is targeted for the network administrator, consultant, or student looking for assistance with NX-OS configuration. It covers the three major Cisco Nexus products and highlights key features of them in a way that makes it easy for the reader to digest and implement.

How This Book Is Organized

This book has been organized following the OSI system model for the initial chapters starting with Layer 2 and then moving to Layer 3. We then add in network-based services such as IP multicast, security, and high availability. Next the embedded serviceability features of NX-OS are explored before moving to emerging data center architecture, Unified Fabrics. Last, and certainly not least, we focus on Nexus 1000V and its capability to provide insight, consistent network policy, and simplified administration to virtualized environments.

Chapters 1 through 9 cover the following topics:

- **Chapter 1, "Introduction to Cisco NX-OS":** Provides the reader with the foundation for building NX-OS configurations including command-line interface (CLI) differences, virtualization capabilities, and basic file system management.
- **Chapter 2, "Layer 2 Support and Configurations":** Focuses on the comprehensive suite of Layer 2 technologies supported by NX-OS including vPC and Spanning Tree Protocol.
- **Chapter 3, "Layer 3 Support and Configurations":** Delves into the three most common network Layer 3 protocols including EIGRP, OSPF, and BGP. Additionally HSRP, GLBP, and VRRP are discussed.
- **Chapter 4, "IP Multicast Configuration":** Provides the reader the information needed to configure IP Multicast protocols such as PIM, Auto-RP, and MSDP.
- **Chapter 5, "Security":** Focuses on the rich set of security protocols available in NX-OS including CTS, ACLs, CoPP, DAI, and more.
- **Chapter 6, "High Availability":** Delves into the high-availability features built into NX-OS including ISSU, stateful process restart, stateful switchover, and non-stop forwarding.
- **Chapter 7, "Embedded Serviceability Features":** Provides the reader with the ability to leverage the embedded serviceability components in NX-OS including SPAN, configuration checkpoints and rollback, packet analysis, and Smart Call Home.
- **Chapter 8, "Unified Fabric":** Explores the industry leading capability for Nexus switches to unify storage and Ethernet fabrics with a focus on FCoE, NPV, and NPIV.
- **Chapter 9, "Nexus 1000V":** Enables the reader to implement Nexus 1000V in a virtualized environment to maximum effect leveraging the VSM, VEM, and port profiles.

Chapter 1

Introduction to Cisco NX-OS

This chapter provides an introduction and overview of NX-OS and a comparison between traditional IOS and NX-OS configurations and terminology. The following sections will be covered in this chapter:

- NX-OS Overview
- NX-OS User Modes
- Management Interfaces
- Managing System Files

NX-OS Overview

Cisco built the next-generation data center-class operating system designed for maximum scalability and application availability. The NX-OS data center-class operating system was built with modularity, resiliency, and serviceability at its foundation. NX-OS is based on the industry-proven Cisco Storage Area Network Operating System (SAN-OS) Software and helps ensure continuous availability to set the standard for mission-critical data center environments. The self-healing and highly modular design of Cisco NX-OS enables for operational excellence increasing the service levels and enabling exceptional operational flexibility. Several advantages of Cisco NX-OS include the following:

- Unified data center operating system
- Robust and rich feature set with a variety of Cisco innovations
- Flexibility and scalability
- Modularity
- Virtualization
- Resiliency

- IPv4 and IPv6 IP routing and multicast features
- Comprehensive security, availability, serviceability, and management features

Key features and benefits of NX-OS include

- **Virtual device contexts (VDC):** Cisco Nexus 7000 Series switches can be segmented into virtual devices based on customer requirements. VDCs offer several benefits such as fault isolation, administration plane, separation of data traffic, and enhanced security.
- **Virtual Port Channels (vPC):** Enables a server or switch to use an EtherChannel across two upstream switches without an STP-blocked port to enable use of all available uplink bandwidth.
- **Continuous system operation:** Maintenance, upgrades, and software certification can be performed without service interruptions due to the modular nature of NX-OS and features such as In-Service Software Upgrade (ISSU) and the capability for processes to restart dynamically.
- **Security:** Cisco NX-OS provides outstanding data confidentiality and integrity, supporting standard IEEE 802.1AE link-layer cryptography with 128-bit Advanced Encryption Standard (AES) cryptography. In addition to CTS, there are many additional security features such as access control lists (ACL) and port-security, for example.
- **Base services:** The default license that ships with NX-OS covers Layer 2 protocols including such features such as Spanning Tree, virtual LANs (VLAN), Private VLANS, and Unidirectional Link Detection (UDLD).
- **Enterprise Services Package:** Provides Layer 3 protocols such as Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), Intermediate System-to-Intermediate System (ISIS), Enhanced Interior Gateway Routing Protocol (EIGRP), Policy-Based Routing (PBR), Protocol Independent Multicast (PIM), and Generic Routing Encapsulation (GRE).
- **Advanced Services Package:** Provides Virtual Device Contexts (VDC), Cisco Trustsec (CTS), and Overlay Transport Virtualization (OTV).
- **Transport Services License:** Provides Overlay Transport Virtualization (OTV) and Multiprotocol Label Switching (MPLS) (when available).

Example 1-1 shows the simplicity of installing the NX-OS license file.

Example 1-1 *Displaying and Installing the NX-OS License File*

```
! Once a license file is obtained from Cisco.com and copied to flash, it can be installed for the chassis.
! Displaying the host-id for License File Creation on Cisco.com:
```

```
congo# show license host-id
License hostid: VDH=TBM14404807
! Installing a License File:
congo# install license bootflash:license_file.lic
Installing license ..done
congo#
```

Note NX-OS offers feature testing for a 120-day grace period. Here is how to enable a 120-day grace period:

```
    congo(config)# license grace-period
```

The feature is disabled after the 120-day grace period begins. The license grace period is enabled only for the default admin VDC, VDC1.

Using the grace period enables customers to test, configure, and fully operate a feature without the need for a license to be purchased. This is particularly helpful for testing a feature prior to purchasing a license.

NX-OS Supported Platforms

NX-OS data center-class operating system, designed for maximum scalability and application availability, has a wide variety of platform support, including the following:

- Nexus 7000
- Nexus 5000
- Nexus 2000
- Nexus 1000V
- Cisco MDS 9000
- Cisco Unified Computing System (UCS)
- Nexus 4000

Cisco NX-OS and Cisco IOS Comparison

If you are familiar with traditional Cisco IOS command-line interface (CLI), the CLI for NX-OS is similar to Cisco IOS. There are key differences that should be understood prior to working with NX-OS, however:

- When you first log into NX-OS, you go directly into EXEC mode.
- NX-OS has a setup utility that enables a user to specify the system defaults, perform basic configuration, and apply a predefined Control Plane Policing (CoPP) security policy.

- NX-OS uses a feature-based license model. An Enterprise or Advanced Services license is required depending on the features required.
- A 120-day license grace period is supported for testing, but features are automatically removed from the configuration after the expiration date is reached.
- NX-OS has the capability to enable and disable features such as OSPF, BGP, and so on via the **feature** configuration command. Configuration and verification commands are not available until you enable the specific feature.
- Interfaces are labeled in the configuration as Ethernet. There aren't any speed designations in the interface name. Interface speed is dynamically learned and reflected in the appropriate **show** commands and interface metrics.
- NX-OS supports Virtual Device Contexts (VDC), which enable a physical device to be partitioned into logical devices. When you log in for the first time, you are in the default VDC.
- The Cisco NX-OS has two preconfigured instances of VPN Routing Forwarding (VRF) by default (management, default). By default, all Layer 3 interfaces and routing protocols exist in the default VRF. The mgmt0 interface exists in the management VRF and is accessible from any VDC. If VDCs are configured, each VDC has a unique IP address for the mgmt0 interface.
- Secure Shell version 2 (SSHv2) is enabled by default. (Telnet is disabled by default.)
- Default login administrator user is predefined as admin; a password has to be specified when the system is first powered up. With NX-OS, you must enter a username and password; you cannot disable the username and password login. In contrast, in IOS you can simply type a password; you can optionally set the login to require the use of a username.
- NX-OS uses a kickstart image and a system image. Both images are identified in the configuration file as the kickstart and system boot variables; this is the same as the Cisco Multilayer Director Switch (MDS) Fibre Channel switches running SAN-OS.
- NX-OS removed the **write memory** command; use the **copy running-config startup-config**; there is also the alias command syntax.
- The default Spanning Tree mode in NX-OS is Rapid-PVST+.

Caution In NX-OS, you have to enable features such as OSPF, BGP, and CTS; if you remove a feature via the **no** feature command, all relevant commands related to that feature are removed from the running configuration.

For example, when configuring vty timeouts and session limits, consider Example 1-2, which illustrates the difference between IOS and NX-OS syntax.

Example 1-2 *vty Configurations and Session Limits, Comparing the Differences Between Traditional IOS and NX-OS*

```
! IOS:
congo#
congo(config)# line vty 0 9
congo(config)# exec-timeout 15 0
congo(config)# login
congo# copy running-config startup-config
-----------------------------------------------------------------
! NX-OS:
congo(config)# line vty
congo(config)# session-limit 10
congo(config)# exec-timeout 15

congo# copy running-config startup-config
```

NX-OS User Modes

Cisco NX-OS CLI is divided into command modes, which define the actions available to the user. Command modes are "nested" and must be accessed in sequence. As you navigate from one command mode to another, an increasingly larger set of commands become available. All commands in a higher command mode are accessible from lower command modes. For example, the **show** commands are available from any configuration command mode. Figure 1-1 shows how command access builds from EXEC mode to global configuration mode.

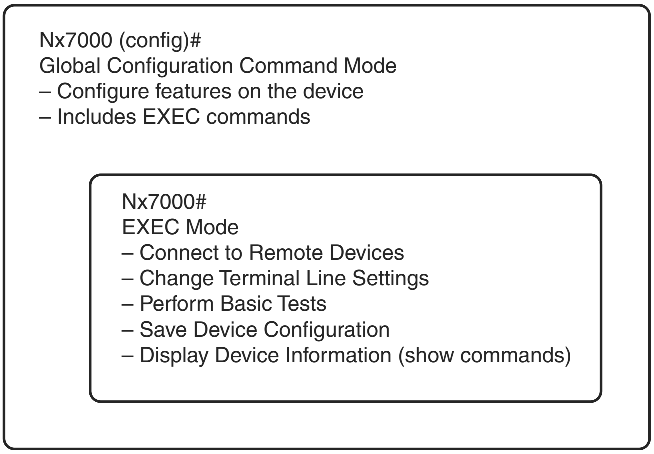

Nx7000 (config)#
Global Configuration Command Mode
– Configure features on the device
– Includes EXEC commands
Nx7000#
EXEC Mode
– Connect to Remote Devices
– Change Terminal Line Settings
– Perform Basic Tests
– Save Device Configuration
– Display Device Information (show commands)

Figure 1-1 *NX-OS Command Access from EXEC Mode to Global Configuration Mode*

EXEC Command Mode

When you first log in, Cisco NX-OS Software places you in EXEC mode. As demonstrated in Example 1-3, the commands available in EXEC mode include the **show** commands that display device status and configuration information, the **clear** commands, and other commands that perform actions that you do not save in the device configuration.

Example 1-3 *Cisco NX-OS EXEC Mode*

```
Congo# show interface ethernet 1/15
Ethernet1/15 is down (SFP not inserted)
  Hardware: 10000 Ethernet, address: 001b.54c2.bbc1 (bia 001b.54c1.e4da)
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA
  auto-duplex, auto-speed
  Beacon is turned off
  Auto-Negotiation is turned off
  Input flow-control is off, output flow-control is off
  Switchport monitor is off
  Last link flapped never
  Last clearing of "show interface" counters never
  30 seconds input rate 0 bits/sec, 0 packets/sec
  30 seconds output rate 0 bits/sec, 0 packets/sec
  Load-Interval #2: 5 minute (300 seconds)
    input rate 0 bps, 0 pps; output rate 0 bps, 0 pps
  L3 in Switched:
    ucast: 0 pkts, 0 bytes - mcast: 0 pkts, 0 bytes
  L3 out Switched:
    ucast: 0 pkts, 0 bytes - mcast: 0 pkts, 0 bytes
! Output omitted for brevity

Congo#
```

Global Configuration Command Mode

Global configuration mode provides access to the broadest range of commands. The term *global* indicates characteristics or features that affect the device as a whole. You can enter commands in global configuration mode to configure your device globally or enter more specific configuration modes to configure specific elements such as interfaces or protocols as demonstrated here:

```
Nx7000# conf t
Nx7000(config)# interface ethernet 1/15
```

Interface Configuration Command Mode

One example of a specific configuration mode that you enter from global configuration mode is interface configuration mode. To configure interfaces on your device, you must specify the interface and enter interface configuration mode.

You must enable many features on a per-interface basis. Interface configuration commands modify the operation of the interfaces on the device, such as Ethernet interfaces or management interfaces (mgmt 0).

Example 1-4 demonstrates moving between the different command modes in NX-OS.

Example 1-4 *Interface Ethernet1/5 Is a 10Gigabit Ethernet Interface—Show How the Interface Is Designated at Ethernet and Not Interface Ten1/15.*

```
congo# conf t
congo(config)# interface ethernet 1/15
congo(config-if)# exit
Congo# show interface ethernet 1/15
Ethernet1/15 is down (SFP not inserted)
  Hardware: 10000 Ethernet, address: 001b.54c2.bbc1 (bia 001b.54c1.e4da)
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA
  auto-duplex, auto-speed
  Beacon is turned off
  Auto-Negotiation is turned off
  Input flow-control is off, output flow-control is off
  Switchport monitor is off
  Last link flapped never
  Last clearing of "show interface" counters never
  30 seconds input rate 0 bits/sec, 0 packets/sec
  30 seconds output rate 0 bits/sec, 0 packets/sec
  Load-Interval #2: 5 minute (300 seconds)
    input rate 0 bps, 0 pps; output rate 0 bps, 0 pps
  L3 in Switched:
    ucast: 0 pkts, 0 bytes - mcast: 0 pkts, 0 bytes
  L3 out Switched:
    ucast: 0 pkts, 0 bytes - mcast: 0 pkts, 0 bytes

Congo#
```

NX-OS supports different Ethernet interface types such as Gigabit Ethernet and 10-Gigabit Ethernet interfaces. All interfaces are referred to *Ethernet*; NX-OS does not designate Gigabit or 10-Gigabit Ethernet interfaces. In Example 1-4, interface 1/15 is a 10-Gigabit Ethernet interface.

Management Interfaces

NX-OS has many different type of management interfaces, all of which the following section covers:

- **Controller Processor (CP)/Supervisor:** Has both the management plane and control plane and is critical to the operation of the network.
- **Connectivity Management Processor (CMP):** Provides a second network interface to the device for use even when the CP is not reachable. The CMP interface is used for out-of-band management and monitoring; the CMP interface is independent from the primary operating system.
- **MGMT0:** Provides true out-of-band management through a dedicated interface and VRF to ensure 100 percent isolation from either control plane or data plane. MGMT0 enables you to manage the devices by the IPv4 or IPv6 address on the MGMT0 interface; the mgmt0 interface is a 10/100/1000 Ethernet interface. When implementing Virtual port-channel (vPC), a best practice is to use the MGMT0 interface for the VPC keepalive link.
- **Telnet:** Provides an unsecure management connection to the NX-OS device.
- **SSH:** Provides a secure management connection to the NX-OS device.
- **Extended Markup Language (XML) management interfaces:** Use the XML-based Network Configuration Protocol (NETCONF) that enables management, monitoring, and communication over the interface with an XML management tool or program.
- **Simple Network Management Protocol (SNMP):** Used by management systems to monitor and configure devices via a set of standards for communication over the TCP/IP protocol.

Controller Processor (Supervisor Module)

The Cisco Nexus 7000 series supervisor module is designed to deliver scalable control plane and management functions for the Cisco Nexus 7000 Series chassis. The Nexus 7000 supervisor module is based on an Intel dual-core processor that enables a scalable control plane. The supervisor modules controls the Layer 2 and Layer 3 services, redundancy capabilities, configuration management, status monitoring, power, and environmental management. The supervisor module also provides centralized arbitration to the system fabric for all line cards. The fully distributed forwarding architecture enables the supervisor to support transparent upgrades to higher forwarding capacity-capable I/O and fabric modules. Two supervisors are required for a fully redundant system, with one supervisor module running as the active device and the other in hot standby mode, providing exceptional high-availability features in data center-class products. Additional features and benefits of the Nexus 7000 supervisor modules to meet demanding data center requirements follow:

- Active and standby supervisor.
- In-Service Software Upgrade (ISSU) with dual supervisor modules.
- Virtual output queuing (VoQ), which is a quality of service (QoS)-aware lossless fabric, avoids the problems associated with head-of-line blocking.
- USB interfaces that enable access to USB flash memory devices for software image loading and recovery.
- Central arbitration that provides symmetrical control of the flow of traffic through the switch fabric helps ensure transparent switchover with no losses.
- Segmented and redundant out-of-band provisioning and management paths.
- Virtualization of the management plane via Virtual Device Contexts (vDC).
- Integrated diagnostics and protocol decoding with an embedded control plane packet analyzer; this is based on the Wireshark open source. (No additional licenses are required.)
- Fully decoupled control plane and data plane with no hardware forwarding on the module.
- Distributed forwarding architecture, enabling independent upgrades of the supervisor and fabric.
- With Central arbitration and VoQ, this enables for Unified Fabric.
- Transparent upgrade capacity and capability; designed to support 40-Gigabit and 100-Gigabit Ethernet.
- System locator and beacon LEDs for simplified operations.
- Dedicated out-of-band management processor for “lights out” management.

Connectivity Management Processor (CMP)

The supervisor incorporates an innovative dedicated connectivity management processor (CMP) to support remote management and troubleshooting of the complete system. The CMP provides a complete out-of-band management and monitoring capability independent from the primary operating system. The CMP enables *lights out* management of the supervisor module, all modules, and the Cisco Nexus 7000 Series system without the need for separate terminal servers with the associated additional complexity and cost. The CMP delivers the remote control through its own dedicated processor, memory, and boot flash memory and a separate Ethernet management port. The CMP can reset all system components, including power supplies; it can also reset the host supervisor module to which it is attached, enabling a complete system restart.

The CMP offer many benefits, including the following:

- Dedicated processor and memory, and boot flash.
- The CMP interface can reset all the system components, which include power, supervisor module, and system restart.

- An independent remote system management and monitoring capability enables *lights out* management of the system.
- Remote monitoring of supervisor status and initiation of resets that removes the need for separate terminal server devices for out-of-band management.
- System reset while retaining out-of-band Ethernet connectivity, which reduces the need for onsite support during system maintenance.
- Capability to remotely view boot-time messages during the entire boot process.
- Capability to initiate a complete system power shutdown and restart, which eliminates the need for local operator intervention to reset power for devices.
- Login authentication, which provides secure access to the out-of-band management environment.
- Access to supervisor logs that enables rapid detection and prevention of potential system problems.
- Capability to take full console control of the supervisor.
- Complete control is delivered to the operating environment.

Example 1-5 shows how to connect to the CMP interface and the available **show** commands available from the CMP interface. Also, note the escape sequence of "~," to get back to the main NX-OS interface. You can also connect from the CMP back to the CP module.

Example 1-5 *Connecting to the CMP Interface, Displaying Available **show** Commands*

```
N7010-1# attach cmp
Connected
Escape character is '~,' [tilde comma]

N7010-1-cmp5 login: admin
Password:
Last login: Tue Aug 11 23:58:12 2009 on ttyS1

N7010-1-cmp5# attach cp
This command will disconnect the front-panel console on this supervisor, and will
clear all console attach sessions on the CP - proceed(y/n)? y
N7010-1#

N7010-1# attach cmp
Connected
Escape character is '~,' [tilda comma]

N7010-1-cmp5 login: admin
Password:
Last login: Wed Aug 12 00:06:12 2009 on ttyS1
```

```
N7010-1-cmp5# show ?
  attach          Serial attach/monitor processes
  clock           Display current date
  cores           Show all core dumps for CMP
  cp              Show CP status information
  hardware        Show cmp hardware information
  interface       Display interface information
  line            Show cmp line information
  logging         Show logging configuration and contents of logfile
  logs            Show all log files for CMP
  processes       Show cmp processes information
  running-config  Current operating configuration
  sprom           Show SPROM contents
  ssh             SSH information
  system          Show system information
  users           Show the current users logged in the system
  version         Show cmp boot information
```

Telnet

NX-OS enables for Telnet server and client. The Telnet protocol enables TCP/IP terminal connections to a host. Telnet enables a user at one site to establish a TCP connection to a login server at another site and then passes the keystrokes from one device to the other. Telnet can accept either an IP address or a domain name as the remote device address.

Note Remember that the Telnet server is disabled by default in NX-OS.

The Telnet server is disabled by default on an NX-OS device. Example 1-6 demonstrates how to enable a Telnet server in NX-OS.

Example 1-6 *Enabling a Telnet Server in NX-OS*

```
N7010-1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
N7010-1(config)# feature telnet
N7010-1(config)# show telnet server
telnet service enabled
N7010-1(config)# copy running-config startup-config
[########################################] 100%
```

SSH

NX-OS supports SSH Server and SSH Client. Use SSH server to enable an SSH client to make a secure, encrypted connection to a Cisco NX-OS device; SSH uses strong encryption for authentication. The SSH server in Cisco NX-OS Software can interoperate with publicly and commercially available SSH clients. The user authentication mechanisms supported for SSH are Remote Authentication Dial-In User Service (RADIUS), Terminal Access Controller Access Control System Plus (TACACS+), and the use of locally stored usernames and passwords.

The SSH client application enables the SSH protocol to provide device authentication and encryption. The SSH client enables a Cisco NX-OS device to make a secure, encrypted connection to another Cisco NX-OS device or to any other device that runs the SSH server.

SSH requires server keys for secure communications to the Cisco NX-OS device. You can use SSH server keys for the following SSH options:

- SSH version 2 using Rivest, Shamir, and Adelman (RSA) public-key cryptography
- SSH version 2 using the Digital System Algorithm (DSA)

Be sure to have an SSH server key-pair with the appropriate version before allowing the SSH service. You can generate the SSH server key-pair according to the SSH client version used. The SSH service accepts two types of key-pairs for use by SSH version 2:

- The *dsa* option generates the DSA key-pair for the SSH version 2 protocol.
- The *rsa* option generates the RSA key-pair for the SSH version 2 protocol.

By default, Cisco NX-OS Software generates an RSA key using 1024 bits.

SSH supports the following public key formats:

- OpenSSH
- IETF Secure Shell (SECSH)

Example 1-7 demonstrates how to enable SSH server and configure the SSH server keys.

Example 1-7 *Enabling SSH Server and Configuring SSH Server Keys*

```
N7010-1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
N7010-1(config)# no feature ssh
XML interface to system may become unavailable since ssh is disabled
N7010-1(config)# ssh key rsa 2048
generating rsa key(2048 bits).....
```

```
..
generated rsa key
N7010-1(config)# feature ssh
N7010-1(config)# exit
N7010-1# show ssh key
**************************************
rsa Keys generated:Thu Aug 13 23:33:41 2009
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA6+TdX+ABH/mq1gQbfhhsjBmm65ksgfQb3Mb3qbwUbNlc
Aa6fjJCGdHuf3kJox/hjgPDChJOdkUXHjESlV59OhZP/NHlBrBq0TGRr+hfdAssD3wG5oPkywgM4+bR/
ssCzoj6jVG41tGmfPip4pr3dqsMzR21DXSKK/tdj7bipWKy1wSkYQzZwatIVPIXRqTJY7L9a+JqVIJEA
0QlJM1l0wZ5YbxccB2GKNKCM2x2BZl4okVgl80CCJg7vmn+8RqIOQ5jNAPNeb9kFw9nsPj/r5xFC1RcS
KeQbdYAjItU6cX1TslRnKjlWewCgIa26dEaGdawMVuftgu0uM97VCOxZPQ==

bitcount:2048
fingerprint:
1f:b7:a3:3b:f5:ca:a6:36:19:93:98:c7:37:ba:27:db
**************************************
could not retrieve dsa key information
**************************************
N7010-1# show ssh server
ssh version 2 is enabled
N7010-1(config)# username nxos-admin password C1sc0123!

N7010-1(config)# username nxos-admin sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEA6+TdX+ABH/mq1gQbfhhsjBmm65ksgfQb3Mb3qbwUbNlcAa6fjJCGdHu
f3kJox/hjgP
DChJOd-
kUXHjESlV59OhZP/NHlBrBq0TGRr+hfdAssD3wG5oPkywgM4+bR/ssCzoj6jVG41tGmfPip4pr3dqsMzR21
DXSKK/tdj7b
ip-
WKy1wSkYQzZwatIVPIXRqTJY7L9a+JqVIJEA0QlJM1l0wZ5YbxccB2GKNKCM2x2BZl4okVgl80CCJg7vmn+
8RqIOQ5jNAP
Neb9kFw9nsPj/r5xFC1RcSKeQbdYAjItU6cX1TslRnKjlWewCgIa26dEaGdawMVuftgu0uM97VCOxZPQ==
N7010-1(config)# show user-account
user:admin
        this user account has no expiry date
        roles:network-admin
user:nxos-admin
        this user account has no expiry date
        roles:network-operator
        ssh public key: ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEA6+TdX+ABH/mq1gQbfhhsjBmm65ksgfQb3Mb3qbwUbNlcAa6fjJCGdHu
f3kJox/hjgP
DChJOd-
kUXHjESlV59OhZP/NHlBrBq0TGRr+hfdAssD3wG5oPkywgM4+bR/ssCzoj6jVG41tGmfPip4pr3dqsMzR21
DXSKK/tdj7b
```

```
ip-
WKy1wSkYQzZwatIVPIXRqTJY7L9a+JqVIJEA0QlJM1l0wZ5YbxccB2GKNKCM2x2BZl4okVgl80CCJg7vmn+
8RqIOQ5jNAP
Neb9kFw9nsPj/r5xFC1RcSKeQbdYAjItU6cX1TslRnKjlWewCgIa26dEaGdawMVuftgu0uM97VCOxZPQ==
N7010-1(config)#
N7010-1# copy running-config startup-config
[########################################] 100%
N7010-1#
```

XML

NX-OS has a robust XML management interface, which can be used to configure the entire switch. The interface uses the XML-based Network Configuration Protocol (NETCONF) that enables you to manage devices and communicate over the interface with an XML management tool or a program. NETCONF is based on RFC 4741 and the NX-OS implementation requires you to use a Secure Shell (SSH) session for communication with the device.

NETCONF is implemented with an XML Schema (XSD) that enables you to enclose device configuration elements within a remote procedure call (RPC) message. From within an RPC message, you select one of the NETCONF operations that matches the type of command that you want the device to execute. You can configure the entire set of CLI commands on the device with NETCONF.

The XML management interface does not require any additional licensing. XML management is included with no additional charge.

XML/NETCONF can be enabled via a web2.0/ajax browser application that uses XML/NETCONF to pull all statistics off all interfaces on the Nexus 7000 running NX-OS in a dynamically updating table.

Figures 1-2, 1-3, and 1-4 demonstrate sample output from the XML/NETCONF interface.

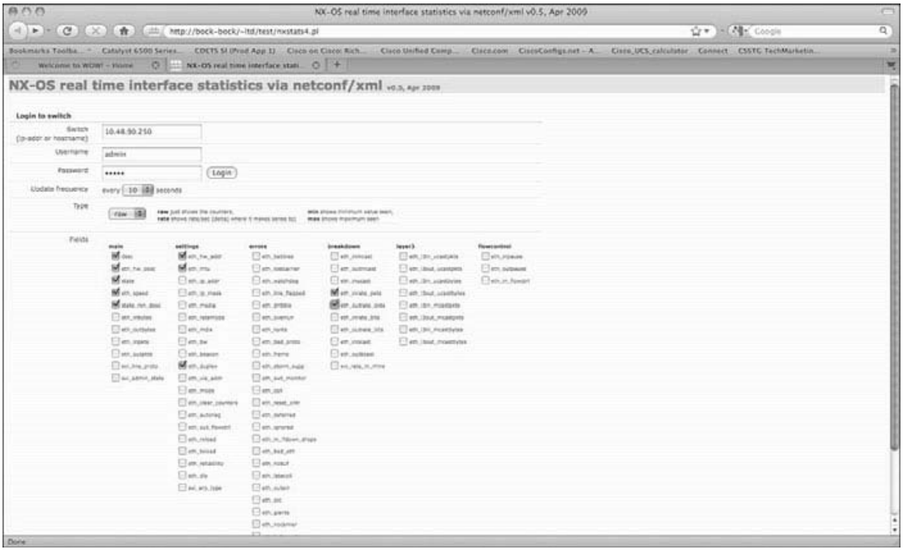
SNMP

The Simple Network Management Protocol (SNMP) is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language used for the monitoring and management of devices in a network.

SNMP has different versions such as SNMPv1, v2, and v3. Each SNMP version has different security models or levels. Most Enterprise customers are looking to implement SNMPv3 because it offers encryption to pass management information (or traffic) across the network. The security level determines if an SNMP message needs to be protected and authenticated. Various security levels exist within a security model:

- **noAuthNoPriv:** Security level that does not provide authentication or encryption.

- **authNoPriv:** Security level that provides authentication but does not provide encryption.
- **authPriv:** Security level that provides both authentication and encryption.



NX-OS real time interface statistics via netconf/xml v0.5, Apr 2009

http://bock-bock/~ltd/test/nxstats4.pl

NX-OS real time interface statistics via netconf/xml v0.5, Apr 2009

Login to switch

Switch (ip-addr or hostname): 10.48.90.250

Username: admin

Password: *****

Update frequency: every 10 seconds

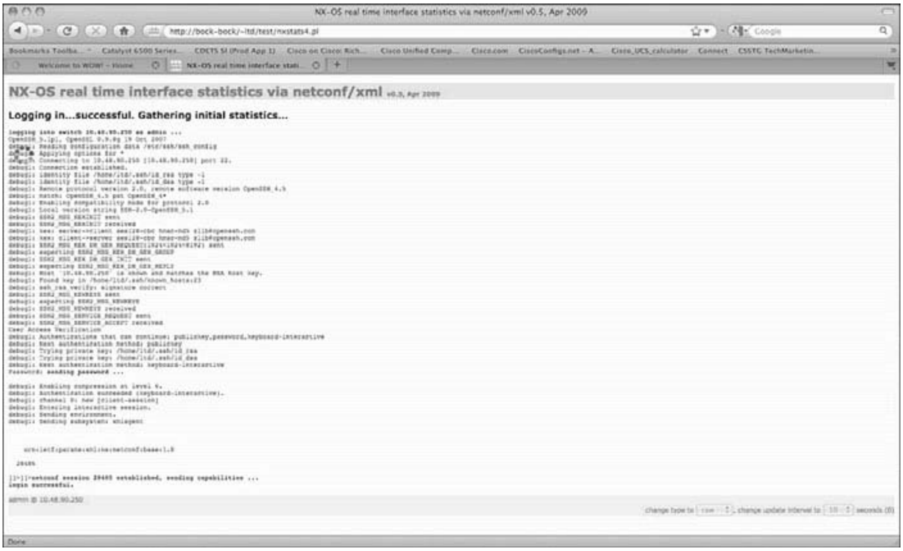
Type: raw

raw just shows the counters, **rate** shows rate/sec (delta) where t makes sense to; **min** shows minimum value seen, **max** shows maximum seen

Fields: main, settings, errors, breakdown, layer3, flowcontrol

Done

Figure 1-2 *Obtaining NX-OS Real-Time Interface Statistics via NETCONF/XML. The IP Address Entered Is the NX-OS mgmt0 Interface.*



NX-OS real time interface statistics via netconf/xml v0.5, Apr 2009

http://bock-bock/~ltd/test/nxstats4.pl

NX-OS real time interface statistics via netconf/xml v0.5, Apr 2009

Logging in...successful. Gathering initial statistics...

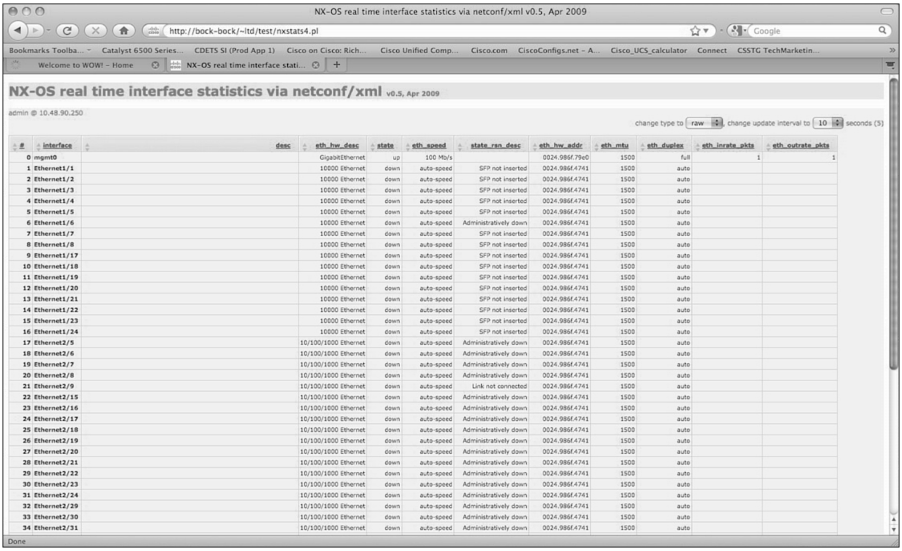
```
logging into switch 10.48.90.250 as admin ...
OpenSSH_5.1p1, OpenSSL 0.9.8g 19 Oct 2007
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: Applying options for *
debug1: Connecting to 10.48.90.250 [10.48.90.250] port 22.
debug1: Connection established.
debug1: identity file /home/ltd/.ssh/id_rsa type -1
debug1: identity file /home/ltd/.ssh/id_dsa type -1
debug1: Remote protocol version 2.0, remote software version OpenSSH_4.5
debug1: match: OpenSSH_4.5 pat OpenSSH_4*
debug1: Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-2.0-OpenSSH_5.1
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: server->client aes128-cbc hmac-md5 zlib@openssh.com
debug1: kex: client->server aes128-cbc hmac-md5 zlib@openssh.com
debug1: SSH2_MSG_KEX_DH_GEX_REQUEST(1024<1024<8192) sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_GROUP
debug1: SSH2_MSG_KEX_DH_GEX_INIT sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_REPLY
debug1: Host '10.48.90.250' is known and matches the RSA host key.
debug1: Found key in /home/ltd/.ssh/known_hosts:23
debug1: ssh_rsa_verify: signature correct
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug1: SSH2_MSG_SERVICE_ACCEPT received
User Access Verification
debug1: Authentications that can continue: publickey,password,keyboard-interactive
debug1: Next authentication method: publickey
debug1: Trying private key: /home/ltd/.ssh/id_rsa
debug1: Trying private key: /home/ltd/.ssh/id_dsa
debug1: Next authentication method: keyboard-interactive
Password: sending password ...

debug1: Enabling compression at level 6.
debug1: Authentication succeeded (keyboard-interactive).
debug1: channel 0: new [client-session]
debug1: Entering interactive session.
debug1: Sending environment.
debug1: Sending subsystem: xmlagent


   urn:ietf:params:xml:ns:netconf:base:1.0

  29485

]]>]]>netconf session 29485 established, sending capabilities ...
login successful.
```

admin @ 10.48.90.250

change type to raw, change update interval to 10 seconds (0)

Done

Figure 1-3 *Login Results to the NX-OS Devices via NETCONF/XML*



NX-OS real time interface statistics via netconf/xml v0.5, Apr 2009

admin @ 10.48.90.250

change type to raw, change update interval to 10 seconds (5)

#	interface	desc	eth_hw_desc	state	eth_speed	state_rsn_desc	eth_hw_addr	eth_mtu	eth_duplex	eth_inrate_pkts	eth_outrate_pkts
0	mgmt0		GigabitEthernet	up	100 Mb/s		0024.986f.79e0	1500	full	1	1
1	Ethernet1/1		10000 Ethernet	down	auto-speed	SFP not inserted	0024.986f.4741	1500	auto		
2	Ethernet1/2		10000 Ethernet	down	auto-speed	SFP not inserted	0024.986f.4741	1500	auto		
3	Ethernet1/3		10000 Ethernet	down	auto-speed	SFP not inserted	0024.986f.4741	1500	auto		
4	Ethernet1/4		10000 Ethernet	down	auto-speed	SFP not inserted	0024.986f.4741	1500	auto		
5	Ethernet1/5		10000 Ethernet	down	auto-speed	SFP not inserted	0024.986f.4741	1500	auto		
6	Ethernet1/6		10000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		
7	Ethernet1/7		10000 Ethernet	down	auto-speed	SFP not inserted	0024.986f.4741	1500	auto		
8	Ethernet1/8		10000 Ethernet	down	auto-speed	SFP not inserted	0024.986f.4741	1500	auto		
9	Ethernet1/17		10000 Ethernet	down	auto-speed	SFP not inserted	0024.986f.4741	1500	auto		
10	Ethernet1/18		10000 Ethernet	down	auto-speed	SFP not inserted	0024.986f.4741	1500	auto		
11	Ethernet1/19		10000 Ethernet	down	auto-speed	SFP not inserted	0024.986f.4741	1500	auto		
12	Ethernet1/20		10000 Ethernet	down	auto-speed	SFP not inserted	0024.986f.4741	1500	auto		
13	Ethernet1/21		10000 Ethernet	down	auto-speed	SFP not inserted	0024.986f.4741	1500	auto		
14	Ethernet1/22		10000 Ethernet	down	auto-speed	SFP not inserted	0024.986f.4741	1500	auto		
15	Ethernet1/23		10000 Ethernet	down	auto-speed	SFP not inserted	0024.986f.4741	1500	auto		
16	Ethernet1/24		10000 Ethernet	down	auto-speed	SFP not inserted	0024.986f.4741	1500	auto		
17	Ethernet2/5		10/100/1000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		
18	Ethernet2/6		10/100/1000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		
19	Ethernet2/7		10/100/1000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		
20	Ethernet2/8		10/100/1000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		
21	Ethernet2/9		10/100/1000 Ethernet	down	auto-speed	Link not connected	0024.986f.4741	1500	auto		
22	Ethernet2/15		10/100/1000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		
23	Ethernet2/16		10/100/1000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		
24	Ethernet2/17		10/100/1000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		
25	Ethernet2/18		10/100/1000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		
26	Ethernet2/19		10/100/1000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		
27	Ethernet2/20		10/100/1000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		
28	Ethernet2/21		10/100/1000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		
29	Ethernet2/22		10/100/1000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		
30	Ethernet2/23		10/100/1000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		
31	Ethernet2/24		10/100/1000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		
32	Ethernet2/29		10/100/1000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		
33	Ethernet2/30		10/100/1000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		
34	Ethernet2/31		10/100/1000 Ethernet	down	auto-speed	Administratively down	0024.986f.4741	1500	auto		

Figure 1-4 *Results of the Selected Attributes, Such as Speed, Duplex, Errors, Counters, MAC Address. The Page Refreshes Every 10 Seconds.*

Cisco NX-OS supports the following SNMP standards:

- **SNMPv1:** Simple community-string based access.
- **SNMPv2c:** RFC 2575-based group access that can be tied into RBAC model.
- **SNMPv3:** Enables for two independent security mechanisms, authentication (Hashed Message Authentication leveraging either Secure Hash Algorithm [SHA-1] or Message Digest 5 [MD5] algorithms) and encryption (Data Encryption Standard [DES] as the default and Advanced Encryption Standard [AES]) to ensure secure communication between NMS station and N7K/NX-OS. Both mechanisms are implemented as demonstrated in Example 1-8.

As NX-OS is truly modular and highly available, the NX-OS implementation of SNMP supports stateless restarts for SNMP. NX-OS has also implemented virtualization support for SNMP; NX-OS supports one instance of SNMP per virtual device context (VDC). SNMP is also VRF-aware, which allows you to configure SNMP to use a particular VRF to reach the network management host.

Example 1-8 demonstrates how to enable SNMPv3 on NX-OS.

Example 1-8 *Enabling SNMPv3 on NX-OS*

```
N7010-1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
N7010-1(config)# snmp-server user NMS auth sha Cisc0123! priv Cisc0123! engineID
```

```
00:00:00:63:00:01:00:10:20:15:10:03
N7010-1(config)# snmp-server host 10.100.22.254 informs version 3 auth NMS
N7010-1(config)# snmp-server community public ro
N7010-1(config)# snmp-server community nxos rw
N7010-1(config)# show snmp
sys contact:
sys location:
0 SNMP packets input
        0 Bad SNMP versions
        0 Unknown community name
        0 Illegal operation for community name supplied
        0 Encoding errors
        0 Number of requested variables
        0 Number of altered variables
        0 Get-request PDUs
        0 Get-next PDUs
        0 Set-request PDUs
        0 No such name PDU
        0 Bad value PDU
        0 Read Only PDU
        0 General errors
        0 Get Responses
45 SNMP packets output
        45 Trap PDU
        0 Too big errors
        0 No such name errors
        0 Bad values errors
        0 General errors
        0 Get Requests
        0 Get Next Requests
        0 Set Requests
        0 Get Responses
        0 Silent drops
Community            Group / Access      context    acl_filter
---------            --------------      -------    ----------
nxos                  network-admin
public                network-operator
______________________________________________________________
                  SNMP USERS
______________________________________________________________
User                          Auth  Priv(enforce) Groups
____                          ____  _____________ ______
admin                         md5   des(no)       network-admin
```

```
nxos-admin                    sha   des(no)       network-operator
______________________________________________________________
 NOTIFICATION TARGET USERS (configured  for sending V3 Inform)
______________________________________________________________
User                          Auth  Priv
____                          ____  ____
NMS                           sha   des
(EngineID 0:0:0:63:0:1:0:10:20:15:10:3)
SNMP Tcp Authentication Flag : Enabled.
-----------------------------------------------------------------------------------
Port Monitor : enabled
-----------------------------------------------------------------------------------
Policy Name  : default
Admin status : Not Active
Oper status  : Not Active
Port type    : All Ports
-----------------------------------------------------------------------------------
Counter          Threshold  Interval Rising Threshold event Falling Threshold
event In Use
-------          ---------  -------- ---------------- ----- ------------------ --
Link Loss        Delta      60       5                4     1                  4
Yes
Sync Loss        Delta      60       5                4     1                  4
Yes
Protocol Error   Delta      60       1                4     0                  4
Yes
Signal Loss      Delta      60       5                4     1                  4
Yes
Invalid Words    Delta      60       1                4     0                  4
Yes
Invalid CRC's    Delta      60       5                4     1                  4
Yes
RX Performance   Delta      60       2147483648       4     524288000          4
Yes
TX Performance   Delta      60       2147483648       4     524288000          4
Yes
-----------------------------------------------------------------------------------
SNMP protocol : Enabled
-------------------------------------------------------------------
Context                          [Protocol instance, VRF, Topology]

N7010-1# show snmp user
______________________________________________________________
```

```
                  SNMP USERS
______________________________________________________________

User                          Auth  Priv(enforce) Groups
____                          ____  _____________ ______
admin                         md5   des(no)       network-admin

nxos-admin                    sha   des(no)       network-operator

______________________________________________________________
 NOTIFICATION TARGET USERS (configured  for sending V3 Inform)
______________________________________________________________

User                          Auth  Priv
____                          ____  ____
NMS                           sha   des
(EngineID 0:0:0:63:0:1:0:10:20:15:10:3)
N7010-1(config)# exit
N7010-1# copy running-config  startup-config
[########################################] 100%
N7010-1#
```

DCNM

Cisco Data Center Network Manager (DCNM) is a management solution that supports NX-OS devices. DCNM maximizes the overall data center infrastructure uptime and reliability, which improves service levels. Focused on the operational management requirements of the data center, DCNM provides a robust framework and rich feature set that fulfills the switching, application, automation, provisioning, and services needs of today's data centers and tomorrow's data center requirements.

DCNM is a client-server application supporting a Java-based client-server application. The DCNM client communicates with the DCNM server only, never directly with managed Cisco NX-OS devices. The DCNM server uses the XML management interface of Cisco NX-OS devices to manage and monitor them. The XML management interface is a programmatic method based on the NETCONF protocol that complements the CLI functionality.

DCNM has a robust configuration and feature support on the NX-OS platform. The following features can be configured, provisioned, and monitored through DCNM enterprise management:

- Physical ports
- Port channels and virtual port channels (vPC)
- Loopback and management interfaces

- VLAN network interfaces (sometimes referred to as switched virtual interfaces [SVI])
- VLAN and private VLAN (PVLAN)
- Spanning Tree Protocol, including Rapid Spanning Tree (RST) and Multi-Instance Spanning Tree Protocol (MST)
- Virtual Device Contexts
- Gateway Load Balancing Protocol (GLBP) and object tracking
- Hot Standby Router Protocol (HSRP)
- Access control lists
- IEEE 802.1X
- Authentication, authorization, and accounting (AAA)
- Role-based access control
- Dynamic Host Configuration Protocol (DHCP) snooping
- Dynamic Address Resolution Protocol (ARP) inspection
- IP Source Guard
- Traffic storm control
- Port security
- Hardware resource utilization with Ternary Content Addressable Memory (TCAM) statistics
- Switched Port Analyzer (SPAN)

DCNM also includes end-end enterprise visibility including topology views, event browsers, configuration change management, device operating system management, hardware asset inventory, logging, and statistical data collection management.

Managing System Files

Directories can be created on bootflash: and external flash memory (slot0:, usb1:, and usb2:); you can also navigate through these directories and use them for files. Files can be created and accessed on bootflash:, volatile:, slot0:, usb1:, and usb2: file systems. Files can be accessed only on the system: file systems. Debug file system can be used for debug log files specified in the **debug** *logfile* command. System image files, from remote servers using FTP, Secure Copy (SCP), Secure Shell FTP (SFTP), and TFTP can also be downloaded.

File Systems

Table 1-1 outlines the parameters for the syntax for specifying a local file system, which is:

```
filesystem:[//module/]
```

Example 1-9 demonstrates some file system commands and how to copy a file.

Table 1-1 *Syntax for Specifying a Local File System*

File System Name	Module	Description
Bootflash	sup-active sup-local	Internal CompactFlash memory located on the active supervisor module used for storing image files, configuration files, and other miscellaneous files. The initial default directory is bootflash.
Bootflash	sup-standby sup-remote	Internal CompactFlash memory located on the standby supervisor module used for storing image files, configuration files, and other miscellaneous files.
slot0	Not applicable	External CompactFlash memory installed in a supervisor module used for storing system images, configuration files, and other miscellaneous files.
volatile	Not applicable	Volatile random-access memory (VRAM) located on a supervisor module used for temporary or pending changes.
Nvram	Not applicable	Nonvolatile random-access memory (NVRAM) located on a supervisor module used for storing the startup-configuration file.
Log	Not applicable	Memory on the active supervisor that stores logging file statistics.
system	Not applicable	Memory on a supervisor module used for storing the running-configuration file.
debug	Not applicable	Memory on a supervisor module used for debug logs.
usb1	Not applicable	External USB flash memory installed in a supervisor module used for storing image files, configuration files, and other miscellaneous files.
usb2	Not applicable	External USB flash memory installed in a supervisor module used for storing image files, configuration files, and other miscellaneous files.

Example 1-9 *File System Commands/Copying a File*

```
N7010-1# dir bootflash:
        311     Jun 20 05:15:05 2009  MDS20090619155920643.lic
        309     Jun 20 05:15:56 2009  MDS20090619155929839.lic
    2470887     Aug 01 08:13:35 2009  dp42
    8533440     Apr 17 23:17:14 2009  lacp_tech_all.log
     308249     Aug 01 09:08:39 2009  libcmd.so
        134     Jun 19 23:06:53 2009  libglbp.log
        175     Jun 20 04:14:22 2009  libotm.log
      49152     Jun 19 22:50:53 2009  lost+found/
   87081184     Jan 02 06:21:20 2008  congo-s1-dk9.4.0.2.bin
   87755113     Dec 11 13:35:25 2008  congo-s1-dk9.4.0.4.bin
   92000595     Apr 16 21:55:19 2009  congo-s1-dk9.4.1.4.bin
   92645614     Apr 08 06:08:35 2009  congo-s1-dk9.4.1.5.bin
   92004757     Jun 02 04:29:19 2009  congo-s1-dk9.4.1.5E2.bin
   99851395     Aug 03 05:17:46 2009  congo-s1-dk9.4.2.0.601.bin
  100122301     Aug 12 04:42:13 2009  congo-s1-dk9.4.2.1.bin
    9905740     Jan 02 06:21:29 2008  congo-s1-epld.4.0.2.img
    9730124     Dec 11 13:42:30 2008  congo-s1-epld.4.0.4.img
   23584768     Jan 02 06:21:26 2008  congo-s1-kickstart.4.0.2.bin
   23785984     Dec 11 13:34:37 2008  congo-s1-kickstart.4.0.4.bin
   24718848     Apr 16 21:52:40 2009  congo-s1-kickstart.4.1.4.bin
   25173504     Apr 08 06:00:57 2009  congo-s1-kickstart.4.1.5.bin
   23936512     Aug 03 05:03:13 2009  congo-s1-kickstart.4.1.5E2.bin
   25333248     Aug 03 05:18:37 2009  congo-s1-kickstart.4.2.0.601.bin
   25234944     Aug 12 04:40:52 2009  congo-s1-kickstart.4.2.1.bin
      12558     Aug 01 08:51:22 2009  shrun
     916893     Apr 17 23:23:03 2009  stp_tech.og
       4096     Dec 11 14:04:50 2008  vdc_2/
       4096     Dec 11 14:04:50 2008  vdc_3/
       4096     Dec 11 14:04:50 2008  vdc_4/
     592649     Apr 17 23:18:16 2009  vpc_tech.log
        942     Jul 10 09:45:27 2009  wireshark
Usage for bootflash://sup-local
  982306816 bytes used
  827592704 bytes free
 1809899520 bytes total
N7010-1# dir bootflash://sup-remote
      12349     Dec 05 02:15:33 2008  7k-1-vdc-all.run
       4096     Apr 04 06:45:28 2009  eem/
      18180     Apr 02 23:47:26 2009  eem_script.cfg
   99851395     Aug 03 05:20:20 2009  congo-s1-dk9.4.2.0.601.bin
  100122301     Aug 12 04:46:18 2009  congo-s1-dk9.4.2.1.bin
```

```
      19021     Apr 03 21:04:50 2009  eem_script_counters.cfg
      19781     Apr 05 23:30:51 2009  eem_script_iptrack.cfg
      29104     Jun 19 22:44:51 2009  ethpm_act_logs.log
          0     Jun 19 22:44:51 2009  ethpm_syslogs.log
        175     Jun 20 04:14:37 2009  libotm.log
      49152     Jun 19 22:38:45 2009  lost+found/
   87755113     Apr 07 23:54:07 2009  congo-s1-dk9.4.0.4.bin
   92000595     Apr 16 21:55:19 2009  congo-s1-dk9.4.1.4.bin
   92645614     Apr 08 06:08:35 2009  congo-s1-dk9.4.1.5.bin
   92004757     Jun 02 04:29:19 2009  congo-s1-dk9.4.1.5E2.bin
   10993389     Mar 22 04:55:13 2009  congo-s1-epld.4.1.3.33.img
   23785984     Apr 07 23:47:43 2009  congo-s1-kickstart.4.0.4.bin
   24718848     Apr 16 21:52:40 2009  congo-s1-kickstart.4.1.4.bin
   25173504     Apr 08 06:00:57 2009  congo-s1-kickstart.4.1.5.bin
   23936512     Jun 02 04:26:35 2009  congo-s1-kickstart.4.1.5E2.bin
   25333248     Aug 03 05:19:26 2009  congo-s1-kickstart.4.2.0.601.bin
   25234944     Aug 12 04:45:24 2009  congo-s1-kickstart.4.2.1.bin
        310     Sep 19 03:58:55 2008  n7k-rhs-1.lic
      12699     Jan 23 14:02:52 2009  run_vpc_jan22
      11562     Mar 13 07:52:42 2009  startup-robert-cfg
      16008     Mar 12 02:02:40 2009  startup-vss-cfg
      17315     Mar 19 06:24:32 2009  startup-vss-cfg_roberto_mar18
         99     Apr 04 06:51:15 2009  test1
       9991     Jun 19 23:12:48 2009  vdc.cfg
       4096     Jan 22 13:37:57 2009  vdc_2/
       4096     Jan 22 00:40:57 2009  vdc_3/

       4096     Sep 11 12:54:10 2008  vdc_4/
     111096     Dec 20 04:40:17 2008  vpc.cap
          0     Feb 03 08:02:14 2009  vpc_hw_check_disable
      18166     Apr 03 03:24:22 2009  vpc_vss_apr02
      18223     Apr 02 22:40:57 2009  vss_vpc_apr2

Usage for bootflash://sup-remote
  863535104 bytes used
  946364416 bytes free
 1809899520 bytes total
N7010-1# copy bootflash://sup
bootflash://sup-1/        bootflash://sup-active/   bootflash://sup-remote/
bootflash://sup-2/        bootflash://sup-local/    bootflash://sup-standby/

N7010-1# copy bootflash://sup-local/congo-s1-epld.4.0.4.img bootflash://sup-
remote/congo-s1-epld.4.0.4.img
N7010-1# dir bootflash://sup-remote
```

```
      12349     Dec 05 02:15:33 2008  7k-1-vdc-all.run
       4096     Apr 04 06:45:28 2009  eem/
      18180     Apr 02 23:47:26 2009  eem_script.cfg
      19021     Apr 03 21:04:50 2009  eem_script_counters.cfg
      19781     Apr 05 23:30:51 2009  eem_script_iptrack.cfg
      29104     Jun 19 22:44:51 2009  ethpm_act_logs.log
          0     Jun 19 22:44:51 2009  ethpm_syslogs.log
        175     Jun 20 04:14:37 2009  libotm.log
      49152     Jun 19 22:38:45 2009  lost+found/
   87755113     Apr 07 23:54:07 2009  congo-s1-dk9.4.0.4.bin
   92000595     Apr 16 21:55:19 2009  congo-s1-dk9.4.1.4.bin
   92645614     Apr 08 06:08:35 2009  congo-s1-dk9.4.1.5.bin
   92004757     Jun 02 04:29:19 2009  congo-s1-dk9.4.1.5E2.bin
   99851395     Aug 03 05:20:20 2009  congo-s1-dk9.4.2.0.601.bin
  100122301     Aug 12 04:46:18 2009  congo-s1-dk9.4.2.1.bin
    9730124     Aug 12 22:02:57 2009  congo-s1-epld.4.0.4.img
   10993389     Mar 22 04:55:13 2009  congo-s1-epld.4.1.3.33.img
   23785984     Apr 07 23:47:43 2009  congo-s1-kickstart.4.0.4.bin
   24718848     Apr 16 21:52:40 2009  congo-s1-kickstart.4.1.4.bin
   25173504     Apr 08 06:00:57 2009  congo-s1-kickstart.4.1.5.bin
   23936512     Jun 02 04:26:35 2009  congo-s1-kickstart.4.1.5E2.bin
   25333248     Aug 03 05:19:26 2009  congo-s1-kickstart.4.2.0.601.bin
   25234944     Aug 12 04:45:24 2009  congo-s1-kickstart.4.2.1.bin
        310     Sep 19 03:58:55 2008  n7k-rhs-1.lic
      12699     Jan 23 14:02:52 2009  run_vpc_jan22
      11562     Mar 13 07:52:42 2009  startup-robert-cfg
      16008     Mar 12 02:02:40 2009  startup-vss-cfg
      17315     Mar 19 06:24:32 2009  startup-vss-cfg_roberto_mar18
         99     Apr 04 06:51:15 2009  test1

       9991     Jun 19 23:12:48 2009  vdc.cfg
       4096     Jan 22 13:37:57 2009  vdc_2/
       4096     Jan 22 00:40:57 2009  vdc_3/
       4096     Sep 11 12:54:10 2008  vdc_4/
     111096     Dec 20 04:40:17 2008  vpc.cap
          0     Feb 03 08:02:14 2009  vpc_hw_check_disable
      18166     Apr 03 03:24:22 2009  vpc_vss_apr02
      18223     Apr 02 22:40:57 2009  vss_vpc_apr2

Usage for bootflash://sup-remote
  873283584 bytes used
  936615936 bytes free
 1809899520 bytes total
N7010-1#
```

Configuration Files: Configuration Rollback

The configuration rollback feature enables you to take a snapshot, or *checkpoint*, of the Cisco NX-OS configuration and then reapply that configuration to your device at any point without having to reload the device. Rollback allows any authorized administrator to apply this checkpoint configuration without requiring expert knowledge of the features configured in the checkpoint.

You can create a checkpoint copy of the current running configuration at any time. Cisco NX-OS saves this checkpoint as an ASCII file that you can use to roll back the running configuration to the checkpoint configuration at a future time. You can create multiple checkpoints to save different versions of your running configuration.

When you roll back the running configuration, you can trigger the following rollback types:

- **Atomic:** Implement the rollback only if no errors occur. This is the default rollback type.
- **Best-effort:** Implement a rollback and skip any errors.
- **Stop-at-first-failure:** Implement a rollback that stops if an error occurs.

When you are ready to roll back to a checkpoint configuration, you can view the changes that will be applied to your current running configuration before committing to the rollback operation. If an error occurs during the rollback operation, you can choose to cancel the operation or ignore the error and proceed with the rollback. If you cancel the operation, Cisco NX-OS provides a list of changes already applied before the error occurred. You need to clean up these changes manually.

Configuration rollback limitations are as follows:

- Allowed to create up to ten checkpoint copies per VDC.
- You are not allowed to apply a checkpoint file of one VDC into another VDC.
- You are not allowed to apply a checkpoint configuration in a nondefault VDC if there is a change in the global configuration portion of the running configuration compared to the checkpoint configuration.
- The checkpoint filenames must be 75 characters or less.
- You are not allowed to start a checkpoint filename with the word *auto*.
- You cannot name a checkpoint file with *summary* or any abbreviation of the word *summary*.
- Only one user can perform a checkpoint, rollback, or copy the running configuration to the startup configuration at the same time in a VDC.
- After execution of **write erase** and **reload** commands, checkpoints are deleted. You can use the **clear checkpoint database** command to clear out all checkpoint files.
- Rollback fails for NetFlow if during rollback you try to modify a record that is programmed in the hardware.

- Although rollback is not supported for checkpoints across software versions, users can perform rollback at their own discretion and can use the best-effort mode to recover from errors.
- When checkpoints are created on bootflash, differences with the running-system configuration cannot be performed before performing the rollback, and the system reports "No Changes."

Example 1-10 demonstrates how to create a configuration rollback.

> **Note** You need to make sure you are in the correct VDC. If you need to change VDCs, use the **switchto vdc** syntax.

Example 1-10 *Creating a Configuration Rollback*

```
N7010-1# checkpoint changes
..........Done
N7010-1# show diff rollback-patch checkpoint changes running-config
Collecting Running-Config
Generating Rollback Patch
Rollback Patch is Empty
N7010-1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
N7010-1(config)# no snmp-server user nxos-admin
N7010-1(config)# exit
N7010-1# show diff rollback-patch checkpoint changes running-config
Collecting Running-Config
Generating Rollback Patch
!!
no username nxos-admin sshkey ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEA6+TdX+ABH/mq1gQbfhhsjBmm65ksgfQb3Mb3qbwUbNlcAa6fjJCGdHu
f3kJ
ox/hjgPDChJOd-
kUXHjESlV59OhZP/NHlBrBq0TGRr+hfdAssD3wG5oPkywgM4+bR/ssCzoj6jVG41tGmfPip4pr3dqsMzR21
DXSK
K/tdj7bipWKy1wSkYQzZwatIVPIXRqTJY7L9a+JqVIJEA0QlJM1l0wZ5YbxccB2GKNKCM2x2BZl4okVgl80
CCJg
7vmn+8RqIOQ5jNAPNeb9kFw9nsPj/r5xFC1RcSKeQbdYAjItU6cX1TslRnKjlWewCgIa26dEaGdawMVuftg
u0uM
97VCOxZPQ==
no username nxos-admin
N7010-1# rollback running-config checkpoint changes
Note: Applying config in parallel may fail Rollback verification
Collecting Running-Config
Generating Rollback Patch
Executing Rollback Patch
```

```
Generating Running-config for verification
Generating Patch for verification
N7010-1# show snmp user nxos-admin
______________________________________________________________
                  SNMP USER
______________________________________________________________

User                          Auth  Priv(enforce) Groups
____                          ____  _____________ ______
nxos-admin                    sha   des(no)       network-operator

You can also enable specific SNMP traps:
N7010-1(config)# snmp-server enable traps eigrp
N7010-1(config)# snmp-server enable traps callhome
N7010-1(config)# snmp-server enable traps link
N7010-1(config)# exit
N7010-1#
```

Operating System Files

Cisco NX-OS Software consists of three images:

- The kickstart image, contains the Linux kernel, basic drivers, and initial file system.
- The system image contains the system software, infrastructure, Layers 4 through 7.
- The *Erasable Programmable Logic Device (EPLD)* image: EPLDs are found on the Nexus 7000 currently shipping I/O modules. EPLD images are not released frequently,\; even if an EPLD image is released, the network administrator is not forced to upgrade to the new image. EPLD image upgrades for I/O modules disrupt traffic going through the I/O module. The I/O module powers down briefly during the upgrade. The EPLD image upgrades are performed one module at a time.

On the Nexus 7000 with dual-supervisor modules installed, NX-OS supports in-service software upgrades (ISSU). NX-OS ISSU upgrades are performed without disrupting data traffic. If the upgrade requires EPLD to be installed onto the line cards that causes a disruption of data traffic, the NX-OS software warns you before proceeding so that you can stop the upgrade and reschedule it to a time that minimizes the impact on your network.

NX-OS ISSU updates the following images:

- Kickstart image
- System image
- Supervisor module BIOS
- Data module image

- Data module BIOS
- Connectivity management processor (CMP) image
- CMP BIOS

The ISSU process performs a certain sequence of events, as outlined here:

Step 1. Upgrade the BIOS on the active and standby supervisor modules and the line cards (data cards/nonsupervisor modules).

Step 2. Bring up the standby supervisor module with the new kickstart and system images.

Step 3. Switch over from the active supervisor module to the upgraded standby supervisor module.

Step 4. Bring up the old active supervisor module with the new kickstart image and the new system image.

Step 5. Upgrade the CMP on both supervisor modules.

Step 6. Perform nondisruptive image upgrade for line card (data cards/nonsupervisor modules), one at a time.

Step 7. ISSU upgrade is complete.

Virtual Device Contexts (VDCs)

The Nexus 7000 NX-OS software supports Virtual Device Contexts (VDCs), VDC(s) allow the partitioning of a single physical Nexus 7000 device into multiple logical devices. This logical separation provides the following benefits:

- Administrative and management separation
- Change and failure domain isolation from other VDCs
- Address, VLAN, VRF, and vPC isolation

Each VDC appears as a unique device and allows for separate Roles-Based Access Control Management (RBAC) per VDC. This enables VDCs to be administered by different administrators while still maintaining a rich, granular RBAC capability. With this functionalit, each administrator can define virtual routing and forwarding instance (VRF) names and VLAN IDs independent of those used in other VDCs safely with the knowledge that VDCs maintain their own unique software processes, configuration, and data-plane forwarding tables.

Each VDC also maintains an individual high-availability (HA) policy that defines the action that the system will take when a failure occurs within a VDC. Depending on the hardware configuration of the system, there are various actions that can be performed. In a single supervisor system, the VDC can be shut down, restarted, or the supervisor can be reloaded. In a redundant supervisor configuration, the VDC can be shut down, restarted, or a supervisor switchover can be initiated.

Note Refer to Chapter 6, "High Availability," for additional details.

There are components that are shared between VDC(s), which include the following:

- A single instance of the kernel which supports all of the processes and VDCs.
- Supervisor modules
- Fabric modules
- Power supplies
- Fan trays
- System fan trays
- CMP
- CoPP
- Hardware SPAN resources

Figure 1-5 shows the logical segmentation with VDCs on the Nexus 7000. A common use case is horizontal consolidation to reduce the quantity of physical switches at the data center aggregation layer. In Figure 1-5, there are two physical Nexus 7000 chassis; the logical VDC layout is also shown.

VDC Configuration

This section shows the required steps to creating a VDC; once the VDC is created, you will assign resources to the VDC. VDC(s) are always created from the default admin VDC context, VDC context 1.

Note The maximum number of VDCs that can be configured per Nexus 7000 chassis is four; the default VDC (VDC 1) and three additional VDC(s).

Example 1-11 shows how to configure the VDC core on Egypt.

Example 1-11 *Creating VDC "core" on Egypt*

```
egypt(config)# vdc core
Note:  Creating VDC, one moment please ...
egypt# show vdc
vdc_id  vdc_name                          state               mac
```

```
------  --------                          -----             ----------
1       egypt                             active         00:1b:54:c2:38:c1
2       core                               active        00:1b:54:c2:38:c2

egypt# show vdc core detail
vdc id: 2
vdc name: core
vdc state: active
vdc mac address: 00:1b:54:c2:38:c2
vdc ha policy: RESTART
vdc dual-sup ha policy: SWITCHOVER
vdc boot Order: 2
vdc create time: Mon Feb 22 13:11:59 2010
vdc reload count: 1
vdc restart count: 0
egypt#
```

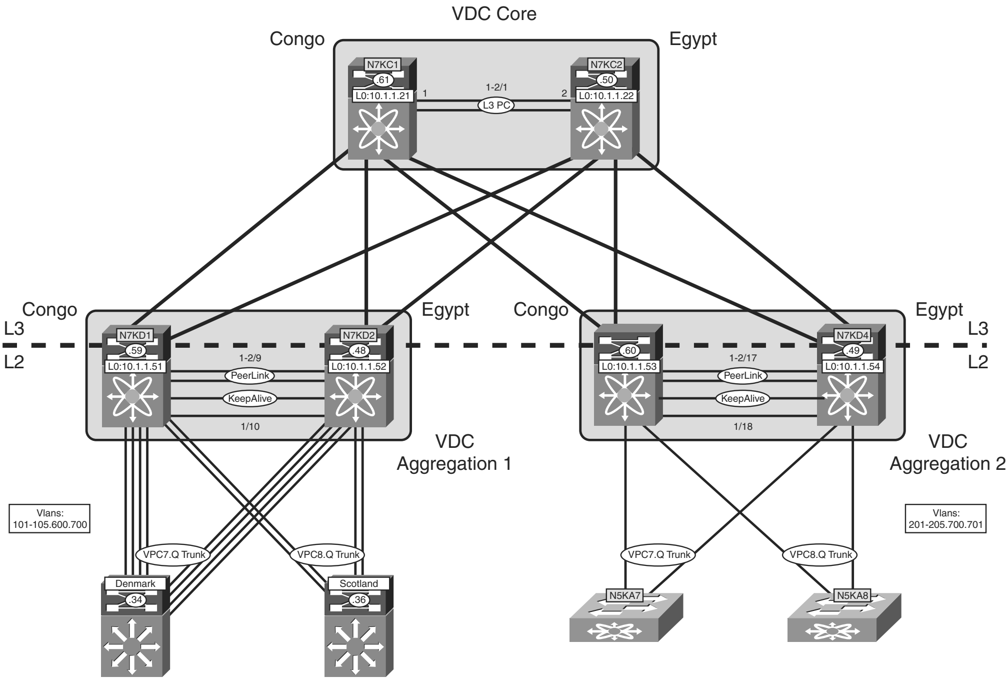


Figure 1-5 *Logical Segmentation with VDCs on the Nexus 7000*

Once the VDC is created, you now have to assign physical interfaces to the VDC. Depending on the Ethernet modules installed in the switch, interface allocation is supported as follows:

The 32-port 10-Gigabit Ethernet Module (N7K-M132XP-12), interfaces can be allocated on a per port-group basis; there are eight port-groups. For example, port-group 1 are interfaces e1, e3, e5, e7; port-group 2 are interfaces e2, e4, e6, e8.

The 48-port 10/100/1000 I/O Module (N7K-M148GT-11) can be allocated on a per-port basis.

The 48-port 1000BaseX I/O Module (N7K-M148GS-11) can be allocated on a per-port basis.

A future module, N7K-D132XP-15, interfaces will be allocated per 2 ports per VDC.

Note It is not possible to virtualize a physical interface and associate the resulting logical interfaces to different VDCs. A supported configuration is to virtualize a physical interface and associate the resulting logical interfaces with different VRFs or VLANs. By default, all physical ports belong to the default VDC.

Example 1-12 demonstrates how to allocate interfaces to a VDC.

Example 1-12 *Allocating Interfaces to a VDC*

```
egypt(config)# vdc core
eqypt(config-vdc)# allocate interface Ethernet1/17
egypt(config-vdc)# allocate interface Ethernet1/18
```

To verify the interfaces allocation, enter the show vdc membership command as demonstrated in Example 1-13.

Example 1-13 *Verifying Interface Allocation to a VDC*

```
egypt(config-vdc)# show vdc membership

vdc_id: 1 vdc_name: egypt interfaces:
        Ethernet1/26          Ethernet1/28          Ethernet1/30
        Ethernet1/32          Ethernet2/2           Ethernet2/4
        Ethernet2/6           Ethernet2/8           Ethernet2/26
        Ethernet2/28          Ethernet2/30          Ethernet2/32
        Ethernet3/4           Ethernet3/5           Ethernet3/6
        Ethernet3/7           Ethernet3/8           Ethernet3/9
        Ethernet3/11          Ethernet3/12          Ethernet3/13
        Ethernet3/14          Ethernet3/15          Ethernet3/16
        Ethernet3/17          Ethernet3/18          Ethernet3/19
        Ethernet3/20          Ethernet3/21          Ethernet3/22
        Ethernet3/23          Ethernet3/24          Ethernet3/25
        Ethernet3/26          Ethernet3/27          Ethernet3/28
        Ethernet3/29          Ethernet3/30          Ethernet3/31
        Ethernet3/32          Ethernet3/33          Ethernet3/34
        Ethernet3/35          Ethernet3/36          Ethernet3/39
        Ethernet3/40          Ethernet3/41          Ethernet3/42
        Ethernet3/43          Ethernet3/44          Ethernet3/45
        Ethernet3/46          Ethernet3/47          Ethernet3/48

vdc_id: 2 vdc_name: core interfaces:
        Ethernet1/17          Ethernet1/18          Ethernet1/19
        Ethernet1/20          Ethernet1/21          Ethernet1/22
        Ethernet1/23          Ethernet1/24          Ethernet1/25
        Ethernet1/27          Ethernet1/29          Ethernet1/31
        Ethernet2/17          Ethernet2/18          Ethernet2/19
        Ethernet2/20          Ethernet2/21          Ethernet2/22
        Ethernet2/23          Ethernet2/24          Ethernet2/25
        Ethernet2/27          Ethernet2/29          Ethernet2/31
        Ethernet3/1           Ethernet3/2           Ethernet3/3
        Ethernet3/10
```

In addition to interfaces, other physical resources can be allocated to an individual VDC, including IPv4 route memory, IPv6 route memory, port-channels, and SPAN sessions. Configuring these values prevents a single VDC from monopolizing system resources. Example 1-14 demonstrates how to accomplish this.

Example 1-14 *Allocating System Resources*

```
egypt(config)# vdc core
egypt(config-vdc)# limit-resource port-channel minimum 32 maximum equal-to-min
egypt(config-vdc)# limit-resource u4route-mem minimum 32 maximum equal-to-min
egypt(config-vdc)# limit-resource u6route-mem minimum 32 maximum equal-to-min
egypt(config-vdc)# limit-resource vlan minimum 32 maximum equal-to-min
egypt(config-vdc)# limit-resource vrf minimum 32 maximum equal-to-min
```

Defining the VDC HA policy is also done within the VDC configuration sub-mode. Use the ha-policy command to define the HA policy for a VDC as demonstrated in Example 1-15.

Example 1-15 *Changing the HA Policy for a VDC*

```
egypt(config)# vdc core
eqypt(config-vdc)# ha-policy dual-sup bringdown
```

The HA policy will depend based on the use-case or VDC role. For example, if you have dual-supervisor modules in the Nexus 7000 chassis or if the VDC role is development/test, the VDC HA policy may be to just shut down the VDC. If the VDC role is for the core and aggregation use case the HA policy would be switchover.

Troubleshooting

The troubleshooting sections introduce basic concepts, methodology, and general troubleshooting guidelines for problems that might occur when configuring and using Cisco NX-OS.

show Commands

Table 1-2 lists sample EXEC commands showing the differences between IOS and NX-OS.

Table 1-2 *Sample EXEC Commands Showing the Differences Between IOS and NX-OS*

Operation	IOS	NX-OS
Displays the running configuration	**show running-config**	**show running-config**
Displays the startup configuration	**show startup-config**	**show startup-config**
Displays the status of a specified port-channel interface	**show etherchannel #**	**show port channel #**
Displays the current boot variables	**show boot**	**show boot**
Displays all environmental parameters	**show environment**	**show environment**
Displays the percentage of fabric utilized per module	**show fabric utilization**	**show hardware fabric-utilization [detail]**
Displays the supervisors high-availability status	**show redundancy**	**show system redundancy status**
Displays CPU and memory usage data	**show process cpu**	**show system resources**
Displays specific VRF information	**show ip vrf** *name*	**show vrf** *name*

debug Commands

Cisco NX-OS supports an extensive debugging feature set for actively troubleshooting a network. Using the CLI, you can enable debugging modes for each feature and view a real-time updated activity log of the control protocol exchanges. Each log entry has a timestamp and is listed chronologically. You can limit access to the debug feature through the CLI roles mechanism to partition access on a per-role basis. Although the **debug** commands show real-time information, you can use the **show** commands to list historical and real-time information.

Caution Use the **debug** commands only under the guidance of your Cisco technical support representative because **debug** commands can impact your network/device performance.

Save **debug** messages to a special log file, which is more secure and easier to process than sending the **debug** output to the console.

By using the **?** option, you can see the options that are available for any feature. A log entry is created for each entered command in addition to the actual **debug** output. The **debug** output shows a timestamped account of the activity that occurred between the local device and other adjacent devices.

You can use the **debug** facility to track events, internal messages, and protocol errors. However, you should be careful when using the **debug** utility in a production environment because some options might prevent access to the device by generating too many messages to the console or creating CPU-intensive events that could seriously affect network performance.

You can filter out unwanted **debug** information by using the **debug-filter** command. The **debug-filter** command enables you to limit the **debug** information produced by related **debug** commands.

Example 1-16 limits EIGRP hello packet **debug** information to Ethernet interface 1/1.

Example 1-16 *Filtering debug Information*

```
switch# debug-filter ip eigrp interface ethernet 1/1
switch# debug eigrp packets hello</code>
```

Topology

Throughout the book, you see a common topology for demonstration purposes. Figure 1-6 depicts the physical topology.

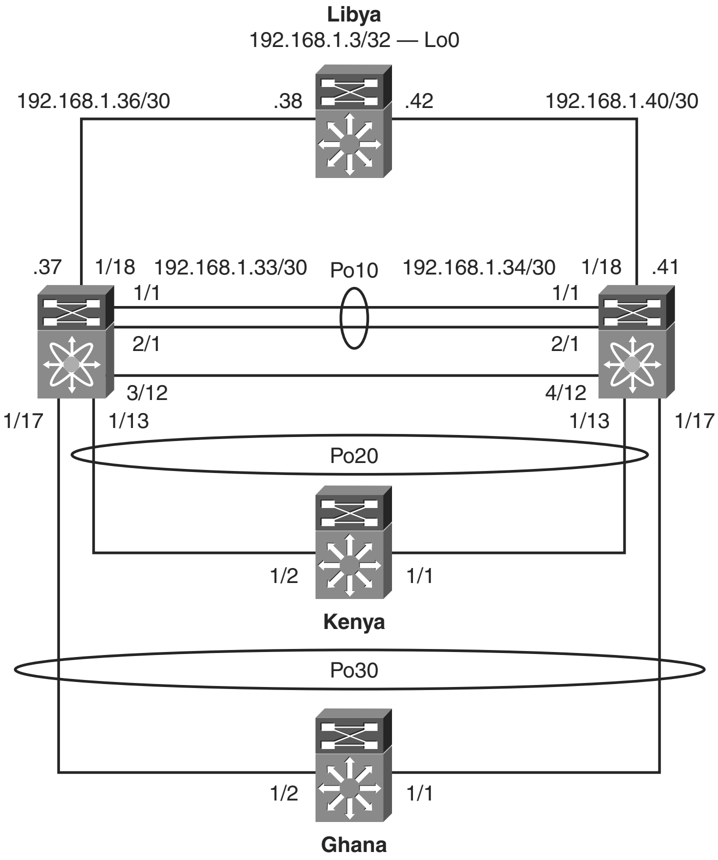


Figure 1-6 *Physical Topology for Book Demonstration Purposes*

Further Reading

NX-OS Feature Navigator: http://tinyurl.com/2btvax

NX-OS Nexus 7000 Supported MIB List: http://tinyurl.com/pzh4gg

NX-OS Nexus 5000 Supported MIB List: http://tinyurl.com/q4pqp5

NX-OS Nexus 1000V Supported MIB List: http://tinyurl.com/nu22mx

This page intentionally left blank

Index