Lab Manual

CCNA VOICE

ciscopress.com

Brent Sieling

FREE SAMPLE CHAPTER

SHARE WITH OTHERS

CCNA Voice Lab Manual

Brent Sieling

Cisco Press

800 East 96th Street

Indianapolis, Indiana 46240 USA

CCNA Voice Lab Manual

Brent Sieling

Published by:
Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

Printed in the United States of America

Second Printing: February 2014

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

ISBN-13: 978-1-58713-299-5

ISBN-10: 1-58713-299-0

Warning and Disclaimer

This book is designed to provide information about CCNA Voice. 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.

Publisher
Paul Boger

Associate Publisher
Dave Dusthimer

Business Operation Manager, Cisco Press
Anand Sundaram

Executive Editor
Mary Beth Ray

Managing Editor
Sandra Schroeder

Development Editor
Ellie Bru

Project Editor
Mandie Frank

Copy Editor
John Edwards

Technical Editor(s)
Brion Washington
Michael H. Valentine

Editorial Assistant
Vanessa Evans

Designer
Mark Shirar

Composition
Tricia Bronkella

Proofreader
Sheri Cain

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

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.

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
168 Robinson Road
#28-01 Capital Tower
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Europe Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: +31 0 800 020 0791
Fax: +31 0 20 357 1100

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.

About the Author

Brent Sieling is an instructor and program director for the two-year Network Specialist Associate Degree at Madison College. He has been teaching networking classes full-time since January 2006 and part-time for three semesters prior to that. He is the lead contact for the Academy Support Center and Instructor Training Center at Madison College's Cisco Networking Academy, providing support to over 30 high schools and technical colleges in the state of Wisconsin. Brent previously worked as a network specialist at the Madison Metropolitan School District, where he managed a network of over 50 schools. Brent currently holds the Cisco Certified Network Associate (CCNA), CCNA Voice, and CCNA Security certifications, and he was previously a Certified Novell Engineer (CNE). He recently completed the Cisco Academy Instructor Trainer Qualification (ITQ) process to become an Academy Instructor Trainer. Brent has been a regular presenter at the Cisco Academy conferences.

About the Contributing Authors

David Bateman is a Certified Cisco Systems instructor with more than 20 years of internetworking experience. David has always enjoyed sharing his knowledge and has been a Cisco instructor for Skyline-ATS since 2000. In addition to teaching he is involved in authoring courses and books including *Configuring Cisco Communication Manager & Unity Connection* released by Cisco Press. David is currently the director of Educational Services for Skyline-ATS. His years of real-world technical and business knowledge allow him to bring a unique perspective to the classroom, where he not only delivers critical technical knowledge but can also explain how technologies can be used to address various business needs.

Brian Morgan, CCIE No. 4865, is a Collaboration Architect with Cisco specializing in Unified Communications and Collaboration technologies. With over 20 years in the networking industry, he has performed in a number of roles, including network consultant, Certified Cisco Systems Instructor, and engineering director for a telecommunications company. When he's not spending time with his family, Brian enjoys working with local high school and college students enrolled in local Cisco Network Academy programs in North Texas.

About the Technical Reviewers

Michael Valentine has more than 15 years of experience in the IT field, specializing in Cisco networking and VoIP technologies. He is a freelance Cisco Certified Systems Instructor, currently associated with Skyline Advanced Technology Services. His accessible, humorous and effective teaching style has demystified Cisco for hundreds of students since he began teaching in 2002. He has developed courseware and lab guides both for Cisco Systems and third-party clients.

Mike received his Bachelor of Arts from the University of British Columbia and currently holds CCNA, CCNA Voice, CCDA, CCNP, CCVP, and CCSI certifications, among others.

Mike has authored, co-authored, and technically reviewed several Cisco Press titles.

Brion Washington has been working with Cisco VOIP since the 4.x release. His expertise is in Cisco Unified Communications Manager (CUCM) and Voice Gateways. He has authored the *CCVP GWGK Quick Reference* (978-1-58714-355-7), *CCNP Voice TVoice 642-427 Quick Reference* (978-1-58714-365-6), and *CCNP Voice CAPPS 642-467 Quick Reference* (978-1-58714-361-8), and served as a technical editor for multiple others. He currently works as a Senior Network Engineer based on the East coast.

Acknowledgments

I have many people to thank for helping put this book together. Thankfully for me, this is not the Academy Awards, where the orchestra can cut me off if I go on too long.

First is Mary Beth Ray of Cisco Press. I owe her thanks for her willingness to work with a first-time author to produce a lab book for the CCNA Voice certification. I owe her even more thanks for not sending the Cisco Press Ninja Assassins after me when I was late on my deadlines, even though I completely deserved it.

To Ellie Bru, Mandie Frank, and John Edwards of Cisco Press. Ellie and Mandie had the unfortunate task of working with a first-time author, trying to get me to complete everything (and in a readable format). John not only had to correct my writing, but also to translate it into proper English.

To David Bateman and Brian Morgan. Despite my years of experience with Cisco's router-based voice solution, there are parts of the voice world, such as Cisco Unified Presence Server, that I had not dabbled in. Their writing both helped to make this book more complete and kept the deadlines from slipping any further.

To Brion Washington and Michael Valentine. As technical reviewers, they both made sure that I was as accurate as possible in explanations, and more than once caught areas where I made mistakes.

To David Shonkwiler and Ken McCullough. As my dean and department head, respectively, they have made it possible to teach in one of the finest two-year schools in the state of Wisconsin, if not the nation. They have made sure that my students have access to the equipment and resources in the classroom to succeed in their future jobs.

To Cheryl Halle and Curt Chambers. As any instructor will tell you, his or her success is in large part due to the people who provide help, and Cheryl and Curt are the best at running our labs. The labs are so well run that the Cisco Academy has held instructor training qualifications here twice in the past year, and the Academy told us it could not have had an easier time.

There are many students who were guinea pigs for my labs over the past year and worked on my beta versions (and many alpha versions). They had to put up with my hands waving in the air as I explained how cool this stuff was, even the QoS portion. I was able to sneak their names into Appendix B, "IP Addressing and Phone Extensions for Cisco Unified Communications Manager Labs (Chapters 8–14 and 16–17)," along with many of my students who achieved their CCNA certification while at Madison College. However, there are two students who went above and beyond and deserve special recognition: Krzysztof Petrynko and John Endries. They both put in numerous hours checking my labs and giving me valuable feedback.

In addition to dedicating the book to my wife Megan, I also have to thank her for not only keeping things running at home (I promise to start cooking dinner more often now that this is done!), but also for doing the initial proofread of all my writing. (She was the one to point out that I needed to hyphenate *ever-patient* in the dedication.)

Finally, I have to thank Cisco itself and, by extension, the Cisco Networking Academy. There are few companies in the world that would spend millions of dollars to develop and maintain excellent curriculums and then give it to schools at no cost. My former students would not have their great jobs without the fantastic CCNA Exploration curriculum that the Academy offers. I have found great success in teaching the Academy curriculum and learned many new things along the way.

Dedication

This book is dedicated to my ever-patient wife Megan Schliesman and my wonderful daughter Lily (who will be mortified that I mentioned her name).

Contents at a Glance

Contents

Icons Used in This Book

Command Syntax Conventions

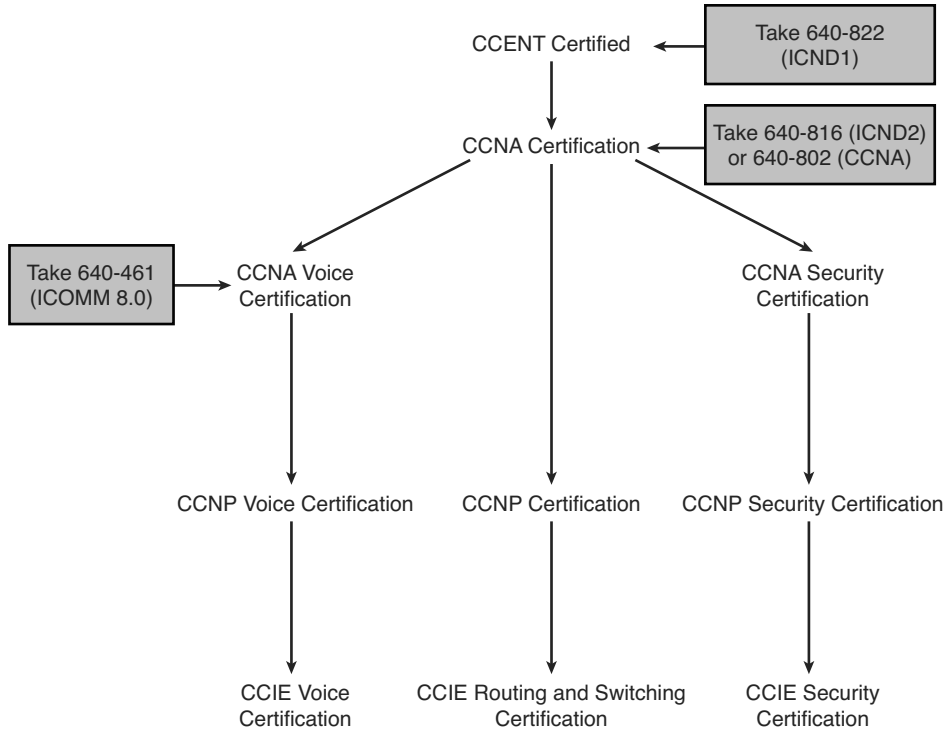
The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- **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.

Introduction

Cisco introduced the CCNA specialty exams in 2008 (CCNA Security, CCNA Voice, and CCNA Wireless) to provide a bridge between the CCNA certification and CCNP-level certifications, allowing networking professionals to get experience in a topic without having to complete the full CCNP-level track. The CCNA specialty exams also provide a common foundation for all the CCNP-level certification exams in a track. Cisco announced a revamped CCNA Voice certification in 2010 that focused more on the Cisco Unified Communications applications. Figure I-1 shows where the current 640-461 CCNA Voice exam fits in with the Cisco certification path.

Figure I-1 Cisco Certifications and CCNA Voice Certification Path



Goals and Methods

While many of the labs in this book can stand on their own, it is expected that this book will be used as a companion to the *CCNA Voice 640-461 Official Certification Guide* from Cisco Press (ISBN 978-1-58720-417-3). The Certification Guide has topics covered on the CCNA Voice certification exam that are not duplicated in this lab book.

Some of the labs (or items in a lab) in this book are marked as optional. In some cases, the lab is designed to match the material in the Certification Guide, but is not required for the CCNA Voice exam (such as Lab 6-1). In other cases, some labs can be completed and require optional hardware to test, but the key concepts can be learned without the hardware.

The goal of this lab book is to help students learn by doing. To that end, these labs contain far more than just step-by-step instructions to complete a task. The ideal scenario is that students understand *why* they are completing each step. When I first started writing labs, they were just simple lists of the commands to complete a task. As an instructor, I found that the minute the student encountered a problem or had something unexpected happen, up went his hand asking for help, as he had no idea

what to do. I started to realize that I was just training "typing monkeys," when my goal was to get students to *think* about why they were typing these commands. My labs grew longer as I added more descriptions and details.

Every time I had more than a few students get stuck or encounter a problem in one place, I would add information on how to avoid the problem or how to troubleshoot the issue in the lab. The result is the labs in this book. I found that most students love the detail I have put into the labs, and when they encounter a problem, the solution is often found nearby. However, I still have students in my classroom that just want to rush through a lab and skim it to find the commands that are printed in bold. Usually, by the fifth lab, they start to realize that I have commands that only apply in some circumstances, and the commands will not work in other cases. They learn to read the lab and carefully go through it. My goal is that they don't type commands without knowing what they are doing. They should "own" the process. After a student understands the process, he does not get stuck as easily and is instead able to figure out what to do to solve a problem.

In almost all cases, the labs build on one another for each of the two different Call Agents:

- For Cisco Unified Communications Manager Express (CUCME), the labs should go in order of Appendix D, "Configuring the Cisco Unified Communications Manager Express Router" (if necessary to load the router), Chapters 3–7, and then Chapter 15.
- For Cisco Unified Communications Manager (CUCM), the labs should go in order of Appendix C, "Building Cisco Unified Communications Manager (CUCM), Cisco Unity Connection (CUC), and Cisco Unified Presence (CUPS) Servers in VMware Workstation" (if necessary to build the servers), Chapters 8–14, Chapter 16, and then Chapter 17.

When proceeding through the labs, it is advisable to save device configurations (or back up servers in the case of CUCM) at the end of each lab.

Hardware Section

For the previous version of the CCNA Voice certification exam, you just needed a router with an IOS that included Cisco Unified Communications Manager Express (CUCME), or CME, as it commonly called, and some phones to learn the material. The current CCNA Voice 640-461 exam is more focused on knowing the functioning and configuration of Cisco Unified Communications applications, such as the Cisco Unified Communications Manager (CUCM), Cisco Unity Connection (CUC), and Cisco Unified Presence (CUPS). Having access to this software is important to understanding the topics for the certification exam. If you do not have access to the Cisco Unified Communications applications, study the figures included with the labs, as figures for the most common or difficult items to configure were included in the labs. Alternatively, check with your local Cisco Networking Academy, as it has the ability to offer classes using the Cisco Unified Communications applications at academic pricing.

Hardware and Software Used in the Labs

This is a list of equipment used to develop the labs. Use the following equipment to ensure the best compatibility. In a later section, alternatives will be explored.

Recommended Resources

These labs were created using the following equipment:

- Cisco 2811 router, running IOS c2800nm-adventerprisek9_ivs-mz.124-24.T2.bin, with a VWIC2-1MFT-T1/E1 card and a PVDM-16 (16-Channel Packet Voice/Fax DSP Module) installed. Some labs require two of these: one router for the HQ location and one for the branch office.
- Cisco WS-C3560-24PS switch with Power over Ethernet (PoE), running ISO c3560-ipservicesk9-mz.122-53.SE.bin.
- One Cisco router, with two or more VWIC2-1MFT-T1/E1 cards and enough digital signal processor (DSP) resources installed to act as a public switched telephone network (PSTN) simulator. (See Appendix E, "PSTN Simulator," for more details.)
- One or more Cisco 7900 Series IP Phones with three or more line buttons. (The Cisco 7962 is specifically demonstrated.)
- One or more Cisco IP Communicator (CIPC) soft phones running on one or more PCs or virtual machines.
- CUCM version 8.6.2 running in a virtual machine. (See Appendix C for more details.)
- CUC version 8.6.2 running in a virtual machine. (See Appendix C for more details.)
- CUPS version 8.6.3 running in a virtual machine. (See Appendix C for more details.)
- To run the Cisco Unified Communications applications on a PC or server, a machine with 8 or more gigabytes of RAM is required (16 gigs or more are recommended). The speed of the processor(s) on this machine is the biggest factor in how fast the virtual machines will perform. Virtualization software, such as VMware Workstation or ESXi Server, is needed.

Alternative Hardware Resources

While the labs were created using the previously listed equipment, in many cases, other equipment will be adequate.

All VoIP phone systems have a few common elements:

- Call Agents
- Phones
- Power for phones
- Network connectivity

Call Agents

The Call Agent is critical in the VoIP system. It controls all the functions of the phone calls. This lab book will use both Cisco Unified Communications Manager Express (CUCME), or CME as it commonly called, and CUCM for the Call Agent. CME is part of selected versions of the IOS on many, but not all, router models. Additionally, the routers will be needed for the gateways in some of the CUCM labs. As mentioned previously, the CUCM software is essential to understanding the topics in the CCNA Voice certification exam. CUCM version 8.6.2 was used to develop the labs, but any

CUCM 8.x version should be nearly identical for the purposes of these labs. Earlier versions, such as CUCM 6.x and 7.x can also be used, but they might require adjustments for items that are different.

The following lists are not completely exhaustive but cover the most common router hardware that supports CME.

Recommended router models:

- 2800 ISR Series (2801, 2811, 2821, 2851)
- 3800 ISR Series (3825 and 3845)
- 2900 ISR2 Series (2901, 2911, 2921, 2951)
- 3900 ISR2 Series (3925, 3925E, 3945, 3945E)
- 1861 Router

Models that will work, but are not as desirable (all of these might require RAM and flash upgrades to get the CME version of IOS to run):

- 2600XM Series (2610XM, 2611XM, 2620XM, 2621XM, 2650XM, 2651XM)
- 800 Series (891, 892, 887)
- 3700 Series (3725, 3745, 3770)
- 1700 Series (1760, 1751-V)
- 2691 Router
- UC500 Series (UC520, UC540, UC560) (CUCM will not support these as gateways, but they will work for the CME labs.)

To determine the IOS that supports CME, review the information in Appendix D, "Configuring the Cisco Unified Communications Manager Express Router."

Phones

The next items required are the voice endpoints, and there are many options to choose from. Following are the recommended models with comments about each:

- Cisco IP Communicator (also known as CIPC) is a soft phone that runs on Microsoft Windows XP, Vista, and 7, and it will work inside a virtual machine. It requires CME version 4.0 or later to work, so older routers with older versions of CME will not support it. An advantage of CIPC is that it does not require power like a desk phone, because the software runs on the PC or in a VM. A downside to CIPC is that it requires that a microphone and speakers be detected on the hardware, or the software will not launch. But it might be possible to trick the software with older audio drivers to believe that an audio source is present, even if one is not.
- 7900 IP Phone Series (any phone in this series will work, unless you are running an older version of CME). Older versions of the 7900 Series, such as the 7940 and 7960, can be purchased on the refurbished/secondary markets quite cheaply.
- 8900 IP Phone Series (any phone in this series will work, but it will require the newer versions of CME found in IOS 15 or later).
- 9900 IP Phone Series (any phone in this series will work, but it will require the newer versions of CME found in IOS 15 or later).

Power for Phones

(If you are using the CIPC, you can skip this section.) All desk phones require power to function. There are four ways to achieve this:

- The cheapest way to power phones is to use a "power brick" to provide power directly to the phones. (Note: The power supplies for Cisco access points will also deliver the –48V DC required to power the phones.) The model number for the Cisco power brick is CP-PWR-CUBE-3=. These are not commonly used in production environments, as the phone will not work if the power is out.
- Not much more expensive is a power injector, which puts Power over Ethernet (PoE) onto the wires between the switch and the phone. (Note: The PoE injectors for Cisco access points will also work to power the phones.) Power injectors are a common solution in a production environment where only one or two devices need power, as the injector can be plugged into the uninterruptible power supply (UPS) in the wiring closet and keeps the phones working when the power is out. Larger, rack-mount units that support dozens of phones are also sometimes used in production environments when upgrading to a PoE switch is not feasible. The model number is CP-PWR-INJ=.
- A switch that supports PoE (or for some older phones, Cisco-proprietary inline power). This solution is used the most often in production, as the switch should be powered by a UPS in the wiring closet and keeps the phones working when the power is out. In the classroom, students can share a switch for more than one pod by using VLANs and separate trunks to different routers. There are many models of switches that provide PoE, but some of the more common are as follows:
  - WS-C3560-24PS-S (24-port Layer 3 switch that also works for the CCNP material)
  - WS-C3560G-24PS-S (24-port Layer 3 switch with all gigabyte ports)
  - WS-C3560-8PC-S (8-port Layer 3 switch)
  - WS-C2960-24PC-L (24-port Layer 2 switch)
  - WS-C2960-24LT-L (24-port Layer 2 switch, but only eight ports with PoE)
  - WS-C3750-24PS-S (stackable 24-port Layer 3 switch)
  - WS-C3750-48PS-S (stackable 48-port Layer 3 switch)
- A router with a PoE switch module installed. (Note: Some models of routers, such as the 2800 and 2900 Series, *require* upgraded power supplies to support PoE switch modules. Make sure that your router has the upgraded power supply. Examples include the PWR-2801-AC-IP and PWR-2811-AC-IP for the 2801 and 2811 routers.) Also note that not all switch modules supply PoE. The modules that do are as follows:
  - HWIC-4ESW-POE
  - HWIC-4ESW with added ILPM-4 module
  - HWIC-D-9ESW-POE

Network Connectivity

If you are using a PoE switch or router with PoE switch modules, you are all set. If you used soft phones or desk phones with power bricks or injectors, a voice-enabled switch is required. A voice-enabled switch is one that allows voice VLANs to be configured. All the more recent switch models have this, including the 2950, 2960, 3550, 3650, and 3750 Series switches.

Additional Items

Several of the labs use an ISDN PRI connection to another router that is configured as a PSTN simulator (see Appendix E for the details). There are a few options for the PRI cards. Each of these cards will require digital signal processor (DSP) resources. For some routers, the DSPs can be added to the motherboard (2800, 3800, 2900, and 3900 Series), some can get the DSP from an Advanced Integration Module (AIM), and some network modules have slots to add DSPs. Various PRI cards that will work with the labs are as follows:

- VWIC-1MFT-T1 (Single-port voice card that only works as a T1. This will *not* work with the Cisco Configuration Professional software for CME.)
- VWIC-2MFT-T1 (Double-port voice card that only works as a T1. This will *not* work with the Cisco Configuration Professional software for CME.)
- VWIC2-1MFT-T1/E1 (Single-port voice card that will work for T1 or E1 configuration. This will work with the Cisco Configuration Professional software for CME.)
- VWIC2-2MFT-T1/E1 (Dual-port voice card that will work for T1 or E1 configuration. This will work with the Cisco Configuration Professional software for CME.)
- VWIC3-1MFT-T1/E1 (Single-port voice card that will work for T1 or E1 configuration. Note: This only works in newer 2900 or 3900 Series routers.)
- VWIC3-2MFT-T1/E1 (Dual-port voice card that will work for T1 or E1 configuration. Note: This only works in newer 2900 or 3900 Series routers.)
- NM-HDV2 (High-density network module that allows additional Voice/WAN Interface Card (VWIC) and DSP resources to be added to select routers.)
- NM-HDV2-1T1/E1 (High-density network module with a single T1/E1 port that allows additional VWIC and DSP resources to be added to select routers.)
- NM-HDV2-2T1/E1 (High-density network module with dual T1/E1 ports that allows additional VWIC and DSP resources to be added to select routers.)

While Lab 6-1 shows how to configure VIC2-2FXO and VIC2-2FXS analog voice cards, these are *not* necessary for the CCNA Voice certification exam. The only reason to have these interfaces (or similar analog connections) would be to use them to study for the first CCNP Voice certification exam, the 642-437 CVOICE v8.0, which includes analog interfaces.

Special Issues with CUCM and IP Phones

CUCM version 8.0 or higher includes Security by Design, which places a certificate on the phone that authenticates the phone to the TFTP server from which it gets the configuration file. As a result, the phone will not get a configuration from any other TFTP server that does not match the certificate. In a classroom with multiple users, or when going back to use CUCME, the certificate can cause problems with registering to other systems. There are two solutions to this problem:

- **Clearing the networking settings:** This might be enough to get the phone to register. On most phone models, press the **Settings** button and then navigate down to the Network Settings section (but do not enter the Network Settings). Press *** * #** (to unlock the settings) and then press the **Erase** key (you might need to press the **More** softkey to see the **Erase** key). The phone will then reset.

- **Removing the Initial Trust List (ITL) security file from the phone:** It might be necessary to clear the ITL file to upgrade or downgrade the firmware on a phone when switching between CUCM and CME. The easiest solution in a lab is to remove the file manually using the **Settings** button on the phone, as follows:
  - For the 7900 Series phones, press **Settings** and then navigate to **Security > Trust List > ITL File > * * #** (to unlock the settings) **> Erase** (you might need to press the **More** softkey to see the **Erase** key). The phone will then reset.
  - For the 8900/9900 Series phones, press **Settings** and then navigate to **Administrator Settings > Reset Settings > Security Settings**. The phone will then reset.

Search the Cisco support forums at http://supportforums.cisco.com for "Migrating IP Phones Between Clusters with CUCM 8 and ITL Files" for more details on ITL files and Security by Design.

CHAPTER 3

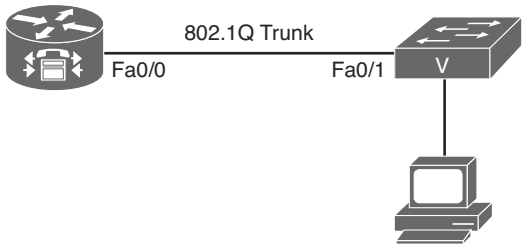
Establishing Network Connectivity and Understanding IP Phone Registration

In both Cisco Unified Communications Manager Express (CUCME) and Cisco Unified Communications Manager (CUCM) environments, phones need to connect to the network to receive services such as IP addresses from DHCP, VLAN assignments for voice traffic, IP information on where to register, and Network Time Protocol (NTP) packets. This chapter focuses on establishing those services.

In this chapter, you will set up a voice network for the fictitious company Shiny Objects Incorporated (SOI). Before phones are connected, you must establish network connectivity and configure needed services.

Lab 3-1: Network Connectivity

Figure 3-1 Topology Diagram



Equipment Required

This lab uses the following equipment:

- Cisco router
- Switch that supports voice VLANs
- PC for testing
- Cisco IP Phone (optional, but useful if switch supports Power over Ethernet [PoE])

Learning Objectives

Upon completion of this lab, you will be able to

- Perform basic router and switch configuration
- Configure VLANs to support data, voice, and network management traffic
- Configure VLAN trunking between a router and a switch using subinterfaces
- Configure router-based DHCP pools for voice and data devices

Scenario

SOI would like to establish its new data network with the expectation of using VoIP in the near future.

These instructions refer to the Pod Addressing Table in Appendix A to determine the IP addresses and VLAN numbers used for your pod. Wherever an *x* is shown, substitute the pod number.

Task 1: Clear and Cable Devices

Because the CCNA is a prerequisite for the CCNA-Voice certification exam, this book assumes that you are familiar with clearing prior configurations.

Step 1-1: Clear Prior Configurations

Clear any prior configuration on the router and switch, and delete the vlan.dat file before reloading both devices.

Step 1-2: Cable Router and Switch

Cable router interface Fast Ethernet 0/0 to switch port Fast Ethernet 0/1, as shown in Figure 3-1. The PC will be connected later.

Task 2: Configure Basic Setup

This task establishes the basic configuration commands on both the router and switch.

Step 2-1: Configure Basic Setup on the Router

Note: Not all devices support the **line vty 0 15** command. If your equipment does not support this command, change it to **line vty 0 4**.

```
Router(config)# hostname RtrPodx
```

For example, Pod 8 would use **hostname RtrPod8**.

```
RtrPodx(config)# no ip domain-lookup
RtrPodx(config)# enable secret class
RtrPodx(config)# line con 0
RtrPodx(config-line)# logging synchronous
RtrPodx(config-line)# exec-timeout 120 0
RtrPodx(config-line)# password cisco
RtrPodx(config-line)# login
RtrPodx(config-line)# line vty 0 15
RtrPodx(config-line)# password cisco
RtrPodx(config-line)# login
RtrPodx(config-line)# exit
```

Note: The **exec-timeout** command shown here is useful in a lab setting. It allows 120 minutes of inactivity before logging you out. (In a production environment, this could be a security risk.)

Step 2-2: Configure Basic Setup on the Switch

```
Switch(config)# hostname SwPodx
```

For example, Pod 3 would use **hostname SwPod3**

```
SwPodx(config)# no ip domain-lookup
SwPodx(config)# enable secret class
```

```
SwPodx(config)# line con 0
SwPodx(config-line)# logging synchronous
SwPodx(config-line)# exec-timeout 120 0
SwPodx(config-line)# password cisco
SwPodx(config-line)# login
SwPodx(config-line)# line vty 0 15
SwPodx(config-line)# password cisco
SwPodx(config-line)# login
SwPodx(config-line)# exit
```

Task 3: Configure the Switch

For the purposes of security and ease of implementing quality of service (QoS), use VLANs to keep voice traffic separate from other traffic.

Step 3-1: Create the VLANs

Create and name VLANs for data, voice, and network management.

```
SwPodx(config)# vlan x0
SwPodx(config-vlan)# name Data
SwPodx(config-vlan)# vlan x1
SwPodx(config-vlan)# name Management
SwPodx(config-vlan)# vlan x5
SwPodx(config-vlan)# name Voice
SwPodx(config-vlan)# exit
```

Step 3-2: Configure the Trunk Port

Configure the trunk port that connects the switch to the router. Layer 3 switches (such as the Cisco Catalyst 3560) require that the trunking protocol be specified with the **switchport trunk encapsulation** command before the interface can be set as a trunk. If you are using a Layer 2 switch (such as a Cisco Catalyst 2950 or 2960), the command is not needed and will be rejected.

Note: Cisco recommends in the "VLAN Security White Paper," to prevent a double-encapsulated 802.1Q/nested VLAN attack, "always pick an unused VLAN as the native VLAN of all the trunks; don't use this VLAN for any other purpose. Protocols like STP, DTP, and UDLD should be the only rightful users of the native VLAN and their traffic should be completely isolated from any data packets." For this reason, the management VLAN is not the native VLAN in this lab. To improve security, it would be better to create another VLAN as the native VLAN that will remain unused, but to simplify this lab, it is not covered.

```
SwPodx(config)# interface fastethernet 0/1
SwPodx(config-if)# switchport trunk encapsulation dot1q
SwPodx(config-if)# switchport mode trunk
SwPodx(config-if)# exit
```

Step 3-3: Configure the Access Ports

Almost all Cisco IP Phones are designed with a three-port switch built inside (one physical port connected to the production switch, one physical port for a PC to connect to the phone, and one internal port for the phone itself). This built-in switch saves money in wiring costs, as existing phone cabling might not meet networking standards. This enables an existing computer to be plugged into the phone, and the phone connects to the switch in the wiring closet.

Prior to the introduction of voice VLANs, a trunk connected an IP Phone to the switch to keep the voice and data traffic separate. Current best practice configures the ports connected to phones and PCs to use access mode but adds a secondary voice VLAN. The switch ports use the access VLAN to send data traffic as untagged frames. However, if the switch detects a Cisco IP Phone using Cisco Discovery Protocol (CDP), it will inform the phone of the VLAN used for voice traffic, which will be tagged using 802.1q. This creates a pseudotrunk that allows only the data and voice VLANs on the link.

Note: If CDP is disabled, or if you are using a non-Cisco IP phone, it requires setting the voice VLAN manually on the IP phone; otherwise, the voice traffic will end up on the data VLAN. For this reason, it is recommended that CDP remains enabled for ports that might have Cisco IP Phones connected.

Use the **interface range** command to assign settings. This is the fastest way to assign settings to more than one switch port at a time.

```
SwPodx(config)# interface range fastethernet 0/2 - 24
SwPodx(config-if-range)# switchport mode access
SwPodx(config-if-range)# switchport access vlan x0
SwPodx(config-if-range)# switchport voice vlan x5
SwPodx(config-if-range)# exit
```

Note: Setting the voice VLAN automatically enables **spanning-tree portfast**, so the switch port does not have to wait for Spanning Tree Protocol (STP) and goes active right away. You can verify this with the **show run** command, as shown in Example 3-1.

Example 3-1 Verify That spanning-tree portfast Is Created by the Voice VLAN Assignment

```
SwPod11# show run

<output omitted>

interface FastEthernet0/1
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface FastEthernet0/2
 switchport access vlan 110
 switchport mode access
 switchport voice vlan 115
 spanning-tree portfast

<output omitted>
```

Step 3-4: Configure the Switch Management Interface

Set up an interface to manage the switch remotely.

```
SwPodx(config)# interface vlan x1
SwPodx(config-if)# ip address 10.x1.0.2 255.255.255.0
SwPodx(config-if)# exit
SwPodx(config)# ip default-gateway 10.x1.0.1
```

Task 4: Configure the Router Subinterfaces

Subinterfaces allow the VLANs to cross a trunk link to the router. Each subinterface will be the default gateway for a paired subnet. When using subinterfaces on a router, it is necessary to assign the correct VLAN to the subinterface before an IP address can be entered. Because there are three VLANs, you need three subinterfaces.

Note: As covered in the note in Step 3-2, there is no native VLAN defined on the router.

Step 4-1: Configure the Data VLAN Subinterface

```
RtrPodx(config-if)# interface fastethernet 0/0.x0
RtrPodx(config-subif)# encapsulation dot1Q x0
RtrPodx(config-subif)# description Data VLAN
RtrPodx(config-subif)# ip address 10.x0.0.1 255.255.255.0
```

Step 4-2: Configure the Management VLAN Subinterface

```
RtrPodx(config-subif)# interface fastethernet 0/0.x1
RtrPodx(config-subif)# encapsulation dot1Q x1
RtrPodx(config-subif)# description Management VLAN
RtrPodx(config-subif)# ip address 10.x1.0.1 255.255.255.0
```

Step 4-3: Configure the Voice VLAN Subinterface

```
RtrPodx(config-subif)# interface fastethernet 0/0.x5
RtrPodx(config-subif)# encapsulation dot1Q x5
RtrPodx(config-subif)# description Voice VLAN
RtrPodx(config-subif)# ip address 10.x5.0.1 255.255.255.0
RtrPodx(config-subif)# exit
```

Step 4-4: Activate the Router Interface

```
RtrPodx(config)# interface fastethernet 0/0
RtrPodx(config-if)# no shutdown
```

Note: You might be thinking "What about IPv6?" CUCME does not support IPv6 until version 8.0, which requires router IOS version 15.0 or higher.

Task 5: Verification

Check the configuration to determine whether it matches what you expect. This will help to avoid future problems.

Step 5-1: Verify Switch VLAN Configuration

Use the **show vlan brief** command to verify the VLAN configuration. This output is from Pod 11; your output will have different VLAN numbers. Notice that Fa0/1 is a trunk port and as such does not have a VLAN assigned to it, so it will not show in the output.

```
SwPod11# show vlan
VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Gi0/1, Gi0/2
110  Data                              active    Fa0/2, Fa0/3, Fa0/4, Fa0/5
                                                 Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                 Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                 Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                 Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                 Fa0/22, Fa0/23, Fa0/24
111  Management                        active
115  Voice                             active    Fa0/2, Fa0/3, Fa0/4, Fa0/5
                                                 Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                 Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                 Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                 Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                 Fa0/22, Fa0/23, Fa0/24
1002 fddi-default                      act/unsup
1003 token-ring-default                act/unsup
1004 fddinet-default                   act/unsup
1005 trnet-default                     act/unsup
```

Step 5-2: Verify Switch Port Assignment

Use the **show interfaces switchport** command to verify the configuration of trunk and access ports. This output is from Pod 11; your output will have different VLAN numbers. Notice that Fa0/1 is a trunk port, while Fa0/2 is a static access port and has a voice VLAN assigned to it.

```
SwPod11# show interfaces switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
```

```
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none

<output omitted>

Name: Fa0/2
Switchport: Enabled
Administrative Mode: static access
Operational Mode: down
Administrative Trunking Encapsulation: negotiate
Negotiation of Trunking: Off
Access Mode VLAN: 110 (Data)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: 115 (Voice)

<output omitted>
```

Step 5-3: Verify Router Subinterface IP Assignment

Use the **show ip interface brief** command to verify that the trunk is assigned correctly. This output is from Pod 11; your output will have different subinterface and IP address numbers.

```
RtrPod11# show ip interface brief
Interface                  IP-Address      OK? Method  Status                Protocol
FastEthernet0/0            unassigned      YES unset   up                    up
FastEthernet0/0.110        10.110.0.1      YES manual  up                    up
FastEthernet0/0.111        10.111.0.1      YES manual  up                    up
FastEthernet0/0.115        10.115.0.1      YES manual  up                    up
```

Task 6: DHCP Services

Note: If you are using another source for DHCP, such as a Windows server or a CUCM server, you can skip this task. However, if the DHCP server is in a different subnet than the clients, it is necessary to use the **ip helper-address** command on each router subinterface to forward the DHCP requests to the server. Regardless of the DHCP server platform you use, make sure to configure the DHCP option 150 as discussed in this task.

While phones and PCs can be assigned IP addresses statically, DHCP can automatically assign IP address leases. Additionally, DHCP can provide additional information to clients, allowing them to locate necessary resources on the network at the same time they receive an IP address. Using the router as a DHCP server is a quick way to provide DHCP services to clients.

The DHCP option 150 tells Cisco IP Phones the IP address of the TFTP server with the initial configuration file. When using CUCME, the router is the TFTP server by default. This lab assigns the default gateway IP address as the option 150 address, as there is only one way to reach the call agent in this network.

Note: If there was redundancy in the network, it would be worthwhile to create a loopback interface and set the option 150 address to the loopback address, as that interface is always up.

Step 6-1: Configure DHCP Pools on the Router

Always enter the **ip dhcp exclude address** command before a DHCP pool is created. This avoids IP addresses that should be excluded from being assigned to devices. Enter the **network** statement as the last command in the pool. Otherwise, if devices are connected, they are assigned an IP address by DHCP right after the **network** statement is entered, even if the default gateway and option 150 are not configured. This can make troubleshooting difficult, as the PCs and phones will receive IP addresses, but the phones will not register and the PCs will not communicate outside their own subnet without the default router (gateway) address.

Create DHCP pools for both the data and voice networks. While it might seem that option 150 is irrelevant in data VLANs, with software on a PC able to emulate a phone (such as the Cisco IP Communicator software), it makes sense to include it for both DHCP pools.

```
RtrPodx(config)# ip dhcp excluded-address 10.x0.0.1 10.x0.0.10
RtrPodx(config)# ip dhcp pool Data
RtrPodx(dhcp-config)# default-router 10.x0.0.1
RtrPodx(dhcp-config)# option 150 ip 10.x0.0.1
RtrPodx(dhcp-config)# network 10.x0.0.0 255.255.255.0
RtrPodx(dhcp-config)# exit
RtrPodx(config)# ip dhcp excluded-address 10.x5.0.1 10.x5.0.10
RtrPodx(config)# ip dhcp pool Voice
RtrPodx(dhcp-config)# default-router 10.x5.0.1
RtrPodx(dhcp-config)# option 150 ip 10.x5.0.1
RtrPodx(dhcp-config)# network 10.x5.0.0 255.255.255.0
RtrPodx(dhcp-config)# exit
```

Task 7: Test and Cleanup

Step 7-1: Test Connectivity

Connect a PC to the switch. Verify that the PC is assigned an IP address from the 10.*x*0.0.0 /24 subnet. Verify that the PC can telnet to both the router and the switch management IP addresses. If not, troubleshoot the configuration.

Step 7-2: Save the Configurations

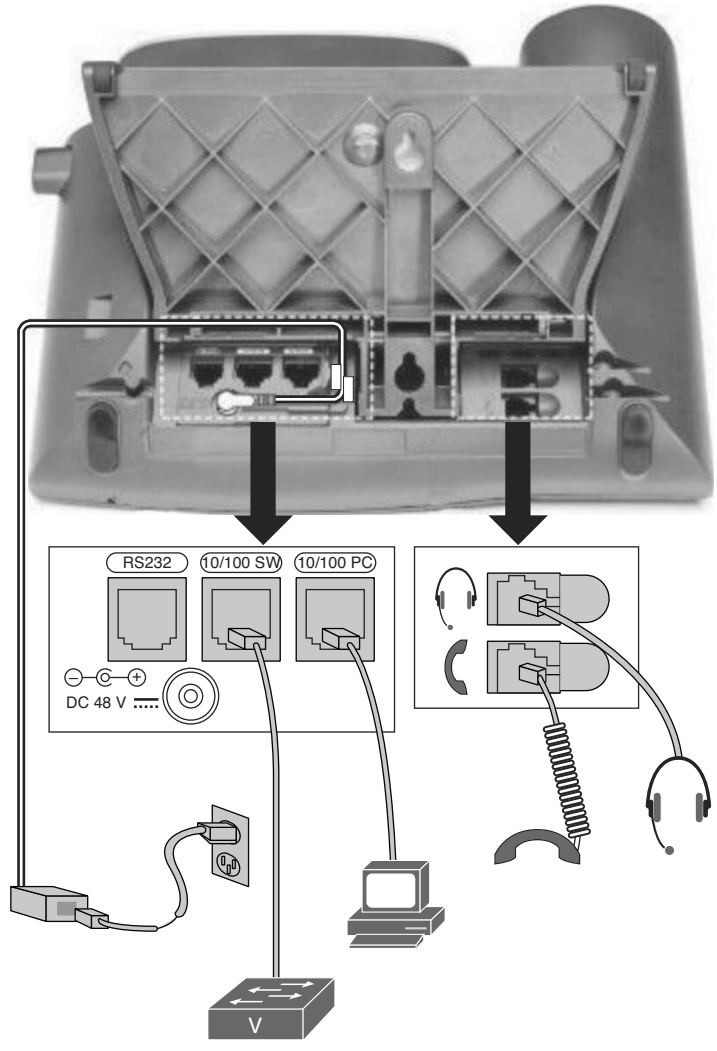
Save the configurations into a text file for both the router and switch. They will be needed for future labs.

Tip: When saving output from the console window, do not forget that some commands are not included and will not be present if pasted back to a device. Common examples include the **no shutdown** command for interfaces and VLAN creation and naming. To avoid problems, add missing commands to the text file or enter a reminder at the top of the text file. An exclamation point (!) at the start of a line makes it a comment for Cisco IOS, and this is an excellent way to add reminders to text output.

Step 7-3: (Optional) Explore Power Over Ethernet on the Switch

If you have a switch that is PoE capable and a Cisco IP Phone or two, monitor the console port of the switch and connect the jack on the phone labeled SW to a switch port. See Figure 3-2 for an example.

Figure 3-2 Cisco IP Phone Connections



If you connect an older Cisco IP Phone (a 7960, for example) that supports only Cisco-proprietary inline power, you might see a message like this on the switch console line:

```
*Mar  1 05:23:55.900: %ILPOWER-7-DETECT: Interface Fa0/3: Power Device detected:
  Cisco PD

*Mar  1 05:23:55.976: %ILPOWER-5-POWER_GRANTED: Interface Fa0/3: Power granted
```

If you connect a newer Cisco IP Phone (a 7975, for example) that supports the IEEE 802.1af standard, you might see a message like this on the switch console line:

```
* Mar  1 05:23:55.858: %ILPOWER-7-DETECT: Interface Fa0/4: Power Device detected:
  IEEE PD

Mar  1 05:23:55.942: %ILPOWER-5-POWER_GRANTED: Interface Fa0/4: Power granted
```

When verifying PoE usage or troubleshooting phone power problems, you can see the existing PoE usage with the **show power inline** command. Knowing the remaining PoE capacity is important, as Cisco sells some switch models that do not have enough PoE to fully power all ports, such as the 24-port Catalyst 2960-24LT-L that supports only eight PoE devices at 15.4 watts.

In this output from Pod 11, there are two Cisco IP Phones attached, consuming 18.3 watts, with 351.7 watts of PoE capacity left on this switch. The 7960 phone (6.3 watts) does not have a PoE class, as it does not support 802.1af, while the 7975 phone (12.0 watts) shows as an IEEE PoE Class 3 device.

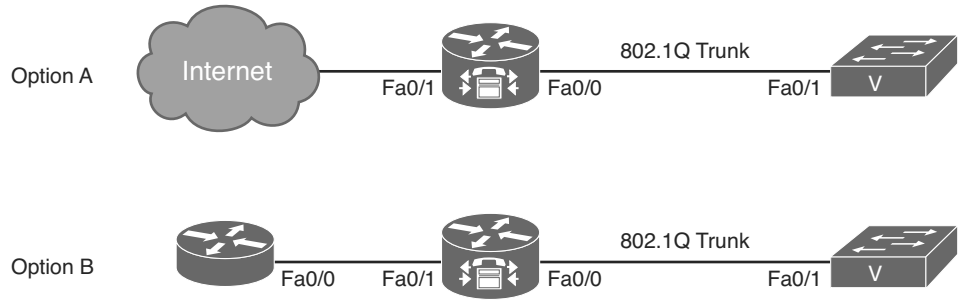
```
SwPod11# show power inline
Available:370.0(w)  Used:18.3(w)  Remaining:351.7(w)

Interface Admin  Oper       Power   Device              Class Max
                            (Watts)
--------- ------ ---------- ------- ------------------ ----- ----
Fa0/1     auto   off        0.0     n/a                 n/a   15.4
Fa0/2     auto   off        0.0     n/a                 n/a   15.4
Fa0/3     auto   on         6.3     IP Phone 7960       n/a   15.4
Fa0/4     auto   on         12.0    IP Phone 7975       3     15.4
Fa0/5     auto   off        0.0     n/a                 n/a   15.4
Fa0/6     auto   off        0.0     n/a                 n/a   15.4
<output omitted>
```

At this point, the phones should be attempting to register and will display a message such as "Registering" or "Configuring CM List" (the message will vary depending on the phone model and the version of firmware on the phone). If the phones show "Configuring IP" for more than a few seconds, the DHCP service is not functioning. The messages displayed on the phones are useful information when troubleshooting.

Lab 3-2: Network Time Protocol

Figure 3-3 Topology Diagram



Equipment Required

This lab uses the following equipment:

- Cisco router (and a second Cisco router if Internet access is not allowed from the first router)
- Switch that supports voice VLANs

Learning Objectives

Upon completion of this lab, you will be able to configure Network Time Protocol (NTP).

Scenario

SOI wants its new data network to use NTP to synchronize time for network devices.

NTP is not only important for synchronizing the time in network device event logs, but also for VoIP to show the correct time on the display of the phones and record the correct timestamp on voicemails, among other uses. The best way to keep everything synchronized is to use an NTP server to coordinate time.

This lab has instructions for two options:

- Option A assumes access to a production network that can reach an NTP server on the Internet.
- Option B configures another Cisco router to provide NTP time to simulate an Internet NTP server.

Note: The NTP server should not be a Microsoft Windows server running the W32Time service, as this uses Simple Network Time Protocol (SNTP), which is not as accurate as NTP and will not sync with most Cisco equipment.

These instructions refer to the Pod Addressing Table in Appendix A to determine the IP addresses and VLAN numbers used for your pod. Wherever an *x* is shown, substitute the pod number.

Task 1: NTP Services

Step 1-1: Load Prior Configurations

Use the configuration from Lab 3-1. If necessary, load the configuration for both the switch and router.

Step 1-2: (Optional) Configure Local Time Zone

NTP is calculated using UTC (Greenwich Mean Time), but you might want to see the time displayed on the router and phones using your local time zone.

Tip: Newer versions of the IOS have the 2007 updated U.S. Daylight Saving Time (DST) start and end dates included. If using an older IOS, or if you have a different DST at your location, you can enter the correct start and end dates as part of the command.

Note: The Cisco IOS does not provide help for time-zone naming conventions. Check Cisco.com for this information.

```
RtrPodx(config)# clock timezone timezone offset-from-GMT
```

For example, U.S. Central Daylight Time would use **clock timezone cdt -6.**

```
RtrPodx(config)# clock summer-time zone recurring
```

For example, U.S. Central Daylight Time would use **clock summer-time cdt recurring.**

Step 1-3: Manually Set the Clock

By manually setting the clock close to the correct time, you reduce the amount of time it takes to synchronize with the NTP server. Ideally, you should be within a minute or two of the correct time.

Use the privileged EXEC mode command **clock set** to manually set time:

```
RtrPodx# clock set hh:mm:ss day month year
```

For example, if the current day is Thursday, August 16, 2012 and the time is 9:40 p.m., you would enter **clock set 21:40:00 16 August 2012**.

Step 1-4 (Option A): Contact an NTP Server on the Internet

(Proceed to Step 1-4 [Option B] if you do not have access to the Internet.)

Configure an interface on the router to reach the Internet. The commands in this step assume that Fast Ethernet 0/1 is cabled to a production network with Internet access and a DHCP server that will assign IP addresses to the router.

```
RtrPodx(config)# interface fastethernet 0/1
RtrPodx(config-if)# ip address dhcp
RtrPodx(config-if)# no shutdown
RtrPodx(config-if)# exit
RtrPodx(config)# ip route 0.0.0.0 0.0.0.0 gateway-of-Fa0/1-network
```

A list of NTP stratum two servers can be found at www.ntp.org (http://support.ntp.org/bin/view/Servers/StratumTwoTimeServers). After selecting one close to you, use the **ntp server** command:

```
RtrPodx(config)# ntp server ntp_server_IP_address
```

Note: Make sure to use the IP address of the NTP server, instead of the Domain Name System (DNS) name, as the router is not configured to translate DNS names.

Skip Step 1-4 (Option B) and go to Step 1-5 to verify NTP operation.

Step 1-4 (Option B): Configure Another Cisco Router to Act as an NTP Server

(Skip this step if you completed Option A.)

The commands in this step assume that Fast Ethernet 0/1 on the voice router is cabled to another Cisco router. (A serial interface could also be used, but the Ethernet interfaces do not require any additional hardware.)

First, configure the other router to connect to the voice router.

```
Router(config)# hostname NTP_Server
NTP_Server(config)# interface fastethernet 0/0
NTP_Server(config-if)# ip address 192.168.0.1 255.255.255.0
NTP_Server(config-if)# no shutdown
NTP_Server(config-if)# exit
```

Set the time zones and clock on the NTP_Server router to match the VoIP router (as you did in Steps 1-2 and 1-3).

```
NTP_Server(config)# clock timezone timezone offset-from-GMT
NTP_Server(config)# clock summer-time zone recurring
NTP_Server# clock set hh:mm:ss day month year
```

Because you are configuring a "fake" NTP server, it is best to use a higher NTP stratum number to avoid conflicting with real NTP servers. Configure the NTP_Server router to be an NTP time source with the **ntp master** *stratum number* command.

Note: If this configuration is used to build the CUCM server (as shown in Appendix C), the CUCM server will not trust an NTP server with a stratum (distance from the atomic clock) of 7 or higher. To account for each device, such as the voice router (that adds 1 to stratum), the starting value is set to 4.

```
NTP_Server(config)# ntp master 4
```

Configure the VoIP router to connect to the NTP_Server router.

```
RtrPodx(config)# interface fastethernet 0/1
RtrPodx(config-if)# ip address 192.168.0.2 255.255.255.0
RtrPodx(config-if)# no shutdown
RtrPodx(config-if)# exit
RtrPodx(config)# ntp server 192.168.0.1
RtrPodx(config)# end
```

Go to Step 1-5 to verify NTP operation.

Step 1-5: Verify That the Time Is Synchronized

Use the following commands to verify that NTP is working:

```
RtrPodx# show ntp status
RtrPodx# show ntp association
RtrPodx# show ntp association detail
```

Note: It can take five to ten minutes to synchronize with the NTP server. To avoid overwhelming NTP servers, the router starts by polling the server every 64 seconds, and it takes several poll intervals for the router to establish confidence in the results.

In Examples 3-2a to 3-2g, the Pod 11 router is shown acquiring NTP time from another router. Your results will vary, but the descriptions will help you understand the various outputs you might see as the router uses NTP to synchronize.

The **show ntp associations** command output start with an "INIT" as the status, while the **show ntp associations detail** command output shows the server as "insane, invalid, unsynced" and the filter error is 16 (showing no polling data).

Example 3-2a Output Showing Pod 11 Router Acquiring NTP Time

```
RtrPod11# show ntp associations
  address          ref clock       st   when   poll reach  delay  offset   disp
 ~192.168.0.1      .INIT.          16      -     64     0  0.000   0.000 16000.
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
RtrPod11# show ntp associations detail
192.168.0.1 configured, insane, invalid, unsynced, stratum 16
ref ID .INIT., time 00000000.00000000 (18:00:00.000 CDT Thu Dec 31 1899)
our mode client, peer mode unspec, our poll intvl 64, peer poll intvl 1024
root delay 0.00 msec, root disp 0.00, reach 0, sync dist 16.00
delay 0.00 msec, offset 0.0000 msec, dispersion 16000.00
precision 2**24, version 4
org time 00000000.00000000 (18:00:00.000 CDT Thu Dec 31 1899)
rec time 00000000.00000000 (18:00:00.000 CDT Thu Dec 31 1899)
xmt time 00000000.00000000 (18:00:00.000 CDT Thu Dec 31 1899)
filtdelay =     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
filtoffset =    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
filterror =    16.00   16.00   16.00   16.00   16.00   16.00   16.00   16.00
minpoll = 6, maxpoll = 10
```

After the INIT phase is done (which can take a minute), the router shows the difference between the NTP server time and the time on the router. The router is now establishing the variation in time between the received time and local time. At the point the incrementing "when" counter equals the "poll" number, the NTP server will be queried again.

Example 3-2b Output Showing Pod 11 Router Acquiring NTP Time

```
RtrPod11# show ntp associations
  address          ref clock       st   when   poll reach  delay  offset   disp
 ~192.168.0.1      127.127.1.1     10     11     64     1  0.000 -348980 7937.5
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
RtrPod11# show ntp associations
  address          ref clock       st   when   poll reach  delay  offset   disp
 ~192.168.0.1      127.127.1.1     10     44     64     1  0.000 -348980 7937.5
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

In Example 3-2c, the router is 3,489.807 seconds off from the NTP server time. This phase will take several minutes.

Note: If the root dispersal value is above 1000, the router might not synchronize with the NTP server.

Example 3-2c Output Showing Pod 11 Router Acquiring NTP Time

```
RtrPod11# show ntp associations detail
192.168.0.1 configured, insane, invalid, stratum 10
ref ID 127.127.1.1   , time D22D35B2.32265329 (00:57:06.195 CDT Wed Sep 28 2011)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
root delay 0.00 msec, root disp 0.30, reach 1, sync dist 7.94
delay 0.00 msec, offset -3489807.6733 msec, dispersion 7937.50
precision 2**24, version 4
org time D22D35B8.CB5A7071 (00:57:12.794 CDT Wed Sep 28 2011)
rec time D22D435A.9A57C7DE (01:55:22.602 CDT Wed Sep 28 2011)
xmt time D22D435A.99DE9D3F (01:55:22.601 CDT Wed Sep 28 2011)
filtdelay =     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
filtoffset = -3489.8    0.00    0.00    0.00    0.00    0.00    0.00    0.00
filterror =     0.00   16.00   16.00   16.00   16.00   16.00   16.00   16.00
minpoll = 6, maxpoll = 10
```

The next phase is to accept the time from the NTP server and establish the accuracy of the local clock. Notice that the router is just over 2 milliseconds off from the NTP server. The "filteroffset" and "filtererror" are now 0 for the first poll.

Example 3-2d Output Showing Pod 11 Router Acquiring NTP Time

```
RtrPod11# show ntp associations detail
192.168.0.1 configured, insane, invalid, stratum 10
ref ID 127.127.1.1   , time D22D36A0.3225413B (01:01:04.195 CDT Wed Sep 28 2011)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
root delay 0.00 msec, root disp 0.21, reach 1, sync dist 7.94
delay 0.00 msec, offset 2.2946 msec, dispersion 7937.50
precision 2**24, version 4
org time D22D36A0.CC349A2D (01:01:04.797 CDT Wed Sep 28 2011)
rec time D22D36A0.CBD66632 (01:01:04.796 CDT Wed Sep 28 2011)
xmt time D22D36A0.CB603CB8 (01:01:04.794 CDT Wed Sep 28 2011)
filtdelay =     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
filtoffset =    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
filterror =     0.00   16.00   16.00   16.00   16.00   16.00   16.00   16.00
minpoll = 6, maxpoll = 10
```

Every 64 seconds, the router polls the NTP server again. You can see the polls show up, as the "filtererror" is gradually set to 0 for each new poll.

Example 3-2e Output Showing Pod 11 Router Acquiring NTP Time

```
RtrPod11# show ntp associations
192.168.0.1 configured, insane, invalid, stratum 10
ref ID 127.127.1.1   , time D22D36D2.32254796 (01:01:54.195 CDT Wed Sep 28 2011)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
root delay 0.00 msec, root disp 0.44, reach 3, sync dist 3.94
delay 0.00 msec, offset 3.1598 msec, dispersion 3937.73
precision 2**24, version 4
org time D22D36DF.CC672189 (01:02:07.798 CDT Wed Sep 28 2011)
rec time D22D36DF.CBD02AD0 (01:02:07.796 CDT Wed Sep 28 2011)
xmt time D22D36DF.CB5A1A5B (01:02:07.794 CDT Wed Sep 28 2011)
filtdelay =     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
filtoffset =    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
filterror =     0.00    0.00   16.00   16.00   16.00   16.00   16.00   16.00
minpoll = 6, maxpoll = 10
```

After enough polls have been completed, the time on the router is NTP synchronized. Your router now considers the NTP server "our master, sane, and valid."

Example 3-2f Output Showing Pod 11 Router Acquiring NTP Time

```
RtrPod11# show ntp associations detail
192.168.0.1 configured, our_master, sane, valid, stratum 10
ref ID 127.127.1.1   , time D22D3756.32286702 (01:04:06.195 CDT Wed Sep 28 2011)
our mode client, peer mode server, our poll intvl 64, peer poll intvl 64
root delay 0.00 msec, root disp 0.39, reach 17, sync dist 0.94
delay 0.00 msec, offset 3.1598 msec, dispersion 939.24
precision 2**24, version 4
org time D22D3760.CCE2CF70 (01:04:16.800 CDT Wed Sep 28 2011)
rec time D22D3760.CBD604CE (01:04:16.796 CDT Wed Sep 28 2011)
xmt time D22D3760.CB5F51F8 (01:04:16.794 CDT Wed Sep 28 2011)
filtdelay =     0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
filtoffset =    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
filterror =     0.00    0.00    0.00    0.00   16.00   16.00   16.00   16.00
minpoll = 6, maxpoll = 10
```

The asterisk (*) in front of the NTP server IP address shows that the server is synchronized. The **show ntp status** command also shows that the server is synchronized.

Example 3-2g Output Showing Pod 11 Router Acquiring NTP Time

```
RtrPod11# show ntp associations
  address         ref clock       st   when   poll reach  delay  offset   disp
*~192.168.0.1     127.127.1.1     10     21     64   377  0.000  15.598  4.689
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
RtrPod11# show ntp status
Clock is synchronized, stratum 11, reference is 192.168.0.1
nominal freq is 250.0000 Hz, actual freq is 249.9998 Hz, precision is 2**24
```

```
reference time is D22D3760.CBD604CE (01:04:16.796 CDT Wed Sep 28 2011)
clock offset is 0.0031 msec, root delay is 0.00 msec
root dispersion is 0.94 msec, peer dispersion is 0.44 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000000722 s/s
system poll interval is 64, last update was 99 sec ago.
```

Step 1-6: (Optional) Configure the Switch to Get NTP from the Router

For the sake of making sure that all networking devices are synchronized using NTP, the switch should use the router as an NTP source.

Note: To avoid overloading public NTP time servers, common practice has only a few edge devices at a company contact the public NTP servers, and all other company resources contact those edge devices.

```
SwPodx(config)# clock timezone timezone offset-from-GMT
SwPodx(config)# clock summer-time zone recurring
SwPodx(config)# ntp server 10.x1.0.1
```

Step 1-7: Save the Configurations

Save the configurations into a text file for both the router and switch. They will be needed for future labs.

Lab 3-3: Phone Boot/Registration Process

Equipment Required

No equipment is required for this lab.

Learning Objectives

Upon completion of this lab, you will better understand the Cisco IP Phone boot and registration process.

Task 1: Questions

The most important part of solving problems with VoIP solutions is knowing how a properly functioning system should work and comparing it to an existing problem. Cisco IP Phones have multiple steps to complete when connected to the network, and understanding the process from booting to registration will speed resolution of problems.

Question 3.1

List at least three ways that a Cisco IP Phone can receive power.

1. _____
2. _____
3. _____

Question 3.2

What two protocols can Cisco IP Phones use to register to the Call Agent (depending on the phone firmware loaded)?

1. _____
2. _____

Question 3.3

Number the following steps in the order they occur during the phone boot process:

___ The phone downloads the configuration file from the TFTP server.

___ The phone gets IP address information from DHCP, including option 150.

___ The phone registers with one or more Call Agents.

___ The phone receives power, which might involve receiving PoE from the switch.

___ The phone learns the VLAN information from CDP.