CCNP Practical Studies: Remote Access

ByWesley Shuo,Dmitry Bokotey,Raymond Morrow,Deviprasad Konda

Publisher: Cisco Press
Pub Date: December 22, 2003
ISBN: 1-58720-073-2
Pages: 528

Gain hands-on experience of CCNP Remote Access topics with lab scenarios for the new 642-821 BCRAN exam.

- Prepare for the CCNP 642-821 BCRAN exam and gain a better, practical understanding of exam concepts
- Experience how remote access concepts work in a real network with practice labs that walk you through their implementation
- Review set-up guides that show you how to prepare a lab for study
- Ready yourself for the new simulation-based questions on the CCNP exams

CCNP Practical Studies: Remote Access (CCNP Self-Study) prepares readers for the CCNP 642-821 BCRAN exam and for workplace challenges in implementing remote access network applications. Designed as a topic-by-topic guide of how to apply remote access concepts in a real network setting, this book is useful in preparing a CCNP candidate for the general exam questions by providing a better understanding of how remote access really works. It is also essential in preparing candidates for the new simulation-based questions that are on the Cisco certification exams. Finally, it serves anyone wanting a guide to real-world application of these concepts, regardless of certification interest.

Each chapter includes a review of the applicable technology, and guides the reader through implementation of the technology. This step-by-step process can be executed on a home- or office-based lab, a remote-accessible lab, some networking simulation software programs, or even as a stand-alone guide.

All of the topics on the new 642-821 BCRAN exam are covered, providing comprehensive exam preparation.

CCNP Practical Studies: Remote Access

ByWesley Shuo,Dmitry Bokotey,Raymond Morrow,Deviprasad Konda

Publisher: Cisco Press

Pub Date: December 22, 2003

ISBN: 1-58720-073-2

Pages: 528

Copyright

Published by:
Cisco Press
800 East 96th Street, 3rd Floor
Indianapolis, Indiana 46240 USA

Printed in the United States of America 1 2 3 4 5 6 7 8 9 0

First Printing December 2003

Library of Congress Cataloging-in-Publication Number: 2002111069

Warning and Disclaimer

This book is designed to provide information about remote-access technologies. Every effort has been made to make this book as complete and 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 authors and are not necessarily those of Cisco Systems, Inc.

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.

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 of the professional technical community.

Reader 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 e-mail at feedback@ciscopress.com. Please be sure to include the book title and ISBN in your message.

Corporate and Government Sales

Cisco Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact: U.S. Corporate and Government Sales 1-800-382-3419corpsales@pearsontechgroup.com

For sales outside of the U.S. please contact: International Sales 1-317-581-3793 international@pearsontechgroup.com

We greatly appreciate your assistance.

Publisher	John Wait
Editor-In-Chief	John Kane
Executive Editor	Brett Bartow
Cisco Representative	Anthony Wolfenden
Cisco Press Program Manager	Sonia Torres Chavez
Cisco Marketing Communications Manager	Scott Miller
Cisco Marketing Program Manager	Edie Quiroz
Managing Editor	Patrick Kanouse
Development Editor	Jill Batistick
Project Editor	Marc Fowler
Copy Editor	Gayle Johnson
Technical Editors	Henry Benjamin, Brian Feeny, Charles Ragan
Team Coordinator	Tammi Barnett
Book Designer	Gina Rexrode
Cover Designer	Louisa Adair
Production Team	Interactive Composition Corporation
Indexer	Larry Sweazy

Corporate 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: 408 526-4100

European Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: 31 0 20 357 1000
Fax: 31 0 20 357 1100

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-7660
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
Capital Tower
168 Robinson Road
#22-01 to #29-01
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Cisco Systems has more than 200 offices in the following countries and regions. Addresses, phone numbers, and fax numbers are listed on the Cisco.com Web site at www.cisco.com/go/offices.

Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China PRC • Colombia • Costa Rica • Croatia • Czech Republic • Denmark • Dubai, UAE • Finland • France • Germany • Greece • Hong Kong SAR • Hungary • India • Indonesia • Ireland • Israel • Italy • Japan • Korea • Luxembourg • Malaysia • Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines • Poland • Portugal • Puerto Rico • Romania • Russia • Saudi Arabia • Scotland • Singapore • Slovakia • Slovenia • South Africa • Spain • Sweden • Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • United States • Venezuela • Vietnam • Zimbabwe

Printed in the USA

Dedications

Wesley Shuo: I'd like to dedicate this book to my uncle and aunt, who passed away in 2001. Many thanks to my parents for always being there. To my sister, Eva, and brother, Jeff, for their continued support. To my best friends, Johnny, Daniel, and Robinson, for being my mentors.

To my dear wife, Flora, and two lovely daughters, Priscilla and Kristina, for putting up with me during the nights and weekends spent working on this book.

Dmitry Bokotey: I would like to dedicate this book to my wife, Alina, for her never-ending patience and support, for being here from the start, for never doubting any of my "silly" ideas, and for her smile that always brightens my day. Special thanks to my daughter, Alyssa, for bringing light and meaning to my existence every day.

Raymond Morrow: To my wife, Liz, for her support and belief in me to finish what I start, and to my children, Justin, Trey, Shelby, and Quentin, for never questioning the time I spent in front of my computer.

Deviprasad Konda: This book is dedicated to my parents, Ahobala and Vimala Raju Konda. Their love and dedication have built the foundation upon which I stand today.

About the Authors

Wesley Shuo, CCIE No. 4116, is a network design consultant with Cisco Systems. In this capacity, he provides IP Telephony (AVVID) consulting services and technical expertise to customers during the planning, design, implementation, and operation phases. Before his current position, he was a solutions consulting engineer in the Service Provider Line of Business at Cisco, where he gained extensive experience with various remote-access and WAN technologies, including ATM, WAN switching, DSL, MPLS, BGP, IS-IS, OSPF, RIP, VoIP, VoDSL, VPNs, and IPSec.

Dmitry Bokotey is a triple CCIE (No. 4460) in Routing and Switching, ISP Dial, and Security. He is one of the first professionals to achieve the new CCIE Security certification. Presently, he is a senior solution consultant for Cisco Systems, where he is responsible for the design and configuration of complex telecom and CLEC/ILEC customer networks. He has more than seven years of experience designing and managing large network installations. Careerbuilder.com recently labeled him "one of the world's top computer network engineers."

Raymond Morrow, CCIE No. 4146, CSS1, Cisco IP telephony design specialist, is currently employed at Northrop Grumman. Previously, he was a principal consultant with Computer Solutions, a San Antonio, Texas-based Cisco Silver Partner with Security and VPN Partner specialization. He has 16 years of experience in the networking arena and designs and implements various networking projects to a diverse customer base. Currently he is studying for his Security CCIE Lab Exam after having passed the Security CCIE Qualification Exam. He is the coauthor of *CCIE Practical Studies: Security*.

Deviprasad Konda is the lead support engineer for Qualcomm's corporate R&D business unit. He manages the Firewall and DMZ Infrastructure, Content Networking Infrastructure, and Quality of Service project teams. He is also part of the design engineering team for Core Backbone Evaluation and Corporate VPN and Remote Access Infrastructure projects. He has more than six years of experience designing and implementing Cisco router- and switch-based enterprise network architectures. He also has extensive network security expertise. He has a B.S. in computer engineering from Graceland University.

About the Technical Reviewers

Henry Benjamin, CCIE No. 4695, is a triple Cisco Certified Internet Expert, having certified Routing and Switching in May 1999, ISP Dial in June 2001, and Communications and Services in May 2002. He has more than ten years of experience in Cisco networks, including planning, designing, and implementing large IP networks running IGRP, EIGRP, BGP, and OSPF. Recently he worked for a large IT organization based in Sydney, Australia, as a key network designer, designing and implementing networks all over Australia and Asia. He is a former CCIE lab proctor.

Brian Feeny, CCIE No. 8036, is the senior network engineer for ShreveNet, Inc., an Internet service provider, where he has worked for the last seven years. He is also a partner in Netjam LLC, which specializes in sales and support of Cisco network equipment. He has more than 11 years of experience in the networking industry.

Charles Ragan, CCIE No. 1764, is an independent technology consultant. His background includes IP routing and switching, various voice over technologies (VoIP, VoFR), and many other desktop and related protocols. He has been in the information technology field for 19 years. His full technical biography can be found at http://www.geocities.com/ciscojock2002. He can be reached at ciscojock2002@yahoo.com.

Acknowledgments

Wesley Shuo: I would like to give special thanks to Brett Bartow and Cisco Press for giving me the once-in-a-lifetime opportunity to work on this book. Many thanks to Dmitry Bokotey, Raymond Morrow, and Deviprasad Konda for working extremely hard with me on this book. Special thanks to Adeel Ahmed for providing me with the lab resource and reviewing my chapter. Thanks to Cisco Systems and my manager, Himanshu Desai, for providing me with such a wonderful learning and working environment. Last but not least, I'd like to thank Dr. Helmut Epp, Dr. Gregory Brewster, Chonghua Zhang, and Ms. Terry Skwarek from DePaul University for giving me the education and environment to learn about the world of networking.

Dmitry Bokotey: Writing a book is never easy. Without help and guidance from my wife, Alina, and Cisco Press, it would have been impossible. This book is the product of collective effort. I would like to thank the lead author, Wesley Shuo, and coauthor Raymond Morrow, for their willingness to synchronize and compromise, and for their professionalism and knowledge. Again, I'm forever grateful to my wife, Alina, for help with writing and editing my second book.

I would also like to thank the team at Cisco Press, especially Brett Bartow, for believing in me and keeping all of us on track.

Big thanks to Cisco Systems' CCIE department, especially Kathe Saccenti, who helped me become a better engineer.

Finally, I want to thank my parents, Peter and Ida, for letting me spend days and nights beside my computer, no matter how pointless they thought it was.

Raymond Morrow: This project is a collaboration of four people with differing backgrounds. I would like to thank my coauthors, Wesley Shuo, Dmitry Bokotey, and Devi Konda, for their efforts and willingness to allow me to write this book with them.

Deviprasad Konda: My work on this book would not have been possible but for the help and guidance of several people.

I would like to thank my friend and coauthor, Wesley Shuo, for his help and support. I owe him an enormous amount of gratitude.

I would also like to thank my manager, Liren Chen, for his constant encouragement throughout the course of this endeavor. His upbeat attitude helped me immensely.

Last but not least, I would like to thank Brett Bartow at Cisco Press for his considerable patience after I missed deadline after deadline. I am very grateful to him for this opportunity.

Foreword

CCNP Practical Studies: Remote Access is designed to provide you with another vehicle to obtain hands-on experience, which is a critical component of any preparation program for the Cisco Certified Network Professional exams. The detailed lab scenarios contained in this book illustrate the application of key internetworking concepts covered on the CCNP BCRAN exam. They help you master the practical skills you need to build, configure, and troubleshoot a remote-access network to interconnect central sites to branch offices and small offices/home offices. With the introduction of performance-based testing elements to the CCNP BCRAN exam, these hands-on skills are of critical importance to succeeding on the exam and in your job as a CCNP professional.

Cisco and Cisco Press present this material in text-based format to provide another learning vehicle for our customers and the broader user community. A publication does not duplicate the instructor-led or e-learning environment, and we acknowledge that not everyone responds in the same way to the same delivery mechanism. It is our intent that presenting this material via a Cisco Press publication will enhance the transfer of knowledge to a broad audience of networking professionals.

Cisco Press will present lab manuals on existing and future exams through these *Practical Studies* titles to help achieve the Cisco Internet Learning Solutions Group's principal objectives: to educate the Cisco community of networking professionals and to enable that community to build and maintain reliable, scalable networks. The Cisco Career Certifications and classes that support these certifications are directed at meeting these objectives through a disciplined approach to progressive learning.

To succeed on the Cisco Career Certifications exams, as well as in your job as a Cisco certified professional, we recommend a blended learning solution that combines instructor-led, e-learning, and self-study training with hands-on experience. Cisco Systems has created an authorized Cisco Learning Partner program to provide you with the most highly qualified instruction and invaluable hands-on experience in lab and simulation environments. To learn more about Cisco Learning Partner programs available in your area, go to www.cisco.com/go/authorizedtraining.

The books that Cisco Press creates in partnership with Cisco Systems meet the same standards of content quality demanded of our courses and certifications. It is our intent that you will find this and subsequent Cisco Press certification and training publications of value as you build your networking knowledge base.

Thomas M. Kelly
Vice President, Internet Learning Solutions Group
Cisco Systems, Inc.

September 2003

Introduction

The Cisco Certified Network Professional (CCNP) program is one of the main certifications offered by Cisco Systems. For many network professionals, it is the logical step before they attempt the prestigious CCIE examination. You obtain CCNP certification by successfully passing four written tests, one of which is the Remote Access examination. This book is intended as a practical guide for candidates who are preparing for the Remote Access examination.

Achieving the CCNP certification can greatly enhance your career possibilities. The rigors of preparing for CCNP certification impart candidates with technical skills that are valuable to many organizations. Just as important, preparing for the certification also gives candidates the tools needed to design and maintain good networks. From a personal standpoint, becoming a CCNP is a milestone for candidates. It proves that they have the knowledge and dedication necessary to attempt and pass the four tests. After reading this book, we hope that you will be in a position to take and pass the Remote Access examination with confidence.

NOTE

The Remote Access examination (RMTAC 640-605) is 75 minutes long and has 50 to 60 questions. It is a computer-based exam and can be taken at any Sylvan Prometric site. You can contact Sylvan at 1-800-829-NETS or at www.2test.com.

Goals of This Book

This book's primary objective is to impart candidates with the practical knowledge needed to pass the Remote Access examination. Theoretical knowledge by itself is insufficient to pass a Cisco examination. You need practical knowledge to complement theory. Building functional, working networks is the best way to use theory and techniques to develop practical knowledge.

A network professional looking to improve his or her remote-access network skills can also use this book as an on-the-job reference. The lab exercises closely follow real-life scenarios to help candidates apply proven Cisco techniques in their work environments.

This book's main objective is to help you pass the Remote Access examination. To that end, it covers the topics you need to know without going into excessive detail. You can judge the areas in which you are weak and focus on them. The primary goal is to help you achieve the practical skills needed to be successful.

Audience

This book is focused on network professionals who are preparing for the CCNP Remote Access written examination. It is assumed that you have CCNA-level knowledge of routing protocols and WANs and working knowledge of remote-access technologies and protocols.

Each chapter begins with a brief overview that describes what the chapter is about. The main part of each chapter covers Scenarios that help you apply theoretical knowledge to real-life environments. The steps needed to configure and verify the Scenarios are laid out. Sample configurations and explanations also are included. You configure a Practical Exercise to test your knowledge of the material just covered. The accompanying Practical Exercise Solution helps you assess your familiarity with the topics. The Summary reviews the chapter's main points. Finally, the Review Questions further test your knowledge of the subjects covered.

The Practical Exercises are meant to emphasize the real-life aspect of the material. The Review Questions, on the other hand, are meant to test your theoretical knowledge of the topics. By putting these together, you will gain an understanding of the technologies and protocols needed to pass the Remote Access examination.

Organization

This book has 14 chapters and 1 appendix. As just described, they have a consistent structure, including an overview, Scenarios with detailed explanations, examples, Practical Exercises, and Review Questions. The chapters are as follows:

- Chapter 1, "Introduction to Remote Access," introduces the various types of remote-access technologies, networks, and their users.
- Chapter 2, "Building a CCNP Remote-Access Lab," covers creating LANs and WANs, as well as asynchronous, ISDN, PPP, DDR, dial backup, AAA, and security labs.
- Chapter 3, "Modem Connections and Operation Overview," covers modem operation, communication, and configuration. Basic and automatic modem configurations are covered in detail.
- Chapter 4, "Using Cable Modems to Access a Central Site," covers cable modems and their configuration. It contains an overview of cable modem technology, including transmission systems, protocols, and technology issues. The configuration of headend and CPE equipment is also covered.
- Chapter 5, "Configuring Point-to-Point Protocol and Controlling Network Access," covers the configuration of PPP. Basic PPP features and operation are described. The configuration covers PPP callback, authentication, and compression.
- Chapter 6, "Using ISDN and DDR Technologies to Enhance Remote Connectivity," covers the basic use of ISDN. Included is an overview of ISDN, including the different kinds of network equipment, ISDN bandwidth, and channels. The process of call setup and teardown is examined. The configuration of ISDN PRI and BRI is examined, including some optional configurations such as Multilink PPP.
- Chapter 7, "Optimizing the Use of DDR with Interface Dialer Profiles and Rotary Groups," covers the more-advanced topic of ISDN in a DDR scenario. The topic of DDR is covered, and DDR configuration is demonstrated. This chapter also covers the optimization of DDR interfaces using features such as dialer groups and dialer profiles.
- Chapter 8, "Using DSL to Access a Central Site," covers the basic use of DSL. It includes an overview of the various flavors of DSL. The different Cisco products in the DSL space are covered. The configuration section covers DSLAM configuration at Layer 2 and PPPoE and PPPoA configuration at Layer 3.
- Chapter 9, "Frame Relay Connectivity and Traffic Flow Control," covers the important topic of Frame Relay. It offers an overview of Frame Relay, including Frame Relay basics and signaling. The configuration of Frame Relay subinterfaces and traffic shaping is demonstrated. Issues and solutions relating to these topics are also covered.
- Chapter 10, "Enabling a Backup to the Permanent Connection," covers the configuration and use of dial backup. The basic theory and operation are discussed, including the various options such as physical versus dialer interfaces and load sharing versus load balancing.
- Chapter 11, "Managing Network Performance with Queuing and Compression," covers queuing and compression and their impact on network performance. Queuing basics are covered, including the various flavors and their operation. The configuration covers the use of Weighted Fair Queuing, priority queuing, and custom queuing. Data compression is

discussed and its configuration demonstrated.

- Chapter 12, "Scaling IP Addressing with Network Address Translation," covers the use and configuration of NAT. The concept of NAT and its components are discussed. The configuration section covers the topics of static NAT, dynamic NAT, and Port Address Translation.
- Chapter 13, "Using AAA to Scale Access Control in an Expanding Network," covers the concept of AAA and Cisco's Cisco Secure product. An overview of AAA and its individual components is given. The Cisco Secure product is examined from both the client and server perspective. Then AAA is configured on both client and server using Cisco Secure.
- Chapter 14, "Securing Remote-Access Networks," covers the configuration and use of VPNs in a remote-access scenario. The overview covers the different components of IPSec, including ESP, AH, and IKE. The various Cisco products in this space are also described. The configuration section covers the different VPN configurations, including router-to-router, VPN client-to-router, and VPN client-to-PIX.
- Appendix A, "Answers to Review Questions," provides answers to the chapter-ending review questions.

How Best to Use This Book

This book emphasizes a practical approach to study. Convenient access to equipment is a big plus, because you can easily follow the examples in the book. However, this luxury is unavailable to many people. Therefore, you don't need the equipment to get the full benefit from this book. Complete configurations are shown in every chapter so that you can get a good understanding of the concepts involved. The troubleshooting sections help you find your way out of potential problems. The command output examples show you what a successful end result looks like.

Equipment

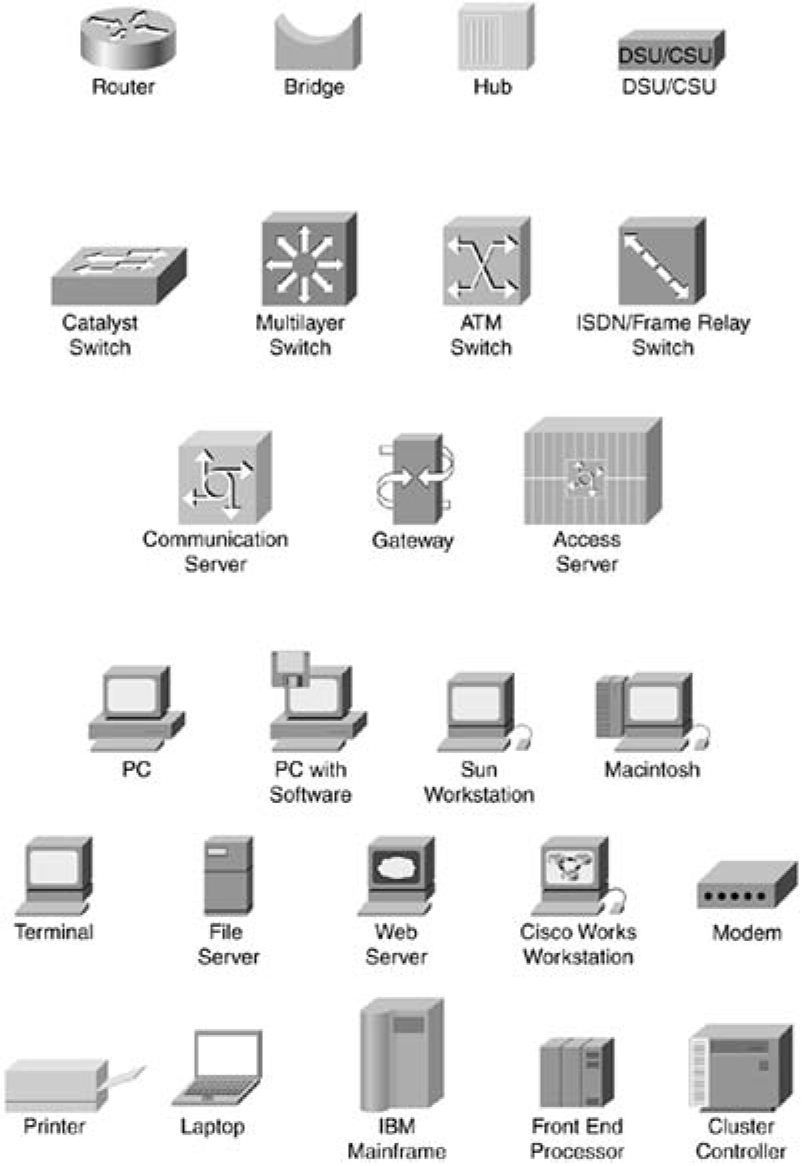
There are many places where you can obtain equipment. The ideal situation is if your place of employment has a lab or spare equipment you can use. If this is not the case, the Internet is a great place for you to find reasonably priced equipment. Also, a number of resellers and Cisco partners sell equipment. Alternatively, many simulators can simulate real-life networks. Cisco's Cisco Interactive Mentor (CIM) is one such product. To find out more about CIM, visit www.ciscopress.com.

Summary

The prestigious CCNP certification has become increasingly popular. It can be a stepping-stone for further achievements, such as the CCIE certification. It shows that you have the skill and dedication required to succeed in the networking industry. In that regard, this book is meant to help you attain that goal. It has been designed to help you take and pass the Remote Access examination.

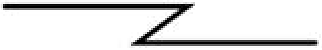
For many, the end of one journey signifies the beginning of another. Successfully achieving the CCNP certifica-tion can inspire you to goals you might not have thought of before. We hope this book helps you in that quest.

Icons Used in This Book



Line: Ethernet

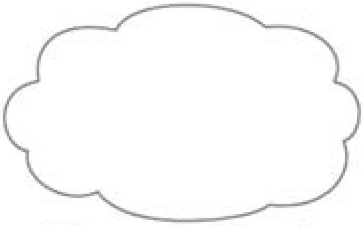
Token Ring



Line: Serial

FDDI

Line: Switched Serial



Network Cloud

Command Syntax Conventions

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

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

Chapter 1. Introduction to Remote Access

This chapter covers the following topics:

- Types of Remote-Access Users
- Remote-Access Technologies

Remote-access networks connect central facilities to remote locations. These can range from remote branch offices connecting to central office sites, to telecommuters connecting back to the office. This chapter introduces the various types of remote-access technologies, networks, and their users.

Types of Remote-Access Users

Remote-access users vary widely in their situations and needs. The type of network they use depends on their specific needs:

- Corporate users in a branch office— These users are connected back to a central office, usually by a Frame Relay or serial link. ISDN links can sometimes be used to back up the primary link. Of late, broadband technologies such as cable are also being used in branch offices.
- Telecommuters working from home— These users use a wide variety of technologies, including digital subscriber line (DSL), cable modems, and dialup links. Virtual private networks (VPNs) can also be employed by these users for added security.
- Traveling users— Also known as road warriors, they use dialup links and VPN technologies to connect to resources. They may also employ wireless technologies such as 802.11x for connectivity.

Remote-Access Technologies

Based on the needs of the users just described, a wide range of technologies can be used to provide remote access. Some of the traditional technologies include Frame Relay, leased lines, ISDN, and dialup links. Newer technologies include DSL, cable modems, and wireless technologies such as 802.11x.

VPNs have also become a significant technology in the past few years. They can provide an alternative to expensive leased lines in a central office/branch office scenario. Also, they can provide security to users who use them over DSL and cable modem networks to connect to the central office. IPSec VPNs are a good example of the latter scenario.

Frame Relay

Over the past few years Frame Relay has been one of the most popular remote-access technologies. It offers a high-speed connection between a central office and a branch office.

One of Frame Relay's benefits is built-in congestion control to combat bursty traffic. As bandwidth needs have increased over the years, this technology has proven very popular. Also, Frame Relay circuits can be ordered from providers in a variety of bandwidths. Starting at 56 kbps, these are usually fractions of a T1. This allows for flexibility when planning.

Some of the services that can be used over Frame Relay networks are data, voice over IP, voice over Frame Relay, and IP Multicast.

Frame Relay operates at Layer 2 by encapsulating Layer 3 traffic such as IP within a Frame Relay frame. To improve performance, Frame Relay relies on higher-layer protocols such as TCP to overcome corrupt or dropped frames that occur during transmission. This is different from protocols such as X.25 that have built-in error checking/correction. Often Frame Relay is described as a successor to X.25.

Frame Relay employs its own addressing scheme at Layer 2 to specify a frame's destination. This feature is called a *Data Link Connection Identifier (DLCI)*. This field in the Frame Relay header tells the Frame Relay switch where to route the frame. The DLCI can be thought of as the Media Access Control (MAC) address in the Frame Relay network.

Another advantage of Frame Relay is its capability to establish one-to-many connections. This is often called *point-to-multipoint*. This capability can potentially allow the redirection of traffic around an outage, provided that a partially-meshed network exists.

One of Frame Relay's drawbacks is the high cost of provisioning links. The high cost can possibly be justified in a branch office scenario, but it might be unsuitable for single remote users.

Serial Links

This type of network has also been historically popular in connecting branch offices. These lines can be ordered from a fractional T1 such as 56 kbps up to DS3s. Possible fractional T1 line speeds include 56 kbps, 128 kbps, 256 kbps, and so on. A full DS3 has a speed of 45 Mbps.

These networks do not provide any of the congestion control and error-detection capabilities that Frame Relay provides. The onus is completely on the higher-layer protocols to provide such services.

As with Frame Relay, cost is also an issue with these links.

ISDN

Integrated Services Digital Network (ISDN) remains one of the most flexible and widely offered services today. Providers all over the world offer ISDN services to users.

ISDN basically comes in two varieties:

- Basic Rate Interface (BRI), which consists of two 64-kbps Bearer (B) channels and one 16-kbps Data (D) channel. This is often represented as 2B+D, for the two B channels and one D channel. This has traditionally been the choice of many remote users who connect to the office from their residences. It remains popular, especially with users who do not have DSL or cable services available at their residences.
- Primary Rate Interface (PRI), which in the U.S. consists of 23 64-kbps B channels and one 64-kbps D channel. This is often represented as 23B+D. In Europe, PRI consists of 30 64-kbps B channels and one 64-kbps D channel. PRI services are often used when greater bandwidth is needed, such as when a connection is needed between a central office and a branch office.

Even though ISDN is offered all around the world, there are differences in the switches that providers use to provide ISDN service. When configuring ISDN, make sure that the design and configuration match the switch type and service being offered.

ISDN makes use of the same wiring used by analog phone lines. However, because ISDN is digital, the signal transmitted across the line is digital instead of analog. This allows for much higher transmission speeds. In addition, call setup for ISDN is very quick compared to that of an analog line. This is because of the use of the separate D channel. The setup is done out-of-band on the D channel, it does not disturb existing user traffic, and it takes a short amount of time. The combination of these factors makes ISDN ubiquitous, fast, and convenient.

ISDN is useful when a variety of applications need to be supported. The higher bandwidth can support applications such as videoconferencing, web browsing, e-mail, and voice services. Also, ISDN can support multiple data sources, as opposed to analog, which typically can support only one data source at a time.

ISDN lends itself to a variety of applications in the remote-access arena. Users who want to connect from home or users in a small office/home office (SOHO) typically can use BRI connections to do so.

However, in a scenario where a branch office needs a connection to a central office, PRI services can be used either as a primary link or as a backup connection that can be activated when the primary line goes down or when additional bandwidth is needed. This is often called a *remote office/branch office (ROBO) scenario*.

ISDN does have some drawbacks. A variety of standards are supported in different parts of the world. This results in a variety of equipment needed to support these standards and interfaces. You have to be careful when ordering, configuring, and maintaining equipment that connects to ISDN providers in different parts of the world.

Another drawback of ISDN is its cost. Because ISDN is charged on a per-usage basis, it can be expensive to operate. This is one of the reasons why ISDN is used in many scenarios as a backup to a serial link that has a flat per-month cost.

Lately ISDN has been replaced in many homes and SOHO environments by technologies such as DSL and cable modems, which offer much higher transfer rates. These services are also cheaper because they offer flat-rate pricing. The combination of these factors has made these technologies more attractive than ISDN.

Analog

Analog dialup service is the most ubiquitous remote access available. All you need is a phone line and a modem. Speeds, which started out around 300 bps, have steadily increased over the years to 56 kbps.

Users using analog dialup usually connect to an access server using a modem. The provider that operates the access server gives the user a phone number. The user connects to the access server using that phone number.

If in the same calling area, the user can connect to the provider using a local phone number. If the user is not in the same calling area, many providers have toll-free numbers. This allows users to connect without incurring long distance charges or using calling cards.

Some providers also offer software that has a list of phone numbers organized by country. Users can use this software to select the appropriate number for their location. They can then connect from all over the world.

Users can also connect to the Internet via dialup and then use VPNs to connect to their corporate networks. Many operating systems now offer native VPN solutions such as Microsoft's Point-to-Point Tunneling Protocol (PPTP).

The most obvious drawback of dialup services is the speed—or lack thereof. With applications becoming more and more bandwidth-intensive and other broadband options becoming more cost-effective, users are turning away from dialup.

DSL

In the past few years, DSL has emerged as one of the technologies that can provide broadband services to homes. This technology can support both high-speed data and voice at the same time. It also can support data transfer rates of up to several megabits. Certain flavors of DSL can deliver speeds of up to 52 Mbps.

These transfer rates are made possible by using unused frequencies on copper telephone lines. The available bandwidth is divided into frequency ranges. One frequency range is used for voice, another is used for upstream data transmission, and another is used for downstream data transmission. For example, voice uses the frequency range of 0 to 3.4 kHz, and Asymmetric DSL uses the frequency ranges of 25 to 138 kHz in the upstream direction and 170 to 1104 kHz in the downstream direction. Splitters are sometimes used to separate these frequencies.

Another feature of DSL is that it is "always on." Unlike ISDN and analog, no dial-in is required. This is an attractive feature, especially for users who are accustomed to the cumbersome call setups and busy signals associated with analog dialup services.

DSL offerings can be broadly divided into two categories:

- Asymmetric DSL
- Symmetric DSL

Asymmetric DSL

In this category, the upload and download speeds differ. Here are some of the different Asymmetric DSL technologies:

- Asymmetric DSL (ADSL)— As noted in the name, this technology offers differing upload and download speeds. This is the most common technology for residential and commercial use. It can be configured to reach rates of 6 Mbps.
- Rate-Adaptive DSL (RADSL)— This technology uses ADSL modems that can adjust to differing line lengths and line qualities. The speed varies in this technology, depending on conditions, up to 7 Mbps.
- Very High Bit Rate DSL (VDSL)— The fastest DSL technology, it has a maximum range of 4500 feet and can deliver rates of up to 52 Mbps.
- Consumer DSL (CDSL)— This technology does not need a splitter like ADSL and RADSL. In those technologies, splitters are used to split the frequency ranges and protect the different ranges from interference. CDSL is slower than ADSL and offers downstream speeds of around 1 Mbps.

Symmetric DSL

In this category, the upload and download speeds are the same. Here are some of the different Symmetric DSL technologies:

- Symmetric DSL (SDSL)— This technology is suited to environments that need higher upload speeds than those offered by ADSL. It is provided over a single telephone line and typically offers rates of around 768 kbps.
- Integrated Services Digital Network DSL (IDSL)— As the name implies, this technology is similar to ISDN in that it can use the same terminal adapter. However, it is different in that it is always on. Also, IDSL is not metered like ISDN. It is a symmetric service offering rates of around 144 kbps.
- High Bit Rate DSL (HDSL)— This technology delivers symmetric data rates of around 1.5 Mbps in both directions. It runs over two-wire pairs.

DSL has a wide range of offerings that users can choose from. Also, DSL's always-on characteristics and its support of a wide range of applications make it an attractive technology for many remote users.

DSL does have its drawbacks. Its distance limitation is a significant issue. DSL services cannot be offered beyond certain distances from the central office. Also, DSL is not as ubiquitous as other services, like dialup and ISDN.

Cable Modem Services

The demand for high-speed Internet access in the past few years has seen the rise of cable modem services as a broadband alternative. The technology takes advantage of the wide reach of cable infrastructure used to deliver television service.

Data is transmitted over the network as radio frequency (RF) signals. The cable modem converts these into digital signals. In addition to television and data signals, analog voice signals can be transmitted over the network. These systems can also perform full-duplex communications. The fiber coming from the homes of subscribers is usually aggregated in remote units, and fiber is used to connect these units to headend routers. This kind of hybrid network is also called a *Hybrid Fiber-Coaxial (HFC) network*.

Different frequency ranges are used to transmit in upstream and downstream directions. The cable modem uses channels in the 5-to-42 MHz range to transmit data in the upstream direction. Similarly, a TV channel in the 50-to-750 MHz range is used for downstream traffic.

Cable can support a significant amount of bandwidth—enough bandwidth to allow subscribers to watch television and be on the Internet at the same time. The cable modem uses 10/100 Ethernet or USB to connect to the user's PC.

In addition to bandwidth, cable is also attractive because of the wide range of applications it can support. Data, voice, and video can all be supported by this medium.

Conversely, because cable is a shared medium, performance can be degraded if too many users are on the same segment. This is the most significant drawback of cable. The shared nature of the medium also raises security concerns, because traffic can potentially be captured using a packet sniffer.

Also, the frequency range used for upstream communications is vulnerable to interference caused by household appliances.

Summary

This chapter provided a brief introduction to the various kinds of remote-access technologies. Some of them, such as leased lines and analog, have been in existence for quite a while. On the other hand, technologies such as DSL and cable modems are more recent offerings and provide bandwidth not usually associated with remote-access technologies.

Each technology has advantages and drawbacks. Different technologies can be used for different needs based on their strengths and weaknesses.

Review Questions

1: What are the main kinds of remote-access users?

2: At what OSI layer does Frame Relay operate?

3: What addressing feature of Frame Relay allows for frame routing?

4: What are some advantages of Frame Relay?

5: What are the two main varieties of ISDN?

6: What are two advantages of ISDN?

7: What are the two main varieties of DSL?

8: What are two advantages of DSL?

9: What are some drawbacks of DSL?

Chapter 2. Building a CCNP Remote-Access Lab

This chapter covers the following topics:

- Creating LANs
- Creating WANs by Using a Cisco Router as a Frame Relay Switch
- Creating Asynchronous, ISDN, PPP, DDR, Dial Backup, AAA, and Security Labs

It is essential to have hands-on experience, because the new exam format requires you to understand how to configure Cisco devices to be able to pass the exam. In the new format, you are given interactive access to routers and are asked to configure the routers. If you have taken the new CCNA or CCNP Routing and Switching exam, you should be familiar with the new exam format.

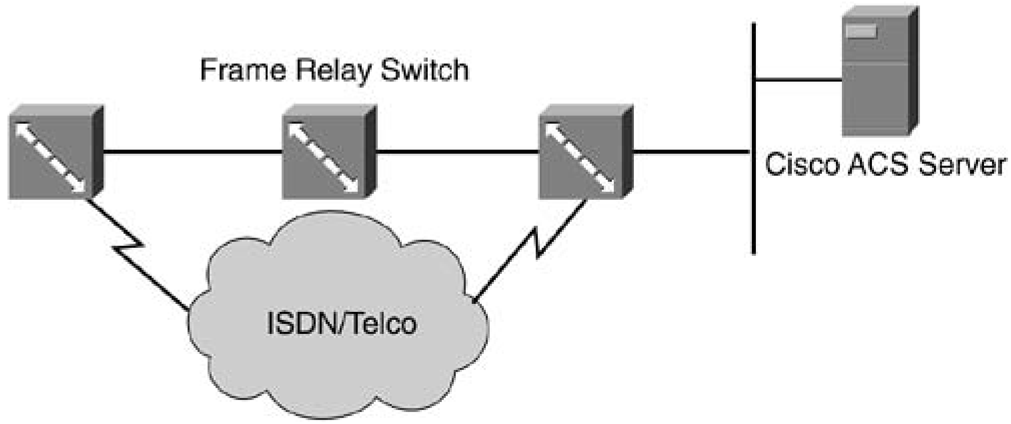
This chapter provides some suggestions on what devices you should acquire to build a lab and which technologies you can practice by using this lab.

Before you begin, review some of the areas in which you can possibly build a home lab to study for the Remote Access exam:

- Asynchronous
- PPP
- ISDN BRI
- Dial-on-demand routing (DDR)
- Frame Relay
- Dial backup
- Queuing and compression
- Network Address Translation (NAT)
- Authentication, authorization, and accounting (AAA)
- Security

Figure 2-1 illustrates the lab topology you can use to study most of the areas mentioned in the preceding list.

Figure 2-1. CCNP Home Lab Topology



Several key components are required to model the remote-access lab. The following list should be viewed more as a list of roles than a list of devices:

- LANs: Switches/hubs and cables
- WANs: Routers and cables
- Routers
- Test hosts and applications

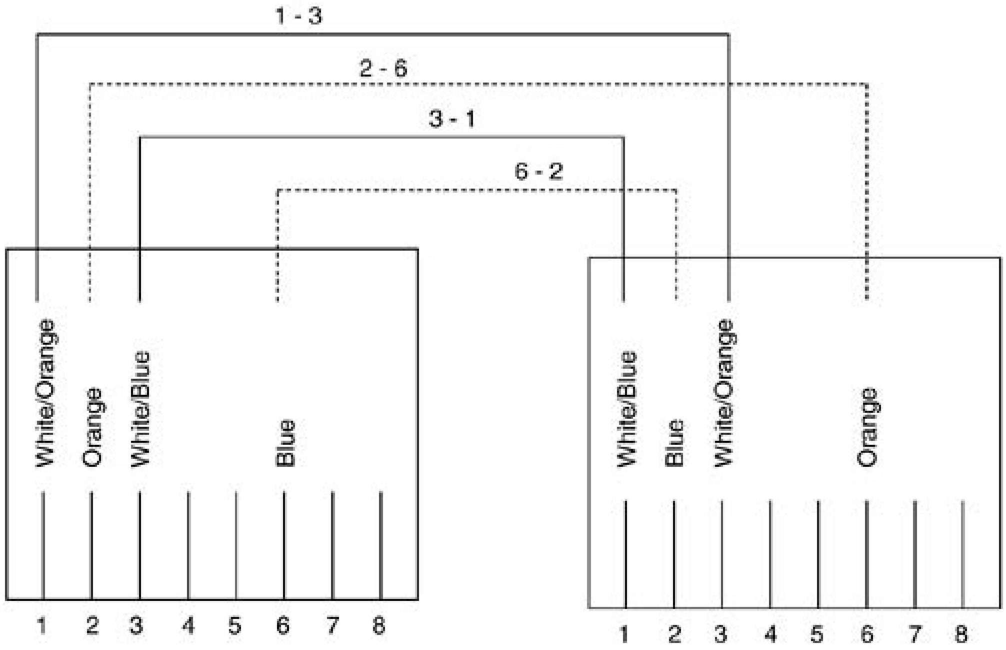
Creating LANs

Some labs require host connections. For example, in Chapter 13, "Using AAA to Scale Access Control in an Expanding Network," you need to test the configuration between the routers and the AAA server. You can use several different methods to model LANs:

- Using switches
- Using hubs
- Using an Ethernet crossover cable

If you will not connect more than two devices, a common method is to use an Ethernet crossover cable. In the lab environment, you can use this cable to connect two routers or to connect a router to one host. An Ethernet crossover cable is just an RJ-45-to-RJ-45 patch cable, pinned out in a crossover pattern. Figure 2-2 illustrates the pinouts (pins 1, 2, 3, and 6 are used) for an Ethernet crossover cable.

Figure 2-2. Pinouts for an Ethernet Crossover Cable



Creating WANs by Using a Cisco Router as a Frame Relay Switch

You can configure any Cisco router with Cisco IOS Release 11.0 or later and at least two serial interfaces as a Frame Relay switch. Two interfaces are needed because the switch is primarily a data communications equipment (DCE) device and requires two routers to serve as the data terminal equipment (DTE) devices. Because the Frame Relay switch is a DCE-only device, it requires DCE serial cables as well.

The most common way to provide Layer 1 WAN connectivity between routers is to connect a female V.35 DCE cable to a male V.35 DTE cable. In any back-to-back configuration, you need to ensure that one side (DCE) of the link sets clocking. To configure an interface's clock rate, use theclock rate [*value*] command. Example 2-1 shows how to set the clocking on a serial interface.

Example 2-1. Configuring the Clock Rate on a DCE Interface

```
fr_switch(config)#interace serial 1
fr_switch(config-if)#clock rate ?
       Speed (bits per second)
  1200
  2400
  4800
  9600
  19200
  38400
  56000
  64000
  72000
  125000
  148000
  250000
  500000
  800000

  1000000

  1300000

  2000000

  4000000


  <300-4000000>    Choose clockrate from list above
```

These cables can be ordered from Cisco Systems—part number CAB-V35MT for the V.35 male DTE cable and part number CAB-V35FC for the female DCE cable. When the cables are connected in a back-to-back mode, sometimes it can be difficult to tell which one is the DCE cable. The show controller command specifies the cable type and whether the cable is DCE or DTE.Example 2-2 shows the output of show controller, where you can tell what the interface type is. As you can see from the example, the interface serial 0 is a V.35 DTE cable. Use V.35 cables whenever possible because of their flexibility in a lab environment.

Example 2-2. show controller Command

```
Router#show controller serial 0

HD unit 0, idb = 0xCED94, driver structure at 0xD3B18

buffer size 1524  HD unit 0, V.35 DTE cable

cpb = 0xE2, eda = 0x4140, cda = 0x4000

RX ring with 16 entries at 0xE24000

00 bd_ptr=0x4000 pak=0x0D66F0 ds=0xE2DDB0 status=80 pak_size=0
```

Configuring a Cisco Router as a Frame Relay Switch

To configure Frame Relay switching, you must perform the following tasks:

Step 1. Enable Frame Relay switching.

You do this with the global configuration command frame-relay switching.

Step 2. Configure the interface LMI and the Frame Relay interface type.

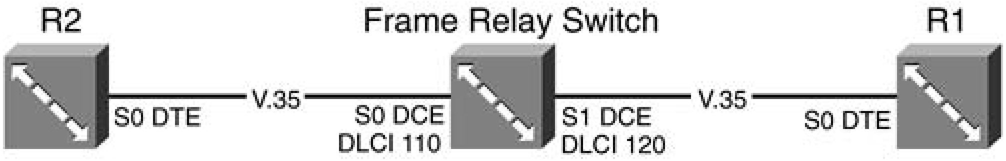
You need to set the encapsulation to Frame Relay with the encapsulation frame-relay command, and you must set the LMI type with the frame-relay lmi-type [ansi | cisco | q993a] command from the interface prompt. To continue configuring the Frame Relay interface, add the frame-relay intf-type dce command. Because the interface is DCE, you also need to use the clock rate*bps* command. The *bps* values range from 1200 to 8000000.

Step 3. Configure PVCs with the frame-relay route command.

You do this with the interface command frame-relay route [*16-1007*]*inbound_DLCI* interface*outbound_serial_interface* [*16-1007*]*outbound_DLCI*. This command creates a PVC on the interface and maps it to another interface.

Figure 2-3 shows the diagram used in this example. It highlights the network from a hardware and service provider perspective. The Frame Relay switch has two V.35 DCE cables to two routers, R1 and R2. These two routers have V.35 DTE male cables connected to their Serial 0 ports. You configure a PVC with DLCI 110 on Serial 0 mapping to DLCI 120 on Serial 1. Other types of cables, such as X.21 or RS232, can be used as well. Cisco also makes back-to-back cable, which can save you a lot of space when you build a lab at home.

Figure 2-3. Basic Frame Relay Configuration



Example 2-3 demonstrates the use of these commands and the basic configuration of a Frame Relay switch.

Example 2-3. Configuring a Basic Frame Relay Switch

```
fr_switch#configuration terminal

Enter configuration commands, one per line.  End with CNTL/Z.

fr_switch(config)#frame-relay switching

fr_switch(config)#interface serial 0

fr_switch(config-if)#encapsulation frame-relay

fr_switch(config-if)#frame-relay intf-type dce

fr_switch(config-if)#frame-relay lmi-type ansi

fr_switch(config-if)#clock rate 128000
fr_switch(config-if)#frame-relay route 110 interface s1 120
fr_switch(config-if)#exit
fr_switch(config)#
fr_switch(config)#interface serial 1
fr_switch(config-if)#encapsulation frame-relay
fr_switch(config-if)#frame-relay intf-type dce
fr_switch(config-if)#frame-relay lmi-type ansi
fr_switch(config-if)#clock rate 128000
fr_switch(config-if)#frame-relay route 120 interface s0 110
fr_switch(config-if)#exit
```

Example 2-4 shows the router's configuration in its entirety.

Example 2-4. Entire Frame Relay Configuration

```
fr_switch#show running-config

hostname fr_switch
!
frame-relay switching
!
interface Serial0
 no ip address
 encapsulation frame-relay
 clockrate 128000
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 110 interface Serial1 120
```

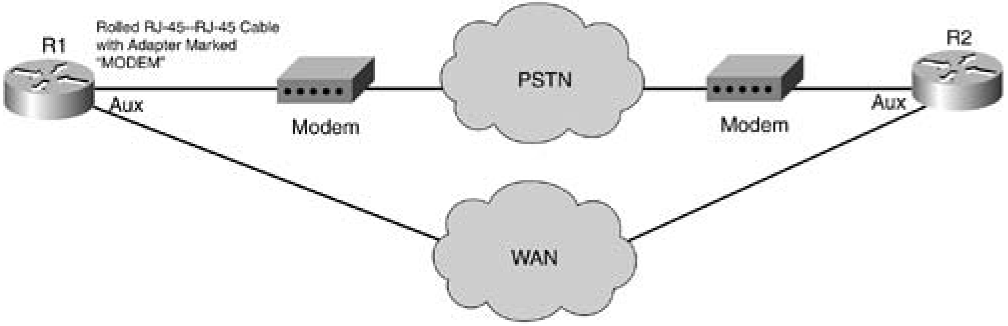
```
!
!
interface Serial1
 no ip address
 encapsulation frame-relay
 clockrate 128000
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 120 interface Serial0 110
!
!
no ip classless
!
line con 0
line aux 0
line vty 0 4
 login
!
end
```

Creating Asynchronous, ISDN, PPP, DDR, Dial Backup, AAA, and Security Labs

Figure 2-4 illustrates the topology you can use to practice asynchronous communication, PPP, DDR, and dial backup.

Figure 2-4. Asynchronous, PPP, DDR, and Dial Backup Lab Topology

[View full size image]



You can use any Cisco router with an auxiliary port, a rolled RJ-45 cable, an adapter marked "MODEM" (Cisco part number CAB-25AS-MMOD), and any modem that is V.34-capable or better to build this lab. If you have one of the following routers, you can also use a SCSI-II 68-pin async port, an eight-to-one octopus cable, and a 25-pin adapter to build this lab:

- Cisco 2509/2510
- Cisco 2511/2512
- Cisco 2600/3600 with SCSI-II 68-pin 16/32-port async port

The part number for the octopus cable is CAB-OCTAL-KIT. It also includes modem head-shells for any asynchronous devices, such as modems.

It might not be feasible to order ISDN lines from your service provider. Getting two physical lines can prove costly, because there is an installation charge for the ISDN circuits, as well as the ongoing call charges as you use and test ISDN within your lab. You still might want to do some research and find out if ISDN cost for your location is reasonable enough for short-term testing. ISDN simulators are also expensive, around $800. You might be able to pick up a secondhand ISDN simulator or even rent one for a couple of months. Investing in an ISDN simulator is definitely worthwhile if you are considering pursuing CCIE certifications such as Routing/Switching and Security in the future.

Cisco Secure Access Control Server software can be downloaded for evaluation from the Cisco website. It offers centralized control from a web-based graphical interface to manage AAA functionality.

Routers are the basic requirement for the CCNP remote-access lab. Three routers should be enough for you to practice most of the areas covered in the CCNP Building Cisco Remote Access

Networks (BCRAN) exam.

Ideally, you should look for 2600XM routers. They are Cisco's current product line. They support all the new technologies, such as VoIP and VPN acceleration through hardware, and they also have software support to allow current Cisco IOS software images to be used. These are modular routers that allow a number of different modules to be included.

Older routers such as the 2500 and the 4000/4500/4700 might also be an option. These routers are less expensive than the 2600XM models and offer a range of interfaces. Memory restrictions on these models might hinder future proofing when newer processor- and memory-intensive Cisco IOS software releases are introduced.

All routers in the lab should have enough DRAM and Flash memory to load and use at least the IP PLUS IPSEC 56 Cisco IOS software feature set. This feature set has all the software functions required for the CCNP remote-access lab, including IPSec.

Summary

This chapter presented a lab topology that should allow you to practice most of the technologies discussed in this book:

- Asynchronous
- PPP
- ISDN BRI
- DDR
- Frame Relay
- Dial backup
- Queuing and compression
- NAT
- AAA
- Security

It can be quite expensive to build a home lab. However, if you intend to pursue other Cisco certifications such as CCIE in the future, building a home lab can turn out to be very cost-effective. Many companies offer online rack-time rental. This is another option for you to consider.

Chapter 3. Modem Connections and Operation Overview

This chapter covers the following topics:

Wide-area communication takes advantage of the existing PSTN for data transfer by converting digital signals into analog signals and vice versa for transmission over the PSTN. The device used to accomplish such conversion is called a *modem* (short for modulator/demodulator).

This chapter concentrates on the following modem-related topics:

- A typical modem connection
- DTE-to-DTE wiring
- Data compression and error control
- Configuring the modem (DCE)

A Typical Modem Connection

The devices involved in a modem connection belong to one of two groups: data terminal equipment (DTE) or data communications equipment (DCE).

NOTE

Interestingly, the Electronic Industries Association (EIA) defines DCE as data communications equipment. However, the International Telecommunication Union-Telecommunications Standards Sector (ITU-TSS, or ITU-T) defines DCE as data circuit-terminating equipment.

Examples of the DTE devices are

- PCs
- Routers
- Mainframe computers

DCE devices include

- Modems
- Channel service units/data service units (CSUs/DSUs)

Communication Between DTE Devices

Communication between DTE devices is accomplished through communication between DCE devices.

In other words, DTE-to-DTE communication involves three stages:

- DTE(1)-to-DCE(1)
- DCE(1)-to-DCE(2)
- DCE(2)-to-DTE(2)

Each of the three stages requires different cabling and configuration. The next section describes how the DTE-to-DCE interface defined by the EIA/TIA-232 standard works. (TIA stands for Telecommunications Industries Association.)

DTE-to-DCE Communication

Out of the 25 pins available in a DB-25 connector, only eight are actually used for signaling to connect a DTE to a DCE. The remaining 17 signals are disregarded. In turn, the eight utilized signals can be divided into three categories. These categories and their corresponding signals are described in Table 3-1.

Table 3-1. DTE-to-DCE Signals

Category	Signal	Function
Data transfer	Transmit Data (TxD)	The DTE transmits data to the DCE.
	Receive Data (RxD)	The DTE receives data from the DCE.
	Ground (GRD) or pin 7	Provides the ground reference for voltage measurements.
Hardware flow control	Request to Send (RTS)	Indicates that the DTE has buffers available to receive from the DCE.
	Clear to Send (CTS)	Indicates that the DCE has buffers available to take data from the DTE.
Modem control	Data Terminal Ready (DTR)	The DTE tells the DCE that it can accept an incoming call.
	Carrier Detect (CD)	The DCE has established a carrier signal with the remote DCE.
	Data Set Ready (DSR) or pin 6	The DCE is ready for use (a pin is not used on modem connections).

Modem Control Functions

The modem control category signals are sent between the DTE and the DCE to open or close the connection. They also check the connection status. An existing connection termination can be initiated by a DTE or a DCE.

When a termination is prompted by a DTE device, the access server drops the DTR signal. The modem must understand that the connection needs to end when a DTR signal is no longer present. In a DTE-initiated termination with an improperly configured modem control, the DTR signal might not be dropped or recognized, and the modem might not hang up as a result.

When a termination is initiated by a DCE device, the modem must correctly reflect the state of the carrier with the CD signal. The access server recognizes that the CD signal is low and therefore drops the connection. During a DCE-initiated termination with an improperly configured modem control, the CD signal might not be dropped or recognized, and you might get into someone else's modem session by mistake.

DCE-to-DCE Communication

When a modem has data to send, the following sequence of events takes place:

1. DTE data enters the sending modem via the TxD pin. When DTE sends data to a DCE and the sending modem's buffer is nearly full, a DCE can control flow (via hardware) by lowering the CTS signal. This way, the DTE knows not to use TxD.
2. Data is compressed. At the data compression stage, the sending and receiving modems agree on the compression algorithm. A standard MNP 5 or V.42bis algorithm is used.
3. Data is packetized. The following tasks are performed:

   a. Windowing

   b. Checksum

   c. Error control

   d. Retransmission
4. Data is modulated from digital into analog signals.
5. Data is sent over the telephone network.

When the receiving modem gets the data, it performs the same steps as just listed. Only this time, the order is reversed and is as follows:

1. The signal is demodulated.
2. The data is depacketized.
3. The data is decompressed.
4. The data is delivered to the destination DTE.

If the receiving DTE is unable to receive data on RxD, it can send an RTS signal.

DTE-to-DTE Wiring

This section examines the DTE-to-DTE wiring functions.

When the DTE-to-DTE Devices Are in the Same Vicinity

If two DTE devices such as a terminal and an access server are located close to one another, it makes more sense to link them back to back instead of using a telephone network and two DCEs. A regular EIA/TIA-232 cable cannot be used for such DTE-to-DTE links, because both DTE devices send on TxD pin 2 and receive on RxD pin 3. In such instances, a null modem can accomplish direct DTE-to-DTE connections. With null modems, pins 2 and 3 are crisscrossed, as well as other corresponding pins of the DB-25 connector, and thus allow the DTEs to communicate with one another.

Alternatively, you can configure some devices, such as serial printers, to act as either a DTE or a DCE. If a device is configured as a DCE, it transmits data on pin 3 and receives data on pin 2. Such configuration forgoes a null modem connection and allows a DTE (such as a PC or server) to be directly connected to a printer with a regular EIA/TIA-232 cable.

RJ-45 Wiring and Cables

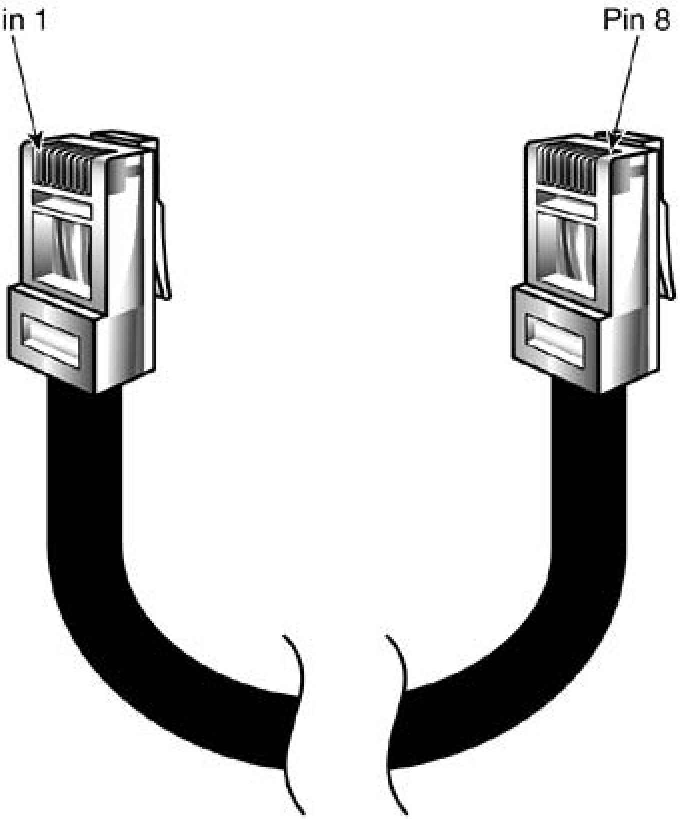
RJ-45 connectors are used for the following ports:

- Console
- Asynchronous
- Auxiliary

No standards define RJ-45 interface pinouts, but Cisco defines them as DTE. If you were to cable the access server port (RJ-45) to an external device (modem or terminal), you would need RJ-45-to-RJ-45 cable and an RJ-45-to-DB-25 adapter. An RJ-45-to-RJ-45 cable can be rollover or straight-through. A rollover cable has its pins reversed, as in 1 to 8, 2 to 7, and so on. A straight-through cable, on the other hand, has the pins going straight in a 1 to 1, 2 to 2 fashion.

To find out which of the two types of cable you have, hold the two connector ends of the same cable side by side. Check the color-coded wires inside the connector. Straight-through cable wires are the same color for the same pins on both connectors. A rolled cable has the wire colors on the two connectors flipped, as shown in Figure 3-1.

Figure 3-1. Identifying Rollover Cable



Pin 1 on one connector and pin 8 on the other connector should be the same color

The octal cable used for the asynchronous port connections functions as a rolled cable.

An RJ-45-to-DB-25 adapter can be either rollover or straight-through. For instance, a male or female DTE adapter (MDTE or FDTE) is straight-through. A male or female DCE adapter (MDCE or FDCE) is rolled. A male modem (MMOD) adapter is rolled and is the only one that supports modems. In it, the MDCE connectors are changed so that DB-25 pin 8 instead of pin 6 is wired to DSR.

The auxiliary and console ports are configured as DTE devices on Cisco access servers. Terminals (such as PCs) are also DTE devices. Two DTE devices cannot be directly connected unless the signals are rolled exactly once. So you must either roll the pins in the cable or in the DB-25 adapter, but not both. To directly connect two DTE devices, you can use either of these formulas:

- DTE + rolled RJ-45 cable + straight DB-25 adapter + DTE
- DTE + straight RJ-45 cable + rolled DB-25 adapter + DTE

DTE-to-DCE Wiring

A DTE-to-DCE connection should not have rolls. The same effect can be achieved with having two rolls and the connector. Cisco routers come with a kit for console and auxiliary port cabling.

The kit includes the following:

- RJ-45-to-RJ-45 rollover cable
- RJ-45-to-DB-9 female DTE adapter (labeled "TERMINAL")
- RJ-45-to-DB-25 female DTE adapter (labeled "TERMINAL")
- RJ-45-to-DB-25 male DCE adapter (labeled "MODEM")

The RJ-45-to-DB-9 female DTE adapter is typically used to connect a PC being used as a console terminal. The RJ-45-to-DB-25 female DTE adapter is used to connect a terminal to the console or auxiliary port. The RJ-45-to-DB-25 male DCE adapter is used to connect the auxiliary port to a modem.Table 3-2 describes the port types for console and auxiliary ports on Cisco routers.

Table 3-2. Port Types for Console and Auxiliary Ports on Cisco Routers

	Routers	
	DB-25	RJ-45
Console port	DCE	DTE
Auxiliary port	DTE	DTE

Data Compression and Error Control

Data compression results depend on the type of data being compressed. Some types, such as ASCII files, can be compressed quite a bit. Other types of data can compress only a little. Even though certain software applications can be used to achieve data compression, normally it's better to leave this operation up to the modem. This is because modem hardware compression algorithms are faster than the ones used by host software.

Compression normally works with error-correction algorithms. Error detection and correction techniques can be used to guarantee data integrity at any transmission speed. Two examples of such techniques are

- Microcom Networking Protocol (MNP)
- Link Access Procedure for Modems (LAP-M)

V.42bis and MNP5 are the compression algorithms that commonly operate over LAP-M or MNP4 correction. The V.42 and V.42bis compression algorithms can be implemented in V.32 and V.34 modems as well as in other equipment with lower speed capability. In theory, V.42bis can provide the 4:1 compression ration. However, in practice, this is rarely accomplished.

V.42bis compression is achieved when both communicating modems agree to use it. In such instances, the software compression option should be turned off. If hardware compression is used, the data transfer between the DTE and DCE can occur at a higher speed.

Modem Modulation Standards

ITU-T defines a number of modem modulation standards, as shown in Table 3-3.

On top of various ITU standards, manufacturers have devised their own versions of modems. This causes some interoperability issues among different kinds of modems, even when the modems come from the same vendor.

Table 3-3. Modulation Standards

Standard	Description
V.32bis	Finalized in July 1991.
V.34	Finished in June 1994.
V.34 annex 12	Supports 33.6 Kbps transmit and receive operation. If compression is used, a transmission rate of up to 133.8 Kbps is possible if the PC can deal with this speed.
V.90	The 56 Kbps standard is the most recent one. Most modem manufacturers now have products that meet this standard. This is despite the fact that a data rate of 53 Kbps is the maximum permitted within the U.S.

When V.34 modems are properly configured, they can adapt to line conditions. Initially, two modems attempt to establish a call at 28.8 Kbps. If this transmission speed isn't possible because of line conditions, the modems can continue to reduce the speed in 2.4 Kbps increments all the way down to a minimum speed of 2.4 Kbps. By the same token, the modems try to increase the speed when line conditions improve.

In contrast, older modems can negotiate a fixed transmission rate only during handshaking, thus continuing transmission at the speed agreed to at the outset by the two modems. This situation might result in a connection failure if an older modem's line becomes particularly bad. If the line quality improves down the road, older modems still can't take advantage of greater bandwidth.

The access server is unaware of modulations, because it is only directly involved with DTE-to-DCE communication. However, the access server-to-modem speed must consider the modulation speed and compression ratio to achieve the best end-to-end performance.

The Relationship Between Modem Speeds and Compression Ratios

DCE-to-DCE speed is modem-to-modem communication speed across the telephone network. DTE-to-DCE speed is the communication speed between the computer and the modem attached to it. If you want to gain maximum benefits from compression, the PC should clock the modem at its speeds equal to the potential compression ratio. In a PC, the DTE should set the modem at its fastest rate to take advantage of compression.

The EIA/TIA-232 serial interface (COM port), found on PCs and some Macs, is sometimes used with Universal Asynchronous Receiver Transmitters (UARTs) and character-oriented communication packages. However, these features are unreliable at higher data rates, and the speed of the interface might fall a good deal short of the full potential of V.34.

If a modem isn't configured properly, it might automatically alter DTE-DCE speeds so that they match DCE-DCE speeds. This is often called *speed mismatch*. You can prevent speed mismatch by locking the DTE-DCE speed so that it remains the same as originally configured. This speed-locking procedure is called *speed conversion*. It is also known as *port-rate adjustment* or *buffered mode*.

Table 3-4 lists the maximum theoretical speeds possible for selected modem modulation standards. You can also see the possible speeds where V.42bis compression is used with the same standards.

Table 3-4. Maximum Theoretical Speeds for Modulation Standards

Standard	Speed	Maximum Speed with 4:1 V.42bis Compression
V.90	56000	224000
V.34	28800	115200
V.32 turbo	19200	76800
V.32bis	14400	57600
V.32	9600	38400

Configuring the Modem (DCE)

In this portion of the chapter, you will learn about the tasks involved in configuring the modem:

- Connecting to the modem (DCE)
- Basic modem configuration
- Modem autoconfiguration

Connecting to the Modem (DCE)

Asynchronous dial-up involves the use of analog modems to convert data into streams of information that can be carried over phone lines. These modems can be attached externally, as with the Cisco 2511 access server, or they can be integrated into the product, as with Cisco AS5200 series access servers. The line that connects the modem can be a physical asynchronous line (external modem) or a virtual line inside an integrated modem module (integrated modem).

In the following sections, you will learn to

- Differentiate between a forward and reverse connection to a modem
- Configure a reverse-Telnet session
- Configure line types

Differentiating Between a Forward and Reverse Connection to a Modem

Cisco access servers support two types of connections to a modem: incoming asynchronous line (forward) and outgoing asynchronous line (reverse). A user who dials into an access server from a remote terminal through an asynchronous line makes a *forward connection*, and a user who connects through an access server to an attached modem to configure that modem makes a *reverse connection*, known as *reverse Telnet*.

A host can make reverse-Telnet protocol connections to devices attached to a Cisco access server. Different port numbers (20*xx*, 40*xx*, and 60*xx*) are used for different device types. This is because each type has its own unique data type and protocol negotiations. The remote host must specify a particular TCP port on the router to connect with individual lines or a rotary group. For example, the remote host might make a reverse-Telnet connection to the modem using port 2097. The TCP port number 2097 specifies a Telnet connection (TCP port 2000) to line 97.

For the Telnet protocol, the base TCP port for individual lines is 2000, and the base TCP port for rotary groups is 3000. If the service provided is the raw TCP protocol (no Telnet), the base TCP port for individual lines is 4000, and the base TCP port for rotary groups is 5000. Telnet protocol (binary mode) uses 6000 as the base TCP port for individual lines and 7000 as the base TCP port for rotary groups. The Xremote protocol uses 9000 as the base TCP port for individual lines and 10000 as the base TCP port for rotary groups.

You need to use the transport input command to specify which protocol to use when connecting to a line using reverse Telnet:

```
Router(config-line)#transport input {all | lat | mop | nasi | none | pad |
  rlogin | telnet | v120}
```

For example, if you enter the command transport input all, all possible command option protocols can be used for the connection. The command options are lat | mop | nasi | none | pad | rlogin | telnet | v120. Each command option protocol can also be specified individually.

Configuring a Reverse-Telnet Session

The EXEC commands described in this section allow you to initiate and control a reverse-Telnet session. You use the telnet command to make a Telnet connection to a host or to a particular port on a host:

```
Router#telnet [host] [port] [/debug]
```

You can specify the target *host* either by host name or by IP address. You can use the optional debug switch to obtain more-detailed information about the connection. If you simply enter the name of the host to which you want to make a connection, the system tries to establish a Telnet session with that host by default. The interface through which the connection is made provides the source IP address for the connection.

You use the disconnect command to cut off a particular session or all sessions:

```
Router#disconnect [session-number]
```

Also, you can put the current session on hold by pressing Ctrl-Shift-6 followed by x.

Configuring Line Types

Access servers use four different line types:

- CON (console port)— All Cisco routers have a console port. This port corresponds to line 0 on all routers.
- AUX (auxiliary port)— Most Cisco routers have an auxiliary port. Its number matches up to the line right after the last TTY line on the router.
- TTY (asynchronous port)— TTY lines and asynchronous interfaces correspond on a one-to-one basis. In other words, a TTY line of TTY*n* corresponds to line number *n*. Only access servers have TTY lines.
- vty (virtual terminal)— vty lines are virtual lines normally associated with incoming Telnet sessions. They are dynamically assigned to the synchronous interfaces. The actual line the vty corresponds to is given by the expression line = last_tty_line + 2 + *m*, where *m* equals the number of the vty line. For instance, on a router with 16 TTY ports, the vty 4 line corresponds to line 22.

A connection to a specific access server line is valuable when that line has a dial-out modem, parallel printer, or serial printer attached to it. To establish a connection to such a line, the remote host or terminal should specify a particular TCP port on the access server. For example, if you were to make a Telnet connection to line 97 (2000 + 97), you would need to enter telnet *ip-address*2097.

Theshow line command displays status information on all line types:

```
Router#show line [line-number]
```

If you want more-detailed information on a particular line (such as baud rate, modem state, and modem hardware state), you need to specify the *line-number* when issuing this command.

Example 3-1 shows the output from the show line command. The absolute line numbers are displayed in the Tty column. The next column (Typ) shows the type of line assigned to each line number—CTY, TTY, AUX, vty, or LPT. The line speed associated with each line is shown in the

Tx/Rx column. For example, the AUX line can transmit and receive at 9600 bps. The line's autoselect state is shown in the column labeled A. A value of F indicates that autobaud has been configured for the line, and a hyphen indicates that it has not.

Example 3-1. show line Command Output

```
Router#show line
Tty Typ      Tx/Rx      A Modem  Roty AccO AccI   Uses   Noise  Overruns   Int
*    0 CTY              -    -      -    -    -      0       0     0/0       -
    65 AUX   9600/9600  -    -      -    -    -      0       1     0/0       -
    66 vty              -    -      -    -    -      0       0     0/0       -
    67 vty              -    -      -    -    -      0       0     0/0       -
    68 vty              -    -      -    -    -      0       0     0/0       -
    69 vty              -    -      -    -    -      0       0     0/0       -
    70 vty              -    -      -    -    -      0       0     0/0       -
```

The type of modem signal configured for the line can be callin, callout, cts-req, DTR-Act, inout, or RIisCD. The Roty column indicates the rotary group configured for the line. If an output or input access list is configured for the line, it is shown in the AccO and AccI columns, respectively. The Uses column shows the number of TCP connections established to or from a line since the system was restarted. The system also reports on the number of times noise has been detected on each line since the last restart. The Overruns column indicates the number of hardware (UART) overruns and software overflows that have occurred on each line since the last system restart. Hardware overruns are buffer overruns—they occur when the UART chip receives bits from the software faster than it can process them. Conversely, software overflows occur when the software receives bits from the hardware faster than it can process them.

Basic Modem Configuration

This portion of the configuration section introduces you to some of the beginning stages of modem configuration:

- Interface configuration
- Modem configuration using standard AT commands
- Nonstandard modem commands
- Chat-script configuration

Configuring an Interface

Theinterface async command and the line command are used to configure an asynchronous port. The interface async command lets you configure the protocol or logical aspects of the asynchronous port. The line command lets you configure the physical aspects of the same port. You use the interface async command to configure internal characteristics, such as protocol encapsulation and authentication schemes. But you use the line command to configure external characteristics such as the basic modem-related parameters on an access server.

To make a successful asynchronous connection, you need to configure both the modem and the access server. You need to have the modem

- Perform hardware flow control
- Hang up when you quit a session
- Lock DTE speed

You should also have the Carrier Detect (CD) signal accurately reflect the carrier state.

On the access server, you need to configure the line to which the modem is attached. You begin by using the line command to specify the particular line being configured:

```
Router(config)#line [aux | console | tty | vty]line-number [ending-line-number]
```

You use the login command to set a login password on the line. This prevents unauthorized connection on the line:

```
Router(config-line)#login
```

You specify the password using the password command:

```
Router(config-line)#passwordstring
```

You configure flow control on the line using the flowcontrol command:

```
Router(config-line)#flowcontrol {none | software [lock] [in | out] | hardware
  [in | out]}
```

Because software flow control (xon and xoff characters) is not recommended for modems used with Cisco routers, use this command to specify that Request to send (RTS) and Clear to send (CTS) signals will be used to control the flow of data on the line by setting flowcontrol hardware.

Getting the modem to lock DTE speed ensures that the modem will always communicate with the access server at the specified speed. You use the speed command to set both the transmit and receive speed:

```
Router(config-line)#speedbps
```

Thespeed command lets you set the maximum transmit and receive speed between the modem and the access server. The chosen speed value needs to be expressed in bps.

You use the transport input all command if you want every protocol to be passed to the access server through the line.

The stopbits command allows you to set the number of stop bits transmitted per byte:

```
Router(config-line)#stopbits {1 | 1.5 | 2}
```

You use the modem command to configure the type of modem signal for the line:

```
Router(config-line)#modem inout
```

If you specify a value of inout, the line uses the modem for both incoming and outgoing calls.

Standard Modem AT Commands

Modem vendors all have their own unique set of modem commands. However, several modem attention (AT) commands are common to most of them. The AT command syntax is AT*argument*. The modem command prefix "AT" may be in uppercase or lowercase, but not mixed case. Any characters that follow the "AT" are treated as commands, and any characters preceding it are ignored.

The standard command for loading factory default settings is AT&F*argument*. The factory default settings are read-only. The *argument* can have a value of 0, 1, or 2, where 1 is the default.

S-registers are low-level modem service registers. You can modify modem behavior by setting numeric values in various modem control registers. The S0 (answer on ring) command, for instance, sets the modem to answer a call on a particular ring when it is in auto-answer mode. Its command syntax is ATS0=*argument*. The command ATS0=1 sets a modem to automatically answer all incoming calls on the first ring.

For lines configured with caller ID, automatic answering on the second ring is recommended. You use the command AT&C1 to get the CD signal to accurately reflect line state. Its command syntax is AT&C*argument*. Specifying 1 (the default value) as the argument causes the modem to send the CD signal when it connects with another modem and to drop the signal when it disconnects.

The command AT&D controls the Data Terminal Ready (DTR) signal from the DTE to the modem. Its command syntax is AT&D*argument*. The standard command for getting the modem to hang up at DTR low is AT&D3.

The characters +++ are used to put a modem in command mode. If you issue this command at the near-end modem, it is transmitted to the far-end modem. If the far-end modem tries to interpret it, this might cause the connection to hang. You can overcome this common bug by entering the ATS2=255 or ATS2=128 commands at the far-end modem. The function of the S2 (escape code character) command is to store the ASCII decimal code for the escape code character. Its command syntax is ATS2=*argument*.

You can get a modem not to echo keystrokes to DTE in command mode by using the command ATE0. You can turn echo back on by entering RATE or RATE1 (1 is the default value).

TheATM(speaker) command is used to control a modem's speaker. Its command syntax is ATM*argument*. The default argument is 1, which means that the speaker is on during dial-string execution and remains on until a carrier is detected or the modem goes on hook. The standard command for turning off external audio output from the modem is ATM0.

TheATZ command returns the Cisco modem user interface to its default state and re-executes the initialization string. ATZ99 returns to the standard Cisco IOS software user interface (EXEC) mode.

Nonstandard Modem Commands

In addition to standard modem commands, a number of nonstandard commands are essential for modems attached to Cisco routers. Let's take a look at some of these commands and how they are used by three prominent modem vendors—Microcom, Hayes, and U.S. Robotics (USR).

Any modem attached directly to a Cisco router needs to be configured for hardware flow control. USR modems use the command AT&H1&R2, where &H1 (transmit flow control) enables hardware flow control (CTS) and &R2 (receive hardware flow control) instructs the modem to send data to the DTE only if RTS is asserted. The AT\Q3 and AT&K3 commands are used to set hardware flow control on Microcom and Hayes modems, respectively.

The modem's serial port needs to be set to a fixed data-transfer rate. This means locking the DTE speed to prevent it from being negotiated down during the initial call setup. You use the DTE data rate command AT&B1 to lock the DTE speed on a USR modem. Specifying 1 as the command argument sets the DTE interface to follow the DTE data rate, regardless of the DCE connection rate. Microcom and Hayes lock the DTE speed using AT\J0 and AT&Q6, respectively.

You need to set the type of error control used on the modem. For a USR modem, you use the commandAT&M4 to configure automatic selection between V.42, MNP error control, and a non-error-controlled data link. In this case, 4 is the default argument. When no error control is selected, an MNP or V.42 link request is ignored. The equivalent Microcom and Hayes commands areAT\N6 and AT&Q5, respectively.

You need to ensure that the best compression algorithm negotiated between two communicating modems is used. For a USR modem, you use the command AT&K1 to configure automatic selection/deselection of Microcom Networking Protocol (MNP) level 5 or V.42bis data compression. This assumes that an MNP or a Link Access Procedure for Modem (LAPM) link has been established. Data compression is enabled only if the DTE data rate is higher than the link rate and the remote DCE supports either the MNP level 5 option in the MNP link request or V.42bis in the LAPM link request. The compression commands used by Microcom and Hayes are

AT%C1 and AT%Q9, respectively.

Show configuration commands allow you to display current modem settings. The USR modem inquiry command ATI4 gets the modem to send one screen of data to the DTE. The display indicates the settings for DTE band rate, parity, word length, S-register values, dial type, AT commands, and so on. AT/S1 and AT&V are the equivalent commands on Microcom and Hayes modems, respectively.

You need to be able to save any changes to the modem's configuration to its own nonvolatile RAM (NVRAM). Microcom, Hayes, and USR all use the command AT&W to achieve this.

You can use a modem's Help command to display all the AT commands for that modem. Microcom and Hayes both use AT$H, whereas USR uses AT$.

Initialization strings are used to send commands to modems before they dial out. No strings are required when you dial into a modem.

Configuring Chat Scripts

Asynchronous modems are not standard. This means that for optimal configuration, you must write custom chat scripts to perform certain tasks. A chat script is a string of text. It defines the handshaking that occurs between two DTE devices or between a DTE and its directly attached DCE (for example, an access server and a modem).

Here is the syntax for the chat-script command:

```
Router(config)#chat-scriptscript-name expect-string send-string
```

A chat script consists of *expect-send* pairs that define the string that the local system expects to see from the remote device and the reply that the local system should send.

Example 3-2 demonstrates a chat script. The chat script name is defined as dial. The *expect-send* pair ABORT ERROR stops the chat script if an error occurs. The *expect-send* pair " " "ATZ" sends the AT command to the modem to reset it using the stored profile. The empty *expect* string means that this task is performed without expecting an input string. OK "ATDT \T" specifies that when the input string OK is received, the AT command is sent to instruct the modem to dial the telephone number in *dialer-string* or the start-chat command. The chat script specifies that the access server will wait up to 30 seconds for the input string CONNECT. The argument\c indicates the end of the chat script.

Example 3-2. Chat Script

```
Router(config)#chat-script dial ABORT ERROR ABORT BUSY "" "ATZ" OK "ATDT \T"
  TIMEOUT 30 CONNECT \c
```

Chat scripts generally perform tasks such as

- Initializing the attached modem
- Instructing the modem to dial out
- Logging into a remote system

Thestart-chat command allows you to manually start a chat script on any asynchronous line that is not currently active. The command syntax is

```
Router#start-chatregexp [line-number [dialer-string]]
```

You can configure chat scripts so that they are executed automatically for specific events. For example, a chat script configured for line activation is triggered by incoming traffic (CD going high).

In addition to line activation, other events commonly trigger the execution of a chat script:

- Connection— Triggered by outgoing traffic such as reverse Telnet
- Line reset— Triggered by async line reset
- Startup— Triggered by access server startup
- Dialer— Triggered by dial-on-demand routing (DDR)

Modem Autoconfiguration

The Cisco IOS software provides a modem autoconfiguration feature that facilitates the configuration of modems on access servers. With the autoconfiguration feature, you can configure modems without having to resort to modem configuration commands. You can use the asynchronous interface to autodiscover the type of modem on the line and to use that modem configuration. You can configure non-Cisco-supported modems by specifying modem information in the modem-autoconfiguration chat scripts.

Using the autoconfiguration feature requires you to manage the modem capability (modemcap) database. This consists of a list of AT configuration commands for setting each modem type's attributes. Modemcap exists as a file in the Cisco IOS software.

With automatic modem configuration, a chat script is executed each time a modem is reset. This sends a string of modem-configuration commands to the modem. The command string is generated automatically whenever the modem is recycled. For example, if you were using an AppleTalk Remote Access Protocol (ARAP) dial-in modem configured with flow control, it would receive a string that included commands to

- Return to factory defaults
- Use hardware flow control
- Turn off error control

To set up autoconfiguration on a modem, you need to

- Connect the phone line and power to the modem
- Execute the **modem autoconfigure** command on the line with the modem

No other setup function is required for most modem configurations.

With automatic modem configuration, modems are configured to match current line settings. This means that the line configuration may be changed if the speed for the modem DTE differs from the current configuration on the line. You should, whenever possible, configure a line to expect a specific modem type. If none is specified, the access server tries to autodiscover the modem type. It does this by sending AT commands to the modem and then evaluating the response using the information in the modemcap database.

The access server's modemcap database has entries for several different modems. The actual entries in any particular modemcap database depend on the hardware and Cisco IOS version. If a particular modem is not currently supported, you can manually add it to the modemcap database so that it will be autodiscovered in future communication.

Modem Autoconfiguration Methods

There are two ways to configure modem autoconfiguration. You can configure modem autodiscovery, or you can specify a particular modem type to be used on the line. You also need to manage the modemcap database.

To configure modem autodiscovery, you use the following command:

```
Router(config-line)#modem autoconfigure discovery
```

Example 3-3 shows modem autodiscovery being configured on lines 1 through 16. The modem autoconfigure discovery command instructs the access server to send the AT string at various baud rates until successful reception is confirmed. This command also tells the access server to send a variety of AT commands in an effort to fully identify the modem from the entries in the access server's modemcap database.

Example 3-3. Configuring Modem Autodiscovery

```
Router(config)#line 1 16

Router(config-line)#modem autoconfigure discovery
```

The access server then builds the configuration string based on the discovered modem type and sends it to the modem. If the access server cannot identify the modem type, the default modem entry in the modemcap is used to build the configuration string.

If you know that the modem can be configured using one of the initialization strings in the modemcap database, you should use the following command to specify that modem type:

```
Router(config-line)#modem autoconfigure typemodem-type
```

Because of the overhead and the possibility of configuration ambiguities associated with modem autodiscovery, you should configure the modem type whenever possible. This means that whenever the line resets, it automatically sends the correct initialization command string to the modem.

If none of the strings in the modemcap database properly initializes the modem, you need to configure the modem manually. Alternatively, you can change the modemcap database.

Configuring the Modemcap Database

Let's take a closer look at the modemcap database. Modem attributes have a full name and a two-or three-letter abbreviation. For example, factory defaults are abbreviated as FD. You should be familiar with these abbreviations for efficient management of the modemcap database.

One of the basic tasks in managing the modemcap database is viewing the modem entries in the modemcap file. You can do this using the show modemcap command. To display the modemcap entry for a particular modem type, you need to include the modem type as an argument to the show modemcap command. Example 3-4 shows the modemcap entry for the Codex 3260.

Example 3-4. show modemcap modem-type Command Output

```
Router#show modemcap codex_3260
Modemcap values for codex_3260
Factory Defaults (FD): &F
Autoanswer (AA): S0=1
Carrier detect (CD): &C1
Drop with DTR (DTR): &D2
Hardware Flowcontrol (HFL): *FL3
Lock DTE speed (SPD): *SC1
Best Error Control (BER): *SM3
Best Compression (BCP): *DC1
No Error Control (NER): *SM1
No Compression (NCP): *DC0
No Echo (NEC): E0
No Result Codes (NRS): Q1
Software Flowcontrol (SFL): [not set]
Caller ID (CID): &S1
On-hook (ONH): H0
Off-hook (OFH): H1
Miscellaneous (MSC): [not set]
Template entry (TPL): default
Modem entry is built-in
```

The modemcap entry includes

- Command description
- Command abbreviation (in brackets)
- Command string

The default modem type has modemcap values for a few of the most common attributes, such as factory defaults and autoanswer. It has no command strings for attributes that vary widely with modem type, such as locking speeds, hardware flow control, compression, and error correction.

You can create a variant modemcap entry using the following command:

```
Router(config)#modemcap editmodem-name attribute at-command
```

This allows you to add a new modem to the modemcap database and add new attributes to an existing modem entry in the modemcap database.

Example 3-5 shows the creation of an entry for a new modem usr_new. The modemcap edit command creates the usr_new entry in the modemcap database and sets the new modem's caller ID to *U1. The second command locks the DTE speed on the usr_new modem. You can use themodemcap edit command to specify up to four layers of templates for the current modemcap entry. A template is another modemcap entry that the current entry points to. It's used to set any value not found in the current modemcap entry. In this example, usr_new points to the usr_courier modemcap entry as its template.

Example 3-5. Editing a Modemcap Entry

```
Router(config)#modemcap edit usr_new caller-id *U1

Router(config)#modemcap edit usr_new speed &B1

Router(config)#modemcap edit usr_new template usr_courier
```

Theshow modemcap command allows you to verify the access server's new modemcap entry. You can verify the new attribute values for a modemcap entry by specifying the modem name as an argument to the show modemcap command. The display for usr_new would be identical to that for usr_courier, except for the lock DTE speed, caller ID, and template attributes. You can also use the show running-config command to verify the attribute settings for a new modemcap entry.

You can remove a modem from the modemcap database by specifying its name as the only argument to the no modemcap edit command. If you specify the modem name and an attribute as arguments, the command removes only that modem attribute from the modem's modemcap entry.

Troubleshooting Modem Autoconfiguration

Here are some commands you can use to verify and debug modem autoconfiguration:

- debug confmodem— Displays the modem-configuration process.
- show line— Shows the type of modem configured on a line.
- clear line— Returns a line to its idle state. Returning a line to its idle state normally means that the line returns to being a terminal line, with the interface left in a down state.

Let's look at some common problems associated with modem autoconfiguration and discuss how you might troubleshoot them. If a modem fails to respond, you should first check that it's plugged in and turned on. You should check whether the power-up configuration is set to load factory defaults. You should determine whether you can connect to the modem through reverse Telnet. It's important to check that there is a dial tone at the phone jack. Also, sometimes the modem could have hung up. This fact can be verified by entering the show line command. If an * appears next to the line, issue the clear line*n* command to reset it.

If a modem is not recognized by modem autoconfigure discovery, you need to check what modem configuration the line is using. You can do this using the show line command. You should establish whether the Cisco access server recognizes the modem. You can overcome modem autodiscovery problems by using the modem autoconfigure type command to specify a particular modem type.

Modem autoconfiguration might fail because of a problem with an original modemcap entry. If you have configured your own modemcap entry and reconfiguration appears to function, you should verify that the DTR attribute is not set to &D3. The manual that accompanies your modem contains information that can be invaluable when troubleshooting any problems.

Scenarios

The scenarios presented in this chapter help you gain a better understanding of modem operation and configuration through practical application. You will go over the necessary configuration tasks in their logical progression. The two scenarios provided cover the following topics:

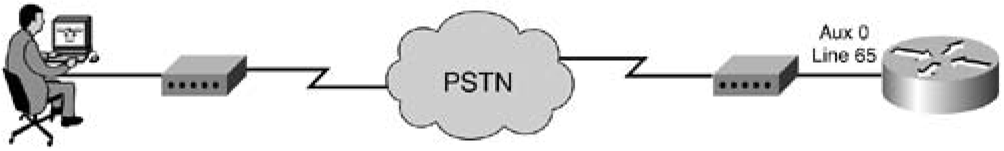
- Configuring the serial interface and asynchronous line on the central router
- Configuring the central-site modem

Scenario 3-1: Configuring the Serial Interface and Asynchronous Line on the Central Router

Several distinct configuration tasks must be performed to successfully establish a remote connection using asynchronous modems. The first are the initial configuration of the central-site router and the configuration of the serial interface and the corresponding line.

Suppose that a PC in a small office needs to communicate with a central site over a standard telephone line. The PC is running Windows 2000 and is fitted with an external modem. At the central site, the connection to the router is to be made through an external modem directly attached to the router's serial 0 port on the serial sync/async network module in slot 3. This scenario is depicted in Figure 3-2.

Figure 3-2. Serial Interface and Asynchronous Line on the Central Router



Two of the commands you'll need to complete the configuration are ip host and physical-layer. Theip host command is a global configuration command that allows you to define a static name-to-address mapping in the host cache:

```
Router(config)#ip host {name | tmodem-telephone-number} [tcp-port-number]
```

```
  {address1 [address2...address8]}
```

The*tcp-port-number* parameter lets you specify a TCP port number when connecting to the host name using Telnet.

The general syntax of the physical-layer command is as follows:

```
Router(config-if)#physical-layer {sync | async}
```

Because the default is synchronous mode, you use the keyword async to set the serial interface to asynchronous mode.

Step 1: Initial Configuration

Before configuring the asynchronous connection, you need to perform an initial configuration of the central-site router. You can do this from a terminal attached to its console port (line 0). You begin by entering global configuration mode. You can then configure the router name using the hostname command. It is also useful to disable the IP domain name system with the no ip domain-lookup command. This keeps the system from trying to translate domain names that have typing errors.

Next you need to select the routing protocol. To configure EIGRP routing, you use the router eigrp command. You must include the autonomous system number. This number is used to tag the routing information and to identify the routes to other EIGRP routers. You use the network command to specify the network serviced by the EIGRP routing protocol. Here it's 10.0.0.0.

You can use the enable secret command to enable a password for entering privileged EXEC mode. Here the password is cisco. This secret password provides an additional layer of security on the router. Passwords are case-sensitive strings that can be up to 80 characters long. They cannot begin with a number.

The central-site router is to connect to its local network through the Ethernet 0 port on the module in slot 0. You enter interface e 0/0 to configure this interface. But you can also use interface ethernet 0/0 and int eth 0/0. You set the IP address for the Ethernet interface using the ip address command. You also have to include a subnet mask. You then activate the interface using the no shutdown command.

To begin configuring the console line, you enter line console 0. You are now in line configuration mode. You use the no exec-timeout command to prevent the console from automatically disconnecting after a period of inactivity. The default timeout is 10 minutes.

The initial configuration of the central-site router is now complete. It's shown in Example 3-6.

NOTE

Don't forget to reset the exec-timeout after the configuration is completed. Leaving it open is a potential security risk.

Example 3-6. Initial Configuration of the Central-Site Router

```
Router#configure terminal

Router(config)#hostname R1

R1(config)#no ip domain-lookup

R1(config)#router eigrp 100

R1(config-router)#network 10.0.0.0

R1(config)#enable secret cisco

R1(config)#interface ethernet 0/0

R1(config-if)#ip address 10.115.0.120 255.255.255.0

R1(config-if)#no shutdown

R1(config-if)#line console 0

R1(config-line)#no exec-timeout
```

Step 2: The Serial Interface and Line

When the initial configuration of the central-site router is complete, you can begin configuring the serial interface and asynchronous line.

Configuring the Asynchronous Line

Let's assume that the external modem is directly attached to the serial 0 port on the serial network module in slot 3 (identified as port serial 3/0). You enter interface serial 3/0 to select the serial 3/0 interface. You are now in interface configuration mode. You must explicitly configure the interface as an asynchronous interface using the physical-layer async command. On the Cisco 3640 router, this adds TTY line 97 (TTY97) to the configuration.

Next you need to configure the line (line 97) with the appropriate physical layer parameters. You enterline 97 to begin the line configuration. This puts you in line configuration mode. To prevent unauthorized connections, you use the login command to enable user login to the interface and to challenge for a password. You set the login password using the password command. Here, it's cisco. To allow incoming and outgoing connections on the line, you enter modem inout. You want to allow any transport protocol to pass to the router through the line. You achieve this by entering the transport input all command.

You use the speed command to set the maximum speed between the router and the modem. Here, it's 115200 (bps). You use the stopbits command to set the number of stopbits per byte of data. This example has one stopbit per byte. You should configure the line to use CTS/RTS signals for hardware flow control.

This completes the configuration of the line. You return to global configuration mode by entering exit. All configuration tasks described in this section are shown in Example 3-7.

Example 3-7. Asynchronous Line Configuration

```
R1(config)#interface serial 3/0

R1(config-if)#physical-layer async

R1(config-if)#line 97

R1(config-line)#login

R1(config-line)#password cisco

R1(config-line)#modem inout

R1(config-line)#transport input all

R1(config-line)#speed 115200

R1(config-line)#stopbits 1

R1(config-line)#flowcontrol hardware

R1(config-line)#exit
```

Configuring Reverse Telnet

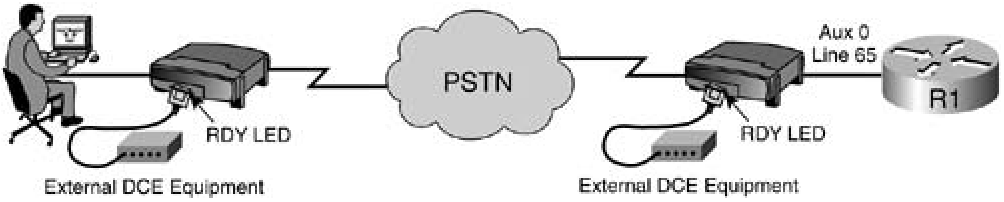
You need to be able to reverse-Telnet to the attached modem. You can assign it a host name and associate it with the router's Ethernet IP address and with the Telnet TCP port corresponding to the line. You enter ip host modem 2097 10.115.0.120 to define the host name "modem" and associate it with TCP port 2097 and IP address 10.115.0.120. You can now exit configuration mode. You can save your configuration to NVRAM by entering copy running-config startup-config.

Scenario 3-2: The Central-Site Modem

Cisco access servers support reverse-Telnet connections. This means that you can connect through an access server to an attached modem to configure that modem.

Suppose that a remote PC fitted with an external modem needs to communicate with a central site over a standard telephone line. At the central site, the connection to the router is to be made through an external modem directly attached to one of the router's serial ports. The external modem has already been assigned the host name "modem." The topology is shown in Figure 3-3.

Figure 3-3. Central-Site Modem



The first part of this scenario covers the manual method of modem configuration, and the second part presents an alternative to the manual method—the autoconfiguration technique.

Manual Modem Configuration

When manually configuring a modem, you follow these steps:

Step 1. Connect via reverse Telnet.

Step 2. Configure the central-site modem.

Step 3. Configure the PC modem.

Step 1: Connecting via Reverse Telnet

Let's look at how you can configure the central-site modem using reverse Telnet. To connect to the modem on line 97 using reverse Telnet, you enter **modem** or **telnet modem**. Here **modem** is the host name of the modem configured to the router's line 97. Remember that the **ip host modem 2097 10.115.0.120** command has already been configured.

The system then prompts you for a login password. You respond by entering the appropriate login password. Here it was previously configured to "cisco." You enter AT (uppercase or lowercase) and press Enter to verify connectivity to the modem. The modem should respond with an OK message.

Step 2: Configuring the Central-Site Modem

You can now begin entering AT commands to configure the modem. These commands are specific to your modem manufacturer and are not always the same on different modems. You should therefore contact your modem manufacturer for a complete list of the AT commands relevant to your modem. You can normally obtain a list of these commands from the modem itself by entering AT$.

Let's assume that a USR Sportster modem is being configured. You use the command AT&F0 to configure the modem to load the factory default settings. The Carrier Detect (CD) and Data Terminal Ready (DTR) signals are used between the DTE and DCE to initiate and receive calls. You use the command AT&C1 to set the modem to operate only when the proper carrier signal is present. This conforms to RS-232 standard operation, where the CD signal should accurately reflect the current line state.

To have the modem to hang up on DTR going low, you enter the command AT&D2. You use the commandAT&H1 to configure the modem for hardware flow control. The modem should send data to the router only if request to send (RTS) is asserted, so you enter the command AT&R2. The command AT&M4 is used to set error correction. This allows the modem to automatically select between V.42, MNP, and no error correction.

To set a fixed data transfer rate on the serial port between the modem and the router, you enter the command AT&B1. To set the modem to autoselect the best compression algorithm (MNP level 5 or V.42bis), you use the command AT&K1. The command AT&W0 is used to store the new configuration to the modem's NVRAM (pattern 0). You can check the configuration by using the command ATI4.

After manually configuring the central-site modem, you can leave the reverse-Telnet session by pressingCtrl+Shift+6 and then x. You must then enter disconnect to clear the Telnet session. If you fail to do this, you will not be able to reconnect.

Step 3: Configuring the PC Modem

After configuring the central-site modem, you need to configure the PC modem. The AT commands required are specific to your modem manufacturer.

Let's assume that the PC modem is a USR Sportster model. You can use the Hyperterminal communications utility to access the modem. Hyperterminal is a communications software utility that comes with Windows OS. When you have access, you need to verify connectivity to the modem by entering the command AT. The modem should respond with an OK message.

For simplicity, let's say that you want to use all the modem's factory default settings. So you enter the command AT&F to load these settings. You then save the configuration to the modem's NVRAM using the command AT&W. You can compound AT commands to speed up configuration. For example, you can enter the command AT&F&W instead of the separate commandsAT&F and AT&W.

After configuring the PC modem, you can connect to the central-site router using the ATDT command and the appropriate phone number. As soon as the modems have successfully synchronized, you should receive the prompt "User access verification" and be asked for the login password. When you have gained access, you should see the console prompt from the central-site router.

To verify connectivity to line 97 (TTY97), you first enter privileged EXEC mode. You do this by enteringenable and then entering the appropriate password. Then you enter either show users orshow line 97 at the prompt. The show users command shows which users have active connections to the router, the lines that they are connected through, and how long they have been connected. The show line command lists the parameters for a specified line. It also shows some activity information associated with the line.

You can verify the running configuration using the show running-config command. In Example 3-8 you can see that domain name lookup is disabled, the modem is bound to IP address 10.115.0.120, and the serial 3/0 interface—the modem-connected interface—is set to asynchronous mode. The output also confirms the configuration settings for line 0 (the console line) and line 97.

Example 3-8. Running the R1 Configuration

```
R1#show running-config
!
hostname R1
!
enable secret 5 $1$FaD0$Xyti5Rkls3LoyxzS8
!
no ip domain-lookup
ip host modem 2097 10.115.0.120
!
interface Serial 3/0
 physical-layer async
 no ip address
!
line con 0
 exec-timeout 0 0
line 65 70
line 97
 password cisco
 login
 modem InOut
 transport input all
 stopbits 1
 speed 115200
 flowcontrol hardware
line aux 0
line vty 0 4
```

To disconnect the PC modem, you enter quit.

Autoconfiguring the Central Modem

You should follow up the modem autoconfiguration by verifying connectivity.

Step 1: Central-Site Modem Autoconfiguration

Let's assume that you are in privileged EXEC mode on the central-site router. You turn on debugging for modem configuration by entering debug confmodem. This allows you to see the processes occurring while the router is configuring the modem.

Next you enter configuration mode using the configure terminal command. You select the configuration for the modem line (line 97) by entering line 97. This puts you in line configuration mode. To set the router to autoconfigure the modem and to find its type using autodiscovery, you enter modem autoconfigure discovery. If you already know that the modem type is usr_sportster, you should avoid autodiscovery by entering this command. Telling the router the modem type reduces unnecessary work for the router's processor.

The debug messages in Example 3-9 show that the modem type has been successfully discovered. The router has automatically loaded AT commands to configure it.

Example 3-9. debug confmodem Messages

```
R1#debug confmodem
TTY97:detection speed(115200) response ---OK---
TTY97:Modem type is usr_sportster
TTY97:Modem command: --AT&F&C1&D2&H1&B2&M4&K1&B1S0=1H0--
TTY97:Modem configuration succeeded
TTY97:locking speed(115200) response ---OK---
TTY97:locked DTE speed at 115200
```

```
TTY97:Done with modem configuration
```

To leave configuration mode, you enter exit at successive prompts. You turn off modem configuration debugging by entering the no debug confmodem command. This helps reduce any unnecessary overhead processing for the router.

Step 2: Verifying Connectivity

As soon as the central-site modem has been autoconfigured, you need to verify connectivity. You can do this using a modem at any remote site—it could be attached to a PC or to another router. Let's assume that the PC modem used to dial in represents a telecommuter. The phone number of the central-site modem is 5551005.

To dial into the central-site router, you use Hyperterminal and the ATDT command. Here you enterATDT 5551005. The router responds by challenging you for a login password. On successful connection, you should see the console prompt from the central-site router. By opening this session with the routers, you verify connectivity. You disconnect from the modem by entering quit.

If the connection fails, from the central-site router you could reverse-Telnet to the modem and check its settings using the command ATI4. You could then check the settings for the line on the router by entering show line 97 at the privileged EXEC prompt. You can also check the status of the router's serial port attaching to the modem by entering show interface serial 3/0.

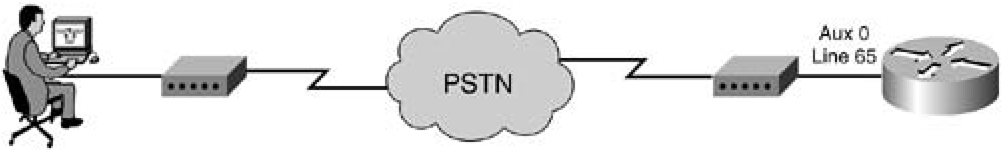
If the settings used by the autoconfiguration process for your modem are inaccurate, or the router has incorrectly detected the modem, you can edit the current entry to suit your modem, put a new entry into the modemcap database, or use a manual configuration process.

Practical Exercise 3-1: Configuring a Modem on the AUX Port for EXEC Dial-in Connectivity

In many situations, it might be necessary to allow a router to accept interactive command processing of Cisco IOS (EXEC) calls with a modem connected to the router's auxiliary (AUX) port. This document provides the necessary configuration tasks to configure such a scenario.

This exercise uses the network setup shown in Figure 3-4.

Figure 3-4. Modem on the AUX Port for EXEC Dial-in Connectivity



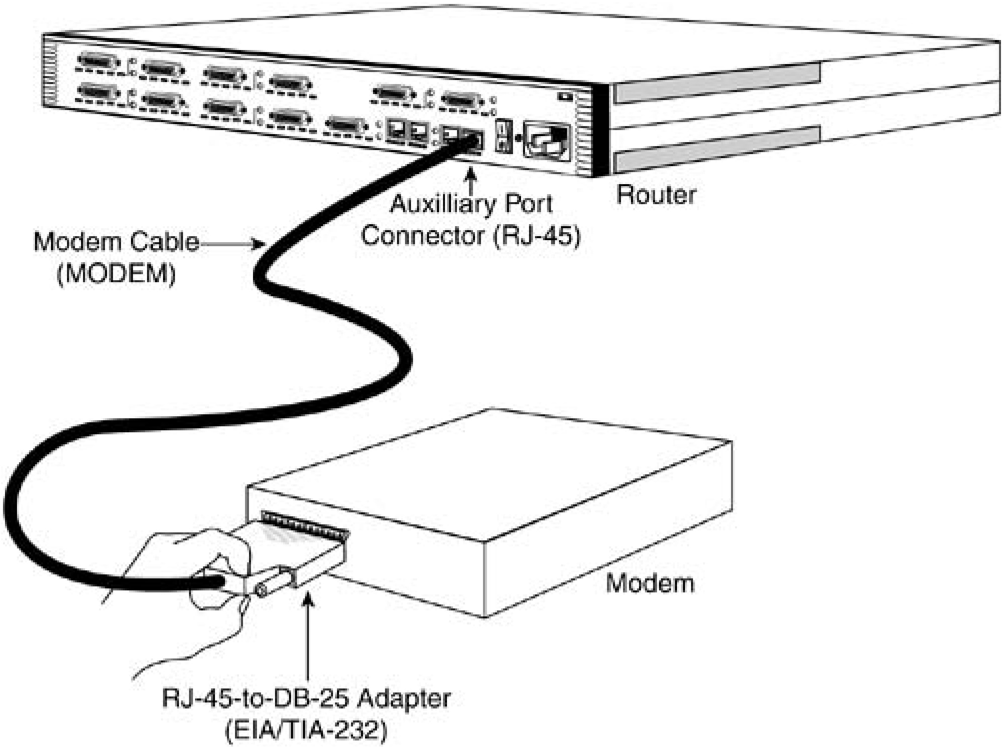
Practical Exercise 3-1 Solution

Use the following steps to configure a modem on the AUX port for EXEC dial-in connectivity:

Step 1. Connect the cable from the router AUX port to the modem.

The AUX port on a Cisco router is either RJ-45 or DB-25. If the AUX port is RJ-45, use a flat-satin rolled RJ-45-RJ-45 cable (part number CAB-500RJ=), which is usually provided with every Cisco router for console connections. You also need an RJ-45-to-DB-25 adapter marked "MODEM" (part number CAB-25AS-MMOD) to connect the rolled cable to the DB-25 port on the modem, as shown in Figure 3-5.

Figure 3-5. Connecting the Rolled Cable to the DB-25 Port on the Modem



If your router has a DB-25 AUX port, use a straight-through DB-25Female-DB25Male RS-232 cable to connect the modem to the router.

Step 2. Use the show line command to determine the AUX port's async interface. Although most routers have the AUX port as line 1, access servers have the AUX port interface after the TTY lines. For example, if your router has 16 async/modem lines, the AUX port is line 17. Configure the AUX port based on the show line outputs. Example 3-10 verifies that the AUX port configuration is on interface line 65.

Example 3-10. show line Command Output

```
R1#show line
   Tty Typ     Tx/Rx    A Modem  Roty AccO AccI   Uses   Noise  Overruns   Int
*    0 CTY              -    -      -    -    -      0       0     0/0        -
    65 AUX   9600/9600  -    -      -    -    -      0       1     0/0        -
    66 vty              -    -      -    -    -      0       0     0/0        -
    67 vty              -    -      -    -    -      0       0     0/0        -
    68 vty              -    -      -    -    -      0       0     0/0        -
    69 vty              -    -      -    -    -      0       0     0/0        -
    70 vty              -    -      -    -    -      0       0     0/0        -
```

Step 3. Use the commands shown in Example 3-11 to configure the router AUX line.

Example 3-11. Line Configuration

```
R1(config)#line 65
R1(config-line)#modem inout
R1(config-line)#speed 115200
R1(config-line)#transport input all
R1(config-line)#flowcontrol hardware
R1(config-line)#login
R1(config-line)#password cisco
```

Step 4. Reverse-Telnet to the modem and configure the appropriate initialization string, as shown in Example 3-12.

Example 3-12. Modem Configuration via Reverse Telnet

```
R1#telnet 172.22.53.145 2065
```

```
Trying 172.22.53.145, 2065 ... Open

at

OK

at&f1s0=1

OK

at&w

OK

R1#disconnect 1

Closing connection to 172.22.53.145 [confirm]

R1#
```

Step 5. Use an analog phone to verify that the phone line is active and functioning. Then connect the analog phone line to the modem.

Step 6. Test the modem connection by initiating an EXEC modem call to the router from another device (for example, a PC). Use a terminal emulation program on the PC, such as Hyperterminal, and access the PC's modem through one of the COM ports. Once you have connected to the PC's modem through the COM port, initiate the dial to the router.

Step 7. As soon as the connection is established, the dial-in client is prompted for a password. Enter the correct password. This password must match the one configured on the AUX port line.

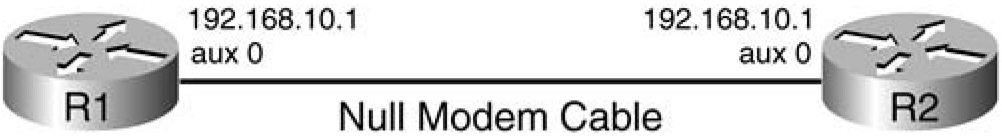
Practical Exercise 3-2: Connecting Routers Back-to-Back Through the AUX Ports

This sample configuration shows you how to directly connect two routers without using a modem or other DCE devices. In this configuration, two Cisco routers are connected back-to-back through the asynchronous AUX ports using a null modem cable (rollover cable). The AUX ports of the two routers are directly connected using a rollover cable with Point-to-Point Protocol (PPP) running on the link. The AUX ports are DTE devices. Connecting DTE to DTE devices requires a null modem cable (rollover cable).

A flat-satin rollover (null modem) cable (part number CAB-500RJ=) is usually provided with every Cisco router to allow for RJ-45 console connectivity. If the AUX port is a DB-25, use an RJ-45-to-DB-25 adapter marked "terminal" with the null modem cable (rollover cable).

This exercise uses the network setup shown in Figure 3-6.

Figure 3-6. Routers Back-to-Back Through the AUX Ports



Practical Exercise 3-2 Solution

Configure the async interface corresponding to the AUX port. Use the show line command to determine which async interface corresponds to the AUX port. Make sure the IP addresses on the AUX ports of both routers are in the same subnet. Specify PPP as the encapsulation for the async interface with the encapsulation ppp command. Allow routing protocols on the link with the async dynamic routing command. The encapsulation ppp and async dynamic routing commands are described in greater detail in Chapter 5, "Configuring Point-to-Point Protocol and Controlling Network Access."

Configure the default route to point to the Async1 (AUX port) interface. Configure the line for the AUX port. Allow all protocols to use the line. Set the RX speed (identical to the other router's TX speed). Set the TX speed (identical to the other router's RX speed). The routers' configuration is shown in Example 3-13.

Example 3-13. Routers' Configuration

```
R1#show running-config
 hostname R1
 !
!
 interface Async1
  ip address 192.168.10.1 255.255.255.0
  encapsulation ppp
  async mode dedicated
 !
 no ip classless
 ip route 0.0.0.0 0.0.0.0 Async1
 logging buffered
 !
 line con 0
  exec-timeout 0 0
 line aux 0
  modem InOut
  transport input all
  speed 38400
  flowcontrol hardware
 line vty 0 4
 !
 end
```

```
R2#show running-config
 hostname Router2
 !
 interface Ethernet0
  ip address 10.1.1.1 255.255.255.0
 !

 interface Async1
  ip address 192.168.10.2 255.255.255.0
  encapsulation ppp
  async mode dedicated
 !
 no ip classless
 ip route 0.0.0.0 0.0.0.0 Ethernet0
 logging buffered
 !
 line con 0
  exec-timeout 0 0
 line aux 0
  modem InOut
  transport input all
  speed 38400
  flowcontrol hardware
 line vty 0 4

 !
 end
```

Summary

This chapter explained modem connections and operation. DTE devices such as PCs communicate with each other through DCE devices such as modems. A modem converts digital signals into analog signals and vice versa. The EIA/TIA-232 standard defines the interface between DTE and DCE.

You have seen the configuration methods and commands used to establish an asynchronous connection through an analog modem. Cisco access servers support two types of connections to a modem—incoming asynchronous line (forward) connections and outgoing asynchronous line (reverse) connections.

You have seen how to configure the central-site modem. You can use reverse Telnet to manually configure the central-site modem or use autoconfiguration. You can verify connectivity to the modem using the modem command AT.

Review Questions

1: Which of following signals does a DTE use to indicate to a DCE that it is ready to accept an incoming call?

A. DSR

B. DTR

C. RTS

D. CTS

2: The DTR, CD, and DSR signals belong to which group of signals?

A. Hardware flow control

B. Modem control

C. Data transfer

3: For which type of connection is null modem cable required?

A. DTE-DCE

B. DCE-DCE

C. DCE-DTE

D. DTE-DTE

4: What command would you use to display status information for all line types?

A. show running-config

B. show line all

C. show line

D. show aux tty vty con

5: Which line type would you associate with line number 0?

A. AUX

B. TTY

C. vty

D. CON

6: Which of the following AT commands are common to most modem types?

A. AT&B1

B. AT&F

C. AT&K1

D. AT&D3

E. ATS2=255

F. AT&M4

7: Why would you use the modem autoconfiguration feature?

A. To configure a modem automatically

B. To autodiscover modems

C. To update the modemcap database

D. To configure non-Cisco modems

Chapter 4. Using Cable Modems to Access a Central Site

This chapter covers the following topics:

- Cable Modem Technology Overview
- Basic Cable Modem Troubleshooting Using Cisco IOS Software Commands

The first section of this chapter covers cable modem technology. Some of the key terminologies are explained briefly before you configure Cisco's Cable Modem Termination System (CMTS) and cable modem (CM).

Cisco uBR7246 is used as an example in this chapter to explain the basic configuration of the Cisco CMTS equipment. Two different configurations for the Cisco cable access router, bridging and routing, are also covered so that you can be educated from both the service provider and end user sides.

Finally, the troubleshooting section helps you understand the cable modem initialization process and learn the Cisco IOS software commands to troubleshoot from both the CMTS and CM sides.

Cable Modem Technology Overview

Data Over Cable Service Interface Specifications (DOCSIS) is a project that was developed to provide a set of necessary communications and operations support interface specifications through which cable companies can achieve cross-platform functionality. In essence, DOCSIS can guarantee interoperability by establishing standards for carrying IP packets over an HFC cable TV network. Figure 4-1 illustrates the DOCSIS protocol stack compared to the OSI model. Some of the key terminologies and DOCSIS specifications will be explained briefly.

Figure 4-1. DOCSIS Protocol Stack

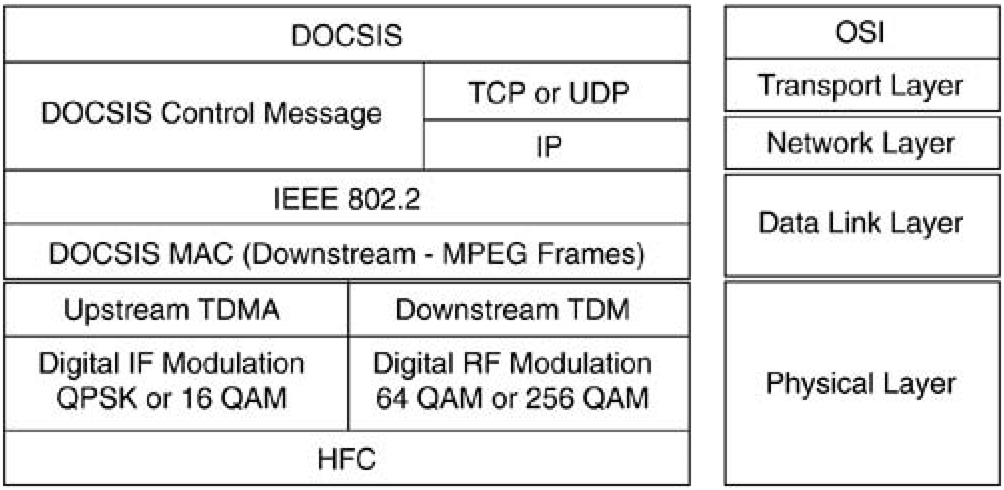
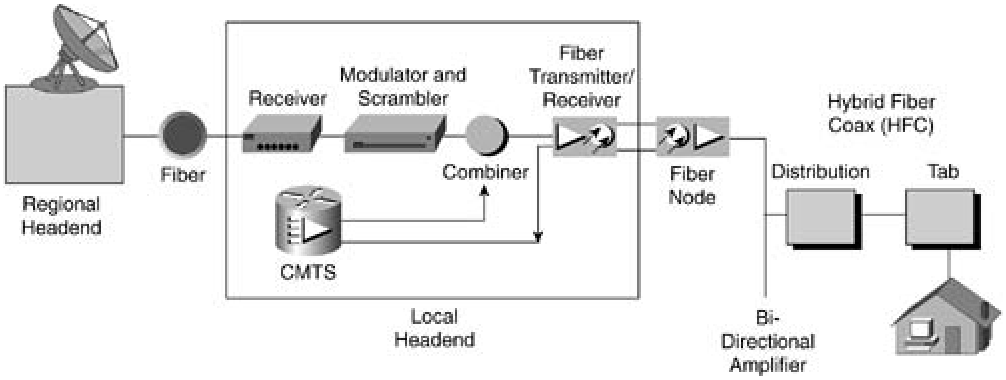


Figure 4-2 illustrates a typical CATV and two-way data network. The Hybrid Fiber Coax (HFC) portion refers to any configuration of fiber optic and coaxial cable that is used to distribute broadband communications such as voice, video, and data. The HFC network connects subscribers to the cable headend and video flows as analog radio frequency or optical signals. Optical fiber brings the signal from the headend to fiber nodes that serve 500 to 2000 homes. Fiber optic is used because it has lower signal power loss and is less susceptible to noise compared to coaxial cable for long distances. Fiber node converts optical signals from fiber to electrical signals on 75-ohm coaxial cable. Coaxial cable has higher signal power loss than fiber, but it is a more cost-effective way to reach subscribers.

Figure 4-2. Typical CATV and Two-Way Data Network



You can see from Figure 4-2 that the regional headend and local headends are connected via the high-speed fiber network. The video signal flows in analog or digital formats over the fiber network. Usually the regional headend receives national channels from satellites and transmits them to various local headends. Local headends may receive local channels as well as national channels from the regional headend. They selectively process and transmit them to subscribers based on individual requests or demographic group needs. All the video channels are modulated and sent to the combiner at the local headend. The downstream port of the CMTS is connected to the up-converter, and the output signal goes into the combiner. In essence, the output signal (6 MHz wide) becomes one of the video channels that is sent downstream for data communication. Please note that to achieve two-way data network, bidirectional amplifiers are required.

Downstream and Upstream

Downstream is the term used for the signal received by the cable modem. In other words, the signal flows from the headend toward the subscribers. It is also called *forward path. Upstream* is the term used for the signal transmitted by the cable modem. The signal flows from the subscribers to the headend. It is also called the *return path* or *reverse path*.

Modulation Modes

Digital modulation is the physical layer of the DOCSIS protocol stack. The different types of modulation modes are as follows:

- Quadrature Phase Shift Keying (QPSK)— A digital modulation method in which 2 data bits are represented with each baud symbol. QPSK is used for upstream transmission.
- Quadrature Amplitude Modulation (QAM)— A digital modulation method in which the value of a symbol consisting of multiple bits is represented by a carrier's amplitude and phase states. Typical QAM types are

  -16-QAM (4 bits per symbol)— Used for upstream transmission.

  -64-QAM (6 bits per symbol)— Used for downstream transmission.

  -256-QAM (8 bits per symbol)— Used for downstream transmission.

Spectrum Sharing

Time-division multiplexing (TDM) permits timeslots within a channel to be shared by multiple subscribers. TDM is used for downstream transmission, in which only one transmitter is involved. *Time-division multiple access (TDMA)* allows multiple subscribers to transmit sequentially to a common receiver. It is used for upstream or return transmission in which a number of transmitters need to communicate with the headend.

DOCSIS Hardware Specifications

Tables 4-1 through 4-4 show the DOCSIS hardware specification. The DOCSIS hardware must meet or exceed the published specifications for the cable access solution to work properly.

Table 4-1 summarizes key parameters of the upstream signal.

Table 4-1. Upstream Characteristics

Frequency Range	5 to 42 MHz
Bandwidth	200, 400, 800, 1600, 3200 KHz
Modulation Mode	QPSK or 16-QAM
Symbol Rates	160, 320, 640, 1280, 256 Ksym/sec

Table 4-2 summarizes key parameters of the downstream signal.

Table 4-2. Downstream Characteristics

Frequency Range	88 to 860 MHz
Bandwidth	6 MHz
Modulation Mode	64-QAM or 256-QAM
Symbol Rates	5.056941 or 5.360537 Msym/sec

Table 4-3 summarizes the incoming upstream signals that need to be supported by the CMTS receiver. Downstream RF output is also specified in this table.

Table 4-3. CMTS Power Level Range

Upstream	Power Level Range
200 KHz	–16 to +14 dBmV[1]
400 KHz	–13 to +17 dBmV
800 KHz	–10 to +20 dBmV
1600 KHz	–7 to +23 dBmV
3200 KHz	–4 to +26 dBmV
Downstream	Power Level Range
6 MHz	+50 to +61 dBmV

[1] A dBmV is the power of a signal in comparison to the power of a 1 mV signal when applied to 75-ohm resistance. The dBmV is used as the unit of radio frequency (RF) power in the cable industry. The coax cables used in the cable industry are usually 75-ohm.

Table 4-4 summarizes input and output signal levels for the cable modem.

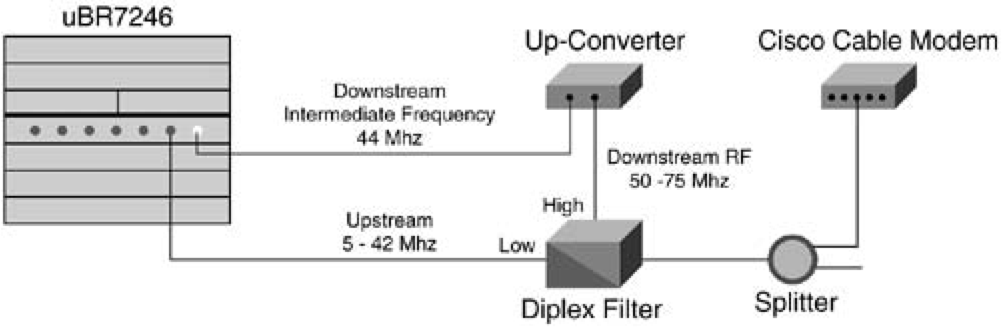
Table 4-4. Cable Modem Power Level Range

Output	Power Level Range
QPSK	+8 to +58 dBmV
16-QAM	+8 to +55 dBmV
Input	Power Level Range
64-QAM/256-QAM	–15 to +15 dBmV

Scenarios

Figure 4-3 illustrates the cable modem lab topology. A Cisco uBR7246 with an MC-16C cable modem card installed in slot 3 is used in the lab. With the MC-16C card, you get one downstream port and six upstream ports. In this lab, the downstream port is connected to the Wavecomm up-converter at 459 MHz, and upstream port 0 is used for the upstream transmission. Note that Dynamic Host Configuration Protocol (DHCP), Time of Day (ToD), and TFTP servers are required but are not shown in Figure 4-3.

Figure 4-3. Cable Modem Lab Topology



Scenario 4-1: Cisco CMTS Minimum Configuration Requirements

In this scenario, you will learn the minimum configuration requirements for the Cisco CMTS. You will also learn the configuration's syntax and commands to verify the configuration.

The Cisco CMTS minimum configuration requirements are as follows. They are required for link establishment between the CMTS and the cable modem:

- Set the upstream frequency
- Enable the upstream port
- Configure the IP address
- Configure the helper address

Setting the Upstream Frequency

You need to configure a fixed frequency of the upstream RF carrier for an upstream port. You should make sure that the upstream frequency of your RF output complies with the expected input frequency of your Cisco MC16C cable modem card. The valid range for a fixed upstream frequency is 5 to 42 MHz.

NOTE

You need to make sure that the upstream frequency you choose does not interfere with the frequencies used for any other upstream applications running in the cable plant. The cable interface will not operate until you either set a fixed upstream frequency or create and configure a spectrum group. This is covered later in this chapter.

Use the following commands to set the upstream frequency in cable interface configuration mode:

```
uBR7246(config-if)#interface cableslot/port

uBR7246(config-if)#cable upstreamportfrequencyreturn frequency
```

InExample 4-1, the upstream frequency is set to 39 MHz. In Example 4-2, the command show controller cable 3/0 upstream 0 displays the upstream frequency. Note that Cisco cable interface line cards always program the upstream's center frequency in 16-KHz increments. This is the frequency displayed by the show controller cable upstream command. In Example 4-2, the actual center frequency is 38.992 MHz.

Example 4-1. Upstream Frequency Configuration

```
uBR7246(config-if)#interface cable 3/0

uBR7246(config-if)#cable upstream 0 frequency 39000000
```

Example 4-2. Verifying the Upstream Frequency

```
uBR7246#show controller cable 3/0 upstream 0

 Cable3/0 Upstream 0 is up

  Frequency 38.992 MHz, Channel Width 3.200 MHz, 16-QAM Symbol
  Rate 2.560 Msps
  SNR 28.6280 dB
  Nominal Input Power Level 0 dBmV, Tx Timing Offset 2744
  Ranging Backoff automatic (Start 0, End 3)
  Ranging Insertion Interval automatic (60 ms)
  Tx Backoff Start 0, Tx Backoff End 4
  Modulation Profile Group 5
  Concatenation is enabled
  part_id=0x3137, rev_id=0x03, rev2_id=0xFF
  nb_agc_thr=0x0000, nb_agc_nom=0x0000
  Range Load Reg Size=0x58
  Request Load Reg Size=0x0E
  Minislot Size in number of Timebase Ticks is = 8
  Minislot Size in Symbols = 128
  Bandwidth Requests = 0xAC3C
  Piggyback Requests = 0x84
  Invalid BW Requests= 0x22
  Minislots Requested= 0x3EAD8
  Minislots Granted  = 0x3EAD8
  Minislot Size in Bytes = 64
  Map Advance (Dynamic) : 2447 usecs
  UCD Count = 303031
 DES Ctrl Reg#0 = C000C043, Reg#1 = 0
```

Enabling the Upstream Port

Each upstream port must be activated to enable upstream data from the cable modems on the HFC network to the Cisco uBR7246.

To activate the upstream ports, use the following commands in global configuration mode:

```
uBR7246(config)#interface cableslot/port

uBR7246(config-if)#no cable upstreamportshutdown
```

Example 4-3 shows how to activate upstream port 0. Recall that an MC16C card is used in this lab. It is installed in slot 3. Upstream port 0 is used for upstream communication between the CMTS and the cable modem.

Example 4-3. Enabling the Upstream Port

```
uBR7246(config)#interface cable 3/0

uBR7246(config-if)#no cable upstream 0 shutdown
```

To verify whether the upstream ports are enabled or disabled, enter the show interface cable command for the upstream port you have configured, as shown in Example 4-4.

Example 4-4. Verifying the Upstream Port

```
uBR7246#show interface cable 3/0 upstream 0

Cable3/0: Upstream 0 is up

     Received 144 broadcasts, 12489 multicasts, 209258 unicasts

...
```

Configuring the IP Address and Helper Address

Configuring IP address in CMTS is the same way when you configure other Cisco IOS routers.

```
uBR7246(config)#interface cableslot/port

uBR7246(config-if)#ip addressIP address IP subnet mask
```

The helper address provides a way for packets from the cable modem and the PC to locate their supporting DHCP server, from which they receive their IP address and the address of their supporting TFTP and ToD servers.

```
uBR7246(config)#interface cableslot/port

uBR7246(config-if)#cable helper-addressIP address
```

Example 4-5 shows the syntax to configure the DHCP server's IP address.

Example 4-5. Configuring the Helper Address

```
uBR7246(config)#interface cable 3/0

uBR7246(config-if)#cable helper-address 10.1.1.5
```

NOTE

DOCSIS mandates that the DHCP, ToD, and TFTP servers be part of the cable access solutions. Cisco Network Registrar software can be used as the DHCP and TFTP servers, or you can configure the DHCP, ToD, and TFTP services on Cisco's CMTS.

Scenario 4-2: Cisco CMTS Optional Configuration

In this scenario, you will learn how to configure some of the optional configuration for Cisco CMTS and command syntax. Some parameters don't have to be modified, but they are listed here for your reference.

Setting the Upstream Input Power Level

The uBR7246 controls the cable modems' output power levels to meet the desired upstream input power level. The default setting of 0 dBmV is the optimal setting for the upstream power level.

> NOTE
>
> If you increase the input power level, the cable modems on your HFC network increase their transmit power level. This might cause an increase in the network's carrier-to-noise ratio (CNR). Be careful if you adjust this parameter. You might violate the upstream return laser design parameters.

To set the upstream input power level, use the following commands in cable interface configuration mode:

```
uBR7246(config-if)#interface cableslot/port

uBR7246(config-if)#cable upstreamportpower-leveldBmV
```

InExample 4-6, the power level is set to 0 dBmV for upstream channel 0. Again, it is the default setting and is the optimal setting for the upstream power level.

Example 4-6. Configuring the Upstream Input Power Level

```
uBR7246(config-if)#interface cable 3/0
```

```
uBR7246(config-if)#cable upstream 0 power-level 0
```

To verify the current value of the upstream input power level, enter the show controller cable command for the upstream port you just configured, as shown in Example 4-7.

Example 4-7. Verifying the Upstream Input Power Level

```
uBR7246#show controller cable 3/0 upstream 0

 Cable3/0 Upstream 0 is up

  Frequency 38.992 MHz, Channel Width 3.200 MHz, 16-QAM Symbol

  Rate 2.560 Msps

  Spectrum Group 20

  SNR 28.6280 dB

  Nominal Input Power Level 0 dBmV, Tx Timing Offset 2744

  Ranging Backoff automatic (Start 0, End 3)

  Ranging Insertion Interval automatic (60 ms)

  Tx Backoff Start 0, Tx Backoff End 4

  Modulation Profile Group 5

  Concatenation is enabled

  part_id=0x3137, rev_id=0x03, rev2_id=0xFF

  nb_agc_thr=0x0000, nb_agc_nom=0x0000

  Range Load Reg Size=0x58

  Request Load Reg Size=0x0E

  Minislot Size in number of Timebase Ticks is = 8

  Minislot Size in Symbols = 128

  Bandwidth Requests = 0xAC3C

  Piggyback Requests = 0x84

  Invalid BW Requests= 0x22

  Minislots Requested= 0x3EAD8
  Minislots Granted  = 0x3EAD8
  Minislot Size in Bytes = 64
  Map Advance (Dynamic) : 2447 usecs
  UCD Count = 303031
  DES Ctrl Reg#0 = C000C043, Reg#1 = 0
```

Configuring the Upstream Channel Bandwidth

By default, the upstream RF bandwidth is set to 1600 KHz. The command to configure the upstream channel bandwidth is as follows:

```
uBR7246(config)#interface cableslot/port
uBR7246(config-if)#cable upstreamportchannel-width [200000 | 400000 | 800000 |
  1600000 |3200000]
```

Example 4-8 shows you how to configure the channel width for upstream port 0. You can also use the show controller cable command to view the channel width configuration of the upstream port you just configured, as shown in Example 4-9.

Example 4-8. Configuring the Upstream Channel Bandwidth

```
uBR7246(config)#interface cable 3/0
uBR7246(config-if)#cable upstream 0 channel-width 3200000
```

NOTE

Before increasing the channel width or modulation, you should perform a thorough analysis of your upstream spectrum using a spectrum analyzer to find a wide-enough band with adequate CNR. Failure to do so can potentially affect other services in your cable network.

Example 4-9. Verifying Upstream Channel Width

```
uBR7246#show controller cable 3/0 upstream 0
 Cable3/0 Upstream 0 is up
  Frequency 38.992 MHz, Channel Width 3.200 MHz, 16-QAM Symbol
  Rate 2.560 Msps
  Spectrum Group 20
  SNR 28.6280 dB
  Nominal Input Power Level 0 dBmV, Tx Timing Offset 2744
  Ranging Backoff automatic (Start 0, End 3)
  Ranging Insertion Interval automatic (60 ms)
  Tx Backoff Start 0, Tx Backoff End 4
  Modulation Profile Group 5
  Concatenation is enabled
  part_id=0x3137, rev_id=0x03, rev2_id=0xFF
  nb_agc_thr=0x0000, nb_agc_nom=0x0000
  Range Load Reg Size=0x58
  Request Load Reg Size=0x0E
  Minislot Size in number of Timebase Ticks is = 4
  Minislot Size in Symbols = 32
  Bandwidth Requests = 0xAC3C
  Piggyback Requests = 0x84
  Invalid BW Requests= 0x22
  Minislots Requested= 0x3EAD8
  Minislots Granted  = 0x3EAD8

  Minislot Size in Bytes = 64

  Map Advance (Dynamic) : 2447 usecs

  UCD Count = 303031

  DES Ctrl Reg#0 = C000C043, Reg#1 = 0
```

Configuring Spectrum Management

Spectrum management is a way to improve performance on upstream signal traffic and to compensate for noise and interference. The spectrum manager monitors the upstream frequencies. If there is too much noise or interference in an upstream channel, the spectrum manager reassigns the upstream channel to a different upstream frequency. Spectrum management is configured and activated using spectrum groups. A spectrum group is a table of frequencies that upstream ports can use to implement a frequency-hopping policy. The commands to configure spectrum management are as follows:

```
uBR7246(config)#cable spectrum-groupgroup-number [timeday hh:mm:ss]frequency

  upstream-frequency [dBmV]

uBR7246(config)#interface cableslot/port

uBR7246(config-if)#cable upstreamportspectrum-groupgroup-number
```

InExample 4-10, three fixed frequencies—29 MHz, 33 MHz, and 39 MHz—are configured under spectrum group 20. Spectrum group 20 is then assigned to upstream port 0.

Example 4-10. Configuring a Spectrum Group and Associating It with the Upstream Port

```
uBR7246(config)#cable spectrum-group 20 frequency 29000000

uBR7246(config)#cable spectrum-group 20 frequency 33000000
```

```
uBR7246(config)#cable spectrum-group 20 frequency 39000000
uBR7246(config)#interface cable 3/0
uBR7246(config-if)#cable upstream 0 spectrum-group 20
```

You can use the show cable spectrum-group command to display the current allocation table and frequency assignment, as shown in Example 4-11.

Example 4-11. Displaying the Spectrum Group Configuration

```
uBR7246#show cable spectrum-group
Group  Frequency      Upstream     Weekly Scheduled            Power   Shared
No.    Band           Port         Availability                Level   Spectrum
       (Mhz)                       From Time:    To Time:      (dBmV)
1      29.000                                                  0       No
10     29.000                                                  0       No
20     29.000                                                  0       No
20     33.000                                                  0       No
20     39.000                                                  0       No
20     38.992 [3.20]  Cable3/0 U0                              0
20     29.008 [1.60]  Cable3/0 U1                              0
20     29.008 [1.60]  Cable3/0 U2                              0
20     29.008 [1.60]  Cable3/0 U3                              0
20     29.008 [1.60]  Cable3/0 U4                              0
20     29.008 [1.60]  Cable3/0 U5                              0
```

Example 4-12 displays the current frequency assignment and spectrum group 20 for upstream port 0.

Example 4-12. Verifying the Upstream Frequency and Its Associated

Spectrum Group Configuration

```
uBR7246#show controller cable 3/0 upstream 0
 Cable3/0 Upstream 0 is up
  Frequency 38.992 MHz, Channel Width 3.200 MHz, 16-QAM Symbol
  Rate 2.560 Msps
  Spectrum Group 20
  SNR 28.6280 dB
  Nominal Input Power Level 0 dBmV, Tx Timing Offset 2744
  Ranging Backoff automatic (Start 0, End 3)
  Ranging Insertion Interval automatic (60 ms)
  Tx Backoff Start 0, Tx Backoff End 4
  Modulation Profile Group 5
  Concatenation is enabled
  part_id=0x3137, rev_id=0x03, rev2_id=0xFF
  nb_agc_thr=0x0000, nb_agc_nom=0x0000
  Range Load Reg Size=0x58
  Request Load Reg Size=0x0E
  Minislot Size in number of Timebase Ticks is = 4
  Minislot Size in Symbols = 32
  Bandwidth Requests = 0xAC3C
  Piggyback Requests = 0x84
  Invalid BW Requests= 0x22
  Minislots Requested= 0x3EAD8
  Minislots Granted  = 0x3EAD8
  Minislot Size in Bytes = 64
  Map Advance (Dynamic) : 2447 usecs
  UCD Count = 303031
  DES Ctrl Reg#0 = C000C043, Reg#1 = 0
```

NOTE

The Cisco uBR MC16S Spectrum Management card and Cisco IOS Release 12.1(7)CX together provide advanced spectrum management features such as intelligent frequency hopping, dynamic upstream modulation, and proactive channel management.

Configuring the Downstream Cable Interface

If the external up-converter is used, the downstream frequency is an information-only command. It should reflect the digital carrier frequency, which is the center frequency of the downstream RF carrier for that downstream port. The configuration controlling the digital carrier frequency is done in the IF-to-RF up-converter that must be installed in the downstream path from the Cisco uBR7246. The commands to configure the downstream frequency are as follows:

```
uBR7246(config)#interface cableslot/port

uBR7246(config-if)#cable downstream frequency54000000-1000000000 Broadcast

  Frequency - Hz
```

NOTE

Thecable downstream frequency command currently has no effect on external up-converters; it is information only.

Annex B is the MPEG framing format used in North America. By default, the downstream carrier MPEG frame format is set to Annex B. Under normal circumstances, this setting does not have to be changed. The commands to configure the framing format are as follows:

```
uBR7246(config)#interface cableslot/port

uBR7246(config-if)#cable downstream annex B
```

Interleaving is used to improve the bit error rate. Larger interleaving values increase noise stability but at the cost of potentially increased transmission time. DOCSIS specifies that the operator can select the interleave depth for best operational throughput. The default value is 32. Optional values are 8, 16, 32, 64, and 128. Under normal circumstances, this setting does not have to be changed. The commands to configure the downstream interleave depth are as follows:

```
uBR7246(config)#interface cableslot/port

uBR7246(config-if)#cable downstream interleave-depth [8 | 16 | 32 | 64 | 128]
```

The default modulation mode for downstream is 64-QAM. You can use the following commands to set the downstream modulation to either 64-QAM or 256-QAM:

```
uBR7246(config)#interface cableslot/port

uBR7246(config-if)#cable downstream modulation [64qam | 256qam]
```

To summarize what you have learned, Example 4-13 shows the basic CMTS downstream configuration. Use the command shown in Example 4-14 to verify the downstream configuration.

Example 4-13. Cisco CMTS Downstream Configuration

```
uBR7246(config)#interface cable 3/0
uBR7246(config-if)#cable downstream annex B
uBR7246(config-if)#cable downstream modulation 64qam
uBR7246(config-if)#cable downstream interleave-depth 32
uBR7246(config-if)#cable downstream frequency 459000000
```

Example 4-14. Displaying the Downstream Characteristics

```
uBR7246#show controller cable 3/0 downstream
 Cable3/0 Downstream is up
  Frequency 459.0000 MHz, Channel Width 6 MHz, 64-QAM, Symbol
  Rate 5.056941 Msps
  FEC ITU-T J.83 Annex B, R/S Interleave I=32, J=4
  Downstream channel ID: 0
```

Scenario 4-3: Cisco Cable Modem Bridging and Routing Configuration

You saw the basic Cisco CMTS configuration in the previous scenarios. Now it is time to learn the configuration of the Cisco cable access router. This scenario presents two types of configuration:

- DOCSIS-compliant bridging
- Routing

DOCSIS-compliant bridging is also known as *plug-and-play bridging*. It is a default configuration for most Cisco cable access routers, such as uBR924 and uBR925. In bridging mode, a Cisco cable access router performs as a DOCSIS 1.0 cable modem and should work with any DOCSIS-qualified CMTS. If you don't intend to implement any advanced data features, such as IPSec or a firewall, bridging mode is easier to configure. You need to configure routing mode for a Cisco cable access router if advanced data features are required. This chapter explains how you can configure basic routing mode for a Cisco cable access router. If you need to configure NAT or IPSec,Chapters 12, "Scaling IP Addressing with Network Address Translation," and 14, "Securing Remote Access Networks," provide you with more information on how to configure these features.

Cisco Cable Access Router DOCSIS-Compliant Bridging Configuration

As mentioned earlier, this is the default mode of operation for a Cisco cable access router. The cable access router functions in its plug-and-play DOCSIS-compliant bridging mode and performs as a DOCSIS-compliant two-way cable modem with this configuration. A Cisco DOCSIS-compliant cable modem supports the following minimum set of features:

- It downloads the DOCSIS configuration file from the CMTS or dedicated server at the headend. It provisions and configures itself automatically.
- It operates in bridge mode and provides Internet connectivity to the CPE devices.

NOTE

The DOCSIS specification requires that a DOCSIS-compliant cable modem download a DOCSIS configuration file during its power-on or reset sequence. Cisco provides a DOCSIS cable modem configuration tool at www.cisco.com/support/toolkit/CableModem. You need a CCO account to access this tool.

If the Cisco cable access router is configured in routing mode, the following steps are necessary to convert it back to bridging mode:

Step 1. Disable IP routing on the cable access router:

```
uBR925(config)#no ip routing
```

Step 2. Remove the IP address from both the Ethernet and cable interfaces:

```
uBR925(config)#interface Ethernet0

uBR925(config-if)#no ip address

uBR925(config-if)#interface cable-modem0

uBR925(config-if)#no ip address
```

Assign both the Ethernet and cable interfaces to a bridge spanning group. For the bridge-group number, you can choose any integer from 1 to 63. Also disable spanning tree on the cable interface.

```
uBR925(config)#interface Ethernet0

uBR925(config-if)#bridge-groupbridge-group

uBR925(config-if)#bridge-groupbridge-groupspanning-disabled

uBR925(config-if)#interface cable-modem0

uBR925(config-if)#bridge-groupbridge-group

uBR925(config-if)#bridge-groupbridge-groupspanning-disabled
```

Step 3. Enable DOCSIS-compliant bridging:

```
uBR925(config)#interface cable-modem0

uBR925(config-if)#cable modem compliant bridge
```

cable modem compliant bridge is the default configuration under interface cable-modem0. Therefore, it doesn't show up in the configuration when you enter that command.

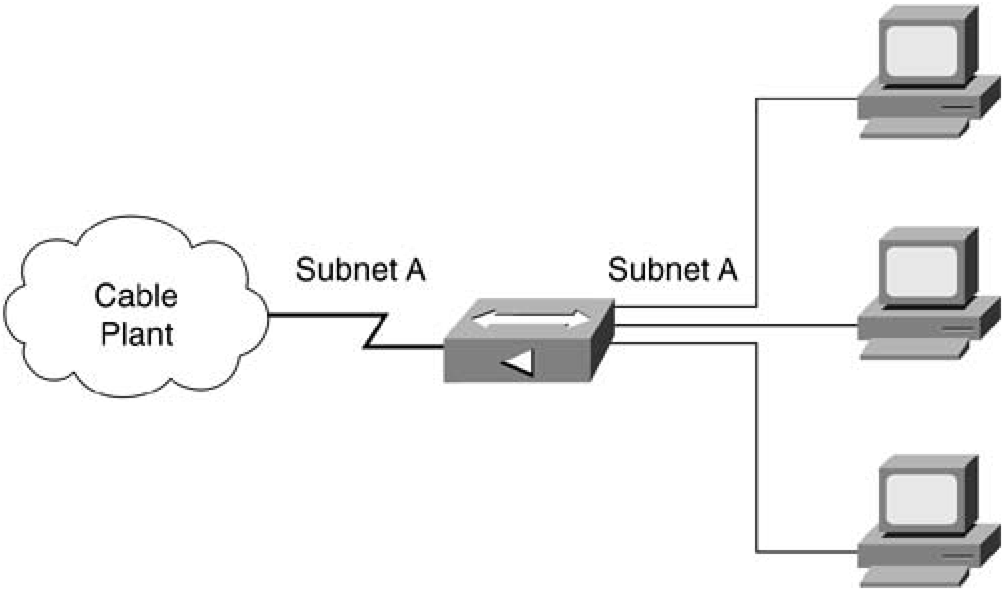
NOTE

With Cisco IOS Release 12.0(5)T and later, a Cisco cable access router such as uBR925 supports a maximum of 254 CPE devices in bridging mode. No limit exists when the cable access router is operating in routing mode.

The MAX CPE field in the DOCSIS configuration file is used to determine the maximum number of CPE devices that can connect the cable network. The default value is 1.

Figure 4-4 illustrates a typical bridging topology, and Example 4-15 displays the basic plug-and-play bridging configuration. Cisco cable access routers do not need additional configuration to provide Internet access for PCs and other customer premises equipment (CPE) devices. However, the PCs and CPE devices must be configured to support DHCP allocation of IP addresses.

Figure 4-4. Typical Cisco Cable Modem DOCSIS-Compliant Bridging Topology



Example 4-15. Cisco Cable Access Router DOCSIS-Compliant Bridging Configuration

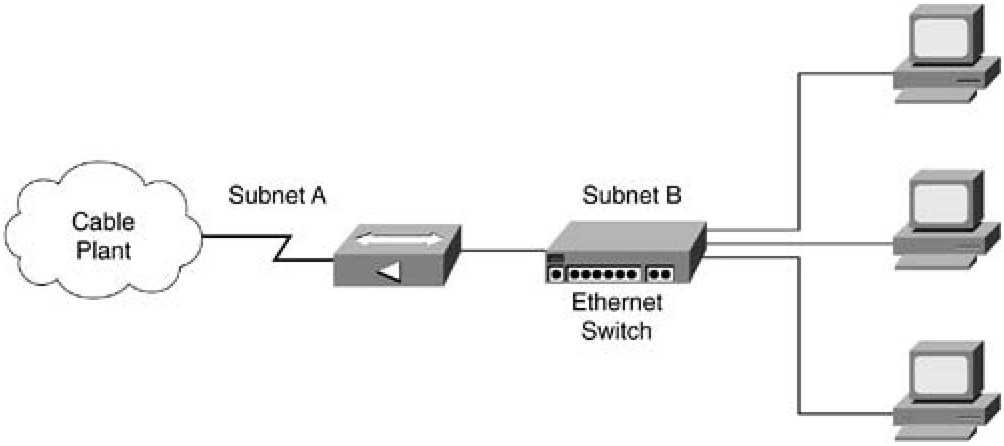
```
no ip routing


interface Ethernet0
```

```
no ip address

bridge-group 59

bridge-group 59 spanning-disabled

!

interface cable-modem0

ip address docsis

bridge-group 59

bridge-group 59 spanning-disabled
```

Cisco Cable Access Router Routing Configuration

Figure 4-5 shows a typical routing topology. If you plan to use advanced features such as IPSec and a firewall, a Cisco cable access router needs to be configured for routing mode. All the CPE devices need to be on a different subnet than the subnet used by the CMTS. For routing protocols, you can configure RIP version 2 or just use the default route.

Figure 4-5. Typical Cisco Cable Modem Routing Topology



Keep in mind that the default configuration is bridging mode. To configure routing mode, follow these steps:

Step 1. Enable IP routing:

```
uBR925(config)#ip routing
```

Step 2. Disable DOCSIS-compliant bridging on the cable interface:

```
uBR925(config)#interface cable-modem0
uBR925(config-if)#no cable-modem compliant bridge
```

Step 3. Remove the bridge group on the cable and Ethernet interfaces:

```
uBR925(config)#interface Ethernet0
uBR925(config-if)#no bridge groupnumber
uBR925(config-if)#interface cable-modem0
uBR925(config-if)#no bridge groupnumber
```

Step 4. Configure the cable interface to receive an IP address from the DHCP server:

```
uBR925(config)#interface cable-modem0
```

```
uBR925(config-if)#ip address docsis
```

Step 5. Enter the Ethernet interface's IP address and subnet mask:

```
uBR925(config)#interface Ethernet0

uBR925(config-if)#ip addressip-address subnet-mask
```

Step 6. Configure the routing protocol (RIP version 2) or configure the default route:

```
uBR925(config)#router rip

uBR925(config-router)#version 2

uBR925(config-router)#networkcable-network-number

uBR925(config-router)#networkEthernet-network-number
```

or

```
uBR925(config)#ip route 0.0.0.0 0.0.0.0ip-address
```

where*ip-address* is the IP address for the CMTS.

Example 4-16 illustrates the cable modem routing configuration. RIP version 2 is used as the routing protocol. Two network statements exist under RIP configuration:

- 172.16.0.0 is for the Ethernet interface and CPE devices.
- 10.0.0.0 is for the cable modem interface and the CMTS.

In most cases the default route configuration should be enough.

Example 4-16. Cisco Cable Access Router Routing Configuration

```
ip routing

...

interface Ethernet0

ip address 172.16.1.1 255.255.255.0

!

interface cable-modem0

ip address docsis

no cable-modem compliant bridge

...

router rip

version 2

network 10.0.0.0

network 172.16.0.0
```

Basic Cable Modem Troubleshooting Using Cisco IOS Software Commands

This section covers some of the useful commands used to perform troubleshooting in Cisco CMTS and cable modems. It also goes into detail about the cable modem initialization sequences and the corresponding status in CMTS.

Basic CMTS Troubleshooting

The Cisco uBR7200 series universal broadband routers maintain a database of flapping cable modems to assist in locating cable plant problems. It tracks the upstream and downstream performance of all DOCSIS-compliant cable modems on the network. Information such as MAC address, up and down transitions, registration events, missed periodic ranging packets, upstream power adjustments, and the physical interface on the Cisco uBR7200 series is maintained in the flap list. Please note that the flap list doesn't affect throughput and incur additional overhead on the network. Cable modems are automatically added to the flap list when any of the following conditions are detected:

- When the cable modem reregisters more frequently than the user-specified insertion time. The default cable flap list insertion time is set to 180 seconds.
- When intermittent keepalive messages are detected between the CMTS and the CM. The default cable flap list miss threshold is set to 6 seconds.
- When the CM upstream transmit power is adjusted beyond the user-specified power threshold. The default cable flap list power adjust is set to 2 dB.

Example 4-17 displays the sample output of the show cable flap-list command. Table 4-5 explains the flap list fields.

Example 4-17. Cable Flap List Output

```
uBR7246#show cable flap-list

MAC Address     Upstream     Ins   Hit   Miss  CRC   P-Adj Flap  Time

0020.4077.7e0c  Cable3/0/U0  147   7080  2499  0     0     293   Aug 10 04:53:30

0020.4077.2be0  Cable3/0/U0  83    7125  1619  0     0     169   Aug 10 04:13:15

0020.4076.d31e  Cable3/0/U0  82    7182  1391  0     0     164   Aug 10 04:07:02

0020.4077.2bfe  Cable3/0/U0  57    7216  977   0     0     116   Aug 10 03:30:11
```

Table 4-5. Explanation of Flap List Fields

Column	Description
MAC Address	The MAC-layer address of a cable modem. It is used to identify the subscribers.
Upstream	The physical upstream interface in the Cisco uBR7200 series. In Example 4-17, the statistic is for a cable modem card in slot 3 and upstream port 0.
Ins	The flapping modem's insertion count. This counts the number of times the RF link was abnormally reestablished. This count can indicate the following: - Intermittent downstream sync loss - DHCP or modem registration problems
Hit	Contains keepalive polling statistics. The link is kept alive using station maintenance intervals. The station maintenance process occurs for every modem about every 25 seconds. When the CMTS receives a response from the cable modem, the event is counted as a hit. Otherwise, the event is counted as a miss. The hit counts should be much greater than the miss counts.
Miss	Contains keepalive polling statistics. High miss counts can indicate the following: - Intermittent upstream because of noise - Laser clipping - Too much or too little upstream attenuation - Common-path distortion
CRC	CRC errors usually indicate noise on a plant. They can indicate the following: - Intermittent upstream because of noise - Laser clipping - Impulsive noise or interference - Common-path distortion
P-Adj	Indicates the number of times the modem power adjustment exceeded the threshold value. This count can indicate the following: - Amplifier degradation - Poor connection - Attenuation problem - Thermal sensitivity
Flap	Indicates the number of times the modem has flapped.

Time	The time stamp indicating the last time the modem flapped.

Theshow cable modem command, shown in Example 4-18, is the most useful command for the CMTS. This command displays information on all cable modems or a particular cable modem on the network.

Example 4-18. Format and Sample Output for the show cable modem Command

```
uBR7246#show cable modem
Interface   Prim Online    Timing Rec    QoS CPE IP address      MAC address
            Sid  State     Offset Power
Cable3/0/U0 1    online    2808    0.00  5   0   30.30.30.9      0007.0e02.c9ed
Cable3/0/U0 2    online    2811    0.00  5   0   30.30.30.10     0003.e3a6.84a1
Cable3/0/U0 3    online    2811    0.00  5   0   30.30.30.7      0007.0e02.cae1
Cable3/0/U0 4    online    2809    0.00  5   0   30.30.30.8      0002.b94a.22a7
```

Table 4-6 lists messages on the CMTS. It shows the different stages of cable modem registration. On the CM side, you can use show controllers cable-modem 0 mac state and look at the MAC state field. debug cable-modem mac log is another useful command on the CM side to capture the CM startup sequences.

Table 4-6. Various CM Statuses When Entering show cable-modem

Message	Description
offline	The modem is considered offline
init(r1)	The modem sent initial ranging
init(r2)	The modem is ranging
init(rc)	Ranging is complete
init(d)	A DHCP request was received
init(i)	A DHCP reply was received, and an IP address was assigned
init(t)	A ToD request was received
init(o)	A TFTP request was received
online	The modem was registered and enabled for data
online(d)	The modem was registered, but network access for the CM is disabled
online(pk)	The modem was registered, BPI was enabled, and KEK was assigned
online(pt)	The modem was registered, BPI was enabled, and TEK was assigned
reject(m)	The cable modem attempted to register, but registration was refused because of a bad Message Integrity Check (MIC)
reject(c)	The cable modem attempted to register, but registration was refused because of a bad class of service (CoS)
reject(pk)	The KEK modem key assignment was rejected
reject(pt)	The TEK modem key assignment was rejected

Theping docsis command, shown in Example 4-19, allows you to quickly diagnose the health of a channel between the Cisco uBR7200 series routers and the cable interface. This command allows you to ping the MAC addresses of the CMs before the registration is complete. In other words, you can use this command to ping CMs that do not have an IP address. The syntax of ping docsis is as follows:

```
uBR7246#ping docsis {IP Address | MAC Address}
```

Example 4-19. ping docsis Command

```
uBR7246#ping docsis 0007.0e02.c9ed
```

```
Queueing 5 MAC-layer station maintenance intervals, timeout is 25 msec:

!!!!!

Success rate is 100 percent (5/5)
```

Basic CM Troubleshooting

Message CRC failures, header CRC failures, sync losses, and pulse losses indicate downstream noise, as shown in Example 4-20. Rerequest is useful information for upstream debugging. You can use show interface cable-modem 0 counters to display this information.

Example 4-20. Displaying Cable Modem-Specific Counters

```
cable-modem#show interface cable-modem 0 counters


Cable specific counters:

Ranging requests sent  : 28176

Downstream FIFO full   : 0

Re-requests            : 0

DS MAC Message Overruns: 0

DS Data Overruns       : 0

Received MAPs          : 83089626

Received Syncs         : 17589748

Message CRC failures   : 0

Header CRC failures    : 0

Data PDUs              : 318963

DS MAC messages        : 100936055

Valid Headers          : 101237079

Sync losses            : 0

Pulse losses           : 0
```

```
BW request failures    : 0
Max TX Rate (pps)      : 0
Max RX Rate (pps)      : 0
ACK table collisions   : 0
ACKs defered           : 0
ACKs dropped           : 0
Packets Concatenated   : 0
Paks not Concatenated  : 17957
Multiple Concatenations: 0
```

Example 4-21 demonstrates another useful command—show controllers cable-modem. Notice that the signal-to-noise ratio (SNR) must be greater than the threshold for the CM to operate properly. In Example 4-21, the SNR is 32.8 dB.

Example 4-21. Cable Modem's MAC State and SNR Information

```
cable-modem#show controllers cable-modem 0

BCM Cable interface 0:
CM unit 0, idb 0x80BF7B08, ds 0x80BF9800, regaddr = 0x2700000, reset_mask 0x80
station address 0003.e3a6.84a1  default station address 0003.e3a6.84a1
PLD VERSION: 1
Concatenation: ON Max bytes Q0: 1600 Q1: 2000 Q2: 2000 Q3: 2000
MAC State is maintenance_state, Prev States = 15
MAC mcfilter 01E02F00  data mcfilter 00000000

MAC extended header ON
DS: BCM 3300 Receiver: Chip id = BCM3300
US: BCM 3300 Transmitter: Chip id = 3300
```

```
Tuner: status=0x00
Rx: tuner_freq 459000000, symbol_rate 5056000, local_freq 11520000
    snr_estimate 328(TenthdB), ber_estimate 0, lock_threshold 26000
    QAM in lock, FEC in lock, qam_mode QAM_64   (Annex B)
Tx: tx_freq 29008112, symbol rate 8 (1280000 sym/sec)
     power_level: 9.0 dBmV (commanded)
                  58  (gain in US AMP units)
                  60  (BCM3300 attenuation in .4 dB units)
```

Theshow controllers cable-modem 0 mac state command, shown in Example 4-22, summarizes the state of the cable MAC layer. The normal operational state of the interface is maintenance_ state. This command gives you most of the information you need about the cable modem's status.

Example 4-22. Partial Output of the show controllers cable-modem 0 mac state Command

```
cable-modem#show controllers cable-modem 0 mac state

MAC State:                  maintenance_state
Ranging SID:                352
Registered:                 TRUE
Privacy Established:        FALSE
...
```

Cable Modem Initialization Process

This section discusses the events during the registration process of a DOCSIS cable modem. You can view these events on the Cisco CM console port by entering the debug cable-modem mac log command. You can view the corresponding events on the CMTS by using the show cable modem command. Refer to Table 4-6 for explanations of each event that occurs on the CMTS.

Examples 4-23 through 4-32 display each event of the CM initialization process.

Event 1: Scanning for a Downstream Channel and Establishing Synchronization with the CMTS

InExample 4-23, the cable modem acquires a downstream channel from the CMTS, saves the last operational frequency in nonvolatile memory, and tries to reacquire the saved downstream channel the next time a request is made.

Example 4-23. CM Begins a Downstream Scan

```
CMAC_LOG_STATE_CHANGE                       ds_channel_scanning_state
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      99/805790200/997799800/6000300
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      98/601780000/799789900/6000300
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      97/403770100/595779700/6000300
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      96/73753600/115755700/6000300
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      95/217760800/397769800/6000300
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      94/121756000/169758400/6000300
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      93/175758700/211760500/6000300
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      92/79753900/85754200/6000300
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      91/55752700/67753300/6000300
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      90/177000000/213000000/6000000
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      89/219000000/225000000/6000000
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      88/141000000/171000000/6000000
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      87/135012500/135012500/6000000
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      86/123012500/129012500/6000000
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      85/405000000/447000000/6000000
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      84/339012500/399012500/6000000
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      83/333025000/333025000/6000000
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      82/231012500/327012500/6000000
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      81/111025000/117025000/6000000
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      80/93000000/105000000/6000000
CMAC_LOG_WILL_SEARCH_DS_FREQUENCY_BAND      79/453000000/855000000/6000000
CMAC_LOG_WILL_SEARCH_SAVED_DS_FREQUENCY     459000000
CMAC_LOG_UCD_MSG_RCVD                       1
CMAC_LOG_DS_64QAM_LOCK_ACQUIRED             459000000
CMAC_LOG_DS_CHANNEL_SCAN_COMPLETED
```

Event 2: Obtaining Upstream Channel Parameters

InExample 4-24, the cable modem waits for an upstream channel descriptor (UCD) message from the CMTS. This is done to retrieve transmission parameters for the upstream channel.

Example 4-24. CM Begins Identifying the Upstream Parameters

```
CMAC_LOG_STATE_CHANGE                       wait_ucd_state
CMAC_LOG_UCD_MSG_RCVD                       1
CMAC_LOG_UCD_MSG_RCVD                       1
CMAC_LOG_ALL_UCDS_FOUND
CMAC_LOG_STATE_CHANGE                       wait_map_state
CMAC_LOG_FOUND_US_CHANNEL                   1
CMAC_LOG_UCD_MSG_RCVD                       1
CMAC_LOG_UCD_NEW_US_FREQUENCY               38992000
CMAC_LOG_SLOT_SIZE_CHANGED                  8
CMAC_LOG_UCD_UPDATED
CMAC_LOG_ADJUST_RANGING_OFFSET              -74
CMAC_LOG_RANGING_OFFSET_SET_TO              12276
CMAC_LOG_MAP_MSG_RCVD
```

Event 3: Starting Ranging for Power Adjustments

The ranging process adjusts the cable modem's transmit power. In Example 4-25, the cable modem performs ranging in stages, ranging state 1 and ranging state 2.

Example 4-25. CM Enters the Ranging 1 and Ranging 2 States

```
CMAC_LOG_INITIAL_RANGING_MINISLOTS             40
CMAC_LOG_STATE_CHANGE                          ranging_1_state  <--- init(r1)
CMAC_LOG_RANGING_OFFSET_SET_TO                 9610
CMAC_LOG_POWER_LEVEL_IS                        28.0 dBmV (commanded)
CMAC_LOG_STARTING_RANGING
CMAC_LOG_RANGING_BACKOFF_SET                   0
CMAC_LOG_RNG_REQ_QUEUED                        0
CMAC_LOG_RNG_REQ_TRANSMITTED
CMAC_LOG_RNG_RSP_MSG_RCVD
CMAC_LOG_RNG_RSP_SID_ASSIGNED                  2
CMAC_LOG_ADJUST_RANGING_OFFSET                 2408
CMAC_LOG_RANGING_OFFSET_SET_TO                 12018
CMAC_LOG_ADJUST_TX_POWER 20
CMAC_LOG_POWER_LEVEL_IS                        33.0 dBmV (commanded)
CMAC_LOG_STATE_CHANGE                          ranging_2_state    <--- init(r2)
CMAC_LOG_RNG_REQ_QUEUED                        2
CMAC_LOG_RNG_REQ_TRANSMITTED
CMAC_LOG_RNG_RSP_MSG_RCVD
CMAC_LOG_ADJUST_RANGING_OFFSET                 -64
CMAC_LOG_RANGING_OFFSET_SET_TO                 11954
CMAC_LOG_RANGING_CONTINUE
CMAC_LOG_RNG_REQ_TRANSMITTED
CMAC_LOG_RNG_RSP_MSG_RCVD
```

```
CMAC_LOG_ADJUST_TX_POWER                       -9
CMAC_LOG_POWER_LEVEL_IS                        31.0 dBmV (commanded)
CMAC_LOG_RANGING_CONTINUE
CMAC_LOG_RNG_REQ_TRANSMITTED
CMAC_LOG_RNG_RSP_MSG_RCVD
CMAC_LOG_RANGING_SUCCESS                       <--- init(rc)
```

Event 4: Establishing IP Connectivity

InExample 4-26, the cable modem invokes DHCP requests to obtain an IP address, which is needed for IP connectivity. The DHCP request also includes the name of a file that contains additional configuration parameters, the TFTP server's address, and the ToD server's address.

Example 4-26. CM Enters the DHCP State

```
CMAC_LOG_STATE_CHANGE                          dhcp_state  <--- init(d)
CMAC_LOG_DHCP_ASSIGNED_IP_ADDRESS              188.188.1.62
CMAC_LOG_DHCP_TFTP_SERVER_ADDRESS              4.0.0.1
CMAC_LOG_DHCP_TOD_SERVER_ADDRESS               4.0.0.32
CMAC_LOG_DHCP_SET_GATEWAY_ADDRESS
CMAC_LOG_DHCP_TZ_OFFSET                        360
CMAC_LOG_DHCP_CONFIG_FILE_NAME                 platinum.cm
CMAC_LOG_DHCP_ERROR_ACQUIRING_SEC_SVR_ADDR
CMAC_LOG_DHCP_COMPLETE                         <--- init(i)
```

Event 5: Establishing the Time of Day

InExample 4-27, the Cisco cable modem accesses the ToD server for the current date and time, which are used to create time stamps for logged events such as those displayed in the MAC log file.

Example 4-27. CM Enters the Time of Day State

```
CMAC_LOG_STATE_CHANGE                  establish_tod_state  <--- init(t)

CMAC_LOG_TOD_REQUEST_SENT

CMAC_LOG_TOD_REPLY_RECEIVED            3234813212

CMAC_LOG_TOD_COMPLETE
```

Event 6: Establishing Security

Keys for privacy are exchanged between the cable modem and the CMTS. In Example 4-28, the CM enters bypass security state. It is not defined for DOCSIS 1.0, but it will be fully defined by DOCSIS 1.1.

Example 4-28. CM Enters the Bypass Security State

```
CMAC_LOG_STATE_CHANGE                        security_association_state

CMAC_LOG_SECURITY_BYPASSED
```

Event 7: Establishing the TFTP

After the DHCP and security operations are successful, in Example 4-29, the cable modem downloads operational parameters from a configuration file stored on a cable company's TFTP server.

Example 4-29. CM Enters the TFTP State

```
CMAC_LOG_STATE_CHANGE                          configuration_file_state

CMAC_LOG_LOADING_CONFIG_FILE                   platinum.cm

CMAC_LOG_CONFIG_FILE_PROCESS_COMPLETE          <--- init(o)
```

Event 8: Performing Registration

InExample 4-30, the cable modem registers with the CMTS. The cable modem is authorized to forward traffic to the cable network after the cable modem is initialized, authenticated, and configured.

Example 4-30. CM Enters the Registration State

```
CMAC_LOG_STATE_CHANGE                       registration_state
CMAC_LOG_REG_REQ_MSG_QUEUED
CMAC_LOG_RNG_REQ_TRANSMITTED
CMAC_LOG_RNG_RSP_MSG_RCVD
CMAC_LOG_REG_REQ_TRANSMITTED
CMAC_LOG_REG_RSP_MSG_RCVD
CMAC_LOG_COS_ASSIGNED_SID                   1/3
CMAC_LOG_RNG_REQ_QUEUED                     3
CMAC_LOG_REGISTRATION_OK
```

Event 9: Establishing Baseline Privacy

Link-level encryption keys are exchanged between the CMTS and the cable modem. In Example 4-31, baseline privacy has not been configured.

Example 4-31. CM Begins the Baseline Privacy Process

```
CMAC_LOG_STATE_CHANGE                       establish_privacy_state
CMAC_LOG_PRIVACY_NOT_CONFIGURED
```

Event 10: Entering the Operational Maintenance State

As soon as the cable modem is completely up and running, it enters operational maintenance state, as shown in Example 4-32.

Example 4-32. CM Enters the Operational State

```
CMAC_LOG_STATE_CHANGE                     maintenance_state <--- online
```

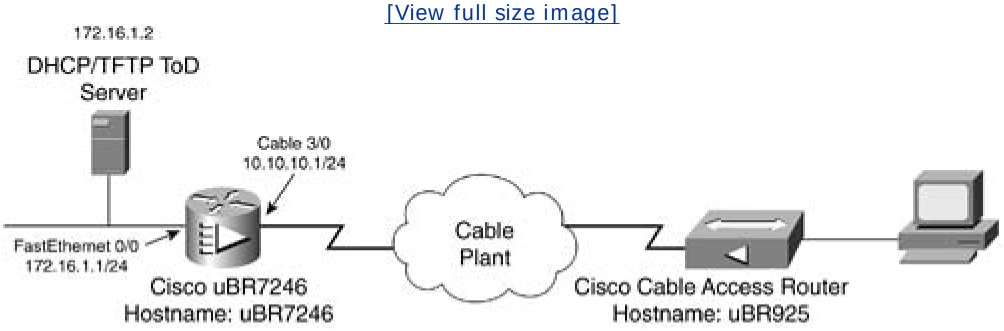
Practical Exercise: The CMTS and DOCSIS-Compliant Bridging Cable Modem Configuration

The practical exercise is designed to test your knowledge of the topics covered in this chapter. The practical exercise begins by giving you some information about a situation and then asks you to work through the solution on your own. The solution is found at the end.

Figure 4-6 presents the network topology for this Practical Exercise.

Figure 4-6. Practical Exercise: Cable Modem Lab Topology

[View full size image]



Before you begin this exercise, be sure the DHCP, ToD, and TFTP servers are properly configured and that the DOCSIS configuration file is available on the TFTP server for Cisco cable access router to download.

In this exercise, you need to configure the CMTS - Cisco uBR7246 shown in Figure 4-6 as follows:

- Activate upstream port 0 of the cable modem card in slot 3.
- Configure spectrum management using group number 30. Configure three upstream frequencies—29 MHz, 33 MHz, and 39 MHz—and then associate this group with upstream port 0.
- Set the power level of upstream port 0 to 0.
- Configure the cable helper address using IP address 172.16.1.2.
- Change the upstream channel bandwidth to 3200000.
- Change the downstream modulation mode to 256-QAM.

If everything is configured correctly in this exercise, the PC in the figure will be able to obtain an IP address from the DHCP server and will have network connectivity.

Practical Exercise Solution

Example 4-33 displays uBR7246's working configuration.

Example 4-33. uBR7246 Configuration

```
uBR7246#show running-config
version 12.1
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
service internal
service udp-small-servers max-servers no-limit
service tcp-small-servers max-servers no-limit
!
hostname uBR7246
!
no logging buffered
enable password cisco
!
cable flap-list size 1000
cable flap-list aging 86400
cable spectrum-group 30 frequency 29000000
cable spectrum-group 30 frequency 33000000
cable spectrum-group 30 frequency 39000000
clock timezone PST -8
clock summer-time PDT recurring
```

```
ip subnet-zero

ip cef

!

interface FastEthernet0/0

 ip address 172.16.1.1 255.255.255.0

 full-duplex

!

interface Cable3/0

 description Connected to Wavecomm Upconverter at 459000000

 ip address 10.10.10.1 255.255.255.0

cable downstream annex B

cable downstream modulation 256qam

cable downstream interleave-depth 32

cable downstream frequency 459000000

cable upstream 0 spectrum-group 30

cable upstream 0 power-level 0

cable upstream 0 channel-width 3200000

no cable upstream 0 shutdown

 cable upstream 1 shutdown

 cable upstream 2 shutdown

 cable upstream 3 shutdown

 cable upstream 4 shutdown

 cable upstream 5 shutdown

 cable dhcp-giaddr policy

cable helper-address 172.16.1.2

!

ip classless

!
```

```
line con 0
line aux 0
line vty 0 4
!
end
```

Example 4-34 shows typical Cisco IOS configurations for a Cisco cable access router that is operating in plug-and-play DOCSIS-compliant bridging mode.

Example 4-34. uBR925 DOCSIS-Compliant Bridging Configuration

```
uBR925#show running-config
version 12.1
no service pad
service timestamps debug datetime msec localtime
service timestamps log uptime
no service password-encryption
!
hostname uBR925
!
enable password cisco
!
!
!
clock timezone - -8
ip subnet-zero
no ip routing
!
interface Ethernet0
```

```
 no ip address
bridge-group 59
bridge-group 59 spanning-disabled
!
interface cable-modem0
 ip address docsis
 cable-modem boot admin 2
 cable-modem boot oper 5
bridge-group 59
bridge-group 59 spanning-disabled
!
ip classless
!
line con 0
line vty 0 4
!
end
```

Summary

After completing this chapter, you should have a basic understanding of the cable modem technology. Basic Cisco CMTS and cable access router configurations were demonstrated in this chapter so that you could learn the minimum configuration requirements for both. Some of the useful IOS commands described here can help you troubleshoot and diagnose cable modem network problems.

Table 4-7 summarizes the CMTS and cable access router commands used in this chapter.

Table 4-7. Summary of Cisco IOS Software Commands Used in This Chapter

Command	Description
interface cable*slot/port*	Specifies the cable interface and downstream port.
cable upstream*port*frequency*return frequency*	Configures the upstream frequency.
show controller cable*slot/port* upstream*port*	Displays the upstream characteristics.
[no]cable upstream*port*shutdown	Activates the upstream port.
cable helper-address*IP address*	Specifies the IP address of a DHCP server to which UDP broadcast packets will be sent.
cable upstream*port*power-level*dBmV*	Sets the upstream input power level.
show interface cable*slot/port*upstream *port*	Displays upstream information on a cable interface.
cable upstream*port*channel-width [200000 | 400000 | 800000 | 1600000 |3200000]	Specifies the upstream channel width for an upstream port.
cable spectrum-group*group-number* [time*day hh:mm:ss*]frequency *upstream-frequency* [*dBmV*]	Creates and configures a spectrum group.
cable upstream*port*spectrum-group *group-number*	Assigns a spectrum group to a single upstream.
show cable spectrum-group [*group-number*] [*detail*]	Displays information about spectrum groups on a Cisco CMTS.
cable downstream frequency *54000000-1000000000 broadcast frequency Hz*	Specifies the downstream center frequency for the cable interface line card. This command is information only for uBR7246.
cable downstream annex [A | B]	Sets the MPEG framing format for a downstream port.

cable downstream interleave-depth [8 |16 | 32 | 64 | 128]	Sets the downstream interleave depth.
cable downstream modulation [64qam |256qam]	Sets the downstream format for a downstream port.
[no]cable-modem compliant bridge	Enables or disables DOCSIS-compliant bridging.
show cable flap-list	Displays the cable flap list on a Cisco CMTS.
show cable modem	Displays information for the registered and unregistered CMs.
ping docsis {*IP address* | *MACaddress*}	Determines whether a specific CM can be reached from the CMTS at the DOCSIS MAC layer.
show interface cable-modem 0 counters	Displays MIB counters on the cable interface.
show controllers cable-modem 0	Displays high-level controller information for the cable access router's cable interface.
show controllers cable-modem 0 mac state	Displays detailed MAC layer information for the cable access router's cable interface.
debug cable-modem mac log	Displays detailed debugging messages for the cable interface MAC layer.

Review Questions

1: What are the downstream and upstream frequency allocations?

2: What type of modulation methods are used for the upstream and downstream?

3: What servers are required for the cable access solution to work?

4: What are the minimum configuration requirements for the CMTS?

5: What MPEG framing format is used in North America?

A. Annex A

B. Annex B

C. Annex C

6: What configuration is recommended to deal with upstream noise and interference?

7: What is the correct syntax to activate upstream port 2 of the cable modem card in slot 4?

A. `interface cable 4/2 upstream no shutdown`

B. `interface cable 4/0 no cable upstream 2 shutdown`

C. `interface cable 2/0 upstream no shutdown`

8: What is the default operating mode of a Cisco cable access router?

9: What are the required steps to configure the routing mode on the cable access router?

10: What command can be used at the CMTS to see the flapping cable modems?

11: What command can be used at the CMTS to find out the registered and unregistered cable modems?

Chapter 5. Configuring Point-to-Point Protocol and Controlling Network Access

This chapter covers the following topics:

- PPP Overview
- Configuring PPP

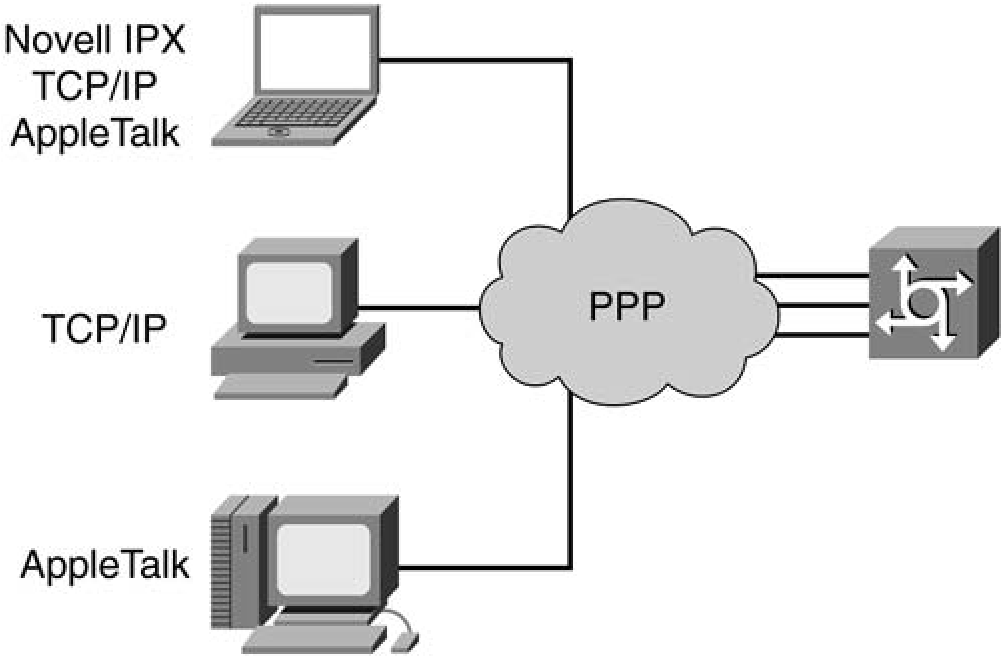
This chapter explores the issues and nature of the Point-to-Point Protocol (PPP) as it relates to remote access. Although PPP is applicable to other networking environments, the focus here is on this side of it in particular.

PPP Overview

To make remote connections possible, users need to have the following components installed on their devices: application software (such as FTP, Telnet, or a web browser), protocol stacks (TCP/IP, IPX, AppleTalk), and link-layer protocols (such as PPP).

When sent out across the dialup connection, the higher-layer protocols are framed in link-layer protocols (such as PPP) much like Ethernet link-layer framing encapsulates IP datagrams on a LAN.Figure 5-1 is a simplified version of a remote connection from an end user to a Network Access Server (NAS) to demonstrate the different types of framing encountered on the way.

Figure 5-1. Framing Types for a Remote Connection



This section introduces the following concepts:

- Common remote-access protocols
- PPP framing
- PPP negotiation phases
- LCP options
- PPP frame format

Common Remote-Access Protocols

For datagram transmission over point-to-point lines, two standard protocols exist: Serial Line Internet Protocol (SLIP) and PPP. SLIP, described in RFC 1055, works only with IP on point-to-point serial connections. PPP, on the other hand, can facilitate multiprotocol connections over synchronous and asynchronous circuits. Therefore, PPP is the most widely used protocol for remote dial access.

PPP Framing

As mentioned, PPP can transmit packets over asynchronous or synchronous links. The packet's framing type is dictated by the medium in use. Asynchronous High-Level Data Link Control (AHDLC) framing is used for asynchronous links, and bit-synchronous framing is used for synchronous links. Cisco supports the following PPP framing types for different interfaces:

- Asynchronous interfaces (modems)— AHDLC framing
- Synchronous interfaces (serial or ISDN)— Bit-synchronous framing
- Virtual terminal (vty) connections via synchronous interface— V.120 framing
- Asynchronous interfaces with special V.110 modems— V.110 framing

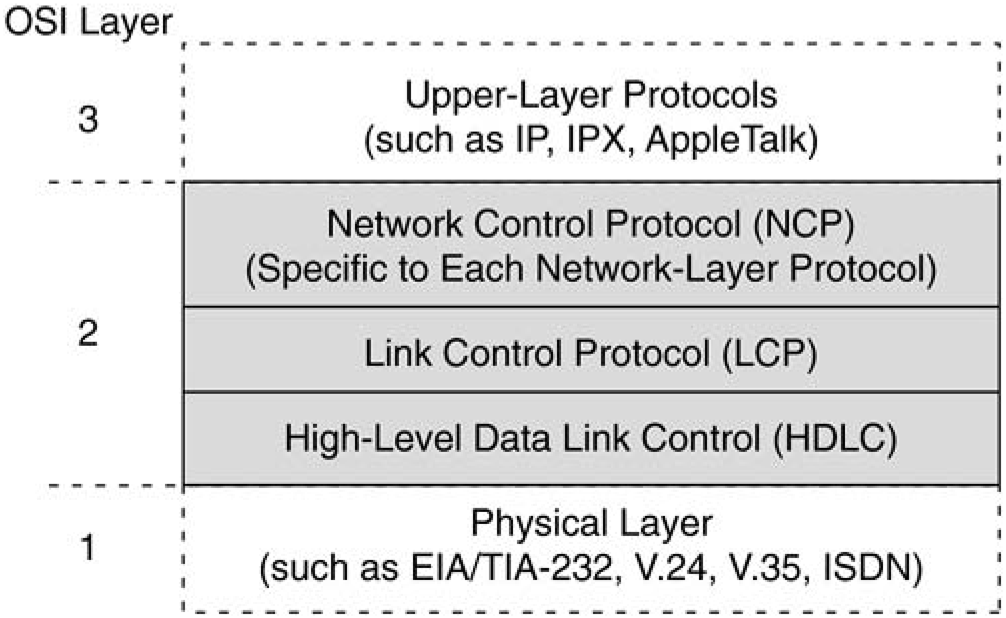
PPP Negotiation Phases

As soon as the encapsulation type described in the preceding section has been confirmed, the link media type is no longer relevant to PPP connection establishment. PPP establishes network protocol connectivity in three functional phases:

- Link Control Protocol (LCP)— Establishes and configures the data-link connection. During this phase, the protocol used in the next phase is negotiated.
- Authentication— Applies security functionality to the connection.
- Network Control Protocol (NCP)— Establishes and configures different network-layer protocols, such as IP, IPX, AppleTalk, DECnet, and bridged data.

Figure 5-2 shows the layered negotiation.

Figure 5-2. PPP Negotiation Phases


OSI Layer
3
Upper-Layer Protocols
(such as IP, IPX, AppleTalk)
Network Control Protocol (NCP)
(Specific to Each Network-Layer Protocol)
2
Link Control Protocol (LCP)
High-Level Data Link Control (HDLC)
1
Physical Layer
(such as EIA/TIA-232, V.24, V.35, ISDN)

The negotiation steps are bidirectional and sequential. In other words, LCP negotiation, including authentication (if configured), must be completed before the NCP negotiation can begin. When a PPP link is operational, it remains in this state until LCP or NCP initiates termination or the physical link fails. When LCP closes the link, all NCP connections associated with the link close as well. Conversely, the NCP-initiated termination is not guaranteed to close the PPP link.

LCP and NCP are discussed in detail in the following sections from a more theoretical perspective. The authentication phase coverage continues in the section "Configuring PPP," along with some authentication-related hands-on tasks.

LCP

LCP deals with options that are link-dependent and protocol-independent. As mentioned, LCP negotiation is bidirectional, which means that both ends of the connection must agree on their options and acknowledge its peer's request. Some of the options negotiated during the LCP phase include magic number (to detect loopback), callback, multilink, link compression, and authentication. Authentication in terms of LCP translates into whether authentication is to be used and, if so, which protocol will facilitate the authentication. However, this is not the actual authentication process.

As soon as the LCP phase has been negotiated successfully, the LCP connection is considered open. Now the authentication process as determined by LCP can begin.

NCP

NCP is the final step of the PPP negotiation process. NCP deals with protocol-dependent options such as the protocol address, protocol compression, and so forth. The individual NCP options correspond to the type of protocol configured on the interface. For instance, if IP is the chosen protocol, IPCP (IP Control Protocol) is negotiated.

The protocol address is the NCP option that is always negotiated. Sometimes it is the only option negotiated. It is possible for the NAS to provide the protocol address to the dial-in client or simply acknowledge whatever protocol address the peer requests. For a remote Cisco router to accept an address from the NAS it has dialed into, the client router needs to be configured to do so. The associated technique is presented in the section "Configuring PPP."

IPCP is the primary NCP and is used here to explain the NCP parameters. As part of the IPCP process, usually three different options are negotiated: the IP address, IP/TCP header compression, and the DNS and WINS primary and secondary servers. Keep in mind that the DNS and WINS options relate to Microsoft Windows PC clients only.

During the IPCP negotiations, the roles of a client and a NAS are different. The access server is required to supply the negotiation parameters (IP address, DNS and WINS address, and so on) for itself and often for a client as well. On the other hand, the client needs to be configured to be able to retrieve this information from the NAS.

Another option in the IPCP negotiations is, as mentioned, the TCP/IP header compression. This might decrease a header's size from 40 to 5 bytes. The negotiation of this option includes whether the peer can accept a packet with the compressed header. This feature is recommended for transmissions whose packet sizes are small, such as Telnet or WWW.

LCP Options

PPP offers a number of features that are negotiated at the LCP level that can prove very useful when implemented in an internetwork:

- Authentication options
- PPP callback
- PPP compression
- Multilink PPP
- Bandwidth Allocation Protocol (BAP)

These services are described next. Authentication is implemented in Scenario 5-3 as well. You will learn how to configure the remaining features in Chapter 6, "Using ISDN and DDR Technologies to Enhance Remote Connectivity."

Authentication

Before you begin learning about the PPP authentication process and techniques, you should become familiar with the following terms:

- Authenticator— The peer demanding authentication. Specifies the authentication protocol to be used during the LCP phase.
- Peer— The opposite end of the link; an entity that is being authenticated by the authenticator.
- Remote authentication— The remote PPP peer authentication of the local NAS.
- Local authentication— The local NAS authenticating its remote peer.

When authentication is requested by either side of the connection during LCP negotiation, the actual authentication takes place after the LCP stage is completed. Authentication is accomplished to check the peer's validity. This is done by verifying the preassigned name (often called the *userid* or *host name*) and the secret (often called the *password*). This book calls them name/secret pairs. The name/secret combination can be stored locally or remotely on an AAA server.

The two most popular PPP authentication techniques are Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP). Both ensure that unauthorized individuals can't access the remote-access server (RAS). Their differences are discussed in greater detail later.

As is the case with all other PPP negotiation phases, authentication is bidirectional. This means that both ends of the connection are required to authenticate one another. Consequently, authentication needs to be enabled at both ends. A notable exception to this rule is discussed in Chapter 6.

The Cisco NAS differentiates among types of call direction. Depending on the type, the NAS takes certain action when it comes to authentication. This is done to protect the network against security violations. Table 5-1 lists the types of calls and the NAS's subsequent responses.

Table 5-1. Call Direction Types

Direction	Description	NASReaction
Callin	Occurs when the NAS is on the receiving end of the call.	The NAS requires the peer to successfully complete local authentication before replying to any requests for remote authentication. This is designed to avoid playback attacks.
Callout	Occurs when the Cisco NAS places the call.	The NAS responds to the remote authentication request without first expecting the completion of local authentication.
Dedicated	Occurs when the Cisco NAS does not recognize to which direction the call belongs.	The NAS responds to the remote authentication request without first expecting the completion of local authentication.

The following subsections describe PAP and CHAP in more detail.

PAP

PAP is the less-secure of the two PPP authentication protocols, because the secret is sent over the wire in clear text. Therefore, if the packet is captured, the secret contained in it can be used in a malicious attack. Understandably, because of this drawback, PAP isn't a preferred method of authentication—unless, of course, it is the only one supported.

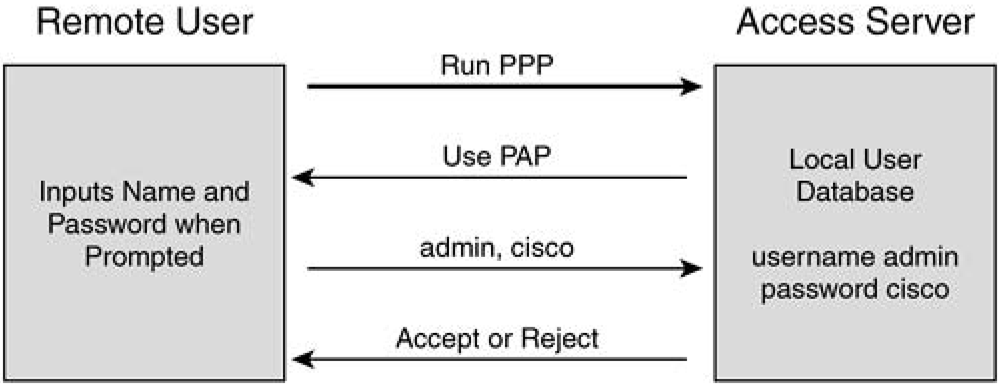
PAP implements a two-way handshake sequence to verify its peer's identity:

Step 1. The peer sends its host name and secret to be checked by the authenticator.

Step 2. The authenticator verifies the offered host name/secret combination against the known value either locally or via an AAA server. If the authenticator determines that the values are legitimate, the authentication is satisfied and acknowledged. If not, the connection is terminated on the spot.

Figure 5-3 shows the handshake process.

Figure 5-3. PAP Authentication



NOTE

The secret in Step 1 does not need to be identical for both peers. They each can have their own.

CHAP

CHAP is quite a bit more secure than PAP. During CHAP authentication, the secret itself is never sent across the connection, some parts of communication are encrypted, and the challenges are constantly repeated to ensure that the connection is authorized at all times. Unlike PAP, CHAP uses a three-way handshake for identification purposes:

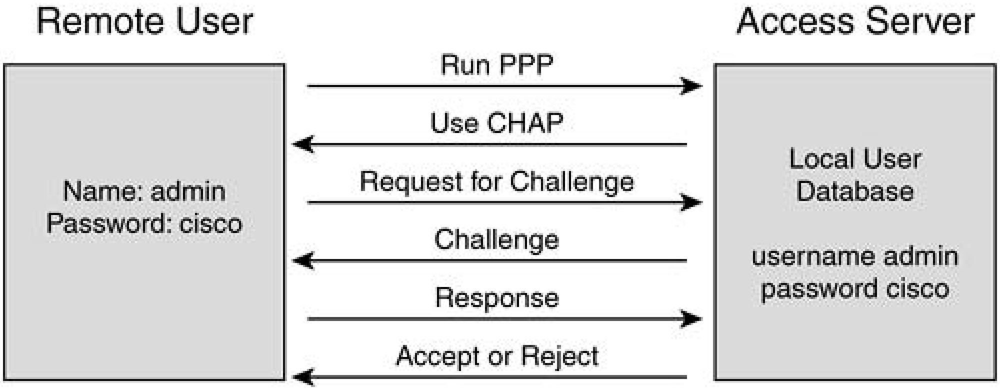
Step 1. The authenticator sends a challenge to the peer. The challenge contains a random number and the authenticator's host name.

Step 2. The peer answers the challenge with a one-way hash value and its own host name. The hash value is calculated via MD5 encryption and is derived from the random number from the challenge message plus the secret associated with the authenticator's host name.

Step 3. When the authenticator receives the response to its challenge, it goes through the same hashing process as the peer, inputting the secret and the random number as the derivatives. After the new MD5 value is calculated, the challenger compares it to the one that came back from the peer. If they match, the authentication is accepted and acknowledged. Otherwise, the connection is dropped.

Figure 5-4 shows the three-way handshake process.

Figure 5-4. CHAP Authentication



NOTE

Because the hash values need to be identical for CHAP authentication to work, the secret value must be shared between both peers. This requirement is different form the PAP implementation.

PPP Callback

PPPcallback is an option negotiated during LCP that allows a caller to request that a called party should place another callback to the initiating peer. For this discussion, the party requesting a callback is the client, and the party accepting the request and making the callback is the server. PPP callback is useful whenever centralized control over a call is desired, such as for the purposes of bill consolidation, dialup call savings, and even security, because the callbacks are placed only to preconfigured numbers.

Although normally authentication is considered an optional PPP feature, it must be enabled and passed for the callback feature to work.

The sequence of a PPP callback is as follows:

Step 1. The callback client places a call to the callback server (NAS) indicating that the callback service is requested. The callback server responds with the callback request acknowledgment. The type of acknowledgment sent in this step signifies simply that the server is generally capable of accepting callback requests.

Step 2. The callback server proceeds further by authenticating the client. As usual, the authentication can take place locally or at an AAA server.

Step 3. As soon as the client has been successfully identified, the server verifies whether the callback service is allowed for the particular client that requested it. If so, the call initially placed by the client is disconnected.

Step 4. After the call is disconnected, the server waits a certain amount of time. Then it initiates a new callback to the client on a preconfigured number. If this call fails, additional attempts are not undertaken.

PPP is negotiated upon the client-initiated call only. The callback does not require a new PPP negotiation.

NOTE

If the server decides that the client is not authorized for a callback service, the response depends on whether dial-on-demand routing is implemented for the connection. If DDR is used, the callback server continues processing the initial call as if there were no callback request to begin with. If you want to disconnect a user who failed callback authorization, you can issue an optional command on the server. If the connection is non-DDR, the callback server disconnects the initial call by default.

PPP Compression

Compression can significantly improve throughput on slow links. Cisco IOS offers PPP compression for all upper-layer protocols through Compression Control Protocol (CCP). This type of compression is considered a per-interface compression.

PPP CCP is an optional feature and is negotiated after the LCP phase. Cisco supports two CCP compression algorithms:

- STAC— Checks the data stream for redundant strings and replaces them with tokens that are smaller. Then it creates tables of tokens with information about where the original type occurs within the data stream. These tables are used to replace redundant strings found in the subsequent data streams. This process uses more CPU but less memory.
- Predictor— Checks the data for previous compression. The already-compressed data is sent as is. This process requires more memory but fewer CPU cycles.

Both of these algorithms base their operation on "dictionaries" of past data compression. When dictionaries become full, information is renewed. The choice of an algorithm depends on each individual case.

Compression should be used with care, because it can be a burden on system resources. Keep in mind that the rate of compression is dependent on the data type. For instance, text files are very good candidates for compression versus already-compressed file formats that would not yield a better than 1:1 compression ratio. Also, whenever possible, hardware compression should be chosen over software compression.

Although PPP compression can be bidirectional, it is recommended that only the remote client side perform compression. This way, the NAS can decompress the client's communication but doesn't compress its own. The reason for this is so that the NAS itself avoids performing compression that can use four times as much CPU power as decompression.

Multilink PPP

MultilinkPPP(MPPP) is a technique of fragmenting packets and sending them over multiple data links to the PPP peer for reassembly. The benefit of MPPP lies in its ability to temporarily use additional bandwidth that's available between the two peers. MPPP is identified by an additional 4-byte header that dictates the fragment sequencing.

MPPP can be used in the following scenarios:

- In circuit-switched topologies for ISDN B channels or asynchronous connections— Although MPPP was not designed exclusively for ISDN networks, it can certainly be successfully employed in such an environment by dynamically combining multiple B channels into a single larger-sized link to achieve *N* * 64 kbps bandwidth. The most usual of the *N* values is 2 because it is cost-effective and widely available. Combining two B channels would yield a total bandwidth of 128 kbps. The concepts of ISDN and its implementation of MPPP are discussed in further detail in Chapter 6.
- Leased line— All group members are synchronous serial lines.
- Dialed or leased lines— Separate links can be of either origin.
- Different bandwidth of individual members— The maximum fragment size is computed based on the slowest of all grouped links.
- Combination of applications producing different-sized datagrams— Intermixing of datagrams without a multilink header.

MPPP Terms

Before you can understand the ins and outs of MPPP, you should become familiar with the following terms:

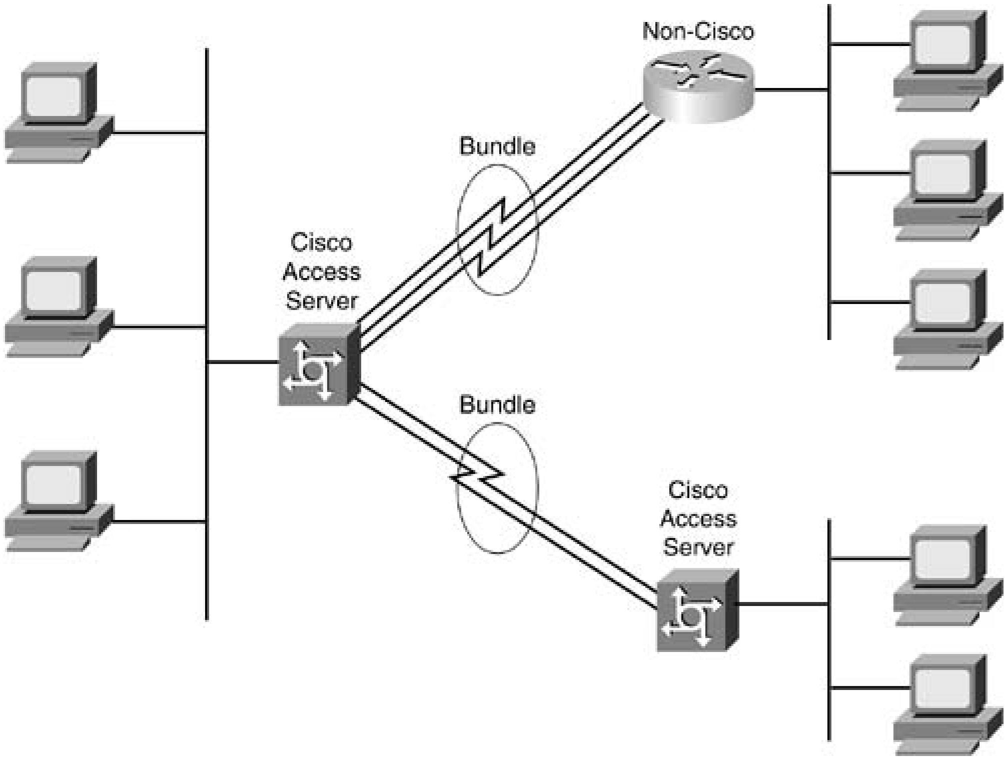
- Bundle— A group of links between two PPP peers combined for MPPP operation.
- Bundle master— An interface in control of a bundle.
- Bundle member— An interface that is a part of a bundle.
- Dialer interface— A rotary group for multiple interfaces such as ISDN BRI/PRI.
- Nondialer interface— A serial interface.
- Virtual-access interface— A temporary logical interface created for the purpose of an MPPP call. Its configuration is cloned from the dialer interface that placed or received the MPPP call.
- Virtual template— Used for MPPP calls over nondialer interfaces to provide configuration information.
- Max-Receive-Reconstructed Unit (MRRU)— An LCP option that indicates whether the LCP packet sender supports MPPP and the link's maximum byte limit.

- Endpoint discriminator— An LCP option that specifies whether an MPPP bundle exists for the sending device.

MPPP Operation

Every MPPP bundle needs to be controlled by a single interface, the bundle master, which is a virtual-access interface (see Figure 5-5).

Figure 5-5. Multilink PPP Bundling



The multilink PPP process starts out with LCP negotiation, including the MRRU option that takes place on the physical interface. PPP LCP negotiation determines whether MPPP can be used on the link. The bundle is identified by a peer's name, its endpoint discriminator, or both. Therefore, the PPP authentication is required to complete so that the peers can identify each other, name the bundle, and check whether another bundle of the same name already exists. If a bundle already exists, the new call simply joins in. No new negotiations of any sort are required for additional calls.

At this point, the NAS sets up a virtual-access interface as the bundle master. From this moment, all PPP negotiations are transferred from the physical interface to the virtual-access interface. The physical interface becomes a part of a bundle governed by a bundle master. Whatever NCP parameters are negotiated for the master are automatically applied to the rest of the bundle members.

MPPP Operation Issues

Three major issues are associated with MPPP's operation:

- A new link in a bundle is brought up and added to the bundle whenever the bundle master's saturation reaches the specified load. This value is represented as a percentage of 255, where 255 is the maximum.
- As mentioned, a new bundle can be created when no other bundle is between the same two peers already in existence. A single bundle can handle multiple connections between the same pair of devices. So the rules are simple: If there is no bundle, one can be built; if there is a bundle, the new call joins it. A bundle's existence is checked by using an expected name. The default order in which the bundles are named is first by the PPP authenticated name and then by the endpoint discriminator if no authentication has been negotiated.
- The links are dropped from a bundle when the bundle master's load falls below the configured threshold for a predetermined amount of time (the idle timer). The link that was added to the bundle last is the first one to be disconnected. With links of unequal bandwidth, the slowest link is dropped first.

Bandwidth Allocation Protocol

The specification for BAP is an extension of the MPPP concept. It was created to control the number of connections that an authorized user is allowed to establish at any time. BAP creates a standard set of rules that let MPPP change bandwidth on demand without the need for end-user participation in the configuration changes. As a result, the NAS can manage the usage of its access ports per caller.

BAP administers the method in which individual links are added to and deleted from an MPPP bundle. While LCP is negotiated, BAP is decided on, and a distinguishing link discriminator is given to every link in an MPPP bundle. It allows peers to specify which link is brought up or disconnected when the bandwidth increase or decrease is requested.

BAP can operate in two different modes: active and passive. *Active mode* means that the device can initiate or accept any type of connection request and determine whether links should be added to or removed from a multilink bundle. Active mode is for dialer interfaces, but not for virtual-template interfaces. *Passive mode* means that the device only responds to calls by accepting a call request, a callback request, or an addition or removal of a link by an active peer. Passive mode can be used for virtual-template interfaces and dialer interfaces.

BAP supports ISDN and asynchronous serial interfaces. When talking about BAP operation over dialer interfaces, only legacy dial-on-demand routing (DDR) dialer configurations are discussed. BAP does not support DDR dialer profiles (covered in detail in later chapters).

BAP Operation

The first member link of the MPPP bundle is not negotiated under BAP. The subsequent member links, however, require BAP management. Although the first link does not belong to BAP, it does carry all BAP information packets. There are a total of eight BAP packet types:

- Call-Request

- Call-Response
- Callback-Request
- Callback-Response
- Link-Drop-Query-Request
- Link-Drop-Query-Response
- Call-Status-Indication
- Call-Status-Response

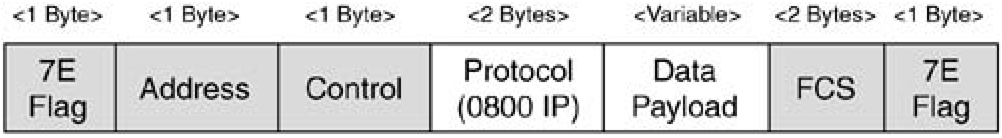
BAP follows Cisco's MPPP implementation in its judgment of load by monitoring the bundle master. The bundle load determines the need for bandwidth aggregation. Only with BAP, both peers have to agree on the bandwidth aggregation decision.

PPP Frame Format

Figure 5-6 illustrates the contents of a PPP frame. Its fields are as follows:

- The Flag, Address, and Control field values are constant.
- The Protocol field reveals the protocol payload (such as TCP/IP or IPX).
- The Data field may be of variable length according to the maximum transmission unit (MTU) of the PPP interface.
- FCS is the frame check sequence.

Figure 5-6. PPP Frame Format



<1 Byte>	<1 Byte>	<1 Byte>	<2 Bytes>	<Variable>	<2 Bytes>	<1 Byte>
7E Flag	Address	Control	Protocol (0800 IP)	Data Payload	FCS	7E Flag

Configuring PPP

This section looks at configuring PPP. First, you will enable PPP for asynchronous ports using modems, followed by verification and troubleshooting. Then, the "Scenarios" section discusses how to implement some of PPP's more-advanced features, including authentication, PPP compression, and PPP multilink.

NOTE

PPP over ISDN is covered in more detail in Chapter 6.

Initial PPP Configuration

This section offers a brief list of the steps necessary to configure PPP for asynchronous interfaces. These steps are described in greater detail in the "Scenarios" section. They are as follows:

Step 1. Attach and configure the modem. Then configure the router's asynchronous port. This step is not covered here because you already learned it in Chapter 3, "Modem Connections and Operation Overview." Review it if you feel the need.

Step 2. Configure PPP's asynchronous interface, including PPP encapsulation and authentication methods.

Step 3. Configure network layer addresses, and enable routing.

Step 4. Configure the asynchronous interface for dial-on-demand routing.

Verification and Troubleshooting

For troubleshooting purposes, it is important that you know and understand the different packet types available with LCP. This will assist you in reading the debug output. Table 5-2 describes the packet types you are most likely to encounter.

Table 5-2. LCP Packet Types

Packet Type	Debug Output	Description
Configure-Request	CONFREQ	Notifies the receiving peer of the local configuration parameters.
Configure-Ack	CONFACK	Acknowledges the receipt of CONFREQ packet. Sends back all the options specified in the CONFREQ received from the peer.
Configure-Nak	CONFNAK	Responds to the unacceptable CONFREQ options with a negative acknowledgment. Includes the unacceptable options, marked as such, along with any values that are OK. Upon receiving the CONFNAK packet, the initiating peer has the option of sending a new CONFREQ with the changed NAKed values or to have those values omitted altogether.
Configure-Reject	CONFREJ	Refuses an option included in the received CONFREQ. Upon receipt of the CONFREJ packet, the transmitting peer needs to retransmit the CONFREQ, this time without the rejected options.
Terminate-Request	TERMREQ	Requests an existing connection termination.
Terminate-Ack	TERMACK	Acknowledges the receipt of a TERMREQ packet.
Echo-Request	Echo-Request	Confirms connectivity and detects loopback.
Echo-Reply	Echo-Reply	Replies to an Echo-Request packet.

Table 5-3 presents several options that can be included in the CONFREQ packet.

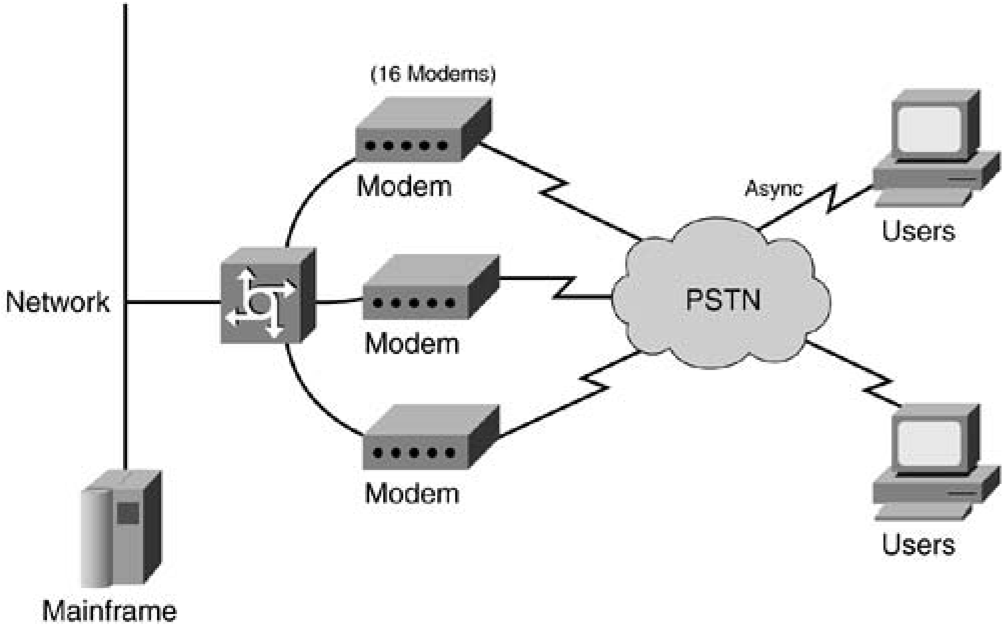
Table 5-3. CONFREQ Packet Options

Packet Option	Description
Maximum-Receive-Unit	Identifies the maximum receive unit (MRU) to the peer. Note that Cisco IOS ignores the peer's request to increase the MRU above the maximum of the interface MTU. Also, if the peer suggests an MRU that is less than the Cisco IOS interface MTU, the CONFREQ is CONFNAKed, indicating the Cisco IOS interface MTU. It is not recommended for a dialup peer to dynamically adjust the interface MTU to match the negotiated MRU/MRRU. Ideally, the MRU/MTU settings should be identical on both peers.
Multilink-MRRU	Indicates that the local device supports MPPP.
Authentication-Protocol	Advertises the desired authentication protocol.
Magic-Number	Identifies a random number to detect a loopback.
Compression	Announces PPP compression.
Callback	Requests a callback from the peer.

Scenarios

As you have recently learned, PPP is widely used by many telecommuters to access their private corporate networks remotely. In this section, you will configure a remote-access setup. Although each scenario completes its own task, together the scenarios form one logical implementation. Every scenario builds on the previous one. They are based on the topology shown in Figure 5-7.

Figure 5-7. PPP Scenarios Topology



The scenarios show you how to

- Configure the PPP communications protocol for operation
- Control network access with PAP authentication
- Configure PPP compression

Scenario 5-1: Initial Access Server and Network Setup

Before turning to the PPP-specific setup, you need to perform some initial configuration on the access server. The configuration in this section is basic and should be familiar to you.

Configure the username admin password cisco combination for an administrator:

```
R1(config)#usernamenamepasswordstring
```

Example 5-1 shows the running configuration on R1.

Example 5-1. show running-config Command Output

```
version 12.2
service timestamps debug datetime localtime
service timestamps log datetime localtime
no service password-encryption
!
username admin password cisco
hostname R1
!
line vty 0 4
 password cisco
```

Next, you use the absolute line number to configure an asynchronous link. The absolute line number changes with different router models, so you should verify it. To figure out to which line number the modem is attached, issue the show line command. The line number displayed in the output is the one that needs configuration. In Example 5-2, you can see that the line number corresponds to TTY port 8.

Example 5-2. show line Command Output

```
R6#show line
   Tty Typ     Tx/Rx    A Modem  Roty AccO AccI   Uses   Noise  Overruns   Int
*    0 CTY              -    -      -    -    -      0       0     0/0       -
*    8 AUX  57600/57600 F inout     -    -    -      0       0     0/0       -
```

After finding the correct line number, you must configure the modem on that line. Because you don't know the modem type, you should use the autoconfigure type default command.

As you can see from Example 5-3, the modem line configuration includes the modem inout and modem autoconfigure commands. If you remember, the default on Cisco routers is to reject the incoming network connections to asynchronous ports. You are also required to specify an incoming transport protocol or, in this case, use the transport input all command to indicate that any type of protocol is allowed.

Example 5-3. Configuring a Line for the Modem

```
R1(config)#line aux 8

R1(config-line)#modem InOut

R1(config-line)#modem autoconfigure discovery

R1(config-line)#login local

R1(config-line)#transport input all

R1(config-line)#flowcontrol hardware
```

NOTE

Example 5-3 demonstrates the configuration of line 8 for a modem. You should be familiar with all these commands from reading Chapter 3. Refer to that chapter to refresh your memory.

Scenario 5-2: Configuring PPP on the Asynchronous Link

Now that you've enabled the basic configuration on the access server, you can move on to the PPP-related tasks.

Enabling the Autosensing Feature

The first of these tasks is to prepare line 8 for PPP use. You enabled the modem functions on the line in the previous scenario. Now it's time to allow a PPP session to start on the router.

A Cisco access server can be configured to accommodate (autosense) a PPP or SLIP session to start automatically or through the user prompt. If autosensing is not configured on your access server, the router does not recognize a connection attempt and does not respond to the client.

The following command is placed on the absolute line number (line 8 in this case), along with the rest of the modem commands:

```
R1(config-line)#autoselect [arap | ppp | slip | during-login]
```

By selecting one of this command's options, you allow the router to start a corresponding process when it receives a starting character. Each of the three protocols as well as the carriage return has a recognizable start character contained in a frame's flag. For instance, when a return character is encountered, the access server knows to start an EXEC session.

Table 5-4 shows the frame flag values in hexadecimal format for the protocols available for autosensing.

Table 5-4. PPP Flag Values

Protocol	Flag Value
Return key	0d
ARAP	10
PPP	7E
SLIP	c0

When configuring the autoselect command, you need to specify which of the three protocols is allowed to start a session. In this scenario, it is obviously PPP. Also, use the autoselect command with the during-login keyword to cause the username/password prompt to come up without the user's having to press Enter. Example 5-4 displays the autoselect commands used for this scenario.

Example 5-4. Enabling Autosensing

```
R1(config)#line 8
```

```
R1(config-line)#autoselect during-login

R1(config-line)#autoselect ppp
```

Configuring the Asynchronous Interface

The next task is to configure the router's asynchronous interface and enable PPP on it. The asynchronous interface in question should match the modem line number. Enter interface configuration mode by issuing the int async65 command. After you are in interface configuration mode, you can proceed with the PPP-specific and general statements.

Enabling PPP Encapsulation

To enable PPP on any type of connection, whether synchronous or asynchronous, you need to define PPP encapsulation at the interface level of both ends of the connection by entering the following command:

```
R1(config-if)#encapsulation ppp
```

Configuring Local Interface Addressing

The next step involves configuring the network layer address on the local asynchronous interface (8, in this case) in the following manner:

```
R1(config-if)#ip addressaddress mask
```

Configuring Interface Addressing of Remote Devices

At this point you need to create a method for assigning an IP address to the PPP client dialing into the router. The IP address for a particular peer can be managed on the NAS in a number of ways:

- Static configuration of IP addresses for each interface.
- A local pool of IP addresses can be configured on the NAS. In such instances, the IP address is allocated by the pool.
- The IP address can be assigned by a Dynamic Host Configuration Protocol (DHCP) server.
- The IP address can be assigned by an AAA server.
- The peer can request a specific IP address, in which case the NAS would only need to acknowledge the request.

As mentioned earlier, you can set up your configuration so that the peer's IP address is assigned centrally. The following command is used to specify a client's source originating address. When a client dials into the appropriate line, the address is allocated from the specified location.

```
R1(config-if)#peer default ip address [ip-address | dhcp | poolpoolname]
```

You can see that the available options include a specific IP address, a local pool of addresses, or a DHCP server. If you choose to specify the pool argument, you need to configure a global address pool that matches the name of the peer default ip address command. Here's the command syntax to configure the local pool:

```
R1(config)#ip local poolpool-name starting-address ending-address
```

If you decide to go with the dhcp option, you need to configure the ip helper address and ip dhcp-server as well.

To specify dynamic addressing (addressing requested by the user at the EXEC level upon connection), issue the following command:

```
R1(config-if)#async dynamic address
```

You may opt to include both default and dynamic options with your configuration. This way, the user will have a choice between the two methods of address assignment. If the user enters the peer's own address, it is used; if the user enters the default keyword, the default address is used instead.

In this scenario, the peer default ip address*ip-address* option is used. The peer is assigned the address of 10.1.1.254. Example 5-5 demonstrates the interface-level commands used in this scenario.

Example 5-5. Enabling PPP at the Interface Level

```
R1(config)#int async65

R1(config-if)#encapsulation ppp

R1(config-if)#ip address 10.1.1.1 255.255.255.0

R1(config-if)#peer default ip address 10.1.1.254
```

Verification

Make sure that the previous steps have resulted in the proper configuration. Look at the interface configuration by issuing the show interface async 65 command, as shown in Example 5-6.

Example 5-6. show interface async 65 Command Output

```
R1#show interfaces async 65

Async65 is up, line protocol is up

  Hardware is Async Serial
  MTU 1500 bytes, BW 57600 Kbit, DLY 100000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set
  DTR is pulsed for 5 seconds on reset
  LCP Open
  Closed: BRIDGECP, IPCP, CCP, CDPCP, LLC2, BACP, IPV6CP
  Last input never, output 00:14:49, output hang never
  Last clearing of "show interface" counters 00:14:59
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/16 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 6 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     1 packets output, 24 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

To verify that the line configuration for the modem has been configured correctly, enter the show line*x* command, as shown in Example 5-7. Note that line 8 is used in this example.

Example 5-7. show line 65 Command Output

```
R1#show line 8
   Tty Typ     Tx/Rx    A Modem  Roty AccO AccI   Uses   Noise  Overruns   Int
*   8  AUX   57600/57600  F inout     -    -    -      0       0     0/0       -
Line  8, Location: "", Type: ""
Length: 24 lines, Width: 80 columns
Baud rate (TX/RX) is 57600/57600, no parity, 2 stopbits, 8 databits
Status: Ready, Active, No Exit Banner, Modem Configuring,
  Modem Speed Locked, Modem Signals Polled, Autoconfig Running
Capabilities: Autobaud Full Range, Hardware Flowcontrol In,
  Hardware Flowcontrol Out, Modem Callout, Modem RI is CD,
  Line usable as async interface, Modem Discovery
Modem state: Ready
Modem hardware state: CTS DSR  DTR RTS
Special Chars: Escape  Hold  Stop  Start  Disconnect  Activation
                ^^x    none   -     -       none
Timeouts:      Idle EXEC    Idle Session   Modem Answer  Session   Dispatch
               00:10:00        never                        none     not set
                            Login-sequence User Response
                             00:00:30
                            Autoselect Initial Wait

Modem type is usr_sportster.
Session limit is not set.
Time since activation: never
Editing is enabled.
History is enabled, history size is 10.
DNS resolution in show commands is enabled
Full user help is disabled
```

```
Allowed input transports are pad v120 telnet rlogin udptn ssh.

Allowed output transports are pad v120 telnet rlogin ssh.

Preferred transport is telnet.

No output characters are padded

No special data dispatching characters
```

Scenario 5-3: Configuring Interface Parameters Available with PPP

A number of interface-level commands can be configured as part of the PPP setup on the NAS. Scenario 5-2 introduced interface parameters that are required with PPP. In this scenario, you will configure optional but highly desirable and widely used PPP services.

Configuring PPP Authentication

Probably the most implemented of all PPP interface parameters is authentication. As you learned earlier in this chapter, PAP and CHAP are the two PPP authentication options. The command to configure authentication is as follows:

```
R1(config-if)#ppp authentication [pap | chap]
```

Theppp authentication command can configure either the PAP or CHAP authentication method. The correct method is indicated by the appropriate keyword.

You may use the ppp authentication, ppp authorization, and ppp accounting commands at an interface level.

Configuring Asynchronous Callback

Cisco router interfaces support PPP asynchronous callback. This configuration assumes that other PPP and modem-related features that were covered in previous scenarios are already enabled.

Configuring Callback PPP Clients

There are two types of clients for which you can enable the callback feature on the NAS:

- Those that support PPP callback per RFC 1570
- Those that do not but instead can put themselves in answer mode, which accepts the router's callback

For clients that are PPP callback-compliant, you can configure the router to accept the clients' callback request with the following command:

```
R1(config-if)#ppp callback accept
```

Use this command to configure the callback feature for clients that are not RFC 1570-compliant:

```
R1(config-if)#ppp callback initiate
```

Because the client can't request callback itself, the router can initiate on behalf of the client.

Configuring IPCP

Think back to the discussion of the NCP IPCP near the beginning of this chapter. You have an option to specifically include several IPCP parameters with your PPP configuration, such as the primary and secondary DNS and WINS server addresses, the peer-requested address, and so on. Here is the general syntax for the ipcp command:

```
R1(config-if)#ppp ipcp [accept-address | dns [reject | accept | primary-ip-address
  [secondary-ip-address] [accept]] | ignore-map | username unique | wins [reject |
  accept | primary-ip-address [secondary-ip-address] [accept]]]
```

If you put a question mark after the ppp ipcp interface command, a list of options appears, as shown in Example 5-8. This Example shows the IPCP available parameters specified on R1.

Example 5-8. Available IPCP Options

```
R1(config-if)#ppp ipcp ?
  accept-address      Accept any non zero IP address from our peer
  address             Additional ipcp address options
  dns                 Specify DNS negotiation options
  header-compression  IPCP header compression option
  ignore-map          Ignore dialer map when negotiating peer IP address
  mask                Specify subnet mask negotiation options
  predictive          Predict peers IPCP requests/replies
  username            Configure how usernames are handled
  wins                Specify WINS negotiation options
```

Table 5-5 explains the options in Example 5-8.

Table 5-5. IPCP Parameters

Option	Description
accept-address	Accepts any nonzero IP address from the peer.
dns [accept | reject]	Domain Name Server. Accepts a peer request for any nonzero server address. Rejects the IPCP option if received from the peer.
ignore-map	Ignores the dialer map when negotiating the peer IP address.
username unique	Ignores a common username when providing an IP address to the peer.
Wins	Windows Internet Naming Service.

Example 5-9 displays the IPCP options configured on R1.

Example 5-9. Configuring IPCP Parameters

```
R1(config)#interface async65

R1(config-if)#ppp ipcp accept-address

R1(config-if)#ppp ipcp header-compression ack

R1(config-if)#ppp ipcp dns 10.1.1.1

R1(config-if)#ppp ipcp wins 10.1.1.12

R1(config-if)#ppp ipcp mask 255.255.255.0

R1(config-if)#ppp ipcp username-unique

R1(config-if)#ppp ipcp ignore-map
```

Scenario 5-4: Configuring the Asynchronous Interface for DDR

It's time to enable DDR tasks on your router. The commands introduced in this scenario are applied to the asynchronous interface.

Dialer Commands

Remember the peer default ip address command you entered earlier? This command allows the router to accept the peer's address:

```
R1(config-if)#dialer in-band
```

Beware of the order in which these commands need to be added to your configuration. The peer default ip address command must come first, followed by the dialer in-band command. If this order is reversed, the peer's IP address will not be accepted.

The following commands allow the definition of interesting traffic to be associated with the interface:

```
R1(config-if)#dialer-grouplist-number
R1(config-if)#dialer idle-timeoutseconds
```

The idle-timeout period specifies how many seconds free of interesting traffic the line tolerates before disconnecting.

Configuring Dedicated or Interactive PPP Sessions

The following command empowers the user to enter PPP commands, such as the IP address, at the EXEC level. If the async dynamic address command you learned earlier is specified, the router must be put into interactive mode. Dedicated mode does not allow user input.

```
R1(config-if)#async mode [dedicated | interactive]
```

Example 5-10 shows all the commands covered in the preceding sections.

Example 5-10. Configuring DDR on the Interface

```
R1(config)#int async65

R1(config-if)#dialer in-band

R1(config-if)#dialer idle-timeout 600

R1(config-if)#dialer-group 8

R1(config-if)#async mode interactive
```

Specifying Interesting Traffic

Now that you've tied the interesting traffic list to the interface, you need to define the interesting traffic parameters under the global configuration. In other words, you need to create a traffic rule that triggers asynchronous calls:

```
R1(config)#dialer-listdialer-groupprotocolprotocol-name [permit | deny | list

  access-list-number | access-group]
```

As you can see, the interesting traffic definition can get quite extensive, especially if you add access lists to the equation. It is not the intention of this chapter to cover DDR in detail. You will have a chance to learn more about it in subsequent chapters. For this scenario, the dialer list can be kept to its bare minimum (IP routing), because the router is not used to initiate calls or route over the link. Example 5-11 shows R1's complete DDR configuration.

Example 5-11. show running-config Command Output

```
interface Async65

no ip address
```

```
encapsulation ppp

no ip route-cache

no ip mroute-cache

dialer in-band

dialer fast-idle 122

dialer string 5551212

dialer hold-queue 100

dialer-group 1

!

dialer-list 1 protocol ip permit
```

Verification

When you complete the preceding tasks, you can test the validity of your configuration. You may check the modem with the debug confmodem and debug modem commands. You can examine the result of issuing these commands in Example 5-12. The AT commands are sent by the router to the modem.

Example 5-12. Output from the debug modem and debug confmodem Commands

```
R1#clear line 65

R1#debug modem

15:25:51: TTY65: DSR came up

15:25:51: tty65: Modem: IDLE->READY

15:25:51: TTY65: Autoselect started

15:27:51: TTY65: Autoselect failed

15:27:51: TTY65: Line reset  <--- Clear line 65

15:27:51: TTY65: Modem: READY->HANGUP

15:27:52: TTY65: dropping DTR, hanging up

15:27:52: tty65: Modem: HANGUP->IDLE
```

```
15:27:57: TTY65: restoring DTR

15:27:58: TTY65: DSR came up


R1#terminal monitor

R1#debug confmodem

Modem Configuration Database debugging is on

*Mar 3 03:06:30.931: TTY1: detection speed (57600) response ---OK---

*Mar 3 03:06:30.963: TTY1: Modem command: --AT&FS0=1--

*Mar 3 03:06:31.483: TTY1: Modem configuration succeeded

*Mar 3 03:06:31.487: TTY1: Detected modem speed 57600

*Mar 3 03:06:31.487: TTY1: Done with modem configuration
```

Now that you know your physical layer is functioning properly, you can verify the upper layers. You can turn on PPP debugging using the following commands:

```
R1#debug ppp negotiation

R1#debug ppp authentication

R1#debug ppp error
```

When initiating a PPP session to the router by a workstation, note the following processes in the debug output for organized troubleshooting:

- PPP initialization when the first PPP string is received
- LCP finishes the PPP negotiation
- Authentication negotiation finishes successfully
- The peer receives the proper IP address

Example 5-13 demonstrates PPP debugging using the debug ppp negotiation, debug ppp authentication, and debug ppp error commands.

Example 5-13. Debugging PPP

```
R1#debug ppp negotiation

*Mar  2 02:25:27.693: %LINK-3-UPDOWN: Interface Async65, changed state to up

*Mar  2 02:25:27.693: Se0/0 PPP: Treating connection as a dedicated line

*Mar  2 02:25:27.693: Se0/0 PPP: Phase is ESTABLISHING, Active Open

*Mar  2 02:25:27.693: Se0/0 LCP: O CONFREQ [Closed] id 11 len 10

*Mar  2 02:25:27.693: Se0/0 LCP:    MagicNumber 0x35C4DB07 (0x050635C4DB07)

*Mar  2 02:25:27.729: Se0/0 LCP: I CONFREQ [REQsent] id 14 len 29

*Mar  2 02:25:27.729: Se0/0 LCP:    MagicNumber 0xBFAE7481 (0x0506BFAE7481)

*Mar  2 02:25:27.729: Se0/0 LCP:    MRRU 1524 (0x110405F4)

*Mar  2 02:25:27.729: Se0/0 LCP:    EndpointDisc 1 Local (0x130F01696F7377616ED3

  23630 3063)

*Mar  2 02:25:27.729: Se0/0 LCP: O CONFREJ [REQsent] id 14 len 23

*Mar  2 02:25:27.729: Se0/0 LCP:    MRRU 1524 (0x110405F4)

*Mar  2 02:25:27.733: Se0/0 LCP:    EndpointDisc 1 Local (0x130F01696F7377616ED3

  23630 3063)

*Mar  2 02:25:27.733: Se0/0 LCP: O CONFACK [REQsent] id 15 len 10

*Mar  2 02:25:27.733: Se0/0 LCP:    MagicNumber 0xBFAE7481 (0x0506BFAE7481)

*Mar  2 02:25:27.733: Se0/0 LCP: State is Open

*Mar  2 02:25:27.733: Se0/0 PPP: Phase is UP

*Mar  2 02:25:27.737: Se0/0 IPCP: O CONFREQ [Closed] id 10 len 16

*Mar  2 02:25:27.737: Se0/0 IPCP:    CompressType VJ 15 slots (0x0206002D0F00)

*Mar  2 02:25:27.737: Se0/0 IPCP:    Address 10.1.30.200 (0x03060A011EC8)

*Mar  2 02:25:27.745: Se0/0 LCP: I CONFACK [Open] id 11 len 10

*Mar  2 02:25:27.745: Se0/0 LCP:    MagicNumber 0x35C4DB07 (0x050635C4DB07)
```

```
*Mar  2 02:25:27.777: Se0/0 LCP: I CONFREQ [Open] id 15 len 10

*Mar  2 02:25:27.781: Se0/0 LCP:    MagicNumber 0xBFAE7481 (0x0506BFAE7481)

*Mar  2 02:25:27.781: Se0/0 LCP: Dropping packet, state is Open

*Mar  2 02:25:27.813: Se0/0 IPCP: I CONFREQ [REQsent] id 105 len 28

*Mar  2 02:25:27.813: Se0/0 IPCP:    CompressType VJ 15 slots (0x0206002D0F00)

*Mar  2 02:25:27.813: Se0/0 IPCP:    Address 0.0.0.0 (0x030600000000)

*Mar  2 02:25:27.813: Se0/0 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)

*Mar  2 02:25:27.813: Se0/0 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)

*Mar  2 02:25:27.813: Se0/0 IPCP: Pool returned 10.1.30.109

*Mar  2 02:25:27.817: Se0/0 IPCP: O CONFREJ [REQsent] id 105 len 16

*Mar  2 02:25:27.817: Se0/0 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)

*Mar  2 02:25:27.817: Se0/0 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)

*Mar  2 02:25:27.817: Se0/0 IPCP: O CONFNAK [REQsent] id 106 len 10

*Mar  2 02:25:27.817: Se0/0 IPCP:    Address 10.1.30.109 (0x03060A011E6D)

*Mar  2 02:25:27.817: Se0/0 IPCP: O CONFACK [REQsent] id 107 len 16

*Mar  2 02:25:27.817: Se0/0 IPCP:    CompressType VJ 15 slots (0x0206002D0F00)

*Mar  2 02:25:27.821: Se0/0 IPCP:    Address 10.1.30.109 (0x03060A011E6D)

*Mar  2 02:25:27.833: Se0/0 IPCP: I CONFACK [ACKsent] id 10 len 16

*Mar  2 02:25:27.833: Se0/0 IPCP:    CompressType VJ 15 slots (0x0206002D0F00)

*Mar  2 02:25:27.833: Se0/0 IPCP:    Address 10.1.30.200 (0x03060A011EC8)

*Mar  2 02:25:27.833: Se0/0 IPCP: State is Open

*Mar  2 02:25:27.837: Se0/0 IPCP: Install route to 10.1.30.109

*Mar  2 02:25:27.837: Se0/0 IPCP: Add link info for cef entry 10.1.30.109

*Mar  2 02:25:27.861: Se0/0 IPCP: I CONFREQ [Open] id 106 len 16

*Mar  2 02:25:27.865: Se0/0 IPCP:    CompressType VJ 15 slots (0x0206002D0F00)

*Mar  2 02:25:27.865: Se0/0 IPCP:    Address 0.0.0.0 (0x030600000000)

*Mar  2 02:25:27.865: Se0/0 IPCP: Dropping packet, state is Open

*Mar  2 02:25:27.881: Se0/0 IPCP: I CONFREQ [Open] id 107 len 16
```

```
*Mar  2 02:25:27.885: Se0/0 IPCP:    CompressType VJ 15 slots (0x0206002D0F00)
*Mar  2 02:25:27.885: Se0/0 IPCP:    Address 10.1.30.109 (0x03060A011E6D)
*Mar  2 02:25:27.885: Se0/0 IPCP: Dropping packet, state is Open
*Mar  2 02:25:28.733: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async65/,
  changed state to up

R1#debug ppp authentication
May 15 22:05:31.868: %LINK-3-UPDOWN: Interface Async65, changed state to up
*May 15 22:05:31.892: %ISDN-6-CONNECT: Interface Async65 is now connected to
  5551212
*May 15 22:05:31.900: ASYNC65 PPP: Treating connection as a callout
*May 15 22:05:31.900: ASYNC65 CHAP: Using alternate hostname cisco
*May 15 22:05:31.984: ASYNC65 CHAP: I CHALLENGE id 50 len 27 from "r8"
*May 15 22:05:31.988: ASYNC65 CHAP: Using alternate hostname cisco
*May 15 22:05:31.992: ASYNC65 CHAP: Username r8 found
*May 15 22:05:31.992: ASYNC65 CHAP: Using default password
*May 15 22:05:31.996: ASYNC65 CHAP: O RESPONSE id 50 len 26 from "cisco"

R1#debug ppp error
PPP Async65(i): rlqr receive failure. successes = 15
PPP: myrcvdiffp = 159 peerxmitdiffp = 41091
PPP: myrcvdiffo = 2183 peerxmitdiffo = 1714439
PPP: threshold = 25
PPP Async65(i): rlqr transmit failure. successes = 15
PPP: myxmitdiffp = 41091 peerrcvdiffp = 159
PPP: myxmitdiffo = 1714439 peerrcvdiffo = 2183
PPP: l->OutLQRs = 1 LastOutLQRs = 1
PPP: threshold = 25
```

```
PPP Async65(i): lqr_protrej() Stop sending LQRs.
PPP Async65(i): The link appears to be looped back.
```

Next, you can test your configuration by following up with pings to appropriate addresses, as shown in Example 5-14.

Example 5-14. ICMP Testing

```
R1#ping 10.10.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.1.2, timeout is 2 seconds:
..!!!
Success rate is 60 percent (3/5), round-trip min/avg/max = 112/114/120 ms
R1#
```

If you want to check the results of all the scenarios in R1 configuration, take a look at Example 5-15, which displays R1's show running-config output.

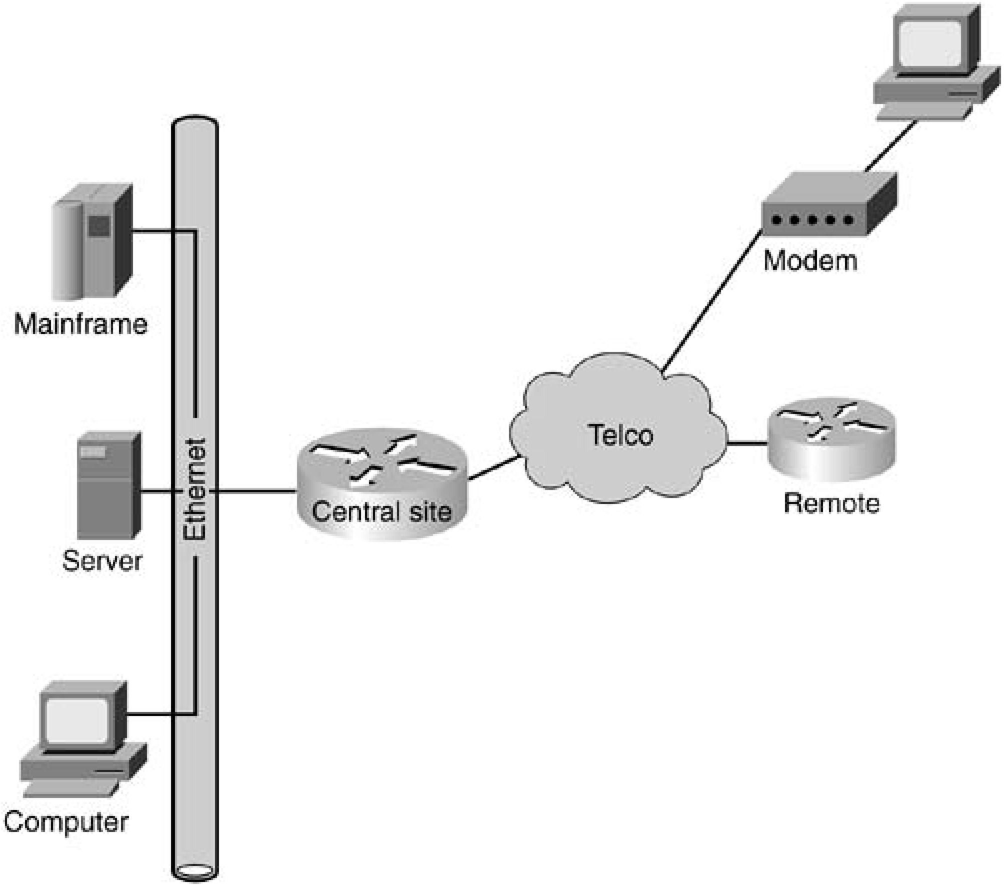
Example 5-15. show running-config Output from R1

```
version 12.2
service timestamps debug datetime localtime
service timestamps log datetime localtime
no service password-encryption
!
hostname R1
!
interface Async65
 no ip address
 encapsulation ppp
 no ip route-cache no ip mroute-cache
 dialer in-band
 dialer fast-idle 122
 dialer string 5551212
 dialer hold-queue 100
 dialer-group 1
 async default routing
 async dynamic address
 async dynamic routing
 async mode interactive
 ppp reliable-link
 ppp encrypt mppe auto
 ppp authentication chap pap ms-chap optional
 ppp direction callin
 ppp link reorders
!
dialer-list 1 protocol ip permit
!
line aux 0
 autobaud
 modem InOut
 modem autoconfigure discovery
 transport input all
 autoselect during-login
 flowcontrol hardware
```

Practical Exercise: Dial In and Dial Out

In this exercise, your goal is to create a working solution to enable router Central Site to receive calls from router Remote as well as a dial-in user. Figure 5-8 illustrates the topology for the exercise.

Figure 5-8. Practical Exercise: Dial In and Dial Out



Practical Exercise Solution

The first step is to build a chat script for the modem, which you learned about in Chapter 3. Second, you apply the configuration to the modem line. Don't forget to verify the line's status with the show line command. Configure PPP and the DDR option on the physical and dialer interfaces that were discussed throughout this chapter's scenarios. Example 5-16 shows the output of the show run command on the Central Site router. You can view all commands necessary to complete the given task.

Example 5-16. show running-config Output

```
chat-script rstmdm "" "AT&FS0=1&B1&C1&D2&H1&K1&M4&R2" OK
chat-script dialnum ABORT ERROR ABORT BUSY "" "ATDT \T" TIMEOUT 60 CONNECT
!
username remote privilege 15 password 7 05080F1C2243
username user1 privilege 15 password 7 05080F1C2243
!
interface Async65
 no ip address
 encapsulation ppp
 no ip route-cache
 dialer in-band
 dialer rotary-group 0
 async default routing
 async dynamic address
 async mode interactive
 ppp reliable-link
 ppp encrypt mppe auto
 ppp authentication chap pap ms-chap optional
 ppp direction callin
 ppp link reorders
```

```
!
interface Dialer0
 no ip address
 encapsulation ppp
 no ip route-cache
 dialer in-band
 dialer map ip 10.1.1.1 name remote 5551212
 dialer-group 1
 no cdp enable
 ppp reliable-link
 ppp encrypt mppe auto
 ppp authentication chap pap ms-chap optional
 ppp direction callin
 ppp ipcp accept-address
 ppp ipcp wins 172.16.5.1
 ppp ipcp mask 255.255.255.0
 ppp link reorders
!
dialer-list 1 protocol ip permit
!
line aux 0
 autobaud
 modem InOut
 modem autoconfigure discovery
 transport input all
 autoselect during-login
 stopbits 1
 flowcontrol hardware
```

Summary

PPP is used to enable multiprotocol transport across a point-to-point link. It allows for link-level services such as authentication, callback, and compression. Multiple PPP links can be combined into a bundle for higher throughput that can be configured to give both peers the proper control over the resources.

PPP is a tremendously important part of remote network connectivity. It is mentioned often throughout this book with various implementations.

Review Questions

1: Which of the following is/are valid PPP authentication methods?

A. PAP

B. CHAP

C. MS-CHAP

D. MS-PAP

2: True or false: The authentication process is part of LCP negotiation.

3: List at least three possible methods for IP address assignment to the client.

4: When you let the client choose his or her own IP address with the async dynamic address command, your router needs to be in ___________.

A. Dedicated mode

B. Interactive mode

C. Either

D. None of the above

5: Which of the following are valid LCP packet types?

A. CONFNAK

B. CONFREJ

C. CONFREQ

D. All of the above

E. A and C

F. None of the above

6: True or false: BAP's active mode can operate under dialer interfaces, but not under virtual-template interfaces.

7: How can you hard-code the subnet mask during the IP PCP negotiation?

8: What are the main types of compression that PPP supports?

A. Compressor

B. Stacker

C. Predictor

D. LZ compression

E. TCP header

9: What command allows the router to accept the peer's address?

10: Name an interface in control of a bundle in MPPP.

Chapter 6. Using ISDN and DDR Technologies to Enhance Remote Connectivity

This chapter covers the following topics:

- ISDN Overview
- DDR
- The ISDN Layer Protocols
- Examining ISDN Call Setup and Teardown
- Configuring ISDN

This chapter provides an overview of Integrated Services Digital Network (ISDN). The first part of this chapter covers a limited amount of theory necessary for sufficient understanding of the ISDN configuration, verification, and troubleshooting of a Cisco Network Access Server (NAS) that follows.

Included in this discussion of ISDN are its advantages over other types of connections, services it can offer, available bandwidth in the form of BRI and PRI, and dial-on-demand routing. When it comes to ISDN, it is also important to comprehend the interface to the service provider cloud as well as circuit-switched access establishment via call setup and its release via call teardown.

ISDN Overview

This section introduces the main components of ISDN. This includes, but is not limited to, the following topics:

- What are integrated services?
- Advantages of ISDN
- ISDN services
- ISDN bandwidth and channels

What Are Integrated Services?

Since the 1960s, the telecommunication networks backbone has been converting to digital. The end-user access, however, such as the telephone and modem connections, has remained mostly analog. ISDN takes advantage of the digital telecommunications backbone and replaces some of the analog service devices with new higher-speed digital equipment. So the beauty of ISDN is that it makes use of the existing backbone technology while enhancing it with cost-effective higher-speed services that were previously unavailable or unjustifiably expensive.

When the digital network is extended end-to-end by ISDN, it eliminates the need to translate (or sample) the analog waveform into a digital pattern. This allows any application, whether voice, video, or data, to transparently transmit over the backbone, because there is no longer a need to differentiate between the various types of network traffic. As a result, diverse sets of services can be integrated into one cost-effective solution.

Advantages of ISDN

ISDN provides a viable alternative to various forms of communication while allowing reliable high-speed access to the Internet and other services. Table 6-1 demonstrates how ISDN compares to a few of these forms of communication.

Table 6-1. Advantages of ISDN

Form of Communication	ISDNAdvantage Over the Specified Form
Analog dialup modem	The transmission rate is up to four times faster.
	Call setup is less than 1 second versus 30 to 45 seconds.
Leased line	The cost is lower.
	The transmission rate is double.

ISDN Services

As mentioned, ISDN can provide a number of different services:

- Data— A widely used ISDN service, referring to the payload type of the ISDN packet. Has an end-to-end synchronous signal.
- Rate adaptation— Allows incompatible equipment to use the ISDN network for data communication. For instance, devices that do not support synchronous connections or 64 kbps speeds nonetheless can use ISDN services. The two rate adaptation standards are as follows:

  -V.110— Can be applied to synchronous and asynchronous applications. It has no error detection or correction. It is based on TDM technology. The frame format lets flags and control bits accommodate different source speeds.

  -V.120— Can be applied to synchronous and asynchronous applications. Unlike V.110, it allows error detection and correction. It is based on STDM (HDLC) technology.

- Voice— Analog or asynchronous data transfer over asynchronous modems.
- DNIS— Identifies a called party number.
- CLID— Identifies a calling party number.

ISDN Bandwidth and Channels

The discussion of ISDN revolves around two variations: BRI and PRI. Before we begin, let's examine the North American digital signal standards and their "T" assignments, because BRI and PRI adhere to those standards. You will also learn the European equivalents of their North American counterparts.

Table 6-2 shows the DS level, its corresponding maximum speed, the "T" designation, and the number of channels for each level.

Table 6-2. North American Digital Hierarchy

Digital Signal Level	Speed	"T" Designation	Channels or DS0s
DS0	64 kbps	—	1
DS1	1.544 Mbps	T1	24
DS2	6.312 Mbps	T2	96
DS3	44.736 Mbps	T3	672
DS4	274.176 Mbps	T4	4032

NOTE

As mentioned, some equipment is incapable of supporting the 64 kbps DS0 standard and can extend to only 56 kbps operation.

ISDN-BRI

BRI specifies the following components:

- It is made up of three DS0s.
- It has two B channels at 64 kbps each, used for data.
- It has one D channel at 16 kbps, used for signaling.
- The remaining 48 kbps is used for framing and synchronization.
- The total speed is measured as follows:

  64 + 64 + 16 + 48 = 192

ISDN-PRI

North American PRI specifies the following components:

- It is made up of DS1 (T1) with 24 channels.
- It has 23 B channels at 64 kbps each, used for data.
- It has one D channel at 64 kbps, used for signaling, carried in timeslot 24.
- The remaining 8 kbps is used for framing and synchronization.
- The total speed is measured as follows:

  (23 * 64) + 64 + 8 = 1544 kbps
- Two encoding schemes are possible: AMI and B8ZS.
- Two separate types of framing are defined: Super-Frame (SF) and Extended Super-Frame (ESF).

European and other countries' PRI specifies the following components:

- It is made up of E1, the equivalent of T1, with 32 channels.
- It has 30 B channels at 64 kbps each, used for data.
- It has one D channel at 64 kbps, used for signaling, carried in timeslot 16.

- The remaining 64 kbps is used for framing and synchronization.
- The total speed is measured as follows:

  (30 * 64) + 64 + 64 = 2048 kbps
- Encoding is HDB3.
- Framing is multiframe.

BRI Functional Groups

BRI defines the following functional groups (ISDN devices):

- TE1— Terminal equipment 1. Specifies an ISDN-compatible device. Can connect to an NT1 or NT2 device (described in this list). Examples of a TE1 device include

  - Router with a native ISDN interface

  - Digital telephone

  - Digital fax
- TE2— Terminal equipment 2. Specifies a device that is not ISDN-compatible. Requires a terminal adapter (described next) for compliance with ISDN. TE2 equipment examples include

  - Router with no native ISDN interface

  - Devices with X.21, X.25, or EIA/TIA-232 interfaces
- TA— Terminal adapter. Used with TE2 to convert electrical signals into the kind recognized by ISDN.
- NT1— Network Termination 1. Links four-wire ISDN customer wiring to the two-wire provider facility.
- NT2— Network Termination 2. Specifies a device that manages traffic to and from subscriber devices and the NT1. Performs switching and concentrating.
- LT— Line Termination. Specifies a provider's side. Functions as an NT1.
- ET— Exchange Termination. Specifies a line card of a subscriber in the ISDN exchange.
- LE— Local Exchange. Specifies LT and ET. It is a provider's ISDN switch.

NOTE

An NT1/NT2 combination device is sometimes called a *Network Termination Unit (NTU)*.

Which Devices Represent the BRI Reference Points

Reference points are interfaces between functional groups. They might or might not manifest in actual physical interfaces. Reference points include the following:

- U— User reference point. Between NT1 and LT.
- T— Terminal reference point. Between NT1 and NT2, or between NT1 and TE1 (or TA) if no NT2 is present.
- S— System reference point. Between NT2 and TE1 (or TA). Has the same characteristics as the T interface.
- R— Rate reference point. Between TA and TE2.

Let's spend a few moments discussing how functional groups and reference points work together.

First, you connect the wall jack to the NT1 with a standard two-wire cable. Then you connect the NT1 to an ISDN terminal or a terminal adapter with a four-wire connector. An eight-wire connector is used for the S/T interface because it requires both NT and TE capabilities.

An S/T interface is a combination of the S and T interfaces. It defines a reference point between a TE1 (or TA) and an NT. You can think of it as a point-to-multipoint bus that multiple ISDN devices can share.

The U interface is a two-wire interface between the NT and the provider cloud normally terminated with an eight-pin RJ-48 connector. In this case, the NAS has built-in NT1 functionality. U interface termination is mostly used in North America.

As far as the Cisco IOS is concerned, there is no real difference between the S/T or U termination when it comes to BRI operation. What you have to keep in mind is that BRI consists of a single D channel for signaling and two B channels for data.

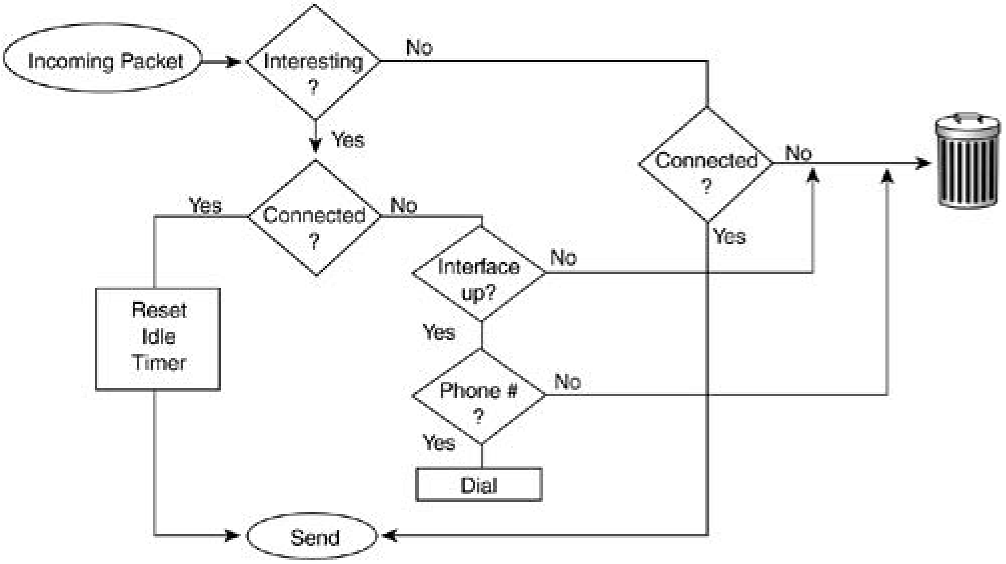
DDR

Dial-on-demand routing (DDR) determines whether to bring up a connection that is not already active based on *interesting* and *uninteresting traffic* coming into the router. Interesting traffic brings up a connection, and uninteresting traffic doesn't.

How does a router know which traffic is interesting and which isn't? Through preconfigured access lists and dialer lists. The section "Configuring ISDN" shows you how to configure interesting traffic.

Figure 6-1 displays the basic process of determining interesting traffic.

Figure 6-1. Interesting Versus Uninteresting Packets



A dialer list specifies interesting traffic that is allowed to make a connection. Numerous dialer list settings can be used in conjunction with access lists that provide more granular control for a dialer list. A dialer list is then assigned to a dial group that refers to it when needed. A physical BRI interface belongs to a dial group and therefore carries out the instructions set up in a dialer list.

It is very important to understand the need for static route entries to prevent routing updates from initiating a call and thus adding unnecessary service charges. DDR can be configured with a number of different options. For instance, an idle timer disconnects a call when no traffic has been transmitted for a predetermined period of time.

It can also be used for other valuable purposes, such as backup for a leased line or Frame Relay connection. In this case, an ISDN link may be brought up after a certain load has been reached on the main line or a preconfigured length of time has lapsed since the line became inactive.

Another DDR concept that is discussed later in this chapter is so-called legacy DDR versus dialer profiles. You can think of legacy DDR as the configuration that applies to the physical interface, unlike dialer profiles, which use logical dialer interfaces to accomplish DDR.

To accomplish DDR configuration, you need to go through the following steps. Each step is discussed further in the section "Configuring ISDN":

- Specify interesting traffic.
- Assign these parameters to an interface.
- Define the destination aspects with legacy DDR or dialer profiles.

The ISDN Layer Protocols

ISDN spans the bottom three layers of the OSI reference model. As mentioned, ISDN uses a multitude of protocols that fall under those layers and govern its operation.

To communicate from the local terminal equipment to the ISDN switch in the central office (CO), ISDN uses a unique collection of protocols. ITU organizes these protocols in the following manner:

- E. series— Describes telephone network standards as they relate to ISDN.
- I. series— Describes theory, terminology, interfaces, and common techniques.
- Q. series— Describes switching and signaling. For instance, Q.921 deals with Link Access Procedure on the D channel (LAPD) processes at Layer 2 of the OSI model. Q.931 deals with Layer 3 of the OSI model. The D channel uses Q.931 signaling.

After the completion of call setup and connection establishment, the ISDN process is identical to conventional calls. ISDN protocols come into play again when the call is disconnected between the local switch and the terminal equipment. This process is fast and typically doesn't affect user data.

ISDN Layer 1

Layer 1 encompasses the physical connection between the ISDN circuit and the CPE. This layer is shared by the B and D channels alike.

Now we can come back to the protocols we touched on earlier. ISDN Layer 1 is governed by the following protocols:

- I.430— For BRI across the S/T interface.
- I.431— For PRI.
- ANSI T1.601— For the BRI U interface. (The U interface is not standardized by ITU-T.)

ISDN Layer 2

Layer 2 deals with the B and D channels separately, offering functions unique to each channel. It specifies LAPD as the framing protocol used for the D channel. On the other hand, the B channel uses High-Level Data Link Control (HDLC) or PPP encapsulation.

Protocol assignment for Layer 2 is as follows:

- Q.920— Specifies the ISDN functions.
- Q.921— Specifies signaling over the network.

As is the case with the conventional LAN setting, the ISDN network needs the hardware addressing to take place between all the linked devices. ISDN Layer 2 is responsible for such addressing. In addition, there is further discrimination within each device when it comes to different processes running in that device. Therefore, a terminal endpoint identifier (TEI), dynamically assigned to each router by the switch at bootup, is used in tandem with a service access point identifier (SAPI), which is a way to identify the types of messages sent across the network.

ISDN Layer 3

At Layer 3, the D channel is controlled by the Q.931 protocol. The Q.931 protocol is a part of the Digital Subscriber Signaling System 1 (DSS1) protocol suite, which deals with message exchange.

The B channel specifications include support for the network layer protocols, such as IP, IPX, and AppleTalk.

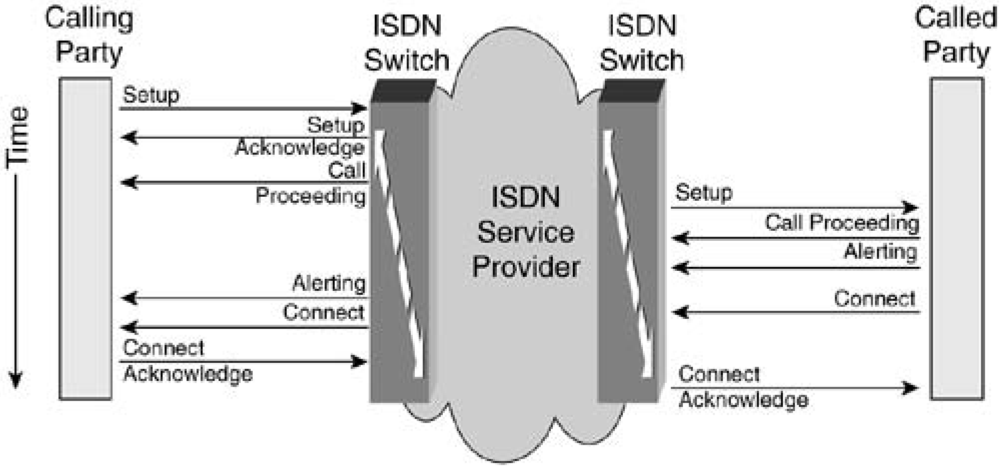
Examining ISDN Call Setup and Teardown

ISDN call setup and teardown reflect the activity of the Layer 3 Q.931 protocol. While an ISDN call is being set up, a number of messages are exchanged between the called and calling parties that identify the progress of a call setup.

Call Setup Process

As you can see in Figure 6-2, the called party requests a call setup. Some steps and messages that are displayed might not necessarily be a part of your particular call setup. It depends on the type of switches used in the exchange and their requirements.

Figure 6-2. ISDN Call Setup



Call Teardown Process

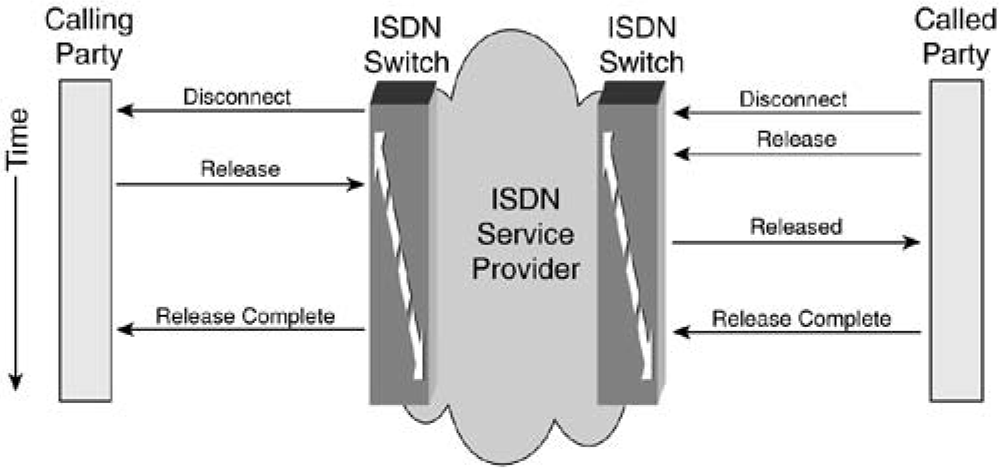
The teardown of a call may be initiated by either party. However, the switch handles the proceedings.

First, the Disconnect message is transmitted on the D channel. After the switch receives the Disconnect message, it starts the release of the B channel circuit and sends a Release message to the downstream switch. The involved switches eventually transmit the Release message to the final switch.

To make sure the call is being disconnected properly, each foregoing switch starts a T12 timer. It expects to receive a Released message from the neighbor switch, upon which it issues a Release Complete message back to the neighbor. If the Release Complete isn't received within the timer period, the Release message is reissued.

Keep in mind as you are consulting Figure 6-3 that call teardown is handled very rapidly throughout the network.

Figure 6-3. ISDN Call Teardown



Configuring ISDN

Configuring ISDN on a router involves setting up a number of global and interface commands. Some are mandatory, and some are optional. The "Scenarios" section discusses both kinds.

Typical tasks are as follows:

- Global parameters(mandatory)— Specify the switch type used by the CO. They set up static routes to various ISDN destinations and select conditions for initiating an ISDN call, such as interesting traffic.
- Interface parameters(mandatory)— Configure interface options, assign the interface to a dialer group, and map ISDN calls to the appropriate destinations.
- Other parameters(optional)— Include idle timers and response times to a call.

Most of these tasks aren't arranged in that particular order. You will probably go back and forth between configuration modes while setting up your ISDN.

Scenarios

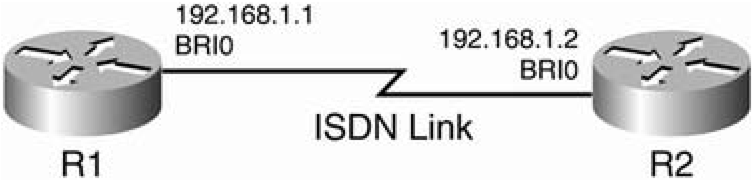
This section presents examples of ISDN configurations. Each new general command is discussed in detail the first time it is encountered. Each subsequent mention of a command is simply shown. This includes commands and concepts covered in previous chapters.

Scenario 6-1: Configuring a Simple ISDN Call

Here you will learn how to set up a simple ISDN call by using PPP encapsulation, defining interesting traffic, and specifying a carrier switch type and other service provider parameters.

In this scenario, DDR is configured to connect R1 to R2. Routing is achieved via a static route. The type of DDR used is "legacy" DDR, which uses dialer maps. Figure 6-4 shows the network layout.

Figure 6-4. Simple ISDN Call Topology



You will first look at R1's configuration, followed by that of R2.

Step 1: Configuring the Switch Type

The first thing you should do is specify the switch type. Table 6-3 shows a number of switches and their IOS command equivalents. As you can see, there are quite a few. The types of switches vary from country to country. Also, most switches are available in either basic or primary implementations for use with BRI or PRI, accordingly.

Table 6-3. Types of ISDN Switches

Command	Description
basic-1tr6	1TR6 ISDN switches (Germany)
basic-5ess	AT&T basic rate switches (U.S.)
basic-dms100	NT DMS-100 (North America)
basic-ni1	National ISDN-1 (North America)
basic-ni2	National ISDN-2 (North America)
basic-1tr6	1TR6 ISDN switches (Germany)
basic-nwnet3	Net3 switches (Norway)
basic-nznet3	Net3 switches (New Zealand)
basic-ts013	TS013 and TS014 switches (Australia)
basic-net3	NET3, also known as E-DSS1 or DSS1 switches (United Kingdom and Europe)
ntt	NTT ISDN switch (Japan)
primary-4ess	AT&T 4ess switch (U.S.)
primary-5ess	AT&T 5ess switch (U.S.)
primary-dms100	NT DMS-100 switch (U.S.)
primary-net5	NET5 switches (Europe)
vn2 to vn5	VN2, VN3, VN4, and VN5 ISDN switches (France)

Find out which one is used by your service provider. Make sure you are clear on the correct type of switch to avoid numerous problems.

The switch type can be configured in either global or interface configuration mode. Global mode controls the type of switch for all ISDN interfaces. The interface mode command applies it to that interface only. You've probably already guessed that if two different switches are specified for global and interface configuration, the interface takes precedence over the global for that particular interface.

To configure your CO's switch type, use

```
R1(config)#isdn switch-typeswitch-identifier
```

or

```
R1(config-if)#isdn switch-typeswitch-identifier
```

Select the AT&T 5ess switch as the CO ISDN switch type for all ISDN interfaces with

```
R1(config)#isdn switch-type basic-5ess
```

After you've specified the type of switch, you can configure the CHAP username and password for the remote router:

```
R1(config)#username R2 password Cisco
```

Step 2: Configuring the ISDN Interface

To specify the interface for use by ISDN, choose one of two available commands. The first one applies to routers with the native ISDN interface TE1:

```
R1(config)#interface brinumber
```

If native TE1 is not a part of your router setup, you need to designate a serial interface for use in ISDN. It becomes TE2 with external TA:

```
R1(config)#interface serialnumber
```

All subsequent commands that govern the interface take place in interface configuration mode. Whether you are using legacy DDR or dialer profiles determines whether most of your interface configuration tasks are applied to a logical or physical interface. Regardless, the ISDN interface is assigned a protocol address, an encapsulation option, a dialer group and, possibly, Service Profile Identifier (SPID) numbers (discussed in the next scenario).

Select the BRI 0 configuration mode:

```
R1(config)#interface bri 0
```

Define the BRI 0 IP address and net mask:

```
R1(config-if)#ip address 192.168.1.1 255.255.255.0
```

Set the PPP encapsulation for BRI 0:

```
R1(config-if)#encapsulation ppp
```

Add CHAP PPP authentication for BRI 0:

```
R1(config-if)#ppp authentication chap
```

Step 3: Configuring the Idle Timer

To prevent the link from staying up indefinitely, you can configure an idle timer. If there is no traffic on the link during the idle timer interval, the connection is terminated. The command to configure the idle timer is

```
R1(config-if)#dialer idle-timeoutseconds
```

Here's an example:

```
R1(config-if)#dialer idle-timeout 3600
```

Step 4: Configuring Dialer Maps

To place a call to a destination, a router needs some way of identifying it. The "DDR" section mentioned the existence of legacy DDR and its more advanced successor, dialer profiles. Here you will become familiar with the legacy DDR configuration through dialer maps.

In short, dialer maps associate the destination router's protocol address with a specific telephone number called the dial string. The command lets other options be specified as well. It's important to understand that it is applied to the physical interface—in this case, BRI 0.

```
R1(config-if)#dialer mapprotocol next-hop-address [namehostname] [speedspeed]

  [broadcast]dial-string
```

This syntax does not include all the options available for this command. The options shown here translate as follows:

- *protocol* is the Layer 3 protocol to which the phone number is mapped.
- *next-hop-address* is the Layer 3 protocol address.
- *hostname* is the name of the remote router used for authentication.
- *speed* is used for rate adaptation to request a lower speed than the standard DS0 64 kbps.
- Withbroadcast, broadcasts, such as routing updates, are forwarded to this address.
- *dial-string* is the destination's telephone number.

Multipledialer map statements identifying different destinations may be used on the same physical interface.

Create the dialer map command to specify IP as the name of the protocol, 192.168.1.2 as the IP address for the BRI interface of the next-hop router, R2 as the CHAP identification name for the remote router, and 2125552222 as the telephone number used to reach the BRI interface on the remote router:

```
R1(config-if)#dialer map ip 192.168.1.2 name R2 speed 56 2125552222
```

Step 5: Specifying Interesting Traffic

You might recall the definition of interesting traffic from the "DDR" section. The dialer-list command is used to identify interesting traffic. dialer-list has two versions: the so-called basic version and one that refers to an access list. The basic version allows or drops only packets belonging to an entire protocol:

```
R1(config)#dialer-listdialer-group-numberprotocolprotocol-name {permit | deny}
```

The access-list version adds the richness of all the options that can be defined by the extended access list:

```
R1(config)#dialer-listdialer-group-numberprotocolprotocol-namelist

  access-list-number
```

Let's look at what each element of this command represents. *dialer-group-number* is the dialer-list identifier that will be used in the next step of DDR configuration to assign this list to an interface. *protocol-name* specifies the Layer 3 protocol to be used.

The*access-list-number* argument matches an extended access list that is defined separately for the purposes of being used with the dialer-list command. The use of access lists to define interesting traffic is covered in the next scenario.

To define interesting traffic for R1, you need to exit interface configuration mode. Associate permitted IP traffic with dialer group 1. This means that the router initiates an ISDN call only for IP traffic.

```
R1(config)#dialer-list 1 protocol ip permit
```

Step 6: Assigning the Dialer List to an Interface

We've already mentioned that the *dialer-group-number* used in the previous command needs to match another command's argument that applies it to an actual interface. By referencing the same number in the dialer-group interface command, the dialer-list command set up in global configuration mode controls which packets are to initiate a call through that interface.

```
R1(config-if)#dialer-groupdialer-group-number
```

Enter interface configuration mode again, and associate the BRI 0 interface with dialer list 1:

```
R1(config-if)#dialer-group 1
```

Step 7: Configuring Routing

You were previously warned that if you choose to advertise routing updates and inadvertently don't prevent those updates from bringing up the link, you might be unpleasantly surprised when you receive a bill from your provider.

Do not despair. ISDN technology offers numerous options to successfully accomplish what you need while keeping charges in check:

- Static routes and default routes
- Floating static routes
- Dynamic routing with passive interfaces
- OSPF demand circuit
- Dialer watch
- Snapshot routing

Here you will examine the static route option; the rest are discussed in later scenarios.

Whenever you have a stub network, as is the case with this scenario, there is no real need to use dynamic routing, because all connections come from and go to the same point.

Set up a static route to R2's 192.168.100.0/24 network using this command:

```
R1(config)#ip route 192.168.100.0 255.255.255.0 192.168.1.2
```

Example 6-1 demonstrates R1's complete ISDN configuration that includes all previous steps.

Example 6-1. R1 ISDN Configuration

```
R1#show run

hostname R1
```

```
isdn switch-type basic-5ess
username R2 password Cisco
interface bri 0
 ip address 192.168.1.1 255.255.255.0
 encapsulation ppp
 dialer idle-timeout 360
 dialer map ip 192.168.1.2 name R2 speed 56 2125552222
 dialer-group 1
 ppp authentication chap
!
ip route 192.168.100.0 255.255.255.0 192.168.1.2
dialer-list 1 protocol ip permit
```

Let's now look at the configuration of R2. It needs to mirror R1.

Establish the CHAP username and password for the remote router:

```
R2(config)#username R1 password Cisco
```

Identify the BRI 0 IP address and net mask:

```
R2(config-if)#ip address 192.168.1.2 255.255.255.0
```

Specify R1 as the CHAP identification name for the remote router and 2125551111 as the telephone number used to dial up the remote router:

```
R2(config-if)#dialer map ip 192.168.1.1 name R1 speed 56 2125551111
```

Example 6-2 shows the ISDN configuration of R2.

Example 6-2. R2 ISDN Configuration

```
R1#show run
hostname R2
isdn switch-type basic-5ess
username R1 password Cisco
interface bri 0
 ip address 192.168.1.2 255.255.255.0
 encapsulation ppp
 dialer idle-timeout 360
 dialer map ip 192.168.1.1 name R1 speed 56 2125551111
 dialer-group 1
 ppp authentication chap
!
ip route 192.168.200.0 255.255.255.0 192.168.1.1
dialer-list 1 protocol ip permit
```

Scenario 6-2: Configuring DDR with Access Lists

In this scenario, you will look at how DDR commands can be used to define an extended access list to initiate ISDN calls. The topology used in the preceding scenario applies to this one as well. Some changes have been requested, however, so your configuration needs to be adjusted accordingly. The service provider switch is changed to a Northern Telecom DMS-100 model, and DDR must be configured on router R1 to connect to R2 for all IP traffic except Telnet and FTP.

Specify a Northern Telecom DMS-100 switch as the one used by the ISDN service provider:

```
R1(config)#isdn switch-type basic-dms100
```

> NOTE
>
> You should reload the router after changing the switch type to make the new configuration effective.

Configuring SPIDs

After you've specified the switch type, you might need to specify a SPID number. Not all switches require a SPID value, especially outside the U.S. Whenever the SPID number is required, you can find out the exact SPID information from your ISDN service provider.

SPIDS are dial-in numbers used by some service providers with certain types of switches, such as National ISDN1 and DMS-100. These numbers, which are similar to regular phone numbers, verify the services provided by your contract. SPIDs are available in spid1 and spid2 categories, one for each B channel.

Sometimes the keyword ldn might have to be placed at the end of the command line. ldn (local directory number) is assigned by the service provider and is used to make sure that calls are properly routed to both B channels.

The syntax for the spid commands is as follows:

```
R1(config-if)#isdn spid1spid-number [ldn]
R1(config-if)#isdn spid2spid-number [ldn]
```

Because you are using a DMS-100 switch, you need to configure SPID numbers:

```
R1(config-if)#isdn spid1 5551212
R1(config-if)#isdn spid2 5551213
```

PPP Authentication with a Different Host Name

In the last scenario, you set up CHAP authentication on each participating router to match the calling router's host name. You might run into a situation in which a username you set up for a calling router does not match its host name. For instance, not knowing a router's host name, dealing with a rotational host name, or simply shortening the task of storing a multitude of host names with their respective passwords would prompt you to skip the real host name and opt for an alternate. Cisco offers such an option for CHAP in its IOS.

To achieve this, perform a combination of actions. On the called router, such as R2, configure the username password command using an alternate host name:

```
R2(config)#username caller password Cisco
```

At the same time, match this alternate host name on the calling router, R1, with the following command:

```
R1(config-if)#ppp chap hostnamealternate-host-name
```

Replace the *alternate-host-name* argument in the real configuration with the word "caller."

Configuring DDR

Because you will configure the new dialer group number 2 for this scenario, first remove the dialer group 1 configured in the previous scenario.

Then associate the BRI 0 interface with dialer list 2 using the dialer-group command:

```
R1(config-if)#dialer-group 2
```

You can apply access lists to a dialer group to initiate dialing. The use of extended access lists when configuring ISDN is more common than specifying conditions in the dialer list itself.

Extended TCP access list entries are defined in the access-list 111 deny commands. They prevent FTP and Telnet packets from triggering calls.

```
R1(config)#access-list 111 deny tcp any any eq ftp

R1(config)#access-list 111 deny tcp any any eq telnet
```

The command access-list 111 permit allows all other IP traffic to start ISDN calls:

```
R1(config)#access-list 111 permit ip any any
```

The next command enables automatic DDR calling. It assigns access list 111 to dialer list 2, which in turn is applied to the BRI 0 interface by the dialer-group command already configured:

```
R1(config)#dialer-list 2 protocol ip list 111
```

Example 6-3 shows R1's new configuration.

Example 6-3. R1 Running Configuration

```
R1#show run
hostname R1
isdn switch-type basic-dms100
isdn spid1 5551212
isdn spid2 5551213
username R2 password Cisco
interface bri 0
 ip address 192.168.1.1 255.255.255.0
 encapsulation ppp
 dialer idle-timeout 360
 dialer map ip 192.168.1.2 name R2 speed 56 2125552222
 dialer-group 2
 ppp authentication chap
 ppp chap hostname caller
ip route 192.168.100.0 255.255.255.0 192.168.1.2
access-list 111 deny tcp any any eq ftp
access-list 111 deny tcp any any eq telnet
access-list 111 permit ip any any
dialer-list 2 protocol ip list 101
```

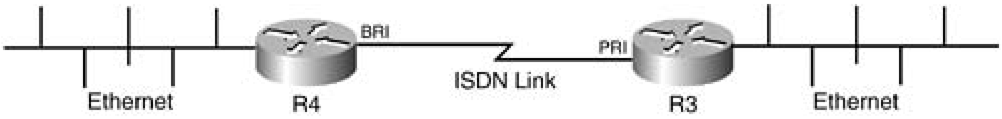
NOTE

The new configuration of R2 would follow the same logical pattern as that of R1. Therefore, it's not included here.

Scenario 6-3: Configuring PRI

Figure 6-5 shows the topology for this scenario. R3 possesses a PRI interface. You need to configure the router to allow R4 to access its Ethernet side via ISDN.

Figure 6-5. PRI Configuration Topology



You can begin your configuration by enabling the Ethernet interface on R3. This includes setting an IP address for the Ethernet interface. Here, it is 10.30.30.1/24.

The next step is to configure the ISDN switch type specified by the telephone company for your PRI connection. In this case, it is primary-5ess:

```
R3(config)#isdn switch-type primary-5ess
```

Now configure a username and password to be used for authentication when R4 tries to connect to R3. Let's say that the username is "R4" and the password is "Cisco":

```
R3(config)#username R4 password Cisco
```

You can now configure a dialer list to specify IP as the type of interesting traffic that initiates a call to R4:

```
R3(config)#dialer-list 1 protocol ip permit
```

Next, create the static route to R4's Ethernet network via its BRI address:

```
R3(config)#ip route 10.40.40.0 255.255.255.0 192.168.1.4
```

It's time to configure the T1 interface.

Configuring PRI

Configuring PRI interfaces involves the PRI-specific tasks discussed next, as well as the DDR-based commands you used in BRI configurations.

You start by configuring the ISDN PRI controller:

```
R3(config)#controller {t1 | e1} {slot/port | unit-number}
```

The t1 part of the command is used for North America and Japan. e1 is used for European facilities and much of the rest of the world. The *slot/port* or *unit-number* specifies the controller's physical slot, port location, or unit number.

You can use the following controller configuration command to select the frame type used by the PRI service provider:

```
R3(config-controller)#framing {sf | esf | crc4 | no-crc4}
```

Older T1 configurations use the sf (superframe) keyword. esf (extended superframe) is used for T1 PRI configurations. The crc4 | no-crc4 (cyclic redundancy check) options are for E1 PRI configurations.

The next command identifies the physical-layer signaling method. You need to satisfy the density requirement of 1s on the provider's digital facility. If there aren't enough 1s in the digital bitstream, the network switches and multiplexers can lose their synchronization for transmitting signals.

```
R3(config-controller)#linecode {ami | b8zs | hdb3}
```

ami means alternate mark inversion. hdb3 (high-density bipolar 3) is used for E1 PRI configurations.

Thelinecode and framing controller commands must match the framing and line-code types that are used at the T1/E1 WAN provider's CO switch.

NOTE

When T1 is used, framing esf and linecode b8zs are usually implemented. If E1 is used,framing crc4 and linecode hdb3 are applied.

Choose the clock source for the T1 with the following command:

```
R3(config-controller)#clock source {line [primary | secondary]| internal}
```

primary or secondary keywords are used for AS5000 to select either the primary or secondary TDM as the clock source.

Now that you've configured the controller, you can specify it for the PRI operation:

```
R3(config-controller)#pri-group timeslotsrange
```

This command identifies how many fixed timeslots the provider allocates. T1 uses values from 1 to 24, and E1 can range from 1 to 31.

The next command sets up an interface for PRI D channel operation:

```
R3(config-controller)#interface serial {slot/port: | unit :}{23 | 15}
```

This creates a serial subinterface to a T1/E1. The 23 argument refers to a T1 interface and designates channels 0 to 22 as the B channels and DS0 23 as the D channel. Alternatively, the 15 parameter is for an E1 interface and designates 30 B channels and timeslot 16 as the D channel.

On R3, configure your PRI as shown in Example 6-4.

Example 6-4. PRI Configuration

```
R3(config)#controller t1 1/0

R3(config-controller)#linecode b8zs

R3(config-controller)#clock source line

R3(config-controller)#framing esf

R3(config-controller)#pri-group timeslots 1-24

R3(config-controller)#interface serial 1/0:23
```

At this point you can continue with the DDR portion of the PRI configuration. The tasks you configure here are already familiar to you, such as PPP encapsulation, authentication, interface IP address, dialer group, and dialer map to the destination. This portion of the configuration is presented in Example 6-5.

Example 6-5. R3 Interface Configuration

```
R3(config-if)#encapsulation PPP
```

```
R3(config-if)#ppp authentication chap

R3(config-if)#ip address 192.168.1.3 255.255.255.0

R3(config-if)#dialer-group 1

R3(config-if)#dialer idle-timeout 90

R3(config-if)#dialer map ip 192.168.1.4 name R4 2125554444
```

Next, you will configure dynamic routing via EIGRP.

Configuring Routing

In the previous scenario, you learned how to set up routing via a static route. In most instances, however, this solution is not sufficient to satisfy the routing requirement. In larger environments, the scalability issue introduces the need for dynamic routing.

In this example, you can set up EIGRP routing with the following command:

```
R3(config)#router eigrp 100
```

This creates a new problem. If you introduce your network into the dynamic protocol, your links will constantly be brought up by the routing updates, right? Well, not if you configure passive interfaces. A passive interface listens to routing updates but doesn't forward them.

```
R3(config-router)#passive-interfaceinterface
```

The following command adapts this syntax for the current scenario:

```
R3(config-router)#passive-interface serial 1/0:23
```

Sometimes a situation occurs in which other networks need to be informed of the stub network existence. So, you need to configure the router to redistribute the static route to other routers in the network. Therefore, the static route will be redistributed into a dynamic protocol of your choice. For this purpose, apply the following:

```
R3(config-router)#redistribute static
```

Verification

You can verify your ISDN configuration by using several commands. For instance, you can check the status of the ISDN link by entering show isdn status, as shown in Example 6-6.

Example 6-6. Verifying ISDN Status

```
R3#show isdn status

The current ISDN Switchtype = primary-5ess

ISDN Serial1/0:23 interface

    Layer 1 Status:

        ACTIVE

    Layer 2 Status:

        TEI = 0, State = MULTIPLE_FRAME_ESTABLISHED
    Layer 3 Status:

        0 Active Layer 3 Call(s)

    Activated dsl 0 CCBs = 0

    Total Allocated ISDN CCBs = 0
```

You can check the D channel configuration by issuing the show interface serial 1/0:23 command, as shown in Example 6-7.

Example 6-7. Verifying the D Channel Subinterface

```
R3#show interface serial 1/0:23

Serial1/0:23 is up, line protocol is up (spoofing)

  Hardware is DSX1

  Internet address is 192.168.1.3/24

  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255

  Encapsulation PPP, loopback not set

  Last input 00:00:04, output 00:00:04, output hang never

  Last clearing of "show interface" counters never

  Input queue: 0/75/0 (size/max/drops); Total output drops: 0

  Queueing strategy: weighted fair

  Output queue: 0/1000/64/0 (size/max total/threshold/drops)

     Conversations  0/1/256 (active/max active/max total)

     Reserved Conversations 0/0 (allocated/max allocated)

  5 minute input rate 0 bits/sec, 0 packets/sec

  5 minute output rate 0 bits/sec, 0 packets/sec

     102 packets input, 618 bytes, 0 no buffer

     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles

     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort

     102 packets output, 571 bytes, 0 underruns
     0 output errors, 0 collisions, 6 interface resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions
  Timeslot(s) Used:24, Transmitter delay is 0 flags
```

Next, you can verify the configuration of the T1 controller. Enter show controller t1 1/0 to do this, as shown in Example 6-8.

Example 6-8. Verifying the Controller Configuration

```
R3#show controller t1 1/0
T1 1/0 is up.
  Applique type is Channelized T1 - unbalanced
  No alarms detected.
  Framing is ESF, Line Code is B8zs, Clock Source is Line.

  Data in current interval (580 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 3 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 3 Severely Err Secs, 0 Unavail Secs
```

If you want to monitor the ISDN connection in real time so that you can see the Layer 2 communication process, you can turn on Q.921 debugging by entering debug isdn q921.

Then you ping R4's Ethernet port. Assume that it is configured with the IP address 10.40.40.1. When the ping returns successful results, you can view the rest of the connection process, because Q.921 debugging has been turned on, as shown in Example 6-9.

Example 6-9. Monitoring the ISDN Connection in Real Time

```
ISDN Q921 packets debugging is on
```

```
R3#
ISDN Se1/0:23: RX <-  RRp sapi = 0  tei = 0 nr = 20
ISDN Se1/0:23: TX ->  RRf sapi = 0  tei = 0  nr = 15
ISDN Se1/0:23: TX ->  RRp sapi = 0  tei = 0 nr = 15
ISDN Se1/0:23: RX <-  RRf sapi = 0  tei = 0  nr = 20
ISDN Se1/0:23: RX <-  RRp sapi = 0  tei = 0 nr = 20
ISDN Se1/0:23: TX ->  RRf sapi = 0  tei = 0  nr = 15
ISDN Se1/0:23: TX ->  RRp sapi = 0  tei = 0 nr = 15
ISDN Se1/0:23: RX <-  RRp sapi = 0  tei = 0 nr = 20
ISDN Se1/0:23: TX ->  RRf sapi = 0  tei = 0  nr = 15
ISDN Se1/0:23: RX <-  RRf sapi = 0  tei = 0  nr = 20
```

NOTE

R4's configuration follows all the logical steps discussed in this and previous scenarios. It is not included here to save space. We believe that you can easily configure R4 on your own based on the information you've learned in this chapter.

Scenario 6-4: Alternative Identification Techniques

You already know how to configure identification through PPP authentication options PAP and CHAP. You also encountered the *alternate-host-name* parameter that can be used with CHAP. In this scenario, you will be introduced to two more identification options that can be used alongside or instead of those you've previously learned.

NOTE

Now that you've mastered general ISDN setup tasks, you can move on to more complex optional ones. This scenario and the subsequent ones in this chapter differ in their layout from the rest in that they don't provide topology examples. There is simply no need to repeat basic ISDN setup to introduce new steps. Any of the optional configuration parameters discussed in this scenario can be applied separately to an ISDN network, provided that basic ISDN has already been configured as described in prior scenarios.

Caller ID

The caller identification feature allows for screening of incoming ISDN calls. When the call is requested, the number supplied in the message is checked against a preexisting table of permitted numbers. This way, the call is not accepted until it is verified.

The syntax for the ISDN caller ID command is as follows:

```
Router(config-if)#isdn callernumber [callback] [exact]
```

This statement is applied to a called router. The *number* argument can be up to 25 characters long and can specify a range of numbers or partially known numbers. If you supply an X for any position in the number, it is treated as a "don't-care" digit, where the router accepts any number that matches the same position. Also, you can assign several numbers to an interface.

Thecallback keyword is used in the callback setups. The optional exact keyword demands the exact match to the configured number. In other words, if you don't have the exact option enabled, your router accepts any number of digits supplied by a caller as long as the same sequence of numbers appears in the configured telephone number.

When configuring caller identification, take care that your switch or access router supports this feature; otherwise, no calls will get through.

Unidirectional PPP Authentication

As mentioned in the previous chapter, the PPP authentication option, along with many other PPP options, must be bidirectional. This means that both routers participating in the connection setup have to authenticate one another.

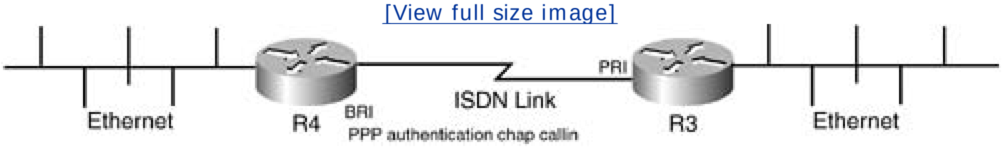
There is a way to bend this rule if you add an optional callin keyword at the end of the ppp authentication [pap | chap] command. This keyword specifies that authentication is to be used only if the router is on the receiving end of the call.

This issue comes up when one of the routers does not support authentication. Take a look at Figure 6-6. In this scenario, if R6 places a call to R2, it allows R2 to challenge R6, but it does not challenge R2 in return. However, if R2 places a call to R6 (a call in), R6 makes an authentication request from R2. The full syntax for the command is

```
R6(config-if)#ppp authentication [pap | chap]callin
```

Figure 6-6. One-Way PPP Authentication

[View full size image]



Scenario 6-5: Alternative Routing Methods

Previously, you learned how to configure static routing and dynamic routing via EIGRP. Here you will see how you can effectively implement OSPF as your dynamic routing method and use static routing as a backup method.

OSPF Demand Circuit

OSPF Demand Circuit (DC) is another feature that enables routing over ISDN without keeping the link constantly open. Perhaps you already know that to maintain neighbor relationships and ensure the accuracy of its link-state databases, OSPF sends Hello packets every 10 seconds and link-state advertisements (LSAs) every 30 minutes. Normally, it would keep the link up indefinitely.

The OSPF DC option was created to stifle periodic Hellos and LSAs. When DC is configured on a router, its Hello packets have a DC bit set, and its LSAs have a DoNotAge (DNA) bit set that suppresses those periodic refreshers. The way this works is at first OSPF creates adjacencies and synchronizes LSA databases in the usual manner. After this is done, OSPF keeps those adjacencies so that the routing updates can initiate an ISDN call only after a topological change has taken place.

To configure the OSPF DC, use the following command on your ISDN interface:

```
R1(config-if)#ip ospf demand-circuit
```

It has been argued by some that this command should be placed on routers at both ends of the call. However, it needs to reside only on the calling router. It is of no use to the receiving router. In instances where both routers can call one another, the use of OSPF DC is not recommended. Otherwise, you might run into a situation where both routers initiate a call simultaneously after the topological change, and the call will never get through.

A number of issues are associated with OSPF DC. If you are not careful while redistributing protocols into OSPF, you might cause routing loops and link flapping that keep the line up indefinitely because of constant "change" in topology.

Also, you might encounter a scenario where the ISDN interface's bandwidth, which figures into the OSPF metric of cost, equals that of the primary link. OSPF cost is based on a formula: cost = 100,100,000/bandwidth (bps). To keep the ISDN interface as a backup, you would have to manually assign it a very high cost to keep it from load balancing:

```
R1(config-if)#ip ospf costcost
```

Floating Static Routes

There are situations where you want your static routes to take the back burner to dynamic routing and be used only if other routes are unavailable. You would have to configure floating static routes. Normally, static routes have a default administrative distance of 1. This means that under ordinary circumstances they are preferred over dynamic routing protocols.

To switch this order manually, assign an administrative distance to the static route that is higher than the one of a dynamic route. We recommend using something above 200. Employ the familiar ip route command, but this time, add an *administrative-distance* argument at the end:

```
R1(config)#ip routedestination-network destination-subnet-mask {local-interface |

  next-hop}administrative-distance
```

Scenario 6-6: Configuring the Interface and the Backup Interface

The backup interface is used as an alternative to floating static routes. When an ISDN interface is configured as a backup, its status changes to standby, and its line protocol state changes to down. They remain that way until something happens to the main link. The command for the backup interface is configured under the principal interface (not the ISDN interface!). The syntax for the command is as follows:

```
R1(config-if)#backup interfaceinterface number
```

A number of optional parameters can be configured under a backup interface setup. The backup delay command specifies the amount of time (in seconds) that will lapse after the main interface fails and before the ISDN backup link is brought up. It also identifies how long after the principal link is repaired the ISDN interface stays up until it becomes inactive again. This command is used in conjunction with the backup interface command under the chief interface configuration. If backup delay is omitted, the ISDN interface kicks in instantaneously after the primary link failure and deactivates after the primary link is back. This isn't a good idea when you're dealing with a flapping connection.

```
R1(config-if)#backup delayactivation-time deactivation-time
```

NOTE

Unlike floating static routes, backup delay works only when the principal interface is physically down. It doesn't work under the administratively down status.

Thebackup load command is used in a bandwidth-on-demand scenario. It controls the percentage of the main link saturation before activating the ISDN interface as well as the percentage in the decrease of traffic before bringing the ISDN link down. It is also used together with the backup interface command.

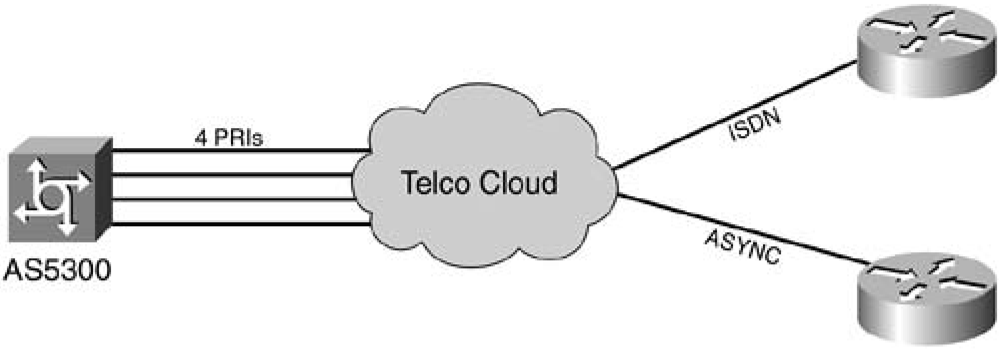
```
R1(config-if)#backup loadactivation-percentage deactivation-percentage
```

backup load can be configured alongside the backup delay command. Then, each one is responsible for its own sphere of influence.

Practical Exercise 6-1: Dialing Out with ISDN

This configuration has an AS5300 with four PRIs to allow Async and ISDN outbound connections, as shown in Figure 6-7. It can support 96 modem calls or a large number of ISDN calls. Static dialer maps are configured on the dialing side for each ISDN/Async connection. Static IP routes are used at both ends of the connection to avoid the unnecessary overhead of a dynamic routing protocol. Adding a remote location would require the addition of a dialer map, a username, and a static route for the new destination on the dialing side. All remote nodes have fixed IP addresses.

Figure 6-7. Dialing Out with ISDN



Your configuration should include the following:

- The PRI switch type, framing, and line coding
- The usernames and passwords of all the remote nodes you will be dialing into
- The IP addressing scheme

Practical Exercise 6-1 Solution

Example 6-10 shows the solution.

Example 6-10. Configuration Output

```
as5300#show running-config

hostname as5300

!

enable password somethingSecret

!

username remoteISDN01 password 0 open4u

ip subnet-zero

!

isdn switch-type primary-5ess

!

controller T1 0

framing esf

clock source line primary

linecode b8zs

pri-group timeslots 1-24

!

controller T1 1

 framing esf

 clock source line secondary 1

 linecode b8zs

 pri-group timeslots 1-24

!
```

```
controller T1 2
 framing esf
 clock source line secondary
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 3
 framing esf
 clock source line secondary
 linecode b8zs
 pri-group timeslots 1-24
!
interface Ethernet0
 ip address 171.68.186.54 255.255.255.240
no ip directed-broadcast
!
interface Serial0:23
encapsulation ppp
dialer rotary-group 2
isdn switch-type primary-5ess
isdn incoming-voice modem
!
interface Serial1:23
encapsulation ppp
dialer rotary-group 2
isdn switch-type primary-5ess
isdn incoming-voice modem
!
```

```
interface Serial2:23

encapsulation ppp

dialer rotary-group 2

isdn switch-type primary-5ess

isdn incoming-voice modem

!

interface Serial3:23

encapsulation ppp

dialer rotary-group 2

isdn switch-type primary-5ess

isdn incoming-voice modem

!

interface Dialer2

ip address 10.1.1.65 255.255.255.192

no ip directed-broadcast

encapsulation ppp


dialer in-band

dialer map ip 10.1.1.66 name remoteISDN01 broadcast 6665800

dialer-group 1

ppp authentication chap

!

ip classless

ip route 10.1.200.0 255.255.255.0 10.1.1.2

ip route 10.1.201.0 255.255.255.0 10.1.1.66

!

dialer-list 1 protocol ip permit
```

```
remoteISDN01#show running-config
!
hostname remoteISDN01
!
enable password somethingSecret
!
username as5300 password 0 open4u
ip subnet-zero
no ip domain-lookup
!
isdn switch-type basic-5ess
!
interface Ethernet0
ip address 10.1.201.1 255.255.255.0
no ip directed-broadcast
!
interface BRI0
ip address 10.1.1.66 255.255.255.192
no ip directed-broadcast
encapsulation ppp
dialer-group 1
isdn switch-type basic-5ess
ppp authentication chap
!
no ip http server
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.1.65
!
```

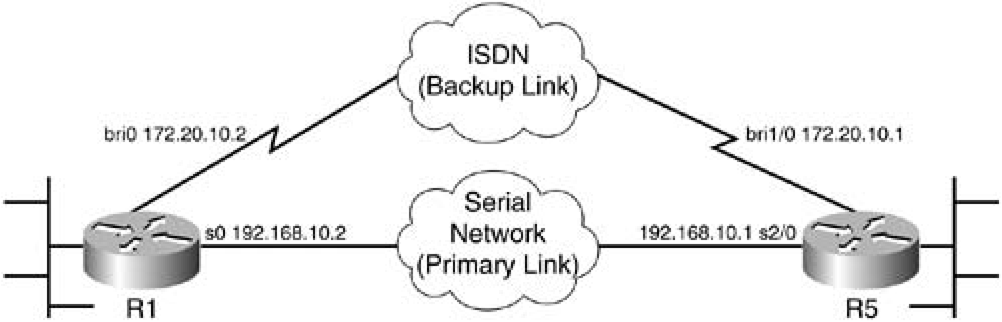
```
dialer-list 1 protocol ip permit
```

Practical Exercise 6-2: ISDN as a Backup

This configuration demonstrates the use of an ISDN BRI line to back up a leased-line connection, as shown in Figure 6-8. The backup interface command places the specified interface in standby mode until the primary interface fails.

Figure 6-8. DDR Backup Using BRIs and the backup interface Command

[View full size image]



This configuration also uses the Open Shortest Path First (OSPF) routing protocol between the two routers. As soon as the backup connection is activated, you must ensure that the routing table is updated to use the new backup route.

Practical Exercise 6-2 Solution

Example 6-11 shows the solution.

Example 6-11. Configuration Output

```
R1#show running-config
Building configuration...
!
!
hostname R1
!
aaa new-model
aaa authentication login default local
aaa authentication login NO_AUTHEN none
aaa authentication ppp default if-needed local
enable secret 5 <deleted>
!
username admin password 7 <deleted>
username R5 password 7 <deleted>
!
ip subnet-zero
no ip finger
!
isdn switch-type basic-ni
!
interface Loopback0
 ip address 172.17.1.1 255.255.255.0
```

```
!
interface Ethernet0
 ip address 172.16.1.1 255.255.255.0
!
interface Serial0
backup delay 10 30
backup interface BRI0
ip address 192.168.10.2 255.255.255.252
encapsulation ppp

no ip mroute-cache
no fair-queue
!
interface BRI0
ip address 172.20.10.2 255.255.255.0
encapsulation ppp
dialer idle-timeout 900
dialer map ip 172.20.10.1 name R5 broadcast 5551111
dialer map ip 172.20.10.1 name R5 broadcast 5551112
dialer load-threshold 1 outbound
dialer-group 1
isdn switch-type basic-ni
isdn spid1 51299699380101 9969938
isdn spid2 51299699460101 9969946
ppp authentication chap
ppp multilink
!
router ospf 5
```

```
 log-adjacency-changes
 network 172.16.0.0 0.0.255.255 area 0
 network 172.17.0.0 0.0.255.255 area 0
 network 172.20.10.0 0.0.0.255 area 0
 network 192.168.10.0 0.0.0.3 area 0
!
ip classless
no ip http server
!
access-list 101 remark Interesting traffic definition for backup link
access-list 101 permit ip any any
dialer-list 1 protocol ip list 101

R5#show running-config
Building configuration...
Current configuration:
!
!
hostname R5
!
aaa new-model
aaa authentication login default local
aaa authentication login NO_AUTHEN none
aaa authentication ppp default if-needed local
enable secret 5 <deleted>
!
username admin password 7 <deleted>
username R1 password 7 <deleted>
```

```
!
ip subnet-zero
!
isdn switch-type basic-ni
!
interface Loopback0
 ip address 172.22.1.1 255.255.255.0
!
interface BRI1/0
ip address 172.20.10.1 255.255.255.0
encapsulation ppp
dialer idle-timeout 900
dialer map ip 172.20.10.2 name R1 broadcast
dialer-group 1
isdn switch-type basic-ni
isdn spid1 51255511110101 5551111
isdn spid2 51255511120101 5551112
ppp authentication chap
ppp multilink
!
interface Serial2/0
ip address 192.168.10.1 255.255.255.252
encapsulation ppp
no fair-queue
clockrate 64000
!
!
router ospf 5
```

```
 network 172.20.10.0 0.0.0.255 area 0
 network 172.22.1.0 0.0.0.255 area 0
 network 192.168.10.0 0.0.0.3 area 0
!
ip classless
no ip http server
!
dialer-list 1 protocol ip any
```

NOTE

Remember that shutting down the primary interface administratively via the shutdown command will not bring up the backup link. You need to physically unplug the cables to verify the configuration's success.

After the backup link is activated, the OSPF table is exchanged, and the new routes using the backup link are installed. The traffic now flows across the backup link. Example 6-12 shows the results of the backup link operation.

Example 6-12. Verifying the Backup Link Functionality

```
R1#show ip route
Gateway of last resort is not set
     172.17.0.0/24 is subnetted, 1 subnets
C       172.17.1.0 is directly connected, Loopback0
     172.16.0.0/24 is subnetted, 1 subnets

C       172.16.1.0 is directly connected, Ethernet0
     172.20.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.20.10.0/24 is directly connected, BRI0
```

```
C       172.20.10.1/32 is directly connected, BRI0
     172.22.0.0/32 is subnetted, 1 subnets
O       172.22.1.1 [110/1563] via 172.20.10.1, 00:00:22, BRI0
R1#show interface BRI 0
    BRI0 is up, line protocol is up
     Hardware is BRI with U interface and external S bus interface
     Internet address is 172.20.10.2, subnet mask is 255.255.255.0
     MTU 1500 bytes, BW 256 Kbit, DLY 100000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
     Encapsulation PPP, loopback not set
     DTR is pulsed for 5 seconds on reset
     LCP Open, multilink Open
     Open: IPCP
```

Summary

This chapter covered the theory and configuration of ISDN. ISDN has numerous advantages over the traditional analog service while maintaining the investment in existing technology and providing high-speed service at a low cost. ISDN requires proper configuration for you to make the most of its services and avoid pitfalls. In this chapter, you learned how to enable legacy ISDN. You also saw the use of some PPP and DDR techniques and how they relate to ISDN. Table 6-4 summarizes the ISDN commands used in this chapter.

Table 6-4. Summary of ISDN Commands Used in This Chapter

Command	Description
isdn switch-type*switch-identifier*	Specifies the central office switch type on the ISDN interface.
interface bri*number*	Configures a BRI interface and enters interface configuration mode.
encapsulation ppp	Enables PPP encapsulation on an interface.
ppp authentication {pap | chap} [callin]	Enables PPP authentication and specifies the type.
ppp chap hostname*alternate-host-name*	Allows CHAP authentication through a name other than the host name.
dialer idle-timeout*seconds* [inbound |either]	Specifies the duration of idle time before a line is disconnected.
dialer map*protocol next-hop-address* [name*hostname*] [speed*speed*] [broadcast]*dial-string*	Configures a serial interface or ISDN interface to call one or multiple sites or to receive calls from multiple sites.
dialer-list*dialer-group-number* protocol*protocol-name* {{permit | deny} | list*access-list-number*}	Defines a DDR dialer list for dialing by protocol or by a combination of a protocol and a previously defined access list.
dialer-group*dialer-group-number*	Controls access by configuring an interface to belong to a specific dialing group.
isdn spid {1 | 2}*spid-number* [ldn]	Associates ISDN LDNs provided by your telephone service provider to the SPID.
framing {sf | esf | crc4 | no-crc4}	Selects the frame type for the T1 or E1 data line.
linecode {ami | b8zs | hdb3}	Defines the line code.
clock source {line [primary | secondary] | internal}	Sets the E1 line clock source for the Cisco AS5200 access server.
pri-group timeslots*range*	Specifies the channels to be controlled by the primary D channel.

interface serial {*slot/port*: | *unit* :}{23 | 15}	Specifies a serial interface created on a channelized E1 or channelized T1 controller (for ISDN PRI, channel-associated signaling or robbed-bit signaling).
ip route*destination-network destination-subnet-mask*{*local-interface* |*next-hop*} [*administrative-distance*]	Establishes static routes and defines the next hop for large-scale dial-out.
backup interface*interface number*	Configures an interface as a secondary or dial backup.
backup delay*activation-time deactivation-time*	Defines how much time should elapse before a secondary line status changes after a primary line status has changed.
backup load*activation-percentage deactivation-percentage*	Sets a traffic load threshold for dial backup service.
isdn caller*number* [callback] [exact]	Configures ISDN caller ID screening and optionally enables ISDN caller ID callback for legacy DDR.
isdn answer {1 | 2}*called-party-number*	Forces the router to verify a called-party number or subaddress number in the incoming setup message for ISDN BRI calls if the number is delivered by the switch.
show isdn status	Displays the status of all ISDN interfaces.
show interface*interface*	Displays information about the physical attributes of the ISDN interface.
show controller {t1*slot/port* | *bri*}	Displays information about the ISDN PRI or BRI.
debug isdn q921	Monitors the ISDN connection in real time.

Review Questions

1: Which of the following digital services does ISDN provide?

A. Voice

B. Data

C. Text

D. Graphics

E. Music

F. Video

G. All of the above

2: Which of the following services does an NT2 device perform?

A. Compression

B. Switching

C. Concentrating

D. Encryption

3: What type of interface can make up the R reference point?

A. EIA/TIA 232-C

B. X.25

C. c.V.24

D. V.35

4: What type of standard cable does the BRI U interface use?

A. Two-wire

B. Four-wire

C. Six-wire

D. BRI-wire

5: What happens when no more traffic is transmitted over the ISDN call?

A. An idle timer starts.

B. The call disconnects.

C. The bandwidth deteriorates.

D. Unidirectional flow changes directions.

6: What happens if the isdn switch-type command is used in global mode?

A. Only one interface accepts that switch type.

B. All ISDN interfaces assume the same switch type.

C. A few ISDN interfaces assume the same switch type.

D. Integrated services are enhanced.

7: True or false: Static routes are used in stub environments to save costs.

8: What type of framing is used for modern T1 PRI configurations?

A. sf

B. esf

C. crc4

D. no-crc4

9: Which linecode type is specified for T1 PRI configuration?

A. ami

B. b8zs

C. hdb3

D. None of the above

10: True or false: Rate adaptation can increase the ISDN channel speed.

Chapter 7. Optimizing the Use of DDR with Interface Dialer Profiles and Rotary Groups

This chapter covers the following topics:

- DDR and Dialer Profiles
- Dialer Rotary Group Overview
- Dialer Profiles and Dialer Rotary Group Configuration

The drawback of legacy dial-on-demand routing (DDR), as discussed in the preceding chapter, is that it cannot differentiate per user by specifying separate characteristics for various users. All calls made over the same physical interface must have the same configuration parameters. To sidestep this requirement, dialer profiles were created. They allow a user-specific profile to be configured on the router by separating the physical interface configurations from the logical configurations. Such profiles establish the characteristics of a particular user and then are dynamically allocated to the same interface for incoming or outgoing DDR calls.

DDR and Dialer Profiles

DDR consists of two portions: logical and physical. Network layer address, encapsulation, and dialer parameters are part of the logical portion of DDR. The interface that places and receives calls is the physical portion. When dialer profiles are implemented, the physical interfaces comprise a dialer pool and are allocated from this pool on an as-needed basis. A physical interface is borrowed from the dialer pool when a call is made. It is returned to the pool when the call is complete. Dialer profiles dynamically bind logical and physical configurations for each call. This allows the physical interface to take on different characteristics according to the requirements of an incoming or outgoing call. Remember that the combination of physical and logical characteristics is only temporary and lasts as long as the call.

Advantages of Dialer Profiles

Table 7-1 discusses the advantages of dialer profiles over legacy DDR.

Table 7-1. Dialer Profiles Versus Legacy DDR

LegacyDDR	Dialer Profiles
All ISDN B channels have the same configuration as the physical interface.	There is one configured logical interface per ISDN B channel.
One dialer map is required for every dialer for every protocol, which makes multiprotocol configurations very complex.	The dialer profile is a point-to-point interface that negates the requirement for a Layer 3-to-Layer 2 mapping and the subsequent complexities of managing multiple maps.
Dial backup is restricted because when a BRI or PRI is used to back up an interface, all the B channels go down, and the whole interface is idle.	Dialer profiles save the ISDN B channels by permitting the ISDN BRI interfaces to belong to multiple dialer pools. This allows a backup interface to be nondedicated and useable when the primary interface is still up.

In addition to the aforementioned perks, dialer profiles provide the ability to separate the logical portion of DDR from the physical interface, allowing you to

- Enable concurrent bridging over DDR interfaces to multiple sites
- Limit the number of minimum or maximum connections taking place on a DDR interface
- Assign different Layer 3 network addresses to different members of a physical interface
- Specify different encapsulations for different B channels
- Configure different members of a DDR interface with different DDR parameters

Dialer profiles support only PPP or HDLC encapsulation. PPP encapsulation is the most popular choice because it's nonproprietary and offers authentication options. This chapter's discussion focuses on PPP.

Dialer Profile Components

A dialer profile is a combination of the following components:

- Dialer interface— A logical portion of a dialer profile. The dialer interface governs all configuration settings for a destination. Each dialer interface can contain multiple dialer maps. Furthermore, different per-call parameters can be assigned to each dialer map defined in a dialer map class. The dialer interface defines the destination network protocol address, encapsulation type, type of PPP authentication, and dialer remote name for PPP PAP or CHAP. Other specified parameters include the dialer string/map, dialer pool number, interesting traffic lists, Multilink PPP, and optional timeouts.

- Dialer map class— An optional portion of a dialer profile that defines call characteristics for a specified destination. Map classes are designed to avoid having to identify the same call characteristics repeatedly for multiple interfaces. If a map class isn't used, a separate call characteristics definition is required for each dialer interface, even if those characteristics are identical for several dialer interfaces. The information included in a map class is tuned for each destination. This information can specify an ISDN speed of 56 kbps, whether it is a semipermanent connection, optional dialer timers such as dialer fast idle, dialer idle timeout, and dialer wait-for-carrier time.

- Dialer pool— A group of one or more physical interfaces of which each dialer interface is a member. Each dialer interface is associated with a dialer pool. A physical interface can be part of more than one dialer pool. You can also configure an optional priority, which determines outbound dialing contention for specific physical interfaces in the pool.

- Physical interfaces— Members of one or more pools. The configuration of a physical interface is limited to the encapsulation parameters. If required, Multilink PPP and PPP authentication are specified to enable identification of the dialer pools to which the interface belongs. The encapsulation method of the physical interface must match that of the dialer interface, which belongs to the same pool as the physical interface.

Dialer Profile Binding Sequence

As you know, dialer profiles specify the technique of dynamically binding the logical and physical configuration. It is the job of the NAS to associate dialer information with a physical port to accommodate the needs of a particular user dialing in to or out of the NAS. When multiple dialer profiles are configured on the NAS, it must determine which profile to bind for every call. The following two sections describe the binding sequence for dialing out and dialing in.

Dialing Out

The binding process for the outgoing calls works as follows:

1. When an outgoing packet arrives at the NAS, a route table lookup is performed, and the incoming packet from the network arrives. A route table lookup points to the destination

via the dialer interface.

2. When it is noted that the dialer interface is a dialer profile, the IOS determines whether an existing connection for this profile exists. If there is none, the software identifies the pool to which the dialer interface belongs.
3. The NAS searches for the first available physical interface of the pool that has the highest pool priority. When it is located, this interface is identified for use in dialing. It is then bound to the dialer interface, taking on the configuration of that dialer interface.
4. The telephone number for the dialer profile is dialed, and the regular DDR process takes place.

Dialing In

What makes the incoming call-binding process more complex than that for the outgoing calls is the fact that the called physical interface may be a member of multiple pools, and the pools, in turn, may be associated with multiple dialer profiles. The incoming call-binding process is as follows:

1. If the physical interface belongs to only one pool, which is associated with one dialer profile, the bind occurs between the physical interface and this dialer profile. If this isn't possible, the next step is a further attempt at binding known as an *approximate match*.
2. This attempt looks for a match of the Call Line ID (CLID) from the call with the dialer number from a dialer profile. However, the search involves only the profiles associated with the pool to which the dialed physical interface belongs. If there is a match, the physical interface is bound to the dialer profile that returned a match. If this step fails as well, proceed with the further binding attempt known as a *complete match*.
3. If PPP authentication is configured on the physical interface, the call is answered, and the caller is authenticated. In this case, the authenticated name is used to match the dialer profile that contains the same name in its configuration. Again, the only profiles that are checked are those that are associated with the same pools of which the called physical interface is a member. If the check returns a match, the physical interface is bound to the found dialer interface. If the complete match fails, the binding cannot occur, and the call is disconnected.

You might have realized that for the last binding attempt to be successful, the physical interface needs to have PPP encapsulation and PPP authentication enabled. Also, the physical interface engages in PPP Link Control Protocol (LCP) layer negotiations (described in Chapter 5, "Configuring Point-to-Point Protocol and Controlling Network Access") before binding to a profile. This means that if a dialer profile is using Multilink PPP, the physical interface must be configured for Multilink PPP as well because LCP negotiations might take place before the dialer profile is located.

After the bind has occurred, this does not mean that the connection has occurred as well. Just because the physical interface found the logical configuration to use, this does not imply that the call cannot be disconnected for other reasons. One such reason can be the maximum threshold configured for inbound calls. When the NAS locates an appropriate profile for an incoming call, it checks whether the profile has reached its maximum connection limit. If the current incoming call puts the profile's connection limit over its configured maximum, the call is disconnected.

Dialer Profile Limitations

Dialer profiles have certain limitations:

- Dialer profiles do not support dynamic encapsulation.
- The only supported encapsulation types are PPP and HDLC. X.25 and Frame Relay are not currently supported.
- The physical and dialer interfaces both require PPP authentication to be enabled.
- The maximum threshold for incoming calls is checked only after the call has been answered, so the charge applies regardless of whether the call is later disconnected because of the exceeded limit.
- Each dialer interface takes up an interface description block (IDB). IDB is an internal structure that manages an interface. Because a limited number of IDBs are available (the exact number depends on the hardware platform), dialer profiles might have certain scalability constraints.

Dialer Rotary Group Overview

Dialer rotary groups are designed to simplify configuration for multiple callers and multiple-destination environments by binding a single configuration to multiple physical interfaces. Synchronous, asynchronous, ISDN BRI, and ISDN PRI interfaces can make up a dialer rotary group. A physical interface that is configured as a member of a rotary group assumes configuration parameters for the group. A rotary group consisting of multiple physical interfaces applies the configuration of a *logical dialer interface*, also called a *virtual dialer interface*, to all its members.

When rotary groups are used, such characteristics as the IP address, interesting traffic definition, and call parameters are connected with the dialer interface rather than the physical interface. When a call comes into the router, the dialer interface selects a physical interface from the pool of physical interfaces.

With rotary groups, users of several BRIs or PRIs might get a single phone number from the service provider. Therefore, they allocate all their interfaces to a single rotary group so that only one number needs to be dialed. This kind of setup requires the remote routers to have only one set of dialer map statements for your destination. In turn, debugging and management on the user side are less complicated.

Dialer Profiles and Dialer Rotary Group Configuration

This section is divided into two portions:

- Configuring dialer profiles
- Configuring dialer rotary groups

Each part briefly describes the general configuration tasks involved in setting up dialer profiles and rotary groups. These tasks and the specific commands needed to configure them are described in more detail in the "Scenarios" section for both dialer profiles and rotary groups.

Configuring Dialer Profiles

Dialer profile configuration involves three separate stages:

1. Configure the logical dialer interface.
2. Configure the physical interface as a member of a dialer pool. At this stage you also specify the service parameters for the physical interface.
3. Optionally define the map class.

Let's briefly look at some commands that let you configure dialer profiles. A number of commands involved in this process create relationships between the elements of a dialer pool. Some of these commands belong to the physical interface configuration portion, and others belong to the dialer interface.

Among the dialer interface commands is the dialer string command, which specifies the destination's phone number. Multiple phone numbers may be included using dialer string. Starting with Cisco IOS Release 12.2(8)T, you can specify the order in which these phone numbers are to be used. Within the dialer string command, you can include the optional keywordclass, followed by the *map-class-name* parameter. When used, they specify a particular map class and pull the configurations from that map class for the call.

Another portion of the dialer profile configuration is to specify the pool of physical interfaces used to reach the target network. The pool is identified by a number between 1 and 255.

You can then associate a physical interface with a numbered pool and place the interface in that pool using a special physical interface command described in Scenario 7-1.

To make your configuration completely functional, two extra steps need to be taken:

1. Specify "interesting" traffic that will cause the link to be brought up.
2. Define the static routes to be used.

Configuring Dialer Rotary Groups

Five configuration stages set up dialer rotary groups:

1. Define interesting traffic.
2. Create a dialer interface.
3. Configure the physical interfaces as a rotary group.
4. Configure static routes.
5. Disable routing updates.

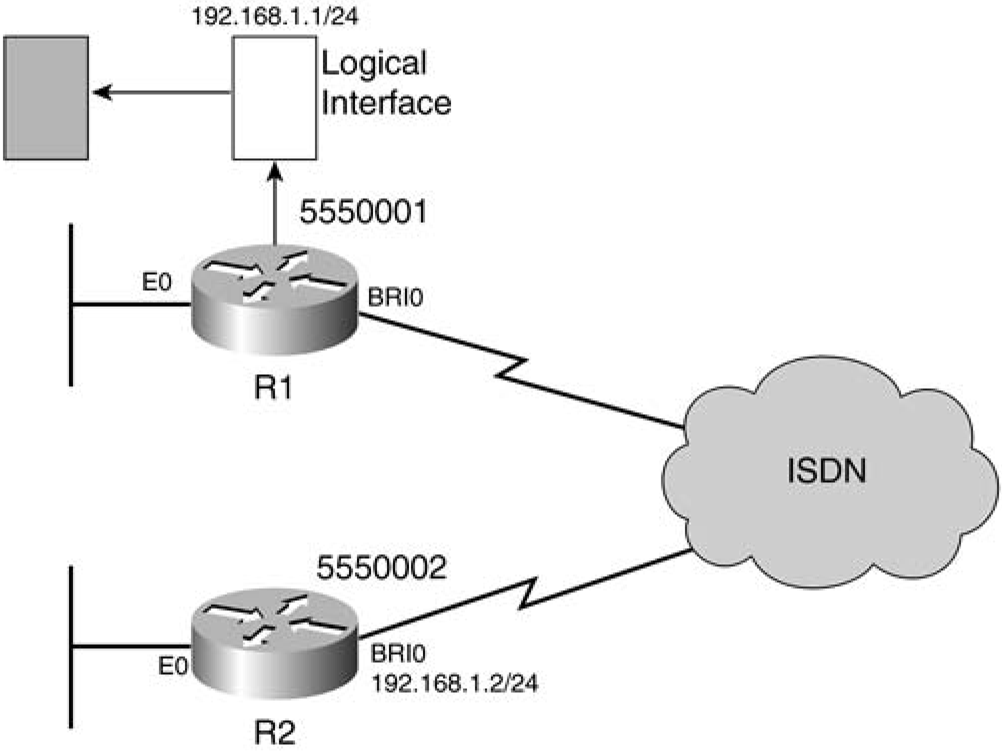
Scenarios

This section presents three scenarios. The first teaches you how to configure dialer profiles with a BRI interface. The second describes dialer profile configuration on a PRI interface. The third offers a dialer rotary group configuration example.

Scenario 7-1: Configuring Dialer Profiles

In this scenario, you enable DDR between R1 and R2, as shown in Figure 7-1.

Figure 7-1. Dialer Profile Configuration Topology



R2 has been preconfigured for legacy DDR, as shown in Example 7-1. Every command shown should already be familiar to you from Chapter 6, "Using ISDN and DDR Technologies to Enhance Remote Connectivity."

Example 7-1. Configuration of R2

```
R2#show running-config
```

```
hostname R2

!

!Output omitted for brevity

!

interface BRI0

 ip address 192.168.1.2 255.255.255.0

 encapsulation ppp

 dialer idle-timeout 120

 dialer map ip 192.168.1.1 name R1 broadcast 5550001

 dialer-group 1

 ppp authentication chap

!

dialer-list 1 protocol ip permit
```

Your assignment is to configure R1 with dialer profiles. As you know, to configure dialer profiles, you need to do the following:

- Configure one or more dialer interfaces.
- Configure an optional dialer map class to define different characteristics on a per-call basis.
- Configure the physical interfaces, and add them to a dialer pool.

Configuring the Dialer Interface

You can now begin the dialer profile configuration on R1. Before you can configure any commands for the dialer interface, you need to create it using the following command:

```
R1(config)#interface dialernumber
```

Theinterface dialer command puts you in dialer interface configuration mode. You can choose a number from 1 to 1000. After the dialer interface is created, you can set up the entire configuration for a destination inside it.

Under the dialer interface configuration, you need to specify the IP address of the dialer interface that the physical interface will later assume when the binding occurs. To assign an IP address to the dialer interface, use the following command:

```
R1(config-if)#ip addressaddress mask
```

Thedialer remote-name command identifies the name of the remote router, R2. This name is checked by CHAP authentication.

```
R1(config-if)#dialer remote-namename
```

The next command defines the destination router's phone number. You also have the option to define map classes.

```
R1(config-if)#dialer stringnumberclassmap-class-name
```

You can use multiple phone numbers with the dialer string command. Before Cisco IOS Release

12.2(8)T, the first telephone number in the dial string list was always the one used for a specific outgoing call. However, Release 12.2(8)T introduced the Rotating Through Dial Strings feature, which lets you customize the dial string usage order. By using this feature, you can specify the dialing order of multiple dial strings.

The syntax to configure the Rotating Through Dial Strings feature is as follows:

```
R1(config-if)#dialer order {sequential | round-robin | last-successful}
```

The options are as follows:

- sequential— The call uses the first dial string in the multiple strings list.
- round-robin— The call uses the next dial string in the list after the most recently successful string.
- last-successful— The call uses the most recently successful string.

Thedialer load-threshold command specifies the traffic load, which causes additional links to be brought up for Multilink PPP:

```
R1(config-if)#dialer load-thresholdload [outbound | inbound | either]
```

Valid*load* values are between 1 and 255, with 255 being 100% load. You also can choose to specify the direction of traffic for which the load is calculated.

Thedialer wait-for-line-protocol command forces the dialer to wait a specified amount of time for a line protocol after establishing a physical connection. If a call is dropped before the timer has expired, the call is considered unsuccessful, which creates conditions for a redial (if this is configured). This command is used only for the PPP encapsulation, because Cisco HDLC encapsulation is the default line protocol and always comes up. To set up the line protocol timer, use the following syntax:

```
R1(config-if)#dialer wait-for-line-protocolseconds
```

Theseconds value can range from 1 to 2147483.

Use the dialer hold-queue command to set the number of packets in queue while the line is coming up:

```
R1(config-if)#dialer hold-queuenumber
```

Thenumber argument is a value between 1 and 100.

Thedialer pool command is used to associate a dialer interface with a dialer pool:

```
R1(config-if)#dialer poolnumber
```

Substitute the *number* argument with a value between 1 and 255.

To tell the dialer interface which dialer list to use to determine the interesting traffic parameters, use the following command:

```
R1(config-if)#dialer-groupdialer-list-number
```

The group numbers should be in the range of 1 to 10.

You specify that Multilink PPP is to be used on the dialer interface with the following command:

```
R1(config-if)#ppp multilink
```

When the ppp multilink command is placed on the logical interface, it deals with outgoing calls; when placed on the physical interface, it is applied to incoming calls. For both incoming and outgoing calls, place this command on both physical and dialer interfaces.

Example 7-2 shows the dialer interface configuration portion of R1. Notice the following elements in the output:

- The phone number of the destination is 0002, with the map class DEPT.
- The dialer interface is assigned to pool 5.
- The IP address and mask are specified under the dialer interface configuration.
- The remote router name is set for CHAP authentication.
- List number 1 is specified for interesting traffic definition.

Example 7-2. Dialer Interface Configuration of R1

```
R1#show running-config

!

interface dialer1

ip address 192.168.1.1 255.255.255.0

 encapsulation ppp
```

```
dialer remote-name R2
dialer string 0002 class DEPT
dialer string 0012 class DEPT
 dialer wait-for-line-protocol 10
 dialer load-threshold 60 either
dialer hold-queue 12
dialer pool 5
dialer-group 1
 dialer order round-robin
 no cdp enable
 ppp authentication chap
 ppp multilink
!
dialer-list 1 protocol ip permit
```

Configuring the Map Class

Now you can configure an optional map class. The map-class dialer command is used to define a map class and subsequently enter map class configuration mode:

```
R1(config)#map-class dialerclass-name
```

In the preceding step, you specified DEPT as the class name. This means that the dialer1 interface on R1 is associated with map class DEPT, created by the map-class dialer command. Class names are case-sensitive.

As soon as you enter map class configuration mode, you can define parameters for the map class. Such commands may vary from one environment to the next. The commands used in this scenario are only examples, not requirements.

For instance, use the dialer isdn speed command to set an ISDN bit rate to 56 kbps for use in the map class:

```
R1(config-map-class)#dialer isdn speedspeed
```

Thedialer idle-timeout command causes the call to be disconnected if there is no activity on the link for the time specified. This helps you avoid unnecessary charges.

```
R1(config-map-class)#dialer idle-timeoutseconds
```

Theidle-timeout default is 20 seconds.

Thedialer fast-idle command is used when a call is waiting for the interface but the idle timeout hasn't yet expired. If the fast-idle command is specified, the current call is disconnected much faster so that the waiting call can get through. The syntax for this command is as follows:

```
R1(config-map-class)#dialer fast-idleseconds
```

The default fast idle time is 20 seconds.

Thedialer wait-for-carrier-time command causes the call to be dropped if no carrier is detected within the specified amount of time. Use the following syntax to issue the command:

```
R1(config-map-class)#dialer wait-for-carrier-timeseconds
```

The default value is 30 seconds. However, for asynchronous lines, the value should be at least 60 seconds to allow for delays in the telephone network.

Example 7-3 shows the configuration of map class DEPT. This is the same map class that is associated with the dialer1 interface. One of the set parameters is for the call to disconnect after 2 minutes of no data traffic.

Example 7-3. Map Class DEPT on R1

```
R1#show running-config

!

interface dialer1

dialer string 0002 class DEPT

!

! Output omitted for brevity

!

map-class dialer DEPT

 dialer isdn speed 56

dialer idle-timeout 120

 dialer fast-idle 20

 dialer wait-for-carrier-time 30

!
```

Configuring the Physical Interface

The final of the three dialer profile configuration tasks is configuring the physical interface and applying it to a dialer pool.

The first part of the physical interface configuration is to assign the interface in question to a dialer pool. Dialer pools can use a combination of synchronous, serial, BRI, or PRI interfaces. To include an interface in a dialer pool, issue the following command:

```
R1(config-if)#dialer pool-membernumber [prioritynumber] [min-linknumber]

  [max-linknumber]
```

Thedialer pool-member command can be used several times to assign the interface to more than one dialer pool. If you use the optional priority keyword, you can assign a priority to this interface within a particular pool. The valid priority numbers range between 1 and 255. The higher the number, the higher the likelihood that the interface will be chosen over other interfaces. The prioritization of interfaces within a pool applies only to dialing out. The min-link andmax-link options reserve the minimum and maximum number of ISDN B channels for an interface. The lowest number requirement is 1, and the highest is 255.

As mentioned, it's important to set PPP encapsulation, authentication, and multilink options on a physical interface for LCP negotiations to be successful and for subsequent profile binding to take place. When configuring the physical interface, don't forget to include these settings.

Example 7-4 shows R1's physical interface configuration. Here a physical interface is assigned to pool number 5. Notice that the Layer 2 protocol parameters have been configured as well.

Example 7-4. Physical Interface Configuration of R1

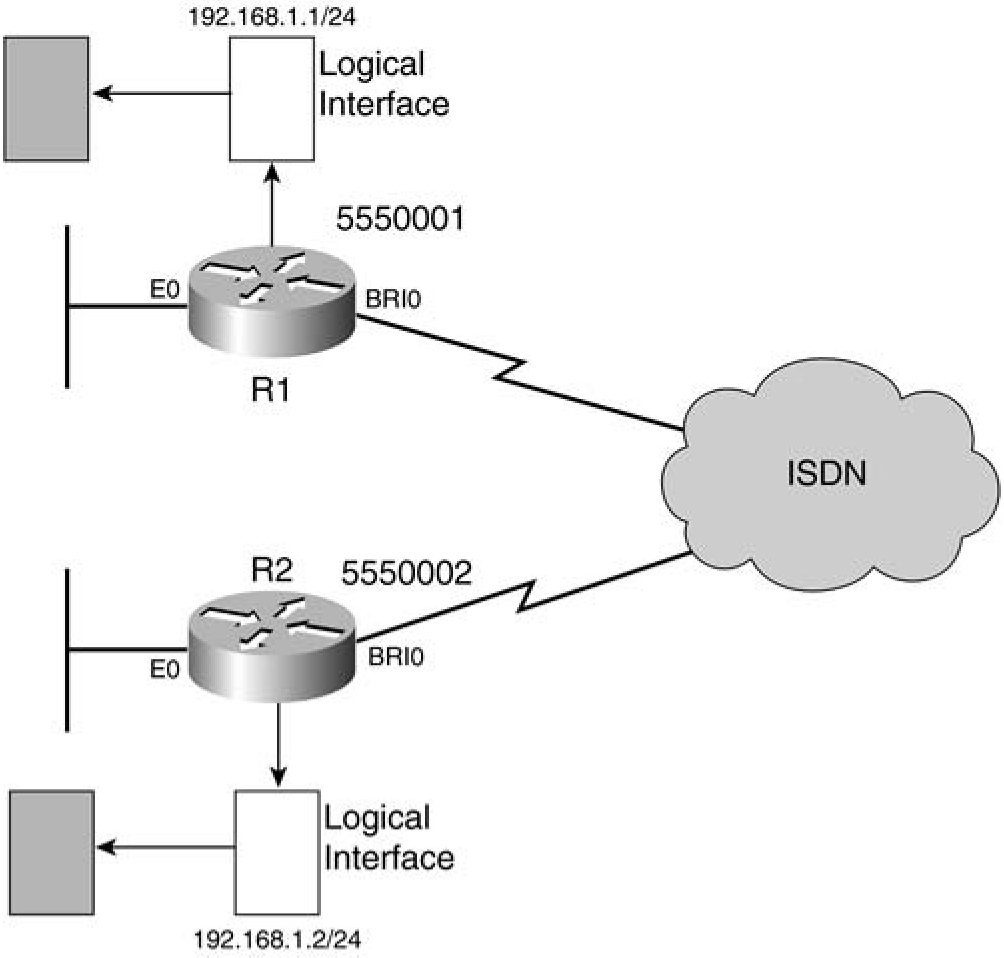
```
R1#show running-config

!

interface BRI0

 no ip address

encapsulation ppp
```

```
dialer pool-member 5

ppp authentication chap

ppp multilink

!

interface dialer1

 dialer pool 5

!

! Output omitted for brevity

!
```

Scenario 7-2: Configuring Dialer Rotary Groups

In this scenario you will learn all five stages of rotary group configuration. You will do so by reconfiguring R2, previously used in Scenario 7-1. Figure 7-2 illustrates the current topology.

Figure 7-2. Rotary Group Configuration Topology



Defining Interesting Traffic

The first step in the rotary group configuration is to define interesting traffic. You learned how to do so in Chapters 5 and 6. This section is a brief reminder. Packets that are considered interesting trigger a DDR call. Interesting traffic criteria can vary. The choices include protocol type, source address, destination address, and port number. To create an interesting traffic definition, you use the following command:

```
R2(config)#dialer-listdialer-group-numberprotocolname [permit | deny | list
  access-list-number]
```

Key components of this command are described in Table 7-2.

Table 7-2. Interesting Traffic Command Arguments

Argument	Description
dialer-group-number	References this dialer list using the same number as in the dialer-group command.
protocol*name*	Specifies which protocol packets are considered interesting for DDR, including IP, IPX, AppleTalk, DECnet, and VINES.
permit | deny	Permits or forbids the named protocol to initiate DDR. Can also optionally specify an access list.
list	References an access list created for greater precision in interesting traffic definition.

R2's interesting traffic definition, shown in Example 7-5, allows IP traffic to initiate DDR but not IPX.

Example 7-5. Defining Interesting Traffic on R2

```
R2#show running-config
!
hostname R2
!
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx deny
!
```

NOTE

In this scenario, IPX traffic would be denied with or without the list statement, because interesting traffic must be explicitly permitted.

Creating the Dialer Interface for Dialer Rotary Groups

The dialer interface created for rotary groups should include all configuration parameters that will later be applied to a physical interface when a call is made. Therefore, configuring a dialer interface has several stages of its own:

Step 1. Create a dialer interface with the following command:

```
R2(config)#interface dialernumber
```

The*number* element is used to produce a dialer interface. It also is used as a reference number for a rotary group. All subsequent configuration steps in this section take place in dialer interface configuration mode.

Step 2. Configure a local network address and mask. This address will be applied to the physical interface at the time of the call.

Step 3. Configure the encapsulation type, such as PPP.

Step 4. When PPP is used, configure PPP authentication.

Step 5. When using internal or external modems, use the following command:

```
R2(config-if)#dialer in-band [no parity | odd parity]
```

This command does not apply to ISDN interfaces, because they use out-of-band dialing on the D channel.

Step 6. Connect the dialer-list interesting traffic definition with the dialer-group command.

Step 7. Map the destination parameters with the dialer map command:

```
R2(config-if)#dialer mapprotocol next-hop-addressnamehostname [broadcast]

  dialer-string
```

Step 8. Force a dialer interface to be connected at all times with the new dialer persistent command. For all intents and purposes, it achieves the same effect as the dialer idle-timeout 0 command. The dialer idle-timeout command should not be configured when the dialer persistent command is present.

```
R2(config-if)#dialer persistent [delay [initial]seconds | max-attempts

  number]
```

The optional delay keyword sets the delay in seconds before a persistent connection attempts to reestablish after a network error. The initial keyword delays a persistent connection establishment after configuration or bootup and without interesting traffic. max-attempts is the maximum number of reconnecting attempts after a network error.

Step 9. Configure the number of redial attempts, the interval between redial attempts, and how long the interface is disabled if all redial attempts fail:

```
R2(config-if)#dialer redial intervalsecondsattemptsredials [re-enable

  disable-time-seconds]
```

Example 7-6 shows the dialer interface configuration on R2. If you compare this output with Example 7-1, you'll notice that the network address, encapsulation-related commands, interesting traffic, and destination coordinates have all moved from the physical interface into the dialer interface.

Example 7-6. Configuring the Dialer Interface on R2

```
R2#show running-config

!

interface dialer1

ip address 192.168.1.2 255.255.255.0

encapsulation ppp

 dialer remote-name R1

dialer map ip 192.168.1.1 name R1 broadcast 5550001

dialer-group 1

 dialer persistent delay initial 60

 dialer persistent delay 10

 dialer persistent max-attempts 5

 dialer redial interval 20 attempts 5 re-enable 3600

ppp authentication chap

!
```

Configuring Physical Interfaces as a Rotary Group

This portion of the configuration includes the physical interface in a rotary group. First, select an interface, BRI0 in this case, to comprise a rotary group. After you enter the configuration mode of that interface, create a dialer rotary group:

```
R2(config-if)#dialer rotary-groupnumber
```

The*number* argument should match the dialer interface number that you want your rotary group configuration to come from. No further configuration of the physical interface is required. All other parameters come from the dialer interface.

Configuring Static Routes

The last two steps in the dialer rotary group setup are not rotary group-specific and are needed for general DDR deployment. The first step is to configure a static route for each DDR calling destination:

```
R2(config)#ip routenetwork mask {address | interface} [distance]
```

Defining Passive Interfaces

The final step is to stop routing updates from triggering a DDR call by making your dialer interface passive:

```
R2(config-router)#passive-interface dialernumber
```

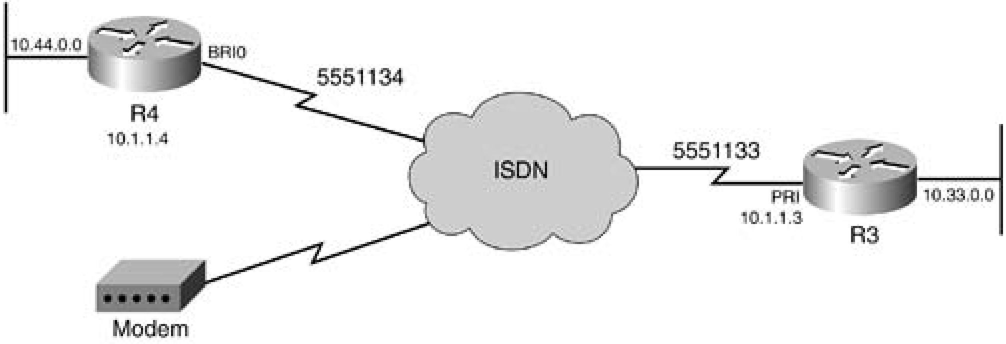
> NOTE
>
> You can also create certain conditions with an access list that prevent updates of a particular routing protocol from passing. You can then add the access list to the dialer-list statement.

Scenario 7-3: Configuring the PRI Interface to Receive Asynchronous Calls and ISDN Calls

In this scenario you will learn how to configure a PRI interface on R3 to receive asynchronous calls and existing ISDN calls from R4, as shown in Figure 7-3. R3 is a Cisco 3640 with a Fast Ethernet network module, a T1/ISDN PRI network module, and a 30-modem network module with five modems installed. This study combines some configuration tasks learned in this chapter and those learned in previous chapters.

Figure 7-3. PRI Interface Configuration Topology

[View full size image]



Assume that you have previously connected to R3 and configured the serial interface ISDN PRI channel with certain parameters. You can also assume that a PC has already been configured for asynchronous calls and R4 for BRI calls. While working on the R3 configuration, you will have to remove some of the existing statements as well as configure some new parameters.

As mentioned, R3 has been preconfigured to receive ISDN calls. Therefore, you should start by removing some of the old configuration. Example 7-7 shows the statements in need of removal.

Example 7-7. Removing the Old Configuration from R3

```
R3(config)#interface serial 1/0:23

R3(config-if)#no dialer idle-timeout 180

R3(config-if)#no dialer map ip 10.1.1.4 name R4 5551134

R3(config-if)#no dialer-group 1

R3(config-if)#no ppp authentication chap

R3(config-if)#no ip address

R3(config-if)#router eigrp 100
```

```
R3(config-router)#no passive-interface serial 1/0:23

R3(config-router)#no redistribute static

R3(config-rotuer)#no ip route 10.44.0.0 255.255.255.0 10.1.1.4
```

Now that the old configuration has been erased, you are ready to configure a PRI interface to receive ISDN and asynchronous calls. This process involves the following steps:

- Reconfiguring the PRI interface and adding it to dialer pool 4
- Creating an asynchronous group interface for use with internal modems
- Creating a dialer interface
- Configuring modem line characteristics
- Disallowing routing protocol updates to trigger DDR calls
- Verifying static routes

Configuring the PRI Interface

Enter serial 1/0:23 configuration mode, and configure the D channel to switch incoming analog calls to the internal modems:

```
R3(config-if)#isdn incoming-voice modem
```

Next you can assign the PRI interface to dialer pool 4:

```
R3(config-if)#dialer pool-member 4
```

Configuring an Asynchronous Group Interface

Here you need to configure an asynchronous group interface for R4's internal modems. Table 7-3 describes the commands needed to configure an asynchronous group interface.

Table 7-3. Asynchronous Group Interface Commands

Command	Description
interface group-async 1	Creates an asynchronous group interface.
ip unnumbered ethernet 0/0	Forces the group interface to use the IP address of the Ethernet port.
encapsulation ppp	Enables the use of PPP encapsulation on the interface.
ppp authentication chap	Specifies the PPP authentication type.
dialer in-band	Enables DDR on the interface and sends the data and the DDR control information over the same line.
dialer idle-timeout 180	Specifies a timeout of 3 minutes if no data is detected on the line.
dialer-group 1	Refers the interface to dialer list 1 for interesting traffic definition.
async mode interactive	Lets the dial-in user run Serial Line Internet Protocol (SLIP) and PPP at EXEC level on the line.
peer default ip address pool bigpool	Specifies that the interface allocates an IP address to any incoming call from the address pool bigpool.
no cdp enable	Disables the Cisco Discovery Protocol.
group-range 60–65	Identifies the modem lines in this group interface.

Creating a Dialer Interface

At this stage you create the dialer interface to allow R4 to connect using its BRI interface. Table 7-4 describes the usual configuration tasks.

Table 7-4. Dialer Interface Configuration Commands

Command	Description
interface dialer 1	Creates the dialer interface.
ip address 10.1.1.3 255.255.255.0	Configures the dialer interface's IP address and mask.
dialer idle-timeout 180	Sets an idle timer.
dialer-group 1	Recalls the dialer list that defines interesting traffic.
encapsulation ppp	Sets the encapsulation to PPP.
ppp authentication chap	Sets the PPP authentication to CHAP.
no peer default ip address	Stops the dialer interface from trying to assign an IP address to incoming calls.
ppp multilink	Enables Multilink PPP.
dialer remote-name R4	Identifies the remote router.
dialer string 5551134	Supplies R4's phone number.
dialer pool 4	Sets the dialer interface to use pool 4.

Configuring Modem Line Features

Now you need to configure the internal modem lines and their physical characteristics. Table 7-5 shows the list of commands needed to accomplish this.

Table 7-5. Modem Line Configuration Commands

Command	Description
line 60 65	Enters modem line configuration mode, which is used for asynchronous calls coming into the PRI interface.
autoselect during-login	Lets the router automatically select the correct protocol during login.
autoselect ppp	Specifies PPP as the autoselect protocol.
login local	Tells the router to check a local login username and password.
modem inout	Sets the modem lines to accept both incoming and outgoing calls.
modem autoconfigure discovery	Tells the router that the modem type is to be automatically discovered and configured for operation.
transport input all	Specifies that the lines will accept all protocols.
stopbits 1	Sets the number of stop bits for the data.
flowcontrol hardware	Configures the router to control flow by using RTS CTS signal lines.

Preventing Routing Updates from Triggering DDR Calls

The last stage of this scenario's configuration is preventing routing updates from making a DDR call. You must ensure that such updates will not be sent over the dialer interface. Table 7-6 lists the needed commands.

Table 7-6. Routing Configuration Commands

Command	Description
router eigrp 100	Enters routing protocol configuration mode.
passive-interface dialer 1	Specifies the dialer 1 interface as passive.
ip route 10.44.0.0 255.255.255.0 dialer 1	Assigns a static route to R4's Ethernet network address over the dialer 1 interface.

Verification

You know that to see the commands you've entered, you can enter show running-config. However, you can use other methods to verify the success of your configuration. Here are some additional testing techniques:

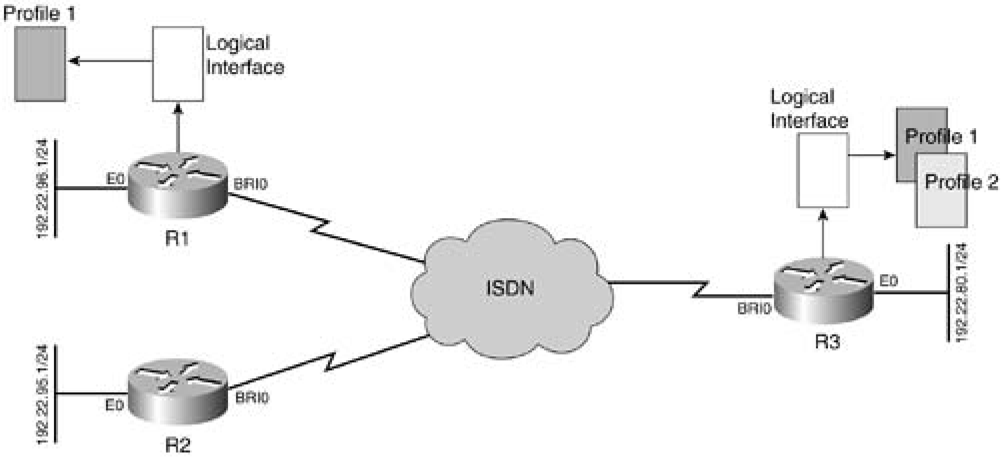
- You can dial into R3's ISDN PRI interface from the PC modem. The PRI interface should pass the call from the PC to the internal modems to be answered. If this action results in the "User access verification" message and a login prompt, the connection has been correctly established.
- You can dial in over the ISDN line from R4 by pinging R3's Ethernet port. To reach the Ethernet network, R4 dials into R3's ISDN PRI interface for connection establishment. You can watch the call setup activity and the ping response on R4 to verify whether the call was successful.
- You can use the debug command you learned in Chapter 6.

Practical Exercise: Configuring Dialer Profiles

In this exercise you provide DDR configuration for the three routers pictured in Figure 7-4. R3 is the central router that R1 (remote branch) and R2 (telecommuter) dial into.

Figure 7-4. Configuring Dialer Profiles

[View full size image]



Practical Exercise Solution

To complete this exercise you need to configure two dialer profiles on R3—one for each caller. Use the map class with the dialer string on R3 for R1. You also have to configure a dialer profile on R1 for its communication with the central site. R2 needs to be configured for legacy DDR.

The addressing scheme is shown in Figure 7-4. Use the EIGRP protocol when configuring R3. After you are finished with your configuration, you can verify it against Examples 7-8,7-9, and 7-10.

Example 7-8 shows the configuration of R3.

Example 7-8. Central Site Configuration

```
R3#show running-config

hostname R3

!

aaa new-model

aaa authentication login default local

aaa authentication ppp default local

!

username admin privilege 15 password 7 cisco

username R1 password 7 cisco

username R2 password 7 cisco

!

isdn switch-type basic-5ess

!

interface Ethernet0

  ip address 192.22.80.4 255.255.255.0

!

interface BRI0

  no ip address
  encapsulation ppp
  dialer pool-member 1
  isdn switch-type basic-5ess
  ppp authentication chap
  ppp multilink
!
interface Dialer0
  ip address 192.22.85.1 255.255.255.0
  encapsulation ppp
  dialer pool 1
  dialer remote-name R1
  dialer string 6661000 class mapclass1
  dialer load-threshold 128 outbound
  dialer-group 5
  ppp authentication chap
  ppp multilink
!
interface Dialer1
  ip address 192.22.86.1 255.255.255.0
  encapsulation ppp
  dialer pool 1
  dialer remote-name R2
  dialer string 6662000
  dialer-group 5
  ppp authentication chap
!
router eigrp 67
  redistribute static
  passive-interface Dialer0
  passive-interface Dialer1
  network 192.22.0.0
  auto-summary
!
ip classless
ip route 192.22.95.0 255.255.255.0 Dialer1
ip route 192.22.96.0 255.255.255.0 Dialer0
!
map-class dialer mapclass1
  dialer idle-timeout 180
  dialer fast-idle 5
!
dialer-list 5 protocol ip permit
!
```

Example 7-9 shows the configuration of R1.

Example 7-9. Remote Branch Configuration

```
R1#show running-config
hostname R1
!
aaa new-model
aaa authentication login default local
aaa authentication ppp default local
!
username admin privilege 15 password 7 cisco
username R3 password 7 cisco

!

ip subnet-zero

!

isdn switch-type basic-5ess

!

interface Ethernet0

ip address 192.22.96.1 255.255.255.0

!

interface BRI0

no ip address

encapsulation ppp

dialer pool-member 10

isdn switch-type basic-5ess

ppp multilink

!

interface Dialer1

ip address 192.22.85.2 255.255.255.0

encapsulation ppp

dialer pool 10

dialer remote-name R3

dialer string 6663000

dialer load-threshold 128 outbound

dialer-group 5

ppp authentication chap

ppp multilink

!

ip classless
ip route 192.22.0.0 255.255.0.0 192.22.80.0

ip route 192.22.80.0 255.255.255.0 Dialer1

!

dialer-list 5 protocol ip permit

!
```

Example 7-10 shows the configuration of R2.

Example 7-10. Telecommuter Configuration

```
R2#show running-config

hostname R2

!

aaa new-model

aaa authentication login default local

aaa authentication ppp default local

!

username admin privilege 15 password 7 cisco

username R3 password 7 cisco

!

isdn switch-type basic-5ess

!

interface Ethernet0

ip address 192.22.95.1 255.255.255.0

!

interface BRI0

ip address 192.22.86.2 255.255.255.0

encapsulation ppp
```

```
dialer map ip 192.22.86.1 name R3 6663000

dialer-group 1

isdn switch-type basic-5ess

ppp authentication chap

!

ip classless

ip route 0.0.0.0 0.0.0.0 192.22.86.1

!

dialer-list 1 protocol ip permit

!
```

Summary

In this chapter you learned how to configure dialer profiles. Dialer profiles allow separation of logical configurations from the physical interfaces that are later bound together when a DDR call is made. Dialer profile components include a dialer interface, dialer pool, physical interfaces, and an optional dialer map class.

To configure dialer profiles, you follow these steps:

Step 1. Configure a dialer interface.

Step 2. Configure an optional map class to be applied to the dialer interface.

Step 3. Configure the physical interfaces, and attach them to the same dialer pool as the appropriate dialer interface.

You also learned how to configure dialer rotary groups. Dialer rotary groups let you call multiple destinations at the same time by allowing a single logical interface configuration to be applied to a set of physical interfaces. Many of the rotary group configuration elements are identical to those of legacy DDR and dialer profiles. To configure dialer rotary groups, you follow these steps:

Step 1. Define interesting traffic.

Step 2. Create a dialer interface.

Step 3. Configure the physical interfaces.

Step 4. Configure static routes.

Step 5. Disable routing updates.

Review Questions

1: What is another name for a dialer interface?

A. Backup dialer interface

B. Ancillary dialer interface

C. Surrogate dialer interface

D. Virtual dialer interface

2: True or false: When a call is triggered, the dialer interface selects a physical interface from the pool.

3: Which of the following cannot be used in the logical configuration?

A. The network layer address

B. Encapsulation

C. The interface media type

D. Dialer parameters

4: True or false: When dialer profiles are used, an active BRI interface can function as a dial backup.

5: Which of the following interfaces can be used with dialer pools? (Choose all that apply.)

A. Frame Relay

B. Serial

C. BRI

D. PRI

6: What is the correct syntax to prohibit routing updates from being sent on the dialer 1 interface?

A. no routing update dialer 1

B. passive-interface dialer 1

C. dialer 1 no update

D. interface-passive dialer 1

7: What is the main advantage of using dialer rotary groups?

A. They simplify configuration for multiple callers and calling destinations.

B. They organize interface selection in a round-robin fashion.

C. They allow Multilink PPP to be implemented, but only on identical interfaces.

D. They are required for ISDN PRI channel selection.

8: What is the correct syntax for assigning a physical interface to a rotary group?

A. dialer rotary 1

B. rotary-group 1

C. dialer rotary-group 1

D. dialer-group 1

Chapter 8. Using DSL to Access a Central Site

This chapter covers the following topics:

- ADSL Overview
- Cisco 6160 DSLAM Overview
- Cisco 6400 UAC Overview
- DSL Access Architectures and Protocols

This chapter focuses on Digital Subscriber Line (DSL) technology. DSL, like cable modem, is one of the most popular broadband access methods and will be a new topic on the CCNP exam.

After completing this chapter, you will understand the basic Asymmetric DSL (ADSL) technology, Cisco 6160 DSL Access Multiplexer (DSLAM) configuration, and Cisco 6400 Universal Access Concentrator (UAC) configuration. You will also understand different access architectures and protocols such as Integrated Routing and Bridging (IRB), Routed Bridge Encapsulation (RBE), Point-to-Point Protocol over ATM (PPPoA), and Point-to-Point Protocol over Ethernet (PPPoE).

Note that there are different flavors of DSL technologies. This chapter focuses on ADSL technology.

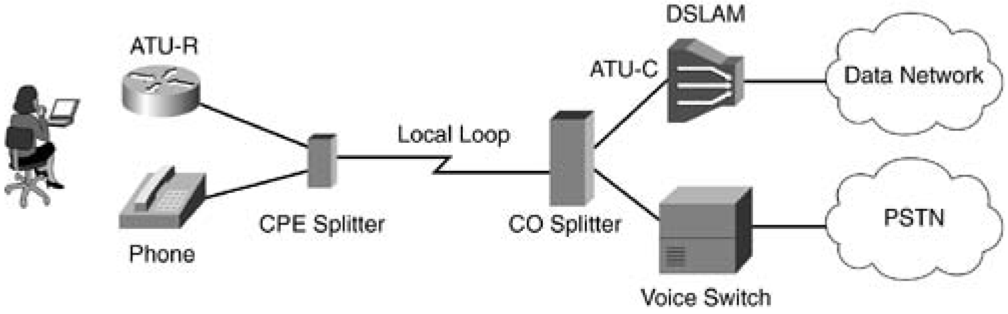
ADSL Overview

DSL technology introduces a new family of products that can provide high-speed data and voice service over existing copper pairs. Several flavors of DSL exist, but each type can be categorized as either SDSL or ADSL. *SymmetricDSL(SDSL)* provides equal bandwidth from the customer premises to the service provider (upstream) and from the service provider to the customer (downstream).*ADSL* provides higher downstream speeds than upstream.

Traditionally, ADSL has been used to provide high-speed data service by encoding data on the local loop by using frequencies (up to 1 MHz) greater than voice (up to 4 kHz) so that existing telephone service would be preserved and would travel simultaneously with the data. At the central office (CO), the voice would be routed to the public switched telephone network (PSTN) using a low-pass frequency filter called a POTS splitter chassis (PSC).

Figure 8-1 depicts a typical end-to-end ADSL system. Beginning at the customer premises, the user's general-purpose computer is connected to the ADSL Terminating Unit-Remote (ATU-R) over an Ethernet connection.

Figure 8-1. Typical End-to-End ADSL System



The ATU-R is typically connected to an external splitter device. In some cases, however, the external splitter is eliminated in lieu of an internal filter in the ATU-R and microfilters attached to plain old telephone service (POTS) devices in the home. From the splitter, the loop is wired to a Network Interface Device (NID) that serves as the demarcation point into the customer premises. From the NID, the loop is connected to a splitter device in the central office that splits off voice traffic and routes it to the PSTN. Data is connected to the ADSL Terminating Unit-Central Office (ATU-C). The user's data traffic is then typically routed across the ATM network to an aggregation gateway or router.

Modulation Methods

Three modulation methods for encoding data onto the local loop are Carrierless Amplitude and Phase (CAP), Discreet MultiTone 2 - Issue 2 (DMT2), and G.lite. DMT was selected as the preferred standard for ADSL modulation. CAP technology is cost-effective and readily available. G.lite is a simplified DMT encoding scheme that provides limited features to facilitate interoperability and minimize end-user interaction.

Table 8-1 shows the maximum data rates for downstream and upstream, line-coding technologies, and maximum reach. Note that the maximum-reach number is best-case, assuming "clean copper."

Table 8-1. ADSL Data Rates

Maximum Data Rate Downlink/Uplink	Line Coding Technology	Maximum Reach
8 Mbps/1 Mbps	CAP, DMT	18,000 feet/5.5 km
1.5 Mbps/640 kbps	G.lite	18,000 feet/5.5 km

Sources of Interference

Many sources of interference can degrade the quality of DSL. For instance, loading coils are used as a low-frequency (300 to 3300 Hz) filter but cannot be present for ADSL operation.

Other sources of interference include the following:

- Impedance changes
- Bridged taps
- Crosstalk
- Impulse hits

Techniques to Solve Interference

Several techniques exist for adjusting to interference:

- Rate-Adaptive DSL (RADSL)— Used to adjust the transmission rate.
- Reed-Solomon Forward Error Correction (FEC)— The process of correcting errors mathematically at the receiving end of a transmission path rather than calling for a retransmission.
- Bit interleaving— Used to avoid having consecutive errors delivered to the FEC algorithm at the receiving end of the circuit.
- Trellis coding— A modulation error-correction technique to improve error performance during reception.

Cisco 6160 DSLAM Overview

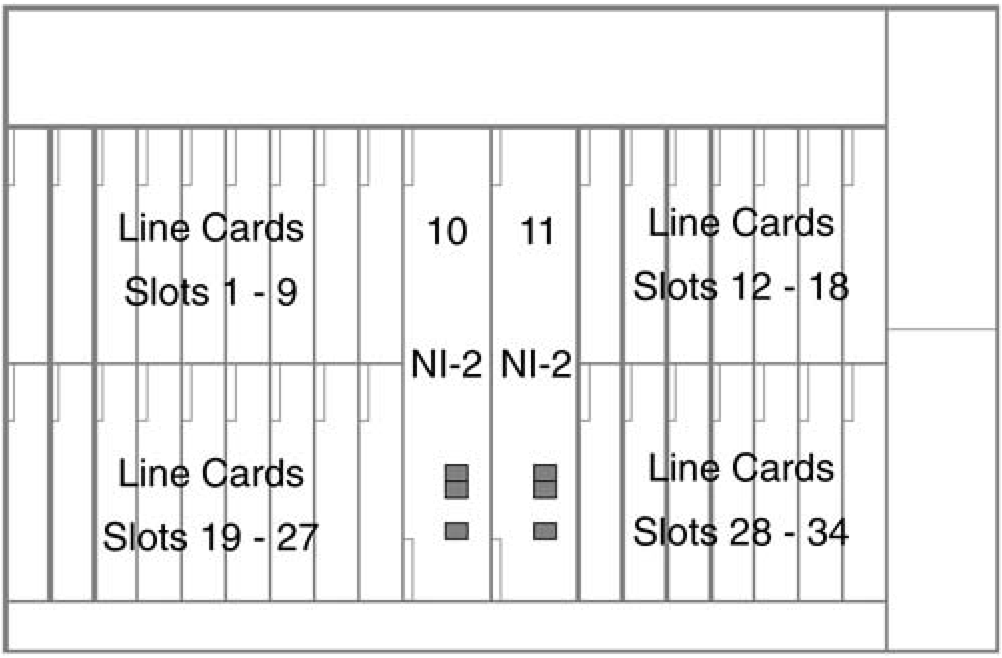
This section provides an overview of the Cisco 6160 DSLAM system and hardware components and discusses basic Cisco DSLAM configuration.

System and Hardware Components

The Cisco 6160 can be operated as a carrier class DSLAM with ADSL, SDSL, and Integrated Services Digital Network DSL (IDSL) interfaces. The Cisco 6160 is intended for use in North American central office facilities. The Cisco 6160 DSLAM can support up to 256 subscribers and concentrate traffic onto a single high-speed WAN trunk.

ExamineFigure 8-2. The chassis has 32 short slots for line cards and two double-length slots for Network Interface (NI-2) cards. Slots 10 and 11 hold the NI-2 cards. Slots 1 to 9 and 12 to 34 hold the line cards. Some of the essential functions the NI-2 card provides are ATM switching, WAN interface, and subtending.

Figure 8-2. Cisco 6160 DSLAM Chassis



WAN interfaces can be either OC-3c or DS3 and can be used for trunking or subtending. Subtending allows up to 12 other chassis to be subtended to a single host DSLAM system, aggregating the subtended systems through a single network uplink.

DSL line cards come in several varieties. In this chapter, the Quad Flexicard is used. It supports four ADSL connections and can be configured with CAP, DMT2, or G.lite line coding.

NOTE

You can install line cards of two or more different types in a single Cisco 6160 chassis. However, mixing different types of cards (Flexi ADSL, SDSL, and/or IDSL) on the same side of the chassis might result in decreased performance.

Basic Cisco 6160 DSLAM Configuration

In this section, you will learn all the necessary information to successfully configure the Cisco 6160 DSLAM.

Interface Numbering

Before you begin the configuration, it is important to know the interface numbering scheme used by the Cisco IOS software in the 6160. Interfaces whose names begin with ATM0 (ATM0/0, ATM0/1, and so forth) are NI-2 card WAN interfaces. ATM0/0 is the ATM switch's interface with the processor. There is no need to configure ATM0/0 unless you plan to use in-band management. ATM0/1 is the trunk port. ATM0/2 and ATM0/3, if present, are subtending interfaces.

Table 8-2 illustrates the interface numbering scheme for Cisco 6160 DSLAM.

Table 8-2. Cisco 6160 DSLAM Interface Numbering

Interface	Description
ATM0/0	The ATM switch's interface
ATM0/1	Trunk interface
ATM0/2	Subtend
ATM*A*/*B*	*A* = 1 to 34 (slot); *B* = 1 to 4 (port)
Ethernet0/0	Management Ethernet port

As shown in Table 8-2, interfaces whose names begin with ATM1 through ATM34 are line card interfaces. Ethernet0/0 is the interface for the LAN that connects the Cisco 6160 to its management system. For line card interfaces, the number before the slash indicates the slot number. The number after the slash indicates the interface or port number. For example, ATM5/4 is port 4 in slot 5.

Configuring Line Cards

Before you can use the Flexicard, you need to configure a slot for a specific card type. Use this command:

```
slotslot# cardtype
```

slot# is the slot number; the range is 1 to 34. *cardtype* is the card type for which you want to configure the slot. You must indicate the type of card. To configure the Quad Flexicard in slot 1 to use DMT modulation, you would enter the following:

```
lab-6160(config)#slot 1 ATUC-4FLEXIDMT
```

NOTE

You can use show hardware command to find out which cards are installed in the Cisco 6160 DSLAM.

Creating DSL Profiles

Except for a few dynamic operational modes, port configuration takes place through a configuration profile rather than by direct configuration. A *profile* is a named list of configuration parameters with a value assigned to each parameter. You can change the value of each parameter in the profile. To configure a subscriber, you need only attach the desired profile to that subscriber. When you change a parameter in a profile, you change the value of that parameter on all ports using that profile. If you want to change a single port or a subset of ports, you can copy the profile, change the desired parameters, and then assign the new profile to the desired ports. Multiple ports can share the same profile, but one port cannot have more than one profile. If you modify an existing profile, that change takes effect on every ADSL port linked to that profile.

Every port is attached to a special profile named "default" by default. You can modify the default profile (but not delete it). This is useful when you want to modify one or two default parameters and apply this to every port in the system (rather than creating a new profile with minor changes and attaching it to every port in the system).

When you create a profile, it inherits all the configuration settings of the default profile at the time of creation. If you subsequently modify the special profile default, the new changes to the default do not propagate to the previously created profiles.

To create a DSL profile, or to select an existing profile for modification, use the following command:

```
dsl-profileprofile-name
```

To delete a DSL profile, use the following command:

```
no dsl-profileprofile-name
```

In both examples, *profile-name* is the name of the profile you want to create, or an existing profile you want to delete or modify. To create a DSL profile called ccnp, you would enter the following:

```
lab-6160#configure terminal

lab-6160(config)#dsl-profile ccnp
```

After the DSL profiles are created, you can customize them with the following parameters:

- Bit rate
- DMT margin
- Check bytes
- Interleaving delay
- Training mode

The following sections discuss these parameters in more detail.

Setting the Bit Rate

To set the maximum and minimum allowed bit rates for the fast-path and interleaved-path profile parameters, use the following command:

```
dmt bitrate max interleaved downstreamdmt-bitrateupstreamdmt-bitrate
```

dmt-bitrate is a multiple of 32 kbps. If you enter a nonmultiple of 32 kbps, the Cisco IOS software aborts the command.

InExample 8-1, the command sets the maximum interleaved-path bit rate of the ccnp profile to 8032 kbps downstream and 832 kbps upstream.

Example 8-1. Setting the Bit Rate

```
lab-6160#configure terminal

lab-6160(config)#dsl-profile ccnp

lab-6160(config-dsl-prof)#dmt bitrate interleaved-path downstream 8032

  upstream 832
```

Setting the Margins

To set upstream and downstream signal-to-noise ratio (SNR) DMT margins, use the following command:

```
dmt margin downstreamdmt-marginupstreamdmt-margin
```

dmt-margin is equal to the upstream and downstream SNR margins in decibels. Values must be nonnegative integers. The range is from 0 to 127 dB.

> NOTE
>
> Research has shown that the optimum margins for DMT service are 6 dB downstream and 6 dB upstream.

InExample 8-2, the command sets the DMT SNR margins of the ccnp profile to 6 dB upstream and 3 dB downstream.

Example 8-2. Setting the Margin

```
lab-6160#configure terminal

lab-6160(config)#dsl-profile ccnp

lab-6160(config-dsl-prof)#dmt margin downstream 3 upstream 6
```

Setting Check Bytes

Check bytes are also called *FECbytes*. They are added to the user data stream to improve error correction, but they slow performance. To set upstream and downstream check bytes, use the following command:

```
dmt check-bytes interleaved downstreambytesupstreambytes
```

bytes values can be 0, 2, 4, 6, 8, 10, 12, 14, and 16. The default is 16 in each direction.

InExample 8-3, the command sets the interleaved check bytes for the ccnp profile to 6 upstream and 12 downstream.

Example 8-3. Setting the Check Bytes

```
lab-6160#configure terminal

lab-6160(config)#dsl-profile ccnp

lab-6160(config-dsl-prof)#dmt check-bytes interleaved

downstream 12 upstream 6
```

Setting Interleaving Delay

To set the interleaving delay parameter, use this command:

```
dmt interleaving-delay downstreamdelay-in-µsecsupstreamdelay-in-µsecs
```

delay-in-µsecs specifies the interleaving delay in microseconds. The default value is 16000 microseconds in each direction. Allowable values are 0, 500, 1000, 2000, 4000, 8000, and 16000 microseconds.

InExample 8-4, the command sets the interleaving delay of the ccnp profile to 2000 microseconds downstream and 4000 microseconds upstream.

Example 8-4. Setting the Interleaving Delay

```
lab-6160#configure terminal

lab-6160(config)#dsl-profile ccnp

lab-6160(config-dsl-prof)#dmt interleaving-delay downstream 2000 upstream 4000
```

Setting the Training Mode

Two training modes are available—standard and quick. Standard train relates to a training procedure specified in ANSI standards document T1.413, which is considered the standards reference for DMT ADSL. *Quick train*, also called *fast train*, uses a vendor-specific training sequence that is shorter than the standard training sequence.

To modify the training mode in a DMT profile, use the following command:

```
dmt training-mode {standard/quick}
```

InExample 8-5, the command sets the ccnp profile's training mode to quick.

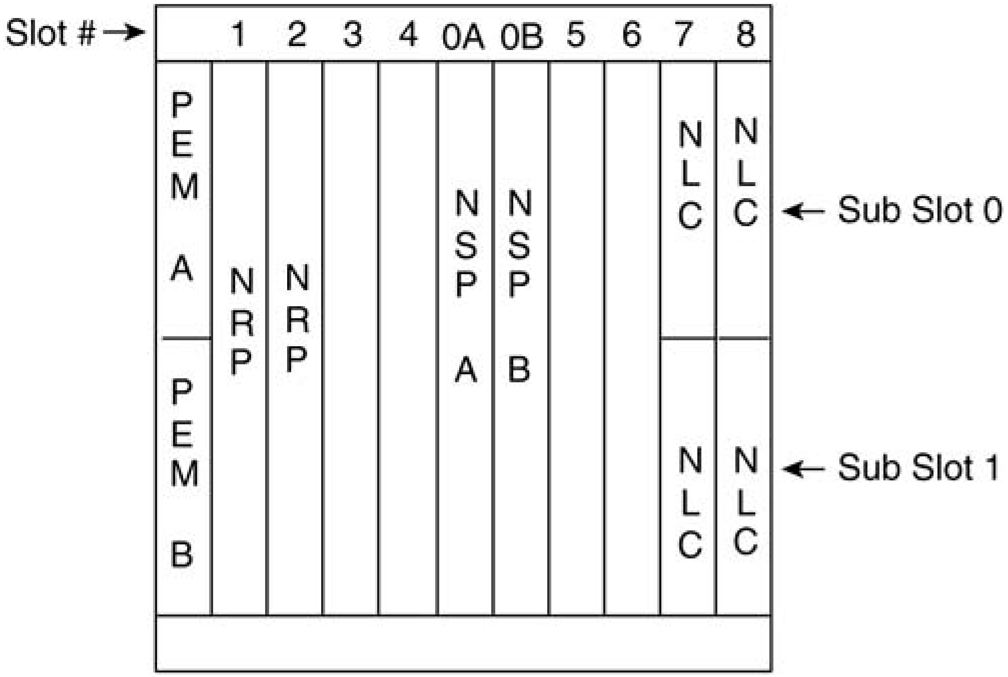
Example 8-5. Setting the Training Mode

```
lab-6160#configure terminal

lab-6160(config)#dsl-profile ccnp

lab-6160(config-dsl-prof)#dmt training-mode quick
```

Cisco 6400 UAC Overview

This section provides an overview of 6400 Universal Access Concentrator (UAC) hardware components (see Figure 8-3). Functional descriptions are provided for each component. How all the components work together within the system is also described.

Figure 8-3. Cisco 6400 UAC Hardware Component



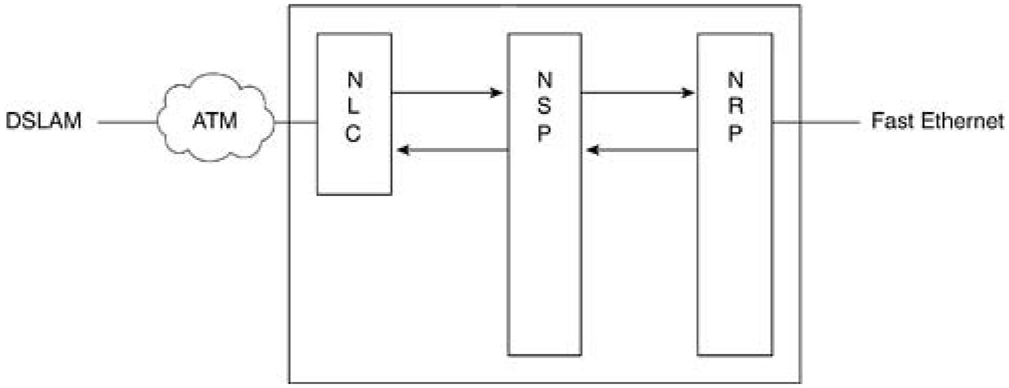
The 6400 is a broadband concentrator that supports Cisco's ATM services, PPP termination, and tunneling. The Cisco 6400 combines ATM switching and routing in a modular and scalable platform.

The 6400 UAC comprises three major functional components:

- Node Line Card (NLC)— A half-height line card. It features two OC-3 ATM interfaces and supports SONET APS 1+1 redundancy.
- Node Switch Processor (NSP)— The centerpiece of the 6400 system. It performs ATM switching and per-flow queuing for the ATM virtual circuits.
- Node Route Processor (NRP)— Based on the Cisco 7200 series router. It supports a variety of configurations, including PPP over ATM and RFC 1483 bridging. It is a full-height line card.

Figure 8-4 illustrates how these components work together.

Figure 8-4. Typical Traffic Flow for the Cisco 6400 UAC



The NLC receives traffic from the DSLAM or other ATM network. The NLC sends this traffic to the NSP. The NSP acts as an ATM switch. The ATM cells must be sent from the NSP to the NRP. The NRP handles routing functions for the 6400. The NRP reassembles the ATM cells into data packets and determines where the data needs to be sent. Direct data connections can be made via a Fast Ethernet port on the NRP. Other data packets are sent back through the NSP to the NLC, where these packets may be routed through the ATM network.

Understanding interface numbering is also important before you configure the 6400. The interface*slot/subslot/port* convention is used for both NLC and NRP. For NLC, the valid subslot and port number are 0 and 1. Because NRP is a full-height card, the subslot and port are always 0. In Example 8-6, NRP is installed in slot 1 and NLC is installed in slot 8, subslot 1.

Example 8-6. Cisco 6400 UAC Interface Numbering

```
interface atm 1/0/0   NRP in slot 1

interface atm 8/1/0   NLC in slot 8, sub-slot 1, port 0
```

All line cards are connected to the ATM backplane to the NSP. This interface is known as interface ATM0/0/0 and can be thought of as the interface to the NSP from an NLC or NRP card's perspective.Example 8-7 shows information about the NSP's ATM backplane.

Example 8-7. Internal Connection to the CPU Card

```
lab-6400NSP#show interface atm 0/0/0

ATM0/0/0 is up, line protocol is up

  Hardware is CPU card
  MTU 4470 bytes, sub MTU 4470, BW 155520 Kbit, DLY 0 usec,

     reliability 255/255, txload 1/255, rxload 1/255

  Encapsulation ATM, loopback not set

  Keepalive not supported

  Encapsulation(s):

  4096 maximum active VCs, 0 current VCCs

  VC idle disconnect time: 300 seconds

  Signalling vc = 35, vpi = 0, vci = 16

  UNI Version = 3.0, Link Side = user
```

To create an ATM PVC on the Cisco 6400, you can use the following command syntax:

interface atm*slot/subslot/port*

atm pvc*vpi vci***interface atm***slot/subslot/port vpi vci*

Example 8-8 shows you how to create an ATM PVC. From the NSP, to create PVC 1/100 coming from NLC 8/0/0 to NRP 1/0/0, the 6400 commands are as shown.

Example 8-8. Creating an ATM PVC from the NLC to the NRP

```
interface atm 8/0/0

atm pvc 1 100 atm1/0/0 1 100
```

DSL Access Architectures and Protocols

The following sections show you the different access architectures and protocols for the DSL service. Four types of access architectures and protocols are covered in this chapter.

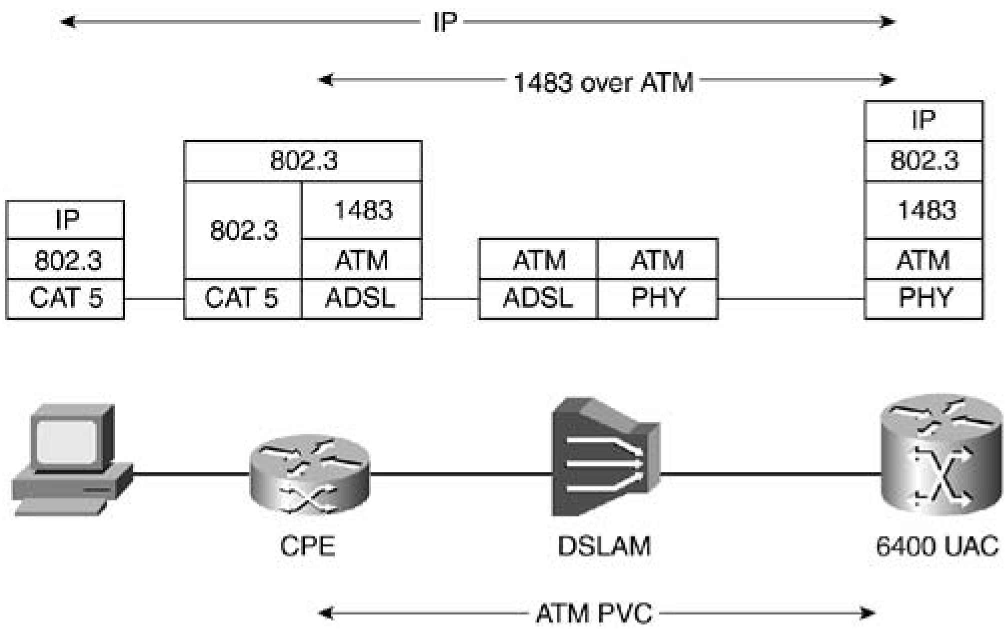
- IRB
- RBE
- PPPoA
- PPPoE

RFC 1483 Bridging and IRB Overview

When configured for RFC 1483 bridging, the ATU-R acts as a half bridge, forwarding all MAC frames not present on the LAN side to the WAN interface. In the case of 1483 bridging, 802.3 MAC frames are encapsulated along with an LLC/SNAP header into cells using AAL5 segmentation. The LLC/SNAP header is used to identify the protocols encapsulated to the remote end. Bridge groups are defined by associating VCs with each other. Bridge groups can be defined in several ways. Bridge members can communicate only with a network host, between each member and use IRB to route out of the bridge.

Bridge Group Virtual Interface (BVI) is a virtual interface that resides in the Cisco 827 and NRP. It acts as an interface between a bridge group and a routed interface. When configured for IRB, the BVI is assigned a number that corresponds to the bridge group that is used to associate the bridge group with the BVI. BVI is used as routed interface with network-layer attributes such as IP address, filtering, and so on. On BVI, routing is enabled on a per-protocol basis. BVI allows you to route a given protocol between routed interfaces and bridge groups. Figure 8-5 illustrates the RFC 1483 bridging protocol stack.

Figure 8-5. RFC 1483 Bridging Protocol Stack



To configure IRB, follow these steps:

Step 1. Enable IRB with the following code:

```
bridge irb
```

Step 2. Specify the bridge protocol to define the type of Spanning Tree Protocol:

```
bridgebridge-groupprotocol {ieee | dec}
```

Step 3. Specify a protocol to be routed in a bridge group:

```
bridge bridge-group route protocol
```

Step 4. Configure the ATM subinterface and aal5snap encapsulation:

```
interface atm slot/0.subinterface-number {multipoint | point-to-point}
  pvc [name]vpi/vci
  encapsulation aal5snap
```

Step 5. Assign a network interface to a bridge group:

```
bridge-group bridge-group
```

Step 6. Enables a bridge group virtual interface:

```
interface bvi bridge-group
```

Example 8-9 demonstrates the IRB configuration of the Cisco 6400 NRP.

Example 8-9. IRB Configuration

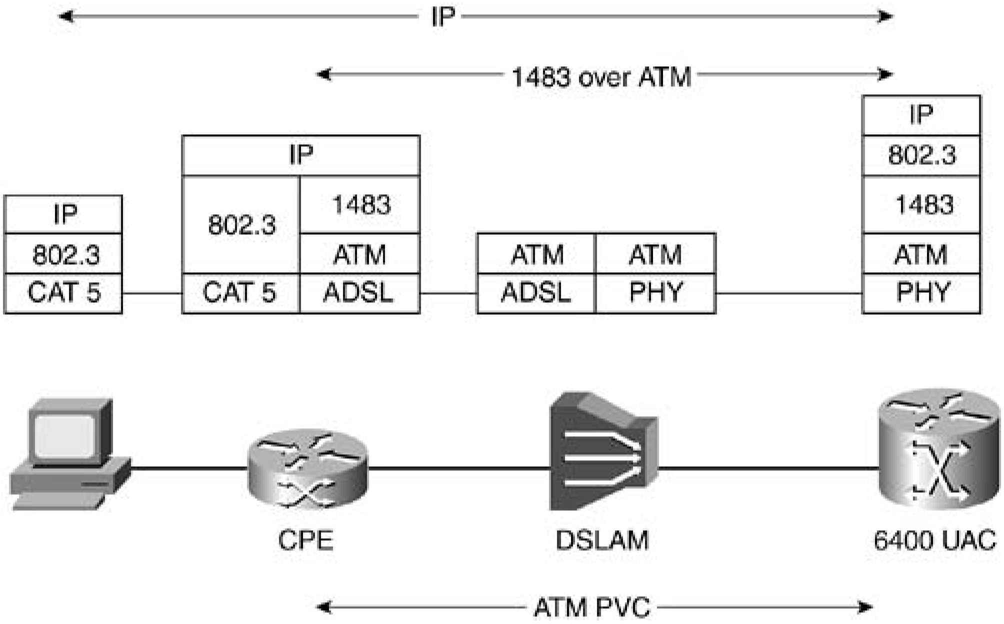
```
bridge irb
bridge 1 protocol ieee
 bridge 1 route ip
!
interface ATM0/0/0.133 point-to-point
 description Integrated
 no ip directed-broadcast
 pvc 1/33
  encapsulation aal5snap
 !
 bridge-group 1
!
interface BVI1
 ip address 10.1.1.1 255.255.255.0
```

RBE Overview

When configured for RBE, the CPE configuration remains the same as that in IRB. RBE is intended to address most of the RFC 1483 bridging issues, such as broadcast storms and security. The ATU-R behaves like the routed-bridge interface that is connected to an Ethernet LAN. For packets sending from the customer side, the destination IP address is examined, and the Ethernet header is skipped. If the destination IP address is in the route cache, the packet is fast-switched to the outbound interface. If the destination IP address is not the route cache, the packet is queued for process switching.

For packets destined for the customer devices, the destination IP address is examined first, and then the destination interface is determined from the IP routing table. To place a destination MAC address in the Ethernet header, the router checks the ARP table for that interface. If the MAC address is not found, the router generates an ARP request for the destination IP address and forwards the ARP request to the destination interface only. If an unnumbered interface is used and multiple subscribers are on the same subnet, the routed-bridge interface uses proxy ARP. All of these can be achieved without using a bridge group or BVI in the aggregation gateway and therefore are more scalable. Figure 8-6 illustrates the RBE protocol stack.

Figure 8-6. RBE Protocol Stack



To configure RBE, follow these steps:

Step 1. Configure the ATM subinterface, and use aal5snap encapsulation. ip unnumbered is used when subscribers are on the same subnet and to conserve IP address space. You can use the ip address command if subscribers are on different subnets.

```
interface atmslot/0.subinterface-number {multipoint | point-to-point}

ip unnumberedinterface-name-number

  pvc [name]vpi/vci

  encapsulation aal5snap
```

Step 2. Associate the RBE command with the ATM subinterface:

```
atm route-bridged ip
```

Step 3. Define the static host route. It is required if the IP unnumbered configuration is used.

```
ip routenetwork-number [network-mask] {address | interface} [distance]
  [namename]
```

Example 8-10 demonstrates the RBE configuration of the Cisco 6400 NRP.

Example 8-10. RBE Configuration

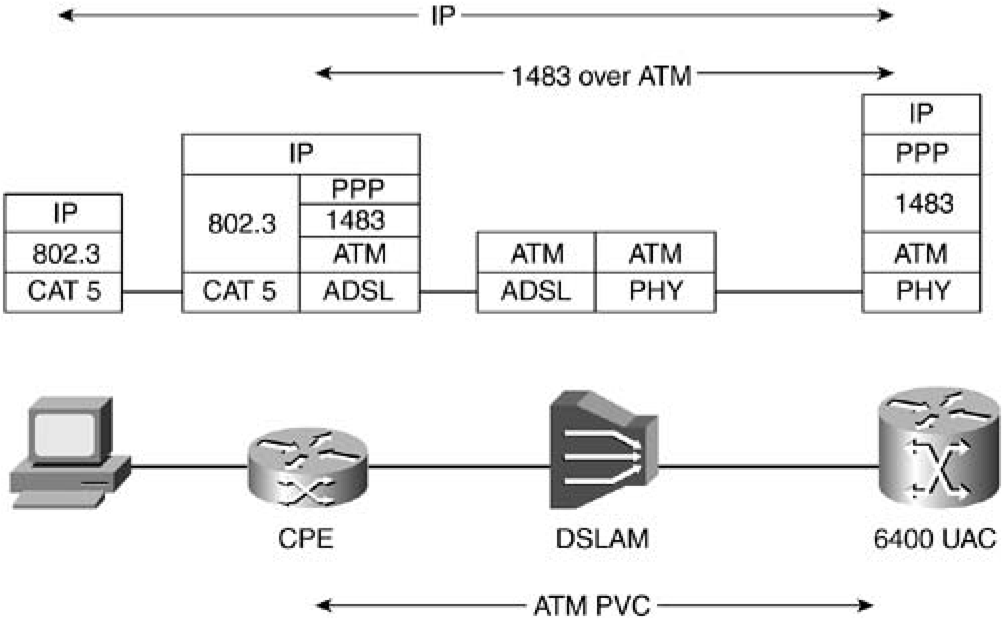
```
interface Loopback0
 ip address 192.168.1.1 255.255.255.0
 no ip directed-broadcast
!
interface ATM0/0/0.1 point-to-point
ip unnumbered Loopback0
 no ip directed-broadcast
 atm route-bridged ip
 pvc 1/35
  encapsulation aal5snap
!
```

```
ip route 192.168.1.2 255.255.255.255 ATM0/0/0.1
```

PPPoA Overview

When configured for PPP over ATM, the ATU-R acts as a router, and additionally provides DHCP and NAT services to the LAN side. In the case of PPP routing, IP packets are encapsulated into a PPP frame and then are segmented into ATM cells through AAL5. The PPP sessions initiated by the subscriber are terminated at the service provider that authenticates users, either using a local database on the router or through a RADIUS server. After the user is authenticated, IPCP negotiation takes place, and then the IP address gets assigned to the CPE. Figure 8-7 illustrates the PPPoA protocol stack.

Figure 8-7. PPPoA Protocol Stack



Follow the next steps to configure PPPoA. (Note that local authentication is used here and that the IP address for the CPE is assigned by the router. RADIUS can be used for these tasks.)

Step 1. Configure a username and password for local authentication:

```
usernamenamepasswordsecret
```

Step 2. Create an ATM subinterface and PVC:

```
interface atmslot/0.subinterface-number {multipoint | point-to-point}
  pvc [name]vpi/vci
```

Step 3. Configure PPPoA encapsulation, and associate a virtual template with it:

```
encapsulation aal5mux ppp virtual-templatenumber
```

aal5mux encapsulation is used for the PPPoA configuration. virtual-template serves as the template, and the virtual-access interface is cloned from the virtual template.

Step 4. Create a virtual template interface:

```
interface virtual-templatenumber
```

Step 5. Conserve IP addresses by configuring the ATM subinterface as unnumbered, and assign the IP address of the interface type you want to leverage:

```
ip unnumberedinterface-name-number
```

Step 6. Create the local IP address pool:

```
ip local poolname begin-ip-address-range [end-ip-address-range]
```

Step 7. Specify the pool for the interface to use:

```
peer default ip address poolpoolname
```

Step 8. Enable CHAP or PAP authentication on the interface:

```
ppp authentication {chap | pap | chap pap | pap chap} [if-needed]
  {default | list-name} [callin]
```

Example 8-11 demonstrates the PPPoA configuration of the Cisco 6400 NRP.

Example 8-11. PPPoA Configuration

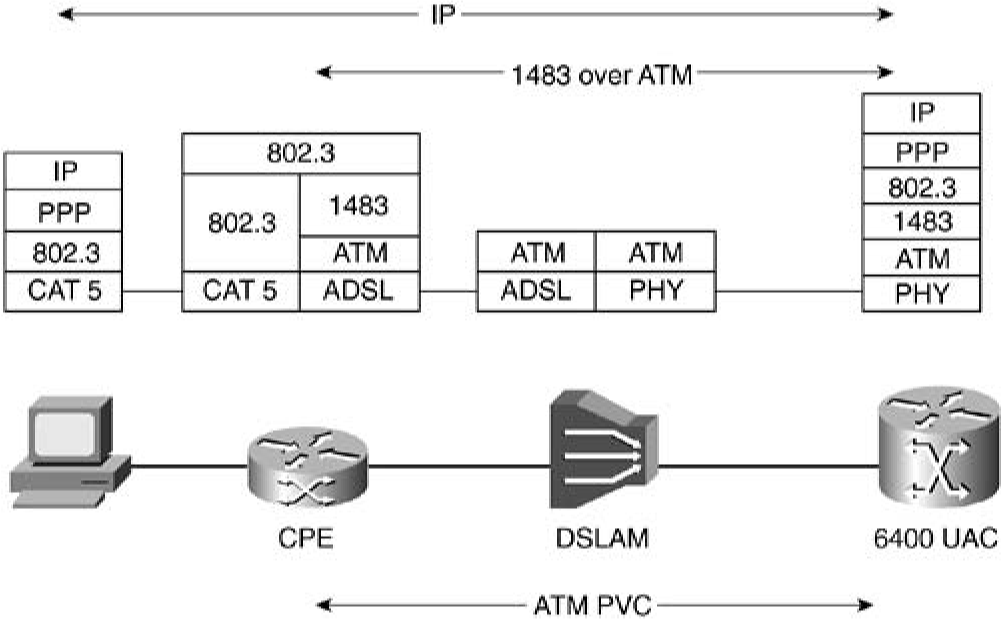
```
username cisco password 0 cisco
!
interface ATM0/0/0.133 point-to-point
 no ip directed-broadcast
 pvc 1/33
  encapsulation aal5mux ppp Virtual-Template1
 !
interface Virtual-Template1
 description PPPoATM
 ip unnumbered FastEthernet0/0/0
 no ip directed-broadcast
 peer default ip address pool ccnp
 ppp authentication chap
!
ip local pool ccnp 10.1.1.10 10.1.1.50
```

PPPoE Overview

For PPPoE, the ATU-R is transparent to this function, bridging the MAC/PPP frames across the WAN interface. The PPPoE feature allows a PPP session to be initiated on a simple bridging Ethernet-connected client. The session is transported over the ATM link via encapsulated Ethernet-bridged frames. The session can be terminated at either a local exchange carrier central office or an Internet service provider point of presence. The termination device is a Cisco 6400 UAC.

In the PPPoE architecture, the IP address allocation for the individual host running the PPPoE client is based on the same principle of PPP in dial mode—that is, via IPCP negotiation. Where the IP address is allocated from depends on the type of service the subscriber has subscribed to and where the PPP sessions are terminated. The PPPoE uses the dialup networking feature of Microsoft Windows. The IP address assigned is reflected with the PPP adapter. The IP address assignment can be either by the UAC or the home gateways if L2TP is used. The IP address is assigned for each PPPoE session. Figure 8-8 illustrates the PPPoE protocol stack.

Figure 8-8. PPPoE Protocol Stack



To configure PPPoE, follow these steps. (Note that local authentication is used here, and the router assigns IP addresses for the hosts. RADIUS can be used for these tasks.)

Step 1. Make sure Cisco Express Forwarding is enabled. If it isn't, use the following command to enable it:

```
ip cef
```

Step 2. Configure the username and password for local authentication:

```
usernamenamepasswordsecret
```

Step 3. Enable the virtual private dialup network (VPDN) configuration:

```
vpdn enable
```

Step 4. Configure the VPDN group to accept the dial-in and to be used to establish PPPoE sessions. Also specify the virtual template that will be used to clone virtual-access interfaces:

```
vpdn-groupnumber

accept-dialin

  protocol pppoe

  virtual-templatetemplate-number
```

Step 5. Create the ATM subinterface and PVC. Also configure AAL5SNAP encapsulation and specify the PPPoE protocol that the VPDN group will use:

```
interface atmslot/0.subinterface-number {multipoint | point-to-point}

pvc [name]vpi/vci

encapsulation aal5snap

protocol pppoe
```

Step 6. Create the virtual template interface:

```
interface virtual-template number
```

Step 7. Conserve IP addresses by configuring the ATM subinterface as unnumbered, and assign the IP address of the interface type you want to leverage:

```
ip unnumbered interface-name-number
```

Step 8. Configure the maximum transmission unit (MTU):

```
ip mtu 1492
```

Because Ethernet has a maximum payload size of 1500 bytes, the PPPoE header is 6 bytes and the PPP ID is 2 bytes, so the PPP MTU must not be greater than 1492 bytes.

Step 9. Create a local IP address pool:

```
ip local poolname begin-ip-address-range [end-ip-address-range]
```

Step 10. Specify the IP address pool for the interface to use:

```
peer default ip address poolpoolname
```

Step 11. Enable CHAP or PAP authentication on the interface:

```
ppp authentication {chap | pap | chap pap | pap chap} [if-needed]
  {default | list-name} [callin]
```

Example 8-12 demonstrates the PPPoE configuration of the Cisco 6400 NRP.

Example 8-12. PPPoE Configuration

```
username cisco password 0 cisco
!
vpdn enable
!
vpdn-group 1
 accept-dialin
```

```
  protocol pppoe
  virtual-template 1
!
ip cef
!
interface ATM0/0/0.133 point-to-point
 no ip directed-broadcast
 pvc 1/33
  encapsulation aal5snap
  protocol pppoe
 !
interface Virtual-Template1
 ip unnumbered FastEthernet0/0/0
 no ip directed-broadcast
 ip mtu 1492
 peer default ip address pool ccnp
 ppp authentication chap
!
ip local pool ccnp 10.1.1.10 10.1.1.50
```

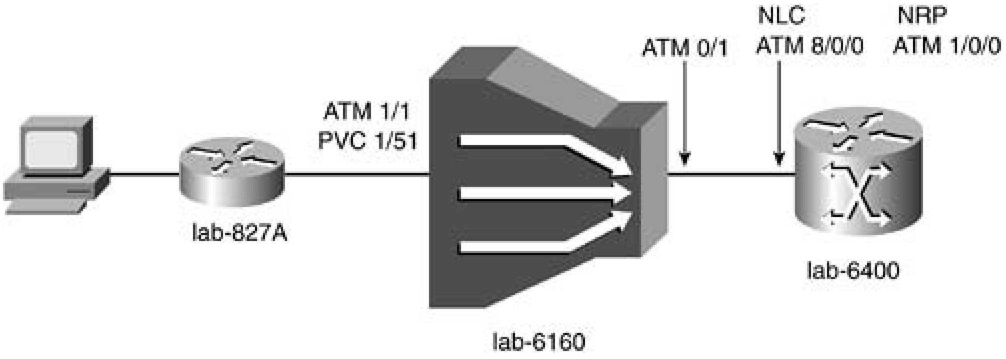
Scenarios

This section presents several examples of DSL access configurations. The scenarios cover the configuration for a DSLAM, a Cisco 6400 UAC NSP, a Cisco 6400 UAC NRP, and a DSL CPE - Cisco 827.

Scenario 8-1: Configuring IRB over DSL

In this scenario, you will configure the DSL solution to support data transport using IRB. When completed, the Cisco 827 should train up with the DSLAM, and you should be able to ping and access all normal network services from a client PC attached to the DSL CPE modem. Figure 8-9 illustrates how these devices are interconnected.

Figure 8-9. IRB Lab Scenario



In Example 8-13, the PVC is mapped from the Cisco 827 DSL connection (ATM1/1) to the DSLAM trunking port (ATM0/1).

Example 8-13. ATM PVC Configuration for the Cisco 6160 DSLAM

```
interface ATM1/1
 description IRB Architecture
 no ip address
 no atm ilmi-keepalive
 atm pvc 1 51  interface  ATM0/1 1 51
```

InExample 8-14, the PVC is configured from the DSLAM to the NSP and NRP. Interface ATM8/0/0 is the network line card, and interface ATM1/0/0 is the NRP.

Example 8-14. ATM PVC Configuration for the NSP

```
interface ATM8/0/0

 description OC3 connection to lab-6160

 no ip address

 no ip directed-broadcast

 no atm ilmi-keepalive

 atm pvc 1 51  interface  ATM1/0/0 1 51
```

A bridge group is configured for IP, and a BVI is created for IRB. The BVI becomes the default gateway for the remote device attached to the CPE equipment (which will be in subnet 10.1.121.0/24). A subinterface is created for a PVC to the NSP. (See the NSP configuration. The NSP maps this PVC to another PVC from the DSLAM, which maps to the subscriber PVC.) In this case, the 1/51 PVC is mapped across the NSP to the 6160. The subinterface is also put in the bridge group. Example 8-15 shows the IRB configuration for the NRP.

Example 8-15. IRB Configuration for the NRP

```
bridge irb

!

interface BVI1

ip address 10.1.121.1 255.255.255.0

no ip directed-broadcast

!

bridge 1 protocol ieee

 bridge 1 route ip

!

interface ATM0/0/0
```

```
no ip address
no ip directed-broadcast
!
interface ATM0/0/0.51 point-to-point
description IRB Configuration
no ip directed-broadcast
pvc 1/51
encapsulation aal5snap
!
bridge-group 1
```

Example 8-16 shows the bridging configuration for the DSL CPE.

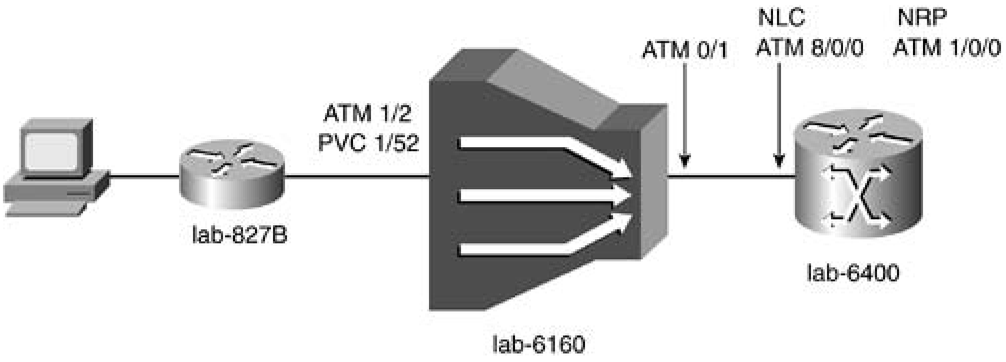
Example 8-16. RFC 1483 Bridging Configuration for the Cisco 827

```
hostname lab-827A
!
ip subnet-zero
no ip routing
!
interface Ethernet0
 ip address 10.1.121.2 255.255.255.0
 no ip directed-broadcast
 no ip mroute-cache
 bridge-group 1
!
interface atm0
 mac-address 0001.96a4.8fae   <--- MAC Address from Ethernet 0
 ip address 10.1.121.2 255.255.255.0
 no ip directed-broadcast
 no ip mroute-cache
 no atm ilmi-keepalive
 pvc 1/51
  encapsulation aal5snap
 !
 bundle-enable
 bridge-group 1
 hold-queue 224 in
!
ip classless
no ip http server
!
bridge 1 protocol ieee
```

Scenario 8-2: Configuring RBE over DSL

In this scenario, you will configure the DSL solution to support data transport using RBE. When completed, the Cisco 827 should train up with the DSLAM, and you should be able to ping and access all normal network services from a client PC attached to the DSL CPE modem. Figure 8-10 illustrates how these devices are interconnected.

Figure 8-10. RBE Scenario



InExample 8-17, the PVC is mapped from the Cisco 827 DSL connection (ATM1/2) to the DSLAM trunking port (ATM0/1).

Example 8-17. ATM PVC Configuration for the Cisco 6160 DSLAM

```
interface ATM1/2
 description RBE Architecture
 no ip address
 no atm ilmi-keepalive
 atm pvc 1 52  interface  ATM0/1 1 52
```

InExample 8-18, the PVC is configured from the DSLAM to the NSP and NRP. Interface ATM8/0/0 is the network line card, and interface ATM1/0/0 is the NRP.

Example 8-18. ATM PVC Configuration for the NSP

```
interface ATM8/0/0
 description OC3 connection to lab-6160
 no ip address
 no ip directed-broadcast
 no atm ilmi-keepalive
 atm pvc 1 52  interface  ATM1/0/0 1 52
```

Example 8-19 shows the RBE configuration for the NRP. You saw the configuration steps in the previous section.

Example 8-19. RBE Configuration for the NRP

```
interface Loopback1
```

```
 ip address 10.1.121.1 255.255.255.0
 no ip directed-broadcast
!
interface ATM0/0/0
no ip address
no ip directed-broadcast
!
interface ATM0/0/0.52 point-to-point
 description RBE Configuration
 ip unnumbered Loopback1
 atm route-bridged ip
 pvc 1/52
  encapsulation aal5snap
!
ip route 10.1.121.2 255.255.255.255 ATM0/0/0.52
```

Example 8-20 shows the bridging configuration for the DSL CPE. As you can see, the CPE configuration is the same when you configure the IRB over DSL.

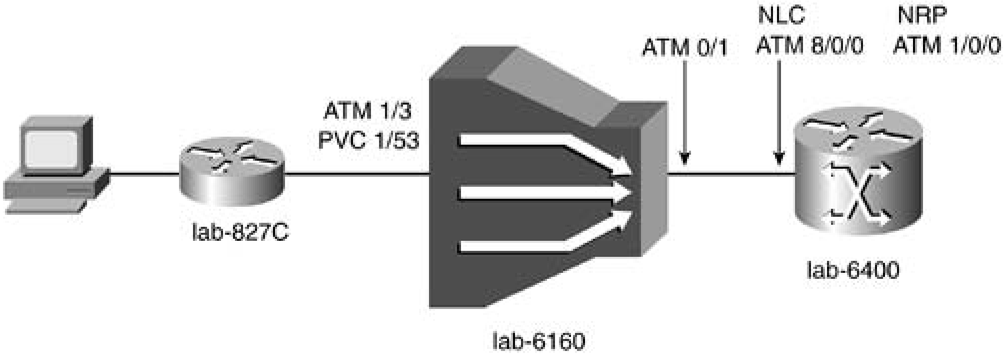
Example 8-20. RFC 1483 Bridging Configuration for the Cisco 827

```
hostname lab-827B
!
ip subnet-zero
no ip routing
!
interface Ethernet0
 ip address 10.1.121.2 255.255.255.0
 no ip directed-broadcast
 no ip mroute-cache
 bridge-group 1
!
interface atm0
 mac-address 0001.96a4.8fae
 ip address 10.1.121.2 255.255.255.0
 no ip directed-broadcast
 no ip mroute-cache
 no atm ilmi-keepalive
 pvc 1/52
  encapsulation aal5snap
 !
 bundle-enable
 bridge-group 1
 hold-queue 224 in
!
ip classless
no ip http server
!
bridge 1 protocol ieee
```

Scenario 8-3: Configuring PPPoA over DSL

In this scenario, you will configure the DSL solution to support data transport using PPPoA. When completed, the Cisco 827 should train up with the DSLAM, and you should be able to ping and access all normal network services from a client PC attached to the DSL CPE modem. Figure 8-11 illustrates how these devices are interconnected.

Figure 8-11. PPPoA Lab Scenario



InExample 8-21, the PVC is mapped from the Cisco 827 DSL connection (ATM1/3) to the DSLAM trunking port (ATM0/1).

Example 8-21. ATM PVC Configuration for the Cisco 6160 DSLAM

```
interface ATM1/3

 description PPPoA Architecture

 no ip address

 no atm ilmi-keepalive

 atm pvc 1 53  interface  ATM0/1 1 53
```

InExample 8-22, the PVC is configured from the DSLAM to the NSP and NRP. Interface ATM8/0/0 is the network line card, and interface ATM1/0/0 is the NRP.

Example 8-22. ATM PVC Configuration for the NSP

```
interface ATM8/0/0

 description OC3 connection to lab-6160

 no ip address

 no ip directed-broadcast

 no atm ilmi-keepalive

 atm pvc 1 53  interface  ATM1/0/0 1 53
```

Example 8-23 shows the PPPoA configuration for the NRP.

Example 8-23. PPPoA Configuration for the NRP

```
username cisco password 0 cisco
!
interface ATM0/0/0
no ip address
no ip directed-broadcast
!
interface ATM0/0/0.53 point-to-point
 description PPPoA Configuration
 pvc 1/53
  encapsulation aal5mux ppp Virtual-Template1
 !
interface Virtual-Template1
 description PPPoA
 ip unnumbered Ethernet0/0/0
 peer default ip address pool ccnp
 ppp authentication chap pap
```

Example 8-24 shows the PPPoA configuration for the DSL CPE.

Example 8-24. PPPoA Configuration for the Cisco 827

```
hostname lab-827C
!
```

```
ip subnet-zero
!
interface Ethernet0
 ip address 10.0.0.1 255.255.255.0
 no ip directed-broadcast
 no ip mroute-cache
!
interface ATM0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 no atm ilmi-keepalive
 pvc 1/53
  encapsulation aal5mux ppp dialer
  dialer pool-member 1
 !
!
interface Dialer1
 ip address negotiated
 no ip directed-broadcast
 encapsulation ppp
 dialer pool 1
 dialer-group 1
 ppp authentication chap callin
 ppp chap hostname cisco
 ppp chap password cisco
!
ip classless
```

```
!
dialer-list 1 protocol ip permit
```

When a PPP connection is made, a virtual interface is created, as shown in Example 8-25. The connection is authenticated with PAP/CHAP (using username "cisco" and password "cisco"). IP addresses are negotiated and handed out from the address pool named ccnp.

Example 8-25. Verifying the Virtual Interface

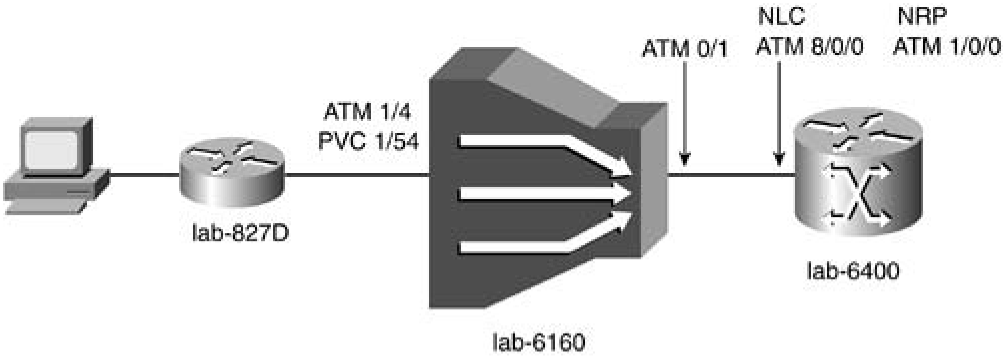
```
lab-6400NRP#show interface virtual-access 1
Virtual-Access1 is up, line protocol is up
  Hardware is Virtual Access interface
  Description: PPPoA
  Interface is unnumbered. Using address of Ethernet0/0/0 (10.1.1.190)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set
  Keepalive set (10 sec)
  DTR is pulsed for 5 seconds on reset
  LCP Open
  Open: IPCP
  Bound to ATM0/0/0.53 VCD: 3, VPI: 1, VCI: 53
  Cloned from virtual-template: 1
  Last input 00:00:03, output never, output hang never
  Last clearing of "show interface" counters 14:05:57
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
```

```
     10239 packets input, 141642 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     21626 packets output, 852074 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

Scenario 8-4: Configuring PPPoE over DSL

In this scenario, you will configure the DSL solution to support data transport using PPPoE. When completed, the Cisco 827 should train up with the DSLAM, and you should be able to ping and access all normal network services from a client PC attached to the DSL CPE modem. Figure 8-12 illustrates how these devices are interconnected.

Figure 8-12. PPPoE Lab Scenario



InExample 8-26, the PVC is mapped from the Cisco 827 DSL connection (ATM1/4) to the DSLAM trunking port (ATM0/1).

Example 8-26. ATM PVC Configuration for the Cisco 6160 DSLAM

```
interface ATM1/4
 description PPPoE Architecture
 no ip address
 no atm ilmi-keepalive

 atm pvc 1 54  interface  ATM0/1 1 54
```

InExample 8-27, the PVC is configured from the DSLAM to the NSP and NRP. Interface ATM8/0/0 is the network line card, and interface ATM1/0/0 is the NRP.

Example 8-27. ATM PVC Configuration for the NSP

```
interface ATM8/0/0

 description OC3 connection to lab-6160

 no ip address

 no ip directed-broadcast

 no atm ilmi-keepalive

 atm pvc 1 54  interface  ATM1/0/0 1 54
```

Example 8-28 shows the PPPoE configuration for the NRP.

Example 8-28. PPPoE Configuration for the NRP

```
username cisco password 0 cisco

!

vpdn enable

!

vpdn-group 1

 accept-dialin

  protocol pppoe

  virtual-template 1

interface ATM0/0/0
```

```
no ip address
no ip directed-broadcast
!
interface ATM0/0/0.54 point-to-point
 description LAB PPPoE Configuration
 pvc 1/54
  encapsulation aal5snap
  protocol pppoe
 !
interface Virtual-Template1
 description PPPoE
 ip unnumbered Ethernet0/0/0
 ip mtu 1492
 peer default ip address pool ccnp
 ppp authentication chap pap
```

For PPPoE over DSL, the DSL CPE is also configured for pure RFC 1483 bridging, as shown in Example 8-29.

Example 8-29. RFC 1483 Bridging Configuration for the Cisco 827

```
hostname lab-827D
!
ip subnet-zero
no ip routing
!
interface Ethernet0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 bridge-group 1
!
interface atm0
 mac-address 0001.96a4.8fae
 ip address 10.1.121.2 255.255.255.0
 no ip directed-broadcast
 no ip mroute-cache
 no atm ilmi-keepalive
 pvc 1/52
  encapsulation aal5snap
 !
 bundle-enable
 bridge-group 1
 hold-queue 224 in
!
ip classless
no ip http server
!
bridge 1 protocol ieee
```

When a PPP connection is made, a virtual interface is created, as shown in Example 8-30. The connection is authenticated with PAP/CHAP (using username "cisco" and password "cisco"). IP addresses are negotiated and handed out from the address pool named ccnp.

Example 8-30. Verifying the Virtual Interface

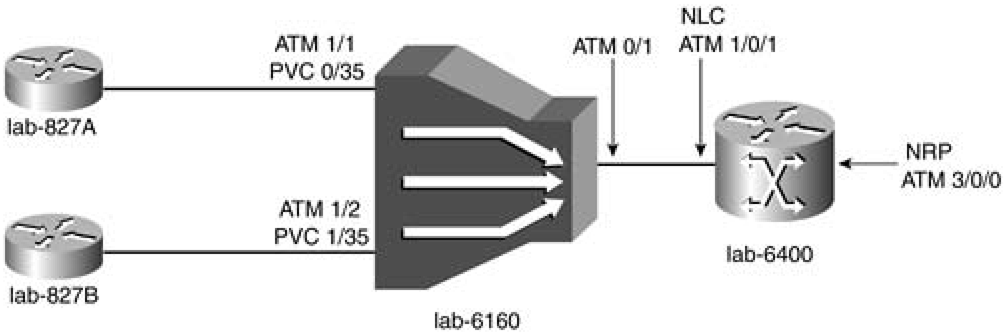
```
lab-6400NRP#show int Virtual-Access3
Virtual-Access3 is up, line protocol is up
```

```
Hardware is Virtual Access interface
Description: PPPoE
Interface is unnumbered. Using address of Ethernet0/0/0 (10.1.1.190)
MTU 1492 bytes, BW 100000 Kbit, DLY 100000 usec,
   reliability 255/255, txload 1/255, rxload 1/255
Encapsulation PPP, loopback not set
Keepalive set (10 sec)
DTR is pulsed for 5 seconds on reset
LCP Open
Open: IPCP
Bound to ATM0/0/0.54 VCD: 4, VPI: 1, VCI: 54
Cloned from virtual-template: 1
Last input 00:00:04, output never, output hang never
Last clearing of "show interface" counters 00:01:34
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
   40 packets input, 2923 bytes, 0 no buffer
   Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
   78 packets output, 6071 bytes, 0 underruns
   0 output errors, 0 collisions, 0 interface resets
   0 output buffer failures, 0 output buffers swapped out
   0 carrier transitions
```

Practical Exercise 8-1: PPPoA over DSL

In this practical exercise, both lab-827A and lab-827B are connected to the DSLAM, as shown in Figure 8-13. You need to create two different DSL profiles—premium and standard. Each of them has a different downstream and upstream speed. Assign a premium DSL profile to lab-827A and a standard DSL profile to lab-827B. In this exercise, you will configure local authentication. IP addresses are assigned to the DSL CPEs from the IP pool configured in the Cisco 6400.

Figure 8-13. Practical Exercise: PPPoA over DSL



Practical Exercise 8-1 Solution

Examples 8-31 through 8-35 show the PPPoA configurations for the DSL CPEs, Cisco 6160 DSLAM, and Cisco 6400.

Example 8-31. Configuration Output for lab-827A

```
lab-827A#show running-config

version 12.2

no service pad

service timestamps debug datetime msec

service timestamps log datetime msec

no service password-encryption

!

hostname lab-827A

!

ip subnet-zero

!

interface Ethernet0

 no ip address

 shutdown

 hold-queue 100 out

!

interface ATM0

 no ip address

 no atm ilmi-keepalive

 pvc 0/35

  encapsulation aal5mux ppp dialer

  dialer pool-member 1
 !
 dsl operating-mode auto
 dsl power-cutback 0
!
interface Dialer1
 ip address negotiated
 encapsulation ppp
 dialer pool 1
 ppp chap hostname cisco
 ppp chap password 0 cisco
!
ip classless
ip route 0.0.0.0 0.0.0.0 Dialer1
ip http server
!
!
!
call rsvp-sync
!
voice-port 1
!
voice-port 2
!
voice-port 3
!
voice-port 4
!
!
```

```
line con 0
 stopbits 1
line vty 0 4
 login
!
scheduler max-task-time 5000
end
```

Example 8-32. Configuration Output for lab-827B

```
lab-827B#show running-config
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname lab-827B
!
!
ip subnet-zero
!
!
!
!
!
interface Ethernet0
```

```
 no ip address
 shutdown
 hold-queue 100 out
!
interface ATM0
 no ip address
 no atm ilmi-keepalive
 pvc 0/35
  encapsulation aal5mux ppp dialer
  dialer pool-member 1
 !
 dsl operating-mode auto
!
interface Dialer1
 ip address negotiated
 encapsulation ppp
 dialer pool 1
 ppp chap hostname cisco
 ppp chap password 0 cisco
!
ip classless
ip route 0.0.0.0 0.0.0.0 Dialer1
ip http server
ip pim bidir-enable
!
!
!
!
```

```
line con 0

 stopbits 1

line vty 0 4

 login

!

scheduler max-task-time 5000

end
```

InExample 8-33, two DSL profiles, premium and standard, are defined. As you can see, each of them is configured with different downstream and upstream speeds.

Example 8-33. Configuration Output for lab-6160

```
lab-6160#show running-config

version 12.2

no service pad

service timestamps debug uptime

service timestamps log uptime

no service password-encryption

!

hostname lab-6160

!

slot 1 ATUC-4FLEXIDMT

slot 10 NI-2-155SM-DS3

!

!

dsl-profile default

!

dsl-profile premium
```

```
 dmt bitrate maximum fast downstream 8064 upstream 864
 dmt bitrate maximum interleaved downstream 0 upstream 0
!
dsl-profile standard
 dmt bitrate maximum fast downstream 6400 upstream 640
 dmt bitrate maximum interleaved downstream 0 upstream 0
!
network-clock-select 1 ATM0/1
redundancy
ip subnet-zero
!
!
no atm oam intercept end-to-end
atm address 47.0091.8100.0000.0030.96fe.db01.0030.96fe.db01.00
atm router pnni
 no aesa embedded-number left-justified
 node 1 level 56 lowest
  redistribute atm-static
!
!
!
!
interface ATM0/0
 no ip address
 atm maxvp-number 0
 atm maxvc-number 4096
 atm maxvci-bits 12
!
```

```
interface Ethernet0/0
 no ip address
 shutdown
!
interface ATM0/1
 no ip address
 no atm ilmi-keepalive
!
interface ATM0/2
 no ip address
 no atm ilmi-keepalive
!
interface ATM0/3
 no ip address
 no atm ilmi-keepalive
!
interface ATM1/1
 no ip address
 dsl profile premium
 no atm ilmi-keepalive
 atm pvc 0 35  interface  ATM0/1 1 35
!
interface ATM1/2
 no ip address
 dsl profile standard
 no atm ilmi-keepalive
 atm pvc 0 35  interface  ATM0/1 2 35
!
```

```
interface ATM1/3
 no ip address
 no atm ilmi-keepalive
!
interface ATM1/4
 no ip address
 no atm ilmi-keepalive
!
ip classless
no ip http server
ip pim bidir-enable
!
!
!
line con 0
line aux 0
line vty 0 4
!
end
```

Example 8-34. Configuration Output for lab-6400NSP

```
lab-6400NSP#show running-config
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
```

```
no service password-encryption
!
hostname lab-6400NSP
!
facility-alarm intake-temperature major 49
facility-alarm intake-temperature minor 40
facility-alarm core-temperature major 53
facility-alarm core-temperature minor 45
ip subnet-zero
!
ip cef
!
atm address 47.0091.8100.0000.0050.7359.3581.0050.7359.3581.00
atm router pnni
 no aesa embedded-number left-justified
 node 1 level 56 lowest
  redistribute atm-static
!
interface ATM0/0/0
 no ip address
 atm maxvp-number 0
!
interface Ethernet0/0/0
 no ip address
 bridge-group 1
!
interface ATM1/0/0
 no ip address
 no atm ilmi-keepalive
!
interface ATM1/0/1
 no ip address
 no atm ilmi-keepalive
!
interface ATM2/0/0
 no ip address
 no atm ilmi-keepalive
!
interface ATM2/0/1
 no ip address
 no atm ilmi-keepalive
!
interface ATM3/0/0
 no ip address
 no atm ilmi-keepalive
 atm pvc 1 35  interface  ATM1/0/1 1 35
 atm pvc 2 35  interface  ATM1/0/1 2 35
!
!
!
ip classless
ip http server
ip pim bidir-enable
!
line con 0
line 1 16
line aux 0
line vty 0 4
!
end
```

Example 8-35. Configuration Output for lab-6400NRP

```
lab-6400NRP#show running-config
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname lab-6400NRP
!
logging rate-limit console 10 except errors
no logging console
!
username cisco password 0 cisco
redundancy
 main-cpu
  auto-sync standard
 no secondary console enable
ip subnet-zero
!
!
!
```

```
!
!
!
interface Loopback1
 ip address 20.1.1.1 255.255.255.255
!
interface ATM0/0/0
 no ip address
 no atm ilmi-keepalive
 hold-queue 500 in
!
interface ATM0/0/0.135 point-to-point
 pvc 1/35
  encapsulation aal5mux ppp Virtual-Template1
 !
!
interface ATM0/0/0.235 point-to-point
 pvc 2/35
  encapsulation aal5mux ppp Virtual-Template1
 !
!
interface Ethernet0/0/1
 ip address negotiated
!
interface Ethernet0/0/0
 no ip address
 shutdown
!
```

```
interface FastEthernet0/0/0
 no ip address
 half-duplex
!
interface Virtual-Template1
 mtu 1460
 ip unnumbered Loopback1
 peer default ip address pool ccnp
 ppp authentication chap
!
ip local pool ccnp 20.1.1.2 20.1.1.10
ip classless
ip http server
!
!
!
!
!
line con 0
line aux 0
line vty 0 4
 login
!
end
```

Example 8-36 shows that lab-827A has successfully passed the PPP negotiation and authentication An IP address is assigned to the DSL connection.

Example 8-36. Output of show ip interface brief for lab-827A

```
lab-827A#show ip interface brief
Interface                IP-Address      OK? Method Status                Protocol
Ethernet0                unassigned      YES NVRAM  administratively down down
ATM0                     unassigned      YES NVRAM  up                    up
Dialer1                  20.1.1.3        YES BOOTP  up                    up
Virtual-Access1          unassigned      YES unset  up                    up
Virtual-Access2          unassigned      YES unset  up                    up
Virtual-Access3          unassigned      YES unset  up                    up
```

Example 8-37 shows that lab-827B has successfully passed the PPP negotiation and authentication An IP address is assigned to the DSL connection.

Example 8-37. Output of show ip interface brief for lab-827B

```
lab-827B#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protoc
Ethernet0                  unassigned      YES NVRAM  administratively down down
ATM0                       unassigned      YES NVRAM  up                    up
Dialer1                    20.1.1.2        YES BOOTP  up                    up
Virtual-Access1            unassigned      YES unset  up                    up
Virtual-Access2            unassigned      YES unset  up                    up
Virtual-Access3            unassigned      YES unset  up                    up
```

InExample 8-38, two virtual interfaces are cloned from the virtual template. They are served as Layer 3 termination for the DSL CPEs—lab-827A and lab-827B. Examples 8-39 and 8-40 show the details of two virtual interfaces.

Example 8-38. Output of show ip interface brief for lab-6400NRP

```
lab-6400NRP#show ip interface brief
Interface                IP-Address      OK? Method Status                Protocol
ATM0/0/0                 unassigned      YES NVRAM  up                    up
ATM0/0/0.135             unassigned      YES unset  up                    up
ATM0/0/0.235             unassigned      YES unset  up                    up
Ethernet0/0/1            unassigned      YES NVRAM  up                    up
Ethernet0/0/0            unassigned      YES NVRAM  administratively down down
FastEthernet0/0/0        unassigned      YES NVRAM  up                    up
Virtual-Access1          20.1.1.1        YES TFTP   up                    up
Virtual-Template1        20.1.1.1        YES TFTP   down                  down
Virtual-Access2          20.1.1.1        YES TFTP   up                    up
Loopback1                20.1.1.1        YES NVRAM  up                    up
```

Example 8-39. Verifying the Virtual Interface for lab-827A

```
lab-6400NRP#show interface virtual-access1
Virtual-Access1 is up, line protocol is up
  Hardware is Virtual Access interface
  Interface is unnumbered. Using address of Loopback1 (20.1.1.1)
  MTU 1460 bytes, BW 100000 Kbit, DLY 100000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set
  Keepalive set (10 sec)
  DTR is pulsed for 5 seconds on reset
  LCP Open
  Open: IPCP
  Bound to ATM0/0/0.135 VCD: 1, VPI: 1, VCI: 35
  Cloned from virtual-template: 1
  Last input 00:00:03, output never, output hang never
  Last clearing of "show interface" counters 00:12:13
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     100 packets input, 1841 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     200 packets output, 49806 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

Example 8-40. Verifying the Virtual Interface for lab-827B

```
lab-6400NRP#show interface virtual-access2
Virtual-Access2 is up, line protocol is up
  Hardware is Virtual Access interface
  Interface is unnumbered. Using address of Loopback1 (20.1.1.1)
  MTU 1460 bytes, BW 100000 Kbit, DLY 100000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set
  Keepalive set (10 sec)
  DTR is pulsed for 5 seconds on reset
  LCP Open
  Open: IPCP
  Bound to ATM0/0/0.235 VCD: 2, VPI: 2, VCI: 35
  Cloned from virtual-template: 1
  Last input 00:00:00, output never, output hang never
  Last clearing of "show interface" counters 00:12:51
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     107 packets input, 1499 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     210 packets output, 53645 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

Example 8-41 displays the DSL profile status for the default profile, premium profile, and standard profile. Keep in mind that the premium and standard profiles are created in this exercise. You can useshow dsl profile [*profile-name*] to display a specific profile, all ports to which the profile is currently attached, and those port settings.

Example 8-41. Output of show dsl profile

```
lab-6160#show dsl profile
dsl profile default:
      Link Traps Enabled: NO
      Alarms Enabled: NO
      ATM Payload Scrambling: Enabled
```

```
 DMT profile parameters
      Maximum Bitrates:
         Interleave Path:   downstream:   640 kb/s,   upstream:   128 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Minimum Bitrates:
         Interleave Path:   downstream:     0 kb/s,   upstream:     0 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Margin:               downstream:     6 dB,     upstream:     6 dB
      Interleaving Delay:   downstream: 16000 usecs,  upstream: 16000 usecs
      Check Bytes (FEC):
         Interleave Path:   downstream:    16,        upstream:    16
         Fast Path:         downstream:     0,        upstream:     0
      R-S Codeword Size:    downstream:  auto,        upstream:  auto
      Trellis Coding:         Disabled
      Overhead Framing:       Mode 3
      Operating Mode:         Automatic
      Training Mode:          Quick
      Minrate blocking:       Disabled
      SNR Monitoring:         Disabled
      Power Management Additional Margin:
                            downstream:     0 dB,     upstream:     0 dB

dsl profile premium:
      Link Traps Enabled: NO
      Alarms Enabled: NO
      ATM Payload Scrambling: Enabled
 DMT profile parameters
      Maximum Bitrates:
         Interleave Path:   downstream:     0 kb/s,   upstream:     0 kb/s
         Fast Path:         downstream:  8064 kb/s,   upstream:   864 kb/s
      Minimum Bitrates:
         Interleave Path:   downstream:     0 kb/s,   upstream:     0 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Margin:               downstream:     6 dB,     upstream:     6 dB
      Interleaving Delay:   downstream: 16000 usecs,  upstream: 16000 usecs
      Check Bytes (FEC):
         Interleave Path:   downstream:    16,        upstream:    16
         Fast Path:         downstream:     0,        upstream:     0
      R-S Codeword Size:    downstream:  auto,        upstream:  auto
      Trellis Coding:         Disabled
      Overhead Framing:       Mode 3
      Operating Mode:         Automatic
      Training Mode:          Quick
      Minrate blocking:       Disabled
      SNR Monitoring:         Disabled
      Power Management Additional Margin:
                            downstream:     0 dB,     upstream:     0 dB
dsl profile standard:
      Link Traps Enabled: NO
      Alarms Enabled: NO
      ATM Payload Scrambling: Enabled

 DMT profile parameters
      Maximum Bitrates:
         Interleave Path:   downstream:     0 kb/s,    upstream:     0 kb/s
         Fast Path:         downstream:  6400 kb/s,    upstream:   640 kb/s
      Minimum Bitrates:
         Interleave Path:   downstream:     0 kb/s,    upstream:     0 kb/s
         Fast Path:         downstream:     0 kb/s,    upstream:     0 kb/s
      Margin:               downstream:     6 dB,      upstream:     6 dB
      Interleaving Delay:   downstream: 16000 usecs,  upstream: 16000 usecs
      Check Bytes (FEC):
         Interleave Path:   downstream:    16,        upstream:    16
         Fast Path:         downstream:     0,        upstream:     0
      R-S Codeword Size:    downstream:  auto,        upstream:  auto
      Trellis Coding:         Disabled
      Overhead Framing:       Mode 3
      Operating Mode:         Automatic
      Training Mode:          Quick
      Minrate blocking:       Disabled
      SNR Monitoring:         Disabled
      Power Management Additional Margin:
                            downstream:     0 dB,      upstream:     0 dB
```

lab-827A is patched to port 1/1, and lab-827B is patched to port 1/2. Example 8-42 displays the status of the DSL subscriber ports on a 6160 chassis.

Example 8-42. Using the show dsl status Command to Display the Status of DSL Ports

```
lab-6160#show dsl status

Subtend Node ID: 0
```

```
                        DOWNSTREAM   UPSTREAM   SUBSCRIBER     CIRCUIT ID
    NAME   ADMIN/OPER         (Kb)       (Kb)   (truncated)    (truncated)
    ----   ----------     --------   --------   -----------    -----------
  ATM1/1      UP/  UP         8064        864
  ATM1/2      UP/  UP         6400        640
  ATM1/3      UP/DOWN            0          0
  ATM1/4      UP/DOWN            0          0
```

Theshow dsl interface atm*slot#/port#* command allows you to display DSL, DMT, and ATM statu for a port, as shown in Example 8-43.

Example 8-43. Displaying DSL, DMT, and ATM Status for Port 1/1

```
lab-6160#show dsl interface atm 1/1
Port Status:
   Subscriber Name:         Circuit ID:
   IOS admin: UP     oper: UP     Card status: ATUC-4FLEXIDMT
   Last Change: 00 days, 00 hrs, 50 min, 28 sec No. of changes: 12
   Line Status: TRAINED
   Test Mode: NONE

ADSL Chipset Self-Test: NONE
CO Modem Firmware Version: 5.38
Configured:
      DMT Profile Name: premium
      Link Traps Enabled: NO
      Alarms Enabled: NO
      ATM Payload Scrambling: Enabled
```

```
 DMT profile parameters
      Maximum Bitrates:
         Interleave Path:   downstream:     0 kb/s,   upstream:     0 kb/s
         Fast Path:         downstream:  8064 kb/s,   upstream:   864 kb/s
      Minimum Bitrates:
         Interleave Path:   downstream:     0 kb/s,   upstream:     0 kb/s
         Fast Path:         downstream:     0 kb/s,   upstream:     0 kb/s
      Margin:               downstream:     6 dB,     upstream:     6 dB
      Interleaving Delay:   downstream: 16000 usecs,  upstream: 16000 usecs
      Check Bytes (FEC):
         Interleave Path:   downstream:    16,        upstream:    16
         Fast Path:         downstream:     0,        upstream:     0
      R-S Codeword Size:    downstream:  auto,        upstream:  auto
      Trellis Coding:         Disabled
      Overhead Framing:       Mode 3
      Operating Mode:         Automatic
      Training Mode:          Quick
      Minrate blocking:       Disabled
      SNR Monitoring:         Disabled
      Power Management Additional Margin:
                            downstream:     0 dB,     upstream:     0 dB
Status:
    Bitrates:
         Interleave Path:   downstream:     0 kb/s,   upstream:     0 kb/s
         Fast Path:         downstream:  8064 kb/s,   upstream:   864 kb/s
    Attainable Aggregate
    Bitrates:
```

```
                            downstream:  9440 kb/s,   upstream:   928 kb/s
    Margin:                 downstream:    12 dB,     upstream:    11 dB
    Attenuation:            downstream:     1 dB,     upstream:     2 dB
    Interleave Delay:       downstream:     0 usecs,  upstream:     0 usecs
    Transmit Power:         downstream:   9.5 dB,        upstream:  12.1 dB
    Check Bytes (FEC):
         Interleave Path:   downstream:     0,        upstream:     0
         Fast Path:         downstream:     0,        upstream:     0
    R-S Codeword Size:      downstream:     1,        upstream:     1
    Trellis Coding:           In Use
    Overhead Framing:         Mode 3
    Line Fault:               NONE
    Operating Mode:           ITU G dmt Issue 1
    Line Type:                Fast Only
    Alarms:
       status:                NONE
ATM Statistics:
   Interleaved-Path Counters:
      Cells:                downstream:         20     upstream:        154
      HEC errors:           downstream:          0     upstream:          2
      LOCD events:          near end:            1     far end:           0
   Fast-Path Counters:
      Cells:                downstream:       1729     upstream:        660
      HEC errors:           downstream:          1     upstream:          1
      LOCD events:          near end:            1     far end:           0
DSL Statistics:
   Init Events:             4
   Far End LPR Events:      0
   Transmitted Superframes: near end:    161749170     far end:           0
   Received Superframes:    near end:    161748691     far end:           0
   Corrected Superframes:   near end:          176     far end:           0
   Uncorrected Superframes: near end:          369     far end:           1
   LOS Events:              near end:            2     far end:           0
   LOF/RFI Events:          near end:            0     far end:           0
   ES Events:               near end:           10     far end:           1
CPE Info:
   Version Number:            0
   Vendor ID:                 34
```

Practical Exercise 8-2: RFC 1483 Bridging over DSL

In this practical exercise, lab-827A and lab-827B are connected to the DSLAM and will be configured using RFC 1483 bridging, as shown in Figure 8-13. DSLAM and NSP configuration remain the same as in the previous exercise. For this exercise, you will assign the ATM interface of the CPEs and subinterfaces of the NRP to Bridge group 1. You will see the configuration output as well as some useful commands to verify the bridging configuration.

Practical Exercise 8-2 Solution

Examples 8-44 and 8-45 show the bridging configurations of both DSL CPE devices.

Example 8-44. Configuration Output for lab-827A

```
lab-827A#show running-config
Building configuration...
Current configuration : 763 bytes
!
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname lab-827A
!
ip subnet-zero
no ip routing
!
interface Ethernet0
 ip address 10.2.2.2 255.255.255.0
 no ip route-cache
 bridge-group 1
 hold-queue 100 out
!
interface ATM0
```

```
 mac-address 0001.96a4.84ac
 ip address 10.2.2.2 255.255.255.0
 no ip route-cache
 no atm ilmi-keepalive
 pvc 0/35
  encapsulation aal5snap
 !
 dsl operating-mode auto
 dsl power-cutback 0
 bridge-group 1
!
ip classless
ip http server
!
bridge 1 protocol ieee
call rsvp-sync
!
line con 0
 stopbits 1
line vty 0 4
!
scheduler max-task-time 5000
end
```

Example 8-45. Configuration Output for lab-827B

```
lab-827B#show running-config
```

```
Building configuration...
Current configuration : 670 bytes
!
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname lab-827B
!
no logging console
!
ip subnet-zero
no ip routing
!
interface Ethernet0
 ip address 10.2.2.3 255.255.255.0
 no ip route-cache
 bridge-group 1
 hold-queue 100 out
!
interface ATM0
 mac-address 0001.96a4.8fae
 ip address 10.2.2.3 255.255.255.0
 no ip route-cache
 no atm ilmi-keepalive
 pvc 0/35
```

```
  encapsulation aal5snap
 !
 dsl operating-mode auto
 bridge-group 1
!
ip classless
ip http server
ip pim bidir-enable
!
bridge 1 protocol ieee
!
line con 0
 stopbits 1
line vty 0 4
!
scheduler max-task-time 5000
end
```

Example 8-46 shows the bridging configuration of the Cisco 6400.

Example 8-46. Configuration Output for lab-6400NRP

```
lab-6400NRP#show running-config
Building configuration...
Current configuration : 907 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime

no service password-encryption

!

hostname lab-6400NRP

!

logging rate-limit console 10 except errors

no logging console

!

redundancy

 main-cpu

  auto-sync standard

 no secondary console enable

ip subnet-zero

!

bridge irb

!

interface ATM0/0/0

 no ip address

 no atm ilmi-keepalive

 hold-queue 500 in

!

interface ATM0/0/0.135 point-to-point

 pvc 1/35

  encapsulation aal5snap

 !

 bridge-group 1

!

interface ATM0/0/0.235 point-to-point
```

```
 pvc 2/35
  encapsulation aal5snap
 !
 bridge-group 1
!
interface Ethernet0/0/1
 ip address negotiated
!
interface Ethernet0/0/0
 no ip address
 shutdown
!
interface FastEthernet0/0/0
 no ip address
 half-duplex
!
interface BVI1
 ip address 10.2.2.1 255.255.255.0
!
ip classless
ip http server
!
!
!
!
bridge 1 protocol ieee
 bridge 1 route ip
!
```

```
line con 0
line aux 0
line vty 0 4
!
end
```

Example 8-47 shows that Bridge group 1 is running the IEEE Spanning Tree Protocol.

Example 8-47. Displaying the Spanning Tree Protocol (IEEE)

```
lab-6400NRP#show spanning-tree 1
 Bridge group 1 is executing the ieee compatible Spanning Tree protocol
  Bridge Identifier has priority 32768, address 0000.0c7f.70fc
  Configured hello time 2, max age 20, forward delay 15
  We are the root of the spanning tree
  Topology change flag not set, detected flag not set
  Number of topology changes 4 last change occurred 00:35:16 ago
          from ATM0/0/0.235
  Times:  hold 1, topology change 35, notification 2
          hello 2, max age 20, forward delay 15
  Timers: hello 0, topology change 0, notification 0, aging 300
 Port 6 (ATM0/0/0.135) of Bridge group 1 is forwarding
   Port path cost 14, Port priority 128, Port Identifier 128.6.
   Designated root has priority 32768, address 0000.0c7f.70fc
   Designated bridge has priority 32768, address 0000.0c7f.70fc
   Designated port id is 128.6, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   BPDU: sent 1663, received 2
 Port 8 (ATM0/0/0.235) of Bridge group 1 is forwarding
   Port path cost 14, Port priority 128, Port Identifier 128.8.
   Designated root has priority 32768, address 0000.0c7f.70fc
   Designated bridge has priority 32768, address 0000.0c7f.70fc
   Designated port id is 128.8, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   BPDU: sent 1527, received 1
```

Example 8-48 shows the IP and MAC addresses of lab-827A and lab-827B. show arp is a useful command to verify whether bridging is configured properly.

Example 8-48. Displaying ARP Information

```
lab-6400NRP#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.2.2.2               31   0001.96a4.84ac  ARPA   BVI1
Internet  10.2.2.3               31   0001.96a4.8fae  ARPA   BVI1
Internet  10.2.2.1                -   0050.7359.35a6  ARPA   BVI1
```

Example 8-49 illustrates that both subinterfaces are in the same bridge group (Bridge group 1), and traffic is passed among them. show bridge is another useful command to debug RFC 1483 bridging.

Example 8-49. Displaying Classes of Entries in the Bridge Forwarding Database

```
lab-6400NRP#show bridge verbose
Total of 300 station blocks, 298 free
```

```
Codes: P - permanent, S - self

BG Hash      Address      Action  Interface      VC    Age   RX count   TX count

 1 21/0   0001.96a4.8fae forward  ATM0/0/0.235    2      0          5          5

 1 28/0   0001.96a4.84ac forward  ATM0/0/0.135    1      0        100        100

Flood ports (BG 1)           RX count    TX count

ATM0/0/0.135                        0           0

ATM0/0/0.235                        0           0
```

Summary

This chapter covered ADSL technology, Cisco DSL hardware components, and the configuration of various DSL access architectures, such as IRB, RBE, PPPoA, and PPPoE. Keep in mind that each DSL access architecture has its advantages and disadvantages. You should further research these architectures to discover the best implementation for your DSL network environment.

Table 8-3 summarizes the commands used in this chapter.

Table 8-3. Summary of Commands Used in This Chapter

Command	Description
slot*slot# cardtype*	Configures a slot for a specific card type.
dsl-profile*profile-name*	Creates a DSL profile.
dmt bitrate max interleaved downstream*dmt-bitrate*upstream *dmt-bitrate*	Sets the maximum and minimum allowed bit rates for the fast-path and interleaved-path profile parameters.
dmt margin downstream*dmt-margin*upstream*dmt-margin*	Sets the upstream and downstream SNR DMT margins.
dmt check-bytes interleaved downstream*bytes*upstream*bytes*	Sets the upstream and downstream check bytes.
dmt interleaving-delay downstream*delay-in-µsecs* upstream*delay-in-µsecs*	Sets the interleaving delay parameter.
dmt training-mode {*standard* | *quick*}	Sets the training mode in a DMT profile.
bridge irb	Enables IRB.
bridge*bridge-group*protocol {ieee | dec}	Specifies the bridge protocol to define the type of Spanning Tree Protocol.
bridge*bridge-group*route*protocol*	Specifies a protocol to be routed in a bridge group.
bridge-group*bridge-group*	Assigns a network interface to a bridge group.
interface bvi*bridge-group*	Enables a bridge group virtual interface.
atm route-bridged ip	RBE command. Typically used to associate with an interface.
username*name*password*secret*	Configures a username and password for local authentication.
encapsulation aal5mux ppp Virtual-Template*number*	Configures PPPoA encapsulation and associates a virtual template with it.
interface virtual-template*number*	Creates a virtual template interface.

ip unnumbered*interface-name-number*	Conserves IP addresses by configuring the interface as unnumbered, and assigns the IP address of the interface type you want to leverage.
ip local pool*name begin-ip-address-range* [*end-ip-address-range*]	Creates the local IP address pool.
peer default ip address pool *poolname*	Specifies the pool for the interface to use.
ppp authentication {chap | pap | chap pap | pap chap} [if-needed] {default | *list-name*} [callin]	Enables CHAP or PAP authentication on the interface.
ip cef	Enables Cisco Express Forwarding switching.
vpdn enable	Enables VPDN configuration.
vpdn-group*number* accept-dialin protocol pppoe virtual-template*template-number*	Configures a VPDN group to accept the dial-in and to be used to establish PPPoE sessions. Specifies the virtual template that will be used to clone virtual-access interfaces.
ip mtu*bytes*	Sets the MTU size of IP packets sent on an interface.
show dsl profile	Displays the DSL profile you changed.
show dsl status	Displays the status of the DSL subscriber ports on a chassis.
show dsl interface atm*slot*/*port*	Shows the status of a DSL port.
show spanning-tree*bridge-group*	Displays information on which Spanning Tree Protocol is running.
show arp	Displays the entries in the ARP table.
show bridge group [verbose]	Displays the status of each bridge group in detail.

Review Questions

1: Which of the following modulation methods is not used for ADSL technology?

A. CAP

B. 2B1Q

C. DMT-2

D. G.lite

2: RFC 1483 when implemented is __________.

A. Bridged

B. Routed

C. Decrypted

D. Encrypted

3: PPPoA when implemented is __________.

A. Bridged

B. Routed

C. Decrypted

D. Encrypted

4: Which of the following interferences degrades DSL services?

A. Impedance changes

B. Bridged taps

C. Crosstalk

D. Impulse hits

E. All of the above

5: What is the function of the POTS splitter?

A. It separates low and high frequencies.

B. It manages ADSL signaling.

C. It generates ringing voltage.

D. It boosts the ADSL signal.

6: The DSL interface on a Cisco 827 is __________.

A. An FDDI interface

B. A Frame Relay interface

C. A serial interface

D. An ATM interface

7: With PPP over ATM, __________. (Choose all that apply.)

A. MAC frames are encapsulated into ATM cells

B. UDP frames are encapsulated using RFC 1483

C. IP packets are encapsulated into PPP frames and then into ATM cells

D. IP packets are encrypted

8: With RFC 1483 bridging, __________.

A. MAC frames are passed across the bridge after LLC/SNAP information is appended

B. IP frames are passed across the bridge unchanged

C. MAC frames are passed across the bridge unchanged

D. IP packets are encrypted

9: Which of the following cards in the Cisco 6400 can be used for Layer 3 packet services?

A. NSP

B. NLC

C. NRP

D. NI-2

10: Which of the following is part of PPPoA configuration?

A. encapsulation aal5mux ppp Virtual-Template 1

B. encapsulation aal5snap

C. atm route-bridged ip

D. bridge 1 protocol ieee

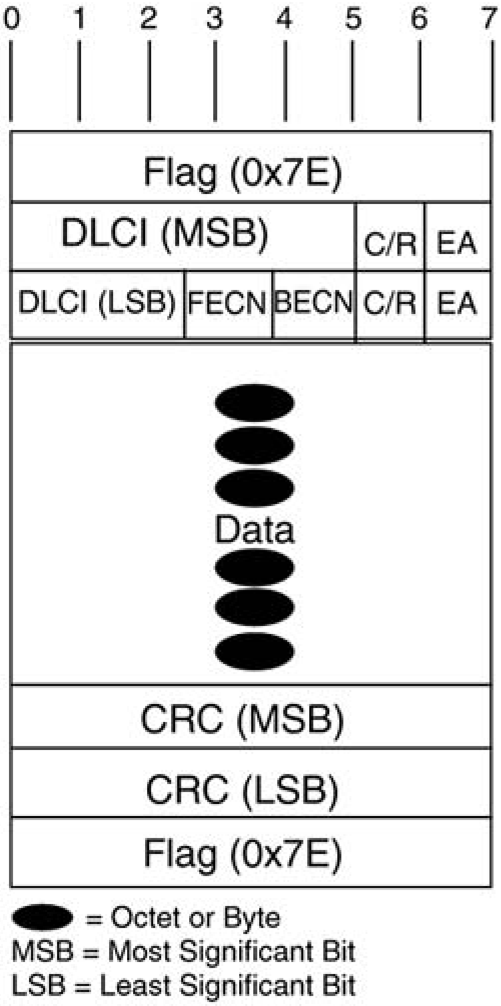
Chapter 9. Frame Relay Connectivity and Traffic Flow Control

This chapter looks at the configuration of Frame Relay and the different traffic flow control options. This chapter covers many topics related to Frame Relay:

Frame Relay Background

Before you begin your adventure with Frame Relay, you need to understand what it is. *Frame Relay* is an industry-standard switched data link layer protocol operating at the physical and data link layers of the OSI model. It can handle multiple virtual circuits (VCs) between Frame Relay-capable devices. Figure 9-1 illustrates an American National Standards Institute (ANSI) T1.618 Frame Relay frame.

Figure 9-1. ANSI T1.618 Frame Relay Format



The fields of the Frame Relay frame are as follows:

- Flag— Used to identify the beginning and end of a frame.
- Data-link connection identifier (DLCI)— Identifies the path through the network to the destination.

- Command/response (C/R)— Not generally used.
- Extended address (EA)— Identifies whether the header octet is followed by another header octet. A value of 0 means that another octet follows. 1 identifies the last octet.
- Forward explicit congestion notification (FECN)— Used to inform the connected devices of congestion in the network.
- Backward explicit congestion notification (BECN)— Used to inform the connected devices of congestion in the network.
- Discard eligible (DE)— Used to identify a packet that is eligible for discard during congestion.
- Data— Can be used to carry any type of information.
- Cyclic redundancy check (CRC)— Used to detect transmission errors and cover the header and data fields.

The 10-bit DLCI value is a logical number in the range of 16 to 107 used to identify the logical connection or permanent virtual circuit (PVC) that will be multiplexed into the physical circuit. The DLCI has significance only between your customer premises equipment (CPE) and your provider's Frame Relay switch. Because the DLCI is used to differentiate between different conversations on the same physical circuit, you can think of it as the heart of the Frame Relay header. Without it, your Frame Relay access device (FRAD) could not identify the different data streams passing through it.

Frame Relay provides you with a packet-switching data communications capability that is used across a data network interface to identify how the traffic will be formatted between your devices (such as routers, switches, multiplexers, and concentrators) and your service provider's network equipment (such as Frame Relay switching nodes). You need to know a couple terms used in Frame Relay. Your devices are often called data terminal equipment (DTE), and your service provider's network equipment is often called data circuit-terminating equipment (DCE).

As the interface between the DTE and DCE, Frame Relay must provide a technique that can statistically multiplex many logical data conversations over your single physical transmission link. If you are familiar with systems that use only time-division multiplexing (TDM) techniques to support multiple data streams, this technique might seem alien to you. Because Frame Relay was conceived to replace less-efficient protocols, its statistical multiplexing provides more flexible and efficient use of available bandwidth than a traditional TDM circuit. You should be aware that you can run Frame Relay on top of the channels provided by a TDM circuit, or you can run Frame Relay without any TDM techniques.

Frame Relay is a high-performance WAN protocol. It is standardized in the U.S. as an ANSI standard and internationally as an International Telecommunication Union Telecommunication Standardization Sector (ITU-T) standard. It operates at the physical and data link layers of the OSI reference model, much like Ethernet and Token Ring.

Frame Relay was created to develop the next-generation protocol to replace X.25 that would be carried across an ISDN interface, but it has been adapted to operate successfully over a wide variety of other network interfaces as well. Originally, WAN technology was developed to operate over low-quality transmission lines. Frame Relay can exploit the recent advances in WAN transmission technology. Because the earlier transmission lines were predominately analog transmission facilities, protocols such as X.25 were overengineered with extensive error checking and correction techniques to combat the quality of the communications across copper transmission lines. Although Frame Relay does not implement error checking, it does frame error checking and sends any error information to upper-layer protocols for any necessary actions, such as a TCP retransmission.

Today's links are much more reliable, often running across fiber media/digital transmission links. Because of this, Frame Relay can leave error detection and correction up to the higher protocol layers. Frame Relay does include a CRC algorithm for use in detecting corrupted bits so that the data can be discarded, but it does not include any protocol mechanisms for correcting bad data.

Frame Relay does not need to provide the explicit, per-VC flow control that X.25 implements. In its place, Frame Relay uses a simple congestion-notification mechanism that allows a network to inform a FRAD that the network resources are close to a congested state. This notification can also be used to alert the higher-layer protocols that flow control might be needed.

By using Frame Relay, you can reduce your network complexity and simplify your network architecture by supporting the three-tiered network model of core, distribution, and access layers. Frame Relay supports many different topologies for the placement of your network equipment, including full, partial, and hybrid meshing.

Frame Relay Terminology

Before continuing with the discussion of Frame Relay, you should take a moment to familiarize yourself with the terms listed in Table 9-1. You will see these terms throughout this chapter.

Table 9-1. Frame Relay Key Technical Terms

Acronym	Definition
Bc	Committed burst rate
Be	Excess burst rate
BECN	Backward explicit congestion notification
CIR	Committed information rate
DCE	Data communications equipment
DE	Discard eligible
DLCI	Data-link connection identifier
DTE	Data terminal equipment
FECN	Forward explicit congestion notification
Frame Relay Data Rate	Identifies the maximum, or peak, Frame Relay data rate. This is computed using the following formula: (Bc + Be) / Be * CIR.
LMI	Local Management Interface
MaxR	Maximum data rate
NNI	Network-to-Network Interface
PVC	Permanent virtual circuit
SVC	Switched virtual circuit
UNI	User-Network Interface
0-CIR service providers (SPs)	Some service providers offer a 0 CIR. Not exceeding CIR in this case means that traffic is not sent across the line. This is ideal for voice implementations, but Service Level Agreements (SLAs) must be negotiated with your SP to ensure that good quality is maintained across your circuit.

Frame Relay Devices

All Frame Relay devices that can attach to a Frame Relay WAN fall into only two general categories:

- DTE
- DCE

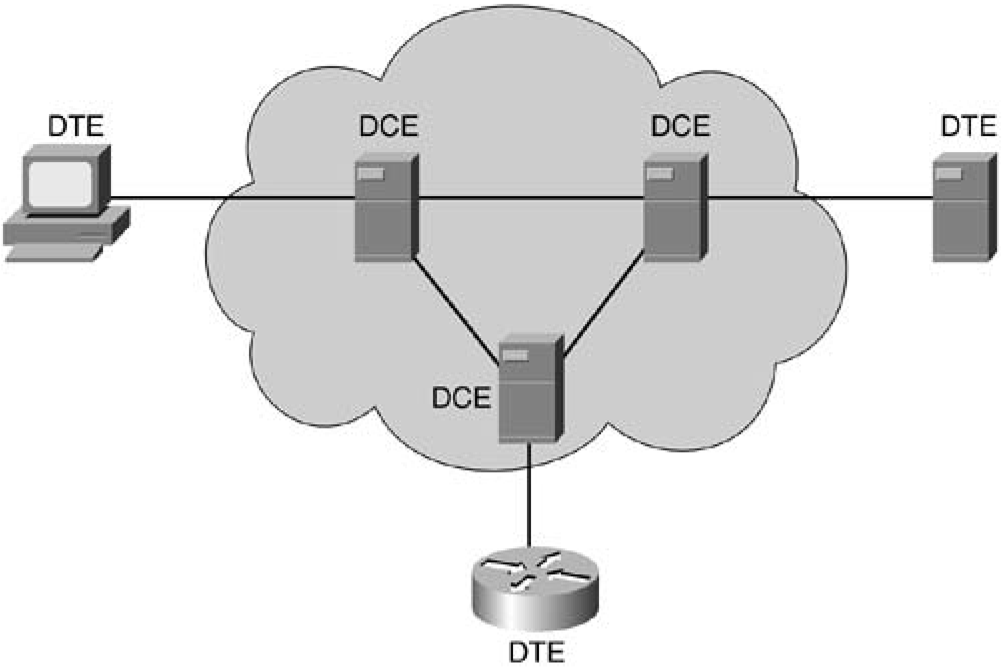
DTE

DTE is commonly the terminating equipment for a specific network that communicates directly with an end user or network. DTE typically is located on the customer premises, usually close to the SP's demarcation point. DTE usually is owned and operated by the customer.

DCE

DCE provides clocking and switching services in a network. The DCE converts user data from your DTE into an acceptable form for the WAN service facility. DCEs are usually the carrier-owned internetworking devices that are responsible for the transmission of data in the WAN. Figure 9-2 illustrates the relationship between a DCE and DTE.

Figure 9-2. DTE/DCE Relationship



Relationship Between DTE and DCE

The connection between the DTE device and the DCE device exists as a physical layer component, but it also contains a link layer component. The link's physical layer component defines the specifications used to connect the devices. The link's link layer component specifies how the connection is established between the DTE device and the DCE device.

The DTE/DCE interface is typically used to identify the boundary of responsibility for the traffic passing between you and your service provider. The physical standards used to specify the DTE/DCE interface include EIA/TIA-232, X.21, EIA/TIA-449, V.24, V.35, and HSSI (Cisco-proprietary).

Frame Relay Topologies

One of the first items you need to consider when designing a Frame Relay network, or any type of regional WAN, is how the connectivity will be laid out. When you are considering your Frame Relay for your choice in WAN mediums, you can choose from three basic design approaches:

- Star topology— A topology in which endpoints on a network are connected to a common central switch by point-to-point links. The star topology's advantages include simplified management and minimized tariff costs. Unfortunately, its disadvantages are considerable. For example, the core router represents a single point of failure and limits overall performance for access to your backbone resources, because each end device arrives through a single physical connection. Another disadvantage is that a star topology is not scalable.
- Full-mesh topology— A topology in which devices are organized in a mesh, with each network node having either a physical circuit or a virtual circuit connecting it to every other network node. The full-mesh topology offers some advantages over the star topology, such as a high level of redundancy and support for all network protocols. One disadvantage is the large number of virtual circuits required (one for every connection between routers), resulting in higher costs. Other disadvantages are replication of a large number of packets/broadcasts and the problems associated with it, and the complexity of configuration resulting from multicast capabilities in nonbroadcast environments.
- Partial-mesh topology— A topology in which devices are organized in a mesh. Some network nodes are organized in a full mesh, but others are connected to only one or two other nodes in a network. When you combine the full-mesh topology with the star topology, you can enjoy the advantages offered by both topologies for your network environment, including improved fault tolerance, without sacrificing performance and management problems. Several forms of partial-mesh topologies exist. They are considered to provide the best balance for regional topologies in terms of the number of virtual circuits, redundancy, and performance.

As you can see, each of these topologies has its advantages and disadvantages. You should consider these in your overall network design.

Frame Relay Virtual Circuits

One reason for Frame Relay's popularity is its capability to logically create multiple connection-oriented data link layer communication paths between two devices across a single physical interface. These VCs provide you with a bidirectional communications path that can exist between a single pair of equipment, commonly called a point-to-point connection, or between multiple pairs of equipment, also known as a partial or full mesh. Each of these VCs is identified by a unique data link connection identifier (DLCI) that differentiates the communications between different devices.

VCs can be mapped across any service provider's Frame Relay network without regard for the number of hops the connection will cross. A VC is not limited to three devices, two DTEs, and a DCE when traveling from source to destination. Just remember that each hop adds to your circuit's overall delay because of the processing that each device needs to do to read the packet and send the packet toward its destination.

VCs can be divided into two separate categorizes—switched virtual circuits (SVCs) or permanent virtual circuits (PVCs).

SVCs

The SVC gives you a way to automatically create temporary connections between DTE devices in the Frame Relay network that can be used in on-demand situations, such as those requiring only sporadic data transfer. An SVC's communication component consists of the following four operational states:

- Call setup— Indicates that the establishment of the SVC between two Frame Relay DTE devices is currently being negotiated.
- Data transfer— Indicates that data is being transmitted between DTE devices over an SVC.
- Idle— Indicates that the SVC between DTE devices is still active, but no data is currently being transferred.
- Call termination— Indicates that the SVC between DTE devices is being terminated.

NOTE

When an SVC remains in an idle state for a defined period of time, the SVC can be torn down and the call terminated.

After the termination of an SVC is complete, if additional data needs to be transmitted between the DTE devices, a new SVC is established. Cisco devices use the same signaling protocols used by ISDN to establish, maintain, and terminate SVCs.

PVCs

The PVC, unlike the SVC, establishes a permanent connection between your DTE devices. This type of circuit is typically used for frequent and consistent data transfers across the Frame Relay network. Because PVC establishment does not require call setup or termination, it is always up. PVCs have only two operational states:

- Data transfer— Indicates that data is currently being transmitted between the DTE devices over the PVC.
- Idle— Indicates that the connection between DTE devices is active, but no data is currently being transferred between DTE devices.

NOTE

Because a PVC connection has no call setup or termination, it is not terminated under any circumstances when in an idle state, unlike the SVC idle state. SVCs encounter startup delays after an idle period.

As soon as the PVC is established, your DTE devices may transfer data whenever they are ready, without the delay associated with the establishment of an SVC.

Frame Relay Configuration Tasks

The actual configuration of Frame Relay when you use IOS is fairly simple. Frame Relay requires that you configure only two items, assuming that Inverse ARP is used, to establish a connection and start passing traffic. The tasks described in the following sections are required for Frame Relay to function.

Enabling Frame Relay Encapsulation

The first step in configuring Frame Relay on your FRAD is to enable Frame Relay encapsulation on the interface that you will use for the connection.

You can configure Frame Relay to support encapsulation of all protocols that conform to RFC 1490 to create interoperability between multiple vendors. You can use the Internet Engineering Task Force (IETF) form of Frame Relay encapsulation if your FRAD is connected to another vendor's equipment across a Frame Relay network. You can use IETF encapsulation on the interface level or on a per-VC basis.

One optional item often overlooked is the fact that you must shut down the interface before changing encapsulation types. By doing this, you ensure that the interface is reset and is using the new encapsulation type.

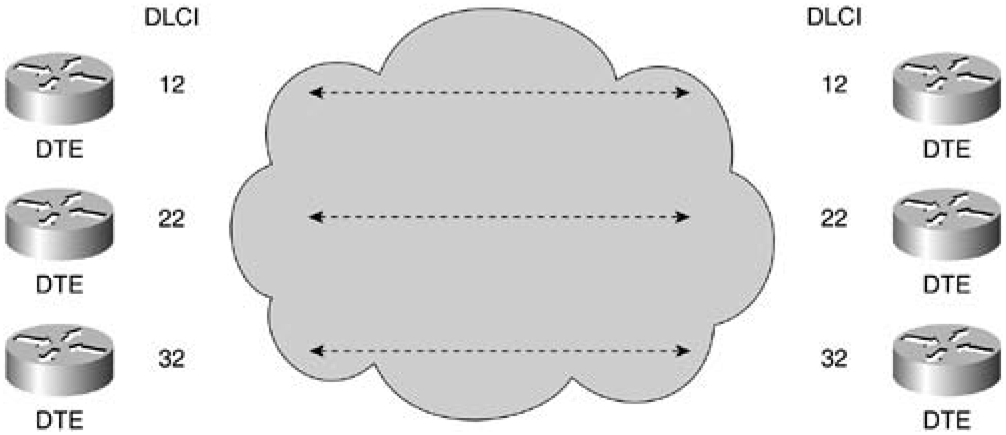
DLCI

You use the DLCI to differentiate the Frame Relay VCs from each other. The DLCI value is usually assigned by the service provider of your Frame Relay circuit.

Frame Relay DLCIs have only local significance to the DTE/DCE pair that they are configured on, which means that their values need only be unique in the LAN. Because of this, any Frame Relay DLCIs may be reused throughout the WAN.

Figure 9-3 illustrates how two different DTE devices can be assigned the same DLCI value within one Frame Relay WAN.

Figure 9-3. Duplicate DLCIs on Each End of a VC



Frame Relay Signaling

Frame Relay was not designed to include a built-in mechanism to address network outages. Instead, the Local Management Interface (LMI) signaling protocol was developed to exchange keepalives and to pass administrative information, such as the addition, deletion, or failure of PVCs. These messages are exchanged only between the DTE/DCE pair and are never transmitted across the network in-band of the PVC.

Within IOS, you can assign the LMI type by using a static assignment or a feature called LMI autosense. A statically defined LMI type comes in three different standards:

- ansi— The Annex D standard defined by the ANSI standard T1.617.
- q.933— The ITU-T Q.933 Annex A standard.
- cisco— The original LMI type defined by the Gang of Four: Cisco, Digital Equipment Corporation, Northern Telecom, and StrataCom. cisco is the default LMI type on a Cisco router.

NOTE

The LMI autosense feature is covered in the next section.

The term LMI refers to a specific signaling protocol, but all three of the definable LMI types available in IOS are generally referred to as LMI. Be careful when deciding which type you will use. Even though all the LMI types are designed to support the same basic functionality, there are enough differences between them that the interfaces on your DTE/DCE pair must run the same LMI type, or you will experience unpredictable results.

By default, the FRAD sends LMI status messages to the WAN every 10 seconds. A full status request is sent as every sixth LMI status query. The WAN responds with a long status message, including any new events that have occurred since the last long status message.

You can use the following command to set the LMI type that your interface will use to communicate with the Frame Relay switch:

```
R4(config-if)#frame-relay lmi-type [cisco | ansi | q933a]
```

LMI Autosense

Cisco FRADs running Cisco IOS Release 11.2 and above support the LMI autosense feature. LMI autosense lets you "sense" the LMI sent by one device that has the LMI type configured, usually on your service provider's WAN equipment, preventing possible misconfiguration.

LMI autosense is automatically enabled in the following situations:

- The router is powered up or the interface changes state to up.
- The line protocol is down, but the line is up.
- The interface is a Frame Relay DTE.
- The LMI type is not explicitly configured on the interface.

When LMI autosense is active, the FRAD sends a full status request in all three LMI message formats to the WAN equipment. It starts with ANSI, and then uses ITU, and finally Cisco in rapid succession. LMI information is passed on DLCI 0 for both the Cisco LMI and Q.933a LMI types. LMI information is passed on DLCI 1023 for the ANSI LMI type. LMI autosense works because the Frame Relay code in IOS can listen to both DLCI 1023 and 0 at the same time.

When the three messages reach the switch, one or more of them elicit a reply, sent back in the form of a status message. Your FRAD then decodes the reply's format to configure the interface's LMI type automatically. Accommodating intelligent switches that can support multiple LMI types and send more than one reply is handled by the FRAD, which configures itself using the last LMI type received. Now, if you look back at the sequence in which the LMI messages are sent, the order should make more sense to you.

If LMI autosense fails to detect the correct LMI type, a retry interval is initiated. For every N391 time interval, which has a 60-second default, LMI autosense retries its automatic LMI configuration sequence.

Disabling or Reenabling Inverse ARP

Inverse ARP is used to build dynamic address mappings in Frame Relay networks running AppleTalk, Banyan VINES, DECnet, IP, Novell IPX, and XNS. Inverse ARP allows your FRAD to discover the protocol address of a device associated with the VC.

Inverse ARP is enabled by default, but you have the option of explicitly disabling it for a given protocol and DLCI pair. You should disable or reenable Inverse ARP under the following conditions:

- You should disable Inverse ARP for a selected protocol and DLCI pair when you know that the protocol is not supported on the other end of the connection.
- You should reenable Inverse ARP for a protocol and DLCI pair if conditions or equipment change and the protocol is then supported on the other end of the connection.

Inverse ARP is not required if you use a point-to-point interface, because there is only a single destination, and discovery is not required.

To enable Inverse ARP for a specific protocol and DLCI pair, use the following command:

```
R4(config-if)#frame-relay inverse-arpprotocol dlci
```

To disable Frame Relay Inverse ARP for a specific protocol and DLCI pair, use the following command:

```
R4(config-if)#no frame-relay inverse-arpprotocol dlci
```

Frame Relay Subinterfaces

To support a partially meshed Frame Relay network, you should use Cisco IOS's subinterface capabilities. Most protocols in use today need to believe that they have transitivity on a logical network. They assume that if Station A can talk to Station B, and Station B can talk to Station C, Station A should be able to talk directly to Station C. Although this concept of transitivity is mostly true on LANs, it is not true on a Frame Relay network unless Station A is directly connected to Station C.

One other item that certain protocols have an issue with on partially meshed networks, such as AppleTalk and transparent bridging, is split horizon. Split horizon states that when a packet is received on an interface, it cannot be sent out the same interface, even if it is received and transmitted on different VCs.

Frame Relay uses subinterfaces to overcome the issues raised by split horizon by ensuring that a single physical interface is treated as multiple virtual interfaces.

Virtual interfaces are seen as being separate from other virtual interfaces. This allows packets that are received on one virtual interface to be forwarded out another virtual interface, even if the virtual interfaces are configured on the same physical interface.

Subinterfaces also address the limitations of Frame Relay networks by giving you a way to subdivide your partially meshed Frame Relay network into a number of smaller, fully meshed (or point-to-point) subnetworks. You assign each subnetwork its own network numbers, making it appear to the protocols as if these networks can be reached through a separate interface. If you have transparent bridging in your networking environment, each subinterface is viewed as a separate bridge port.

NOTE

A point-to-point subinterface can be implemented as an unnumbered interface when used with IP, reducing the addressing burden that might otherwise result.

Subinterface Addressing

When you use point-to-point subinterfaces, the destination is identified or implied by the use of the`frame-relay interface-dlci` command. When you use multipoint subinterfaces, the destinations can be dynamically resolved through the use of Frame Relay Inverse ARP or can be statically mapped through the use of the `frame-relay map` command.

Addressing on Point-to-Point Subinterfaces

You can use the following command to address a point-to-point subinterface:

```
R2(config-subif)#frame-relay interface-dlcidlci
```

NOTE

If you define a subinterface as a point-to-point subinterface, you cannot reassign the same subinterface number as a multipoint subinterface without first rebooting the device.

Inverse ARP on Multipoint Subinterfaces

Inverse ARP provides dynamic address mapping for Frame Relay. Inverse ARP sends a request to map the next-hop protocol address for a specific connection given a DLCI. Responses to the Inverse ARP request are entered in an address-to-DLCI mapping table on the FRAD. The table entries are then used to supply the next-hop protocol address or the DLCI for outgoing traffic.

Because you now have a physical interface that is logically divided into multiple subinterfaces, you must provide information so that a specific subinterface can be distinguished from the physical interface and can be associated with a specific DLCI.

You can use the following command to accomplish this task:

```
R2(config-if)#frame-relay interface-dlcidlci
```

Inverse ARP is enabled by default for all the protocols it supports, but you can disable it for specific protocol-DLCI pairs. Because of this option, you can use dynamic mapping for some protocols and static mapping for others on the same DLCI. You can also explicitly disable Inverse ARP for a protocol-DLCI pair if you know that the protocol is not supported on the other end of the connection.

Static Address Mapping on Multipoint Subinterfaces

You can use a static map to link a specified next-hop protocol address to a specified DLCI. When you use static mapping, inverse ARP is automatically disabled for the specified protocol on the specified DLCI.

You are required to use static mapping if the router at the other end either does not support inverse ARP or does not support inverse ARP for a specific protocol you want to use.

You can use the following commands to establish static mapping according to your network needs:

```
R2(config-if)#frame-relay mapprotocol protocol-address dlci [broadcast] [ietf]

  [cisco]

R2(config-if)#frame-relay map clnsdlci [broadcast]

R2(config-if)#frame-relay map bridgedlci [broadcast] [ietf]
```

Table 9-2 lists the supported protocols and their corresponding keywords.

Table 9-2. Supported Protocols for Static Mapping

Supported Protocol	Corresponding Keyword
IP	ip
DECnet	decnet
AppleTalk	appletalk
XNS	xns
Novell IPX	ipx
VINES	vines
ISO CLNS	clns

You must use the broadcast keyword for routing protocols such as OSI protocols and the Open Shortest Path First (OSPF) protocol.

Configuring a Backup Interface for a Subinterface

You can use a backup interface with both a point-to-point subinterface and a multipoint Frame Relay subinterface. This allows individual PVCs to be backed up in case of failure rather than depending on the entire Frame Relay connection to fail before any redundancy takes over. You can configure a subinterface for backup on failure only, not for backup based on loading of the line.

Any backup interface you configure for the main interface has precedence over any subinterface backup interface you have configured when a complete loss of connectivity is experienced. Because of this, a subinterface backup is activated only if the main interface is up or if it is down and does not have a backup interface defined. If a subinterface fails while the backup interface is in use, and the main interface goes down, the backup subinterface remains connected.

You can use the following commands to configure a backup interface for a Frame Relay subinterface:

```
R2(config)#backup interfacetype number

R2(config-if)#backup delayenable-delay disable-delay
```

Network-to-Network Interface

One item of concern for a service provider is the possibility that a Frame Relay network might cross between two networks that might not be Cisco equipment. Because each vendor supports Frame Relay standards, they are also given the option of providing customizations that differentiate their product form another vendor's product.

To facilitate intervendor communication, the Network-to-Network Interface (NNI) port was defined as a bidirectional protocol to allow configuration, administration, and control information to be communicated between two networks. NNI consists of two independent unidirectional signaling protocols, one from each network, to provide bidirectional communication.

NNI supports status exchanges between the two networks, much like the exchanges between a DTE/DCE pair. The biggest difference between NNI communications and DTE/DCE pair communications is that both sides can initiate a query message exchange, and both sides can respond with either a short or long status message.

User-Network Interface

The User-Network Interface (UNI) port defines a unidirectional protocol that allows your FRAD to request information about all available PVCs in your service provider's Frame Relay equipment. Your FRAD can then use this information to ensure its proper configuration for the transmission or acceptance of any DLCI defined on your service provider's equipment.

Because of the UNI's nature (it is the signaling protocol used between the DTE and the DCE), it does not allow for full configuration, administration, and control between two peer devices.

Congestion-Control Mechanisms

Congestion is a problem that plagues the WAN environment more than the LAN environment in networking today. With the speed of today's LAN networks (up to 10 Gb on some interfaces), congestion is not as big a problem in the LAN environment as it has been in the past. Congestion usually occurs when you try to shove 10 Gb of information through a 1.5 Mbps T1 Frame Relay.

One reason that congestion still is a problem in today's environment is that the developers of WAN protocols must deal with the overhead associated with any type of congestion control, causing an increase in congestion. When you are paying a premium for limited speed, you don't want a significant amount of management traffic taking resources away from the critical data that the link was originally purchased for. Frame Relay can reduce this network overhead by implementing a simple congestion-notification mechanism rather than explicit, per-VC flow control.

Shortcomings of CIR

Frame Relay networks provide guaranteed throughput to your critical traffic as long as your data rate falls below the established CIR. If your data rate exceeds the established CIR, the network devices can set a DE bit on the exceeded frames. The DE bit is covered in more detail later in this chapter.

Unfortunately, CIR is not an adaptable setting that can provide flexibility when your traffic rates vary. Service providers often offer their customers the option of bursting above CIR for a defined period of time to handle the bursty nature of LAN traffic crossing a serial interface. Committed burst (Bc) size and excess burst (Be) size define the amount of traffic you can burst above your CIR.

Bc defines the maximum amount of bursty traffic under normal conditions. Be defines the maximum amount of bursty traffic in excess of Bc that Frame Relay attempts to transfer over a set period of time. If the number of frames entering the Frame Relay network is greater than Bc + Be, and the DE bit is set to 1, these frames are discarded.

> NOTE
>
> Be is used to determine the maximum data rate (MaxR) for the Frame Relay circuit. MaxR is measured in bits per second and uses the following formula:
>
> MaxR = [(Bc + Be) / Bc] * CIR

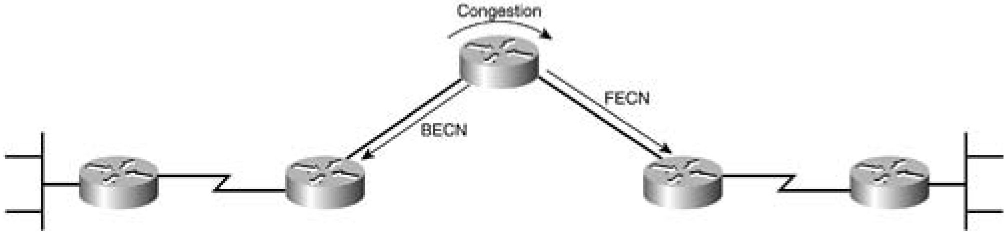
If congestion is encountered in a Frame Relay network, two different congestion-notification mechanisms can be used to inform the devices on the circuit:

- FECN
- BECN

Figure 9-4 illustrates the directions of FECN and BECN. Notice that FECN travels in the direction of congestion, and BECN travels in the opposite direction of congestion.

Figure 9-4. FECN and BECN Directions

[View full size image]



FECN and BECN each use a single bit in the Frame Relay frame header for congestion notification. Frame Relay also reserves another bit in the header, the DE bit, to mark traffic that may be discarded in the event of congestion.

The FECN Bit

The FECN bit is located in the Address field of the Frame Relay header. The FECN mechanism, used when a DTE device sends frames into the Frame Relay network, is set to 1 when congestion is present. After the frames reach the destination DTE device, the Address field (with the FECN bit set) can be examined. If the bit is set to 1, the frame experienced congestion along the path from the source DTE to the destination DTE. This information can then be sent to a higher-layer protocol for processing. Depending on the implementation of the higher-layer protocol, this information may be used to initiate some type of flow control, or it may be ignored.

The BECN Bit

The BECN bit is located in the Address field of the Frame Relay header. The value of the BECN is set to 1 by the DCE device for frames traveling in the opposite direction of frames that have their FECN bit set. This information tells the receiving DTE device that this particular path through the network is currently experiencing congestion. This information can then be sent to a higher-layer protocol for processing. Depending on the implementation of the higher-layer protocol, this information may be used to initiate some type of flow control, or it may be ignored.

NOTE

A quick review of the FECN/BECN bit tells you that a set FECN bit indicates that a frame encountered congestion, and a set BECN bit notifies the sender of congestion conditions on the circuit. The BECN frame might or might not have encountered any congestion of its own.

Frame Relay Discard Eligibility

Just like traffic that traverses your LAN, certain traffic crossing your WAN needs to have a higher priority than other traffic. There has to be a mechanism for you to ensure that traffic used for business purposes has a higher priority than traffic used to update someone's stock ticker (unless, of course, the stock market is your business). The Frame Relay DE bit indicates frames of priority lower than frames you identify as business-essential. The DE bit is located in the Address field in the Frame Relay header.

When your DTE sets the DE bit to 1, it indicates to the network that this is a frame of lower priority that is eligible for discard (assuming that you have negotiated this in your SLA). On notification of congestion, the DCE begins discarding frames that have the DE bit set before discarding those that do not. This simple management mechanism reduces the likelihood that business-critical traffic will be dropped during periods of congestion.

You can create DE lists that identify the characteristics of packets you want to be eligible for discarding. You can also specify DE groups to identify the DLCI that is affected.

You can use the following command to define a DE list to specify the packets that can be dropped when the Frame Relay switch is congested:

```
R2(config)#frame-relay de-listlist-number {protocolprotocol | interface
  typenumber}characteristic
```

You can base your DE lists on the protocol or the interface, and on characteristics such as fragmentation of the packet, a specific TCP or User Datagram Protocol (UDP) port, an access control list (ACL) number, or a packet size.

You can use the following command to define a DE group specifying the DE list and DLCI affected:

```
R2(config-if)#frame-relay de-groupgroup-number dlci
```

Frame Relay Error Checking

Frame Relay uses the CRC, used in many applications such as the file systems in today's popular operating systems, to provide an error-checking mechanism. The CRC works by comparing two calculated values to determine if any errors in the frames were encountered along the transmission path from source to destination. Frame Relay uses the CRC to reduce network overhead caused by error-checking mechanisms. By leaving the extensive error checking up to the higher-layer protocols you run, Frame Relay is not required to retransmit a packet. Instead, the upper-layer protocols retransmit any required packets.

Frame Relay Traffic Shaping

Traffic shaping with Frame Relay applies to both PVCs and SVCs. You can configure Frame Relay traffic shaping by performing the following tasks:

- Enable Frame Relay encapsulation on an interface (covered earlier in this chapter).
- Define VCs for different types of traffic.
- Enable Frame Relay traffic shaping on an interface.
- Enable LMI.
- Specify a traffic-shaping map class for an interface.
- Define a map class with queuing and traffic-shaping parameters.
- Define an ACL.
- Define priority queue lists for the map class.
- Define custom queue lists for the map class.

The following traffic-shaping features are available when you use Cisco IOS Release 11.2 or above:

- Rate enforcement on a per-VC basis— The peak rate for your outbound traffic. This value can be set to match CIR or any other value.
- Dynamic traffic throttling on a per-VC basis— When BECN packets indicate congestion on the network, the outbound traffic rate is automatically stepped down; when congestion eases, the outbound traffic rate is increased.
- Enhanced queuing support on a per-VC basis— Either custom queuing or priority queuing can be configured for individual VCs.

Defining VCs for Different Types of Traffic

You can perform virtual TDM on the same line by defining separate VCs for different types of traffic and specifying queuing and an outbound traffic rate for each VC. In this manner, you can provide guaranteed bandwidth for each traffic type that crosses the line. This enhances your ability to throttle outbound traffic from a high-speed LAN line in your central office to a lower-speed WAN line going to your remote locations, easing congestion and data loss in your network. Enhanced queuing mechanisms can also prevent congestion-caused data loss.

Enabling Frame Relay Traffic Shaping on the Interface

By enabling Frame Relay traffic shaping on an interface, you enable both traffic shaping and per-VC queuing on all PVCs and SVCs defined on the interface. Remember that traffic shaping lets your FRAD control the circuit's output rate and, if configured, react to congestion notification information.

You can use the following command to enable Frame Relay traffic shaping on a specified interface:

```
R2(config-if)#frame-relay traffic-shaping
```

Specifying a Traffic-Shaping Map Class for the Interface

When you specify a Frame Relay map class for a main interface, all the VCs you define on its subinterfaces also inherit the traffic-shaping parameters defined for the class.

You can use the following command to specify a map class for a specified interface:

```
R2(config-if)#frame-relay classmap-class-name
```

You can override the default for a specific DLCI on a specific subinterface by using the class VC command to explicitly assign the DLCI to a different class.

Defining a Map Class with Queuing and Traffic-Shaping Parameters

You can specify the average and peak rates, in bits per second, that you want to allow on a VC by defining and associating it with a map class. You can also specify a custom queue list or a priority queue group for use by the VC associated with the map class. You can use the following commands to define a map class:

This command specifies a map class:

```
R2(config)#map-class frame-relaymap-class-name
```

This command defines the traffic rate:

```
R2(config-map-class)#frame-relay traffic-rateaverage [peak]
```

This command specifies a custom queue list:

```
R2(config-map-class)#frame-relay custom-queue-listlist-number
```

This command specifies a priority queue list:

```
R2(config-map-class)#frame-relay priority-grouplist-number
```

To select BECN or ForeSight as a congestion backward-notification mechanism to which traffic shaping adapts, use this command:

```
R2(config-map-class)#frame-relay adaptive-shaping {becn | foresight}
```

Defining ACLs

When you use custom queuing, you can specify an ACL to identify the traffic it will use. You associate the lists through the list numbers. For more information on defining ACLs, refer to the *Traffic Filtering and Firewalls* configuration guide for the IOS version you are using.

Defining Priority Queue Lists for the Map Class

You have the option of defining a priority list for a protocol and also a default priority list. You use the number you specified for a specific priority list to associate it to the Frame Relay priority group defined for a specified map class.

For example, when you enter the frame-relay priority-group 2 command for the map class fast_vcs, and then you enter the priority-list 2 protocol decnet high command, that priority list is used for the fast_vcs map class. The average and peak traffic rates you defined for the fast_vcs map class are used for DECnet traffic.

Defining Custom Queue Lists for the Map Class

You have the option of defining a queue list for a protocol and a default queue list. You also have the option of specifying the maximum number of bytes to be transmitted in any given cycle. You use the number you specified for a specific queue list to associate it to the Frame Relay custom queue list defined for a specified map class.

For example, when you enter the frame relay custom-queue-list 1 command for the map class slow_vcs and then you enter the queue-list 1 protocol ip list 100 command, that queue list is used for the slow_vcs map class. The access-list 100 definition is also used for that map class and queue. The average and peak traffic rates you defined for the slow_vcs map class are used for IP traffic that meets the access-list 100 criteria.

Troubleshooting Frame Relay Connectivity

Now that your Frame Relay is configured, there might come a time when you need to ensure that it is working correctly. Fortunately, Cisco provides many different ways to verify configurations. Two easy ways to accomplish this through a CLI are the show and debug suite of commands available in Cisco IOS.

Becoming familiar with the show and debug commands available for Frame Relay allows you to quickly troubleshoot and correct most problems without becoming overloaded with a lot of excess information. In this chapter, only commands that relate to the information already covered are explored. For a complete list of available show and debug commands, refer to the *IOS WAN Configuration Guide* for the IOS version you will be using.

show frame-relay lmi Command

Because all traffic crossing a Frame Relay circuit rides over the LMI configured for that circuit, theshow frame-relay lmi command can provide you with valuable information. The output of this command contains a lot of information. When you start to troubleshoot a connectivity problem or verify that the circuit is operational, two fields, Num Status Enq. Sent and Num Update Status Rcvd, give you an idea of the circuit's health. Example 9-1 shows the output of this command as issued on the R4 router.

Example 9-1. Output of the show frame-relay lmi Command

```
R4#show frame-relay lmi

LMI Statistics for interface Serial0 (Frame Relay DTE) LMI TYPE = ANSI
  Invalid Unnumbered info 0             Invalid Prot Disc 0
  Invalid dummy Call Ref 0              Invalid Msg Type 0
  Invalid Status Message 0              Invalid Lock Shift 0
  Invalid Information ID 0              Invalid Report IE Len 0
  Invalid Report Request 0              Invalid Keep IE Len 0
  Num Status Enq. Sent 296              Num Status msgs Rcvd 293
  Num Update Status Rcvd 0              Num Status Timeouts 0
```

Looking at Example 9-1, you can see that the circuit is sending and receiving Status messages without any timeouts, which are vital to the operation of Frame Relay. This output also supplies the LMI type that the circuit is using for operation—in this case, ANSI. If you were experiencing a problem with the configured LMI type, you would see output similar to Example 9-2.

Example 9-2. Mismatched LMI

```
R4#show frame-relay lmi

LMI Statistics for interface Serial0 (Frame Relay DTE) LMI TYPE = ANSI
  Invalid Unnumbered info 0             Invalid Prot Disc 0
  Invalid dummy Call Ref 0              Invalid Msg Type 0
  Invalid Status Message 0              Invalid Lock Shift 0
  Invalid Information ID 0              Invalid Report IE Len 0
  Invalid Report Request 0              Invalid Keep IE Len 0
  Num Status Enq. Sent 96               Num Status msgs Rcvd 3
  Num Update Status Rcvd 0              Num Status Timeouts 93
```

As you can see, your Num Status Timeouts are increasing, indicating a misconfigured circuit.

show frame-relay pvc Command

After you have confirmed that your LMI matches the service provider's, you can verify that you have the proper PVC(s) configured. Use the show frame-relay pvc [*dlci* | *interface*] command to display information about the DLCIs that the router is aware of. Example 9-3 shows output from this command.

Example 9-3. Output of the show frame-relay pvc Command

```
R4#show frame-relay pvc

PVC Statistics for interface Serial0 (Frame Relay DTE)

              Active     Inactive      Deleted       Static
```

```
  Local          3            0            0            0
  Switched       0            0            0            0
  Unused         0            0            0            0

DLCI = 110, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.1

  input pkts 78            output pkts 78           in bytes 21770
  out bytes 22404          dropped pkts 0           in FECN pkts 0
  in BECN pkts 0           out FECN pkts 0          out BECN pkts 0
  in DE pkts 0             out DE pkts 0
  out bcast pkts 63        out bcast bytes 20844
  pvc create time 01:00:23, last time pvc status changed 00:59:45

DLCI = 120, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.2

  input pkts 10            output pkts 20           in bytes 1040
  out bytes 2080           dropped pkts 0           in FECN pkts 0
  in BECN pkts 0           out FECN pkts 0          out BECN pkts 0
  in DE pkts 0             out DE pkts 0
  out bcast pkts 0         out bcast bytes 0
  pvc create time 01:00:17, last time pvc status changed 00:59:47

DLCI = 130, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.2

  input pkts 15            output pkts 16           in bytes 1560
  out bytes 1620           dropped pkts 0           in FECN pkts 0
  in BECN pkts 0           out FECN pkts 0          out BECN pkts 0
  in DE pkts 0             out DE pkts 0
  out bcast pkts 0         out bcast bytes 0
  pvc create time 01:00:19, last time pvc status changed 00:59:49
```

NOTE

Notice that the output of the show frame-relay pvc command displays information about all the PVCs the router knows about. If you want more-specific information about a specific interface or DLCI, you can supply the proper keyword with the command and receive only that information.

If you analyze the output of the show frame-relay pvc command in Example 9-3, you will notice that all the configured PVCs are in an active state. PVCs are in one of three states at any given time:

- ACTIVE— Your PVC is active and can pass traffic.
- INACTIVE— Your local connection to Frame Relay is operational, but the remote router's connection is not operational.
- DELETED— You are not receiving LMIs, or the physical layer is encountering a problem.

Other areas of interest in this output include the pvc create time, which tells you when the PVC was created, and the last time pvc status changed time, which tells you the last time the PVC state time changed. Both of these items can provide invaluable troubleshooting information.

If you are looking for information about congestion, this is the command to use, because it shows the counters related to FECN and BECN packets the router has processed.

debug frame-relay lmi Command

Like most technologies supported by Cisco IOS, Frame Relay supports debugging of numerous configuration items. The one debug command this chapter examines, debug frame-relay lmi, is a useful command when you start troubleshooting (see Example 9-4). If you require the use of other debugging commands for Frame Relay, refer to the *IOS WAN Configuration Guide* for your IOS version.

Example 9-4. Output of the debug frame-relay lmi Command

```
R4#debug frame-relay lmi
Frame Relay LMI debugging is on
Displaying all Frame Relay LMI data
```

```
R4#
01:51:51: Serial0(out): StEnq, myseq 31, yourseen 31, DTE up
01:51:51: datagramstart = 0xE30BD8, datagramsize = 14
01:51:51: FR encap = 0x00010308
01:51:51: 00 75 95 01 01 01 03 02 1F 1F
01:51:51:
01:51:51: Serial0(in): Status, myseq 31
01:51:51: RT IE 1, length 1, type 1
01:51:51: KA IE 3, length 2, yourseq 32, myseq 31
01:52:01: Serial0(out): StEnq, myseq 32, yourseen 32, DTE up
01:52:01: datagramstart = 0xE30BD8, datagramsize = 14
01:52:01: FR encap = 0x00010308
01:52:01: 00 75 95 01 01 01 03 02 20 20
01:52:01:
01:52:01: Serial0(in): Status, myseq 32
01:52:01: RT IE 1, length 1, type 1
01:52:01: KA IE 3, length 2, yourseq 33, myseq 32
01:52:11: Serial0(out): StEnq, myseq 33, yourseen 33, DTE up
01:52:11: datagramstart = 0xE30BD8, datagramsize = 14
01:52:11: FR encap = 0x00010308
01:52:11: 00 75 95 01 01 01 03 02 21 21
01:52:11:
01:52:11: Serial0(in): Status, myseq 33
01:52:11: RT IE 1, length 1, type 1
01:52:11: KA IE 3, length 2, yourseq 34, myseq 33
R4#
```

You can see from this output that this router is successfully exchanging LMIs with the service provider's Frame Relay switch. You know this because the fields myseq and yourseq are increasing. The router adds 1 to the received sequence number with each successive message sent. If this field were not increasing, LMI exchanges would not be occurring. If three successive LMI messages are sent without a reply, where only one field is increasing, the link is reset, and the process restarts.

Scenarios

The Scenarios presented in this chapter help you gain a better understanding of modem operation and configuration through practical application. You will go through the necessary configuration tasks in their logical progression. The two Scenarios cover the following topics:

- Enabling Frame Relay
- Subinterface types
- Assigning IP addressing and DLCI
- Addressing on multipoint subinterfaces
- Inverse ARP
- Configuring multipoint subinterfaces
- Configuring traffic shaping on a PVC
- Configuring guaranteed rates on an interface

Before configuring Frame Relay, you need to perform an initial configuration of your routers. You perform your initial configuration on router R1. You need to complete this section on the remaining routers in this chapter when you use them. Your initial configuration can be done from a terminal attached to its console port (line 0). You begin by entering global configuration mode. You then configure the router name using the hostname command. It is also useful to disable the IP domain name system with the no ip domain-lookup command. This keeps the system from trying to translate domain names that have typing errors.

You can use the enable secret command to enable a password for entering privileged EXEC mode. Here the password is cisco. This secret password provides an additional layer of security on the router. Passwords are case-sensitive strings that can be up to 80 characters long. They cannot begin with a number.

To begin configuring the console line, you enter line console 0. You are now in line configuration mode. You use the no exec-timeout command to prevent the console from automatically disconnecting after a period of inactivity. The default timeout is 10 minutes. The initial configuration of R1 is now complete, as shown in Example 9-5.

NOTE

Don't forget to reset the exec-timeout after the configuration is completed. Leaving it open is a potential security risk.

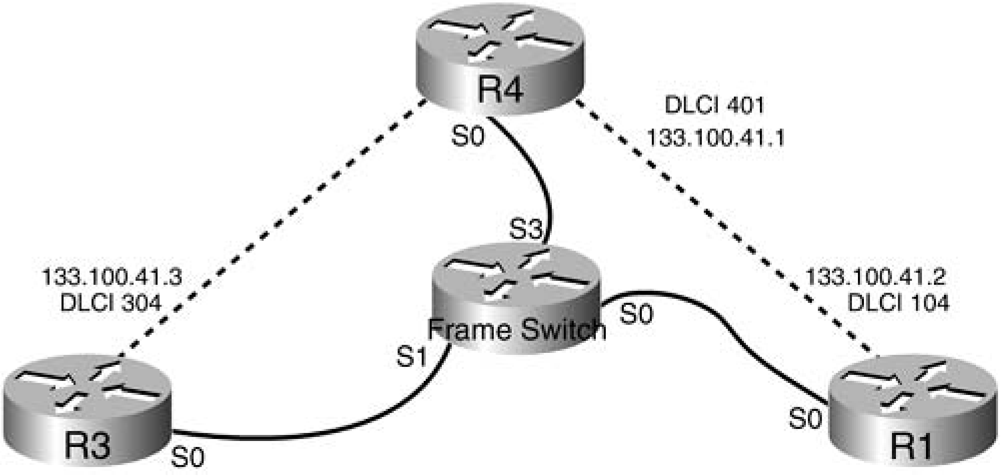
Example 9-5. Initial Configuration of R1

```
Router#configure terminal

Router(config)#hostname R1

R1(config)#no ip domain-lookup

R1(config)#enable secret cisco

R1(config-if)#line console 0

R1(config-line)#no exec-timeout
```

Scenario 9-1: Enabling Frame Relay

Enabling Frame Relay is the logical place for you to start your configurations. In this Scenario, you will enable Frame Relay on the necessary serial interfaces of Routers R1 and R3. You will configure the interface for Router R4 in Scenario 9-2. Figure 9-5 illustrates the Frame Relay cloud that you will be configuring in this Scenario. R4 is the hub router, and R1 and R3 are spoke routers.

Figure 9-5. Frame Relay Cloud Topology



Step 1. Enable Frame Relay encapsulation on the interface:

```
R1(config)#interface type number
```

```
R1(config-if)#encapsulation frame-relay [ietf]
```

Example 9-6 shows the commands necessary to complete this Scenario.

Example 9-6. Enabling Frame Relay for R1 and R3

```
Configuration items for R1:
interface serial 0
encapsulation frame-relay

Configuration items for R3:
interface serial 0
encapsulation frame-relay
```

Scenario 9-2: Subinterface Types

You have two choices of subinterface types: point-to-point and multipoint, neither of which is the default. Follow these steps to configure a subinterface for use on a Frame Relay network:

Step 1. Create the subinterface:

```
R4(config)#interface typenumber.subinterface-number {multipoint |
  point-to-point}
```

Step 2. Enable the Frame Relay encapsulation:

```
R4(config-if)#encapsulation frame-relay [ietf]
```

In this Scenario, you will enable Frame Relay on R4's serial 0 interface. You will configure two point-to-point subinterfaces—serial 0.1 for R3 and serial 0.2 for R1. Example 9-7 shows the required steps to complete this Scenario.

Example 9-7. Creating Subinterfaces for R4

```
Configuration items for R4:

interface serial 0.1 point-to-point

encapsulation frame-relay

!

interface serial 0.2 point-to-point

encapsulation frame-relay
```

Scenario 9-3: Assigning IP Addressing and DLCIs

After configuring this Scenario, you will have a functioning Frame Relay topology. You will finish your configuration of the three routers by assigning IP addresses and DLCIs. Use the IP addresses and DLCIs shown in Figure 9-5.Example 9-8 shows the needed configurations to complete this task.

Example 9-8. Assigning IP Addresses and DLCIs

```
Configuration items for R1:

interface serial 0

ip address 133.100.41.2 255.255.255.252

frame-relay interface-dlci 104

Configuration items for R3:
```

```
interface serial 0

ip address 133.100.43.2 255.255.255.252

frame-relay interface-dlci 304


Configuration items for R4:

interface serial 0.1 point-to-point

ip address 133.100.41.1 255.255.255.252

frame-relay interface-dlci 401

!

interface serial 0.2 point-to-point

ip address 133.100.43.1 255.255.255.252

frame-relay interface-dlci 403
```

Scenario 9-4: Addressing on Multipoint Subinterfaces

When you use a multipoint subinterface, you have a few choices of how you can address the subinterface. You can use Inverse ARP to dynamically map the IP-to-DLCI, or you can statically define the IP-to-DLCI mapping, turning off the Inverse ARP feature. Be aware that not all protocols support dynamic address mapping and must use static address mapping.

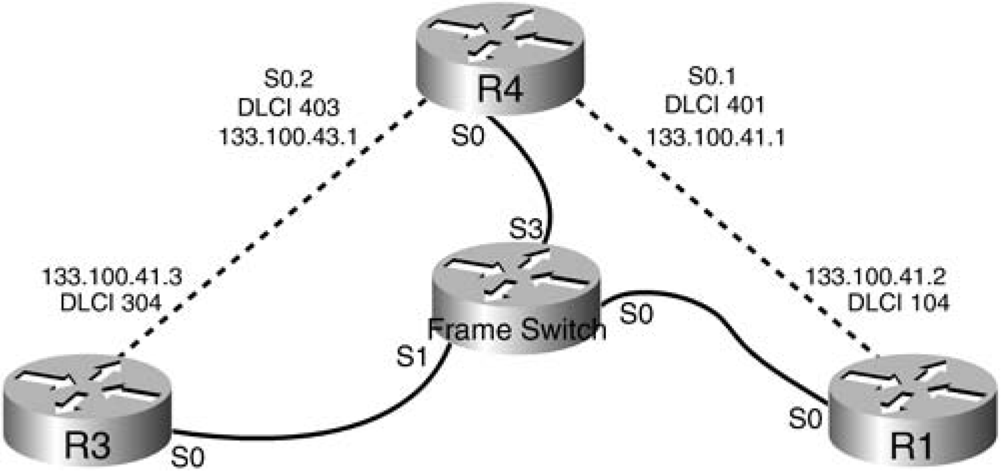
You can create an IP-to-DLCI mapping using the following command:

```
R4(config-if)#frame-relay map [ip | Apollo | appletalk | bridge | clns | decnet |

  dlsw | ip | ipx | llc2 | qllc | rsrb | stun | vines | xns] {a.b.c.d}

  {dlci-number} [active | broadcast | cisco | ietf | nocompress |

  payload-compression | tcp]
```

Remember that Frame Relay by default does not forward a Layer 3 broadcast. Several routing protocols that you can use with Frame Relay do not operate correctly unless you use the broadcast keyword when you create your map.

Figure 9-6 illustrates the topology you will use for this Scenario.

Figure 9-6. frame-relay map Topology



In this Scenario, you configure R4 with a multipoint interface—in this case, the physical serial 0 interface. You will create IP-to-DLCI mappings for R1 and R3 to communicate. Example 9-9 illustrates the configuration needed to accomplish this task.

Example 9-9. Multipoint Interface Example

```
Configuration items for R1:
interface serial 0
ip address 133.100.41.2 255.255.255.240
encapsulation frame-relay
frame-relay map ip 133.100.41.3 104 broadcast
frame-relay map ip 133.100.41.1 104 broadcast

Configuration items for R3:
interface serial 0
ip address 133.100.41.3 255.255.255.240
```

```
encapsulation frame-relay
frame-relay map ip 133.100.41.2 304 broadcast
frame-relay map ip 133.100.41.1 304 broadcast
```

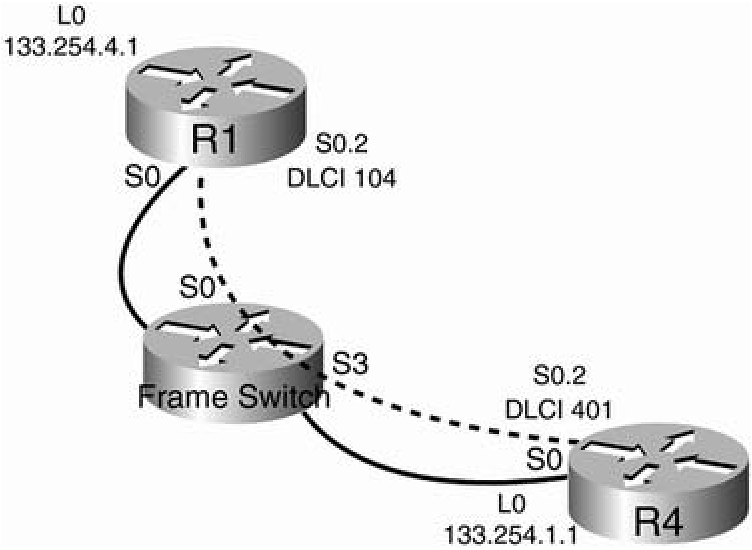
Configuration items for R4:

```
interface serial 0
ip address 133.100.41.1 255.255.255.240
encapsulation frame-relay
frame-relay interface-dlci 401
```

Practical Exercise 9-1: Unnumbered Frame Relay

In this Practical Exercise, you will configure IP unnumbered over subinterfaces at both ends of a point-to-point connection. You will use the IP addresses of the loopback interfaces for each end of the Frame Relay. Figure 9-7 illustrates the topology you will work with in this Practical Exercise.

Figure 9-7. IP Unnumbered Topology



Practical Exercise 9-1 Solution

Follow these steps to configure your Frame Relay topology:

Step 1. Create your loopback interface. You can choose to create a loopback interface with just about any number you want to use. In this Practical Exercise, you will use 0. Address the loopback interface as shown in Figure 9-7.

```
Configuration items for R1:

R1(config)#interface loopback 0

R1(config-if)#ip address 133.254.1.1 255.255.255.0


Configuration items for R4:

R4(config)#interface loopback 0

R4(config-if)#ip address 133.254.4.1 255.255.255.0
```

Step 2. You can enable Frame Relay on your serial interfaces. Although it is not necessary to enter the ip address command, it is shown here for completeness:

```
Configuration items for R1:

R1(config)#interface serial 0

R1(config-if)#no ip address

R1(config-if)#encapsulation frame-relay IETF


Configuration items for R4:
```

```
R4(config)#interface serial 0

R4(config-if)#no ip address

R4(config-if)#encapsulation frame-relay IETF
```

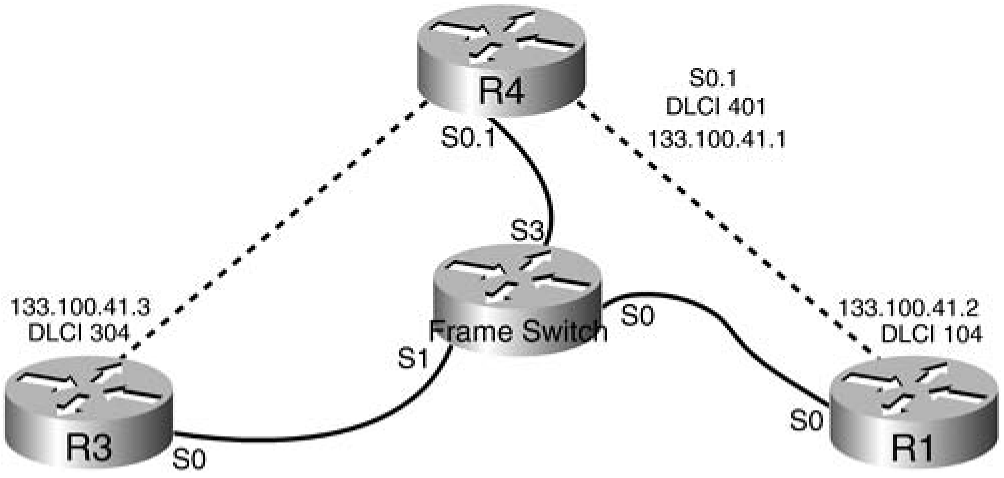
Step 3. Create your subinterfaces, and turn them into unnumbered interfaces. You also need to assign the appropriate DLCIs to your subinterfaces:

```
Configuration items for R1:

R1(config)#interface serial 0.2 point-to-point

R1(config-if)#ip unnumbered loopback0

R1(config-if)#frame-relay interface-dlci 20


Configuration items for R4:

R4(config)#interface serial 0.2 point-to-point

R4(config-if)#ip unnumbered loopback0

R4(config-if)#frame-relay interface-dlci 30
```

Practical Exercise 9-2: Configuring Multipoint Subinterfaces

In this Practical Exercise, you will configure a multipoint subinterface on R4 with point-to-point subinterfaces on R1 and R3. You will configure the necessary static mapping to allow IP connectivity across the circuits. Figure 9-8 illustrates your next topology, similar to the topology you used earlier. The difference is that here you use a multipoint interface to complete the configuration.

Figure 9-8. Multipoint Interface Topology



Practical Exercise 9-2 Solution

Follow these steps to configure your multipoint subinterface Frame Relay topology:

Step 1. Enable Frame Relay on your serial interfaces:

```
Configuration items for R1:

R1(config)#interface serial 0

R1(config-if)#encapsulation frame-relay


Configuration items for R3:

R3(config)#interface serial 0

R3(config-if)#encapsulation frame-relay


Configuration items for R4:

R4(config)#interface serial 0

R4(config-if)#encapsulation frame-relay
```

Step 2. Create your subinterfaces, and place the appropriate IP addresses on them:

```
Configuration items for R1:

R1(config)#interface serial 0.1 point-to-point

R1(config-if)#ip address 133.100.41.2 255.255.255.240
```

```
Configuration items for R3:

R3(config)#interface serial 0.1 point-to-point

R3(config-if)#ip address 133.100.41.3 255.255.255.240
```

```
Configuration items for R4:

R4(config)#interface serial 0.1 multipoint

R4(config-if)#ip address 133.100.41.1 255.255.255.240
```

Step 3. Configure the DLCI for R4:

```
Configuration items for R4:

R4(config)#interface serial 0.1 multipoint

R4(config-if)#frame-relay interface-dlci 401
```

Step 4. Create the appropriate mappings for R1 and R3:

```
Configuration items for R1:

R1(config)#interface serial 0.1 point-to-point

R1(config-if)#frame-relay map ip 133.100.41.3 104 broadcast

R1(config-if)#frame-relay map ip 133.100.41.1 104 broadcast
```

Configuration items for R3:

```
R3(config)#interface serial 0.1 point-to-point

R3(config-if)#frame-relay map ip 133.100.41.2 304 broadcast

R3(config-if)#frame-relay map ip 133.100.41.1 304 broadcast
```

Practical Exercise 9-3: Configuring Traffic Shaping on a PVC

In this Practical Exercise, you will configure traffic shaping for a PVC that will carry voice over Frame Relay traffic. You will use the Cisco-proprietary fragmentation on the class associated with the PVC. You will use 100 for the fragmentation value, 64 Kb for CIR, and 25 Kb for voice.

Practical Exercise 9-3 Solution

Follow these steps to configure your Frame Relay topology:

Step 1. Create your map class called vofr-class with the specified settings:

```
Configuration items for R2:

R2(config)#map-class frame-relay vofr-class

R2(config-map-class)#frame-relay fragment 100

R2(config-map-class)#frame-relay fair-queue

R2(config-map-class)#frame-relay cir 64000

R2(config-map-class)#frame-relay voice bandwidth 25000
```

Step 2. Enable Frame Relay on the serial interface:

```
Configuration items for R2:

R2(config)#interface serial 0

R2(config-if)#encapsulation frame-relay
```

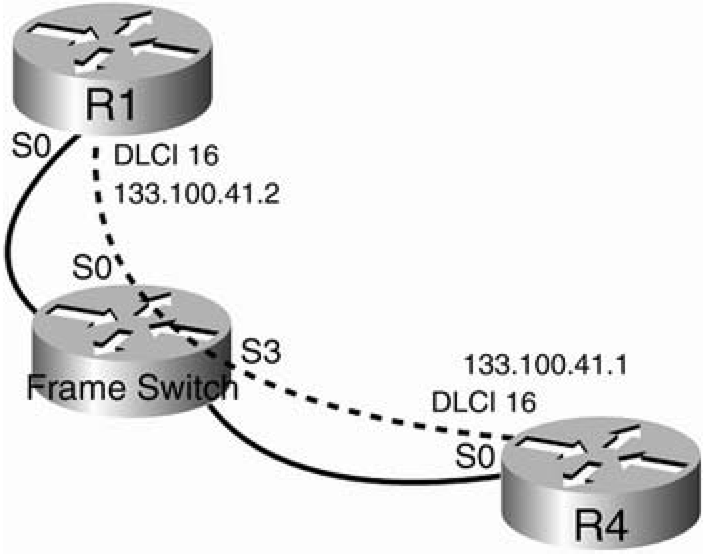
Step 3. Enable Frame Relay traffic shaping, and assign the map class to the serial interface:

```
Configuration items for R2:

R2(config)#interface serial 0

R2(config-if)#frame-relay traffic-shaping

R2(config-if)#frame-relay interface-dlci 108

R2(config-if)#frame-relay class vofr-class
```

Practical Exercise 9-4: Configuring Guaranteed Rates on an Interface

In this Practical Exercise, you will configure a hub with a physical rate of 192 Kbps and a guaranteed rate of 32 Kbps and a remote site with a physical rate of 64 Kbps and a guaranteed rate of 32 Kbps. You will configure traffic shaping so that each end has an average transmit rate of 64 Kbps. If needed, your hub site can burst above this. In case of congestion, it can drop to a minimum of 32 Kbps. Traffic shaping will be configured to adapt to BECN congestion notification. Figure 9-9 illustrates your next topology.

Figure 9-9. Frame Relay Traffic-Shaping Topology



Practical Exercise 9-4 Solution

Follow these steps to configure your Frame Relay topology:

Step 1. Create your map class with the specified rates:

```
Configuration items for R1:

R1(config)#map-class frame-relay cisco

R1(config-map-class)#frame-relay cir 64000

R1(config-map-class)#frame-relay mincir 32000

R1(config-map-class)#frame-relay adaptive-shaping becn

R1(config-map-class)#frame-relay bc 8000

R1(config-map-class)#frame-relay be 16000


Configuration items for R4:

R4(config)#map-class frame-relay cisco

R4(config-map-class)#frame-relay cir 64000

R4(config-map-class)#frame-relay mincir 32000

R4(config-map-class)#frame-relay adaptive-shaping becn

R4(config-map-class)#frame-relay bc 8000
```

Step 2. Enable Frame Relay encapsulation and traffic shaping on your serial interfaces:

```
Configuration items for R1:

R1(config)#interface Serial0

R1(config-if)#encapsulation frame-relay

R1(config-if)#frame-relay traffic-shaping


Configuration items for R4:

R4(config)#interface Serial0

R4(config-if)#encapsulation frame-relay

R4(config-if)#frame-relay traffic-shaping
```

Step 3. Create your subinterfaces, and apply the appropriate IP addresses and DLCIs:

```
Configuration items for R1:

R1(config)#interface Serial0.1 point-to-point

R1(config-subif)#frame-relay interface-dlci 16

R1(config-subif)#frame-relay class cisco


Configuration items for R4:

R4(config)#interface Serial0.1 point-to-point

R4(config-subif)#frame-relay interface-dlci 16

R4(config-subif)#frame-relay class cisco
```

Step 4. Apply your map class to the appropriate subinterfaces:

```
R1(config-subif)#interface Serial0.1 point-to-point

R1(config-subif)#frame-relay class cisco


Configuration items for R4:

R4(config-subif)#interface Serial0.1 point-to-point

R4(config-subif)#frame-relay class cisco
```

Practical Exercise 9-5: Configuring Frame Relay Switching

Frame Relay switching is the process of switching packets based on their assigned DLCI values. You have the option of configuring your FRAD to perform switching in a Frame Relay network. There are two parts you need to be concerned with in a Frame Relay network:

- Frame Relay DTE (the router or access server)
- Frame Relay DCE switch

This step is required before you can configure Frame Relay switching on a Frame Relay DTE or DCE, or with NNI support. You can use the following command to enable packet switching:

```
Frame_Switch(config)#frame-relay switching
```

You can configure an interface as a DTE device or a DCE switch, or as a switch connected to a switch, to support NNI connections. You can use the following command to accomplish this task:

```
Frame_Switch(config-if)#frame-relay intf-type [dce | dte | nni]
```

For PVC switching to operate, you must specify a static route:

```
Frame_Switch(config-if)#frame-relay routein-dlciinterfaceout-interface-type
  out-interface-number out-dlci
```

NOTE

You cannot configure static routes over a tunnel interface on the Cisco 800 series, 1600 series, and 1700 series platforms. You can configure static routes only over tunnel interfaces on platforms that have the Enterprise feature set.

Practical Exercise 9-5 Solution

In this Practical Exercise, you will configure R10 to switch DLCIs 100, 110, 120, and 130 between three interfaces. Follow these steps:

Step 1. Enable Frame Relay switching on your router:

```
R10(config)#frame-relay switching
```

Step 2. Enable Frame Relay on each of your interfaces:

```
R10(config)#interface Serial0

R10(config-if)#encapsulation frame-relay

R10(config)#exit

R10(config)#interface Serial1

R10(config-if)#encapsulation frame-relay

R10(config-if)#exit

R10(config)#interface Serial2

R10(config-if)#encapsulation frame-relay
```

Step 3. Specify the interfaces' clock rate:

```
R10(config)#interface Serial0
R10(config-if)#clockrate 2000000
R10(config-if)#exit
R10(config)#interface Serial1
R10(config-if)#clockrate 2000000
R10(config-if)#exit
R10(config)#interface Serial2
R10(config-if)#clockrate 2000000
```

Step 4. Specify the role the interface will play in the Frame Relay network—in this case, the DCE:

```
R10(config)#interface Serial0
R10(config-if)#frame-relay intf-type dce
R10(config-if)#exit
R10(config)#interface Serial1
R10(config-if)#frame-relay intf-type dce
R10(config-if)#exit
R10(config)#interface Serial2
R10(config-if)#frame-relay intf-type dce
```

Step 5. Configure the mapping for the Frame Relay switching:

```
R10(config)#interface Serial0

R10(config-if)#frame-relay route 130 interface Serial1 110

R10(config)#exit

R10(config)#interface Serial1

R10(config-if)#frame-relay route 100 interface Serial1 120

R10(config-if)#frame-relay route 110 interface Serial0 130

R10(config-if)#exit

R10(config)#interface Serial2

R10(config-if)#frame-relay route 120 interface Serial1 100
```

Summary

This chapter looked at the Frame Relay technology as supported by Cisco devices. Cisco IOS supports the Frame Relay standard as defined by both ANSI and the ITU-T. The chapter started by reviewing the configuration items needed to support a sample network. The theory behind the commands was examined, and a configuration of the items was reviewed. The chapter then looked at the many different commands you can use to troubleshoot and maintain your Frame Relay network.

Review Questions

1: Frame Relay is what kind of technology?

A. Packet-switched

B. Frame-switched

C. Time-switched

D. DVC-switched

2: Name and briefly describe the two kinds of packet-switching techniques discussed in this chapter.

3: Describe the difference between SVCs and PVCs.

4: What is a data-link connection identifier (DLCI)?

5: Describe how LMI Frame Relay differs from basic Frame Relay.

6: True or false: IP unnumbered can be used with Frame Relay.

7: Can Cisco routers connect to other vendor devices over Frame Relay?

8: Is Frame Relay inverse-arp on by default?

9: Is special configuration required to run OSPF over Frame Relay?

10: Is TCP header compression available for use with priority queuing?

Chapter 10. Enabling a Backup to the Permanent Connection

This chapter covers the following topics:

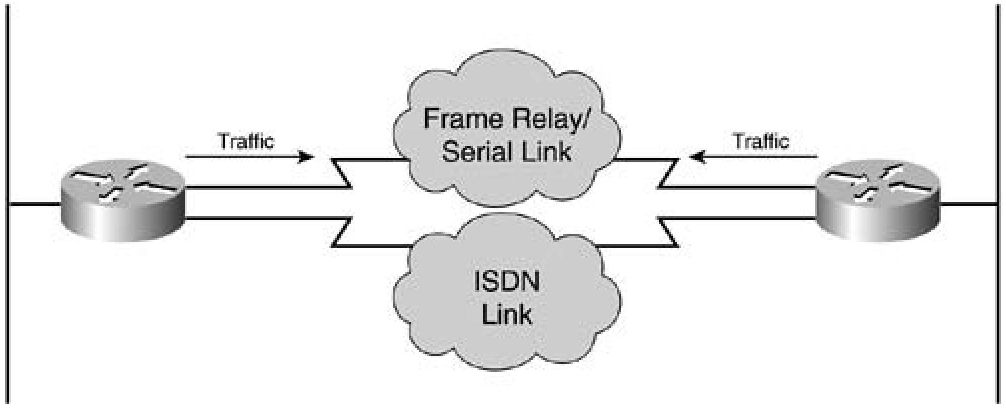
- Backup Overview
- Triggering Dial Backup

Redundancy is a crucial requirement in today's networks. It is especially important in WANs, where leased lines provide permanent connections. In these situations, backup interfaces can be configured to provide valuable redundancy to permanent connections.

Backup Overview

As you can see from Figure 10-1, a backup interface can be either a physical interface, such as an ISDN BRI interface, or an interface assigned to a dialer pool. When a backup interface is specified, it remains in standby mode until activated. When the interface is in standby mode, it remains idle, and the backup route between the routers does not appear in the routing table.

Figure 10-1. Backup Interfaces



As soon as a backup interface is configured, the router monitors the following on the primary link:

- Carrier detect signal
- Keepalives

If the router misses either a carrier detect signal or a keepalive, the primary link is assumed to be down, and the route is withdrawn from the routing table. When this happens, the backup interface is activated, and the backup route appears in the routing table.

Routers can also be configured to bring up a backup interface when the load on the primary interface meets or exceeds a certain limit. This limit can be configured.

Backup interfaces can be activated when certain events occur. Backup interfaces can include

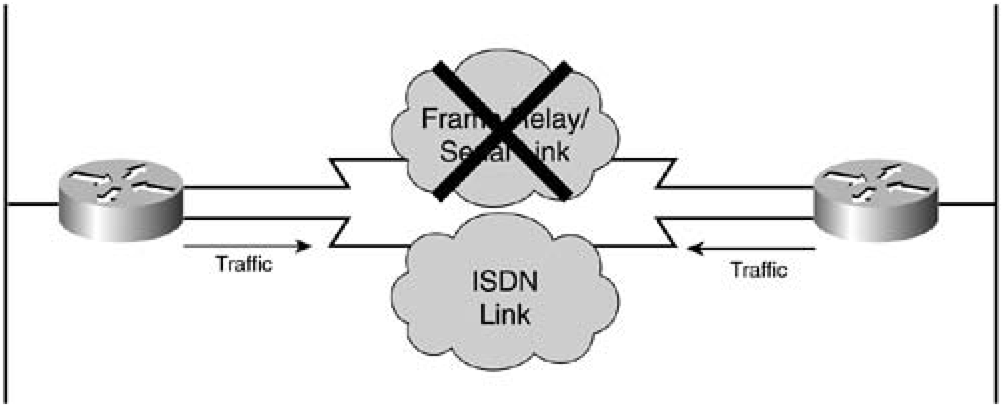
- Serial interfaces
- ISDN interfaces
- Asynchronous interfaces
- Dialer pools

Triggering Dial Backup

Dial backup can be triggered in two main ways:

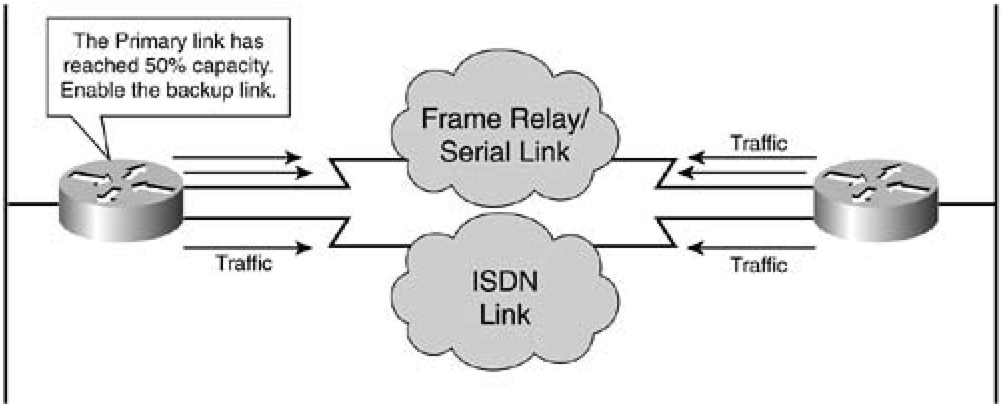
- Failure of the primary link— In this scenario, shown in Figure 10-2, the primary serial or Frame Relay link has failed. The backup ISDN interface can be configured to come up and provide redundancy. As soon as the ISDN link is up, traffic flows across the backup link.

Figure 10-2. Dial Backup When the Primary Link Fails



- Traffic on the primary link reaches or exceeds a threshold— In this scenario, shown inFigure 10-3, the load on the primary link is monitored, and a 5-minute moving average is computed. As soon as this average exceeds a threshold, the backup link is activated.

Figure 10-3. Dial Backup to Support the Primary Link

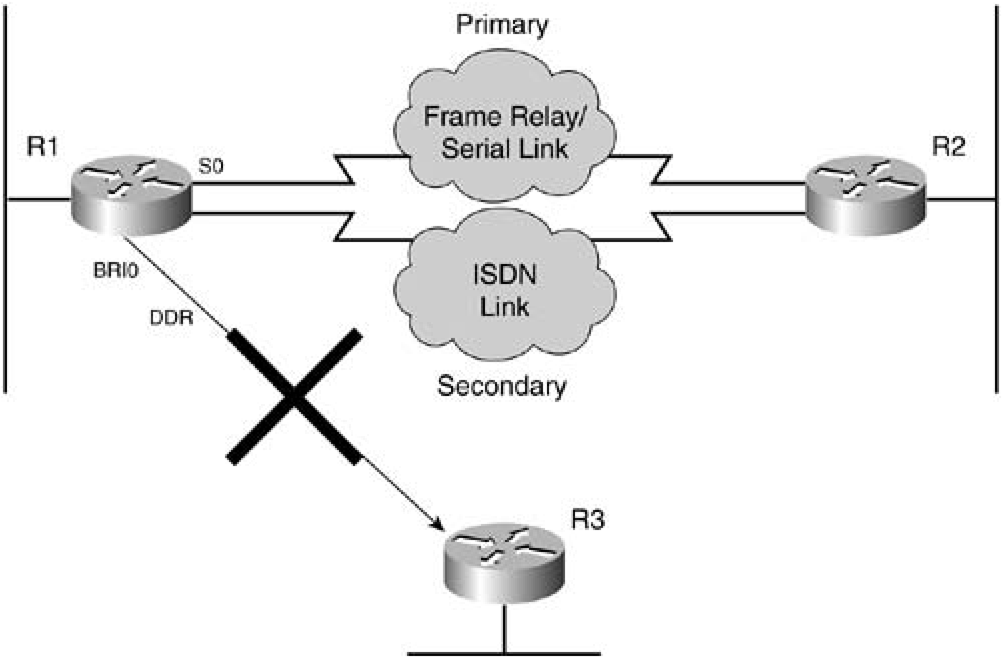


Using Physical Interfaces for Backup

As mentioned, when a backup interface is configured, it is placed in standby mode until it is activated. When in standby mode, it cannot be used. If the need arises for this interface to connect to another site, this is not possible.

InFigure 10-4, R1 has a primary link to R2 via its serial interface, S0. R1 also has an ISDN interface BRI0 configured as a backup interface. In this situation, BRI0 is placed in standby mode and is idle. If the network administrator wants to use the BRI0 interface to connect to R3, because the BRI0 interface is in standby mode, it is not possible. This is a limitation of using a physical interface for backup. The workaround is to use dialer profiles, as described in the following section.

Figure 10-4. Using a Physical Interface for Backup



Using Dialer Profiles for Backup

One way of overcoming the shortcoming just discussed is to use dialer profiles. Using dialer profiles, the ISDN BRI interface can be used to back up the primary serial interface. It can also be used to simultaneously connect to R3 via DDR.

You can configure one dialer profile to act as the backup interface. It is then placed in standby mode, as just described. You can then configure another dialer profile for Legacy DDR to connect to R3. As soon as the two dialer profiles have been configured, the physical BRI interface has to be made a member of both dialer pools.

Floating Static Routes as a Backup

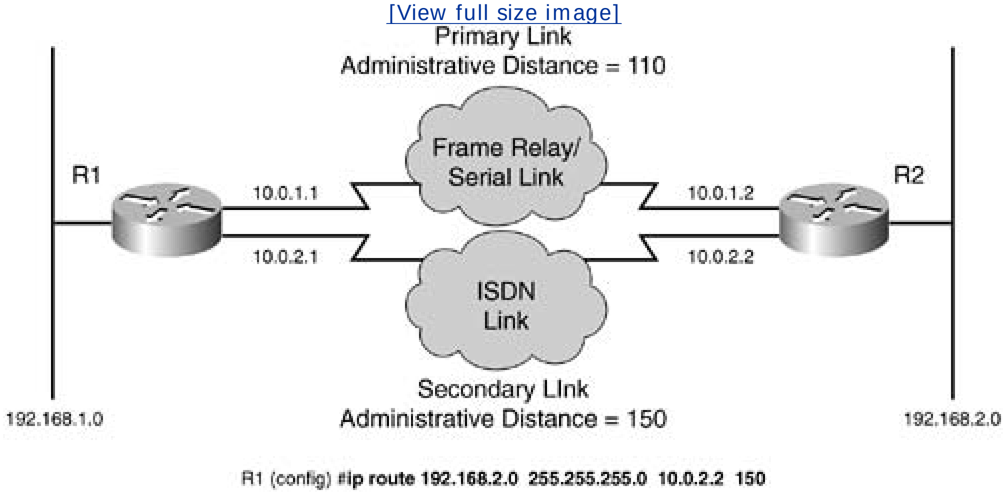
Floating static routes can also be used to back up a primary link. With this method, a static route is configured to the destination network whose administrative distance is greater than that of the dynamic route.

In the scenario shown in Figure 10-5, R1 is connected to R2 via a primary serial link as well as an ISDN BRI interface. OSPF is being used to advertise the Ethernet networks across the serial link. On R1, a static route is configured to Ethernet network 192.168.2.0 /24 via the BRI interface. This static route is configured as follows:

```
R2(config)#ip route 192.168.2.0 255.255.255.0 10.0.2.2 150
```

Figure 10-5. Using a Floating Static Route for Backup

[View full size image]


Primary Link
Administrative Distance = 110
Frame Relay/
Serial Link
R1
R2
10.0.1.1
10.0.1.2
10.0.2.1
10.0.2.2
ISDN
Link
Secondary LInk
Administrative Distance = 150
192.168.1.0
192.168.2.0
R1 (config) #ip route 192.168.2.0 255.255.255.0 10.0.2.2 150

This static route is configured with an administrative distance of 150. Because the administrative distance of the OSPF route (110) is more attractive, the static route is not used. If the primary link fails, the dynamic route is removed, and the static route is installed in the routing table.

Routing with the Load Backup Feature

When the load backup feature is on, load sharing occurs in different ways, depending on the routing protocol used.

Load Backup with OSPF

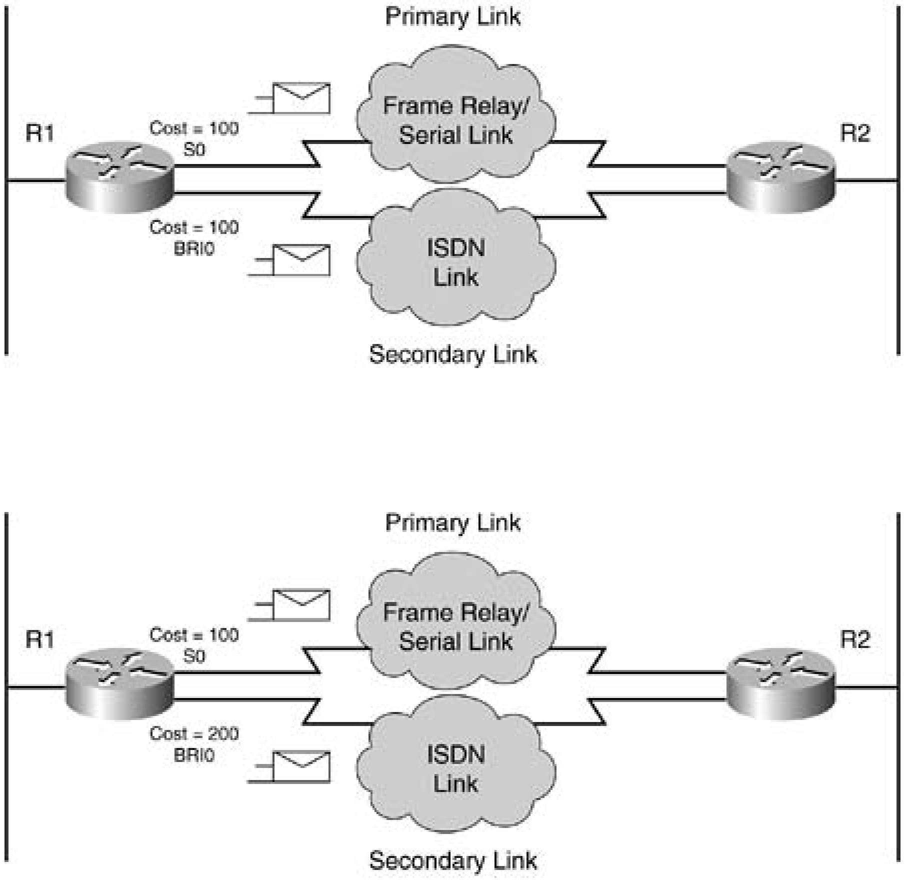
If both the primary and backup links are up at the same time and OSPF is the routing protocol being used, the load backup feature tries to load-share between the two links.

However, this is dependent on the respective costs of the two links. Because cost is the deciding metric in OSPF, it tries to pick the path with the lesser cost. So if load sharing is to occur, both links must have an equal cost. If one path has a lesser cost, all traffic uses that link.

OSPF load-shares only if both paths have an equal cost. This is illustrated in Figure 10-6.

Figure 10-6. Load Sharing with OSPF

[View full size image]



InFigure 10-6, if cost is equal on both links, traffic is sent over both links. However, if one path's cost is less, all traffic is sent over that link.

Load Backup with IGRP and EIGRP

If IGRP or EIGRP is configured, and both the primary and backup links are up, the load backup feature tries to load-share between the two links. However, the metric of both the links must be equal for this to happen. This is similar to the behavior of OSPF.

However, the variance command can be used in the case of IGRP and EIGRP to load-share between links of unequal metrics.

The*multiplier* number specified after the variance command determines which paths to use. For instance, in a simple scenario, if the number is 2, even if the backup path is twice as worse as the primary path, it is used to load-share traffic. This is usually dependent on other factors as well.

Scenarios

This section provides two scenarios of how backup lines can be configured to provide redundancy fo primary lines. Each scenario outlines the steps involved in the setup. The results are shown and the verified. Finally, the complete configuration of the routers is shown.

Scenario 10-1: Configuring Dial Backup for Primary Line Failures

In this scenario, Routers R1 and R2 are connected via a serial link. The serial link is the primary lin The two routers are also connected via an ISDN line, which is designated as the backup link. This backup link provides redundancy in case the primary link fails.

The following steps are required to configure dial backup for line failures:

Step 1. Define the primary interface:

```
Router(config)#interfacetype number
```

Step 2. While in interface configuration mode on the primary interface, define the backup interface to be used:

```
Router(config-if)#backup interfacetype number
```

Step 3. Specify how long to wait after the primary link goes down before enabling the backu link:

```
Router(config-if)#backup delay {enable-delay | never} {disable-delay |
  never}
```

NOTE

This assumes that you have already successfully configured both the primary and backup links

Example 10-1 illustrates the configuration.

Example 10-1. Configuring Dial Backup for Primary Link Failures

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int s0/0
R1(config-if)#backup interface bri0/0
R1(config-if)#
00:39:163208757247: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0/0, TEI 66
  changed to down
00:39:158913789952: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BR0/0, TEI 66
  changed to down
00:39:37: %LINK-5-CHANGED: Interface BRI0/0, changed state to standby mode
R1(config-if)#backup delay 5 10
R1(config-if)#^Z
```

The commands in Example 10-1 designate BRI0/0 as the backup interface for S0/0. The backup lin set to come up 5 seconds after the primary link goes down and is disabled 10 seconds after the pri link comes back up.

It is noteworthy that the backup interface immediately gets placed in standby mode when the bac interface command is issued.

Example 10-2 shows that the BRI0/0 interface is now in standby mode.

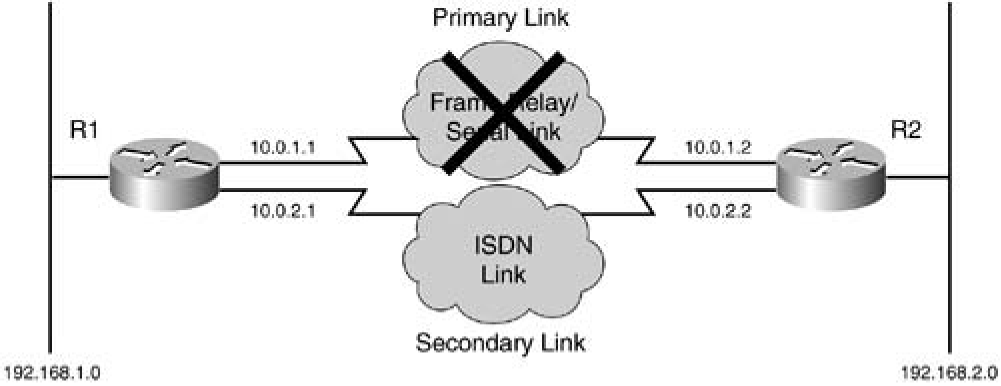
Example 10-2. Backup Interface in Standby Mode

```
R1#show interface bri0/0
BRI0/0 is standby mode, line protocol is down
  Hardware is PQUICC BRI with U interface
  Internet address is 10.0.2.1/24
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Last input 00:05:55, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/16 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     606 packets input, 2497 bytes, 0 no buffer
     Received 1 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     607 packets output, 2502 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
2 carrier transitions
```

The process is illustrated in Figure 10-7. The primary link between Routers R1 and R2 fails. The ba ISDN interface is then brought up to restore connectivity.

Figure 10-7. Primary Link Failure Topology

[View full size image]



Example 10-3 shows what happens on R1 when a primary line fails.

Example 10-3. Backup Interface Comes Up When the Primary Fails

```
R1#
02:27:31: %LINK-3-UPDOWN: Interface Serial0/0, changed state to down
02:27:31: %OSPF-5-ADJCHG: Process 111, Nbr 10.0.2.2 on Serial0/0 from FULL to
  DOWN, Neighbor Down: Interface down or detached
02:27:32: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
  state to down
02:27:36: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to down
02:27:36: %LINK-3-UPDOWN: Interface BRI0/0:2, changed state to down
02:27:156792760292: %ISDN-6-LAYER2UP: Layer 2 for Interface BR0/0, TEI 66 changed
  to up
02:27:36: %LINK-3-UPDOWN: Interface BRI0/0, changed state to up
02:28:02: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to up
```

```
02:28:03: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0/0:1, changed state
  to up
02:28:08: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 4082222222
02:28:12: %OSPF-5-ADJCHG: Process 111, Nbr 10.0.2.2 on BRI0/0 from LOADING to
  FULL, Loading Done
```

InExample 10-3, you can see that when the s0/0 link on R1 goes down, the BRI0/0 interface, whicl configured as its backup, comes up to take its place. When the BRI0/0 interface comes up, it estab an OSPF adjacency with R2 so that routing updates can be exchanged.

Running the show interface command on the backup interface tells you that it is now active, as sl inExample 10-4 . Note that it is no longer in standby mode.

Example 10-4. Backup Interface Is Now Active

```
R1#show interface bri0/0
BRI0/0 is up, line protocol is up (spoofing)
  Hardware is PQUICC BRI with U interface
  Internet address is 10.0.2.1/24
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Last input 00:00:00, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/16 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
```

```
     703 packets input, 2926 bytes, 0 no buffer
     Received 3 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     704 packets output, 2931 bytes, 0 underruns
     0 output errors, 0 collisions, 3 interface resets
     0 output buffer failures, 0 output buffers swapped out
     5 carrier transitions
```

When the primary line is restored, the backup again transitions back to standby mode, as shown ir Example 10-5. Note in the output of the show ip route command that the remote network 192.16 /24 is now advertised via the serial link.

The output of the show ip ospf neighbor command, shown in Example 10-5, shows that R2 is vis only via the primary serial link.

Example 10-5. Primary Link Gets Restored

```
R1#
02:30:03: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
02:30:04: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
  state to up
02:30:13: %ISDN-6-DISCONNECT: Interface BRI0/0:1  disconnected from 4082222222 ,
  call lasted 131 seconds
02:30:13: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to down
02:30:13: %OSPF-5-ADJCHG: Process 111, Nbr 10.0.2.2 on BRI0/0 from FULL to DOWN,
  Neighbor Down: Interface down or detached
02:30:55834574848: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0/0, TEI 66
  changed to down
02:30:55834574848: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BR0/0, TEI 66
  changed to down
```

```
02:30:13: %OSPF-5-ADJCHG: Process 111, Nbr 10.0.2.2 on Serial0/0 from LOADING to
  FULL, Loading Done
02:30:13: %LINK-5-CHANGED: Interface BRI0/0, changed state to standby mode
02:30:14: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0/0:1, changed state
  to down
R1#
R1#
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.0.1.0 is directly connected, Serial0/0
C    192.168.1.0/24 is directly connected, FastEthernet0/0
O    192.168.2.0/24 [110/74] via 10.0.1.2, 00:14:57, Serial0/0
R1#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.0.2.2          1   FULL/  -        00:00:33    10.0.1.2        Serial0/0
R1#
```

Example 10-6 provides the complete configuration of R1 for backup in the case of a primary link fa

Example 10-6. R1's Configuration

```
R1#show running-config
Building configuration...

Current configuration : 1070 bytes
!
version 12.1

hostname R1

!
isdn switch-type basic-net3
!
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 no ip redirects
 speed 100
 full-duplex
!
interface Serial0/0
backup delay 5 10
backup interface BRI0/0
 ip address 10.0.1.1 255.255.255.0
 no ip redirects
 clockrate 512000
!
interface BRI0/0
 ip address 10.0.2.1 255.255.255.0
 dialer map ip 10.0.2.2 broadcast 4082222222
 dialer-group 1
 isdn switch-type basic-net3
!
router ospf 111
 log-adjacency-changes
 network 10.0.1.0 0.0.0.255 area 0
 network 10.0.2.0 0.0.0.255 area 0
 network 192.168.1.0 0.0.0.255 area 0

ip classless

dialer-list 1 protocol ip permit

end
```

Example 10-7 provides the complete configuration of R2 for backup in the case of a primary link fa

Example 10-7. R2's Configuration

```
R2#show running-config
Building configuration...

Current configuration : 1295 bytes
```

```
version 12.1

hostname R2

!

ip subnet-zero

!

isdn switch-type basic-net3

!

interface Ethernet0/0

 ip address 192.168.2.1 255.255.255.0

 no ip redirects

!

interface Serial0/0

 backup delay 5 10

 backup interface BRI1/0

 ip address 10.0.1.2 255.255.255.0

 no ip redirects

 no fair-queue

!

interface BRI1/0

 ip address 10.0.2.2 255.255.255.0

 dialer map ip 10.0.2.1 broadcast 4081111111

 dialer-group 1

 isdn switch-type basic-net3

!


router ospf 111

 log-adjacency-changes

 network 10.0.1.0 0.0.0.255 area 0
 network 10.0.2.0 0.0.0.255 area 0

 network 192.168.2.0 0.0.0.255 area 0

!

ip classless

!

dialer-list 1 protocol ip permit

!

end
```

Scenario 10-2: Configuring Dial Backup for Load Sharing

In this scenario, Routers R1 and R2 are connected via a serial line that is designated as the primar The two routers are also connected via an ISDN line that is the backup link. This backup link is sup to be activated when the load on the primary line crosses a set threshold.

The following steps are required to configure dial backup to support high loads on the primary link

Step 1. Define the primary interface:

```
Router(config)#interfacetype number
```

Step 2. While in interface configuration mode on the primary interface, define the backup interface to be used:

```
Router(config-if)#backup interfacetype number
```

Step 3. Specify the traffic load threshold at which the backup link is to be activated:

```
Router(config-if)#backup load {enable-threshold | never} {disable-load |
  never}
```

This assumes that you have already successfully configured both the primary and backup links.

NOTE

Because load is calculated on an interface basis, the backup load command cannot be used on subinterfaces.

Example 10-8 illustrates the configuration.

Example 10-8. Configuring Dial Backup to Support Primary Links

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface serial0/0
R1(config-if)#backup interface bri0/0
R1(config-if)#
03:03:206158430208: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0/0, TEI 66
  changed to down
03:03:206158430208: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BR0/0, TEI 66
  changed to down
03:03:48: %LINK-5-CHANGED: Interface BRI0/0, changed state to standby mode
```

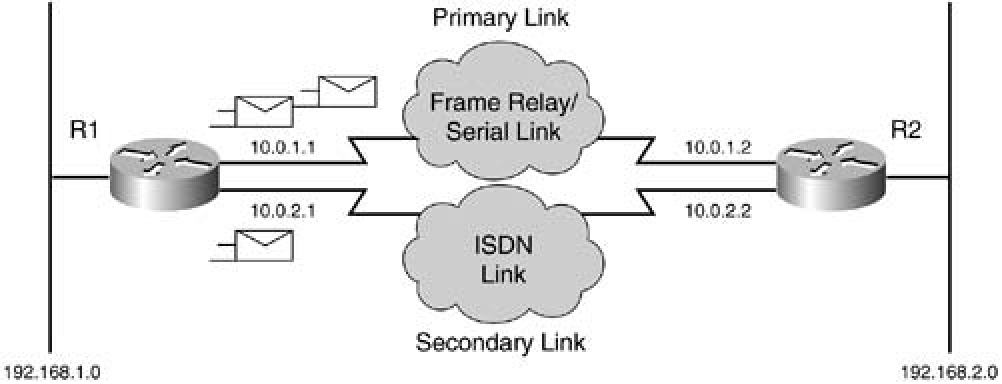
```
R1(config-if)#

R1(config-if)#

R1(config-if)#backup load 25 20

R1(config-if)#^Z
```

InExample 10-8, the S0/0 interface is supported by the BRI0/0 interface. Note that BRI0/0 is agair placed in standby mode. Also, the backup load command specifies that the BRI 0/0 interface will activated when the load on the primary link exceeds 25 percent and will be deactivated when the l drops below 20 percent.

Figure 10-8 illustrates how the ISDN interfaces on Routers R1 and R2 are used to provide backup f primary serial interfaces.

Figure 10-8. Primary Link Support Topology

[View full size image]



When the load on the primary link is below the specified threshold, the backup link is in standby m On the primary link you can see the configured parameters, as shown in Example 10-9.

Example 10-9. Primary and Backup Interfaces with Low Traffic

```
R1#show interface s0/0

Serial0/0 is up, line protocol is up

  Hardware is PowerQUICC Serial

  Internet address is 10.0.1.1/24

  Backup interface BRI0/0, failure delay 0 sec, secondary disable delay 0 sec,
    kickin load 25%, kickout load 20%
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:06, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     19 packets input, 1913 bytes, 0 no buffer
     Received 7 broadcasts, 0 runts, 0 giants, 0 throttles
     1 input errors, 0 CRC, 1 frame, 0 overrun, 0 ignored, 0 abort
     21 packets output, 3143 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up

R1#show interface bri0/0
BRI0/0 is standby mode, line protocol is down
  Hardware is PQUICC BRI with U interface
  Internet address is 10.0.2.1/24
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/16 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 4 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

When the load on the primary link exceeds the configured threshold, the backup interface is broug action, as shown in Example 10-10.

Example 10-10. Backup Interface Is Brought Up

```
R1#
00:08:00: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to down
00:08:00: %LINK-3-UPDOWN: Interface BRI0/0:2, changed state to down
00:08:00: %LINK-3-UPDOWN: Interface BRI0/0, changed state to up
```

```
00:08:4294967295: %ISDN-6-LAYER2UP: Layer 2 for Interface BR0/0, TEI 66 changed

  to up

00:08:01: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0/0:1, changed state

  to down

00:08:01: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0/0:2, changed state

  to down

R1#
```

Example 10-11 shows that the load on the primary link has exceeded the specified threshold. Also, you look at the backup link, you can see that it is now up.

Example 10-11. Primary and Backup Interfaces When the Load Threshol Exceeded

```
R1#show interface s0/0

Serial0/0 is up, line protocol is up

  Hardware is PowerQUICC Serial

  Internet address is 10.0.1.1/24

  Backup interface BRI0/0, failure delay 0 sec, secondary disable delay 0 sec,

    kickin load 25%, kickout load 20%

  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,

    reliability 255/255, txload 64/255, rxload 1/255

  Encapsulation HDLC, loopback not set

  Keepalive set (10 sec)

  Last input 00:00:00, output 00:00:00, output hang never

  Last clearing of "show interface" counters never

  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 148

  Queueing strategy: weighted fair

  Output queue: 63/1000/64/148 (size/max total/threshold/drops)
```

```
     Conversations  1/2/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 392000 bits/sec, 426 packets/sec
     124 packets input, 8885 bytes, 0 no buffer
     Received 63 broadcasts, 0 runts, 0 giants, 0 throttles
     1 input errors, 0 CRC, 1 frame, 0 overrun, 0 ignored, 0 abort
     215234 packets output, 24533109 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up


R1#show interface bri0/0
BRI0/0 is up, line protocol is up (spoofing)
  Hardware is PQUICC BRI with U interface
  Internet address is 10.0.2.1/24
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Last input 00:00:00, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/16 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     20 packets input, 99 bytes, 0 no buffer
     Received 1 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     22 packets output, 139 bytes, 0 underruns
     0 output errors, 0 collisions, 5 interface resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions
```

When the load drops below the specified disable load, the backup interface is again put back in sta mode, as shown in Example 10-12.

Example 10-12. Backup Interface Is Put Back in Standby Mode

```
R1#
00:11:240518168575: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BRI0/0, TEI 66
  changed to down
00:11:236223201280: %ISDN-6-LAYER2DOWN: Layer 2 for Interface BR0/0, TEI 66
  changed to down
00:11:55: %LINK-5-CHANGED: Interface BRI0/0, changed state to standby mode
R1#
R1#
R1#
R1#show interface bri0/0
BRI0/0 is standby mode, line protocol is down
  Hardware is PQUICC BRI with U interface
  Internet address is 10.0.2.1/24
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Last input 00:02:37, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/16 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     83 packets input, 391 bytes, 0 no buffer
     Received 1 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     90 packets output, 531 bytes, 0 underruns
     0 output errors, 0 collisions, 5 interface resets
     0 output buffer failures, 0 output buffers swapped out
     2 carrier transitions
```

Example 10-13 provides the complete configuration of R1 for load backup.

Example 10-13. R1's Configuration for Load Backup

```
R1#show running-config
Building configuration...

Current configuration : 1070 bytes
!
```

```
version 12.1

!

hostname R1

!

ip subnet-zero

!

isdn switch-type basic-net3

!

interface FastEthernet0/0

 ip address 192.168.1.1 255.255.255.0

 no ip redirects

 speed 100

 full-duplex

!

interface Serial0/0

backup interface BRI0/0

backup load 25 20

 ip address 10.0.1.1 255.255.255.0

 no ip redirects

 clockrate 512000

!

interface BRI0/0

 ip address 10.0.2.1 255.255.255.0

 dialer map ip 10.0.2.2 broadcast 4082222222

 dialer-group 1

 isdn switch-type basic-net3

!

router ospf 111
```

```
 log-adjacency-changes
 network 10.0.1.0 0.0.0.255 area 0
 network 10.0.2.0 0.0.0.255 area 0
 network 192.168.1.0 0.0.0.255 area 0
!
ip classless
!
dialer-list 1 protocol ip permit
!
end
```

Example 10-14 provides the complete configuration of R2 for load backup.

Example 10-14. R2's Configuration for Load Backup

```
R2#show running-config
Building configuration...

version 12.1
!
hostname R2
!
ip subnet-zero
!
isdn switch-type basic-net3
!
interface Ethernet0/0
 ip address 192.168.2.1 255.255.255.0
 no ip redirects

!

interface Serial0/0

backup load 25 20

backup interface BRI1/0

 ip address 10.0.1.2 255.255.255.0

 no ip redirects

 no fair-queue

!

interface BRI1/0

 ip address 10.0.2.2 255.255.255.0

 dialer map ip 10.0.2.1 broadcast 4081111111

 dialer-group 1

 isdn switch-type basic-net3

!

router ospf 111

 log-adjacency-changes

 network 10.0.1.0 0.0.0.255 area 0

 network 10.0.2.0 0.0.0.255 area 0

 network 192.168.2.0 0.0.0.255 area 0

!

ip classless

!

dialer-list 1 protocol ip permit

!

end
```

Scenario 10-3: Configuring Dialer Profiles for Backup

As mentioned, the use of dialer profiles allows the use of a physical interface for both backup and D connections to another site. In this scenario, Routers R1 and R2 are connected via a primary serial The routers are also connected via an ISDN line, which has been designated as the backup link. Th scenario shows the use of dialer profiles to provide backup for the primary link.

The following steps are required to configure the backup using dialer profiles:

Step 1. Create and configure a dialer interface:

```
Router(config)#interface dialernumber

Router(config-if)#encapsulation ppp

Router(config-if)#dialer remote-namename

Router(config-if)#dialer stringstring

Router(config-if)#dialer poolnumber

Router(config-if)#dialer-groupnumber
```

Step 2. Specify the physical interface that will support the backup. Configure it for PPP encapsulation:

```
Router(config)#interfacetype number

Router(config-if)#encapsulation ppp

Router(config-if)#ppp authentication chap
```

Step 3. Make the backup interface a member of the dialer pool:

```
Router(config-if)#dialer pool membernumber
```

Step 4. Specify the primary interface to be backed up. Specify the dialer interface configurec Step 1 to be used for backup:

```
Router(config)#interfacetype number

Router(config-if)#backup interface dialernumber
```

Step 5. Specify what kind of backup is required. For primary link failures, use the backup d command. For load sharing, use the backup load command:

```
Router(config-if)#backup delay {enable-delay | never} {disable-delay |

  never}

Router(config-if)#backup load {enable-threshold | never} {disable-load |

  never}
```

Step 6. Specify interesting traffic that will bring up the backup interface using the dialer-lis command:

```
Router(config)#dialer-listdialer-groupprotocolprotocol-name {permit |

  deny | listaccess-list-number | access-group}
```

Example 10-15 shows the use of dialer profiles for backup.

Example 10-15. Configuring the Dialer Interface for Dialer Profiles

```
R1(config)#interface dialer 0

R1(config-if)#ip unnumbered loopback0

R1(config-if)#encapsulation ppp

R1(config-if)#dialer remote-name R2

R1(config-if)#dialer pool 1

R1(config-if)#dialer string 4082222222

R1(config-if)#dialer-group 1

R1(config-if)#exit

R1(config)#dialer-list 1 protocol ip permit

R1(config-if)#^Z
```

InExample 10-15, the dialer interface is configured. You can see how most of the legacy DDR commands that used to be configured on a BRI interface are now configured here. You can also see thedialer-list command is used to define interesting traffic. The number following the dialer-list command should be the dialer-group number specified under the interface.

Example 10-16 shows the configuration of the physical BRI interface used for backup. Note that the dialer pool-member command is used to match the physical BRI backup interface to the logical d interface.

Example 10-16. Configuring the Physical BRI Interface for Dialer Profiles

```
R1(config)#interface bri0/0
```

```
R1(config-if)#encapsulation ppp
R1(config-if)#dialer pool-member 1
R1(config-if)#ppp authentication chap
R1(config-if)#^Z
```

Example 10-17 shows the configuration of the primary link to be backed up. Notice that the backup interface specified is the virtual dialer interface instead of the physical BRI interface.

Example 10-17. Configuring the Primary Interface for Dialer Profiles

```
R1(config)#int serial 0/0
R1(config-if)#backup interface dialer 0
R1(config-if)#backup delay 5 10
R1(config-if)#^Z
```

As soon as the configuration steps are complete, running show commands on the various interface gives the results shown in Example 10-18.

Example 10-18. Output of show interface Commands on the Interfaces

```
R1#show interface s0/0
Serial0/0 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 10.0.1.1/24
  Backup interface Dialer0, failure delay 5 sec, secondary disable delay 10 sec,
  kickin load not set, kickout load not set
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:03, output 00:00:03, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/2/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     5458 packets input, 365859 bytes, 0 no buffer
     Received 2900 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     5472 packets output, 364550 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 output buffer failures, 0 output buffers swapped out
     5 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up

R1#show interface dialer0
Dialer0 is standby mode, line protocol is down
  Hardware is Unknown
  Interface is unnumbered. Using address of Loopback0 (10.0.3.1)
  MTU 1500 bytes, BW 56 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set
  DTR is pulsed for 1 seconds on reset
  Last input never, output never, output hang never
  Last clearing of "show interface" counters 00:02:55
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/16 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes
     0 packets output, 0 bytes

R1#show interface bri0/0
BRI0/0 is up, line protocol is up (spoofing)
  Hardware is PQUICC BRI with U interface
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 254/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set
  Last input 00:00:00, output never, output hang never
  Last clearing of "show interface" counters 00:01:58
  Input queue: 0/75/1/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/16 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     141 packets input, 570 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     1 input errors, 1 CRC, 0 frame, 0 overrun, 0 ignored, 1 abort
     40 packets output, 166 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     6 carrier transitions
```

Notice that the serial interface now specifies the dialer interface as its backup. Also note that while dialer interface is in standby mode, the BRI interface is up.

When the primary link goes down, the dialer interface comes up, as shown in Example 10-19.

Example 10-19. Backup Using Dialer Profiles

```
R1#
07:50:04: %LINK-3-UPDOWN: Interface Serial0/0, changed state to down
07:50:04: %OSPF-5-ADJCHG: Process 111, Nbr 192.168.2.1 on Serial0/0 from FULL to
  DOWN, Neighbor Down: Interface down or detached
07:50:05: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
  state to down
07:50:11: %LINK-3-UPDOWN: Interface Dialer0, changed state to up
07:50:49411108400: %LINK-3-UPDOWN: Interface BRI0/0:1, changed state to up
07:50:51539607551: %DIALER-6-BIND: Interface BR0/0:1 bound to profile Di0
07:50:14: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0/0:1, changed state
07:50:17: %ISDN-6-CONNECT: Interface BRI0/0:1 is now connected to 4082222222 R2
07:50:31: %OSPF-5-ADJCHG: Process 111, Nbr 192.168.2.1 on Dialer0 from LOADING to
  FULL, Loading Done
R1#
```

InExample 10-19, you can see that when the primary serial link goes down, the dialer interface co up. Also, you can see that OSPF reconverges to run over the dialer interface. This is demonstrated the output of the show ip route command, as shown in Example 10-20.

Example 10-20. Output of show ip route After the Backup Link Is Up

```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.3.0/24 is directly connected, Loopback0
C       10.0.3.2/32 is directly connected, Dialer0
C    192.168.1.0/24 is directly connected, FastEthernet0/0
O    192.168.2.0/24 [110/1795] via 10.0.3.2, 00:01:08, Dialer0
R1#ping 192.168.2.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/33/36 ms
R1#
```

The output of show ip route reveals that the remote Ethernet segment 192.168.2.0 /24 can now b reached via the dialer interface. Pinging R2's Ethernet interface 192.168.2.1 via the backup link sh that the link is up, as shown in Example 10-20.

The complete configuration of R1 for dialer profiles is shown in Example 10-21.

Example 10-21. R1's Configuration for Dialer Profiles

```
R1#show running-config
Building configuration...
!
version 12.1
!
hostname R1
!
username R2 password 0 cisco
!
isdn switch-type basic-net3
!
interface Loopback0
 ip address 10.0.3.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 no ip redirects
 speed 100
 full-duplex
!
interface Serial0/0
 backup delay 5 10
 backup interface Dialer0
 ip address 10.0.1.1 255.255.255.0
 no ip redirects
 clockrate 512000
!
interface BRI0/0
 no ip address
 encapsulation ppp
 dialer pool-member 1
 isdn switch-type basic-net3
 ppp authentication chap
!
interface Dialer0
 ip unnumbered Loopback0
 encapsulation ppp
 dialer pool 1
 dialer remote-name R2
 dialer string 4082222222
 dialer-group 1
!
router ospf 111
 log-adjacency-changes
 network 10.0.1.0 0.0.0.255 area 0
 network 10.0.2.0 0.0.0.255 area 0
 network 10.0.3.1 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
!
ip classless
!
```

```
dialer-list 1 protocol ip permit
!
end
```

The complete configuration of R2 for dialer profiles is shown in Example 10-22.

Example 10-22. R2's Configuration for Dialer Profiles

```
R2#show running-config
version 12.1
!
hostname R2
!
username R1 password 0 cisco
!
isdn switch-type basic-net3
!
interface Loopback0
 ip address 10.0.3.2 255.255.255.0
!
interface Ethernet0/0
 ip address 192.168.2.1 255.255.255.0
 no ip redirects
!
interface Serial0/0
 backup delay 5 10
 backup interface Dialer0
 ip address 10.0.1.2 255.255.255.0
 no ip redirects
 no fair-queue
!
interface BRI1/0
 no ip address
 encapsulation ppp
 dialer pool-member 1
 isdn switch-type basic-net3
 ppp authentication chap
!
interface Dialer0
 ip unnumbered Loopback0
 encapsulation ppp
 dialer pool 1
 dialer remote-name R1
 dialer string 4081111111
 dialer-group 1
!
router ospf 111
 log-adjacency-changes
 network 10.0.1.0 0.0.0.255 area 0
 network 10.0.2.0 0.0.0.255 area 0
 network 10.0.3.2 0.0.0.0 area 0
 network 192.168.2.0 0.0.0.255 area 0
!
ip classless
!
dialer-list 1 protocol ip permit
```

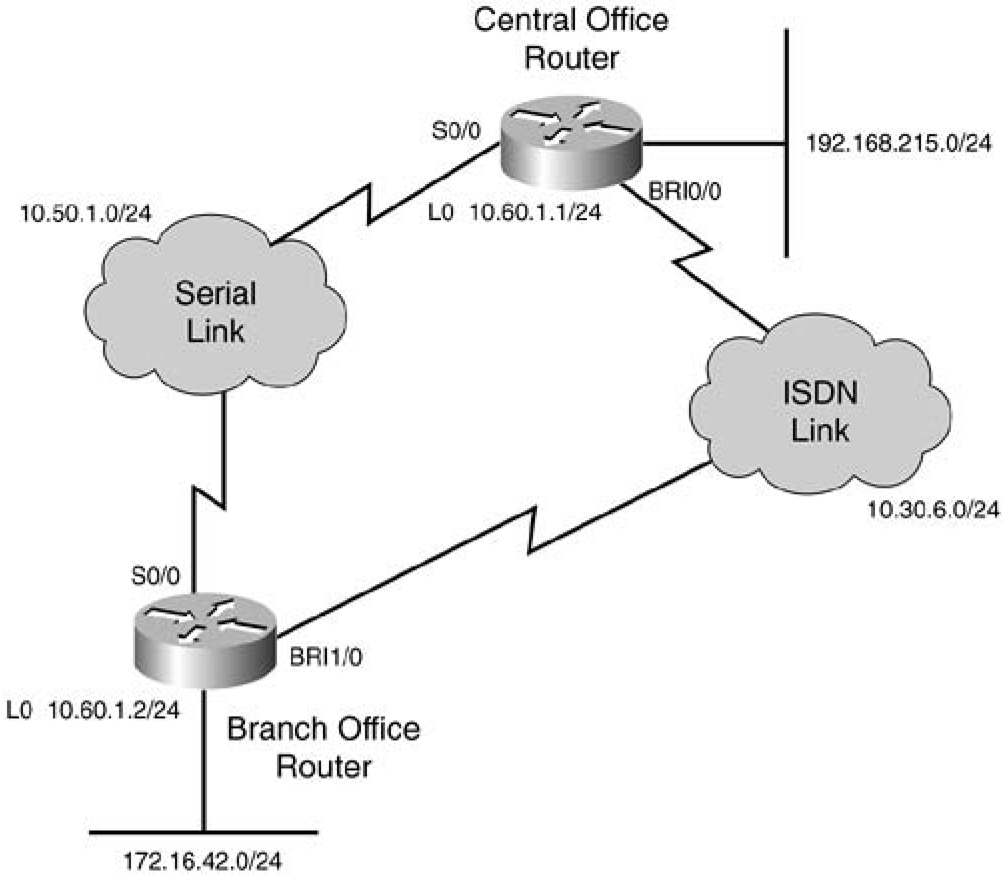
```
!

end
```

Practical Exercise: Enabling Backup for a Primary Link

A corporation has a branch office connected via a serial link of 512 kbps to the central office. The network administrator at the remote site wants to back up the primary serial link with an ISDN BRI link. Configure the backup link at the remote site such that the BRI link can be used for both backup operations and connections to other remote sites. OSPF should be configured on all interfaces. Verify connectivity by pinging the central Ethernet and loopback interfaces from the branch router.

Figure 10-9 illustrates this topology.

Figure 10-9. Practical Exercise Topology



Practical Exercise Solution

The solution for the practical exercise is shown in Example 10-23.

Example 10-23. Solution to the Practical Exercise

```
Branch#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Branch(config)#interface dialer 0
Branch(config-if)#ip unnumbered loopback0
Branch(config-if)#encapsulation ppp
Branch(config-if)#dialer remote-name Central
Branch(config-if)#dialer pool 1

Branch(config-if)#dialer-group 1
Branch(config-if)#exit
Branch(config)#dialer-list 1 protocol ip permit
Branch(config)#int bri1/0
Branch(config-if)#encapsulation ppp
Branch(config-if)#ppp authentication chap
Branch(config-if)#dialer pool-member 1
Branch(config-if)#exit
Branch(config)#int s0/0
Branch(config-if)#backup interface dialer0
Branch(config-if)#backup delay 5 5
Branch(config-if)#^Z
```

If the primary link is disabled, the backup link should come up, as shown in Example 10-24.

Example 10-24. Verifying Dial Backup Operation

```
Branch#
00:14:15: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed
  state to down
00:14:16: %LINK-3-UPDOWN: Interface Serial0/0, changed state to down
00:14:16: %OSPF-5-ADJCHG: Process 111, Nbr 10.60.1.1 on Serial0/0 from FULL to
  DOWN, Neighbor Down: Interface down or detached
00:14:94489280576: %LINK-3-UPDOWN: Interface BRI1/0:1, changed state to up
00:14:98784247807: %DIALER-6-BIND: Interface BR1/0:1 bound to profile Di0
00:14:23: %LINK-3-UPDOWN: Interface Dialer0, changed state to up
00:14:25: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI1/0:1, changed state
  to up
00:14:28: %ISDN-6-CONNECT: Interface BRI1/0:1 is now connected to 4081111111
  Central
00:14:34: %OSPF-5-ADJCHG: Process 111, Nbr 10.60.1.1 on Dialer0 from LOADING to
  FULL, Loading Done
```

The output of show ip route should show that OSPF is now running over the backup dialer interface, as shown in Example 10-25.

Example 10-25. Routing Over the Backup Link

```
Branch#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.42.0 is directly connected, Ethernet0/0
O    192.168.215.0/24 [110/1786] via 10.60.1.1, 00:00:28, Dialer0
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.60.1.1/32 is directly connected, Dialer0
C       10.60.1.0/24 is directly connected, Loopback0
```

You should be able to ping the remote Ethernet and loopback interfaces, as shown in Example 10-26.

Example 10-26. Verifying Connectivity to the Central Network

```
Branch#ping 192.168.215.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.215.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/33/36 ms
Branch#ping 10.60.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.60.1.2, timeout is 2 seconds:
```

```
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

The complete configuration of the branch router is shown in Example 10-27.

Example 10-27. Branch Router Configuration

```
Branch#show running-config
version 12.1
!
hostname Branch
!
username Central password 0 cisco
!
ip subnet-zero
isdn switch-type basic-net3
!
interface Loopback0
 ip address 10.60.1.2 255.255.255.0
!
interface Ethernet0/0
 ip address 172.16.42.1 255.255.255.0
!
interface Serial0/0
 backup delay 5 5
 backup interface Dialer0
 ip address 10.50.1.2 255.255.255.0
 no ip redirects
 no fair-queue
!
interface BRI1/0
 no ip address
 encapsulation ppp
 dialer pool-member 1
 isdn switch-type basic-net3
 ppp authentication chap
!
interface Dialer0
 ip unnumbered Loopback0
 encapsulation ppp
 dialer pool 1
 dialer remote-name Central
 dialer string 4081111111
 dialer-group 1
!
router ospf 111
 log-adjacency-changes
 network 10.50.1.0 0.0.0.255 area 0
 network 10.60.1.2 0.0.0.0 area 0
 network 172.16.42.0 0.0.0.255 area 0
!
ip classless
!
dialer-list 1 protocol ip permit
!
end
```

The complete configuration of the central router is shown in Example 10-28.

Example 10-28. Central Router Configuration

```
Central#show running-config
version 12.1
!
hostname Central
!
username Branch password 0 cisco
!
ip subnet-zero
!
isdn switch-type basic-net3
!
interface Loopback0
 ip address 10.60.1.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 192.168.215.1 255.255.255.0
 no ip redirects
 speed 100
 full-duplex
!
interface Serial0/0
 backup delay 5 5
 backup interface Dialer0
 ip address 10.50.1.1 255.255.255.0
 no ip redirects
 clockrate 512000
!
interface BRI0/0
 no ip address
 encapsulation ppp
 dialer pool-member 1
 isdn switch-type basic-net3
 ppp authentication chap
!
interface Dialer0
 ip unnumbered Loopback0
 encapsulation ppp
 dialer pool 1
 dialer remote-name Branch
 dialer string 4082222222
 dialer-group 1
!
router ospf 111
 log-adjacency-changes
 network 10.50.1.0 0.0.0.255 area 0
 network 10.60.1.1 0.0.0.0 area 0
 network 192.168.215.0 0.0.0.255 area 0
!
ip classless
!
dialer-list 1 protocol ip permit
```

```
!
end
```

Summary

With redundancy being a crucial component of today's networks, backup links are proving more and more valuable. This chapter covered their use to support permanent primary links. There are a number of ways to accomplish the goal of backup. Configuration examples were provided to show these different approaches. Specifically, primary link failures and primary link load support issues were addressed. Figures and verification steps for these scenarios were also shown. Finally, a practical exercise was provided to help your understanding of the concepts presented.

Review Questions

1: Is it possible to specify the backup load command on subinterfaces? Why or why not?

2: What two circumstances can trigger dial backup?

3: What is a drawback of using physical interfaces for backup?

4: Which interfaces can be used as backup interfaces?

5: What is one reason that ISDN interfaces are used mostly for backup interfaces instead of primary interfaces?

6: Which command specifies interesting traffic to bring up an ISDN interface?

7: Which command specifies the amount of time before a backup interface is activated in case of a primary link failure?

8: Which command specifies the load threshold at which a backup interface is brought up in case of load sharing?

9: What is a possible alternative to dial backup?

10: Which commands associate a virtual dialer interface with a physical interface when you configure dialer profiles?

Chapter 11. Managing Network Performance with Queuing and Compression

This chapter covers the following topics:

- Considerations for Traffic Prioritization
- Queuing Operations
- Configuring and Verifying Queuing
- Compression

Many networks today need to support a diverse mixture of applications and protocols. These applications can range from delay-sensitive traffic such as desktop videoconferencing to file transfers that use FTP. Because these different types of traffic share the same network infrastructure, they can negatively affect each other.

Depending on the applications and the overall available bandwidth, congestion can occur. Often this congestion occurs at routers where there is a disparity in speed between two interfaces. For instance, packets might arrive on a Fast Ethernet interface that need to go out on a low-speed WAN link. These packets might arrive faster than the router can send them. At this point, the need for congestion management arises.

Congestion management consists of the following:

- Prioritizing traffic so that applications such as videoconferencing are assigned a higher priority than FTP traffic
- Creating different queues for different priorities of traffic
- Assigning traffic to its appropriate queues
- The order in which these queues are serviced (the order in which traffic is sent)

Congestion management can ensure that even if congestion occurs, traffic can be sent in a prioritized manner so that network performance is not affected and the impact on users is minimized. Queuing mechanisms on routers are an important way of reducing congestion.

NOTE

Queuing is done only on output interfaces.

Considerations for Traffic Prioritization

The following are the main considerations for prioritizing traffic:

- Is there congestion in the network? If not, there is no need to prioritize traffic.
- Delay-sensitive traffic such as voice over IP (VoIP) and videoconferencing are more sensitive to delay and hence need a higher priority than FTP traffic.
- WAN links with speeds of T1/E1 or lower can benefit from prioritization.
- If a WAN is constantly congested, prioritization might not solve the problem. Additional bandwidth needs to be added.

Queuing Operations

If the network is congested, the need for queuing arises. There are four main types of queuing:

- First-in, first-out (FIFO) queuing— Traffic is not prioritized or classified. Packets are transmitted in the order in which they were received.
- Weighted Fair Queuing (WFQ)— An automated method that divides bandwidth fairly among the different types of traffic based on weight.
- Priority queuing (PQ)— A strict method in which high-priority packets always get priority over lower-priority traffic.
- Custom queuing (CQ)— With CQ, bandwidth can be proportionally assigned to the different types of traffic.

The last three methods are covered in this chapter.

Weighted Fair Queuing

WFQ is an automated mechanism that provides an equitable distribution of bandwidth to all network protocols. It helps ensure that available bandwidth is shared by all the protocols and that low-volume traffic is not detrimentally affected by higher-volume traffic consuming a large portion of the bandwidth.

Flow-based WFQ classifies traffic into flows based on certain characteristics in the packet header. These characteristics can include source and destination IP or MAC addresses, protocol, source and destination port numbers, and type of service (ToS) values.

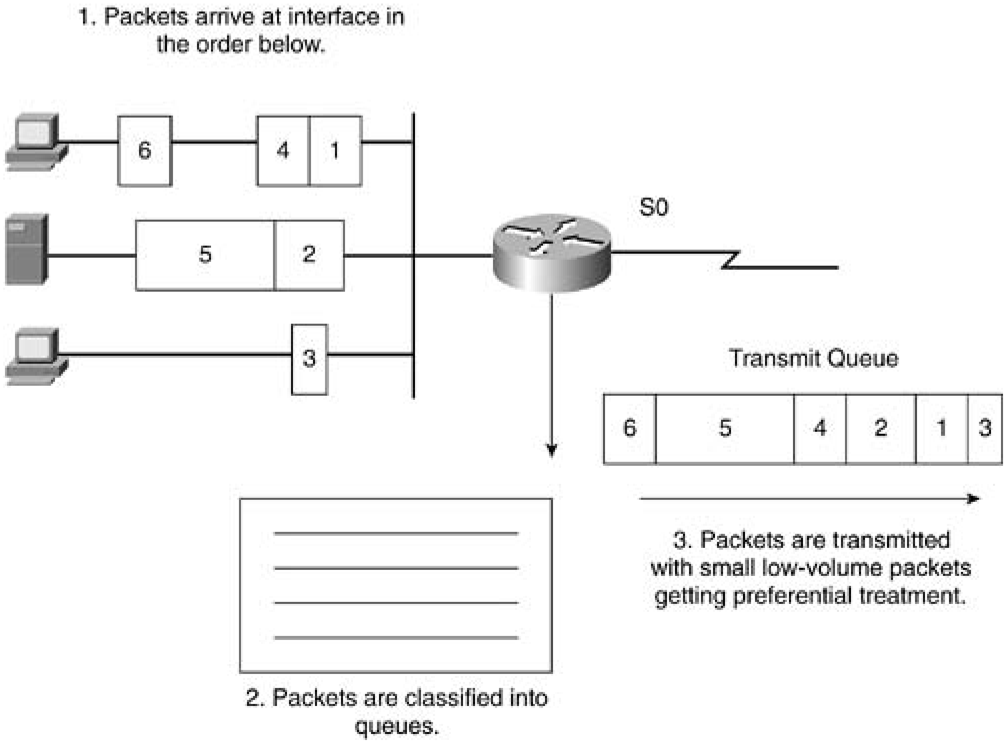
These flows are then classified as either low-volume or high-volume. By definition, the WFQ algorithm gives low-volume flows preferential treatment over high-volume flows. After the low-volume flows have been sent, high-volume flows share the remaining bandwidth. This method ensures that low-bandwidth conversations, which make up the majority of traffic, are serviced in a timely manner.

NOTE

WFQ is the default queuing mechanism for all physical interfaces whose bandwidth is E1 and lower. T1 links (1.544 Mbps) are popular in the United States, and E1 links (2.048 Mbps) are widely used in other parts of the world, including Europe.

The WFQ process is illustrated in Figure 11-1.

Figure 11-1. Weighted Fair Queuing


1. Packets arrive at interface in the order below.
6
4
1
5
2
3
S0
Transmit Queue
6
5
4
2
1
3
3. Packets are transmitted with small low-volume packets getting preferential treatment.
2. Packets are classified into queues.

Each packet's virtual time of delivery determines the order in which it is transmitted. This ensures that smaller packets are given preference, as demonstrated by Packet 3's being the first packet sent in Figure 11-1.

High-volume applications often generate series of packets of associated data. These are called *packet trains*. Packet trains can consume large amounts of bandwidth and starve lower-volume traffic in a FIFO queuing environment. WFQ provides an automatic, elegant solution to this problem.

Although WFQ might work well in a lot of environments, there are some caveats. For instance, it is not supported on interfaces that do tunneling or encryption. Also, certain interfaces such as ATM do not support WFQ.

Priority Queuing

Priority queuing is a mechanism that strictly enforces priority as the criterion for selecting which packets to send first on an interface. This method ensures that high-priority traffic is not delayed by less-important traffic.

When using priority queuing, you create four traffic queues:

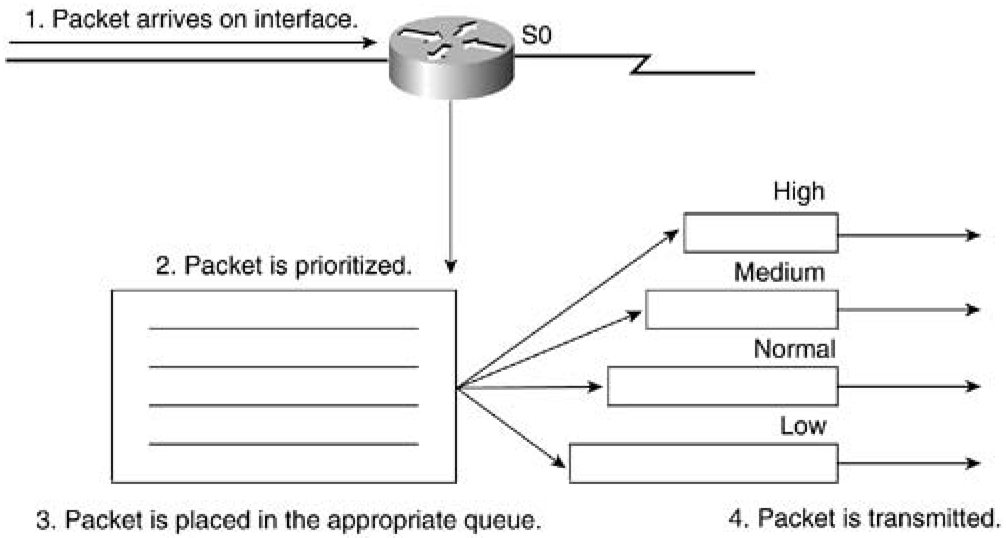
- High
- Medium
- Normal
- Low

Then you configure a set of filters to allow the router to place traffic in these four queues. These filters can be based on traffic characteristics such as protocol or TCP port number.

After the traffic is placed in the queues, the high-priority queue is always emptied before the medium-priority queue, and so on. This process is repeated every time a packet needs to be sent. This ensures that time-sensitive or mission-critical traffic is always given precedence over other traffic. However, note that medium-priority or low-priority packets are not serviced while packets are in the high-priority queue.

This process is illustrated in Figure 11-2.

Figure 11-2. Priority Queuing


1. Packet arrives on interface.
S0
2. Packet is prioritized.
High
Medium
Normal
Low
3. Packet is placed in the appropriate queue.
4. Packet is transmitted.

Priority queuing gives the network administrator the most control over which traffic gets forwarded. This is because the administrator defines the traffic filters. These filters, also called priority lists, assign traffic to the four queues.

Although priority queuing offers the most control over what traffic is transmitted first, it also requires some degree of manual configuration. Traffic prioritization and queue size are two things that the administrator needs to configure. This kind of static configuration cannot respond to a dynamic environment.

It should also be noted that although priority queuing is a good method of ensuring absolute priority for mission-critical traffic, there is a danger that lower-priority traffic could be drowned out. In a worst-case scenario, high-priority traffic could consume 100 percent of the bandwidth, and lower-priority traffic might not even be sent. Care should be taken that this does not happen.

Custom Queuing

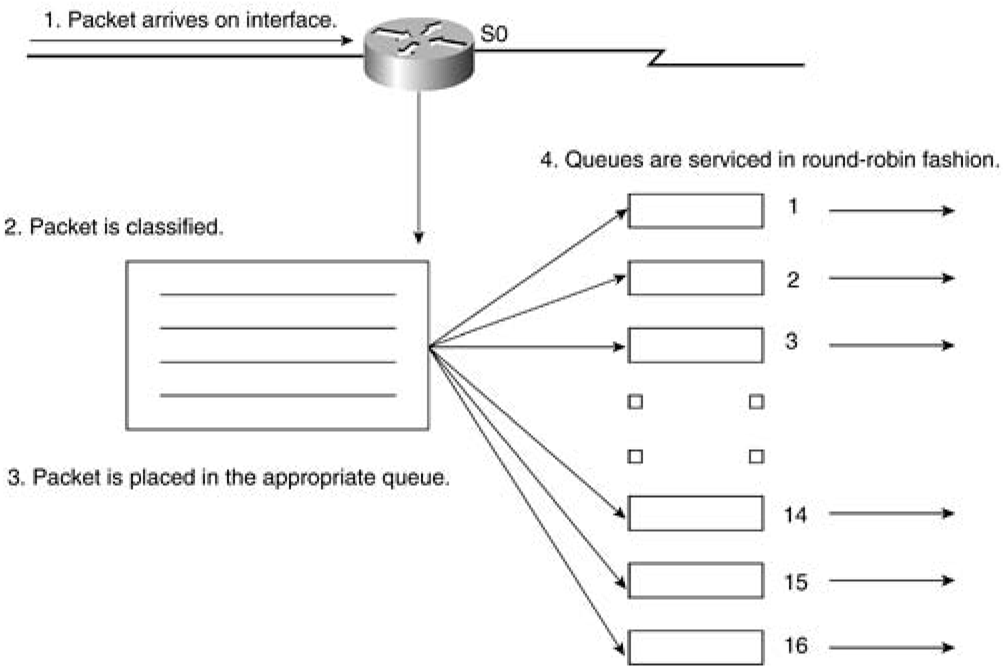
Custom queuing is a method that lets the network administrator guarantee bandwidth by giving queue space to each protocol. This overcomes a potential priority queuing problem in which lower-priority traffic languishes if higher-priority traffic needs to be sent.

Custom queuing has 16 queues to which you can assign traffic. You can define a set of traffic filters called custom queue lists to determine which protocol you want to place in a particular queue. You can also define how many bytes to transmit from each queue.

The queues are then serviced in round-robin fashion, with the specified number of bytes sent each time. As noted, you can set this byte count value. However, custom queuing does not fragment a packet to fit a queue's byte count. When the byte count has been reached, or when no more traffic needs to be sent, the next queue is serviced.

This process is described in Figure 11-3.

Figure 11-3. Custom Queuing


1. Packet arrives on interface.
S0
4. Queues are serviced in round-robin fashion.
2. Packet is classified.
3. Packet is placed in the appropriate queue.
1
2
3
14
15
16

Custom queuing can ensure that no one protocol starves others out of bandwidth. Also, if a particular protocol doesn't use the bandwidth allocated to it, the bandwidth can be dynamically used by other protocols.

However, like priority queuing, custom queuing requires some manual configuration. The byte count value for each queue has to be carefully selected so that the desired results are achieved. Also, because custom queuing configuration is static, it cannot adapt to a changing environment.

Configuring and Verifying Queuing

This section covers the configuration and verification of the three types of queuing just discussed. The various commands are covered. Examples are also provided to help you understand the concepts.

Weighted Fair Queuing

The following interface configuration mode command enables flow-based WFQ on an interface:

```
fair-queue [congestive-discard-threshold [dynamic-queues [reservable-queues]]]
```

The*congestive-discard-threshold* number is the threshold beyond which messages for high-volume traffic are not queued.

Example 11-1 configures WFQ with a congestive discard threshold of 64 on R1's serial 0/2 interface.

Example 11-1. Configuring Weighted Fair Queuing

```
R1#config terminal

R1#(config)#interface serial0/2

R1#(config)# (config-if)#fair-queue ?

  <1-4096>  Congestive Discard Threshold

  <cr>

R1(config-if)#fair-queue 64
```

NOTE

WFQ is the default queuing mechanism for interfaces with speeds of E1 and less.

Example 11-2 shows the different flows going through the serial 0/2 interface. Specifically, two flows are shown. The first is for Telnet traffic, and the second shows an FTP flow. Also, you can see that WFQ is the queuing strategy and that the congestive discard threshold is set to 64, as configured.

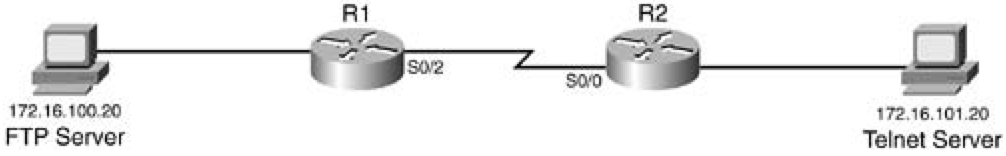
Example 11-2. Weighted Fair Queuing on an Interface

```
R1#show queue serial0/2
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 6/1000/64/0 (size/max total/threshold/drops)
     Conversations  2/3/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 42 kilobits/sec

  (depth/weight/total drops/no-buffer drops/interleaves) 1/32384/0/0/0
  Conversation 31, linktype: ip, length: 44
  source: 172.16.100.20, destination: 172.16.101.20, id: 0x6FA9, ttl: 127,
  TOS: 0 prot: 6, source port 3723, destination port 23

  (depth/weight/total drops/no-buffer drops/interleaves) 5/32384/0/0/0
  Conversation 147, linktype: ip, length: 1376
  source: 172.16.100.20, destination: 172.16.101.20, id: 0x6FA2, ttl: 127,
  TOS: 0 prot: 6, source port 20, destination port 1036
```

Figure 11-4 shows the topology relating to this configuration.

Figure 11-4. Weighted Fair Queuing Topology

[View full size image]



Example 11-3 shows R1's full configuration.

Example 11-3. R1's Full Configuration

```
R1#show running-config

Building configuration...


Current configuration : 1009 bytes

!

version 12.1

!

enable password cisco

!

hostname R1

!

!

ip subnet-zero

!

interface Ethernet0/0

 ip address 172.16.100.1 255.255.255.0

 half-duplex

!

interface Serial0/2
```

```
 ip address 10.1.2.1 255.255.255.252
!
router ospf 1
 log-adjacency-changes
 network 10.1.2.0 0.0.0.3 area 0
 network 172.16.100.0 0.0.0.255 area 0
!
ip classless
!
line con 0
line aux 0
line vty 0 4
!
end
```

Example 11-4 shows R2's full configuration.

Example 11-4. R2's Full Configuration

```
version 12.1
!
hostname R2
!
ip subnet-zero
!
interface Ethernet0/0
 ip address 172.16.101.1 255.255.255.0
 half-duplex
!
interface Serial0/0

 ip address 10.1.2.2 255.255.255.252
!
router ospf 1
 log-adjacency-changes
 network 10.1.2.0 0.0.0.3 area 0
 network 172.16.101.0 0.0.0.255 area 0
!
ip classless
!
line con 0
line aux 0
line vty 0 4
!
end
```

Priority Queuing

This section covers the configuration and verification of priority queuing. The commands related to priority queuing are shown and discussed. Examples are also provided to clarify the concepts being discussed.

Priority Queuing Commands

The commands used to configure priority queuing are

- priority-list protocol
- priority-list interface

- priority-list default
- priority-list queue-limit
- priority-group

Priority Queuing Configuration

Two main steps are required to configure priority queuing:

Step 1. Define a priority list.

Step 2. Assign the priority list to an interface.

Defining a Priority List

A*priority list* is a list of filters that determine which queue a packet is to be placed in. These queuing priorities can be based on the following:

- Protocol type
- TCP/UDP port numbers
- Interface that the packet came in on
- IP precedence
- Source IP address
- Packet size in bytes

NOTE

Packets that do not match the priority list rules must be explicitly placed in the default queue.

The following global configuration mode command defines a priority list based on protocol type:

```
Router(config)#priority-listlist-numberprotocolprotocol-name {high |medium |
  normal |low}queue-keyword keyword-value
```

The following global configuration mode command defines a priority list based on the interface the packet came in on:

```
Router(config)#priority-listlist-numberinterfaceinterface-type interface-number

   {high | medium | normal | low}
```

The following examples demonstrate the creation of priority lists.

The command shown in Example 11-5 places IP packets in the high-priority queue.

Example 11-5. Priority Lists Using Protocol Type

```
R1(config)#priority-list 10 protocol ip high
```

The command shown in Example 11-6 places packets coming in on interface Ethernet0/0 in the medium-priority queue.

Example 11-6. Priority Lists Based on Interface

```
R1(config)#priority-list 10 interface ethernet 0/0 medium
```

The commands shown in Example 11-7 place IP traffic from a source network of 66.218.71.0 in the high-priority queue. This is a two-step process. First you create an access list that differentiates the traffic you want. Then you tell the router to place the specific traffic in the high-priority queue.

Example 11-7. Priority Lists Based on Source IP Address

```
R1(config)#access-list 1 permit 66.218.71.0 0.0.0.255

R1(config)#priority-list 10 protocol ip high list 1
```

The commands shown in Example 11-8 place Telnet traffic in the high-priority queue and place TFTP traffic in the low-priority queue, respectively.

Example 11-8. Priority Lists Based on TCP/UDP Port Numbers

```
R1(config)#priority-list 10 protocol ip high tcp 23

R1(config)#priority-list 10 protocol ip low udp 69
```

You can create priority lists using multiple rules. The rules are searched in order for a match. If a match is not found, the packet is placed in the default queue.

Assigning the Priority List to an Interface

After you create it, you can assign a priority list to an interface. Only one priority list can be applied to an interface at a time. The priority list is then applied to all traffic going through that interface.

The following command assigns a priority list to an interface:

```
Router(config-if)#priority-grouplist-number
```

InExample 11-9, the first command enters interface configuration mode on Serial 0/2, and the second command applies priority list 10 to that interface.

Example 11-9. Applying a Priority List to an Interface

```
R1(config)#interface serial 0/2

R1(config-if)#priority-group 10
```

Custom Queuing

This section covers the configuration and verification of custom queuing. The commands needed to configure custom queuing are shown and discussed. Examples are also provided to help you understand the concepts.

Custom Queuing Commands

The commands used to configure custom queuing are

- queue-list protocol
- queue-list interface
- queue-list default
- queue-list queue limit
- queue-list queue byte-count
- custom-queue-list

Custom Queuing Configuration

Two main steps are required to configure custom queuing:

Step 1. Define a custom queue list.

Step 2. Assign the custom queue list to an interface.

NOTE

Packets that do not match the custom queue list rules must be explicitly placed in the default queue.

Defining a Custom Queue List

A*custom queue list* is a list of filters that determine which queue a packet is to be placed in. These queuing priorities can be based on the following:

- Protocol type
- TCP/UDP port numbers
- Input interface
- Source IP address

The following global configuration mode command defines a custom queue list based on protocol type:

```
Router(config)#queue-listlist-numberprotocolprotocol-name queue-number

  queue-keyword keyword-value
```

The following global configuration mode command defines a custom queue list based on interface type:

```
Router(config)#queue-listlist-numberinterfaceinterface-type interface-number

  queue-number
```

NOTE

You can use custom queuing to place traffic in one of 16 possible queues. Queue 0 is reserved for time-sensitive system traffic such as keepalives and routing protocol messages.

The following examples demonstrate the creation of custom queue lists.

The command shown in Example 11-10 places IP packets in Queue 2.

Example 11-10. Custom Queue Lists Using the Protocol Type

```
R1(config)#queue-list 1 protocol ip 2
```

The command shown in Example 11-11 places packets coming in on Ethernet 0/0 in Queue 3.

Example 11-11. Custom Queue Lists Using the Input Interface

```
R1(config)#queue-list 1 interface ethernet 0/0 3
```

The two commands shown in Example 11-12 place traffic coming from network 10.15.20.0 /24 in Queue 4. This is a two-step process. The first command defines a simple access list that differentiates traffic from the target network. The second command places that traffic coming from the target network in the desired queue.

Example 11-12. Custom Queue Lists Using the Source IP Address

```
R1(config)#access-list 1 permit 10.15.20.0 0.0.0.255

R1(config)#queue-list 1 protocol ip 4 list 1
```

InExample 11-13, the first command puts HTTP traffic in Queue 5, and the second command places DNS traffic in Queue 6.

Example 11-13. Custom Queue Lists Based on TCP and UDP Port Numbers

```
R1(config)#queue-list 1 protocol ip 5 tcp 80
```

```
R1(config)#queue-list 1 protocol ip 6 udp 53
```

The command shown in Example 11-14 places default traffic in Queue 10.

Example 11-14. Custom Queue Lists for Default Traffic

```
R1(config)#queue-list 1 default 10
```

You can create a custom queue list using multiple rules. The list is searched in order for a match. When traffic matches a rule, it is placed in the appropriate queue. If no match occurs, the traffic is placed in the queue configured for default traffic.

Assigning the Custom Queue List to an Interface

After you create it, you can assign a custom queue list to an interface. Only one custom queue list can be applied to an interface at a time. The custom queue list is then applied to all traffic going through that interface.

The following command assigns a custom queue list to an interface:

```
Router(config-if)#custom-queue-listlist
```

InExample 11-15, the first command enters interface configuration mode on serial 0/2. The second command applies custom queue list 1 to that interface.

Example 11-15. Assigning a Custom Queue List to an Interface

```
R1(config)#interface serial 0/2

R1(config-if)#custom-queue-list 1
```

Compression

In addition to queuing, data compression is a useful way to increase network performance over a WAN link. By reducing the size of the frame to be transmitted, throughput can be increased. This section discusses the various kinds of compression. The commands needed to configure compression also are shown.

The kinds of compression supported by Cisco routers are as follows:

- Link compression
- Payload compression
- TCP header compression
- Microsoft Point-to-Point Compression (MPPC)

These methods are discussed briefly in the following sections.

NOTE

By default, frames are transmitted across a link uncompressed.

Link Compression

Also known as per-interface compression, this technique involves compressing both the header and a data frame's payload.

Two main algorithms are used to compress the traffic:

- Predictor— This algorithm is based on predicting the next sequence of characters in the data stream. This method is memory-intensive.
- STAC— This algorithm searches for redundant strings and replaces them with tokens, which are shorter than the original strings. This method is CPU-intensive.

Payload Compression

This technique involves compressing the data portion of a data frame. This is especially useful in an internetwork made up of different WAN networks, such as X.25, Frame Relay, and ATM.

It is also called *per-virtual circuit compression*. Payload compression uses the STAC compression algorithm.

TCP Header Compression

This technique is based on the Van Jacobson algorithm detailed in RFC 1144. This method is protocol-specific. Because only the TCP/IP header is compressed, the Layer 2 header is left unchanged.

This method is CPU-intensive and is good for protocols that have a small payload size, such as Telnet.

Microsoft Point-to-Point Compression

This technique, based on RFC 2118, uses an LZ compression mechanism. It can be used when communicating with a host using MPPC across a WAN link.

Configuring Compression

The following interface mode commands enable compression.

This command configures compression for an LAPB, PPP, or HDLC link:

```
Router(config-if)#compress [predictor |stac |mppc]
```

This command enables STAC compression on a Frame Relay point-to-point interface or subinterface:

```
Router(config-if)#frame-relay payload-compress
```

The following command enables TCP header compression. The passive option compresses outgoing TCP packets only if incoming TCP packets are compressed. If the passive option is not specified, all packets are compressed.

```
Router(config-if)#ip tcp header-compression [passive]
```

Scenarios

This section provides two scenarios of how queuing can be configured to manage network performance. Each scenario outlines the steps involved. The results are shown and verified. The complete configuration of the routers is also provided.

Scenario 11-1: Configuring Priority Queuing

A network administrator wants to give priority to VoIP traffic. He also wants to give FTP traffic medium priority. All other traffic should be placed in the normal queue. He wants to use priority queuing to accomplish this task.

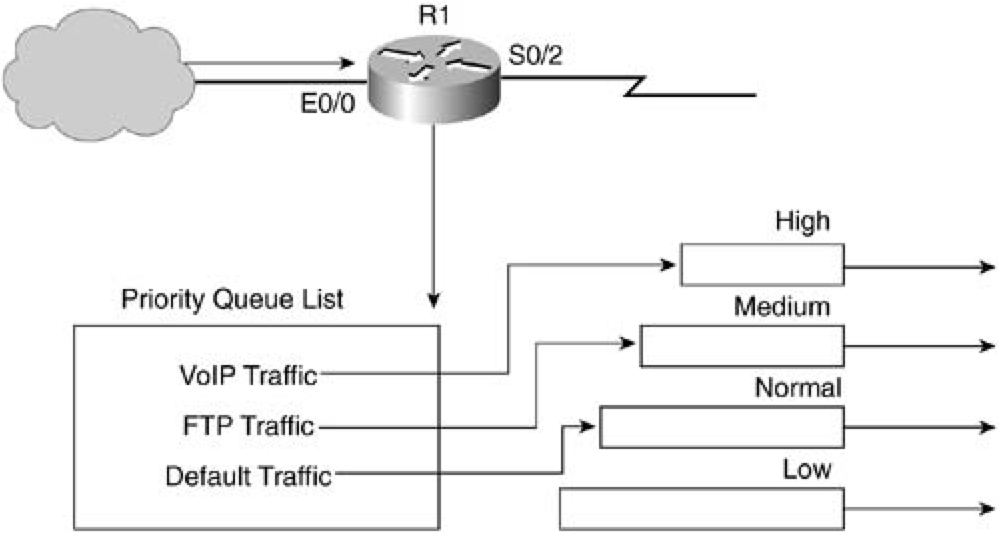
Because VoIP packets generally are 64 bytes in size, the administrator wants to ensure that packets less than 100 bytes are placed in the high queue. This will ensure that VoIP packets are given priority so that jitter can be avoided.

NOTE

Priority queuing is used to give VoIP traffic precedence in this example for purposes of illustration. In most situations, low-latency queuing or class-based WFQ would be used to give VoIP traffic precedence.

The topology for this scenario is shown in Figure 11-5.

Figure 11-5. Priority Queuing Scenario Topology



The following commands create and apply the priority list.

The first command shown next gives priority to packets that are less than 100 bytes. This puts VoIP traffic in the high-priority queue. The second command places FTP traffic in the medium queue. The third command places all other traffic in the normal queue.

```
R1(config)#priority-list 1 protocol ip high lt 100

R1(config)#priority-list 1 protocol ip medium tcp 20

R1(config)#priority-list 1 default normal
```

The following two commands take the newly created priority list and apply it to interface serial 0/0:

```
R1(config)#interface serial0/0

R1(config-if)#priority-group 1
```

Example 11-16 shows the full configuration of R1.

Example 11-16. R1's Full Configuration

```
R1#show running-config

version 12.1

!

hostname R1
```

```
!
ip subnet-zero
!
interface Ethernet0/0
 ip address 172.16.100.1 255.255.255.0
 half-duplex
!
interface Serial0/2
ip address 10.1.2.1 255.255.255.252
 priority-group 1
!
router ospf 1
 log-adjacency-changes
 network 10.1.2.0 0.0.0.3 area 0
 network 172.16.100.0 0.0.0.255 area 0
!
ip classless
!
priority-list 1 protocol ip high lt 100
priority-list 1 protocol ip medium tcp ftp-data
!
end
```

Theshow queueing priority and debug priority commands can be used to verify priority queuing operations, as shown in Example 11-17 and Example 11-18, respectively.

Example 11-17. Output of show queueing priority

```
R1#show queueing priority
```

```
Current DLCI priority queue configuration:

Current priority queue configuration:


List   Queue  Args

1      high   protocol ip          lt 100

1      medium protocol ip          tcp port ftp-data
```

Example 11-18. Output of debug priority

```
R1#debug priority

4d01h: PQ: Serial0/2: ip (70 bytes) -> high

4d01h: PQ: Serial0/2: ip (tcp 20) -> medium

4d01h: PQ: Serial0/2: ip (70 bytes) -> high

4d01h: PQ: Serial0/2: ip (tcp 20) -> medium

4d01h: PQ: Serial0/2: ip (70 bytes) -> high

4d01h: PQ: Serial0/2: ip (tcp 20) -> medium

4d01h: PQ: Serial0/2: ip (70 bytes) -> high

4d01h: PQ: Serial0/2: ip (tcp 20) -> medium

4d01h: PQ: Serial0/2: ip (70 bytes) -> high

4d01h: PQ: Serial0/2: ip (tcp 20) -> medium

4d01h: PQ: Serial0/2: ip (70 bytes) -> high

4d01h: PQ: Serial0/2: ip (tcp 20) -> medium

4d01h: PQ: Serial0/2: ip (70 bytes) -> high
```

Example 11-18 shows that the smaller VoIP packets are placed in the high-priority queue. Also, the FTP data packets are placed in the medium-priority queue.

Theping and extended ping commands can also be used to verify the scenario, as shown in Example 11-19 and Example 11-20, respectively. By varying the size of the ping packet, you can verify the queuing operation.

Example 11-19. Using Pings to Verify Queuing Operations

```
R1#ping 10.1.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/34/36 ms
R1#
4d01h: PQ: Serial0/2 output (Pk size/Q 24/0)
4d01h: PQ: Serial0/2 output (Pk size/Q 72/0)
4d01h: PQ: Serial0/2: ip (defaulting) -> normal
4d01h: PQ: Serial0/2 output (Pk size/Q 104/2)
4d01h: PQ: Serial0/2: ip (defaulting) -> normal
4d01h: PQ: Serial0/2 output (Pk size/Q 104/2)
4d01h: PQ: Serial0/2: ip (defaulting) -> normal
4d01h: PQ: Serial0/2 output (Pk size/Q 104/2)
4d01h: PQ: Serial0/2: ip (defaulting) -> normal
4d01h: PQ: Serial0/2 output (Pk size/Q 104/2)
4d01h: PQ: Serial0/2: ip (defaulting) -> normal
4d01h: PQ: Serial0/2 output (Pk size/Q 104/2)
```

The debug output shows that because the ping packets are 104 bytes each, they are placed in the default queue of normal priority.

Example 11-20. Using Extended Pings to Verify Queuing Operations

```
R1#ping
Protocol [ip]:
Target IP address: 10.1.2.2
Repeat count [5]:
Datagram size [100]: 64
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 64-byte ICMP Echos to 10.1.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/24/25 ms
R1#
4d02h: PQ: Serial0/2 output (Pk size/Q 24/0)
4d02h: PQ: Serial0/2 output (Pk size/Q 72/0)
4d02h: PQ: Serial0/2: ip (68 bytes) -> high
4d02h: PQ: Serial0/2 output (Pk size/Q 68/0)
4d02h: PQ: Serial0/2: ip (68 bytes) -> high
4d02h: PQ: Serial0/2 output (Pk size/Q 68/0)
4d02h: PQ: Serial0/2: ip (68 bytes) -> high
4d02h: PQ: Serial0/2 output (Pk size/Q 68/0)
4d02h: PQ: Serial0/2: ip (68 bytes) -> high
4d02h: PQ: Serial0/2 output (Pk size/Q 68/0)
4d02h: PQ: Serial0/2: ip (68 bytes) -> high
4d02h: PQ: Serial0/2 output (Pk size/Q 68/0)
```

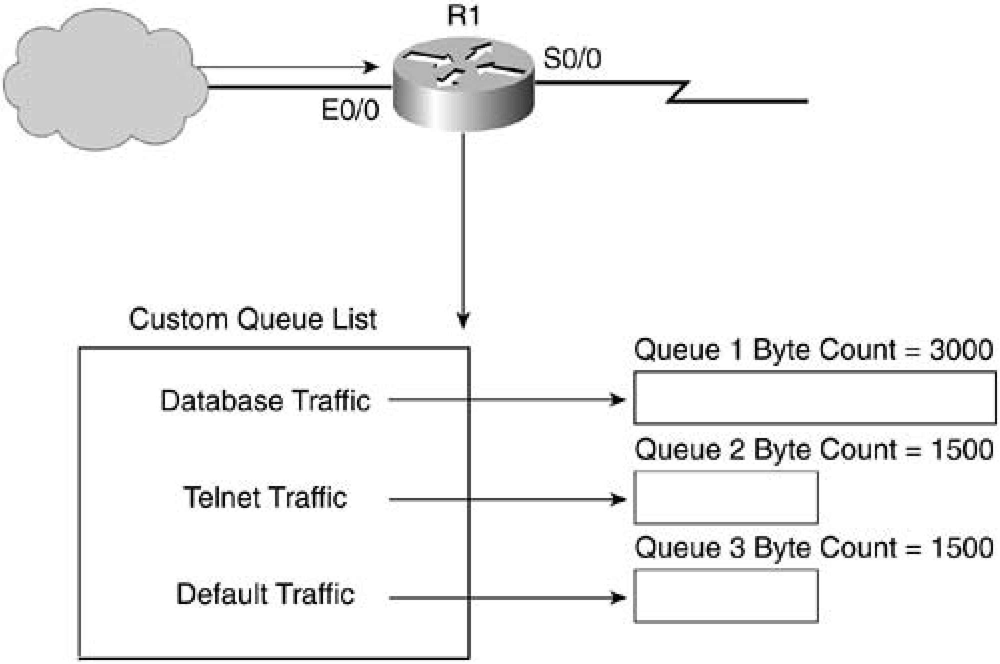
InExample 11-20, the extended ping command is used to set the size of the ping packet to 64 bytes. Because the packet is less than 100 bytes long, the router places it in the high-priority queue.

Scenario 11-2: Configuring Custom Queuing

A corporation has a branch office connected via a T1 to its central site. The network administrator at the branch site wants to divide bandwidth between a database application and other traffic. He wants the database application to receive a greater share of traffic. He also wants Telnet traffic to be placed in its own queue.

Figure 11-6 shows the topology for this scenario.

Figure 11-6. Custom Queuing Scenario Topology


R1
S0/0
E0/0
Custom Queue List
Database Traffic
Telnet Traffic
Default Traffic
Queue 1 Byte Count = 3000
Queue 2 Byte Count = 1500
Queue 3 Byte Count = 1500

The configuration steps are described next.

```
R1(config)#queue-list 1 protocol ip 1 tcp 1521

R1(config)#queue-list 1 protocol ip 2 tcp 23

R1(config)#queue-list 1 default 3

R1(config)#queue-list 1 queue 1 byte-count 3000
```

The first of the preceding commands places the database application traffic, which uses TCP 1521, in Queue 1. The second command places all Telnet traffic in Queue 2. The third command places all other traffic in Queue 3. The last command allocates 3000 bytes to Queue 1. This is the queue that services the database application. Because the default byte count for the other queues is 1500, the database application is allocated more bandwidth than any other type of traffic.

These commands assign the custom queue list to interface serial0/0:

```
R1(config)#interface serial0/0

R1(config-if)#custom-queue-list 1
```

NOTE

Packet sizes of the various applications and protocols play a crucial part in bandwidth allocation when the queue-list queue byte-count command is used. For a complete understanding of this topic, refer to the Cisco Press book *IP Quality Of Service* by Srinivas Vegesna.

Example 11-21 shows R1's full configuration.

Example 11-21. R1's Full Configuration

```
R1#show running-config

version 12.1

!

hostname R1

!

ip subnet-zero
```

```
!
interface Ethernet0/0
 ip address 172.16.100.1 255.255.255.0
 half-duplex
!
interface Serial0/0
ip address 10.1.2.1 255.255.255.252
 custom-queue-list 1
!
router ospf 1
 log-adjacency-changes
 network 10.1.2.0 0.0.0.3 area 0
 network 172.16.100.0 0.0.0.255 area 0
!
queue-list 1 protocol ip 1 tcp 1521
queue-list 1 protocol ip 2 tcp telnet
queue-list 1 default 3
queue-list 1 queue 1 byte-count 3000
!
end
```

Example 11-22 and Example 11-23 show how the show queueing custom and debug custom-queue commands, respectively, can be used to verify custom queuing operation.

Example 11-22. Output of show queueing custom

```
R1#show queueing custom
Current custom queue configuration:
```

```
List   Queue  Args
1      3      default
1      1      protocol ip          tcp port 1521
1      2      protocol ip          tcp port telnet
1      1      byte-count 3000
```

Example 11-23. Output of debug custom-queue

```
4d02h: CQ: Serial0/2 output (Pk size/Q: 786/3) Q # was 3 now 3
4d02h: CQ: Serial0/2 output (Pk size/Q: 786/3) Q # was 3 now 4
4d02h: CQ: Serial0/2 output (Pk size/Q: 1486/1) Q # was 4 now 1
4d02h: CQ: Serial0/2 output (Pk size/Q: 1486/1) Q # was 1 now 1
4d02h: CQ: Serial0/2 output (Pk size/Q: 1486/1) Q # was 1 now 2
4d02h: CQ: Serial0/2 output (Pk size/Q: 114/2) Q # was 2 now 2
4d02h: CQ: Serial0/2 output (Pk size/Q: 114/2) Q # was 2 now 2
4d02h: CQ: Serial0/2 output (Pk size/Q: 114/2) Q # was 2 now 2
4d02h: CQ: Serial0/2 output (Pk size/Q: 114/2) Q # was 2 now 2
4d02h: CQ: Serial0/2 output (Pk size/Q: 114/2) Q # was 2 now 2
4d02h: CQ: Serial0/2 output (Pk size/Q: 114/2) Q # was 2 now 2
4d02h: CQ: Serial0/2 output (Pk size/Q: 114/2) Q # was 2 now 2
4d02h: CQ: Serial0/2 output (Pk size/Q: 114/2) Q # was 2 now 2
```

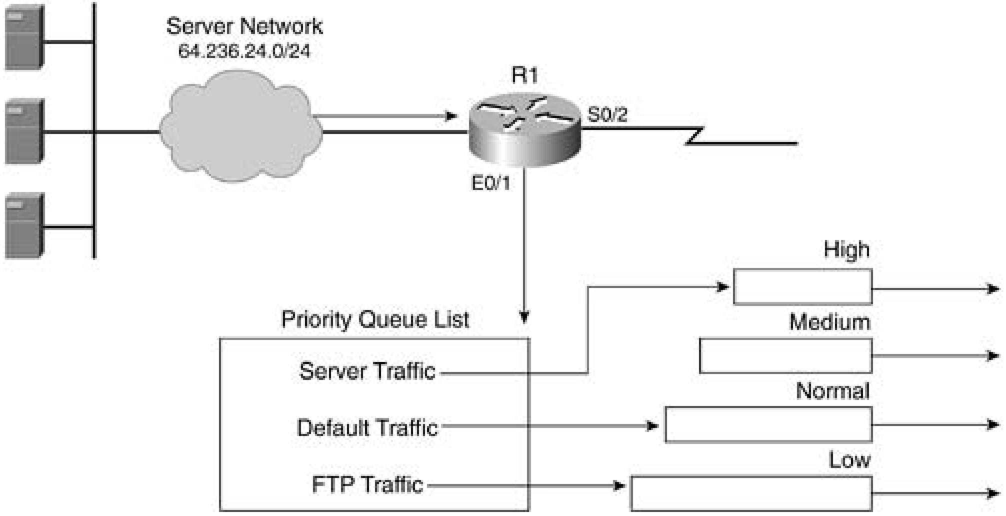
Practical Exercise: Configuring Priority Queuing

A corporation is using a T1 to connect its central office to a remote branch office. The network administrator at the branch office wants to give priority to traffic from a mission-critical server network at the branch office. This network has an IP address of 64.236.24.0 /24. The network administrator also wants to give lower priority to FTP traffic. All other traffic should have normal priority.

Figure 11-7 shows the topology relating to this Practical Exercise.

Figure 11-7. Practical Exercise Topology

[View full size image]



Practical Exercise Solution

Example 11-24 shows the solution to the Practical Exercise, and Example 11-25 shows the output of show queueing priority.

Example 11-24. Solution to Practical Exercise

```
R1#show running-config
version 12.1
!
hostname R1
ip subnet-zero
!
!
interface Ethernet0/0
 ip address 172.16.100.1 255.255.255.0
 half-duplex
!
interface Ethernet0/1
 ip address 64.236.24.1 255.255.255.0
 half-duplex
!
interface Serial0/2
ip address 10.1.2.1 255.255.255.252
 priority-group 1
!
router ospf 1
 log-adjacency-changes
 network 10.1.2.0 0.0.0.3 area 0
 network 172.16.100.0 0.0.0.255 area 0
 network 64.236.24.0 0.0.0.255 area 0
!
ip classless
!
access-list 1 permit 64.236.24.0 0.0.0.255
priority-list 1 protocol ip high list 1
priority-list 1 protocol ip low tcp ftp-data
!
end
```

Example 11-25. Output of show queueing priority

```
R1#show queueing priority
Current DLCI priority queue configuration:
Current priority queue configuration:

List   Queue  Args
1      high   protocol ip          list 1
1      low    protocol ip          tcp port ftp-data
```

Example 11-26 shows the output of the debug priority command. Note that packets from network 64.236.24.0/24 are placed in the high-priority queue. You can also see that FTP traffic is being placed in the low-priority queue.

Example 11-26. Output of debug priority

```
R1#
```

```
4d02h: PQ: Serial0/2: ip (s=64.236.24.25, d=172.16.101.26) -> high
4d02h: PQ: Serial0/2: ip (s=64.236.24.25, d=172.16.101.26) -> high
4d02h: PQ: Serial0/2: ip (s=64.236.24.25, d=172.16.101.26) -> high
4d02h: PQ: Serial0/2: ip (s=64.236.24.25, d=172.16.101.26) -> high
4d02h: PQ: Serial0/2: ip (s=64.236.24.25, d=172.16.101.26) -> high
4d02h: PQ: Serial0/2: ip (s=64.236.24.25, d=172.16.101.26) -> high
4d02h: PQ: Serial0/2: ip (s=64.236.24.25, d=172.16.101.26) -> high
4d02h: PQ: Serial0/2: ip (s=64.236.24.25, d=172.16.101.26) -> high
4d02h: PQ: Serial0/2: ip (s=64.236.24.25, d=172.16.101.26) -> high
4d02h: PQ: Serial0/2: ip (s=64.236.24.25, d=172.16.101.26) -> high
4d02h: PQ: Serial0/2: ip (s=64.236.24.25, d=172.16.101.26) -> high
4d02h: PQ: Serial0/2: ip (defaulting) -> normal
4d02h: PQ: Serial0/2: ip (tcp 20) -> low
4d02h: PQ: Serial0/2: ip (defaulting) -> normal
4d02h: PQ: Serial0/2: ip (tcp 20) -> low
4d02h: PQ: Serial0/2: ip (defaulting) -> normal
4d02h: PQ: Serial0/2: ip (tcp 20) -> low
4d02h: PQ: Serial0/2: ip (defaulting) -> normal
4d02h: PQ: Serial0/2: ip (tcp 20) -> low
4d02h: PQ: Serial0/2: ip (defaulting) -> normal
4d02h: PQ: Serial0/2: ip (tcp 20) -> low
4d02h: PQ: Serial0/2: ip (defaulting) -> normal
4d02h: PQ: Serial0/2: ip (tcp 20) -> low
```

Example 11-27 uses extended pings to verify queuing operation.

Example 11-27. Using Extended Pings to Verify Queuing Operation

```
R1#ping
Protocol [ip]:
Target IP address: 172.16.101.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface:
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.101.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/33/36 ms

R1#
4d02h: PQ: Serial0/2: ip (s=64.236.24.1, d=172.16.101.1) -> high
4d02h: PQ: Serial0/2 output (Pk size/Q 104/0)
4d02h: PQ: Serial0/2: ip (s=64.236.24.1, d=172.16.101.1) -> high
4d02h: PQ: Serial0/2 output (Pk size/Q 104/0)
4d02h: PQ: Serial0/2: ip (s=64.236.24.1, d=172.16.101.1) -> high
4d02h: PQ: Serial0/2 output (Pk size/Q 104/0)
4d02h: PQ: Serial0/2: ip (s=64.236.24.1, d=172.16.101.1) -> high
4d02h: PQ: Serial0/2 output (Pk size/Q 104/0)
```

```
4d02h: PQ: Serial0/2: ip (s=64.236.24.1, d=172.16.101.1) -> high
4d02h: PQ: Serial0/2 output (Pk size/Q 104/0)

R1#ping
Protocol [ip]:
Target IP address: 172.16.101.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.16.100.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.101.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/34/36 ms
R1#
4d02h: PQ: Serial0/2: ip (defaulting) -> normal
4d02h: PQ: Serial0/2 output (Pk size/Q 104/2)
4d02h: PQ: Serial0/2: ip (defaulting) -> normal
4d02h: PQ: Serial0/2 output (Pk size/Q 104/2)
4d02h: PQ: Serial0/2: ip (defaulting) -> normal
4d02h: PQ: Serial0/2 output (Pk size/Q 104/2)
```

```
4d02h: PQ: Serial0/2: ip (defaulting) -> normal

4d02h: PQ: Serial0/2 output (Pk size/Q 104/2)

4d02h: PQ: Serial0/2: ip (defaulting) -> normal

4d02h: PQ: Serial0/2 output (Pk size/Q 104/2)

4d02h: PQ: Serial0/2 output (Pk size/Q 24/0)
```

Summary

Queuing and compression can provide ways to help you manage your bandwidth and network performance. A number of queuing techniques are available. This chapter discussed the three main methods of queuing. Configuration examples were provided to show these methods. Scenarios for priority and custom queuing were also shown. A Practical Exercise was also provided to help your understanding of the concepts presented.

Review Questions

1: What is the default queuing mechanism for interfaces with speeds of E1 and less?

2: Which queuing mechanism should you use to give absolute priority to critical traffic?

3: Which queuing mechanism ensures that packet trains do not adversely affect critical traffic?

4: What is the default congestive discard threshold for Weighted Fair Queuing?

5: How many configurable queues are available for custom queuing?

6: What is the default byte count for queues in custom queuing?

A. 1024

B. 1500

C. 512

D. 256

7: Which of the following cannot be used to classify packets for priority queuing?

A. Protocol type

B. Ingress interface

C. Packet size in bytes

D. Egress interface

8: Queuing is done on which interface?

A. Ingress interface

B. Egress interface

C. Example interface

D. Weighted interface

Chapter 12. Scaling IP Addressing with Network Address Translation

This chapter covers the following topics:

- NAT Operation
- Configuring NAT
- NAT Order of Operation
- When to Use NAT
- NAT Configuration Task List

One of the problems facing anyone connecting to the Internet today is the depletion of IP addresses. The IP version 4 address space was originally designed so that 4,294,967,296 (2^{32}) hosts could be assigned a unique address. Because addresses are reserved for multicasting, testing, and other purposes, and because the nonreserved address space is divided into classes, this range is actually somewhere between 3,200,000,000 and 3,300,000,000 addresses. With the exponential growth of companies doing business over the Internet, IP address assignments became a major concern across the networking world.

For your computer to effectively communicate on the Internet, it must have a unique 32-bit IP address. This IP address identifies the location of your computer on a network, much like your phone number distinguishes your phone from the millions of other phones out there.

With the unpredicted popularity of the Internet and the continuing increase in the number of home and business networks, the number of available IP addresses is simply not enough. IP version 6 is being developed to eliminate these issues, but it will take several years to implement, because it will require modifying the Internet's infrastructure. Because of this lag in deployment, Network Address Translation (NAT) was defined in RFC 1631, *The IP Network Translator*. In the simplest of terms, NAT allows a single device to act as an agent between the Internet (or "public network") and a local (or "private") network. This allows you to use a single unique IP address to represent your entire internal network to anything or anyone outside your network. Besides NAT's obvious benefits when it comes to addressing the shortage of IP addresses, you also gain security and administrative benefits from it.

NAT Operation

NAT can be confused with a proxy server, but there are definite differences between the two. NAT is transparent to the source and destination computers, but a proxy server is not. The source computer has to be specifically configured to communicate with a proxy server, whereas the destination computer thinks that the proxy server is the source computer. Proxy servers usually operate at Layer 4 (the transport layer of the OSI Reference Model) or higher, and NAT operates at Layer 3 (the network layer). Because proxy servers are usually an add-on application, they might be slower than NAT, because NAT is accomplished in hardware.

NAT is configured on the device you use to connect to an external network, whether it is a firewall, router, or computer. Before you get too far into the operation of NAT, you need to have a basic understanding of its many forms and the several ways in which it can be used:

- Static NAT— Used to map an unregistered IP address, such as a private address, to a registered IP address, usually provided by your Internet service provider (ISP), on a one-to-one basis. Also used to map one external public address to one internal private address.

- Dynamic NAT— Used to map an unregistered IP address to a registered IP address from a group of registered IP addresses. Dynamic NAT is usually accomplished with the assistance of a pool or a range of addresses that you configure on your NAT device.

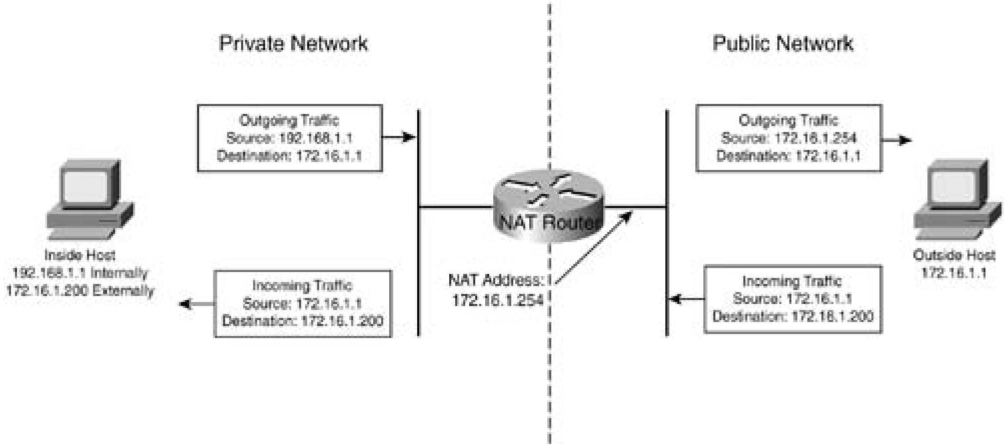
- Overloading— A form of dynamic NAT used to map multiple unregistered IP addresses to a single registered IP address by using different ports. More commonly known as Port Address Translation (PAT) or port-level multiplexed NAT.

- Overlapping— Used when the IP address of your internal network is registered for use on another network. Your NAT device must maintain some type of lookup table of these addresses so that it can intercept them and replace them with registered unique IP addresses. This means that your NAT device must be able to translate the "internal" addresses to registered unique addresses. It also must be able to translate the "external" registered addresses to addresses that are unique to the private network. You can implement this NAT method through the use of static NAT or through the use of a DNS entry and dynamic NAT.

One fact that might need to be mentioned at this point is that your internal network, or LAN, can often be referred to as a stub domain. When used in this manner, a stub domain is a LAN that uses IP addresses internally, with most of the network traffic having a local destination. Although you are allowed to have both registered and unregistered IP addresses in your stub domain, any network device that uses an unregistered IP addresses must use NAT to communicate with the outside world. Figure 12-1 illustrates a NAT operation in which a host on a private network communicates with a host on a public network and a host on the public network communicates with a host on the private network.

Figure 12-1. NAT Operation

[View full size image]



One other benefit of implementing dynamic NAT on your device is that it can automatically create a simple firewall between your internal network and outside networks or the Internet. NAT does this by allowing only connections that originate inside your stub domain. This lets you limit a computer on an external network from reaching your computer unless your computer initiated the contact. Using static NAT allows you to define where a connection initiated by an external device can connect on your computers. For instance, you might want to connect an inside global address to a specific inside local address that is assigned to your web server. Keep in mind that this simple firewall should not be considered a replacement for items such as the Cisco Secure PIX Firewall or the Cisco IOS Firewall Feature Set, because TCP packets may be forged by an unauthorized user to gain access to your "protected" devices.

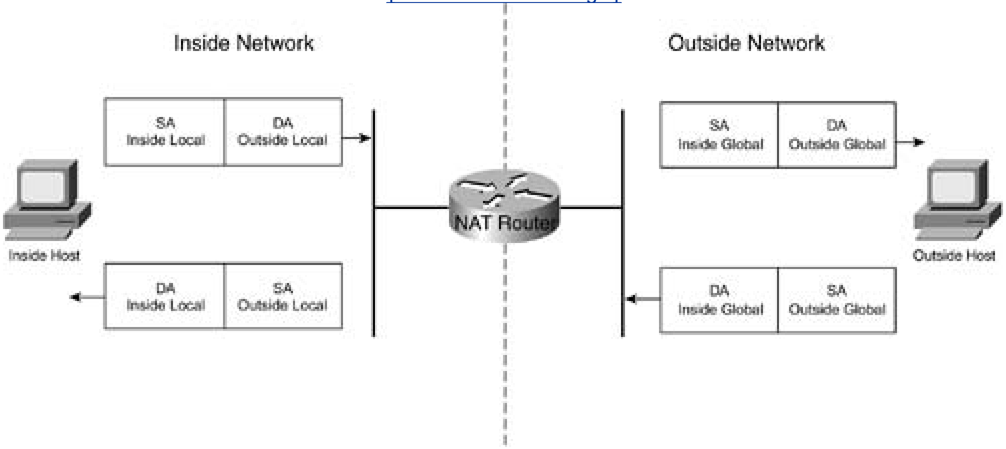
Configuring NAT

When you configure a router to use NAT, you configure one interface to the inside of your network and another to the outside of your network. Any packets that have a source address belonging to the "inside" portion of your network have an *inside local address* as the source address and an *outside local address* as the destination address. The packet resides on the "inside" portion of your network. When that same packet gets switched to the "outside" network, the packet's source is known as the *inside global address*, and the packet's destination is known as the *outside global address*.

For any packet that has a source address belonging to the "outside" portion of your network, while it is on the "outside" network, its source address is known as the outside global address. The packet's destination is known as the *inside global address*. When the same packet gets switched to the "inside" of your network, the source address is known as the outside local address, and the packet's destination is known as the inside local address. Figure 12-2 illustrates this.

Figure 12-2. "Inside" and "Outside" Sample Topology

[View full size image]



The following are the different types of addressing that are associated with NAT:

- Inside local address— An IP address that is assigned to a host on your inside network.
- Inside global address— A legitimate IP address that represents one or more of your inside local IP addresses to the outside world.
- Outside local address— An IP address of an outside host as it appears to your inside network.
- Outside global address— An IP address assigned to a host on the outside network by the owner of the host that is allocated from the globally routable address or network space.

A typical NAT implementation has NAT configured on the exit router between a stub domain and backbone, such as the Internet. When a packet leaves your domain, NAT translates the locally significant source address into a globally unique address and records it to memory. If the return packet matches what NAT has recorded, the packet is allowed back into the network. Otherwise, when a packet enters your domain, NAT translates the globally unique destination address into a local address if it's configured. Remember, if your domain has more than one exit point, each NAT process must have the same translation table to ensure proper translation. If NAT runs out of available addresses, the packet is dropped, and an ICMP host unreachable message is returned to the packet's originator.

When using PAT, in which several internal addresses are translated to only one or a few external addresses, additional translations of the packet are performed. Because each internal address may be translated to a single external address, PAT translates each packet's source port to a unique source port number, a 16-bit number or 65,536 ports per IP address, on the inside global IP address. This distinguishes them from other packets that are being translated. PAT tries to preserve the original source port. However, if the source port is already used in a translation, PAT attempts to find the first available port number, starting from the beginning of the appropriate port group—0 to 511, 512 to 1023, or 1024 to 65535. If PAT cannot allocate another port number from the appropriate group, and you configured more than one IP address, PAT moves to the next IP address and tries to allocate the original source port again. This process continues until PAT runs out of available IP addresses and ports.

When your router is configured to use NAT, it must not advertise local networks to the outside. However, routing information that NAT receives from the outside may still be advertised in the stub domain as usual.

NAT Order of Operation

As noted, NAT is based on whether a packet goes from your inside network to your outside network or from your outside network to your inside network. Table 12-1 illustrates the processing order in relation to where the packet originates. Note that when NAT performs the global-to-local or local-to-global translation, it is different in each flow.

Table 12-1. NAT Order of Operation

Inside-to-Outside	Outside-to-Inside
1. If IPSec, check the input access list 2. Decryption—for CET (Cisco Encryption Technology) or IPSec 3. Check the input access list 4. Check the input rate limits 5. Input accounting 6. Inspect 7. Policy routing 8. Routing 9. Redirect to the web cache 10. NAT inside-to-outside (local-to-global translation) 11. Crypto (check the map and mark it for encryption) 12. Check the output access list 13. Inspect 14. TCP intercept 15. Encryption	1. If IPSec, check the input access list 2. Decryption—for CET or IPSec 3. Check the input access list 4. Check the input rate limits 5. Input accounting 6. Inspect 7. NAT outside-to-inside (global-to-local translation) 8. Policy routing 9. Routing 10. Redirect to the web cache 11. Crypto (check the map and mark it for encryption) 12. Check the output access list 13. Inspect 14. TCP intercept 15. Encryption

As you can see from Table 12-1, NAT occurs after the router processes several items. NAT inside-to-outside also occurs in a different place than NAT outside-to-inside.

When to Use NAT

NAT is a very versatile feature that can be used for the following purposes:

- You use private or unregistered IP addresses on your internal network, but you want to connect to the Internet. NAT provides the necessary translations of your internal local addresses to globally unique IP addresses before sending packets to the outside network.
- You must change your internal addresses, but you don't want to. NAT can be used in this case to translate these addresses.
- You want to do basic load sharing of TCP traffic. With NAT, you can map a single global IP address to many local IP addresses by using the TCP load distribution feature.
- NAT can be used as a practical solution to a connectivity problem only when relatively few hosts in a stub domain communicate outside the domain at the same time. When this is the case, only a small subset of the IP addresses in the domain must be translated into globally unique IP addresses when outside communication is necessary. These addresses can be reused when they are no longer in use.

NAT Configuration Task List

To configure NAT, you must know the inside local address and inside global address you will translate. As soon as your NAT translation is configured, you may optionally do the following:

- Translate inside source addresses
- Overload an inside global address
- Translate overlapping addresses
- Provide TCP load distribution
- Change translation timeouts
- Deploy NAT between an IP phone and the Cisco CallManager

Translating Inside Source Addresses

You can translate your unregistered IP addresses into globally unique IP addresses to communicate outside your network using one of the following methods:

- Static translation— Establishes a one-to-one mapping between your inside local address and an inside global address.
- Dynamic translation— Establishes a mapping between an inside local address and a pool of global addresses.

Configuring Static Translation

You can use the following commands to configure static NAT translation.

Use this command to establish static translation between an inside local address and an inside global address:

```
R2(config)#ip nat inside source {list {access-list number | name}poolname

  [overload] | staticlocal-ip global-ip}
```

This command establishes static translation of an outside source address:

```
R2(config)#ip nat outside source {list {access-list number | name}poolname |
  staticglobal-ip local-ip}
```

Use this command to enter interface configuration mode and specify the inside interface:

```
R2(config)#interfacetype number
```

This command marks the interface as connected to the inside:

```
R2(config-if)#ip nat inside
```

To enter interface configuration mode and specify the outside, use this command:

```
R2(config)#interfacetype number
```

This command marks the interface as connected to the outside:

```
R2(config-if)#ip nat outside
```

These steps are the minimum you must configure to implement NAT. You can use multiple inside and outside interfaces if you are required to.

Configuring Dynamic Translation

You can use the following commands to configure dynamic inside source address translation.

This command defines a pool of global addresses to be allocated as needed:

```
R2(config)#ip nat poolname start-ip end-ip {netmasknetmask | prefix-length
  prefix-length}
```

To define a standard access list permitting addresses that requires translation, use this command:

```
R2(config)#access-listaccess-list-numberpermitsource [source-wildcard]
```

Use this command to establish dynamic source translation, specifying the access list defined in the prior step:

```
R2(config)#ip nat inside source listaccess-list-numberpoolname
```

Use this command to enter interface configuration mode and specify the inside interface:

```
R2(config)#interfacetype number
```

This command marks the interface as connected to the inside:

```
R2(config-if)#ip nat inside
```

To enter interface configuration mode and specify the outside interface, use this command:

```
R2(config)#interfacetype number
```

This command marks the interface as connected to the outside:

```
R2(config-if)#ip nat outside
```

Overloading an Inside Global Address

You can overload a single global address to translate many local addresses to conserve addresses in the inside global address pool. This overloading forces the router to maintain enough information from higher-level protocols, such as TCP or UDP port numbers, to allow it to translate the global address back to the correct local address.

You can use the following commands to configure overloading of inside global addresses.

To define a pool of global addresses to be allocated as needed, use this command:

```
R2(config)#ip nat poolname start-ip end-ip {netmasknetmask | prefix-length

  prefix-length}
```

To define a standard access list, use this command:

```
R2(config)#access-list access-list-number permit source [source-wildcard]
```

This command establishes dynamic source translation, specifying the access list defined in the prior step:

```
R2(config)#ip nat inside source list access-list-number pool name overload
```

This command specifies the inside interface:

```
R2(config)#interface type number
```

This command marks the interface as connected to the inside:

```
R2(config-if)#ip nat inside
```

This command specifies the outside interface:

```
R2(config)#interfacetype number
```

This command marks the interface as connected to the outside:

```
R2(config-if)#ip nat outside
```

Translating Overlapping Addresses

In most cases, NAT is used to translate private IP addresses into legal addresses that can be routed on the Internet. It can also be used to connect two networks that are using the same IP addressing on their internal networks. This scenario is called *overlapping addresses*.

You can use the following commands to configure static SA address translation.

To establish static translation between an outside local address and an outside global address, use this command:

```
R2(config)#ip nat outside source staticglobal-ip local-ip
```

This command specifies the inside interface:

```
R2(config)#interfacetype number
```

This command marks the interface as connected to the inside:

```
R2(config-if)#ip nat inside
```

This command specifies the outside interface:

```
R2(config)#interfacetype number
```

This command marks the interface as connected to the outside:

```
R2(config-if)#ip nat outside
```

You can use the following commands to configure dynamic outside source address translation.

To define a pool of local addresses to be allocated as needed, use this command:

```
R2(config)#ip nat poolname start-ip end-ip {netmasknetmask | prefix-length
  prefix-length}
```

This command defines a standard access list:

```
R2(config)#access-listaccess-list-numberpermitsource [source-wildcard]
```

To establish dynamic outside source translation, specifying the access list defined in the prior step, use this command:

```
R2(config)#ip nat outside source listaccess-list-numberpoolname
```

This command specifies the inside interface:

```
R2(config)#interfacetype number
```

This command marks the interface as connected to the inside:

```
R2(config-if)#ip nat inside
```

This command specifies the outside interface:

```
R2(config)#interfacetype number
```

This command marks the interface as connected to the outside:

```
R2(config-if)#ip nat outside
```

Providing TCP Load Distribution

When NAT comes up in everyday conversation, you probably think of it as a translation mechanism that allows your company to access the Internet. NAT has another function that is unrelated to this feature. If your company has multiple hosts that communicate with a heavily used host or server, you can use NAT to establish a virtual host on the inside network that coordinates load sharing among multiple real hosts. Allocation is done on a round-robin basis from a rotary pool of real addresses when a new connection is opened from the outside to the inside. Any non-TCP traffic is still passed without translation, unless other translations are in effect.

Use the following commands to configure destination address rotary translation to allow you to map one virtual host to many real hosts.

To define a pool of addresses containing the addresses of the real hosts, use this command:

```
R2(config)#ip nat poolname start-ip end-ip {netmasknetmask | prefix-length

  prefix-length}type rotary
```

To define an access list permitting the address of the virtual host, use this command:

```
R2(config)#access-listaccess-list-numberpermitsource [source-wildcard]
```

Use this command to establish dynamic inside destination translation, specifying the access list defined in the prior step:

```
R2(config)#ip nat inside destination listaccess-list-numberpoolname
```

This command specifies the inside interface:

```
R2(config)#interfacetype number
```

This command marks the interface as connected to the inside:

```
R2(config-if)#ip nat inside
```

This command specifies the outside interface:

```
R2(config)#interfacetype number
```

This command marks the interface as connected to the outside:

```
R2(config-if)#ip nat outside
```

Changing Translation Timeouts

If left to the default value, a dynamic address translation times out after some period of nonuse. When overloading is not in use, simple translation entries time out after 24 hours. You can use the following command to change this value:

```
R2(config)#ip nat translation timeoutseconds
```

Overloading gives you more control over translation entry timeout, because each entry contains more context about the traffic using it. You can use the following commands to change timeouts or extended entries.

This command changes the UDP timeout value from 5 minutes:

```
R2(config)#ip nat translation udp-timeoutseconds
```

This command changes the DNS timeout value from 1 minute:

```
R2(config)#ip nat translation dns-timeoutseconds
```

This command changes the TCP timeout value from 24 hours:

```
R2(config)#ip nat translation tcp-timeoutseconds
```

This command changes the finish and reset timeout value from 1 minute:

```
R2(config)#ip nat translation finrst-timeoutseconds
```

This command changes the ICMP timeout value from 1 minute:

```
R2(config)#ip nat translation icmp-timeoutseconds
```

This command changes the synchronous (SYN) timeout value from 1 minute:

```
R2(config)#ip nat translation syn-timeoutseconds
```

Deploying NAT Between an IP Phone and Cisco CallManager

Communication and registration between a Cisco IP phone and the Cisco CallManager (CCM) use the Selsius Skinny Station protocol. The Skinny protocol allows messages to flow back and forth between the devices that include IP address and port information used to identify other IP phone users with which a call can be placed.

When you use NAT with CCM and IP phones, NAT needs to be able to identify and understand the information passed within the Skinny protocol. When an IP phone attempts to make a connection with CCM and its IP address matches your NAT translation rules, NAT translates the original source IP address and replaces it with one from the configured pool. This new address is used to represent the IP phone to CCM as well as other IP phone users.

To specify the port number on which the CCM is listening for skinny messages, use this command:

```
R2(config)#ip nat service skinny tcp portnumber
```

Monitoring and Maintaining NAT

By default, dynamic address translations time out from the NAT translation table after a set amount of time. You can use the following commands to clear the entries before the configured timeout.

To clear all dynamic address translation entries from the NAT translation table, use this command:

```
R2#clear ip nat translation *
```

To clear a simple dynamic translation entry containing an inside translation, or both inside and outside translation, use this command:

```
R2#clear ip nat translation insideglobal-ip local-ip [outsidelocal-ip global-ip]
```

This command clears a simple dynamic translation entry containing an outside translation:

```
R2#clear ip nat translation outsidelocal-ip global-ip
```

This command clears an extended dynamic translation entry:

```
R2#clear ip nat translationprotocolinsideglobal-ip global-port local-ip
  local-port [outsidelocal-ip local-port global-ip global-port]
```

You can use one of the following commands to display translation information:

This command displays active translations:

```
R2#show ip nat translations [verbose]
```

This command displays translation statistics:

```
R2#show ip nat statistics
```

Scenarios

The scenarios presented in this chapter help you gain a more complete understanding of NAT opera
and configuration through practical application. You will go through the necessary configuration tas
their logical progression. The scenarios cover the following topics:

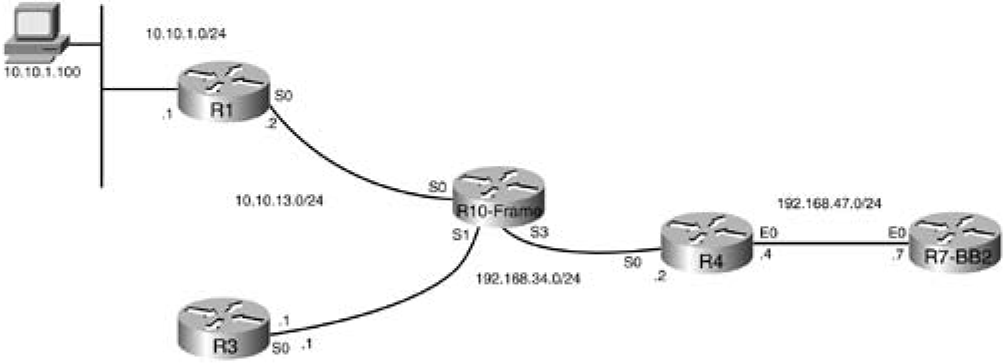
- Simple NAT topology
- Simple static NAT inside-to-outside translation
- Simple static NAT outside-to-inside translation
- Combining static NAT translation
- Overloading an IP address with NAT
- Using NAT with overlapping addresses
- Configuring TCP load distribution

Scenario 12-1: Simple NAT Topology

To further examine NAT, you will configure a simple network topology to examine the results of sev
different scenarios involving NAT. Figure 12-3 shows the topology used in this exercise.

Figure 12-3. NAT Test Topology

[View full size image]



In this scenario, you will configure R1 to translate the inside local address of 10.10.1.100 to 10.10.

Step 1: Initial Configuration

Before configuring any of the NAT configurations, you need to perform an initial configuration of all
routers you will use throughout the scenarios. Although you need to apply these configurations, yo

concentrate on R1 for now. You can do this from a terminal attached to R1's console port (line 0). Y begin by entering global configuration mode. You can then configure the router name using the ho command. It is also useful to disable the IP domain name system with the no ip domain-lookup command. This keeps the system from trying to translate domain names that have typographical e

You can use the enable secret command to enable a password for entering privileged EXEC mode the password is cisco. This secret password provides an additional layer of security on the router. Passwords are case-sensitive strings that can be up to 80 characters long. They cannot begin with number.

Because your router has a connection to its local network through the Ethernet 0 port, You enter in e 0 to configure this interface. But you can also use interface ethernet 0 and int eth 0. You set t address for the Ethernet interface using the ip address command. You also have to include a subn mask. You then activate the interface using the no shutdown command.

Your next configuration is to bring up the Serial 0 interface and configure its IP address. You enter interface s 0 to configure this interface. But you can also use interface serial 0 and int ser 0. Yo activate the interface using the no shutdown command. You can now create two subinterfaces, Se and Serial 0.2, and set the appropriate IP addresses for the Serial subinterfaces using the ip addre command. You also have to include a subnet mask.

You can optionally configure your console line to prevent it from automatically disconnecting you a default 10-minute idle time. To begin configuring the console line, enter line console 0. You are n line configuration mode. You use the no exec-timeout command to prevent the automatic disconn after a period of inactivity. The initial configuration of the R1 router is now complete. It is shown in Example 12-1.

NOTE

Don't forget to reset the exec-timeout after the configuration is complete. Leaving it open is potential security risk.

Example 12-1. Initial Configuration of R1

```
Router#configure terminal

Router(config)#hostname R1

R1(config)#no ip domain-lookup

R1(config)#enable secret cisco

R1(config)#interface ethernet 0

R1(config-if)#ip address 10.10.1.1 255.255.255.0

R1(config-if)#no shutdown

R1(config)#interface serial 0
```

```
R1(config-if)#no shutdown

R1(config)#interface serial 0.1

R1(config-if)#ip address 10.10.13.1 255.255.255.0

R1(config)#interface serial 0.2

R1(config-if)#ip address 10.10.14.1 255.255.255.0

R1(config-if)#line console 0

R1(config-line)#no exec-timeout
```

You can refer back to this section, substituting the information for the particular router you are configuring, whenever you encounter an unconfigured router.

Step 2: NAT Translation Configuration

In this step, you configure R1 with the required NAT configuration, as shown in Example 12-2.

Example 12-2. R1 Configuration for NAT Operation Test

```
! Configuration items for R1:


R1(config)#ip nat inside source static 10.10.1.100 10.10.15.100

R1(config)#interface serial 0.1 point-to-point

R1(config-if)#ip nat inside

R1(config-if)#exit

R1(config)#interface serial0.2 point-to-point

R1(config-if)#ip nat outside

R1(config-if)#exit
```

Step 3: Static Routing Configuration

The last step is to configure static routing. You could use dynamic routing to ensure connectivity fo networks, but that is outside the scope of this chapter. See Example 12-3.

Example 12-3. R1 Configuration for Static Routing

```
! Configuration items for R1:

R1(config)#ip route 0.0.0.0 0.0.0.0 10.10.14.2

R1(config)#ip route 10.10.15.1 255.255.255.0 10.10.13.2
```

You can now view the translation table on R1 to verify that the intended translation exists by using following command:

```
R1#show ip nat translation
```

Example 12-4 shows the results of issuing this command on your NAT router.

Example 12-4. show ip nat translation Command Output from R1

```
R1#show ip nat translation

Pro Inside global      Inside local       Outside local      Outside global

--- 10.10.1.100      10.10.15.100      ---                ---
```

By examining Example 12-4, you can tell that 10.10.1.100 is indeed translated to 10.10.15.100 as intended. You begin your examination of NAT operation by issuing a ping from 10.10.1.100 to the interface of R7 at 192.168.47.7. To see the packets crossing the network, you need to issue the fol commands on R1:

```
R1#debug ip packet detail

R1#debug ip nat
```

Example 12-5 shows the output generated on R1.

Example 12-5. Debug of IP Packets and NAT on R1

```
R1#debug ip packet detail

R1#debug ip nat

NAT: s=10.10.1.100->10.10.15.100, d=192.168.47.7 [481]

IP: s=10.10.15.100 (Serial0), d=192.168.47.7 (Serial1), g=172.16.47.145,

  len 100, forward ICMP type=8, code=0

R1#undebug all

All possible debugging has been turned off
```

ExaminingExample 12-5 shows that your packets are being translated by NAT as expected. Your ro must have valid routes for both the outside device and the inside device, or NAT will not be able to the packets correctly. One other thing to remember is that return packets must be translated befor can be routed.

Scenario 12-2: Simple Static NAT Inside-to-Outside Translation

In this scenario, you configure your NAT router, R1, so that when it receives a packet with a source address of 10.10.1.100 on its inside interface, it translates it to 10.10.14.100. Example 12-6 shows required configuration of R1 to complete this scenario.

Example 12-6. Inside-to-Outside Static Translation

```
! Configuration items for R1:


R1(config)#ip nat inside source static 10.10.1.100 10.10.14.100

R1(config)#interface ethernet 0

R1(config-if)#ip nat inside

R1(config-if)#exit

R1(config)#interface serial 0

R1(config-if)#ip nat outside
```

Scenario 12-3: Simple Static NAT Outside-to-Inside Translation

In this scenario, you configure R1 so that when it receives a packet with a source address of 10.10
on its outside interface, the source address is translated to 10.10.1.200. Example 12-7 shows R1's
configuration required to complete this scenario.

Example 12-7. Outside-to-Inside Static NAT Translation

```
! Configuration items for R1:


R1(config)#ip nat outside source static 10.10.14.200 10.10.1.200

R1(config)#interface ethernet 0

R1(config-if)#ip nat inside

R1(config-if)#exit

R1(config)#interface serial 0

R1(config-if)#ip nat outside
```

Scenario 12-4: Combining Static NAT Translation

In this scenario, you combine the functionality of the previous three scenarios. In other words, you
configure R1 so that when it receives a packet with a source address of 10.10.1.100 on its inside in
it translates it to 10.10.14.100. You also configure R1 so that when it receives a packet on its outsi

interface with a source address of 10.10.14.200, the source address is translated to 10.10.1.200. E 12-8 outlines a possible configuration for R1 that completes this scenario.

Example 12-8. Combining Static NAT Translations

```
! Configuration items for R1:

R1(config)#ip nat inside source static 10.10.1.100 10.10.14.100

R1(config)#ip nat outside source static 10.10.14.200 10.10.1.200

R1(config)#interface ethernet 0

R1(config-if)#ip nat inside

R1(config-if)#exit

R1(config)#interface serial 0

R1(config-if)#ip nat outside
```

Scenario 12-5: Overloading an IP Address with NAT

To complete this scenario, you configure R7 so that it uses Serial 0's IP address for overload. You a enable an outside e-mail server to originate traffic on port 25 to your Loopback 0 address. Exampl illustrates the overload keyword in a configuration.

Example 12-9. overload Keyword

```
! Configuration items for R7:

R7(config)#ip nat inside source list 7 interface serial 0 overload

R7(config)#ip nat inside source static tcp 10.10.7.7 25 10.10.14.7 25
```

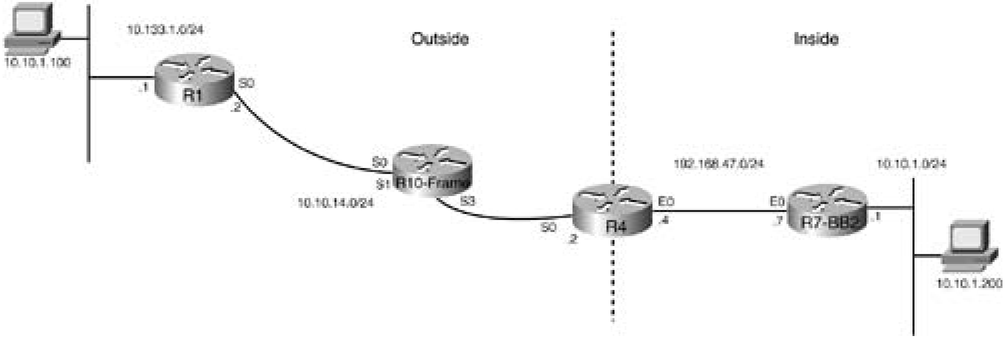
By using the overload keyword and associating it with an interface, you allow more than one insid address to be dynamically translated to the same global address. You also add a second entry to st configure NAT so that packets sourced from local address 100.133.7.7 with TCP port 25 (SMTP) are translated to Serial 0's IP address with TCP port 25. This static NAT entry gives e-mail servers on t outside the ability to originate SMTP (TCP port 25) packets to the global address of 10.10.14.7.

Scenario 12-6: Using NAT with Overlapping Addresses

In this scenario, you use the topology illustrated in Figure 12-4.

Figure 12-4. Scenario 12-4 Topology

[View full size image]



You first need to configure R7 in a manner that will allow it to translate the inside device located at 10.10.1.200 to an address from a NAT pool you will configure. You also need to configure a second translate the outside device located at 10.10.1.100 to a second NAT pool. Example 12-10 illustrate configuration required on R7.

Example 12-10. NAT Pools for Overlapping Networks

```
! Configuration items for R7:

R7(config)#ip nat pool inside 192.168.48.200 192.168.48.205 prefix-length 24

R7(config)#ip nat pool outside 192.168.48.210 192.168.48.215 prefix-length 24

R7(config)#ip nat inside source list 7 pool inside

R7(config)#ip nat outside source list 7 pool outside

R7(config)#interface loopback 0

R7(config-if)#ip address 10.10.7.7 255.255.255.0

R7(config-if)#ip nat inside

R7(config-if)#exit
```

```
R7(config)#interface ethernet 0

R7(config-if)#ip address 192.168.47.7 255.255.255.0

R7(config-if)#ip nat outside

R7(config-if)#exit

R7(config)#ip route 0.0.0.0 0.0.0.0 192.168.47.1

R7(config)#access-list 7 permit 10.10.1.0 0.0.0.255
```

When your inside device sends a DNS query to the DNS server residing outside the NAT domain, th query source address (the address of the inside device) is translated because of the ip nat inside commands. When the DNS server sends a DNS reply, the DNS reply payload gets translated becau ip nat outside commands. If you didn't have this static entry, NAT would not look at the DNS repl payload.

When you are trying to establish connectivity between two overlapping networks by running dynam on a single Cisco router, you must use DNS to create an outside-local-to-outside-global translation choose not to use DNS, you can still gain connectivity with static NAT, but it will be more difficult fo manage.

Scenario 12-7: Configuring TCP Load Distribution

In this scenario, your goal is to define a virtual address to distribute connections among a set of re You define a pool containing the addresses of the real hosts. You define an access control list (ACL) specifies the virtual address. If a translation does not already exist, TCP packets from the outside r on serial 0 with destinations that match your defined ACL are translated to an address from the po Example 12-11 shows a configuration to complete this example.

Example 12-11. Load-Balancing Example

```
! Configuration items for R4:


R4(config)#ip nat pool real-hosts 192.168.50.3 192.168.50.15 prefix-length 28 type

R4(config)#ip nat inside destination list 2 pool real-hosts

R4(config)#interface serial 0

R4(config-if)#ip address 192.168.50.129 255.255.255.240

R4(config-if)#ip nat outside
```

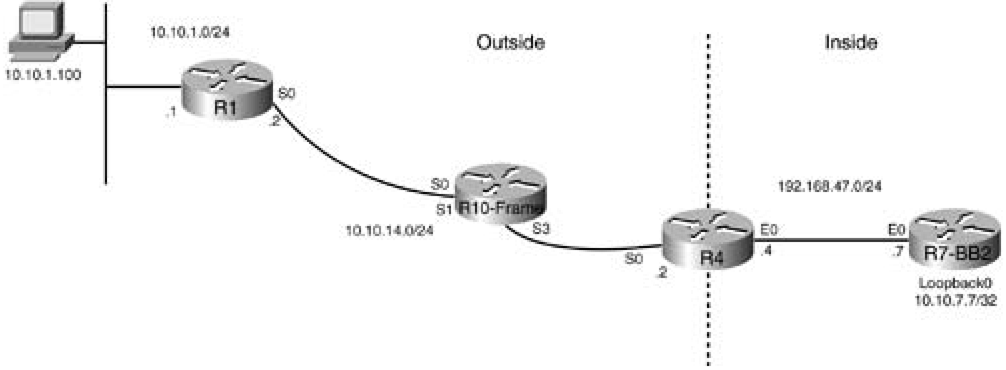
```
R4(config-if)#exit

R4(config)#interface ethernet 0

R4(config-if)#ip address 192.168.50.1 255.255.255.240

R4(config-if)#ip nat inside

R4(config-if)#exit

R4(config)#access-list 2 permit 192.168.50.2
```

Practical Exercise 12-1: Dynamic NAT Using an Outside Source List

In some situations you might need to use dynamic NAT instead of static NAT. One such situation is when you receive only a single routable IP address from your ISP. In this case, you need to configure an access list and associate it with an ip nat command to translate the IP addresses. In this Practical Exercise, you will configure the topology shown in Figure 12-5 using the ip nat outside source list command. This allows traffic from the host at 10.10.1.100 to reach the address of R7's Loopback 0 interface.

Figure 12-5. Outside Source List Topology

[View full size image]



You will assign an access list on R4 so that any packet sourced from 10.10.1.100 to 10.10.7.7 will be translated from a NAT pool to 172.16.48.250. You need to set your routing up so that none of the packets will be dropped, regardless of the address in use.

Practical Exercise 12-1 Solution

1. Configure your network interfaces:

```
! Configuration items for R1:

R1(config)#interface ethernet 0
R1(config-if)#ip address 10.10.1.1 255.255.255.0
R1(config-if)#exit
R1(config)#interface Serial0
R1(config-if)#ip address 10.10.14.1 255.255.255.252
R1(config-if)#exit

! Configuration items for R4:

R4(config)#interface serial 0
R4(config-if)#ip address 10.10.14.2 255.255.255.252
R4(config-if)#ip nat outside
R4(config-if)#exit
R4(config)#interface ethernet 0
R4(config-if)#ip address 172.16.47.1 255.255.255.0
R4(config-if)#ip nat inside
R4(config-if)#exit

! Configuration items for R7:
```

```
R7(config)#interface Loopback0
R7(config-if)#ip address 10.10.7.7 255.255.255.255
R7(config-if)#exit
R7(config)#interface ethernet 0
R7(config-if)#ip address 172.16.47.7 255.255.255.0
R7(config-if)#exit
```

2. Configure your static routing to ensure network connectivity. Remember that you can also use a routing protocol to accomplish this task.

```
! Configuration items for R1:

R1(config)#ip route 0.0.0.0 0.0.0.0 10.10.14.2

! Configuration items for R4:

R4(config)#ip route 10.10.1.0 255.255.255.0 192.168.14.1
R4(config)#ip route 10.10.7.7 255.255.255.255 172.16.47.7
R4(config)#ip route 172.16.48.254 255.255.255.0 Serial1

! Configuration items for R7:
R7(config)#ip route 0.0.0.0 0.0.0.0 172.16.47.1
```

3. Define an access list on R4 so that any traffic that originates from R7's loopback interface, 10.7.7.7/32, is dynamically translated:

```
R4(config)#access-list 1 permit 10.10.7.7 0.0.0.0
```

4. Define your dynamic NAT pool on R4. You will name the pool ccna_lab and give it an address range of 172.16.48.250 to 172.16.48.254. You will also associate this pool with an outside translation:

```
R4(config)#ip nat pool ccna_lab 172.16.48.250 172.16.48.254 netmask
  255.255.255.0
R4(config)#ip nat outside source list 1 pool ccna_lab
```

5. Assign the appropriate interfaces into NAT:

```
R4(config)#interface serial 0
R4(config-if)#ip nat outside
R4(config)#interface ethernet 0
R4(config-if)#ip nat inside
```

Practical Exercise 12-2: Combining Dynamic and Static NAT

In some situations, you might be required to combine dynamic NAT with static NAT. Before starting this Practical Exercise, you need to remember a few things. When you work with dynamic NAT, a translation does not exist in the NAT table until your router receives traffic that requires translation. A dynamic translation has a timeout period after which it is purged from your router's translation table. A static NAT translation exists in your router's NAT translation table as soon as you configure the static NAT command. It remains in the translation table until you delete the entry.

You will continue to use the topology outlined in Figure 12-5 for this Practical Exercise. The first task in merging dynamic and static NAT is to configure R4 so that outside devices address the Loopback 0 interface, 10.10.7.7, as 192.168.48.250. You will also configure a dynamic address pool of ten addresses starting at 192.168.48.200 for use in dynamic translation of R4's Ethernet segment.

Practical Exercise 12-2 Solution

These are the steps necessary to configure this Practical Exercise:

1. Configure your network interfaces:

```
! Configuration items for R1:

R1(config)#interface ethernet 0
R1(config-if)#ip address 10.10.1.1 255.255.255.0
R1(config-if)#exit
R1(config)#interface Serial0
R1(config-if)#ip address 10.10.14.1 255.255.255.252
R1(config-if)#exit

! Configuration items for R4:

R4(config)#interface serial 0
R4(config-if)#ip address 10.10.14.2 255.255.255.252
R4(config-if)#ip nat outside
R4(config-if)#exit
R4(config)#interface ethernet 0
R4(config-if)#ip address 172.16.47.1 255.255.255.0
R4(config-if)#ip nat inside
R4(config-if)#exit
```

```
! Configuration items for R7:

R7(config)#interface Loopback0

R7(config-if)#ip address 10.10.7.7 255.255.255.255

R7(config-if)#exit

R7(config)#interface ethernet 0

R7(config-if)#ip address 172.16.47.7 255.255.255.0

R7(config-if)#exit
```

2. Configure your static routing to ensure network connectivity. Remember that you can also use a routing protocol to accomplish this task.

```
! Configuration items for R1:


R1(config)#ip route 0.0.0.0 0.0.0.0 10.10.14.2


! Configuration items for R4:


R4(config)#ip route 10.10.1.0 255.255.255.0 192.168.14.1

R4(config)#ip route 10.10.7.7 255.255.255.255 172.16.47.7

R4(config)#ip route 172.16.48.254 255.255.255.0 Serial1


! Configuration items for R7:


R7(config)#ip route 0.0.0.0 0.0.0.0 172.16.47.1
```

3. Define an access list on R4 so that any traffic that originates from R4's Ethernet 0 network, 10.10.17.0/24, is dynamically translated:

```
R4(config)#access-list 1 permit 10.10.17.0 0.0.0.255
```

4. Define your dynamic NAT pool on R4. You will name the pool ccna_lab and give it an address range of 172.16.48.200 to 172.16.48.209. You will also associate this pool with an outside translation.

```
R4(config)#ip nat pool ccna_lab 172.16.48.200 172.16.48.209 netmask

  255.255.255.0

R4(config)#ip nat outside source list 1 pool ccna_lab
```

5. Assign the appropriate interfaces into NAT:

```
R4(config)#interface serial 0

R4(config-if)#ip nat outside

R4(config)#interface ethernet 0

R4(config-if)#ip nat inside
```

You can view the contents of your translation table by issuing the show ip nat translations command.Example 12-12 shows the output of this command when it is issued on R4.

Example 12-12. show ip nat translations Command Output on R4

```
R4#show ip nat translations

Pro Inside global    Inside local    Outside local    Outside global

--- 192.168.48.200   10.10.17.107      ---              ---
```

Notice that you see only the static translation you created in this output. This entry translates the inside global address back into the inside local address, giving devices on the outside of your network access to the Loopback 0 interface on your network.

Dynamic entries do not appear in the translation table until it receives a packet on its inside interface with a source address permitted by the ACL you created—in this case, ACL 7.

One point to note when working with dynamic NAT is that a device on the outside can't access a device governed by dynamic NAT if the translation does not exist. When your router receives a packet destined for one of the dynamic NAT global addresses, it checks its translation table for an existing translation. Because no match is found, it tries to route the packet, which in this case means back out the serial interface.

The dynamic NAT configuration you have done in this scenario works well when communication between inside and outside network devices is originated only by the inside devices. It does not work well if you decide to add an e-mail server on your network that needs to receive packets originated by the outside. The second part of this scenario is to configure a static NAT entry so that an e-mail server on the outside can originate communication with the e-mail server on your inside network.

Summary

In this chapter, you learned about the operation of NAT, how NAT translates addresses, and when to use NAT. You learned to configure NAT to translate private addresses in many different scenarios, such as configuring static NAT and dynamic NAT. You looked at some show command output as well as debug commands that relate to NAT to verify connectivity.

Review Questions

1: Network Address Translation is used to connect private IP internetworks that use __________ IP addresses to connect to the Internet.

A. routable

B. standard

C. nonroutable

D. nonstandard

2: When does the NAT operation take place on a router for inside-to-outside translation?

A. Before the IPSec operation

B. Before the routing decision

C. After the IPSec operation

D. After the routing decision

3: True or false: Cisco IOS NAT cannot be applied to subinterfaces.

4: What allows a single NAT-enabled router to allow some users to use NAT and other users on the same Ethernet interface to continue with their own IP addresses?

A. Access list

B. Route map

C. Policy map

D. Priority map

5: What is used to translate internal (inside local) private addresses to one or more outside (inside global—usually registered) IP addresses?

A. Overboard

B. Network Address Translation

C. Interface Address Translation

D. Port Address Translation

6: When using PAT, also known as NAT overloading, how many theoretical translations can be made for each inside global IP address?

A. 30,000

B. 25,655

C. 65,535

D. 100,000

7: PAT additionally translates which port to keep track of individual conversations?

A. Inside source

B. Outside source

C. Inside destination

D. Outside destination

8: IP address __________ refers to a situation in which two locations use the same IP address range but still want to communicate.

A. overloading

B. underloading

C. overlapping

D. underlapping

9: True or false: Static and dynamic NAT may be used on the same router.

Chapter 13. Using AAA to Scale Access Control in an Expanding Network

This chapter covers the following topics:

- AAA Overview
- Authentication
- Authorization
- Accounting
- AAA Protocols
- AAA Method List
- Configuring AAA

Access control in an expanding environment can be a daunting task. Authentication, authorization, and accounting (AAA) provides you with a mechanism you can use to track a user's access and usage. AAA also allows you to set the user's level of access, as well as what he may connect to and when he is allowed to connect.

AAA Overview

AAA combines three independent security functions in a modular fashion that allows you to configure access control to your network devices, such as routers and switches. The three modules you will be concerned with in this chapter are as follows:

- Authentication— Provides the methods you will use to identify your users before allowing them access to your network services. These methods include challenge and response, login and password dialog, encryption, and messaging support.
- Authorization— Provides the methods you will use for remote access control, such as per-user account list and profile, support of IP and Telnet, one-time authorization or authorization for each service, and user group support.
- Accounting— Provides the method you will use to collect and send security server information. You may use this information for auditing, billing, or reporting.

These modules are discussed further in the following sections.

Authentication

Authentication is the method used to identify your user before he or she is allowed access to your network and its services. A simple way of looking at configuring AAA authentication is defining a named list consisting of the authentication methods you want and then applying your defined list to your identified interface(s). You use the method list to define the types of authentication you want to be performed and the sequence in which you want them to be performed. With one exception, the method list named "default," you must apply the method list to a specific interface before any of your defined authentication methods are used. The default method list is automatically applied to any interface you have not applied a method list to. You must define all authentication methods, with the exception of local, line password, and enable authentication, through AAA. When you choose to implement authorization, your users must be authenticated before any authorization can take place.

Authorization

Authorization is designed to work by assembling a set of attributes you define to determine if a user is authorized to perform a certain task. Your defined attributes are compared to the information stored in the database for a given user. The result (the user's capabilities and restrictions) is returned to AAA. You can define the database locally on the network device or host it remotely on a RADIUS or TACACS+ security server, such as Cisco Secure Access Control Server (ACS). TACACS+ and RADIUS security servers authorize your users for their specific rights by using attribute-value (AV) pairs, which associate their rights with the appropriate user. All authorization methods must be defined through AAA. Just like authentication, you configure AAA authorization through the use of a named list of authorization methods and then apply your defined list to your specific interface(s).

Accounting

Accounting lets you track the services your users are accessing, as well as the amount of network resources they are consuming. AAA accounting accomplishes this by reporting your user's activity to the RADIUS or TACACS+ security server in the form of accounting records. These accounting records are comprised of accounting AV pairs. They are stored on the ACS for future analysis of network management, client billing, and/or auditing. You must define all the accounting methods through AAA. Much like the previous AAA modules, you configure AAA accounting through the use of named lists defining your accounting methods and then apply that list to your specified interface(s).

AAA Protocols

AAA uses two major security server protocols—TACACS+ and RADIUS. You can use either of these protocols to authenticate a large number of your users, because each creates a database of usernames and passwords. Both protocols share many features, because Cisco Systems modeled the TACACS+ architecture after the existing RADIUS standard. You can implement a TACACS+ or RADIUS server on a UNIX platform or Windows platform.

RADIUS is covered in the following RFCs:

- RFC 2138, *Remote Authentication Dial In User Service (RADIUS)*
- RFC 2139, *RADIUSAccounting*
- RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*
- RFC 2866, *RADIUSAccounting*
- RFC 2867, *RADIUSAccounting Modifications for Tunnel Protocol Support*
- RFC 2868, *RADIUSAttributes for Tunnel Protocol Support*
- RFC 2869, *RADIUSExtensions*

TACACS+ is covered by the following Internet Draft and RFC:

- *The TACACS+ Protocol Version 1.78* (draft-grant-tacacs-02.txt)
- RFC 1492, *An Access Control Protocol, Sometimes Called TACACS*

AAA Transport Protocols

Just like any packet that travels across your IP network, both TACACS+ and RADIUS use the TCP/IP stack. This is also one area in which they differ: RADIUS uses the UDP protocol for communications between the client and the security server, whereas TACACS+ uses the TCP protocol. TACACS+ operates over TCP port 49, and RADIUS operates over UDP port 1812 for authentication and UDP port 1813 for accounting. In some RADIUS implementations, you might see RADIUS operate over port 1645 for authentication and port 1646 for accounting.

Packet Encryption

One other area in which RADIUS and TACACS+ differ is their use of encryption. RADIUS encrypts only the user password in a client-to-server access request packet. Other items in the packet, such as username, authorized services, and accounting, are sent across the network in clear text.

TACACS+ encrypts the entire packet to the server with the exception of the unencrypted TACACS+ header. This unencrypted header contains a field specifying whether that packet's payload is encrypted.

AAA Method Lists

You create a method list by defining a sequential list of authentication methods that you want to use to authenticate a user. Method lists let you define a backup authentication system for authentication in case of a failure by configuring one or more security protocols to be used for authentication. Your network devices will use the first method you list to authenticate users; in the case of a failure, your network devices will use the next authentication method defined in the method list. This process continues until either your user is authenticated through the successful communication with a listed authentication method or the authentication method list is exhausted, in which case authentication fails. Authentication with the next defined authentication method is tried only if there is no response from the previous authentication method.

NOTE

A FAIL response differs from an ERROR response. A FAIL signals that the user does not meet the defined criteria required to be authenticated. The authentication process stops when a FAIL response is returned. However, an ERROR indicates that the security server has not responded to an authentication query. Because authentication has not been attempted, AAA selects the next authentication method you defined in the authentication method list and reattempts authentication.

Configuring AAA

After you decide which AAA service you want to use, you can use the following steps to configure AAA on your network device:

Step 1. Enable AAA.

Step 2. Configure security protocol parameters.

Step 3. Define the method lists for authentication.

Step 4. Apply the method lists to a particular interface or line.

Step 5. Optionally configure authorization.

Step 6. Optionally configure accounting.

Step 1: Enabling AAA

Before you can use the AAA network security services available to you, you must enable AAA. To accomplish this, use the following command:

```
R8(config)#aaa new-model
```

NOTE

Upon enabling AAA, IOS no longer lets you use the older TACACS or extended TACACS protocols.

If desired, you can disable AAA functionality using the following command:

```
R8(config)#no aaa new-model
```

Step 2: Configuring Security Protocol Parameters

Deciding which parameters you want to configure for your selected security protocol depends on the protocol you want to use. Because the parameters are protocol-specific, they are explained in the following sections.

Step 3: Defining the Method Lists for Authentication

AAA security services offer many varied authentication methods:

- Login authentication
- PPP authentication
- ARAP authentication
- NASI authentication

You also have the option of defining the following parameters:

- Specifying the amount of time for login input
- Enabling password protection at the privileged level
- Changing the text displayed at the password prompt
- Configuring message banners for AAA authentication
- Configuring AAA packet of disconnect
- Enabling double authentication
- Enabling automated double authentication

Some of these items are discussed further in the following sections.

Configuring Login Authentication Using AAA

Login authentication is used to enable AAA authentication regardless of the supported login authentication method you decide to use. You create one or more lists of authentication methods that will be tried at login and apply them to the login authentication command. To configure a login authentication list using AAA, use this command:

```
R8(config)#aaa authentication login {default | list-name}method1 [method2...]
```

list-name is a character string you use to name the list you are creating. The *method* arguments refer to the actual method the authentication algorithm tries. If you specify more than one method of authentication, they are used only if the previous method returns an error, not if it fails. You can use the none keyword as the final method in the command line to specify that authentication should succeed even if all other defined methods return an error. By using the default keyword, you can specify a default list that is applied to all interfaces automatically. Table 13-1 lists the wide variety of supported login authentication methods.

Table 13-1. AAA Login Authentication Methods

Keyword	Description
enable	The enable password is used for authentication.
line	The line password is used for authentication.
local	The local username database is used for authentication.
local-case	Makes the local username case-sensitive.
none	No authentication is used.
group radius	The list of all defined RADIUS servers is used for authentication.
group tacacs+	The list of all defined TACACS+ servers is used for authentication.
group*group-name*	A subset of RADIUS or TACACS+ servers, defined by the aaa group server radius or aaa group server tacacs+ command, is used for authentication.
krb5	Kerberos 5 is used for authentication.
krb5-telnet	When using Telnet to connect to the device, the Kerberos 5 Telnet authentication protocol is used for authentication. This keyword must be the first method in the method list.

Configuring PPP Authentication Using AAA

Your network might require giving your users remote access through some type of dialup connection, such as async or ISDN through an access server. Both of these dialup services present a unique problem when you are trying to control access through AAA. Neither uses the command-line interface of the network device. Instead, they start a network protocol, such as PPP or ARA, as soon as the connection is established. Fortunately, the AAA security service provides a solution to this problem by offering a variety of authentication methods for use on serial interfaces using PPP.

You can use the following command to configure AAA authentication methods for serial lines using PPP. It creates a local authentication list:

```
R8(config)#aaa authentication ppp {default | list-name}method1 [method2...]
```

Table 13-2 lists the authentication methods available with PPP authentication.

Table 13-2. AAA Authentication Methods for PPP

Keyword	Description
if-needed	No authentication is required if the user has already been authenticated on a TTY line.
local	The local username database is used for authentication.
local-case	A case-sensitive local username is used for authentication.
none	No authentication is attempted.
group radius	A defined list of all RADIUS servers is used for authentication.
group tacacs+	A defined list of all TACACS+ servers is used for authentication.
group*group-name*	A subset of RADIUS or TACACS+ servers, defined by the aaa group server radius or aaa group server tacacs+ command, is used for authentication.
krb5	When used with PAP authentication, Kerberos 5 is used for authentication.

Configuring ARAP Authentication Using AAA

You can use the following command to configure AAA authentication with the AppleTalk Remote Access Protocol (ARAP). It enables authentication for ARAP users:

```
R8(config)#aaa authentication arap {default | list-name}method1 [method2...]
```

Table 13-3 lists ARAP's supported login authentication methods.

Table 13-3. AAA Authentication Methods for ARAP

Keyword	Description
auth-guest	Guest logins are allowed if the user has already logged into EXEC.
guest	Guest logins are allowed.
line	The line password is used for authentication.
local	The local username database is used for authentication.
local-case	A case-sensitive local username is used for authentication.
group radius	A defined list of all RADIUS servers is used for authentication.
group tacacs+	A defined list of all TACACS+ servers is used for authentication.

Configuring NASI Authentication Using AAA

When a user attempts to log into the device using the NetWare Asynchronous Services Interface (NASI), you can use the following commands. It enables authentication for NASI users:

```
R8(config)#aaa authentication nasi {default | list-name}method1 [method2...]
```

Table 13-4 lists the NASI authentication methods you may choose from.

Table 13-4. AAA Authentication Methods for NASI

Keyword	Description
enable	The enable password is used for authentication.
line	The line password is used for authentication.
local	The local username database is used for authentication.
local-case	Makes the local username case-sensitive.
none	No authentication is used.
group radius	The list of all defined RADIUS servers is used for authentication.
group tacacs+	The list of all defined TACACS+ servers is used for authentication.
group*group-name*	A subset of RADIUS or TACACS+ servers, defined by the aaa group server radius or aaa group server tacacs+ command, is used for authentication.

Specifying the Amount of Time for Login Input

By default, the system waits 30 seconds for login input before timing out. You can use the following command to change this amount of time:

```
R8(config-line)#timeout login responseseconds
```

Enabling Password Protection at the Privileged Level

You can require a user to be authenticated by the AAA subsystem when entering the privileged EXEC command level (the "enable" level) using the following command:

```
R8(config)#aaa authentication enable defaultmethod1 [method2...]
```

Requests for authentication sent to a RADIUS server include a username of $enab15$. Requests sent to a TACACS+ server include the username that is entered for login authentication.

Table 13-5 lists the supported enable authentication methods.

Table 13-5. AAA Authentication Methods for Enable

Keyword	Description
enable	The enable password is used for authentication.
line	The line password is used for authentication.
none	No authentication is used.
group radius	The list of all defined RADIUS servers is used for authentication.
group tacacs+	The list of all defined TACACS+ servers is used for authentication.
group*group-name*	A subset of RADIUS or TACACS+ servers, defined by the aaa group server radius or aaa group server tacacs+ command, is used for authentication.

Step 4: Applying the Method Lists to a Particular Interface or Line

After you have defined your method list, the next step is to apply it to either a line or an interface. You can use one of the following commands to enter line or interface configuration mode.

Use this command to enter line configuration mode if you want to apply your method list to a line:

```
R8(config)#line [aux | console | tty | vty]line-number [ending-line-number]
```

Use this command to enter interface configuration mode if you want to apply your method list to an interface:

```
R8(config)#interfaceinterface-type interface-number
```

You can use the following command to apply your login method list to a line or set of lines:

```
R8(config-line)#login authentication {default | list-name}
```

You can use the following command to apply the PPP authentication list to a line or set of lines. *protocol1* and *protocol2* represent the CHAP, MS-CHAP, and PAP protocols.

```
R8(config-if)#ppp authentication {protocol1 [protocol2...]} [if-needed]
  {default | list-name} [callin] [one-time]
```

You can use the following command to optionally enable autoselection of ARAP under a line:

```
R8(config-line)#autoselect arap
```

You can use the following command to optionally start the ARAP session automatically during user login:

```
R8(config-line)#autoselect during-login
```

You can use the following command to optionally enable TACACS+ authentication on a line:

```
R8(config-line)#arap authenticationlist-name
```

You can use the following command to optionally enable NASI authentication on a line:

```
R8(config-line)#nasi authenticationlist-name
```

Step 5: Optionally Configuring Authorization

AAA authorization builds on AAA authentication by allowing you to limit which of your services a user can access. With AAA authorization, a user's profile is used to retrieve information from the local user database or the security server to configure the user's session to grant access to a requested service. Access is allowed only if you granted the access in the user's profile.

Much like method lists you configure for authentication, a method list for authorization defines the manner in which authorization will be performed, as well as the sequence in which these methods will be executed. Several different authorization types are available for you to define in your method lists:

- Commands— Used to apply authorization to the EXEC mode commands a user may use. Command authorization is attempted for all EXEC mode commands associated with a specific privilege level.
- EXEC— Applies to the user's attributes during an EXEC terminal session.
- Network— Used with network connections.
- Auth-proxy— Used to apply security policies on a per-user basis.
- Reverse-access— Used with reverse-Telnet sessions.

AAA gives you five different methods you can use with authorization:

- None— Authorization information is not requested or required.
- Local— A local database, defined by the username command, is consulted for authorization.
- If-Authenticated— If the user was previously authenticated, he or she is allowed access to the function without further authorization.
- TACACS+— A TACACS+ security daemon is used for authorization defined by associating attribute-value pairs with a user's assigned rights.
- RADIUS— A RADIUS security server is used for authorization defined by associating attribute-value pairs with a user's assigned rights.

Before you can configure AAA authorization, you must perform the following tasks:

- Enable AAA on your network device.
- Configure AAA authentication, because authorization requires authentication to work properly.
- Define the characteristics of your security server if you are defining RADIUS or TACACS+ authorization.
- Define a local database. Use the username command if you are using local authorization.

Both RADIUS and TACACS+ authorization use attributes to define the specific rights you want to grant your users. The attributes for both RADIUS and TACACS+ are defined on the security server, associated with your user, and sent to your network device, when requested. There the attributes are applied to your user's connection. Because both TACACS+ and RADIUS support many different attributes, you should consult your server's documentation to determine which attributes you can use.

Configuring AAA Authorization

Three steps are required to configure AAA authorization:

Step 1. Configure AAA authorization with named method lists.

Step 2. Disable AAA authorization for global configuration commands.

Step 3. Configure AAA authorization for reverse Telnet.

Each of these steps is looked at in further detail in the following sections.

Step 1: Configuring AAA Authorization with Named Method Lists

You can use the following command to configure AAA authorization for a particular authorization type and enable authorization using named method lists:

```
R8(config)#aaa authorization {auth-proxy | network | exec | commandslevel |
  reverse-access | configuration | ipmobile} {default | list-name} [method1
  [method2...]]
```

You can use one of the following commands to alternatively apply your authorization list to an interface or set of interfaces:

```
R8(config-line)#authorization {arap | commandslevel | exec | reverse-access}
  {default | list-name}
```

or

```
R8(config-line)#ppp authorization {default | list-name}
```

Step 2: Disabling AAA Authorization for Global Configuration Commands

If you decide to implement AAA authorization for all EXEC mode commands, you might encounter a problem in which AAA authorization becomes confused by the fact that some configuration commands are identical to some EXEC-level commands. You can prevent this behavior by stopping your network device from attempting configuration command authorization using the following command:

```
R8(config)#no aaa authorization config-commands
```

Step 3: Configuring AAA Authorization for Reverse Telnet

In most circumstances, you will use Telnet to gain remote access to your network devices. Other times, you might be required to establish a reverse-Telnet session to a device. A reverse-Telnet session is simply a Telnet connection that you establish in the opposite direction you normally would, such as from inside your network to an access server on your network edge, to gain access to a modem. You would also use reverse Telnet to provide your users with dial-out capability using Telnet to access modem ports attached to your access server.

Authentication during reverse Telnet is accomplished using the standard AAA login procedure specified for Telnet. In other words, the user provides a username and password to establish either a Telnet or reverse-Telnet session. Reverse Telnet builds on AAA authentication by providing a second level of security by requiring the additional step of authorization before authentication is completed. Reverse-Telnet authorization also provides the following benefits:

- It ensures that users attempting to gain access to reverse-Telnet activities are authorized to access a specific asynchronous port using reverse Telnet.
- It provides a second method of managing reverse-Telnet authorization.

You can configure your network device to request authorization information from a TACACS+ or RADIUS server before allowing a user to establish a reverse-Telnet session by using the following command:

```
R8(config)#aaa authorization reverse-accessmethod1 [method2...]
```

Although enabling this feature lets your network device request reverse-Telnet authorization information from the security server, you still have to configure the specific privileges for your user regarding reverse Telnet.

Step 6: Optionally Configuring Accounting

AAA accounting lets you track the services your users are accessing and the amount of network resources they are consuming. Your network device reports your users' activities to your TACACS+ or RADIUS security server in the form of accounting records. Each accounting record is composed of accounting AV pairs and is stored on the security server.

Much like authentication and authorization, AAA accounting uses method lists to define the manner and order in which accounting will be performed. Named accounting method lists let you designate specific security protocols for specific lines or interfaces, with the default method list, the only exception, automatically applied to all interfaces that you have not defined a named method list explicitly for. You can define a method list for each specific type of accounting you are interested in. Six different types of AAA accounting are supported:

- Network— Supplies information on all PPP, SLIP, or ARAP sessions.
- EXEC— Supplies information on user EXEC sessions on your network devices.
- Commands— Supplies information about commands a user issues while in EXEC mode for a specific privilege level.
- Connection— Supplies information about outbound connections, such as Telnet, made from your network device.
- System— Supplies information about system-level events. System accounting can be defined only with the default list for AAA accounting.
- Resource— Supplies "start" and "stop" records for calls that have passed user authentication. Also provides "stop" records for calls that fail to authenticate.

After they are defined, you must apply your AAA accounting method lists to specific lines or interfaces before any of your defined methods are performed. If you use the aaa accounting command for a particular accounting type without specifying a named method list, the default method list is automatically applied. If you do not define a default method list, you cannot use accounting.

Currently, only two accounting methods are supported:

- TACACS+— User activity is reported to the TACACS+ security server in the form of accounting records. Each accounting record is composed of accounting AV pairs and is stored on the security server.

- RADIUS— User activity is reported to the RADIUS security server in the form of accounting records. Each accounting record is composed of accounting AV pairs and is stored on the security server.

"Start" and "stop" records are provided by AAA accounting for calls that have passed user authentication so that you may manage and maintain your network. These "start" and "stop" records, called start-stop records, send a "start" record at every call setup and a corresponding "stop" record at the call's completion. A second start-stop record lets you track a user's management progress. Both of these start-stop accounting records can be associated with each other through the use of a unique session ID for the call. Additionally, "stop" records are provided for calls that fail to reach the user authentication stage of a call setup sequence. If you choose to do so, you can disable the sending of a "start" record, because most of the information in the typical "start" record is also included in the "stop" record.

AAA Broadcast Accounting

If your networking environment has several AAA servers, you can take advantage of the AAA broadcast feature. The AAA broadcast feature for accounting allows accounting information to be broadcast to several AAA servers at the same time.

Broadcasting can be used for a group of RADIUS or TACACS+ servers. Each server group can define backup servers for failover independently of other groups.

Before you can successfully configure AAA accounting through named method lists, you complete the following tasks:

- Configure and enable AAA on your network devices.

- If you are using RADIUS or TACACS+ authorization, you must define the characteristics of your RADIUS or TACACS+ security server.

Configuring AAA Accounting

You follow these steps to configure AAA accounting:

Step 1. Configure AAA accounting named method lists.

Step 2. Suppress generation of accounting records for null username sessions.

Step 3. Generate interim accounting records.

Step 4. Generate accounting records for the failed login or session.

Step 5. Specify accounting NETWORK-Stop records before EXEC-Stop records.

Step 6. Configure AAA resource failure stop accounting.

Step 7. Configure AAA resource accounting for start-stop records.

Step 8. Configure AAA broadcast accounting.

Each of these configuration tasks is discussed in further detail in the following sections.

Step 1: Configuring AAA Accounting Named Method Lists

AAA accounting named method lists are specific to the indicated type of accounting:

- network— Used to create a method list to enable authorization for all network-related service requests.
- exec— Used to create a method list to provide accounting records detailing user EXEC terminal sessions on the network devices.
- commands— Used to create a method list for accounting information about specific, individual EXEC commands associated with a specific privilege level.
- connection— Used to create a method list for accounting information about all outbound connections made from the network device.
- resource— Used to create a method list that provides accounting records for calls that have passed user authentication or calls that failed to be authenticated.

If you want to receive only a minimal amount of accounting information, you can use the stop-only keyword. This keyword instructs the specified method, whether RADIUS or TACACS+, to send a stop record accounting notice only at the end of the requested user process. If you want to receive more accounting information, you can use the start-stop keyword to send a start accounting notice at the beginning of the requested event and a stop accounting notice at the completion of the event. If you do not want to receive any accounting information from a line or interface, you can use the none keyword.

You use the method argument to refer to the actual method that AAA uses to determine whether to report accounting information. AAA accounting supports the following methods:

- group radius— Specifies a list of all RADIUS servers for accounting.
- group tacacs+— Specifies a list of all TACACS+ servers for accounting.
- group*group-name*— Specifies a subset of RADIUS or TACACS+ servers for accounting that you define using the server group *group-name*.

AAA accounting supports the following methods to determine where to send accounting records:

- group tacacs— Tells the network device to send accounting information to a TACACS+ security server.
- group radius— Tells the network device to send accounting information to a RADIUS security server.
- group*group-name*— Specifies a subset of RADIUS or TACACS+ servers to use as the accounting method.

You can use the following commands to create an accounting method list and enable accounting:

```
R8(config)#aaa accounting {system | network | exec | connection | commandslevel}
  {default | list-name} {start-stop | stop-only | none} [method1 [method2...]]
```

After you create your accounting method list, you can use one of the following commands to apply the method list to a line or interface:

```
R8(config-line)#accounting {arap | commandslevel | connection | exec} {default |
  list-name}
```

or

```
R8(config-if)#ppp accounting {default | list-name}
```

Step 2: Suppressing Generation of Accounting Records for Null Username Sessions

AAA accounting generates accounting records for all users on the system, including users whose username string is NULL, because of protocol translation. You can use the following command to prevent the generation of accounting records for NULL username sessions:

```
R8(config)#aaa accounting suppress null-username
```

Step 3: Generating Interim Accounting Records

When you use the aaa accounting update command, your network device sends interim accounting records for all users currently using the device. You can use the newinfo keyword to send interim accounting records to your accounting server whenever new accounting information is generated.

When you use the periodic keyword, interim accounting records are generated periodically as often as defined by the *number* argument. The interim accounting record is composed of all the accounting information recorded for that user up to the time the interim accounting record is sent. You can use the following command to enable generation of periodic interim accounting records:

```
R8(config)#aaa accounting update {[newinfo] [periodic]number}
```

Step 4: Generating Accounting Records for the Failed Login or Session

AAA accounting does not, by default, generate accounting records for users who fail login authentication or who succeed in login authentication but fail PPP negotiation for some reason. You can use the following command to generate accounting stop records for users who fail to authenticate at login or during session negotiation:

```
R8(config)#aaa accounting send stop-record authentication failure
```

Step 5: Specifying Accounting NETWORK-Stop Records Before EXEC-Stop Records

If you are required by your company policies to keep your network start-stop records together, such as for billing purposes, you can specify that NETWORK records be generated before EXEC-stop records. You can use the following command to nest accounting records for user sessions:

```
R8(config)#aaa accounting nested
```

Step 6: Configuring AAA Resource Failure Stop Accounting

You can use the following command to enable resource failure stop accounting to generate a "stop" record for any call that does not reach user authentication:

```
R8(config)#aaa accounting resourcemethod-liststop-failure groupserver-group
```

Step 7: Configuring AAA Resource Accounting for Start-Stop Records

You can use the following command to enable full resource accounting for start-stop records:

```
R8(config)#aaa accounting resourcemethod-liststart-stop groupserver-group
```

Step 8: Configuring AAA Broadcast Accounting

You can use the following command to configure AAA broadcast accounting by modifying the aaa accounting command with the broadcast keyword:

```
R8(config)#aaa accounting {system | network | exec | connection | commandslevel}
  {default | list-name} {start-stop | stop-only | none} [broadcast]method1
  [method2...]
```

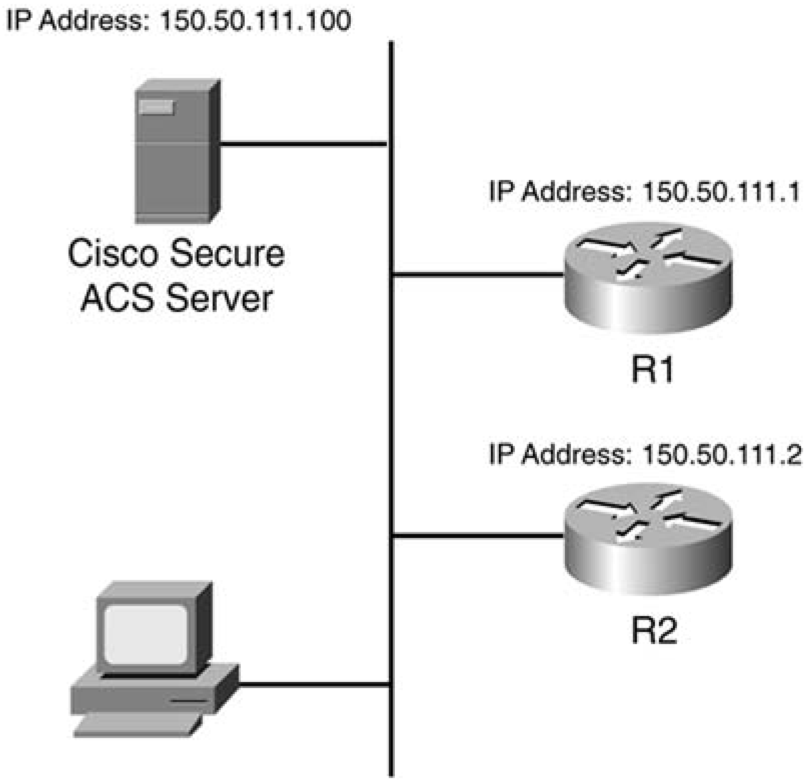
You also can configure AAA broadcast accounting for dialed number identification service (DNIS) on a per-call basis by modifying the aaa dnis map accounting network command with the broadcast keyword:

```
R8(config)#aaa dnis mapdnis-numberaccounting network [start-stop | stop-only |
  none] [broadcast]method1 [method2...]
```

Scenarios

This section offers some examples of configuring authentication, authorization, and accounting. Cisco Secure ACS server is used as the TACACS+ and RADIUS server. Figure 13-1 illustrates the lab topology that is used throughout various scenarios.

Figure 13-1. Lab Topology for AAA Configuration



Scenario 13-1: Configuring Authentication Using TACACS+

In this scenario, you configure authentication using TACACS+. The default login is the TACACS+ server. If there is no response from the server, use the local username/password database or enable secret. Authentication is applied to the Telnet session but not to the console port. The TACACS+ server is configured with R1's Ethernet IP address and uses the key cisco. Example 13-1 shows you the commands to configure this scenario.

Example 13-1. Authentication Commands with TACACS+

```
service timestamps debug datetime msec
service timestamps log datetime msec

no service password-encryption

!

hostname R1

!

aaa new-model

aaa authentication login default group tacacs+ local enable

aaa authentication login no_login none

enable secret 5 $1$mKTM$dS1tLOKpFMXI1gbcmdoMe0

!

username raymond password 0 raymond

username wesley password 0 wesley

memory-size iomem 15

ip subnet-zero

!

!

!

interface Ethernet0/0

 ip address 150.50.111.1 255.255.255.0

 half-duplex

!

interface Ethernet0/1

 no ip address

 shutdown

 half-duplex

!

ip classless

ip tacacs source-interface Ethernet0/0
ip http server
!
!
!
tacacs-server host 150.50.111.100 single-connection
tacacs-server key cisco
!
line con 0
login authentication no_login
line aux 0
line vty 0 4
!
!
end
```

Scenario 13-2: Configuring Authentication Using RADIUS

In this scenario, you configure authentication using RADIUS. The default login is the RADIUS server using the older RADIUS ports. If there is no response from the server, use the local username/password database or enable secret. Authentication is applied to the Telnet session but not to the console port. The RADIUS server is configured with R1's Ethernet IP address and uses the key cisco. Example 13-2 shows the commands to configure this scenario.

Example 13-2. Authentication Commands with RADIUS

```
no service single-slot-reload-enable
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R2
```

```
!
logging rate-limit console 10 except errors
aaa new-model
aaa authentication login default group radius local enable
aaa authentication login no_login none
enable secret 5 $1$mKTM$dS1tLOKpFMXI1gbcmdoMe0
!
username raymond password 0 raymond
username wesley password 0 wesley
memory-size iomem 15
ip subnet-zero
!
!
no ip finger
!
!
!
interface Ethernet0/0
 ip address 10.1.1.42 255.255.255.0
 half-duplex
!
interface Ethernet0/1
 no ip address
 shutdown
 half-duplex
!
ip classless
ip http server
```

```
!
ip radius source-interface Ethernet0/0
radius-server host 10.1.1.111.100 auth-port 1645 acct-port 1646 key cisco
radius-server retransmit 1
radius-server authorization permit missing Service-Type
!
!
!
!
line con 0
login authentication no_login
transport input none
line aux 0
line vty 0 4
!
no scheduler allocate
```

Scenario 13-3: Configuring Authorization Using TACACS+

In this scenario, you configure authentication and authorization using TACACS+. The default login is the TACACS+ server. If there is no response from the server, use the local username/password database or enable secret. Authentication and authorization are applied to the Telnet session but not to the console port. The TACACS+ server is configured with R1's Ethernet IP address and uses the key cisco. The aaa authorization exec command is used to determine if the user is allowed to access an EXEC shell and what shell attributes are permitted or denied. Also, you change the privilege level for certain commands that users are authorized to use.Example 13-3 demonstrates the commands to configure this scenario.

Example 13-3. Authentication and Authorization Commands with TACACS+

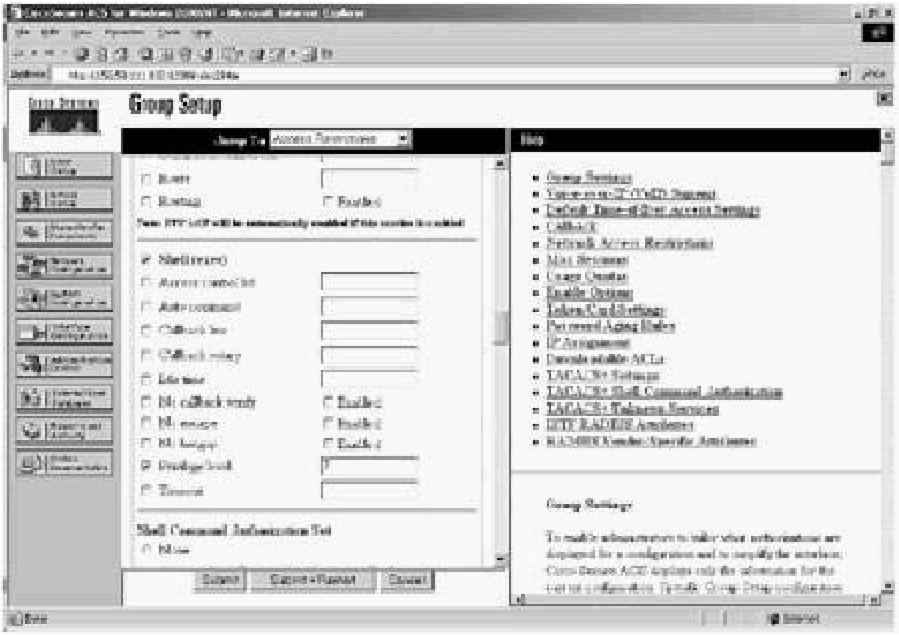
```
version 12.1
```

```
no service single-slot-reload-enable
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1
!
logging rate-limit console 10 except errors
aaa new-model
aaa authentication login default group tacacs+ local enable
aaa authentication login no_login none
aaa authorization exec default group tacacs+ local
aaa authorization exec no_login none
enable secret 5 $1$mKTM$dS1tLOKpFMXI1gbcmdoMe0
!
username raymond privilege 7 password 0 raymond
username wesley privilege 7 password 0 wesley
memory-size iomem 15
ip subnet-zero
!
interface Ethernet0/0
 ip address 150.50.111.1 255.255.255.0
 half-duplex
!
interface Ethernet0/1
 no ip address
 shutdown
 half-duplex
```

```
!
ip classless
ip tacacs source-interface Ethernet0/0
ip http server
!
tacacs-server host 150.50.111.100 single-connection key cisco
!
privilege configure level 7 ntp
privilege configure level 7 ntp server
privilege exec level 7 ping
privilege exec level 7 configure terminal
!
line con 0
authorization exec no_login
login authentication no_login
 transport input none
line aux 0
line vty 0 4
```

Figure 13-2 shows the configuration of Cisco Secure ACS server to assign privilege levels to users. In Group Settings, make sure that Shell(exec) is checked and that 7 is entered in the Privilege level box.

Figure 13-2. Cisco Secure ACS Configuration



Group Setup

Jump To Access Restrictions

□ Routing ☐ Enabled

Note: PPP LCP will be automatically enabled if this service is enabled

☑ Shell (exec)

☐ Access control list

☐ Auto command

☐ Callback line

☐ Callback rotary

☐ Idle time

☐ No callback verify ☐ Enabled

☐ No escape ☐ Enabled

☐ No hangup ☐ Enabled

☑ Privilege level 7

☐ Timeout

Shell Command Authorization Set

○ None

Help

- Group Settings
- Voice-over-IP (VoIP) Support
- Default Time-of-Day Access Settings
- Callback
- Network Access Restrictions
- Max Sessions
- Usage Quotas
- Enable Options
- Token Card Settings
- Password Aging Rules
- IP Assignment
- Downloadable ACLs
- TACACS+ Settings
- TACACS+ Shell Command Authorization
- TACACS+ Unknown Services
- IETF RADIUS Attributes
- RADIUS Vendor-Specific Attributes

Group Settings

To enable administrators to tailor what authorizations are displayed for a configuration and to simplify the interface, Cisco Secure ACS displays only the information for the current configuration. Specific Group Setup configuration

As shown in Example 13-4, after the user accesses the router and is authenticated, the show privilege command shows what the privilege is, and ? displays what commands are available with this privilege.

Example 13-4. Using the show privilege Command and Verifying Available Commands

```
R1#150.50.111.1

Trying 150.50.111.1 ... Open




User Access Verification



Username:wesley

Password:



R1#show privilege

Current privilege level is 7

R1#config t
```

```
Enter configuration commands, one per line.  End with CNTL/Z.

R1(config)#?

Configure commands:

  default  Set a command to its defaults

  end      Exit from configure mode

  exit     Exit from configure mode

  help     Description of the interactive help system

  no       Negate a command or set its defaults

  ntp      Configure NTP


R1(config)#
```

Scenario 13-4: Configuring Accounting Using TACACS+

In this scenario, you configure accounting with TACACS+. You configure the router to run start-stop accounting for all character mode service requests, all commands at privilege level 15, and all system-level events not associated with users, such as configuration changes and reloads. Example 13-5 shows the sample configuration of accounting.

Example 13-5. Some Accounting Commands with TACACS+

```
version 12.2

service timestamps debug datetime msec

service timestamps log datetime msec

no service password-encryption

!

hostname R1

!

logging queue-limit 100

enable secret 5 $1$mKTM$dS1tLOKpFMXI1gbcmdoMe0
```

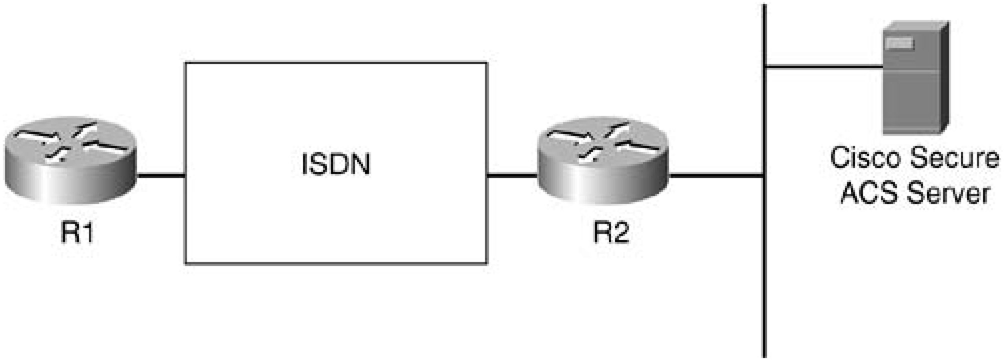
```
!
username wesley privilege 7 password 0 wesley
clock timezone PST -8
aaa new-model
!
!
aaa authentication login default group tacacs+ local enable
aaa authentication login no_login none
aaa authorization exec default group tacacs+ local
aaa authorization exec no_login none
aaa accounting exec default start-stop group tacacs+
aaa accounting commands 15 default start-stop group tacacs+
aaa accounting system default start-stop group tacacs+
aaa session-id common
ip subnet-zero
!
!
interface Ethernet0/0
 ip address 150.50.111.1 255.255.255.0
!
ip classless
ip tacacs source-interface Ethernet0/0
ip http server
no ip http secure-server
!
!
!
!
```

```
tacacs-server host 150.50.111.100 single-connection key cisco

tacacs-server directed-request

!

line con 0

 authorization exec no_login

 login authentication no_login

line aux 0

line vty 0 4

!

end
```

Practical Exercise: ISDN Callback Using TACACS+

In this Practical Exercise, you have a chance to work on the packet mode of AAA configuration. Two routers with ISDN BRI interface and Cisco Secure ACS server are used in this lab. R1 calls R2, and R2 points to the Cisco Secure ACS server running TACACS+ for user information. The TACACS+ server is configured to call back R1 when username R1 is sent by router R1. Figure 13-3 shows the lab topology for this Practical Exercise.

Figure 13-3. Cisco Secure ACS Configuration for ISDN Using TACACS+



Practical Exercise Solution

Example 13-6 shows the configuration of R1, which is the callback client.

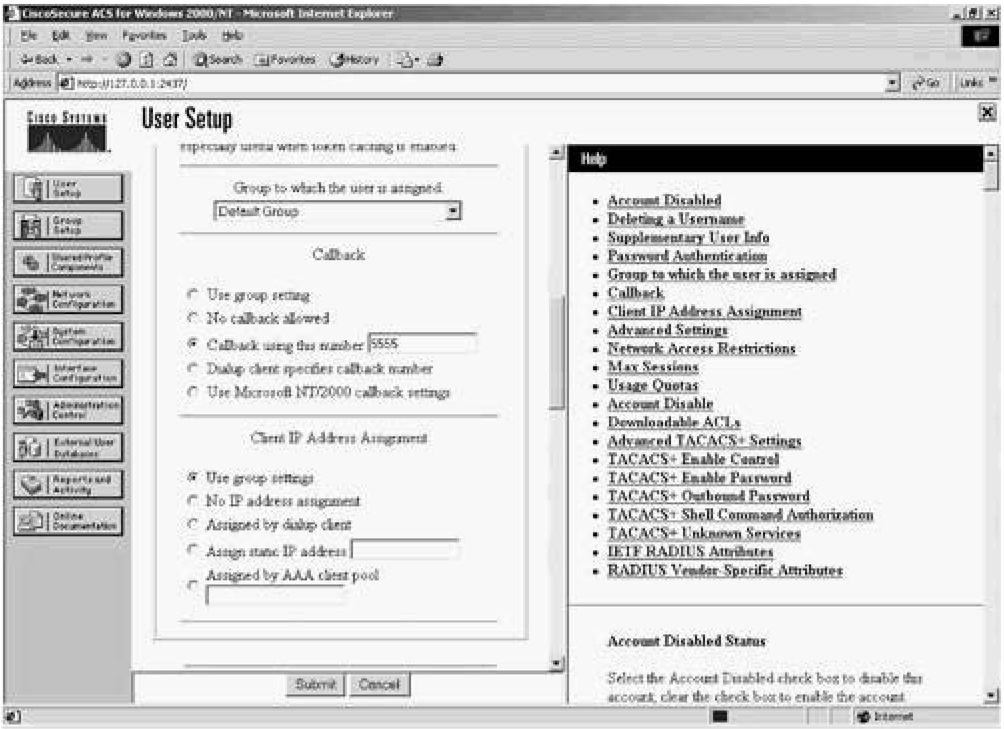
Example 13-6. R1 Configuration from the show running-config Command

```
service timestamps debug datetime msec

service timestamps log datetime msec

no service password-encryption

!

hostname R1

!

!

username R2 password 0 cisco

ip subnet-zero

!

!

no ip domain lookup

!

isdn switch-type basic-ni

!

interface FastEthernet0/0

 ip address 192.168.1.1 255.255.255.0

 !

interface BRI0/0

 no ip address

 encapsulation ppp

 dialer pool-member 1
 isdn switch-type basic-ni
 isdn tei-negotiation first-call
 isdn spid1 6661 5555
 isdn spid2 6662 5555
 no cdp enable
 ppp authentication chap
!
interface Dialer1
 ip address 172.16.35.1 255.255.255.0
 encapsulation ppp
 dialer pool 1
 dialer idle-timeout 60
 dialer string 6666
 dialer hold-queue 20
 dialer-group 1
 no peer default ip address
 no fair-queue
 no cdp enable
 ppp callback request
 ppp authentication chap
 ppp chap hostname R1
 ppp chap password 0 cisco
!
ip classless
ip route 0.0.0.0 0.0.0.0 BRI0/0
no ip http server
!
dialer-list 1 protocol ip permit
```

```
line con 0

line aux 0

line vty 0 4
```

Figures 13-4 and 13-5 illustrate the configuration of Cisco Secure ACS required to complete this Practical Exercise.

Figure 13-4. Cisco Secure ACS Configuration for ISDN Callback



User Setup

especially useful when token caching is enabled.

Group to which the user is assigned:

Default Group

Callback

○ Use group setting
○ No callback allowed
● Callback using this number 5555
○ Dialup client specifies callback number
○ Use Microsoft NT/2000 callback settings

Client IP Address Assignment

● Use group settings
○ No IP address assignment
○ Assigned by dialup client
○ Assign static IP address
○ Assigned by AAA client pool

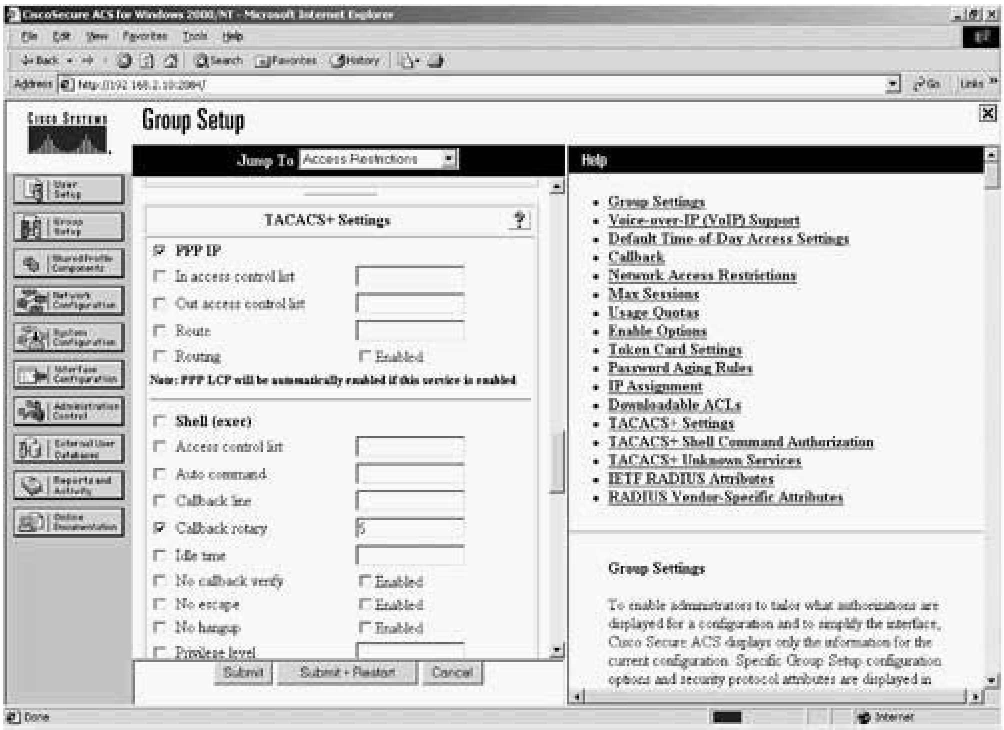
Help

- Account Disabled
- Deleting a Username
- Supplementary User Info
- Password Authentication
- Group to which the user is assigned
- Callback
- Client IP Address Assignment
- Advanced Settings
- Network Access Restrictions
- Max Sessions
- Usage Quotas
- Account Disable
- Downloadable ACLs
- Advanced TACACS+ Settings
- TACACS+ Enable Control
- TACACS+ Enable Password
- TACACS+ Outbound Password
- TACACS+ Shell Command Authorization
- TACACS+ Unknown Services
- IETF RADIUS Attributes
- RADIUS Vendor-Specific Attributes

Account Disabled Status

Select the Account Disabled check box to disable this account; clear the check box to enable the account.

Figure 13-5. Cisco Secure ACS Configuration for ISDN Callback



Group Setup

TACACS+ Settings

☑ PPP IP
☐ In access control list
☐ Out access control list
☐ Route
☐ Routing ☐ Enabled
Note: PPP LCP will be automatically enabled if this service is enabled

☐ Shell (exec)
☐ Access control list
☐ Auto command
☐ Callback line
☑ Callback rotary 5
☐ Idle time
☐ No callback verify ☐ Enabled
☐ No escape ☐ Enabled
☐ No hangup ☐ Enabled
☐ Privilege level

- Group Settings
- Voice-over-IP (VoIP) Support
- Default Time-of-Day Access Settings
- Callback
- Network Access Restrictions
- Max Sessions
- Usage Quotas
- Enable Options
- Token Card Settings
- Password Aging Rules
- IP Assignment
- Downloadable ACLs
- TACACS+ Settings
- TACACS+ Shell Command Authorization
- TACACS+ Unknown Services
- IETF RADIUS Attributes
- RADIUS Vendor-Specific Attributes

Group Settings

To enable administrators to tailor what authorizations are displayed for a configuration and to simplify the interface, Cisco Secure ACS displays only the information for the current configuration. Specific Group Setup configuration options and security protocol attributes are displayed in

Example 13-7 shows the configuration of R2, which is the callback server. Notice that with the aaa authorization network default group tacacs+ command, callback information is obtained from the TACACS+ server.

Example 13-7. R2 Configuration from the show running Command

```
service timestamps debug datetime msec

service timestamps log datetime msec

no service password-encryption

!

hostname R2

!

enable password 7 05080F1C2243

!

aaa new-model

aaa authentication ppp default group tacacs+

aaa authorization network default group tacacs+
!
isdn switch-type basic-ni
!
!
!
interface BRI0/0
 no ip address
 encapsulation ppp
 dialer rotary-group 5
 dialer-group 1
 isdn switch-type basic-ni
 isdn spid1 8881 6666
 isdn spid2 8882 6666
 no cdp enable
 ppp authentication chap
!
interface FastEthernet0/1
 ip address 192.168.2.1 255.255.255.0
 duplex auto
 speed auto
 no cdp enable
!
interface Dialer5
 ip address 172.16.35.2 255.255.255.0
 encapsulation ppp
 dialer in-band
 dialer idle-timeout 60
 dialer enable-timeout 5
 dialer hold-queue 20
 dialer aaa
 dialer-group 1
 no peer default ip address
 ppp callback accept
 ppp authentication chap callin
!
ip tacacs source-interface FastEthernet0/1
!
dialer-list 1 protocol ip permit
!
tacacs-server host 192.168.2.10 single-connection key cisco
tacacs-server directed-request
no ip http server
!
line con 0
line aux 0
line vty 0 4
```

Summary

This chapter looked at the many ways you can configure AAA. You looked at the requirements for configuring authentication, authorization, and accounting for your network devices. You completed the chapter by configuring different AAA scenarios in the Practical Exercise. Table 13-6 summarizes the commands you used in this chapter.

Table 13-6. Summary of Commands Used in This Chapter

Command	Description
aaa new-model	Enables AAA on the router.
tacacs-server host*ip-address*single-connection	Indicates the address of the Cisco Secure ACS server and specifies the use of the TCP single-connection feature of the Cisco Secure ACS server. This feature improves performance by maintaining a single TCP connection for the life of the session between the network access server and the Cisco Secure ACS server rather than opening and closing TCP connections for each session, which is the default behavior.
tacacs-server key*key*	Establishes the shared secret encryption key between the network access server and the Cisco Secure ACS server.
radius-server host*ip-address*	Specifies a RADIUS AAA server.
radius-server key*key*	Specifies an encryption key to be used with the RADIUS AAA server.
ip tacacs source-interface*interface-name*	To use the IP address of a specified interface for all outgoing TACACS+ packets, use the ip tacacs source-interface command in global configuration mode.
ip radius source-interface*interface-name*	To force RADIUS to use the IP address of a specified interface for all outgoing RADIUS packets, use the ip radius source-interface global configuration command.
aaa authentication login {default | list-name}*method1* [*method2* [*method3* [*method4*]]]	Sets AAA authentication at login in global configuration mode. If default is configured, when a user logs in, the listed authentication methods that follow this argument as the default list of methods are used. *list-name* is used to name the list of authentication methods activated when a user logs in. The *method*s are enable, line, local, group tacacs+, group radius, none, local-case, andgroup*group-name*.

aaa authentication ppp {default | *list-name*} *method1* [*method2* [*method3* [*method4*]]]	Specifies one or more AAA authentication methods for use on serial interfaces that are running PPP in global configuration mode. If default is configured, when a user logs in, the listed authentication methods that follow this argument as the default list of methods are used. *list-name* is used to name the list of authentication methods activated when a user logs in. The *method*s are if-needed, local, local-case, none, group radius, group tacacs+, andgroup*group-name*.
login authentication {default | *list-name*}	Enables AAA authentication for logins in line configuration mode.
aaa authorization exec {default | *list-name*} *method1* [*method2* [*method3* [*method4*]]]	Used in global configuration mode for the EXEC process and the method of authorization.
aaa authorization network {default | *list-name*}*method1* [*method2* [*method3* [*method4*]]]	Used in global configuration mode for all network services, including SLIP, PPP, and ARAP, and the method of authorization.
authorization exec [default | *list-name*]	Enables AAA authorization for a specific line or group of lines in line configuration mode.
aaa accounting exec {default | *list-name*} {start-stop | stop-only |wait-start | none} [broadcast]group *group-name*	Audits the EXEC process. start-stop sends a start accounting notice at the beginning of a process and a stop accounting notice at the end of a process. The start accounting record is sent in the background. The requested user process begins regardless of whether the start accounting notice was received by the accounting server.stop-only sends a stop accounting notice at the end of the requested user process. wait-start sends both a start and stop accounting notice to the accounting server. With the wait-start keyword, the requested user service does not begin until the start accounting notice is acknowledged. A stop accounting notice is also sent.
aaa accounting commands*level* {default | *list-name*} {start-stop | stop-only |wait-start | none} [broadcast]group *group-name*	Audits all commands at the specified privilege level (0 to 15).
aaa accounting system {default | *list-name*} {start-stop | stop-only |wait-start | none} [broadcast]group *group-name*	Audits all system-level events such as reload.

Review Questions

1: What does AAA stand for?

2: What are the two modes supported by AAA commands except for the aaa accounting system command?

3: Which protocol encrypts the entire body of the packet—RADIUS or TACACS+?

4: Which protocol encrypts only the password in the access request packet from the client to the server—RADIUS or TACACS+?

5: True or false: RADIUS uses UDP, and TACACS+ uses TCP.

6: Which of the following commands are used for packet mode operation?

A. aaa authentication login default group tacacs+

aaa authorization network default group tacacs+

B. aaa authentication login default group tacacs+

aaa authorization exec default group tacacs+

C. aaa authentication ppp default group tacacs+

aaa authorization exec default group tacacs+

D. aaa authentication ppp default group tacacs+

aaa authorization network default group tacacs+

Chapter 14. Securing Remote-Access Networks

This chapter covers the following topics:

- Internet Protocol Security
- Cisco VPN Products
- Virtual Private Networks
- Memory and CPU Considerations
- Monitoring and Maintaining IPSec
- Clearing IKE Connections
- Troubleshooting IKE
- Quality of Service for Virtual Private Networks
- Configuring QoS for VPN Support
- Monitoring and Maintaining QoS for VPNs

Allowing users to access your resources can open your network to a new set of security issues. You should consider allowing access to your network only when you have a valid and working security policy. This chapter aims to give you the information necessary to implement your remote access as securely as required.

Internet Protocol Security

Cisco implements the Internet Protocol Security (IPSec) protocol suite, as detailed in the open standards developed by the Internet Engineering Task Force (IETF), to provide you with security for the transmission of sensitive information over an unprotected network in both Cisco IOS and the PIX's Finesse software.

Cisco's IPSec implementation is based on RFC 2401, *Security Architecture for the Internet Protocol*, Internet Draft, with Cisco IOS IPSec using RFC 1828, *IP Authentication Using Keyed MD5*, and RFC 1829, *TheESPDES-CBCTransform*, for backward compatibility.

IPSec operates at the network layer to provide protection and authentication of IP packets between IPSec peers. IPSec provides you with the following optional network security services. They should be used in accordance with your local security policy:

- Data confidentiality— Lets the IPSec sender encrypt packets before transmitting them across a network.
- Data integrity— Lets the IPSec receiver authenticate packets sent by an IPSec sender to ensure that the data has not been altered during transmission.
- Data origin authentication— Lets the IPSec receiver authenticate the source of the IPSec packets. This service is dependent on the data integrity service.
- Anti-replay— Lets the IPSec receiver detect and reject replayed packets.

IPSec Architecture

IPSec provides you with the framework used to protect one or more data flows between IPSec peers. IPSec consists of the following two main protocols:

- Authentication header (AH)— Provides data authentication and optional anti-replay services by being embedded in the data to be protected, a full IP datagram. The AH security protocol is implemented per the latest version of the IP Authentication Header Internet Draft. It also provides backward compatibility with RFC 1828.
- Encapsulating Security Payload (ESP)— Provides data privacy services, optional data authentication, and anti-replay services by encapsulating the data to be protected. The ESP security protocol is implemented per the latest version of the IP Encapsulating Security Payload Internet Draft. It also provides backward compatibility with RFC 1829.

IPSec implements several standards that are supported by both Cisco IOS and the PIX Firewall:

- IPSec— Provides the framework of open standards to provide data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer. IPSec also uses the Internet Key Exchange (IKE) protocol to handle negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec.
- IKE— A hybrid protocol that implements Oakley and SKEME key exchanges inside an Internet Security Association and Key Management Protocol (ISAKMP) framework. IKE

provides the mechanism to authenticate the IPSec peers, negotiate IPSec security associations (SAs), and establish the IPSec keys.

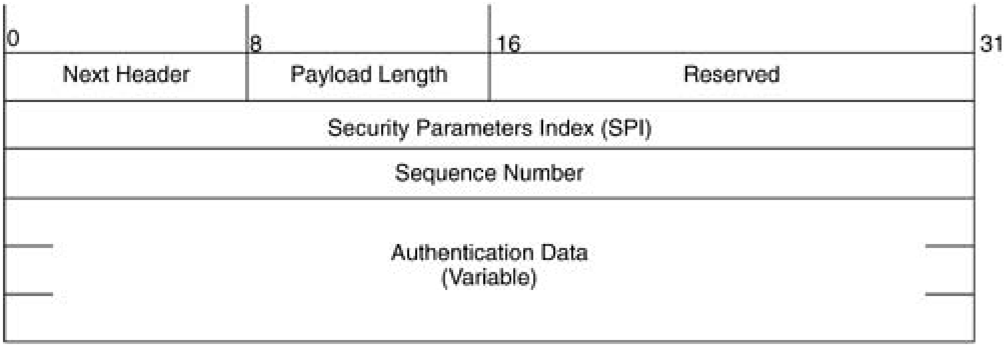
- Message Digest 5 (MD5)— A one-way hashing algorithm that produces a 128-bit hash that, along with the Secure Hash Algorithm, is a variation of MD4, which is designed to strengthen the security of this hashing algorithm.
- Secure Hash Algorithm (SHA)— A one-way hash published by the National Institute of Standards and Technology (NIST). SHA is closely modeled after MD4 and produces a 160-bit digest. Because SHA produces a 160-bit digest, it is more resistant to brute-force attacks than 128-bit hashes, such as MD5, but it can be slower to compute.
- Data Encryption Standard (DES)— An algorithm used to encrypt packet data. IKE implements the 56-bit DES-CBC with Explicit IV standard. Cipher block chaining (CBC) requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPSec packet. Depending on which software version you are using, you also might be able to use Triple DES (168-bit) encryption.
- Diffie-Hellman (D-H)— A public-key cryptography protocol designed to allow two parties to establish a shared secret over an unsecured communications channel. IKE uses D-H to establish session keys.
- RSA signatures— RSA is a public-key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Adleman. RSA signatures provide nonrepudiation.
- Certification authority (CA)— A third-party entity that has the responsibility of issuing and revoking certificates. Each device that has its own certificate and the CA's public key can authenticate every other device within a given CA's domain.

The following sections look at IPSec in more detail.

Authentication Header

The AH, shown in Figure 14-1, provides you with a mechanism to authenticate and verify the integrity of IP datagrams passing between two systems by applying a keyed one-way hash function to the datagram to create a message digest. If the datagram is changed in any way while transiting the network, the receiver detects this when it compares the message digest value it comes up with by performing the same one-way hash function on the datagram sent by the sender. The datagram's authenticity can be guaranteed because the one-way hash mechanism requires the use of a secret shared between the two peers.

Figure 14-1. AH Protocol Packet



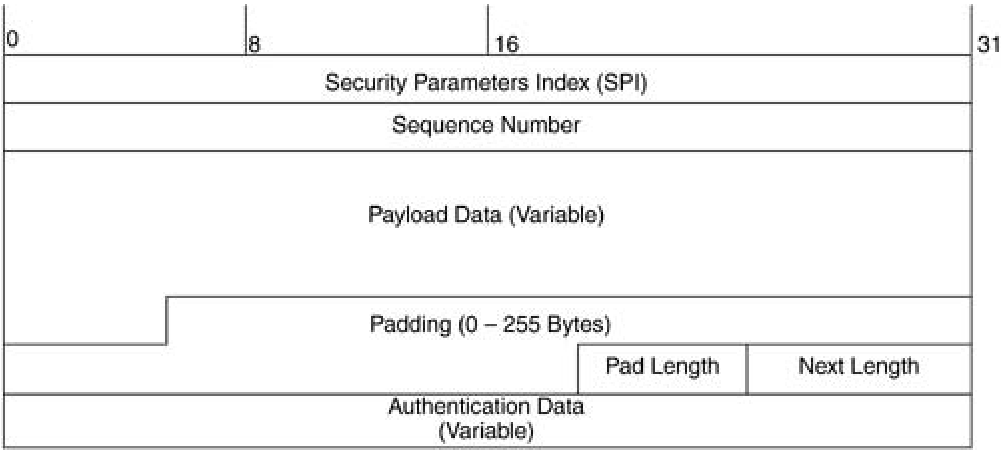
AH can require the receiver to set a bit in the header indicating that a packet has been sent to facilitate anti-replay protection. If the replay bit is not used, an unauthorized user might be able to resend the same packet many times.

AH is applied to the entire datagram, with the exception of any IP header fields that might be changed in normal operation while the datagram is in transition from one peer to the other.

ESP

ESP provides confidentiality (encryption), data origin authentication, integrity, optional anti-replay service, and limited traffic flow confidentiality. Figure 14-2 shows an ESP packet.

Figure 14-2. ESP Protocol Packet



ESP provides confidentiality by performing encryption at the IP packet layer through the use of symmetric encryption algorithms. The default algorithm is 56-bit DES, but support for 168-bit 3DES is also allowed. You may select to use confidentiality independent of any other services.

DES Algorithm

DES is used to encrypt and decrypt select packet data. DES does this by using an encryption algorithm, based on a 56-bit key, to turn clear text into cipher text at the sending peer. DES also turns the cipher-text back into clear text by using a decryption algorithm on the remote peer. Both peers need shared secret keys to enable the encryption and decryption of the packets.

Triple DES Algorithm

3DES is an encryption protocol based on 56-bit DES. 3DES is similar to DES in operation, except that each 64-bit block of data is processed three times, with an independent 56-bit key each time. 3DES essentially doubles the encryption strength offered by 56-bit DES.

Advanced Encryption Standard

The Advanced Encryption Standard (AES) feature, developed by NIST, lets you support the new encryption standard, AES, with CBC mode. AES, developed to replace DES, is a privacy transform for IPSec and IKE that provides a larger key size than DES. AES uses a 128-bit default key, a 192-bit key, or a 256-bit key.

IKE

IKE, often called ISAKMP, is a hybrid protocol designed to provide utility services to IPSec. These services include authentication of the IPSec peers, negotiation of IKE and IPSec security associations, and establishment of keys for encryption algorithms used by IPSec.

IKE key negotiation is done in two phases. Phase 1 is used to negotiate the IKE SA, or key, between two IKE peers. This key lets IKE peers communicate securely during Phase 2. Phase 2 is used to negotiate SAs, or keys, for other applications, such as IPSec.

Phase 1 negotiation occurs using one of two modes: main mode or aggressive mode. Main mode protects all information during the negotiation. During main-mode negotiation, the identities of the two peers are hidden. Main mode has one drawback: It requires time to complete its negotiations. Aggressive mode, on the other hand, requires less time to negotiate keys between peers. Although aggressive mode accomplishes the same end result as main mode, it gives up some of the security provided by main-mode negotiation.

IKE Phase 1 can use three methods to authenticate its IPSec peer:

- Preshared keys— A key value entered into each peer manually (out of band) is used to authenticate the peer.
- RSA signatures— Uses a digital certificate authenticated by an RSA signature.
- RSA encrypted nonces— Uses RSA encryption to encrypt a nonce value (a random number generated by the peer) and other values.

IKE also uses a common value used by all authentication methods—the peer identity (ID). Some examples of the ID are

- The peer's IP address

- The peer's fully qualified domain name (FQDN)

You can create multiple, prioritized policies on each peer to ensure that at least one policy matches the policy of a remote peer.

During the IKE negotiation process, IKE peers agree on the following parameters:

- An encryption algorithm
- A hashing algorithm
- An authentication method
- The lifetime of the SA

Table 14-1 defines each of the five security parameters used to define the IKE policy.

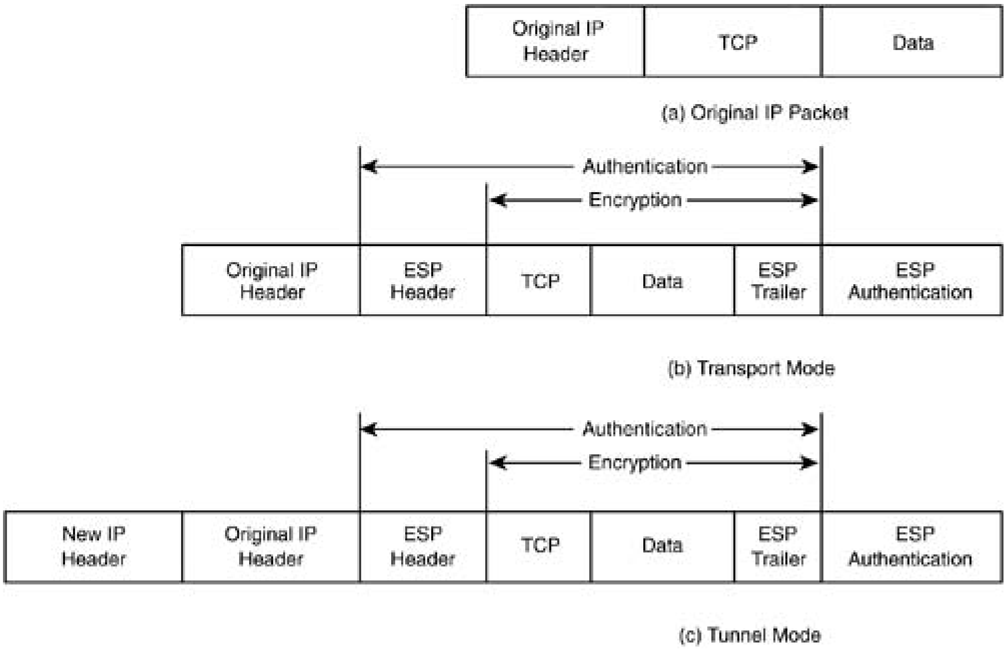
Table 14-1. IKE Security Parameters

Parameter	Accepted Values	Keyword	Default Value
Encryption algorithm	56-bit DES-CBC 168-bit DES	des 3des	56-bit DES-CBC 168-bit DES
Hash algorithm	SHA-1 (HMAC variant) MD5 (HMAC variant)	sha md5	SHA-1
Authentication method	RSA signatures RSA encrypted nonces Preshared keys	rsa-sig rsa-encr pre-share	RSA signatures
D-H group identifier	768-bit D-H 1024-bit D-H 1536-bit D-H	1 2 5	768-bit DH
Lifetime of the security association	Any number of seconds	—	86400 seconds (one day)

Tunnel and Transport Modes

AH and ESP can be run in one of two modes: transport or tunnel. Figure 14-3 shows the packet layout of ESP in the two modes as compared to the packet's original format.

Figure 14-3. ESP Encryption and Authentication



Each of these modes is discussed in the following sections.

Transport Mode

Transport mode protects the upper-layer protocols. When used with IPv4, as shown in Figure 14-3 part (a), the ESP header is inserted into the IP packet before the transport layer header. If authentication is used, an ESP Authentication Data field is added immediately following the ESP trailer. Encryption occurs across the entire transport level segment plus the ESP trailer. Authentication is used to authenticate all cipher text plus the ESP header. This format is shown inFigure 14-3 part (b). When used with IPv6, the payload is the data that normally follows both the IP header and any IPv6 extension headers that are present, with the possible exception of the destination options header, which may be included in the protection.

You typically use transport mode for end-to-end communication between two hosts. ESP in transport mode encrypts and optionally authenticates the IP payload but does not touch the IP header. When AH is used in transport mode, it authenticates the IP payload along with select portions of the IP header. All IPv4 packets contain a Next Header field used to identify the payload protocol. This field is set to decimal 50 for ESP and decimal 51 for AH. AH and ESP headers also contain a Next Header field.

Transport-mode operation eliminates the requirement to implement individual mechanisms in each application to provide confidentiality if the application can take advantage of transport mode. It can also be considered efficient, because it adds very little to the IP packet's total length. Be aware that traffic analysis is still possible with transport mode.

Tunnel Mode

Tunnel mode encapsulates the entire IP packet within a second IP packet, ensuring that the original packet cannot be changed during transport through the network. This essentially "tunnels" the entire original, or inner, packet from one peer to the other without any device in between having the need or capability to view the original packet. For ESP, this concept is illustrated in Figure 14-3 part (c). A new IP header with the necessary routing information is appended to the encapsulated block, containing the ESP header plus cipher text and authentication data if present.

Because the new header does not generally contain the original IP source or destination address, tunnel mode is used when one or both ends of an SA are a security gateway, such as a firewall or router that implements IPSec, making traffic analysis impossible. One added benefit of tunnel mode is that any number of hosts behind the IPSec peer may participate in secure communications without having to be modified to implement IPSec. Their unprotected packets are tunneled through external networks by tunnel-mode SAs set up by the IPSec process on the peers transparently to them.

IPSec Transform Sets

A transform set, a combination of individual IPSec transforms, is used to define a specific security policy for your traffic. It is within this transform set that you establish which security protocols and algorithms you will use for your secure communications. You have the option of defining multiple transform sets. However, during the ISAKMP IPSec security association negotiation that occurs in IKE phase 2 quick mode, the peers must agree on a matching transform set to protect a particular data flow.

Transform sets combine the following IPSec factors:

- Mechanism for payload authentication— AH transform
- Mechanism for payload encryption— ESP transform
- IPSec mode— Transport versus tunnel

Transform sets equal a combination of an AH transform plus an ESP transform plus the IPSec mode (either tunnel or transport mode). Table 14-2 lists the acceptable transform sets you may select from.

Table 14-2. IPSec Transform Sets

AHTransform (Pick up to one)		ESPEncryption Transform (Pick up to one)		ESPAuthentication Transform (Pick up to one only if you also selected the esp-des transform [not esp-rfc1829])	
Transform	Description	Transform	Description	Transform	Description
ah-md5-hmac	AH with the MD5 (HMAC variant) authentication algorithm	esp-des	ESP with the 56-bit DES encryption algorithm	esp-md5-hmac	ESP with the MD5 (HMAC variant) authentication algorithm
ah-sha-hmac	AH with the SHA (HMAC variant) authentication algorithm	esp-3des	ESP with the 168-bit DES encryption algorithm	esp-sha-hmac	ESP with the SHA (HMAC variant) authentication algorithm
ah-rfc1828	Older version of the AH protocol (per RFC 1828)	esp-null	Null encryption algorithm	—	—
—	—	esp-rfc1829	Older version of the ESP protocol (per RFC 1829). Does not allow an accompanying ESP authentication transform.	—	—

Cisco VPN Products

Cisco offers many products that can give you the building blocks you need for your virtual private network (VPN) solutions:

- Cisco PIX 500 series firewall
- Cisco security routers and switches
- Cisco VPN 3000 series concentrators
- Cisco VPN 3000 client

Each of these products is discussed further in the following sections.

Cisco PIX 500 Series Firewall

The Cisco PIX 500 series firewall is a reliable, scalable, functional appliance that provides the following benefits:

- Stateful firewall with per-application content filtering, Java blocking, denial-of-service (DoS) protection, intrusion detection, and time-based ACLs
- Support for L2TP/PPTP-based VPN services suitable for site-to-site VPNs and remote-access VPNs
- Triple DES VPN throughput scalable up to 100 Mbps
- DoS protection against most major types of attacks

Cisco Security Routers and Switches

Cisco has directly integrated security functionality into your network infrastructure through enhanced security features and functionality in Cisco routers and switches, enabling sophisticated security policy enforcement throughout the network. Cisco IOS software's enhanced VPN software features include the following:

- Quality of service (QoS) in the form of application-aware packet classification, congestion management, packet queuing, and traffic shaping and policing
- Stateful IOS firewall with per-application content filtering and Java blocking, DoS protection, intrusion detection, and time-based ACLs
- VPN resiliency through the use of dynamic route recovery using routing protocols through IPSec secured generic routing encapsulation (GRE) tunnel, and dynamic tunnel recovery using IPSec keepalives
- Automated tunnel provisioning using IPSec tunnel endpoint discovery for large mesh network deployments

- Full Layer 3 routing and broad interface support

Cisco VPN 3000 Series Concentrators

The Cisco VPN 3000 series concentrators are remote-access VPN platforms that combine high availability, high performance, and scalability with the most advanced encryption and authentication techniques available. Cisco VPN 3000 series concentrator features include the following:

- High-performance, distributed-processing architecture using Cisco SEP modules to provide hardware-based encryption and large-scale tunneling support for IPSec, PPTP, and L2TP/IPSec connections.
- Scalability with modular design, up to four expansion slots, with redundancy and system architecture designed to provide consistent, high-availability performance. An all-digital design offers high reliability and continuous 24-hour operation with runtime monitoring and alerts.
- Microsoft compatibility offers large-scale client deployment and seamless integration with related systems.
- Security through support of current and emerging security standards allows for integration of external authentication systems and interoperability with third-party products. Firewall capabilities through stateless packet filtering and address translation ensure the required security for a corporate LAN.
- High availability through redundant subsystems and multichassis failover capabilities ensure maximum system uptime.
- Robust management using any standard web browser (HTTP or HTTPS), as well as Telnet, Secure Telnet, SSH, or a console port.

Cisco VPN 3000 Client

The Cisco VPN 3000 client is a software package you use to provide secure connectivity for remote-access VPNs, including support for e-commerce, mobile user, and telecommuting applications. Some of its features include the following:

- Compatibility with most of the major operating systems, including Windows, Linux, Solaris, and Macintosh
- Complete implementation of IPSec standards, including DES and Triple DES encryption
- Authentication through digital certificates, one-time password tokens, and preshared keys

Virtual Private Networks

A VPN can be thought of as a private network you deploy on top of a shared infrastructure that employs the same security, management, and throughput policies you apply to your private network. You currently have three main VPN solutions to choose from:

- Access VPN— Used to provide remote access to an enterprise customer's intranet or extranet over a shared infrastructure. Access VPNs use analog, dial, ISDN, digital subscriber line (DSL), mobile IP, and cable technologies to securely connect mobile users, telecommuters, and branch offices.
- Site-to-site VPN— Used to link enterprise customer headquarters, remote offices, and branch offices to an internal network over a shared infrastructure using dedicated connections. Intranet VPNs differ from extranet VPNs in that they are designed to allow access only to the enterprise customer's employees instead of access to everyone.
- Extranet VPN— Used to link outside customers, suppliers, partners, or communities of interest to an enterprise customer's network over a shared infrastructure using dedicated connections. Extranet VPNs differ from intranet VPNs in that they allow access to users outside the enterprise.

Memory and CPU Considerations

Running IPSec can affect your device's memory usage and CPU utilization. There are several reasons that IPSec packets might be processed slower than packets that are processed through classic crypto:

- IPSec introduces packet expansion, which is more likely to require fragmentation and the corresponding reassembly of IPSec datagrams.
- Encrypted packets probably will be authenticated, which means that two cryptographic operations are performed for every packet.
- The authentication algorithms can be slow.

In addition, the D-H key exchange used in IKE is an exponentiation of very large numbers (between 768 and 1024 bytes) and can take several seconds to compute on some platforms. RSA performance is dependent on the size of the prime number chosen for the RSA key pair.

For each router, the SA database takes approximately 300 bytes of memory, plus an additional 120 bytes of memory for each SA stored in it. Because an IPSec connection requires two SAs, one inbound and one outbound, 540 bytes of memory are required. Each IKE SA entry requires approximately 64 bytes of memory for storage.

There might also be a small decrease in performance for unencrypted packets going through an interface that is doing crypto, because all packets are checked against the crypto map. There should be no performance impact on packets traversing the router that avoid an interface doing crypto.

Monitoring and Maintaining IPSec

Certain configuration changes you make take effect only when you negotiate subsequent SAs. If you want your new settings to take effect immediately, you must clear the existing SAs so that they will be renegotiated with the new configuration. When using manually established SAs, you must clear and reinitialize them, or your changes will never be picked up. If the peer is actively processing IPSec traffic, you can selectively clear only the portion of the SA database that is affected by your configuration changes.

You can use one of the following commands to clear and reinitialize IPSec SAs:

```
R1(config)#clear crypto sa

R1(config)#clear crypto sa peer {ip-address | peer-name}

R1(config)#clear crypto sa mapmap-name

R1(config)#clear crypto sa entrydestination-address protocol spi
```

You can use one or more of the following commands to view information about your IPSec configuration.

This command displays your transform set configuration:

```
R1#show crypto ipsec transform-set
```

This command displays your crypto map configuration:

```
R1#show crypto map [interfaceinterface | tagmap-name]
```

This command displays information about IPSec SAs:

```
R1#show crypto ipsec sa [mapmap-name | address | identity] [detail]
```

This command displays information about dynamic crypto maps:

```
R1#show crypto dynamic-map [tagmap-name]
```

This command displays global SA lifetime values:

```
R1#show crypto ipsec security-association lifetime
```

Clearing IKE Connections

You can use the following commands to clear IKE connections.

To display existing IKE connections, taking note of the connection identifiers for connections you want to clear, use this command:

```
R1#show crypto isakmp sa
```

Use this command to clear an IKE connection:

```
R1#clear crypto isakmp [connection-id]
```

Troubleshooting IKE

You can use the following commands to troubleshoot IKE.

This command displays the parameters for each configured IKE policy:

```
R1#show crypto isakmp policy
```

This command displays all current IKE SAs:

```
R1#show crypto isakmp sa
```

This command displays the crypto map configuration:

```
R1#show crypto map
```

This command verifies an IKE configuration:

```
R1#show running-config
```

This command displays debug messages about IKE events:

```
R1#debug crypto isakmp
```

QoS for Virtual Private Networks

By implementing QoS, you can grant the appropriate service levels to your mission-critical applications. Because remote-access users do not usually care about the network topology or the high level of security/encryption or firewalls that handle their traffic, your solution must be able to give them what they do care about: an acceptable response time for their applications.

Your users' acceptance levels for delays will vary, depending on the application they are using at the time. What is an acceptable level of delay for FTP might not meet with the same acceptance when accessing a database or running voice over IP.

QoS gives you the mechanisms necessary to give your users this level of performance. QoS is a vital tool designed to ensure that all applications coexist and function at acceptable levels of performance. The primary QoS features you will be concerned with, especially when dealing with VPNs, are as follows:

- Packet classification using committed access rate (CAR)
- Bandwidth management by policing with CAR, shaping with Generic Traffic Shaping/Frame Relay Traffic Shaping (GTS/FRTS), and bandwidth allocation with WFQ
- Congestion avoidance using WRED
- Continuity of packet priority over Layer 2 and Layer 3 VPNs with tag switching/Multiprotocol Label Switching (MPLS)

Each of these features is discussed in the following sections.

Packet Classification

The end result of packet classification efforts is to group packets based on your predefined criteria so that the resulting groups of packets can then be subjected to specific packet treatments. This can include faster forwarding by intermediate devices or reducing the probability of a packet's being dropped because of lack of buffering resources. It is often necessary that your traffic be classified before tunneling and encryption, because a tunnel header appended to an IP packet might make the QoS markings in the IP header invisible to intermediate routers/switches.

With classification, you can base decisions on a number of match criteria before your traffic leaves:

- IP addresses
- TCP/UDP port numbers
- IP precedence—the 3 bits in the type of service (ToS) field of the IP packet header
- URL and sub-URL
- MAC addresses
- Time of day

As soon as your packets are classified based on your match criteria, the next step is to mark, or color, the packets with a unique ID to ensure that your classification is honored from end to end. The easiest way to do this is to set the IP ToS field in the header of an IP datagram. This marking of packets is the means you use to ensure that downstream QoS features, such as scheduling and queuing, are used for the proper treatment of the packets you have marked.

Differentiated services let network traffic receive premium treatment at the expense of other less-critical traffic on the same WAN link.

Bandwidth Management

After your selected traffic has been classified, the next step is to ensure that it receives the special treatment it requires from the devices. You do this through the use of queuing and scheduling.

You have the choice of two different implementations of Weighted Fair Queuing (WFQ):

- Flow-based WFQ— Packet classification is based on a flow. Each flow is placed in a separate output queue. When your packet is identified as belonging to a particular flow, it is placed in the associated queue. During times of congestion, WFQ allocates a portion of the available bandwidth for use by each active queue.
- Class-based WFQ— Packets receive the functionality of WFQ with user-defined traffic classes. You create these traffic classes through such mechanisms such as access control lists. After the traffic is classified, you can assign it a fraction of the output interface bandwidth.

Traffic Shaping

Traffic shaping lets you shape Layer 3 traffic into a desired set of rate parameters to enforce a maximum traffic rate. Its end result is a smooth traffic stream at the IP layer through the use of traffic-shaping queues based on the Service Level Agreement (SLA).

Traffic shaping is based on the concept that bursty traffic can be queued, causing the TCP sender to back off its rate of sending, ultimately ensuring that future transmissions conform to your desired rate.

Selecting a Traffic Policer Versus a Traffic Shaper

Policing is used to drop excess traffic, and shaping is used to allow excess traffic to be queued. Shaping can be a better choice where applications are concerned, because shaped traffic does not require a retransmission (dropped traffic does). In this case, Generic Traffic Shaping (GTS) might be the better tool.

Be aware that excessive shaping can result in very deep queues on the shaping device. This might cause the sender to retransmit because of a perceived delay. Policing/dropping of excess traffic is the better choice for IP multicasts or TCP-based traffic related to non-mission-critical applications.

Congestion Avoidance

Congestion avoidance is the ability to recognize and act on congestion in the output direction of an interface in an attempt to reduce or minimize the effects of that congestion.

Congestion produces unwanted effects on a VPN and should be avoided if possible. Tools such as Weighted Random Early Detection (WRED), an implementation of the Random Early Detection (RED) algorithm, let you differentiate between treatment of traffic by adding per-class queue thresholds that determine when packet drops will occur. These thresholds can be configured by the user.

Packet dropping is based on the ideal that adaptive flows such as TCP will back off and retransmit when they detect congestion. By monitoring the average output queue depth and by dropping packets from selected flows, WRED tries to prevent the ramp-up of too many TCP sources at once. Without WRED, TCP synchronization might result.

WRED works by dropping packets from low-priority traffic before it drops packets from high-priority traffic. WRED allows you to select up to six such traffic classes.

QoS for VPN Tunnels

One issue you might face when implementing QoS in a VPN tunnel is the requirement that the QoS parameter you normally find in the header of the IP packet needs to be reflected in the tunnel packet header regardless of the type of tunnel you choose to use. The four primary tunneling protocols used with VPNs are

- Layer 2 Tunneling Protocol (L2TP)
- IPSec
- Layer 2 Forwarding (L2F)
- GRE

L2TP is commonly used for node-to-node applications, with the tunnel terminating at the edge of the user's network. L2TP is based on an IETF-based standard that merges Cisco's L2F tunnel protocol with Microsoft's Point-to-Point Tunneling Protocol (PPTP). L2TP uses third-party security schemes such as IPSec to provide security to packet-level information. L2TP is used primarily with PPP traffic.

GRE tunnels are based on RFC 1702, which allows any protocol to be tunneled inside an IP packet. You can encapsulate data using either IPSec or GRE, both of which can copy the IP ToS values from the packet header into the tunnel header.

This allows devices between GRE-based tunnel endpoints to adhere to the precedence bits you set, improving the routing of premium-service packets. This also gives you the means to use QoS technologies such as policy routing, WFQ, and WRED on intermediate devices between GRE tunnel endpoints.

IETF Differentiated Services

Differentiated Services, or DiffServ (DS), can redefine the IP ToS byte into a DiffServ byte (the

DS byte). The DS byte relays a packet's required QoS level. It is also used to classify packets. DS uses per-hop behaviors (PHBs) to enable common QoS behaviors in the network. The aim is to provide the basis for standards-based QoS in a VPN from end to end.

Committed Access Rate

CAR implements both classification services and policing through rate limiting. You can use CAR's classification services to set the IP precedence for packets entering your network. This allows you to partition your network into multiple priority levels or classes of service. Networking devices within your network can then use the assigned IP precedence values to determine how to treat the traffic. You can use the 3 precedence bits in the ToS field of the IP header to define up to six classes of service.

Your policies can be based on physical port, source or destination IP or MAC address, application port, IP protocol type, or other criteria that can be specified by access lists or extended access lists. You also have the option of classifying packets by categories that are external to the network—for example, by customer. After a packet has been classified, a network can either accept or override and reclassify the packet according to a specified policy. CAR includes commands you can use to classify and reclassify packets.

Custom Queuing

Custom queuing (CQ) is designed to handle traffic by specifying the number of packets or bytes to be serviced for each class of traffic. It services the queues in a round-robin fashion, sending only the allocated portion of bandwidth for each queue before moving to the next queue. If a queue is empty, the device moves to the next queue and sends packets from it, assuming that it has packets ready to send.

When you enable CQ on an interface, the system creates and maintains 17 output queues for that interface. You have the option of configuring queues 1 through 16 by associating a configurable byte count, specifying how many bytes of data to send before moving to the next queue.

Queue 0 is a reserved system queue and is emptied before any of the other queues are processed. The system queue is used for high-priority packets, such as keepalive packets and signaling packets. Other traffic cannot use this queue.

For queues 1 through 16, the system cycles through the queues sequentially, sending the configured byte count from each queue in each cycle, delivering packets in the current queue before moving on to the next one. When a particular queue is being processed, packets are sent until the number of bytes sent exceeds the queue byte count or the queue is empty. You can specify the bandwidth a particular queue can use indirectly by specifying a byte count and queue length. CQ is statically configured and does not automatically adapt to changing network conditions.

The bandwidth that a custom queue is allocated is determined by the following formula:

> (queue byte count / total byte count of all queues) * the interface's bandwidth capacity

where bandwidth capacity equals the interface bandwidth minus the bandwidth for priority queues.

Priority Queuing

Priority queuing (PQ) is used to define how traffic is prioritized in your network. You can configure up to four traffic priorities with a series of filters based on packet characteristics to place traffic in these four queues. The queue with the highest priority is serviced first until it is empty, and then the lower queues are serviced in sequence.

This means that PQ gives priority queues absolute preferential treatment over low-priority queues. Packets are classified based on criteria you specify and are placed in one of the four output queues—high, medium, normal, or low—based on your assigned priority. Packets that you do not classify by priority are placed in the normal queue.

You can set a queue's maximum length by defining the length limit. When a queue is longer than the queue limit, all additional packets are dropped.

A priority list defines a set of rules on how packets are assigned to priority queues. A priority list can also define a default priority or the queue size limits of the various priority queues.

You can classify packets by the following criteria:

- Protocol or subprotocol type
- Incoming interface
- Packet size
- Fragments
- Access list

Keepalive packets sourced by the device are always assigned to the high-priority queue. You must specifically configure all other management traffic into queues. Packets that are not classified by the priority list mechanism are assigned to the normal queue.

Frame Relay Traffic Shaping

Frame Relay Traffic Shaping (FRTS) builds on existing support of congestion control by adding capabilities that improve a Frame Relay network's scalability and performance, increasing the density of VCs and improving response time.

FRTS can be used to eliminate bottlenecks in Frame Relay networks that have high-speed connections at your central site and low-speed connections at your branch sites. You can configure rate enforcement, a peak rate configured to limit outbound traffic, to set a limit on the rate at which data is sent down a VC at your central site.

By using FRTS, you can configure rate enforcement to either the committed information rate (CIR) or some other defined value, such as the excess information rate on a per-VC basis. This ability allows you to share the medium with multiple VCs. Bandwidth can be allocated to each VC, essentially creating a virtual time-division multiplexing (TDM) network.

You also can define PQ, CQ, and WFQ at the VC or subinterface level to achieve finer granularity in the prioritization and queuing of traffic, giving you more control over the traffic flow on an individual VC. If you combine per-VC queuing and rate enforcement with CQ, your VCs can carry multiple traffic types, such as IP, SNA, and Internetwork Packet Exchange (IPX), with a bandwidth guaranteed for each traffic type.

By using backward explicit congestion notification (BECN)-tagged packets, FRTS can dynamically throttle traffic by holding packets in the router's buffers to reduce the data flow from the router into the Frame Relay network. The throttling is done on a per-VC basis. The transmission rate is adjusted based on the number of BECN-tagged packets received.

Configuring QoS for VPN Support

You can configure the QoS for VPNs feature only on tunnel and virtual template interfaces and in crypto map configuration submodes.

When used with GRE and IP-in-IP (IPIP) tunnel protocols, you configure QoS on the tunnel interface, making QoS for VPNs a configuration option on a per-tunnel basis.

When used with the L2F and L2TP protocols, you apply the configuration to the virtual template interface. L2TP clients belonging to identical virtual private dialup network (VPDN) groups inherit the preclassification setting. This feature can be configured on a per-VPDN tunnel basis.

For IPSec tunnels, QoS is applied in the crypto map, allowing configuration on a per-tunnel basis. QoS features on the physical interface on which the crypto map is configured apply classification to the packets before encryption is applied.

You can use the following commands to configure the QoS for VPNs feature on a tunnel or virtual interface basis.

To enter interface configuration mode and specify the tunnel or virtual interface to configure, use this command:

```
R1(config)#interface [tunnel-name | virtual-template-name]
```

Use this command to enable the QoS for VPNs feature:

```
R1(config-if)#qos pre-classify
```

You can use the following commands to configure the QoS for VPNs feature on a crypto map configuration basis.

To enter crypto map configuration mode and specify a previously defined crypto map to configure, use this command:

```
R1(config)#crypto map [map-name]
```

This command enables the QoS for VPNs feature:

```
R1(config-if)#qos pre-classify
```

Monitoring and Maintaining QoS for VPNs

You can use the following commands to monitor and maintain the QoS for VPNs feature.

To display information on the tunnel or the virtual template, including the queuing strategy, use this command:

```
R1#show interfaces [tunnel-name | virtual-template-name]
```

This command displays information on the crypto map:

```
R1#show crypto map [map-name]
```

Scenarios

The scenarios presented in this chapter help you gain a more complete understanding of configuring IPSec through practical application. You will go through the necessary configuration tasks in their logical progression. The scenarios cover the following topics:

- Defining IKE parameters
- Defining IPSec transform sets

Scenario 14-1: Defining IKE Parameters

In this scenario, you define two IKE proposals. The first uses DES, MD5, preshared keys, D-H group 1, and a lifetime of 600 seconds. The second proposal uses 3DES, SHA, RSA signatures, D-H group 2, and a lifetime of 1 day.

Step 1. Identify the policy to create:

```
R1(config)#crypto isakmp policypriority
```

Step 2. Specify the encryption algorithm to use:

```
R1(config-isakmp)#encryption 3des
```

Step 3. Specify the hash algorithm to use:

```
R1(config-isakmp)#hash {sha | md5}
```

Step 4. Specify the authentication method:

```
R1(config-isakmp)#authentication {rsa-sig | rsa-encr | pre-share}
```

Step 5. Specify the D-H group to use:

```
R1(config-isakmp)#group {1 | 2}
```

Step 6. Specify the lifetime, in seconds, for the security association:

```
R1(config-isakmp)#lifetimeseconds
```

Example 14-1 shows the commands you can use to complete this scenario.

Example 14-1. Defining IKE Proposals

```
R1(config)#crypto isakmp policy 10

R1(config-isakmp)#hash md5

R1(config-isakmp)#authentication pre-share

R1(config-isakmp)#group 1

R1(config-isakmp)#lifetime 600

R1(config-isakmp)#exit

R1(config)#crypto isakmp policy 20

R1(config-isakmp)#encryption 3des

R1(config-isakmp)#hash sha

R1(config-isakmp)#authentication rsa-sig

R1(config-isakmp)#group 2

R1(config-isakmp)#lifetime 86400
```

Scenario 14-2: Defining IPSec Transform Sets

In this scenario, you define two transform sets. The first, named set1, uses the authentication SHA HMAC variant using transport mode. The second, named set2, uses the 3DES encryption algorithm with the SHA authentication algorithm using tunnel mode.

Follow these steps to define your transform set:

Step 1. Define your transform set:

```
R1(config)#crypto ipsec transform-set transform-set-name transform1

  [transform2 [transform3]]
```

Step 2. Optionally define the mode to use with the transform set:

```
R1(cfg-crypto-tran)#mode [tunnel | transport]
```

Example 14-2 shows the commands necessary on R1 to complete this scenario.

Example 14-2. Defining IPSec Transform Sets

```
R1(config)#crypto ipsec transform-set set1 ah-sha-hmac
R1(cfg-crypto-tran)#mode transport
R1(cfg-crypto-tran)#exit
R1(config)#crypto ipsec transform-set set2 esp-3des esp-sha-hmac
R1(cfg-crypto-tran)#mode tunnel
```

Practical Exercise 14-1: IPSec Router-to-Router

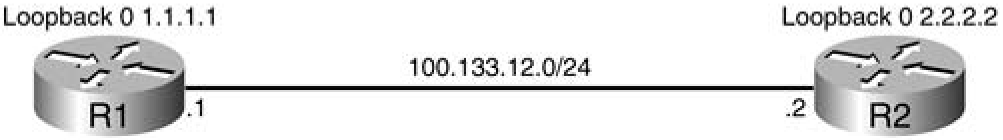
Complete the tasks outlined in this Practical Exercise. Also review the Practical Exercise solution to see how you did and to see what concepts you might need to review.

In this Practical Exercise, you will configure your R1 router to initiate an IPSec router-to-router connection to R2. IKE will use an MD5 hash along with preshared keys. R1 will always initiate the tunnel between the two routers and will be configured to initiate in aggressive mode. R2 will use a dynamic crypto map to accept the tunnel parameters from R1, although it could also have a standard LAN-to-LAN tunnel configuration applied.

Background Information

You are the administrator of R1. You need to configure a LAN-to-LAN connection to R2, as shown inFigure 14-4.

Figure 14-4. IPSec Router-to-Router Topology



Task 1: Verify Compatibility with Existing Access Lists

To run IKE and IPSec, you need to ensure that any existing access lists are compatible with both protocols. Any existing access lists must allow the ports required by IKE and IPSec to pass through them.

Task 2: Define IKE Parameters

Step 1. At the R1 console, provide all the configuration required to set the following IKE settings:

- Define an ISAKMP policy.
- Set the ISAKMP keepalive interval.
- Define the ISAKMP peer and aggressive mode.

Step 2. At the R2 console, provide all the configuration required to set the following IKE settings:

- Define an ISAKMP policy.

- Set the ISAKMP keepalive interval.

- Define the ISAKMP peer and key.

Task 3: Define IPSec Parameters

Step 1. At the R1 console, provide all the configuration required to set the following IPSec settings:

- Define a route to the peer network.

- Define a crypto access list.

- Define an IPSec transform set.

- Define the IPSec crypto map.

- Associate the crypto map to the Ethernet 0 interface.

Step 2. At the R2 console, provide all the configuration required to set the following IPSec settings:

- Define a route to the peer network.

- Define an IPSec transform set.

- Define a dynamic IPSec crypto map.

- Define the IPSec crypto map.

- Associate the crypto map to the Ethernet 0 interface.

Practical Exercise 14-1 Solution

The following is a step-by-step discussion of the Practical Exercise solution.

Task 1 Solution

IKE uses UDP port 500. The IPSec ESP and AH protocols use protocol numbers 50 and 51. You must ensure that any existing access lists you might already have configured do not block protocol 50, 51, and UDP port 500 traffic at any interface used by IPSec. In some cases you might need to reconfigure an existing access list to explicitly permit this traffic.

Task 2 Solution

Step 1. At the R1 console, provide all the configuration required to set the following IKE settings:

Define an ISAKMP policy:

```
R1(config)#crypto isakmp policy 1

R1(config-isakmp)#hash md5

R1(config-isakmp)#authentication pre-share
```

Set the ISAKMP keepalive interval:

```
R1(config)#crypto isakmp keepalive 30 5
```

Define the ISAKMP peer and aggressive mode:

```
R1(config)#crypto isakmp peer address 100.133.12.2

R1(config-isakmp)#set aggressive-mode password cisco123

R1(config-isakmp)#set aggressive-mode client-endpoint ipv4-address

  100.133.12.1
```

Step 2. At the R2 console, provide all the configuration required to set the following IKE settings:

Define an ISAKMP policy:

```
R2(config)#crypto isakmp policy 1

R2(config-isakmp)#hash md5

R2(config-isakmp)#authentication pre-share
```

Set the ISAKMP keepalive interval:

```
R2(config)#crypto isakmp keepalive 30 5
```

Define the ISAKMP peer and key:

```
R2(config)#crypto isakmp key cisco123 address 100.133.12.1
```

Task 3 Solution

Step 1. At the R1 console, provide all the configuration required to set the following IPSec settings:

Define a route to the peer network:

```
R1(config)#ip route 0.0.0.0 0.0.0.0 100.133.12.2
```

Define a crypto access list:

```
R1(config)#access-list 100 permit ip 1.1.1.0 0.0.0.255 2.2.2.0 0.0.0.255
```

Define an IPSec transform set:

```
R1(config)#crypto ipsec transform-set myset esp-3des esp-md5-hmac
```

Define the IPSec crypto map:

```
R1(config)#crypto map mymap 1 ipsec-isakmp

R1(config-crypto-m)#set peer 100.133.12.2

R1(config-crypto-m)#set transform-set myset

R1(config-crypto-m)#match address 100
```

Associate the crypto map to the Ethernet 0 interface:

```
R1(config)#interface ethernet 0

R1(config-if)#crypto map mymap
```

Step 2. At the R2 console, provide all the configuration required to set the following IPSec settings:

Define a route to the peer network:

```
R1(config)#ip route 0.0.0.0 0.0.0.0 100.133.12.1
```

Define an IPSec transform set:

```
R2(config)#crypto ipsec transform-set myset esp-3des esp-md5-hmac
```

Define a dynamic IPSec crypto map:

```
R2(config)#crypto dynamic-map mymap 10
R2(config-crypto-m)#set transform-set myset
```

Define the IPSec crypto map:

```
R2(config)#crypto map mainmap 1 ipsec-isakmp dynamic mymap
```

Associate the crypto map to the Ethernet 0 interface:

```
R2(config)#interface ethernet 0
R2(config-if)#crypto map mainmap
```

Practical Exercise 14-2: Three Full-Mesh IPSec Routers

Complete the tasks outlined in this Practical Exercise. Also review the Practical Exercise solution to see how you did and to see what concepts you might need to review.

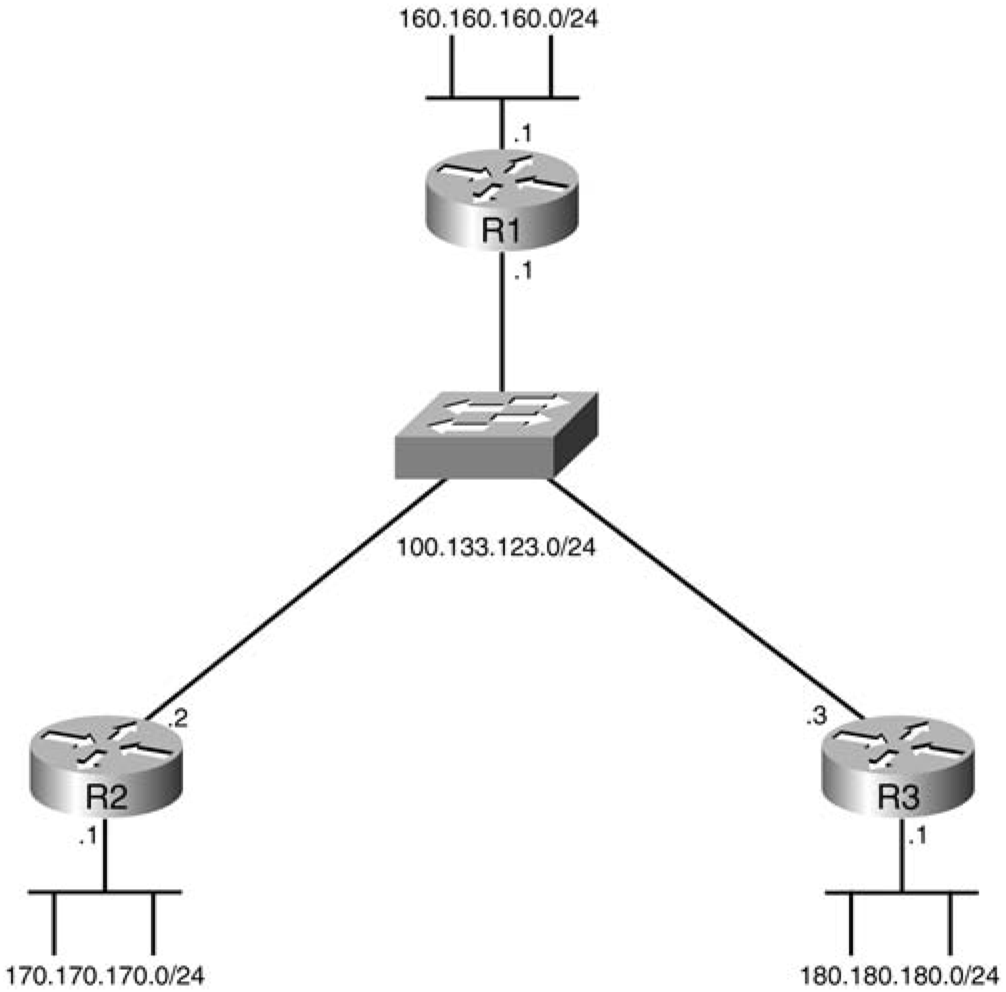
In this Practical Exercise, you are the administrator of a set of routers—R1, R2, and R3—and you are required to configure an IPSec VPN between them. The VPNs are required to provide redundancy between the sites in case of a line failure. You are required to have connectivity between the networks behind each of a router's two peers. Encryption is to be done as follows:

- From network 160.160.160.*x* to network 170.170.170.*x*
- From network 160.160.160.*x* to network 180.180.180.*x*
- From network 170.170.170.*x* to network 180.180.180.*x*

Background Information

You will configure a VPN between three routers, as illustrated in Figure 14-5.

Figure 14-5. Three Full-Mesh IPSec Routers Topology



Task 1: Verify Compatibility with Existing Access Lists

To run IKE and IPSec, you need to ensure that any existing access lists are compatible with both protocols. Any existing access lists must allow the ports required by IKE and IPSec to pass through them.

Task 2: Define IKE Parameters

Step 1. At the R1 console, provide all the configuration required to set the following IKE settings:

- Define an ISAKMP policy.
- Define the ISAKMP peer and key.

Step 2. At the R2 console, provide all the configuration required to set the following IKE settings:

- Define an ISAKMP policy.

- Define the ISAKMP peer and key.

Step 3. At the R3 console, provide all the configuration required to set the following IKE settings:

- Define an ISAKMP policy.
- Define the ISAKMP peer and key.

Task 3: Define IPSec Parameters

Step 1. At the R1 console, provide all the configuration required to set the following IPSec settings:

- Define a route to the peer network.
- Define a crypto access list.
- Define an IPSec transform set.
- Define the IPSec crypto map.
- Associate the crypto map to the Ethernet 0 interface.

Step 2. At the R2 console, provide all the configuration required to set the following IPSec settings:

- Define a route to the peer network.
- Define a crypto access list.
- Define an IPSec transform set.
- Define the IPSec crypto map.
- Associate the crypto map to the Ethernet 0 interface.

Step 3. At the R3 console, provide all the configuration required to set the following IPSec settings:

- Define a route to the peer network.
- Define a crypto access list.
- Define an IPSec transform set.
- Define the IPSec crypto map.
- Associate the crypto map to the Ethernet 0 interface.

Practical Exercise 14-2 Solution

The following is a step-by-step discussion of the Practical Exercise solution.

Task 1 Solution

IKE uses UDP port 500. The IPSec ESP and AH protocols use protocol numbers 50 and 51. You must ensure that any existing access lists you might already have configured do not block protocol 50, 51, and UDP port 500 traffic at any interface used by IPSec. In some cases you might need to reconfigure an existing access list to explicitly permit this traffic.

Task 2 Solution

Step 1. At the R1 console, provide all the configuration required to set the following IKE settings:

Define an ISAKMP policy:

```
R1(config)#crypto isakmp policy 1

R1(config-isakmp)#authentication pre-share
```

Define the ISAKMP peer and key:

```
R1(config)#crypto isakmp key cisco123 address 100.133.123.2

R1(config)#crypto isakmp key cisco123 address 100.133.123.3
```

Step 2. At the R2 console, provide all the configuration required to set the following IKE settings:

Define an ISAKMP policy:

```
R2(config)#crypto isakmp policy 1

R2(config-isakmp)#authentication pre-share
```

Define the ISAKMP peer and key:

```
R2(config)#crypto isakmp key cisco123 address 100.133.123.1

R2(config)#crypto isakmp key cisco123 address 100.133.123.3
```

Step 3. At the R3 console, provide all the configuration required to set the following IKE settings:

Define an ISAKMP policy:

```
R3(config)#crypto isakmp policy 1

R3(config-isakmp)#authentication pre-share
```

Define the ISAKMP peer and key:

```
R3(config)#crypto isakmp key cisco123 address 100.133.123.1

R3(config)#crypto isakmp key cisco123 address 100.133.123.2
```

Task 3 Solution

Step 1. At the R1 console, provide all the configuration required to set the following IPSec settings:

Define a route to the peer network:

```
R1(config)#ip route 170.170.170.0 255.255.255.0 100.133.123.2

R1(config)#ip route 180.180.180.0 255.255.255.0 100.133.123.3
```

Define a crypto access list:

```
R1(config)#access-list 170 permit ip 160.160.160.0 0.0.0.255 170.170.170.0

  0.0.0.255

R1(config)#access-list 180 permit ip 160.160.160.0 0.0.0.255 180.180.180.0

  0.0.0.255
```

Define an IPSec transform set:

```
R1(config)#crypto ipsec transform-set 170cisco esp-des esp-md5-hmac
R1(cfg-crypto-trans)#exit
R1(config)#crypto ipsec transform-set 180cisco esp-des esp-md5-hmac
```

Define the IPSec crypto map:

```
R1(config)#crypto map mymap 17 ipsec-isakmp
R1(config-crypto-m)#set peer 100.133.123.2
R1(config-crypto-m)#set transform-set 170cisco
R1(config-crypto-m)#match address 170
R1(config-crypto-m)#exit
R1(config)#crypto map mymap 18 ipsec-isakmp
R1(config-crypto-m)#set peer 100.133.123.3
R1(config-crypto-m)#set transform-set 180cisco
R1(config-crypto-m)#match address 180
```

Associate the crypto map to the Ethernet 0 interface:

```
R1(config)#interface ethernet 0

R1(config-if)#crypto map mymap
```

Step 2. At the R2 console, provide all the configuration required to set the following IPSec settings:

Define a route to the peer network:

```
R2(config)#ip route 160.160.160.0 255.255.255.0 100.133.123.1

R2(config)#ip route 180.180.180.0 255.255.255.0 100.133.123.3
```

Define a crypto access list:

```
R2(config)#access-list 160 permit ip 170.170.170.0 0.0.0.255 160.160.160.0

  0.0.0.255

R2(config)#access-list 180 permit ip 170.170.170.0 0.0.0.255 180.180.180.0

  0.0.0.255
```

Define an IPSec transform set:

```
R2(config)#crypto ipsec transform-set 160cisco esp-des esp-md5-hmac
```

```
R2(cfg-crypto-trans)#exit
R2(config)#crypto ipsec transform-set 180cisco esp-des esp-md5-hmac
```

Define the IPSec crypto map:

```
R2(config)#crypto map mymap 16 ipsec-isakmp
R2(config-crypto-m)#set peer 100.133.123.1
R2(config-crypto-m)#set transform-set 160cisco
R2(config-crypto-m)#match address 160
R2(config-crypto-m)#exit
R2(config)#crypto map mymap 18 ipsec-isakmp
R2(config-crypto-m)#set peer 100.133.123.3
R2(config-crypto-m)#set transform-set 180cisco
R2(config-crypto-m)#match address 180
```

Associate the crypto map to the Ethernet 0 interface:

```
R2(config)#interface ethernet 0
R2(config-if)#crypto map mymap
```

Step 3. At the R3 console, provide all the configuration required to set the following IPSec settings:

Define a route to the peer network:

```
R3(config)#ip route 160.160.160.0 255.255.255.0 100.133.123.1
R3(config)#ip route 170.170.170.0 255.255.255.0 100.133.123.2
```

Define a crypto access list:

```
R3(config)#access-list 160 permit ip 180.180.180.0 0.0.0.255 160.160.160.0
  0.0.0.255
R3(config)#access-list 170 permit ip 180.180.180.0 0.0.0.255 170.170.170.0
  0.0.0.255
```

Define an IPSec transform set:

```
R3(config)#crypto ipsec transform-set 160cisco esp-des esp-md5-hmac
R3(cfg-crypto-trans)#exit
R3(config)#crypto ipsec transform-set 170cisco esp-des esp-md5-hmac
```

Define the IPSec crypto map:

```
R3(config)#crypto map mymap 16 ipsec-isakmp
R3(config-crypto-m)#set peer 100.133.123.1
R3(config-crypto-m)#set transform-set 160cisco
R3(config-crypto-m)#match address 160
R3(config-crypto-m)#exit
R3(config)#crypto map mymap 17 ipsec-isakmp
R3(config-crypto-m)#set peer 100.133.123.2
R3(config-crypto-m)#set transform-set 170cisco
R3(config-crypto-m)#match address 170
```

Associate the crypto map to the Ethernet 0 interface:

```
R3(config)#interface ethernet 0
R3(config-if)#crypto map mymap
```

Practical Exercise 14-3: IPSec Router-to-Router Hub and Spoke

Complete the tasks outlined in this Practical Exercise. Also review the Practical Exercise solution to see how you did and to see what concepts you might need to review.

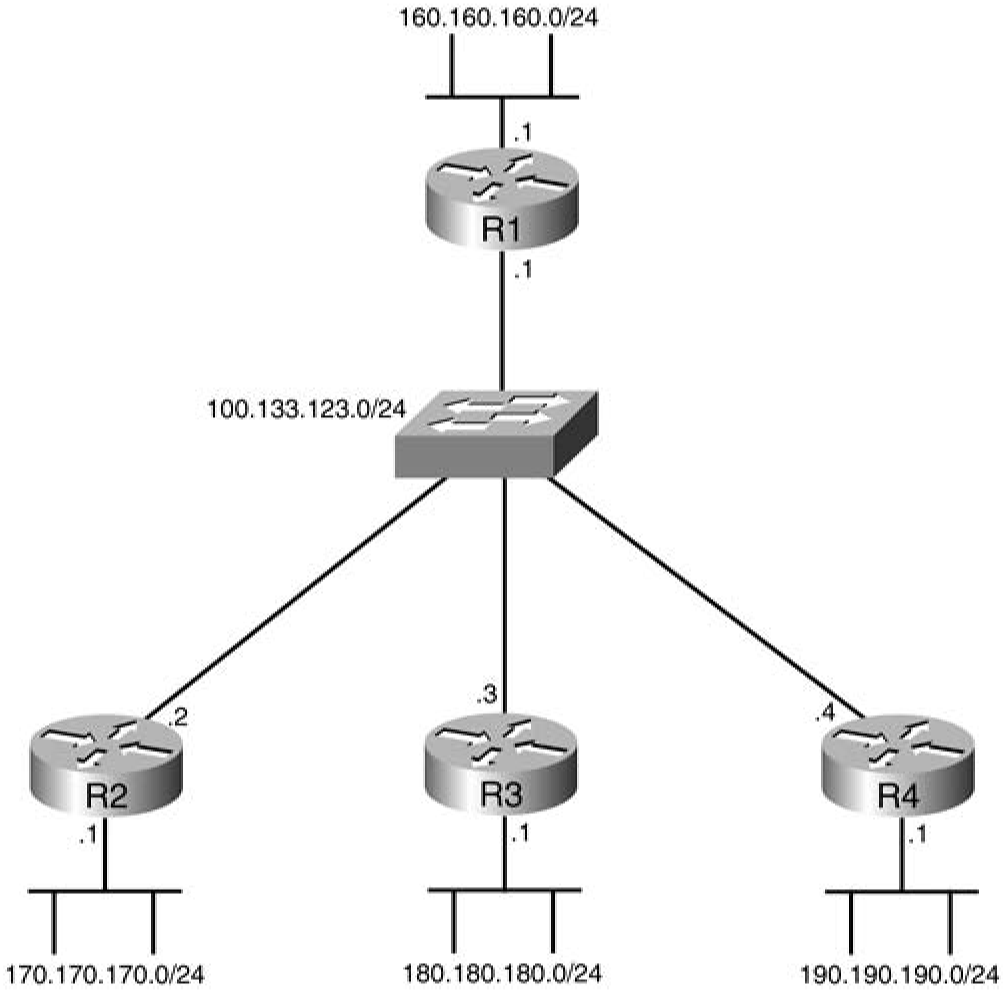
In this Practical Exercise, you are the administrator of a set of routers—R1, R2, R3, and R4. You are required to configure an IPSec VPN between them. R1 is your hub router, and the remaining routers form spokes around it. You will define a single crypto map on the hub router, specifying the networks behind each of its three peers. The crypto maps on each of the spoke routers specify the network behind the hub router. Encryption will be done between the following networks:

- From network 160.160.160.*x* to network 170.170.170.*x*
- From network 160.160.160.*x* to network 180.180.180.*x*
- From network 160.160.160.*x* to network 190.190.190.*x*

Background Information

You will configure a VPN between a hub-and-spoke router configuration, as illustrated in Figure 14-6.

Figure 14-6. IPSec Router-to-Router Hub-and-Spoke Topology



Task 1: Verify Compatibility with Existing Access Lists

To run IKE and IPSec, you need to ensure that any existing access lists are compatible with both protocols. Any existing access lists must allow the ports required by IKE and IPSec to pass through them.

Task 2: Define IKE Parameters

Step 1. At the R1 console, provide all the configuration required to set the following IKE settings:

- Define an ISAKMP policy.

- Define the ISAKMP peer and key.

Step 2. At the R2 console, provide all the configuration required to set the following IKE settings:

- Define an ISAKMP policy.

- Define the ISAKMP peer and key.

Step 3. At the R3 console, provide all the configuration required to set the following IKE settings:

- Define an ISAKMP policy.
- Define the ISAKMP peer and key.

Step 4. At the R4 console, provide all the configuration required to set the following IKE settings:

- Define an ISAKMP policy.
- Define the ISAKMP peer and key.

Task 3: Define IPSec Parameters

Step 1. At the R1 console, provide all the configuration required to set the following IPSec settings:

- Define a route to the peer network.
- Define a crypto access list.
- Define an IPSec transform set.
- Define the IPSec crypto map.
- Associate the crypto map to the Ethernet 0 interface.

Step 2. At the R2 console, provide all the configuration required to set the following IPSec settings:

- Define a route to the peer network.
- Define a crypto access list.
- Define an IPSec transform set.
- Define the IPSec crypto map.
- Associate the crypto map to the Ethernet 0 interface.

Step 3. At the R3 console, provide all the configuration required to set the following IPSec settings:

- Define a route to the peer network.
- Define a crypto access list.
- Define an IPSec transform set.
- Define the IPSec crypto map.
- Associate the crypto map to the Ethernet 0 interface.

Step 4. At the R4 console, provide all the configuration required to set the following IPSec settings:

- Define a route to the peer network.
- Define a crypto access list.
- Define an IPSec transform set.
- Define the IPSec crypto map.
- Associate the crypto map to the Ethernet 0 interface.

Practical Exercise 14-3 Solution

The following is a step-by-step discussion of the Practical Exercise solution.

Task 1 Solution

IKE uses UDP port 500. The IPSec ESP and AH protocols use protocol numbers 50 and 51. You must ensure that any existing access lists you might already have configured do not block protocol 50, 51, and UDP port 500 traffic at any interface used by IPSec. In some cases you might need to reconfigure an existing access list to explicitly permit this traffic.

Task 2 Solution

Step 1. At the R1 console, provide all the configuration required to set the following IKE settings:

Define an ISAKMP policy:

```
R1(config)#crypto isakmp policy 1

R1(config-isakmp)#authentication pre-share
```

Define the ISAKMP peer and key:

```
R1(config)#crypto isakmp key cisco170 address 100.133.123.2

R1(config)#crypto isakmp key cisco180 address 100.133.123.3

R1(config)#crypto isakmp key cisco190 address 100.133.123.4
```

Step 2. At the R2 console, provide all the configuration required to set the following IKE settings:

Define an ISAKMP policy:

```
R2(config)#crypto isakmp policy 1

R2(config-isakmp)#authentication pre-share
```

Define the ISAKMP peer and key:

```
R2(config)#crypto isakmp key cisco170 address 100.133.123.1
```

Step 3. At the R3 console, provide all the configuration required to set the following IKE settings:

Define an ISAKMP policy:

```
R3(config)#crypto isakmp policy 1

R3(config-isakmp)#authentication pre-share
```

Define the ISAKMP peer and key:

```
R3(config)#crypto isakmp key cisco180 address 100.133.123.1
```

Step 4. At the R4 console, provide all the configuration required to set the following IKE settings:

Define an ISAKMP policy:

```
R4(config)#crypto isakmp policy 1

R4(config-isakmp)#authentication pre-share
```

Define the ISAKMP peer and key:

```
R4(config)#crypto isakmp key cisco190 address 100.133.123.1
```

Task 3 Solution

Step 1. At the R1 console, provide all the configuration required to set the following IPSec settings:

Define a route to the peer network:

```
R1(config)#ip route 170.170.170.0 255.255.255.0 100.133.123.2
R1(config)#ip route 180.180.180.0 255.255.255.0 100.133.123.3
R1(config)#ip route 190.190.190.0 255.255.255.0 100.133.123.4
```

Define a crypto access list:

```
R1(config)#access-list 170 permit ip 160.160.160.0 0.0.0.255 170.170.170.0
  0.0.0.255
R1(config)#access-list 180 permit ip 160.160.160.0 0.0.0.255 180.180.180.0
  0.0.0.255
R1(config)#access-list 180 permit ip 160.160.160.0 0.0.0.255 190.190.190.0
  0.0.0.255
```

Define an IPSec transform set:

```
R1(config)#crypto ipsec transform-set 170cisco esp-des esp-md5-hmac
R1(cfg-crypto-trans)#exit
R1(config)#crypto ipsec transform-set 180cisco esp-des esp-md5-hmac
R1(cfg-crypto-trans)#exit
R1(config)#crypto ipsec transform-set 190cisco esp-des esp-md5-hmac
```

Define the IPSec crypto map:

```
R1(config)#crypto map mymap 17 ipsec-isakmp
R1(config-crypto-m)#set peer 100.133.123.2
R1(config-crypto-m)#set transform-set 170cisco
R1(config-crypto-m)#match address 170
R1(config-crypto-m)#exit
R1(config)#crypto map mymap 18 ipsec-isakmp
R1(config-crypto-m)#set peer 100.133.123.3
R1(config-crypto-m)#set transform-set 180cisco
R1(config-crypto-m)#match address 180
R1(config)#crypto map mymap 19 ipsec-isakmp
R1(config-crypto-m)#set peer 100.133.123.4
R1(config-crypto-m)#set transform-set 190cisco
R1(config-crypto-m)#match address 190
```

Associate the crypto map to the Ethernet 0 interface:

```
R1(config)#interface ethernet 0
R1(config-if)#crypto map mymap
```

Step 2. At the R2 console, provide all the configuration required to set the following IPSec settings:

Define a route to the peer network:

```
R2(config)#ip route 160.160.160.0 255.255.255.0 100.133.123.1
```

Define a crypto access list:

```
R2(config)#access-list 170 permit ip 170.170.170.0 0.0.0.255 160.160.160.0
  0.0.0.255
```

Define an IPSec transform set:

```
R2(config)#crypto ipsec transform-set 170cisco esp-des esp-md5-hmac
```

Define the IPSec crypto map:

```
R2(config)#crypto map mymap 17 ipsec-isakmp

R2(config-crypto-m)#set peer 100.133.123.1

R2(config-crypto-m)#set transform-set 170cisco

R2(config-crypto-m)#match address 170
```

Associate the crypto map to the Ethernet 0 interface:

```
R2(config)#interface ethernet 0

R2(config-if)#crypto map mymap
```

Step 3. At the R3 console, provide all the configuration required to set the following IPSec settings:

Define a route to the peer network:

```
R3(config)#ip route 160.160.160.0 255.255.255.0 100.133.123.1
```

Define a crypto access list:

```
R3(config)#access-list 180 permit ip 180.180.180.0 0.0.0.255 160.160.160.0

  0.0.0.255
```

Define an IPSec transform set:

```
R3(config)#crypto ipsec transform-set 180cisco esp-des esp-md5-hmac
```

Define the IPSec crypto map:

```
R3(config)#crypto map mymap 18 ipsec-isakmp
R3(config-crypto-m)#set peer 100.133.123.1
R3(config-crypto-m)#set transform-set 180cisco
R3(config-crypto-m)#match address 180
```

Associate the crypto map to the Ethernet 0 interface:

```
R3(config)#interface ethernet 0
R3(config-if)#crypto map mymap
```

Step 4. At the R4 console, provide all the configuration required to set the following IPSec settings:

Define a route to the peer network:

```
R3(config)#ip route 160.160.160.0 255.255.255.0 100.133.123.1
```

Define a crypto access list:

```
R3(config)#access-list 190 permit ip 190.190.190.0 0.0.0.255 160.160.160.0
  0.0.0.255
```

Define an IPSec transform set:

```
R3(config)#crypto ipsec transform-set 190cisco esp-des esp-md5-hmac
```

Define the IPSec crypto map:

```
R3(config)#crypto map mymap 19 ipsec-isakmp
R3(config-crypto-m)#set peer 100.133.123.1
```

```
R3(config-crypto-m)#set transform-set 190cisco

R3(config-crypto-m)#match address 190
```

Associate the crypto map to the Ethernet 0 interface.

```
R3(config)#interface ethernet 0

R3(config-if)#crypto map mymap
```

Practical Exercise 14-4: IPSec Between Three Routers Using Private Addresses

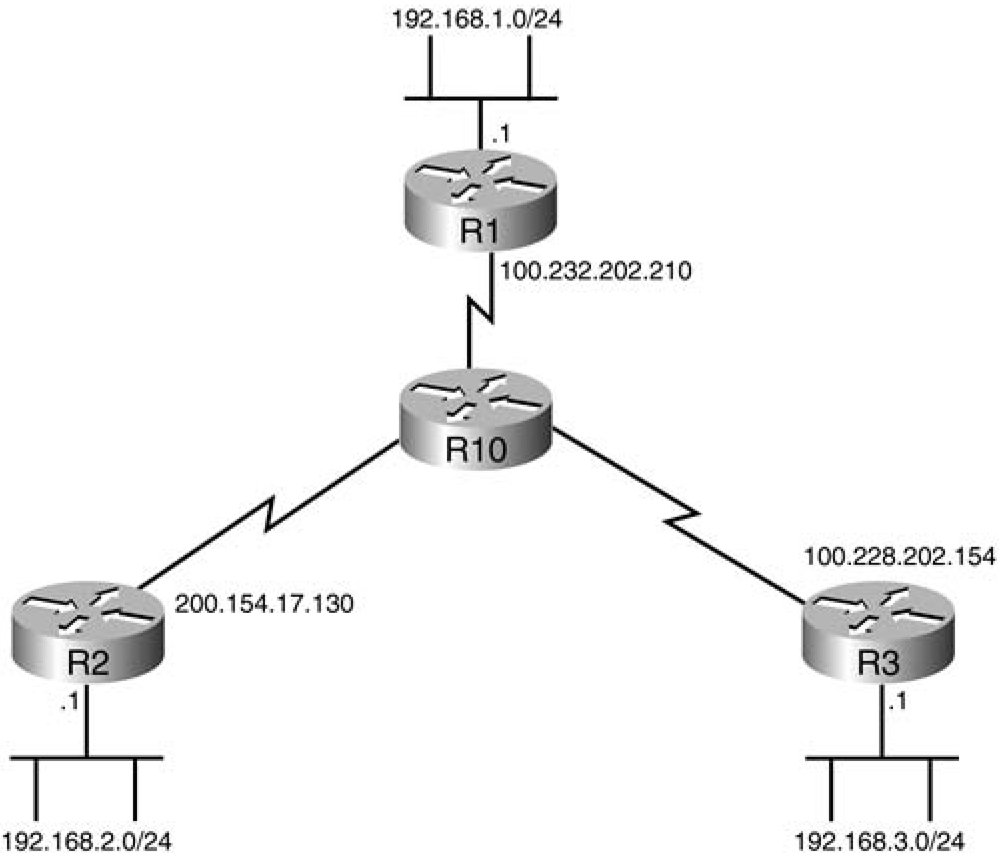
Complete the tasks outlined in this Practical Exercise. Also review the Practical Exercise solution to see how you did and to see what concepts you might need to review.

In this Practical Exercise, you are the administrator of a set of routers—R1, R2, and R3—and you are required to configure an IPSec VPN between them. You will configure your routers so that they form a full mesh with connectivity to the private networks behind each peer router.

Background Information

You will configure a VPN between three routers with private networks, as illustrated in Figure 14-7.

Figure 14-7. IPSec Between Three Routers Using Private Addresses



Task 1: Verify Compatibility with Existing Access Lists

To run IKE and IPSec, you need to ensure that any existing access lists are compatible with both protocols. Any existing access lists must allow the ports required by IKE and IPSec to pass through them.

Task 2: Create Network Address Translation

Step 1. At the R1 console, provide all the configuration required to set the following IKE settings:

- Define traffic to undergo NAT.
- Define an access list for NAT.
- Define the NAT route map.
- Define the NAT interfaces.

Step 2. At the R2 console, provide all the configuration required to set the following IKE settings:

- Define traffic to undergo NAT.
- Define an access list for NAT.
- Define the NAT route map.
- Define the NAT interfaces.

Step 3. At the R3 console, provide all the configuration required to set the following IKE settings:

- Define traffic to undergo NAT.
- Define an access list for NAT.
- Define the NAT route map.
- Define the NAT interfaces.

Task 3: Define IKE Parameters

Step 1. At the R1 console, provide all the configuration required to set the following IKE settings:

- Define an ISAKMP policy.
- Define the ISAKMP peer and key.

Step 2. At the R2 console, provide all the configuration required to set the following IKE settings:

- Define an ISAKMP policy.

- Define the ISAKMP peer and key.

Step 3. At the R3 console, provide all the configuration required to set the following IKE settings:

- Define an ISAKMP policy.
- Define the ISAKMP peer and key.

Task 4: Define IPSec Parameters

Step 1. At the R1 console, provide all the configuration required to set the following IPSec settings:

- Define a crypto access list.
- Define an IPSec transform set.
- Define the IPSec crypto map.
- Associate the crypto map to the Ethernet 0 interface.

Step 2. At the R2 console, provide all the configuration required to set the following IPSec settings:

- Define a crypto access list.
- Define an IPSec transform set.
- Define the IPSec crypto map.
- Associate the crypto map to the Ethernet 0 interface.

Step 3. At the R3 console, provide all the configuration required to set the following IPSec settings:

- Define a crypto access list.
- Define an IPSec transform set.
- Define the IPSec crypto map.
- Associate the crypto map to the Ethernet 0 interface.

Practical Exercise 14-4 Solution

The following is a step-by-step discussion of the Practical Exercise solution.

Task 1 Solution

IKE uses UDP port 500. The IPSec ESP and AH protocols use protocol numbers 50 and 51. You must ensure that any existing access lists you might already have configured do not block protocol 50, 51, and UDP port 500 traffic at any interface used by IPSec. In some cases you might need to reconfigure an existing access list to explicitly permit this traffic.

Task 2 Solution

Step 1. At the R1 console, provide all the configuration required to set the following IKE settings:

Define traffic to undergo NAT:

```
R1(config)#ip nat inside source route-map nonat interface Serial0 overload
```

Define an access list for NAT:

```
R1(config)#access-list 150 deny ip 192.168.1.0 0.0.0.255 192.168.2.0
  0.0.0.255
R1(config)#access-list 150 deny ip 192.168.1.0 0.0.0.255 192.168.3.0
  0.0.0.255
R1(config)#access-list 150 permit ip 192.168.1.0 0.0.0.255 any
```

Define the NAT route map:

```
R1(config)#route-map nonat permit 10
R1(config-route-map)#match ip address 150
```

Define the NAT interfaces:

```
R1(config)#interface serial0
R1(config-if)#ip nat outside
R1(config-if)#exit
R1(config)#interface ethernet0
R1(config-if)#ip nat inside
```

Step 2. At the R2 console, provide all the configuration required to set the following IKE settings:

Define traffic to undergo NAT:

```
R2(config)#ip nat inside source route-map nonat interface Serial0 overload
```

Define an access list for NAT:

```
R2(config)#access-list 150 deny ip 192.168.2.0 0.0.0.255 192.168.1.0
  0.0.0.255
R2(config)#access-list 150 deny ip 192.168.2.0 0.0.0.255 192.168.3.0
  0.0.0.255
R2(config)#access-list 150 permit ip 192.168.2.0 0.0.0.255 any
```

Define the NAT route map:

```
R2(config)#route-map nonat permit 10
R2(config-route-map)#match ip address 150
```

Define the NAT interfaces:

```
R2(config)#interface serial0
R2(config-if)#ip nat outside
R2(config-if)#exit
R2(config)#interface ethernet0
```

```
R2(config-if)#ip nat inside
```

Step 3. At the R3 console, provide all the configuration required to set the following IKE settings:

Define traffic to undergo NAT:

```
R3(config)#ip nat inside source route-map nonat interface Serial0 overload
```

Define an access list for NAT:

```
R3(config)#access-list 150 deny ip 192.168.3.0 0.0.0.255 192.168.1.0
  0.0.0.255
R3(config)#access-list 150 deny ip 192.168.3.0 0.0.0.255 192.168.2.0
  0.0.0.255
R3(config)#access-list 150 permit ip 192.168.3.0 0.0.0.255 any
```

Define the NAT route map:

```
R3(config)#route-map nonat permit 10
R3(config-route-map)#match ip address 150
```

Define the NAT interfaces:

```
R3(config)#interface serial0

R3(config-if)#ip nat outside

R3(config-if)#exit

R3(config)#interface ethernet0

R3(config-if)#ip nat inside
```

Task 3 Solution

Step 1. At the R1 console, provide all the configuration required to set the following IKE settings:

Define an ISAKMP policy:

```
R1(config)#crypto isakmp policy 4

R1(config-isakmp)#authentication pre-share
```

Define the ISAKMP peer and key:

```
R1(config)#crypto isakmp key cisco1234 address 100.228.202.154

R1(config)#crypto isakmp key cisco1234 address 200.154.17.130
```

Step 2. At the R2 console, provide all the configuration required to set the following IKE settings:

Define an ISAKMP policy:

```
R2(config)#crypto isakmp policy 4

R2(config-isakmp)#authentication pre-share
```

Define the ISAKMP peer and key:

```
R2(config)#crypto isakmp key cisco1234 address 100.228.202.154

R2(config)#crypto isakmp key cisco1234 address 100.232.202.210
```

Step 3. At the R3 console, provide all the configuration required to set the following IKE settings:

Define an ISAKMP policy:

```
R3(config)#crypto isakmp policy 4
```

```
R3(config-isakmp)#authentication pre-share
```

Define the ISAKMP peer and key:

```
R3(config)#crypto isakmp key cisco1234 address 100.232.202.210

R3(config)#crypto isakmp key cisco1234 address 200.154.17.130
```

Task 4 Solution

Step 1. At the R1 console, provide all the configuration required to set the following IPSec settings:

Define a crypto access list:

```
R1(config)#access-list 105 permit ip 192.168.1.0 0.0.0.255 192.168.2.0

  0.0.0.255

R1(config)#access-list 106 permit ip 192.168.1.0 0.0.0.255 192.168.3.0

  0.0.0.255
```

Define an IPSec transform set:

```
R1(config)#crypto ipsec transform-set encrypt-des esp-des
```

Define the IPSec crypto map:

```
R1(config)#crypto map combined local-address serial0
R1(config)#crypto map combined 20 ipsec-isakmp
R1(config-crypto-m)#set peer 100.228.202.154
R1(config-crypto-m)#set transform-set encrypt-des
R1(config-crypto-m)#match address 106
R1(config-crypto-m)#exit
R1(config)#crypto map combined 30 ipsec-isakmp
R1(config-crypto-m)#set peer 200.154.17.130
R1(config-crypto-m)#set transform-set encrypt-des
R1(config-crypto-m)#match address 105
```

Associate the crypto map to the Ethernet 0 interface:

```
R1(config)#interface ethernet 0
R1(config-if)#crypto map combined
```

Step 2. At the R2 console, provide all the configuration required to set the following IPSec settings:

Define a crypto access list:

```
R2(config)#access-list 105 permit ip 192.168.2.0 0.0.0.255 192.168.1.0
  0.0.0.255
R2(config)#access-list 106 permit ip 192.168.2.0 0.0.0.255 192.168.3.0
  0.0.0.255
```

Define an IPSec transform set:

```
R2(config)#crypto ipsec transform-set encrypt-des esp-des
R2(config)#crypto ipsec transform-set 1600_box esp-des
```

Define the IPSec crypto map:

```
R2(config)#crypto map combined local-address serial0
R2(config)#crypto map combined 7 ipsec-isakmp
R2(config-crypto-m)#set peer 100.232.202.210
R2(config-crypto-m)#set transform-set 1600_box
R2(config-crypto-m)#match address 105
R2(config-crypto-m)#exit
```

```
R2(config)#crypto map combined 8 ipsec-isakmp

R2(config-crypto-m)#set peer 100.228.202.154

R2(config-crypto-m)#set transform-set 1600_box

R2(config-crypto-m)#match address 106
```

Associate the crypto map to the Ethernet 0 interface:

```
R2(config)#interface ethernet 0

R2(config-if)#crypto map combined
```

Step 3. At the R3 console, provide all the configuration required to set the following IPSec settings:

Define a crypto access list:

```
R3(config)#access-list 105 permit ip 192.168.3.0 0.0.0.255 192.168.1.0

  0.0.0.255

R3(config)#access-list 106 permit ip 192.168.3.0 0.0.0.255 192.168.2.0

  0.0.0.255
```

Define an IPSec transform set:

```
R3(config)#crypto ipsec transform-set encrypt-des esp-des

R3(config)#crypto ipsec transform-set 1600_box esp-des
```

Define the IPSec crypto map:

```
R3(config)#crypto map combined local-address serial0

R3(config)#crypto map combined 7 ipsec-isakmp

R3(config-crypto-m)#set peer 100.232.202.210

R3(config-crypto-m)#set transform-set encrypt-des

R3(config-crypto-m)#match address 106

R3(config)#crypto map combined 8 ipsec-isakmp

R3(config-crypto-m)#set peer 200.154.17.130

R3(config-crypto-m)#set transform-set 1600_box

R3(config-crypto-m)#match address 105
```

Associate the crypto map to the Ethernet 0 interface:

```
R3(config)#interface ethernet 0

R3(config-if)#crypto map combined
```

Practical Exercise 14-5: IPSec/GRE with NAT

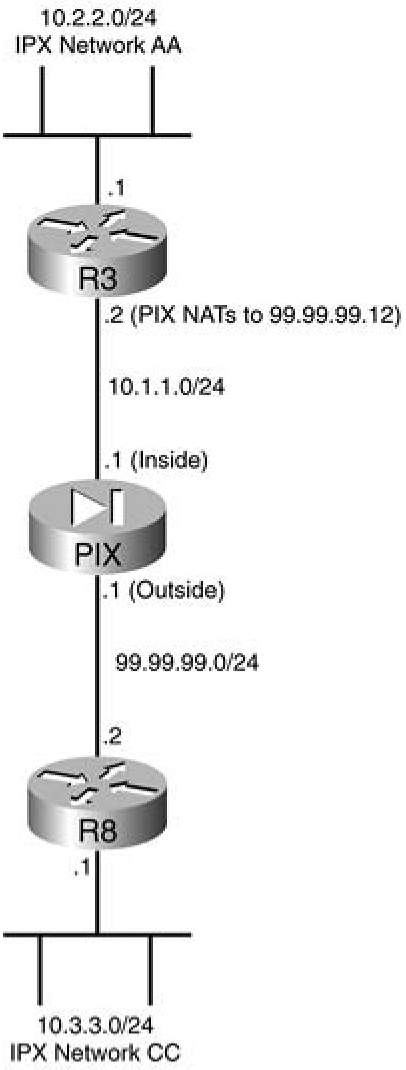
Complete the tasks outlined in this Practical Exercise. Also review the Practical Exercise solution to see how you did and to see what concepts you might need to review.

In this Practical Exercise, you are the administrator of a set of routers, R1 and R2, along with a Cisco PIX. You are required to configure a GRE tunnel with encryption between the routers so that you can pass IPX traffic across the firewall, which is also running NAT.

Background Information

You will configure a GRE tunnel with encryption between two routers with a firewall in the middle, as illustrated in Figure 14-8.

Figure 14-8. IPSec/GRE with NAT



Task 1: Configure PIX

Step 1. At the PIX console, provide all the configuration required to enable traffic flow to and from the PIX firewall:

- Assign addresses to the interfaces.
- Define NAT.
- Associate a global statement to NAT.
- Define the static services allowed from the external network.
- Define the traffic allowed into the network.
- Define routing for the PIX traffic.

Task 2: Configure IPX

Step 1. At the R3 console, provide all the configuration required to configure an IPX network:

- Enable IPX routing.
- Assign addresses to the interfaces.

Step 2. At the R8 console, provide all the configuration required to configure an IPX network:

- Enable IPX routing.
- Assign addresses to the interfaces.

Task 3: Configure IP

Step 1. At the R3 console, provide all the configuration required to configure an IP network:

- Assign addresses to the interfaces.

Step 2. At the R8 console, provide all the configuration required to configure an IP network:

- Assign addresses to the interfaces.

Task 4: Configure the Tunnel

Step 1. At the R3 console, provide all the configuration required to configure the tunnel interface:

- Assign the tunnel source.
- Assign the tunnel destination.
- Define static routing for the tunnel.

Step 2. At the R8 console, provide all the configuration required to configure the tunnel interface:

- Assign the tunnel source.
- Assign the tunnel destination.
- Define static routing for the tunnel.

Task 5: Configure NAT on R8

Step 1. At the R8 console, provide all the configuration required to configure the tunnel interface:

- Identify traffic for NAT to apply to.
- Define the type of NAT to use.
- Apply NAT to the appropriate interfaces.

Task 6: Define IKE Parameters

Step 1. At the R3 console, provide all the configuration required to set the following IKE settings:

- Define an ISAKMP policy.
- Define the ISAKMP peer and key.

Step 2. At the R8 console, provide all the configuration required to set the following IKE settings:

- Define an ISAKMP policy.
- Define the ISAKMP peer and key.

Task 7: Define IPSec Parameters

Step 1. At the R3 console, provide all the configuration required to set the following IPSec settings:

- Define a crypto access list.
- Define an IPSec transform set.
- Define the IPSec crypto map.
- Associate the crypto map to the Ethernet 0 interface.

Step 2. At the R8 console, provide all the configuration required to set the following IPSec settings:

- Define a crypto access list.
- Define an IPSec transform set.
- Define the IPSec crypto map.
- Associate the crypto map to the Ethernet 0 interface.

Practical Exercise 14-5 Solution

The following is a step-by-step discussion of the Practical Exercise solution.

Task 1 Solution

> Step 1. At the PIX console, provide all the configuration required to enable traffic flow to and from the PIX firewall:
>
> Assign addresses to the interfaces:

```
PIX(config)#ip address outside 99.99.99.1 255.255.255.0

PIX(config)#ip address inside 10.1.1.1 255.255.255.0
```

Define NAT:

```
PIX(config)#nat (inside) 1 0.0.0.0 0.0.0.0 0 0
```

Associate a global statement to NAT:

```
PIX(config)#global (outside) 1 99.99.99.50-99.99.99.60
```

Define the static for the services allowed from the external network:

```
PIX(config)#static (inside,outside) 99.99.99.12 10.1.1.2 netmask
  255.255.255.255 0 0
```

Define the traffic allowed into the network:

```
PIX(config)#conduit permit esp host 99.99.99.12 host 99.99.99.2
PIX(config)#conduit permit udp host 99.99.99.12 eq isakmp host 99.99.99.2
```

Define routing for the PIX traffic:

```
PIX(config)#route outside 0.0.0.0 0.0.0.0 99.99.99.2 1
PIX(config)#route inside 10.2.2.0 255.255.255.0 10.1.1.2 1
```

Task 2 Solution

Step 1. At the R3 console, provide all the configuration required to configure an IPX network:

Enable IPX routing:

```
R3(config)#ipx routing 0030.1977.8f80
```

Assign addresses to the interfaces:

```
R3(config)#interface Tunnel0
R3(config-if)#ipx network BB
R3(config)#interface ethernet0
R3(config-if)#ipx network AA
```

Step 2. At the R8 console, provide all the configuration required to configure an IPX network:

Enable IPX routing:

```
R8(config)#ipx routing 0030.80f2.2950
```

Assign addresses to the interfaces:

```
R8(config)#interface Tunnel0

R8(config-if)#ipx network BB

R8(config)#interface ethernet1

R8(config-if)#ipx network CC
```

Task 3 Solution

Step 1. At the R3 console, provide all the configuration required to configure an IP network:

Assign addresses to the interfaces:

```
R3(config)#interface Tunnel0

R3(config-if)#ip address 192.168.100.1 255.255.255.0

R3(config)#interface ethernet0

R3(config-if)#ip address 10.2.2.1 255.255.255.0

R3(config)#interface ethernet1

R3(config-if)#ip address 10.1.1.2 255.255.255.0
```

Step 2. At the R8 console, provide all the configuration required to configure an IP network:

Assign addresses to the interfaces:

```
R8(config)#interface Tunnel0

R8(config-if)#ip address 192.168.100.2 255.255.255.0

R8(config)#interface ethernet0

R8(config-if)#ip address 99.99.99.2 255.255.255.0

R8(config)#interface ethernet1

R8(config-if)#ip address 10.3.3.1 255.255.255.0
```

Task 4 Solution

Step 1. At the R3 console, provide all the configuration required to configure the tunnel interface:

Assign the tunnel source:

```
R3(config)#interface Tunnel0

R3(config-if)#tunnel source ethernet0
```

Assign the tunnel destination:

```
R3(config-if)#tunnel destination 10.3.3.1
```

Define static routing for the tunnel:

```
R3(config)#ip route 10.3.3.0 255.255.255.0 Tunnel0

R3(config)#ip route 10.3.3.1 255.255.255.255 10.1.1.1
```

Step 2. At the R8 console, provide all the configuration required to configure the tunnel interface:

Assign the tunnel source:

```
R8(config)#interface Tunnel0

R8(config-if)#tunnel source ethernet1
```

Assign the tunnel destination:

```
R8(config-if)#tunnel destination 10.2.2.1
```

Define static routing for the tunnel:

```
R8(config)#ip route 0.0.0.0 0.0.0.0 Tunnel0
```

```
R8(config)#ip route 10.2.2.1 255.255.255.255 99.99.99.1
```

Task 5 Solution

Step 1. At the R8 console, provide all the configuration required to configure the tunnel interface:

Identify traffic for NAT to apply to:

```
R8(config)#access-list 1 permit 10.3.3.0 0.0.0.255

R8(config)#ip nat inside source list 1 pool mynat

R8(config)#ip nat pool mynat 99.99.99.70 99.99.99.80 netmask 255.255.255.0
```

Apply NAT to the appropriate interfaces:

```
R8(config)#interface ethernet0

R8(config-if)#ip nat outside

R8(config)#interface ethernet1

R8(config-if)#ip nat inside
```

Task 6 Solution

Step 1. At the R3 console, provide all the configuration required to set the following IKE settings:

Define an ISAKMP policy:

```
R3(config)#crypto isakmp policy 10

R3(config-isakmp)#hash md5

R3(config-isakmp)#authentication pre-share
```

Define the ISAKMP peer and key:

```
R3(config)#crypto isakmp key cisco123 address 99.99.99.2
```

Step 2. At the R8 console, provide all the configuration required to set the following IKE settings:

Define an ISAKMP policy:

```
R8(config)#crypto isakmp policy 10

R8(config-isakmp)#hash md5

R8(config-isakmp)#authentication pre-share
```

Define the ISAKMP peer and key:

```
R8(config)#crypto isakmp key cisco123 address 99.99.99.12
```

Task 7 Solution

Step 1. At the R3 console, provide all the configuration required to set the following IPSec settings:

Define a crypto access list:

```
R3(config)#access-list 101 permit gre host 10.2.2.1 host 10.3.3.1
```

Define an interface for use as an identifier:

```
R3(config)#crypto map mymap local-address ethernet1
```

Define an IPSec transform set:

```
R3(config)#crypto ipsec transform-set myset esp-des esp-md5-hmac
```

Define the IPSec crypto map:

```
R3(config)#crypto map mymap 10 ipsec-isakmp
R3(config-crypto-m)#set peer 99.99.99.2
R3(config-crypto-m)#set transform-set myset
R3(config-crypto-m)#match address 101
```

Associate the crypto map to the Ethernet 0 interface:

```
R3(config)#interface Tunnel0
R3(config-if)#crypto map mymap
R3(config)#interface ethernet1
R3(config-if)#crypto map mymap
```

Configure routing to the peer:

```
R3(config)#ip route 99.99.99.0 255.255.255.0 10.1.1.1
```

Step 2. At the R8 console, provide all the configuration required to set the following IPSec settings:

Define a crypto access list:

```
R8(config)#access-list 101 permit gre host 10.3.3.1 host 10.2.2.1
```

Define an IPSec transform set:

```
R8(config)#crypto ipsec transform-set myset esp-des esp-md5-hmac
```

Define an interface for use as an identifier:

```
R8(config)#crypto map mymap local-address FastEthernet0/0
```

Define the IPSec crypto map:

```
R8(config)#crypto map mymap 10 ipsec-isakmp

R8(config-crypto-m)#set peer 99.99.99.12

R8(config-crypto-m)#set transform-set myset

R8(config-crypto-m)#match address 101
```

Associate the crypto map to the Ethernet 0 interface:

```
R8(config)#interface Tunnel0

R8(config-if)#crypto map mymap

R8(config)#interface ethernet0

R8(config-if)#crypto map mymap
```

Configure routing to the peer:

```
R8(config)#ip route 99.99.99.12 255.255.255.255 99.99.99.1
```

Practical Exercise 14-6: Router to VPN Client with a Preshared Key and NAT

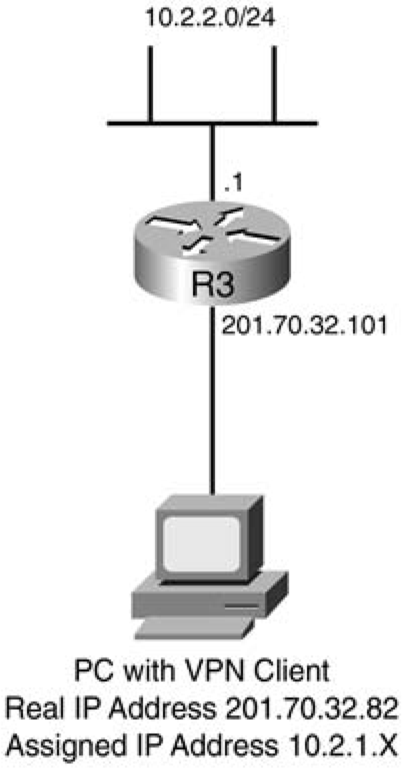
Complete the tasks outlined in this Practical Exercise. Also review the Practical Exercise solution to see how you did and to see what concepts you might need to review.

In this Practical Exercise, you are the administrator of a router that will be the terminating endpoint for VPNs from a VPN client.

Background Information

You will configure your router with the following options. Your router will issue the user an IP address from a pool of addresses, wildcard preshared keys, and NAT. This will allow an off-site user to gain access to your network and have an internal IP address, making it appear to the user that he or she is inside your network. Because you are using private addressing, NAT is involved, and your router must be told what to translate and what not to translate. You will use the topology shown in Figure 14-9.

Figure 14-9. Router to VPN Client with a Preshared Key and NAT



Task 1: Verify Compatibility with Existing Access Lists

To run IKE and IPSec, you need to ensure that any existing access lists are compatible with both protocols. Any existing access lists must allow the ports required by IKE and IPSec to pass through them.

Task 2: Create Network Address Translation

Step 1. At the R3 console, provide all the configuration required to set the following NAT settings:

- Define a NAT pool.
- Define an access list for NAT.
- Define the NAT route map.
- Define the NAT interfaces.

Task 3: Define IKE Parameters

Step 1. At the R3 console, provide all the configuration required to set the following IKE settings:

- Define an ISAKMP policy.
- Define the ISAKMP peer and key.
- Define the address assignment for the client.

Task 4: Define IPSec Parameters

Step 1. At the R3 console, provide all the configuration required to set the following IPSec settings:

- Define an IPSec transform set.
- Define the IPSec dynamic crypto map.
- Define the IPSec crypto map.
- Define the IPSec crypto map.
- Associate the crypto map to the Ethernet 0 interface.

Task 5: Define the Client Parameters

Step 1. On the client PC, provide all the configuration required to create the connection IPSec settings:

- Create the connection.
- Identify the remote peer.
- Identify the Phase 1 information.
- Identify the Phase 2 information.
- Identify the other connection information.

Practical Exercise 14-6 Solution

The following is a step-by-step discussion of the Practical Exercise solution.

Task 1 Solution

IKE uses UDP port 500. The IPSec ESP and AH protocols use protocol numbers 50 and 51. You must ensure that any existing access lists you might already have configured do not block protocol 50, 51, and UDP port 500 traffic at any interface used by IPSec. In some cases you might need to reconfigure an existing access list to explicitly permit this traffic.

Task 2 Solution

Step 1. At the R3 console, provide all the configuration required to set the following NAT settings:

Define a NAT pool:

```
R3(config)#ip local pool ourpool 10.2.1.1 10.2.1.254

R3(config)#ip nat pool outsidepool 201.70.32.150 201.70.32.160 netmask

  255.255.255.0

R3(config)#ip nat inside source route-map nonat pool outsidepool
```

Define an access list for NAT:

```
R3(config)#access-list 101 deny ip 10.2.2.0 0.0.0.255 10.2.1.0 0.0.0.255

R3(config)#access-list 101 permit ip 10.2.2.0 0.0.0.255 any
```

Define the NAT route map:

```
R3(config)#route-map nonat permit 10

R3(config-route-map)#match ip address 101
```

Define the NAT interfaces:

```
R3(config)#interface Ethernet0

R3(config-if)#ip nat outside

R3(config-if)#exit

R3(config)#interface Serial1

R3(config-if)#ip nat inside
```

Task 3 Solution

Step 1. At the R3 console, provide all the configuration required to set the following IKE settings:

Define an ISAKMP policy:

```
R3(config)#crypto isakmp policy 1
R3(config-isakmp)#hash md5
R3(config-isakmp)#authentication pre-share
```

Define the ISAKMP peer and key:

```
R3(config)#crypto isakmp key cisco123 address 0.0.0.0
```

Define the address assignment for the client:

```
R3(config)#crypto isakmp client configuration address-pool local ourpool
```

Task 4 Solution

Step 1. At the R3 console, provide all the configuration required to set the following IPSec settings:

Define an IPSec transform set:

```
R3(config)#crypto ipsec transform-set trans1 esp-des esp-md5-hmac
```

Define the IPSec dynamic crypto map:

```
R3(config)#crypto dynamic-map dynmap 10
R3(config-crypto-m)#set transform-set trans1
```

Define the IPSec crypto map:

```
R3(config)#crypto map intmap 10 ipsec-isakmp dynamic dynmap
```

Define the IPSec parameters:

```
R3(config)#crypto map intmap client configuration address initiate
R3(config)#crypto map intmap client configuration address respond
```

Associate the crypto map to the Ethernet 0 interface:

```
R3(config)#interface Ethernet0

R3(config-if)#crypto map intmap
```

Task 5 Solution

Step 1. On the client PC, provide all the configuration required to create the connection IPSec settings:

Create the connection:

```
1- Myconn

My Identity = ip address

Connection security: Secure

Remote Party Identity and addressing

ID Type: IP subnet

10.2.2.0

Port all Protocol all
```

Identify the remote peer:

```
Connect using secure tunnel

ID Type: IP address

201.70.32.101
```

Identify the Phase 1 information:

```
Authentication (Phase 1)

Proposal 1

Authentication method: pre-shared key

Encryp Alg: DES

Hash Alg: MD5

SA life: Unspecified

Key Group: DH 1
```

Identify the Phase 2 information:

```
Key exchange (Phase 2)

Proposal 1

Encapsulation ESP

Encrypt Alg: DES

Hash Alg: MD5

Encap: tunnel

SA life: Unspecified

no AH
```

Identify any other connection information:

2- Other Connections

Connection security: Non-secure

Local Network Interface

Name: Any

IP Addr: Any

Port: All

Practical Exercise 14-7: PIX to Cisco Secure VPN Client with a Preshared Key

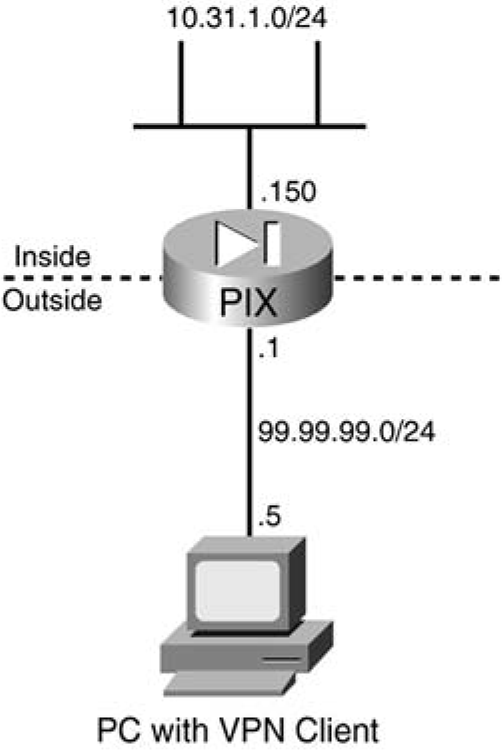
Complete the tasks outlined in this Practical Exercise. Also review the Practical Exercise solution to see how you did and to see what concepts you might need to review.

In this Practical Exercise, you are the administrator of a PIX firewall that will be the terminating endpoint for VPNs from a VPN client.

Background Information

You will configure a VPN client to connect to a PIX firewall using wildcards, mode-config, and the sysopt connection permit-ipsec command. This is used to implicitly permit any packet that came from an IPSec tunnel. It bypasses the checking of an associated access list, conduit, or access group command statement for IPSec connections. The user will have access to everything on your network. You will use the topology illustrated in Figure 14-10.

Figure 14-10. PIX to Cisco Secure VPN Client with a Preshared Key



Task 1: Configure PIX

Step 1. At the PIX console, provide all the configuration required to configure the PIX firewall:

- Define traffic for the mode pool.
- Define the mode pool.
- Prevent NAT for the pool.
- Enable IPSec sysopt.
- Enable ISAKMP.
- Define IKE parameters.
- Define IPSec parameters.

Task 2: Define the Client Parameters

Step 1. On the client PC, provide all the configuration required to create the connection IPSec settings:

- Create the connection.
- Identify the remote peer.
- Identify the Phase 1 information.
- Identify the Phase 2 information.
- Identify the other connection information.

Practical Exercise 14-7 Solution

The following is a step-by-step discussion of the Practical Exercise solution.

Task 1 Solution

Step 1. At the PIX console, provide all the configuration required to configure the PIX firewall:

Define traffic for the mode pool:

```
PIX(config)#access-list 108 permit ip 10.31.1.0 255.255.255.0 172.16.1.0
  255.255.255.0
```

Define the mode pool:

```
PIX(config)#ip local pool test 172.16.1.1-172.16.1.255
```

Prevent NAT for the pool:

```
PIX(config)#nat (inside) 0 access-list 108
```

Enable IPSec sysopt:

```
PIX(config)#sysopt connection permit-ipsec
```

Enable ISAKMP:

```
PIX(config)#isakmp enable outside
```

Define IKE parameters:

```
PIX(config)#isakmp key cisco1234 address 0.0.0.0 netmask 0.0.0.0
PIX(config)#isakmp identity address
PIX(config)#isakmp client configuration address-pool local test outside
PIX(config)#isakmp policy 10 authentication pre-share
PIX(config)#isakmp policy 10 encryption des
PIX(config)#isakmp policy 10 hash md5
PIX(config)#isakmp policy 10 group 1
PIX(config)#isakmp policy 10 lifetime 86400
```

Define IPSec parameters:

```
PIX(config)#crypto ipsec transform-set myset esp-des esp-md5-hmac

PIX(config)#crypto dynamic-map dynmap 10 set transform-set myset

PIX(config)#crypto map mymap 10 ipsec-isakmp dynamic dynmap

PIX(config)#crypto map mymap client configuration address initiate

PIX(config)#crypto map mymap client configuration address respond

PIX(config)#crypto map mymap interface outside
```

Task 2 Solution

Step 1. On the client PC, provide all the configuration required to create the connection IPSec settings:

Create the connection:

```
1- TACconn

My Identity

Connection security: Secure

Remote Party Identity and addressing

ID Type: IP subnet

10.31.1.0

255.255.255.0
```

Port all Protocol all

Identify the remote peer:

Connect using secure tunnel

ID Type: IP address

99.99.99.1

Pre-shared Key=cisco1234

Identify the Phase 1 information:

Authentication (Phase 1)

Proposal 1

Authentication method: pre-shared key

Encryp Alg: DES

Hash Alg: MD5

SA life: Unspecified

Key Group: DH 1

Identify the Phase 2 information:

```
Key exchange (Phase 2)

Proposal 1

Encapsulation ESP

Encrypt Alg: DES

Hash Alg: MD5

Encap: tunnel

SA life: Unspecified

no AH
```

Identify an other connection information:

```
2- Other Connections

Connection security: Non-secure

Local Network Interface

Name: Any

IP Addr: Any

Port: All
```

Practical Exercise 14-8: PIX to Cisco VPN 3000 Client

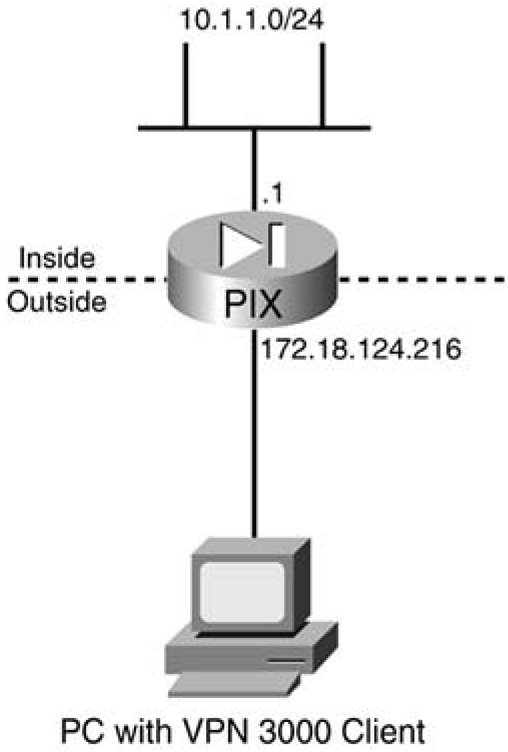
Complete the tasks outlined in this Practical Exercise. Also review the Practical Exercise solution to see how you did and to see what concepts you might need to review.

In this Practical Exercise, you are the administrator of a PIX firewall that will be the terminating endpoint for VPNs from a VPN 3000 client.

Background Information

You will configure your firewall to accept connections from both the Cisco VPN Client 2.5.X and the Cisco VPN Client 3.x. The 2.5.X client will use D-H group 1, the PIX default, and the 3.x client will use D-H group 2. The isakmp policy # group 2 command lets the 3.x clients make a connection. You will define multiple ISAKMP policies to allow the different versions of the VPN 3000 clients to use your firewall as its tunnel endpoint. You will assign IP addresses to the clients as they connect. You will use the topology illustrated in Figure 14-11.

Figure 14-11. PIX to Cisco VPN 3000 Client



Task 1: Configure PIX

Step 1. At the PIX console, provide all the configuration required to configure the PIX firewall:

- Define traffic for the mode pool.

- Define the mode pool.

- Prevent NAT for the pool.

- Enable IPSec sysopt.

- Enable ISAKMP.

- Define IKE parameters for VPN 3000 3.x.

- Define IKE parameters for VPN 3000 2.x.

- Define IKE parameters for all clients.

- Define IPSec parameters.

Task 2: Define the Client Parameters

Step 1. On the client PC, provide all the configuration required to create the connection IPSec settings:

- Click New to create a new connection, and assign a name to your entry in the Connection Entry box.

- Enter the IP address of the destination's public interface.

- Under Group Access Information, enter the group name and group password.

- Click Finish to save the profile in the Registry.

- Click Connect to test the connection.

Practical Exercise 14-8 Solution

The following is a step-by-step discussion of the Practical Exercise solution.

Task 1 Solution

> Step 1. At the PIX console, provide all the configuration required to configure the PIX firewall:
>
> Define traffic for the mode pool:

```
PIX(config)#access-list 101 permit ip 10.1.1.0 255.255.255.0 10.1.2.0
  255.255.255.0
```

Define the mode pool:

```
PIX(config)#ip local pool ippool 10.1.2.1-10.1.2.254
```

Prevent NAT for the pool:

```
PIX(config)#nat (inside) 0 access-list 101
```

Enable IPSec sysopt:

```
PIX(config)#sysopt connection permit-ipsec
```

Enable ISAKMP:

```
PIX(config)#isakmp enable outside
PIX(config)#isakmp identity address
```

Define IKE parameters for VPN 3000 3.x:

```
PIX(config)#isakmp policy 10 authentication pre-share
PIX(config)#isakmp policy 10 encryption des
PIX(config)#isakmp policy 10 hash md5
PIX(config)#isakmp policy 10 group 2
PIX(config)#isakmp policy 10 lifetime 86400
```

Define IKE parameters for VPN 3000 2.x:

```
PIX(config)#isakmp policy 20 authentication pre-share

PIX(config)#isakmp policy 20 encryption des

PIX(config)#isakmp policy 20 hash md5

PIX(config)#isakmp policy 20 group 1

PIX(config)#isakmp policy 20 lifetime 86400
```

Define IKE parameters for all clients:

```
PIX(config)#vpngroup vpn3000 address-pool ippool

PIX(config)#vpngroup vpn3000 dns-server 10.1.1.2

PIC(config)#vpngroup vpn3000 wins-server 10.1.1.2

PIX(config)#vpngroup vpn3000 default-domain cisco.com

PIX(config)#vpngroup vpn3000 idle-time 1800

PIX(config)#vpngroup vpn3000 password cisco

PIX(config)#vpngroup vpn3000 split-tunnel 101
```

Define IPSec parameters:

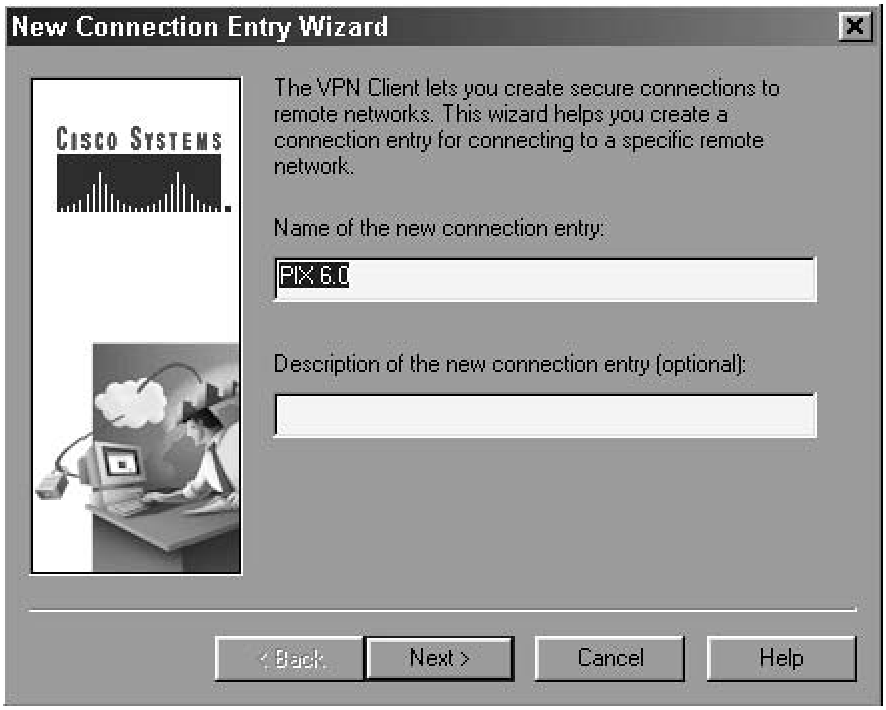
```
PIX(config)#crypto ipsec transform-set myset esp-des esp-md5-hmac
PIX(config)#crypto dynamic-map dynmap 10 set transform-set myset

PIX(config)#crypto map mymap 10 ipsec-isakmp dynamic dynmap

PIX(config)#crypto map mymap interface outside

PIX(config)#crypto dynamic-map dynmap 10 set transform-set myset

PIX(config)#crypto map mymap 10 ipsec-isakmp dynamic dynmap

PIX(config)#crypto map mymap interface outside
```

Task 2 Solution

Step 1. On the client PC, provide all the configuration required to create the connection IPSec settings:

- Click New to create a new connection, and assign a name to your entry in the Connection Entry box, as shown in Figure 14-12.

Figure 14-12. Naming the Entry



New Connection Entry Wizard

The VPN Client lets you create secure connections to remote networks. This wizard helps you create a connection entry for connecting to a specific remote network.

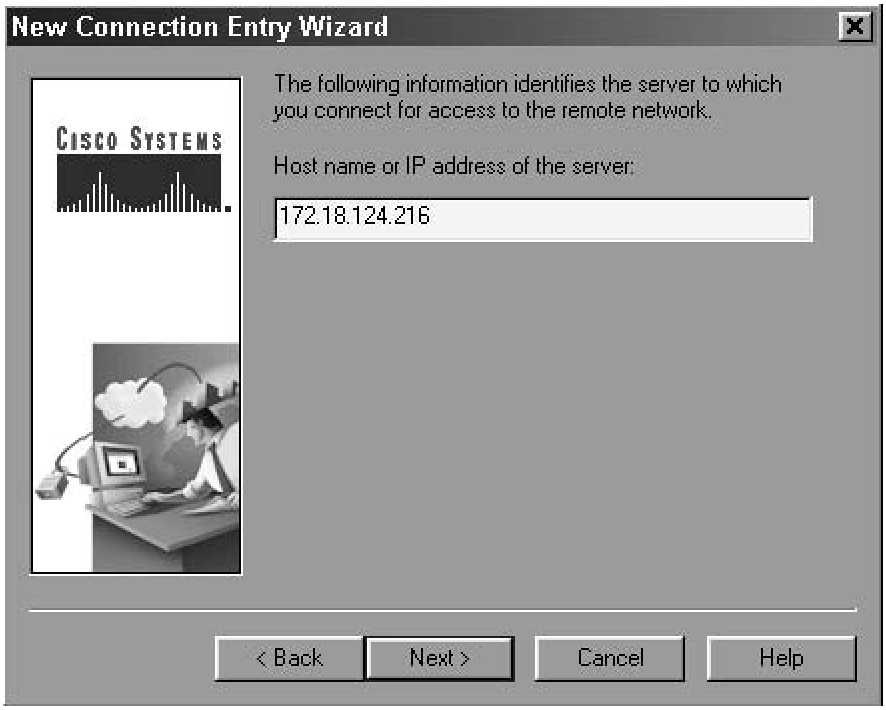
Name of the new connection entry:

PIX 6.0

Description of the new connection entry (optional):

- Enter the IP address of the destination's public interface, as shown in Figure 14-13.

Figure 14-13. Adding the Destination's IP Address

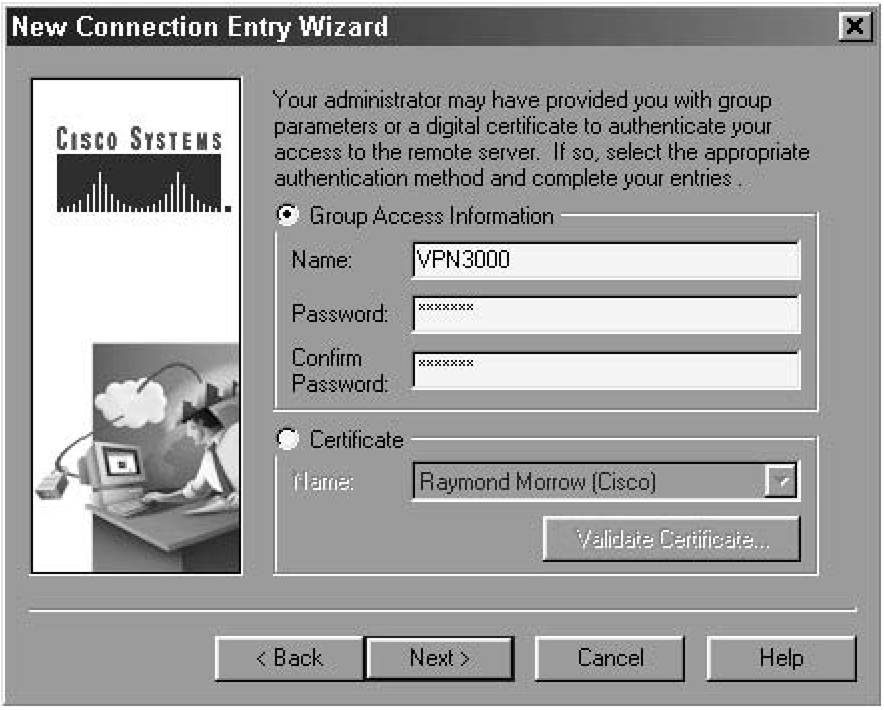


The following information identifies the server to which you connect for access to the remote network.

Host name or IP address of the server: 172.18.124.216

- Under Group Access Information, enter the group name and group password, as shown in Figure 14-14.

Figure 14-14. Adding the Group Name and Group Password



Your administrator may have provided you with group parameters or a digital certificate to authenticate your access to the remote server. If so, select the appropriate authentication method and complete your entries .

● Group Access Information

Name: VPN3000

Password: ********

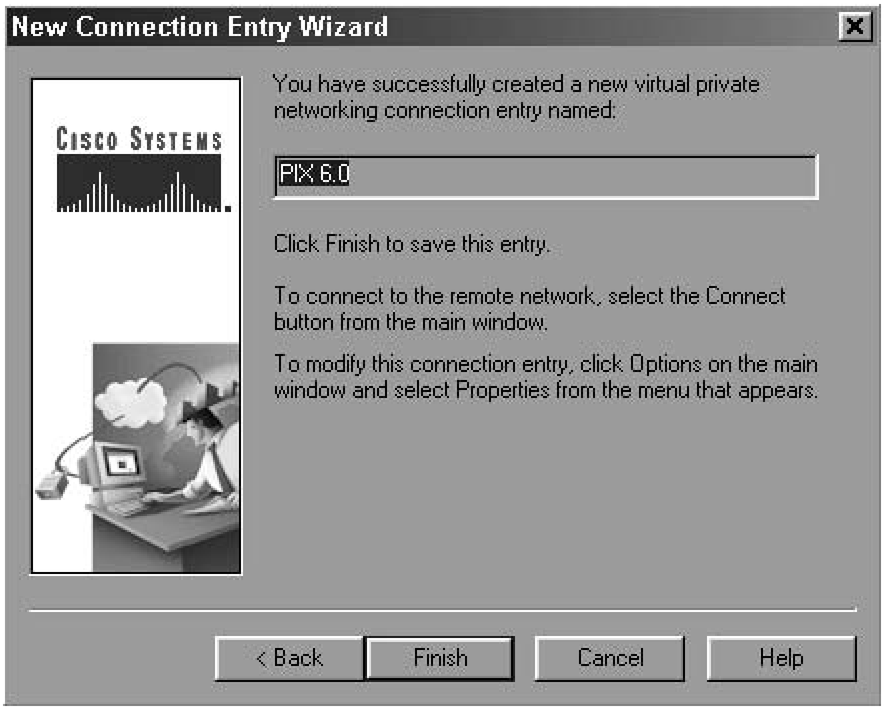
Confirm Password: ********

○ Certificate

Name: Raymond Morrow (Cisco)

- Click Finish to save the profile in the Registry, as shown in Figure 14-15.

Figure 14-15. Saving the Entry



New Connection Entry Wizard

You have successfully created a new virtual private networking connection entry named:

PIX 6.0

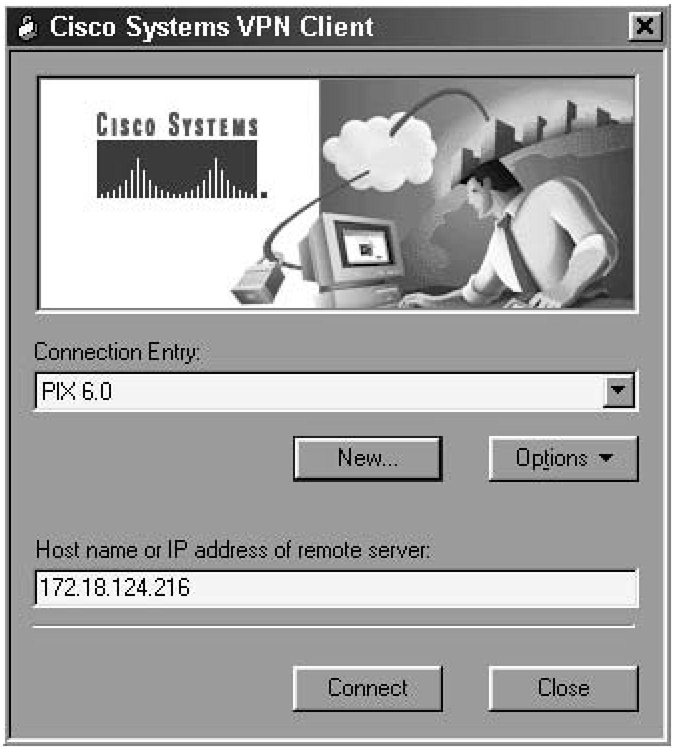
Click Finish to save this entry.

To connect to the remote network, select the Connect button from the main window.

To modify this connection entry, click Options on the main window and select Properties from the menu that appears.

- Click Connect to test the connection, as shown in Figure 14-16.

Figure 14-16. Connecting to the Destination



Connection Entry: PIX 6.0

Host name or IP address of remote server: 172.18.124.216

Practical Exercise 14-9: Layer 2 Tunneling Protocol over IPSec

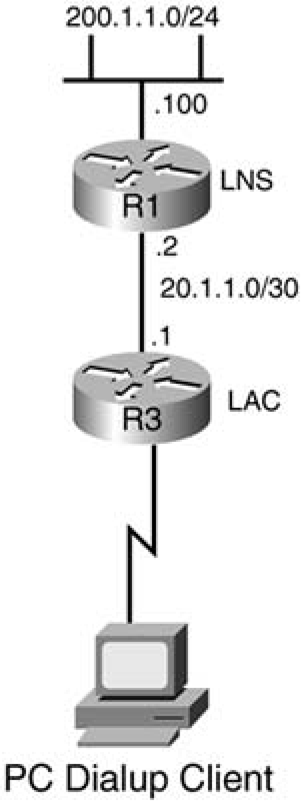
Complete the tasks outlined in this Practical Exercise. Also review the Practical Exercise solution to see how you did and to see what concepts you might need to review.

In this Practical Exercise, you are the administrator of an L2TP Network Server (LNS), R1, and an L2TP Access Concentrator (LAC), dR3, which will be the terminating endpoint for remote dial-in users.

Background Information

You will configure your LAC and LNS to accept incoming L2TP encrypted IPSec connections from remote users. You will use the topology illustrated in Figure 14-17.

Figure 14-17. L2TP over IPSec



Task 1: Configure R3

Step 1. At the R3 console, provide all the configuration required to configure the router as the LAC:

- Create a local account.
- Enable VPDN.
- Create a local IP pool.
- Define an access list that specifies L2TP traffic as interesting.
- Configure an async line.
- Create an IKE policy.
- Define the IKE peer and key.
- Create an IPSec transform set.
- Create a crypto map.
- Assign the crypto map to an interface.

Task 2: Configure R1

Step 1. At the R1 console, provide all the configuration required to configure the router as the LNS:

- Create a local account.
- Enable VPDN.
- Create a local IP pool.
- Define an access list that specifies L2TP traffic as interesting.
- Create a VPDN group to accept tunnel requests.
- Configure the virtual template for cloning.
- Create an IKE policy.
- Define the IKE peer and key.
- Create an IPSec transform set.
- Create a crypto map.
- Assign the crypto map to an interface.

Practical Exercise 14-9 Solution

The following is a step-by-step discussion of the Practical Exercise solution.

Task 1 Solution

Step 1. At the R3 console, provide all the configuration required to configure the router as the LAC:

Create a local account:

```
R3(config)#username LAC password
```

Enable VPDN:

```
R3(config)#vpdn enable

R3(config)#vpdn search-order domain

R3(config)#vpdn-group 1

R3(config-vpdn)#request dialin l2tp ip 20.1.1.2 domain cisco.com

R3(config-vpdn)#local name LAC
```

Create a local IP pool:

```
R3(config)#ip local pool my_pool 10.31.1.100 10.31.1.110
```

Define an access list that specifies L2TP traffic as interesting:

```
R3(config)#access-list 101 permit udp host 20.1.1.1 eq 1701 host 20.1.1.2
  eq 1701
```

Configure an async line:

```
R3(config)#interface Async1
R3(config-if)#ip unnumbered Ethernet0
R3(config-if)#encapsulation ppp
R3(config-if)#async mode dedicated
R3(config-if)#peer default ip address pool my_pool
R3(config-if)#ppp authentication chap
R3(config-if)#exit
R3(config)#line 1
R3(config-line)#autoselect during-login
R3(config-line)#autoselect ppp
R3(config-line)#modem InOut
R3(config-line)#speed 38400
```

```
R3(config-line)#flowcontrol hardware
```

Create an IKE policy:

```
R3(config)#crypto isakmp policy 1
R3(config-isakmp)#authentication pre-share
R3(config-isakmp)#group 2
R3(config-isakmp)#lifetime 3600
```

Define the IKE peer and key:

```
R3(config)#crypto isakmp key cisco address 20.1.1.2
```

Create an IPSec transform set:

```
R3(config)#crypto ipsec transform-set testtrans esp-des
```

Create a crypto map:

```
R3(config)#crypto map l2tpmap 10 ipsec-isakmp
R3(config-crypto-m)#set peer 20.1.1.2
R3(config-crypto-m)#set transform-set testtrans
R3(config-crypto-m)#match address 101
```

Assign the crypto map to an interface:

```
R3(config)#interface Serial0
R3(config-if)#crypto map l2tpmap
```

Task 2 Solution

Step 1. At the R1 console, provide all the configuration required to configure the router as the LNS:

Create a local account:

```
R1(config)#username LNS password cisco
```

Enable VPDN:

```
R1(config)#vpdn enable
```

Create a local IP pool:

```
R1(config)#ip local pool mypool 200.1.1.1 200.1.1.10
```

Define an access list that specifies L2TP traffic as interesting:

```
R1(config)#access-list 101 permit udp host 20.1.1.2 eq 1701 host 20.1.1.1
  eq 1701
```

Create a VPDN group to accept tunnel requests:

```
R1(config)#vpdn-group 1
R1(config-vpdn)#accept dialin l2tp virtual-template 1 remote LAC
R1(config-vpdn)#local name LNS
```

Configure a virtual template for cloning:

```
R1(config)#interface Virtual-Template1
R1(config-if)#ip unnumbered Ethernet0
R1(config-if)#peer default ip address pool mypool
R1(config-if)#ppp authentication chap
```

Create an IKE policy:

```
R1(config)#crypto isakmp policy 1
R1(config-isakmp)#authentication pre-share
R1(config-isakmp)#group 2
R1(config-isakmp)#lifetime 3600
```

Define the IKE peer and key:

```
R1(config)#crypto isakmp key cisco address 20.1.1.1
```

Create an IPSec transform set:

```
R1(config)#crypto ipsec transform-set testtrans esp-des
```

Create a crypto map:

```
R1(config)#crypto map l2tpmap 10 ipsec-isakmp

R1(config-crypto-m)#set peer 20.1.1.1

R1(config-crypto-m)#set transform-set testtrans

R1(config-crypto-m)#match address 101
```

Assign the crypto map to an interface:

```
R1(config)#interface Serial0

R1(config-if)#crypto map l2tpmap
```

Summary

In this chapter, you reviewed the many options available when you're considering the security of your remote-access connection. You read about the Internet Key Exchange (IKE or ISAKMP) protocol and the IP Security (IPSec) protocol, used to achieve a secure connection. You examined the many options available in their implementation. You looked at quality of service (QoS) issues when running on top of these security protocols. You also saw the available show commands.

Review Questions

1: What optional network security services does IPSec offer?

2: When would you apply quality of service parameters to a tunnel interface?

3: Which IPSec options does an IPSec transform set define?

4: What are the two main protocols used with IPSec as implemented by Cisco Systems?

5: IKE is considered what type of protocol and provides IPSec with which services?

6: What is one issue you might encounter when trying to implement QoS within a VPN?

7: What two modes can the authentication header or encapsulating security payload protocols be run in?

8: What four items do IKE peers agree on during negotiations?

9: What three types of VPNs are available to you?

10: What match criteria can you use when classifying packets for QoS?

Appendix A. Answers to Review Questions

Chapter 1

1: What are the main kinds of remote-access users?

A1: Answer:

Corporate users in a branch office

Telecommuters working from home

Traveling users/road warriors

2: At what OSI layer does Frame Relay operate?

A2: Answer: Layer 2

3: What addressing feature of Frame Relay allows for frame routing?

A3: Answer:DLCI(data-link connection identifier)

4: What are some advantages of Frame Relay?

A4: Answer:

It has built-in congestion control.

The ability of traffic to burst.

In a partially meshed network, it can allow for the redirection of traffic around an outage.

5: What are the two main varieties of ISDN?

A5: Answer:BRI(Basic Rate Interface) andPRI(Primary Rate Interface)

6: What are two advantages of ISDN?

A6: Answer:

Quick call setup

It supports a variety of applications.

7: What are the two main varieties of DSL?

A7: Answer: Symmetric and asymmetric

8: What are two advantages of DSL?

A8: Answer:

High bandwidth

It's always on.

9: What are some drawbacks of DSL?

A9: Answer:

Distance limitations

Availability

Speed limitations

Chapter 3

1: Which of following signals does a DTE use to indicate to a DCE that it is ready to accept an incoming call?

A. DSR

B. DTR

C. RTS

D. CTS

A1: Answer: B

2: The DTR, CD, and DSR signals belong to which group of signals?

A. Hardware flow control

B. Modem control

C. Data transfer

A2: Answer: B

3: For which type of connection is null modem cable required?

A. DTE-DCE

B. DCE-DCE

C. DCE-DTE

D. DTE-DTE

A3: Answer: D

4: What command would you use to display status information for all line types?

A. show running-config

B. show line all

C. show line

D. show aux tty vty con

A4: Answer: C

5: Which line type would you associate with line number 0?

A. AUX

B. TTY

C. vty

D. CON

A5: Answer: D

6: Which of the following AT commands are common to most modem types?

A. AT&B1

B. AT&F

C. AT&K1

D. AT&D3

E. ATS2=255

F. AT&M4

A6: Answer: B, D, and E

7: Why would you use the modem autoconfiguration feature?

A. To configure a modem automatically

B. To autodiscover modems

C. To update the modemcap database

D. To configure non-Cisco modems

A7: Answer: A, B, and D

Chapter 4

1: What are the downstream and upstream frequency allocations?

A1: Answer: TheDOCSISupstream frequency is from 5 to 42 MHz. TheDOCSIS downstream frequency is from 88 to 860 MHz.

2: What type of modulation methods are used for the upstream and downstream?

A2: Answer: For the upstream,QPSKor 16-QAMis used. For the downstream, 64-QAMor 256-QAMis used.

3: What servers are required for the cable access solution to work?

A3: Answer: TheDHCP, ToD, and TFTP servers are required.

4: What are the minimum configuration requirements for the CMTS?

A4: Answer:

Set the upstream frequency

Enable the upstream port

Configure the IP address(es)

Configure the helper address

5: What MPEG framing format is used in North America?

A. Annex A

B. Annex B

C. Annex C

A5: Answer: B

6: What configuration is recommended to deal with upstream noise and interference?

A6: Answer: Spectrum management or advanced spectrum management if a Cisco MC16S cable modem card is used.

7: What is the correct syntax to activate upstream port 2 of the cable modem card in slot 4?

A. interface cable 4/2 upstream no shutdown

B. interface cable 4/0 no cable upstream 2 shutdown

C. interface cable 2/0 upstream no shutdown

A7: Answer: B

8: What is the default operating mode of a Cisco cable access router?

A8: Answer: Plug-and-playDOCSIS-compliant bridging mode.

9: What are the required steps to configure the routing mode on the cable access router?

A9: Answer:

1. Enable IP routing.

2. Use the no cable-modem compliant bridge interface command to disableDOCSIS-compliant bridging on the cable interface.

3. Remove the bridge group on the cable and Ethernet interfaces with the no bridge-group interface command.

4. Configure a routing protocol, such as RIP version 2.

10: What command can be used at the CMTS to see the flapping cable modems?

A10: Answer: show cable flap-list

11: What command can be used at the CMTS to find out the registered and unregistered cable modems?

A11: Answer: show cable modem

Chapter 5

1: Which of the following is/are valid PPP authentication methods?

A. PAP

B. CHAP

C. MS-CHAP

D. MS-PAP

A1: Answer: A, B, C

2: True or false: The authentication process is part of LCP negotiation.

A2: Answer: False

3: List at least three possible methods for IP address assignment to the client.

A3: Answer:

ViaAAA

Via the peer default ip address command

Statically assigned

4: When you let the client choose his or her own IP address with the async dynamic address command, your router needs to be in ___________.

A. Dedicated mode

B. Interactive mode

C. Either

D. None of the above

A4: Answer: B

5: Which of the following are valid LCP packet types?

A. CONFNAK

B. CONFREJ

C. CONFREQ

D. All of the above

E. A and C

F. None of the above

A5: Answer: D

6: True or false: BAP's active mode can operate under dialer interfaces, but not under virtual-template interfaces.

A6: Answer: True

7: How can you hard-code the subnet mask during the IP PCP negotiation?

A7: Answer: With the ppp ipcp mask command:

```
ppp ipcp [accept-address | dns [reject | accept | primary-ip-address
  [secondary-ip-address] [accept]] | ignore-map | username unique | wins
  [reject | accept | primary-ip-address [secondary-ip-address] [accept]]]
```

8: What are the main types of compression that PPP supports?

A. Compressor

B. Stacker

C. Predictor

D. LZ compression

E. TCP header

A8: Answer: B, C, E

9: What command allows the router to accept the peer's address?

A9: Answer: dialer in-band

10: Name an interface in control of a bundle in MPPP.

A10: Answer: Bundle master

Chapter 6

1: Which of the following digital services does ISDN provide?

A. Voice

B. Data

C. Text

D. Graphics

E. Music

F. Video

G. All of the above

A1: Answer: G

2: Which of the following services does an NT2 device perform?

A. Compression

B. Switching

C. Concentrating

D. Encryption

A2: Answer: B, C

3: What type of interface can make up the R reference point?

A. EIA/TIA 232-C

B. X.25

C. c.V.24

D. V.35

A3: Answer: A, C, D

4: What type of standard cable does the BRI U interface use?

A. Two-wire

B. Four-wire

C. Six-wire

D. BRI-wire

A4: Answer: A

5: What happens when no more traffic is transmitted over the ISDN call?

A. An idle timer starts.

B. The call disconnects.

C. The bandwidth deteriorates.

D. Unidirectional flow changes directions.

A5: Answer: A

6: What happens if the isdn switch-type command is used in global mode?

A. Only one interface accepts that switch type.

B. All ISDN interfaces assume the same switch type.

C. A few ISDN interfaces assume the same switch type.

D. Integrated services are enhanced.

A6: Answer: B, C

7: True or false: Static routes are used in stub environments to save costs.

A7: Answer: False

8: What type of framing is used for modern T1 PRI configurations?

A. sf

B. esf

C. crc4

D. no-crc4

A8: Answer: B

9: Which linecode type is specified for T1 PRI configuration?

A. ami

B. b8zs

C. hdb3

D. None of the above

A9: Answer: B

10: True or false: Rate adaptation can increase the ISDN channel speed.

A10: Answer: False

Chapter 7

1: What is another name for a dialer interface?

A. Backup dialer interface

B. Ancillary dialer interface

C. Surrogate dialer interface

D. Virtual dialer interface

A1: Answer: D

2: True or false: When a call is triggered, the dialer interface selects a physical interface from the pool.

A2: Answer: True

3: Which of the following cannot be used in the logical configuration?

A. The network layer address

B. Encapsulation

C. The interface media type

D. Dialer parameters

A3: Answer: C

4: True or false: When dialer profiles are used, an active BRI interface can function as a dial backup.

A4: Answer: True

5: Which of the following interfaces can be used with dialer pools? (Choose all that apply.)

A. Frame Relay

B. Serial

C. BRI

D. PRI

A5: Answer: B, C, D

6: What is the correct syntax to prohibit routing updates from being sent on the dialer 1 interface?

A. no routing update dialer 1

B. passive-interface dialer 1

C. dialer 1 no update

D. interface-passive dialer 1

A6: Answer: B

7: What is the main advantage of using dialer rotary groups?

A. They simplify configuration for multiple callers and calling destinations.

B. They organize interface selection in a round-robin fashion.

C. They allow Multilink PPP to be implemented, but only on identical interfaces.

D. They are required for ISDN PRI channel selection.

A7: Answer: A

8: What is the correct syntax for assigning a physical interface to a rotary group?

A. dialer rotary 1

B. rotary-group 1

C. dialer rotary-group 1

D. dialer-group 1

A8: Answer: C

Chapter 8

1: Which of the following modulation methods is not used for ADSL technology?

A. CAP

B. 2B1Q

C. DMT-2

D. G.lite

A1: Answer: B

2: RFC 1483 when implemented is __________.

A. Bridged

B. Routed

C. Decrypted

D. Encrypted

A2: Answer: A

3: PPPoA when implemented is __________.

A. Bridged

B. Routed

C. Decrypted

D. Encrypted

A3: Answer: B

4: Which of the following interferences degrades DSL services?

A. Impedance changes

B. Bridged taps

C. Crosstalk

D. Impulse hits

E. All of the above

A4: Answer: E

5: What is the function of the POTS splitter?

A. It separates low and high frequencies.

B. It manages ADSL signaling.

C. It generates ringing voltage.

D. It boosts the ADSL signal.

A5: Answer: A

6: The DSL interface on a Cisco 827 is __________.

A. An FDDI interface

B. A Frame Relay interface

C. A serial interface

D. An ATM interface

A6: Answer: D

7: With PPP over ATM, __________. (Choose all that apply.)

A. MAC frames are encapsulated into ATM cells

B. UDP frames are encapsulated using RFC 1483

C. IP packets are encapsulated into PPP frames and then into ATM cells

D. IP packets are encrypted

A7: Answer: C, D

8: With RFC 1483 bridging, __________.

A. MAC frames are passed across the bridge after LLC/SNAP information is appended

B. IP frames are passed across the bridge unchanged

C. MAC frames are passed across the bridge unchanged

D. IP packets are encrypted

A8: Answer: A

9: Which of the following cards in the Cisco 6400 can be used for Layer 3 packet services?

A. NSP

B. NLC

C. NRP

D. NI-2

A9: Answer: C

10: Which of the following is part of PPPoA configuration?

A. encapsulation aal5mux ppp Virtual-Template 1

B. encapsulation aal5snap

C. atm route-bridged ip

D. bridge 1 protocol ieee

A10: Answer: A

Chapter 9

1: Frame Relay is what kind of technology?

A. Packet-switched

B. Frame-switched

C. Time-switched

D. DVC-switched

A1: Answer: A

2: Name and briefly describe the two kinds of packet-switching techniques discussed in this chapter.

A2: Answer: With variable-length switching, variable-length packets are switched between network segments to best use network resources until the final destination is reached. Statistical multiplexing techniques essentially use network resources in a more efficient way.

3: Describe the difference between SVCs and PVCs.

A3: Answer: A switched virtual circuit (SVC) is created for each data transfer and is terminated when the data transfer is complete. SVCs have a setup and teardown time associated with them. A permanent virtual circuit (PVC) is a permanent network connection that does not terminate when the transfer of data is complete. Previously not widely supported by Frame Relay equipment, SVCs are gaining popularity in many of today's networks.

4: What is a data-link connection identifier (DLCI)?

A4: Answer: ADLCIis a value assigned to each virtual circuit andDTEdevice connection point in the Frame Relay WAN. Two different connections can be assigned the same value within the same Frame Relay WAN—one on each side of the virtual connection—but two virtual circuits may not share the sameDLCIon a local host.

5: Describe how LMI Frame Relay differs from basic Frame Relay.

A5: Answer:LMIFrame Relay adds a set of enhancements, called extensions, to the features supported by basic Frame Relay. KeyLMIextensions provide global addressing, virtual circuit status messages, and multicasting.

6: True or false: IP unnumbered can be used with Frame Relay.

A6: Answer: True

7: Can Cisco routers connect to other vendor devices over Frame Relay?

A7: Answer: As long as you remember that Cisco routers use a proprietary Frame Relay encapsulation, cisco, by default. To interoperate with other vendors' devices, you should specify the Internet Engineering Task Force (IETF) encapsulation format. You can specifyIETFencapsulation on an interface or per-DLCIbasis.

8: Is Frame Relay inverse-arp on by default?

A8: Answer: inverse-arp is on by default, but the inverse-arp command does not show up in your configuration.

9: Is special configuration required to run OSPF over Frame Relay?

A9: Answer: Frame Relay is treated as a nonbroadcast medium by the Open Shortest Path First (OSPF) routing protocol by default, requiring you to configureOSPFneighbors. There are other methods of handlingOSPFover Frame Relay, depending on whether your network is fully meshed.

10: Is TCP header compression available for use with priority queuing?

A10: Answer: You can use TCP header compression with priority queuing, but this is not recommended. This is because TCP header compression uses an algorithm that requires packets to arrive in order. If packets arrive out of order, a regular TCP/IP packet is reconstructed, but it does not match the original packet, because priority queuing changes the order in which packets are transmitted.

Chapter 10

1: Is it possible to specify the backup load command on subinterfaces? Why or why not?

A1: Answer: No. Because load is calculated on a per-interface basis, the backup load command cannot be configured on subinterfaces.

2: What two circumstances can trigger dial backup?

A2: Answer:

Failure of the primary link

Traffic on the primary link reaching or exceeding the set threshold

3: What is a drawback of using physical interfaces for backup?

A3: Answer: Physical interfaces are placed in standby mode when they are idle and cannot be used to connect to other sites. Backup using dialer profiles overcomes this shortcoming.

4: Which interfaces can be used as backup interfaces?

A4: Answer:

Serial interfaces

ISDNinterfaces

Asynchronous interfaces

Dialer pools

5: What is one reason that ISDN interfaces are used mostly for backup interfaces instead of primary interfaces?

A5: Answer: Cost is the main reason thatISDNinterfaces are used primarily in a backup role.

6: Which command specifies interesting traffic to bring up an ISDN interface?

A6: Answer: The dialer-list command specifies interesting traffic that brings up anISDNinterface.

7: Which command specifies the amount of time before a backup interface is activated in case of a primary link failure?

A7: Answer: The backup delay command allows you to specify the amount of time before the backup interface is activated.

8: Which command specifies the load threshold at which a backup interface is brought up in case of load sharing?

A8: Answer: The backup load command allows you to specify the load threshold at which the backup interface is brought up.

9: What is a possible alternative to dial backup?

A9: Answer: Floating static routes

10: Which commands associate a virtual dialer interface with a physical interface when you configure dialer profiles?

A10: Answer: The dialer pool command on the dialer interface and the dialer pool-member command on the physical interface.

Chapter 11

1: What is the default queuing mechanism for interfaces with speeds of E1 and less?

A1: Answer: Weighted Fair Queuing

2: Which queuing mechanism should you use to give absolute priority to critical traffic?

A2: Answer: Priority queuing

3: Which queuing mechanism ensures that packet trains do not adversely affect critical traffic?

A3: Answer: Weighted Fair Queuing

4: What is the default congestive discard threshold for Weighted Fair Queuing?

A4: Answer: 128

5: How many configurable queues are available for custom queuing?

A5: Answer: 16

6: What is the default byte count for queues in custom queuing?

A. 1024

B. 1500

C. 512

D. 256

A6: Answer: B

7: Which of the following cannot be used to classify packets for priority queuing?

A. Protocol type

B. Ingress interface

C. Packet size in bytes

D. Egress interface

A7: Answer: D

8: Queuing is done on which interface?

A. Ingress interface

B. Egress interface

C. Example interface

D. Weighted interface

A8: Answer: B

Chapter 12

1: Network Address Translation is used to connect private IP internetworks that use __________ IP addresses to connect to the Internet.

A. routable

B. standard

C. nonroutable

D. nonstandard

A1: Answer: C

2: When does the NAT operation take place on a router for inside-to-outside translation?

A. Before the IPSec operation

B. Before the routing decision

C. After the IPSec operation

D. After the routing decision

A2: Answer: D

3: True or false: Cisco IOS NAT cannot be applied to subinterfaces.

A3: Answer: False

4: What allows a single NAT-enabled router to allow some users to use NAT and other users on the same Ethernet interface to continue with their own IP addresses?

A. Access list

B. Route map

C. Policy map

D. Priority map

A4: Answer: A

5: What is used to translate internal (inside local) private addresses to one or more outside (inside global—usually registered) IP addresses?

A. Overboard

B. Network Address Translation

C. Interface Address Translation

D. Port Address Translation

A5: Answer: D

6: When using PAT, also known as NAT overloading, how many theoretical translations can be made for each inside global IP address?

A. 30,000

B. 25,655

C. 65,535

D. 100,000

A6: Answer: C

7: PAT additionally translates which port to keep track of individual conversations?

A. Inside source

B. Outside source

C. Inside destination

D. Outside destination

A7: Answer: A

8: IP address __________ refers to a situation in which two locations use the same IP address range but still want to communicate.

A. overloading

B. underloading

C. overlapping

D. underlapping

A8: Answer: C

9: True or false: Static and dynamic NAT may be used on the same router.

A9: Answer: True

Chapter 13

1: What does AAA stand for?

A1: Answer: Authentication, authorization, and accounting

2: What are the two modes supported by AAA commands except for the aaa accounting system command?

A2: Answer: Character and packet mode

3: Which protocol encrypts the entire body of the packet—RADIUS or TACACS+?

A3: Answer: TACACS+

4: Which protocol encrypts only the password in the access request packet from the client to the server—RADIUS or TACACS+?

A4: Answer:RADIUS

5: True or false: RADIUS uses UDP, and TACACS+ uses TCP.

A5: Answer: True

6: Which of the following commands are used for packet mode operation?

A. aaa authentication login default group tacacs+

aaa authorization network default group tacacs+

B. aaa authentication login default group tacacs+

aaa authorization exec default group tacacs+

C. aaa authentication ppp default group tacacs+

aaa authorization exec default group tacacs+

D. aaa authentication ppp default group tacacs+

aaa authorization network default group tacacs+

A6: Answer: D

Chapter 14

1: What optional network security services does IPSec offer?

A1: Answer: Data confidentiality, data integrity, data origin authentication, anti-replay

2: When would you apply quality of service parameters to a tunnel interface?

A2: Answer: When you are usingGREand IP-in-IP (IPIP) tunnel protocols.

3: Which IPSec options does an IPSec transform set define?

A3: Answer:

Mechanism for payload authentication—AHtransform

Mechanism for payload encryption—ESPtransform

IPSec mode—Transport versus tunnel

ESPtransform of the quality of service parameters

4: What are the two main protocols used with IPSec as implemented by Cisco Systems?

A4: Answer: The authentication header and the encapsulation security payload are both used with IPSec.

5: IKE is considered what type of protocol and provides IPSec with which services?

A5: Answer:IKEis considered a hybrid protocol. It is used to provide IPSec with utility services, such as the establishment of a shared secret.

6: What is one issue you might encounter when trying to implement QoS within a VPN?

A6: Answer: One issue you might face when implementing QoS in aVPNtunnel is the requirement that the QoS parameter you normally find in the header of the IP packet needs to be reflected in the tunnel packet header, regardless of the type of tunnel you choose to use.

7: What two modes can the authentication header or encapsulating security payload protocols be run in?

A7: Answer: They can be run in tunnel mode or transport mode.

8: What four items do IKE peers agree on during negotiations?

A8: Answer: An encryption algorithm, a hashing algorithm, an authentication method, the lifetime of the SA.

9: What three types of VPNs are available to you?

A9: Answer: Access, site-to-site, extranet

10: What match criteria can you use when classifying packets for QoS?

A10: Answer:

IP addresses

TCP/UDPport numbers

IP precedence—the 3 bits in the ToS field of the IP packet header

URL and sub-URL

MACaddresses

Time of day

[SYMBOL] [A] [B] [C] [D] [E] [F] [G] [H] [I] [L] [M] [N] [O] [P] [Q] [R] [S] [T] [U] [V] [W]