IP COMMUNICATIONS

Fax, Modem, and Text for IP Telephony

The Definitive Resource for Understanding, Designing, Configuring, and Troubleshooting Fax, Modem, and Text in Today's IP Networks

David Hanes, CCIE® No. 3491
Gonzalo Salgueiro, CCIE No. 4541

ciscopress.com

Fax, Modem, and Text for IP Telephony

David Hanes, CCIE No. 3491
Gonzalo Salgueiro, CCIE No. 4541

Cisco Press

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

Fax, Modem, and Text for IP Telephony

David Hanes, Gonzalo Salgueiro

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

Printed in the United States of America

First Printing June 2008

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

ISBN-13: 978-1-58705-269-9

ISBN-10: 1-58705-269-5

Warning and Disclaimer

This book is designed to provide information about fax, modem, and text technologies for IP Telephony. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

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

The opinions expressed in this book belong to the 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.

Corporate and Government Sales

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

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

Feedback Information

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

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

We greatly appreciate your assistance.

Publisher	Paul Boger
Associate Publisher	Dave Dusthimer
Cisco Representative	Anthony Wolfenden
Cisco Press Program Manager	Jeff Brady
Executive Editor	Chuck Toporek
Managing Editor	Patrick Kanouse
Senior Development Editor	Christopher Cleveland
Project Editor	Mandie Frank
Copy Editor	Keith Cline
Technical Editors	Richard Collette, John Combs, Bryan Deaver, Steve Ganem, Paul Giralt, Dr. Judy Harkins, Paul E. Jones, Aaron Leonard, Matthew Miller, Robert Moran, Thomas Runyan, Anne Smith, Michael Whitley, Brett Wiggins
Editorial Assistant	Vanessa Evans
Book Designer	Louisa Adair
Cover Designer	Louisa Adair
Composition	Octal Publishing, Inc.
Indexer	Tim Wright
Proofreader	Williams Woods Publishing Services, LLC

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

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

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

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

About the Authors

David Hanes, CCIE No. 3491, currently works as an engineer for the Cisco Customer Assurance Engineering (CAE) group based out of Research Triangle Park, North Carolina supporting various emerging technologies through product testing and field trials. In addition, David is a technical expert for Cisco in the area of fax over IP technologies and assists with network design and troubleshooting for critical fax over IP deployments. Since joining Cisco in 1997, he has worked as a Technical Assistance Center (TAC) engineer for the WAN, WAN Switching, and Multiservice Voice teams, a team lead for the Multiservice Voice team, and an Escalation Engineer covering a variety of voice and fax technologies. David has troubleshot escalated issues in Cisco customer networks worldwide and remains a technical resource for other Cisco employees and customers. Before working at Cisco, David was a Systems Engineer for Sprint, where he gained his first computer networking experience working on the Frame Relay and X.25 protocols. He holds a bachelor of science degree in electrical engineering from North Carolina State University.

Gonzalo Salgueiro, CCIE No. 4541, is a senior engineer for the Unified Communications Infrastructure Escalation team of the Technical Assistance Center (TAC) in Research Triangle Park, North Carolina. In his current role, he is a technical leader for fax and voice over IP technologies working directly with Cisco development engineering, TAC support teams, and product serviceability organization, providing support for various Unified Communications products and technologies. Over the past 12 years at Cisco, he has specialized in troubleshooting complex issues for some of the largest VoIP networks and has provided technical leadership for some of the most critical worldwide voice and fax deployments. Before joining the Escalation team in 1999, Gonzalo had roles as a TAC engineer for both the Access/Dial and Multiservice Voice teams and as a team lead for the Access/Dial team. Gonzalo has developed and delivered all levels of training and documentation on these technologies both internally to Cisco technical teams and externally to Cisco customers worldwide. He holds a bachelor of science degree in physics from Jacksonville University and a master of science degree in physics from the University of Miami.

About the Technical Reviewers

Richard Collette is a senior project manager for Sagem-Interstar, where his primary focus is the development and troubleshooting of Voice over IP applications. He holds a bachelor's degree in electrical engineering from l'École Polytechnique de Montréal. Richard has been involved in the design and development of fax applications for more than 15 years.

John Combs is manager of Accessibility/Safety/Telecom for the Cisco Compliance & Certification department. He is responsible for ensuring that Cisco products meet all applicable accessibility, safety, and telecom government regulatory requirements in all the countries where Cisco products are sold. He started the first Cisco accessibility program and test lab in 1999, and is a technical expert on analog and digital telephony, and Telephone Devices for the Deaf (TDD/TTY). John has been at Cisco for 10 years and has worked in the field of regulatory compliance testing for 25 years.

Bryan Deaver is an escalation engineer in the Technical Assistance Center in San Jose, California. He has worked in various technologies and positions in the TAC since joining Cisco in 1993. He primarily focuses on Cisco VoIP products and also has experience in WAN and access technologies. He is a double CCIE in Routing and Voice and works on some of the most complex field issues.

Steve Ganem is the former vice president of World Wide Customer Service for Brooktrout Technology where he was responsible for providing support and training worldwide to customers developing, installing, and maintaining Brooktrout based applications. Steve spent 18 years with Brooktrout (later Cantata and Dialogic) and was heavily involved with Brooktrout's fax products since their introduction in 1990. Steve previously worked in product development at Motorola where he developed telephony circuit boards and embedded software for Motorola communications devices. Steve holds a bachelor's degree in electrical and computer engineering from Northeastern University. Steve currently provides consulting services to technology companies in the Boston area who are looking to improve their customer support and sustaining engineering teams.

Paul Giralt, CCIE Voice, CCIE Routing and Switching No. 4973, is a customer support engineer in the Cisco Voice Technology Group (VTG) and has been with Cisco for 10 years. He assists in designs, troubleshoots problems, and does testing for VTG's most strategic customers. Paul provides feedback to the development teams on feature requirements and participates in feature design meetings to help ensure features are designed to work in real-world customer environments. He also assists in alpha and early field trials of VTG products with the goal of finding and resolving issues in a real-world environment before the products get into customer hands.

Dr. Judy Harkins is a professor in Gallaudet University's Department of Communication Studies. She is the founding director of the Technology Access Program, and has directed approximately 15 sponsored projects on access to communications and educational applications of technology. She is a principal investigator on two Rehabilitation Engineering Research Centers (RERC)—one on Telecommunications Access in cooperation with the Trace Center, University of Wisconsin, and one on Hearing Enhancement, in cooperation with the Department of Audiology and Speech-Language Pathology at Gallaudet. These centers are funded by the National Institute on Disability and Rehabilitation Research. Dr. Harkins also teaches coursework on communication accessibility to Gallaudet undergraduates.

Paul E. Jones has been involved in research and development of protocols and system architectures in the area of multimedia communications, including voice, video, and data conferencing over IP networks, since 1996. In addition to architecture and software development activities within the Cisco Voice Technology Group, he has actively participated in a number of standards and industry organizations, including the ITU, TIA, IETF, ETSI, and the IMTC. Most notably, he served as editor of ITU-T Recommendation H.323 and more recently as Rapporteur for the H.323 experts group. He is also an active participant in accessibility-related work in the ITU and TIA, with a particular emphasis on technology needed for the deaf and hard of hearing. He is presently engaged in the study of an advanced multimedia system within the ITU-T, which is envisaged to enable users to use a multiplicity of devices and networks in parallel to realize richer multimedia communication capabilities.

Aaron Leonard is a customer support engineer in Cisco TAC's Core Infrastructure escalation team. He's been with Cisco for 12 years, providing last-line technical support for TCP/IP stacks, modems, and now 802.11. In his 20+ years in the networking field, Aaron has done everything from vampire-tapping a transceiver into a 10BASE5 cable to chasing down audio glitches heard on WiFi phones in a pet supplies store.

Matthew Miller, product manager at Sagem-Interstar, is responsible for the entire XMediusFAX product line life cycle, from strategic planning to tactical activities. He also manages the Technology Partner Program, building and maintaining relationships with key technology vendors and their associated partnership programs. Matthew has worked for Sagem-Interstar for the past six years, held positions in both the Support and Sales Engineering groups, and has extensive knowledge of IP and fax over IP technology.

Robert Moran is the fax product line director at Dialogic Corporation. Rob is responsible for the profitability of Dialogic's fax products, including the Dialogic Brooktrout TR1034, SR140, and TruFax product lines. Rob defines the product requirements for engineering, supports the Dialogic sales team, and manages other aspects of his portfolio such as pricing and forecasts to ensure the success of his products. Rob has been a product manager in the telecommunications industry for more than 11 years. Before Dialogic, Rob held several positions with Lucent Technologies.

Thomas Runyan is a customer support engineer for Cisco Customer Assurance Engineering. He is part of a team that evaluates and supports Cisco emerging technologies through field trials, internal lab testing, and critical accounts support. Tom was introduced to facsimile and teletype when he started working at 3M in the 1970s. At that time, 3M was a leader in the facsimile and telecommunications field. Since then, he has spent the majority of his career in the telecommunications field. Over the past 10 years, Tom has held various positions within Cisco, working on emerging technologies, voice, and telecommunications projects.

Anne Smith is a technical writer in the CallManager support group at Cisco. Her expertise lies in comprehending and distilling complex technical information into comprehensive, accurate, and readable documentation.

Michael Whitley, CCIE R&S and Voice No. 3645, recently transitioned into a sales engineering role in Cisco Federal Area, specializing in Unified Communications. In his previous eight years at Cisco, Michael supported Cisco UC products and technologies on the TAC Escalation team. Responsibilities included finding, re-creating, and documenting problems while working escalated customer cases. Michael has been at Cisco for nearly 12 years.

Brett Wiggins, CCIE No. 4998, is a technical marketing engineer for the Cisco Voice Solutions Engineering team in Research Triangle Park, North Carolina. This team's main charter is to publish the Unified Communications *Solution Reference Network Design* (SRND) guides on Cisco.com. He has been focused on Cisco voice solutions in a variety of roles for the majority of his 10 year career at Cisco. He graduated from North Carolina State University with a bachelor of science degree in 1996.

Dedications

David Hanes: I dedicate this book to my three girls—my loving wife, Holly, and our beautiful daughters, Haley and Hannah. You all are true blessings and the joys of my life.

Gonzalo Salgueiro: I dedicate this book to my loving family. To my wife, Rebecca, the love of my life, who has unconditionally supported and encouraged me throughout this long endeavor. To my amazing son, Alejandro, who has given me a new perspective on life.

I also dedicate this book to my parents, Alberto and Elena, whose love, support, and example have served as life-long inspiration.

Acknowledgments

David Hanes: First, I would like to thank Gonzalo Salgueiro for co-authoring this book with me. I can honestly say that this book would never have happened without him. Gonzalo's diligence in ensuring that this book achieved the highest level of excellence and accuracy was unparalleled. His meticulous attention to detail and long hours late at night working in the lab and reviewing chapters required many sacrifices and time away from his family. In addition to being my co-author in the writing of this book, Gonzalo is also a great friend. For his tireless dedication and countless hours spent on this book, I will always be appreciative.

Thanks to Anne Smith for being my guide and mentor throughout this book writing process. What started out as a simple e-mail asking how to get started writing a technical book has resulted in this. From the proposal submission to chapter reviews to helping me deal with all the different twists and turns of being an author, Anne was always there to help out. She never failed to take the time out of her own busy schedule to answer questions, edit chapters, and provide wisdom, insight, and perspective based on her own extensive publishing experience.

Thanks to Paul Giralt for providing the inspiration and motivation to write this book. While writing a book was always a goal of mine, it was not until Paul and company wrote *Troubleshooting Cisco IP Telephony* that I really decided to make this goal a reality. I can only hope that this book receives half the acclaim and notoriety of Paul's book, a tour de force in the technical book world that I still refer to quite regularly. Also, I want to thank Paul for always taking the time to provide assistance with anything related to Unified Communications and for being a technical reviewer of this book.

Thanks to my manager Jim Hofmann and to Robert Santiago, John Selden, Mike Quinn, and the rest of the CAE organization for their support of this book. I have always felt very fortunate for the opportunity to work with such a wonderful group of professionals in an organization where I can truly make a difference. A special personal thanks goes to Jim, who has been especially understanding and patient with the numerous deadlines and delays that come along with authoring a book. It is an honor and a privilege to work for someone like Jim while being a part of such a great organization like CAE.

Thanks to all the technical reviewers—Richard Collette, John Combs, Bryan Deaver, Steve Ganem, Paul Giralt, Dr. Judy Harkins, Paul E. Jones, Aaron Leonard, Matthew Miller, Rob Moran, Tom Runyan, Anne Smith, Mike Whitley, and Brett Wiggins—for their dedication and long hours in proofreading chapters within this book. Additional thanks goes to Bryan Deaver and Mike Whitley, two top engineers at Cisco whom I deeply respect and admire who served as full-book reviewers and technically reviewed every chapter. Mike and Bryan possess a combination of incomparable technical depth and breadth of IP telephony along with a punctilious attention to detail that helped ensure the technical integrity of this book.

Thanks to Tomas DeLeon, Paul Giralt, Dave Goodwin, Marty Hussey, Jose Martinez, Peng Mok, Rafael Muller, Steve Penna, Joe Pinkus, Gonzalo Salgueiro, Markus Schneider, Mike Whitley, and all the people whom I served with in the old RTP VNT team. I learned something new practically every day I worked with all of you, and I will always be proud to have been a part of such an amazing group of engineers.

Thanks to Steve Penna for teaching me so much about faxes and modems over the years and for always being there to help out no matter what the problem is. From complex fax and modem corruption issues to lending a hand with customer proof-of-concept testing, Steve is a wealth of knowledge and expertise, and he never ceases to amaze me. Even more important, Steve is a loyal friend whose admirable qualities of honesty, integrity, and generosity are unsurpassed.

Thanks to Gordon Earl, Wei Wang, Ron Rappel, Hanh Luong, and the rest of the folks in the Santa Barbara DSP team for always taking the time to answer questions and provide detailed explanations. You guys are always a pleasure to work with, and your assistance with understanding the complex operation of the DSPs in Cisco voice gateways was crucial to this book.

Thanks to my teammates Mike O'Brien and Jim Frauenthal for lending their expertise and experience and taking the time to review and edit the loss planning section.

Thanks to Stacy Shepherd for her assistance with helping me understand the various IOS processes and how they relate to the DSP and for always being willing to give me a hand with IOS- and DSP-related issues.

Thanks to Sébastien Biore-Lavigne, Marco Brugge, Richard Collette, Nick Diciaccio, Steve Ganem, Bob Green, Matthew Miller, Rob Moran, and Jennifer Van Lent for educating me on fax server technology. I have enjoyed working with each and every one of you, and thanks for always being there to answer my questions.

Thanks to Chris Cleveland, Mandie Frank, and Kristin Weinberger at Cisco Press for all their hard work in getting this book published. Your professionalism and dedication made working on this project together a pleasure.

I also want to thank the following individuals and groups for their contributions to the book: Arun Arunachalam for answering all of my SIP questions; Adam Gensler for assisting with all of those lab recreates; Christina Hattingh for always finding an answer to any VoIP-related question; Wenqing Jin for answering our questions on T.37; Jeff Johnson for his help with secure modem relay; John Kane for developing my idea into a proposal that ultimately led to this book; James Rafferty for assisting with the fax history section; Chester Rieman for answering SP-related fax questions; Brent Rindal and Matt Portoni for their assistance with the Cisco fax server; Andrea Saks for answering questions on text telephony; Avery Till for answering our VG248 questions; Steven White for his white papers and vision into the FoIP world; all the worldwide TAC engineers whom I have worked with over the years, especially the RTP engineers for being the best support engineers in the business; all the DEs and technical people within Cisco who have taken time to answer questions and write documents explaining how fax, modem, and text works with Cisco voice gateways; and of course, all the customers who I have worked with over the years for teaching me and helping me grow as an engineer.

Gonzalo Salgueiro: First and foremost, I want to extend my deepest gratitude to my friend and coauthor David Hanes for presenting me with the once-in-a-lifetime opportunity to be a part of this project. Despite the endless hours, grueling work, and stressful deadlines, this experience was both fulfilling and enjoyable because of his involvement. His single-minded dedication and willingness to persevere through all obstacles have made this book possible.

Heartfelt thanks to Michael Whitley for his steadfast commitment to the success of this book. His guidance and counsel on so many aspects of this project were vital, especially taking on the colossal task of reviewing the entire manuscript. Mike's unmatched technical expertise and willingness to help are surpassed only by his character and integrity.

I'd like to give special recognition to Bryan Deaver for providing his expertise in voice and fax to this project and investing countless hours in carefully reviewing the entire book and in the process making it superior.

Sincere thanks to Aaron Leonard for all his technical material on modems that he made freely available to me and his willingness to answer my many questions.

I'd like to thank Carla Kochmann for all that she has taught me over the years about modems and her assistance on many difficult customer issues. I'm also grateful to Brooks Read for his help on better understanding the operation of modem relay for this book.

Thanks to Budd Carle and his team for years of valuable development support. Special thanks goes to Stacy Shepherd for her assistance on this project and for years of tireless help and countless discussions.

Thanks to Gordon Earl for his endless help and support over the past decade of working on myriad fax and voice DSP issues. Also, thanks to Dan Lai and Gordon for always making their development teams available to me for assistance, especially Hanh Luong, who has been instrumental on many fax-related issues.

Thanks to Steven White for his help and support of this book and the chance to support and lend design assistance to some of the biggest fax customers of Cisco. I appreciate the opportunity to contribute to the development of some of the new fax and text over IP features that Cisco has released over the years.

Thanks to all the technical reviewers: Richard Collette, John Combs, Bryan Deaver, Steve Ganem, Paul Giralt, Dr. Judy Harkins, Paul E. Jones, Aaron Leonard, Matthew Miller, Rob Moran, Tom Runyan, Anne Smith, Mike Whitley, and Brett Wiggins. This book wouldn't be the same without your dedication, hard work, and technical expertise.

Wholehearted thanks to Steve Penna for the countless hours he has spent imparting his amazing knowledge and immeasurable talents in teaching me about voice and dial technologies over the past dozen years. He remains the gold standard, and I owe much of the success in my career at Cisco to him.

Thanks to Paul Giralt for his help with this book and for always having the good will to share his mastery of Unified Communications with me.

Thanks to Michael Whitley, Steve Penna, Paul Giralt, David Hanes, Jose Martinez, Rafael Muller, Joe Pinkus, Markus Schneider, and Tomas DeLeon of the RTP Voice Network Team (VNT), whom I have had the pleasure to work with for many years and from whom I have learned a tremendous amount about voice and fax.

A big "thank you" goes out to the production team for this book, especially Christopher Cleveland, Mandie Frank, and Kristin Weinberger, who have been incredibly professional, responsive, and a true pleasure to work with. In addition, I want to thank John Kane for all of his initial work in taking this idea off the ground and turning this book into a reality.

Thanks to my teammates Peng Mok, Arun Arunachalam, and Wes Sisk for answering my questions and providing superb and skillful help during this project.

Thanks to my managers Michael Stallings and Scott Lawrence for their constant support and encouragement during the course of this two year endeavor. This thanks extends to Joe Novak, Marty Martinez, and the rest of the Customer Advocacy (CA) leadership team who have allowed me to pursue this unique opportunity.

Many thanks to Bob Green from Dialogic, and Jennifer Van Lent and Marco Brugge from Captaris, for their cooperation and assistance and for the opportunity to communicate our knowledge of FoIP to the fax server community.

Thanks to the engineers of the worldwide Voice and Access support teams whom I have worked with over my years in technical support. A special thanks goes to Adam Gensler and George Matroni for providing lab gear and assisting with portions of the testing done for this book.

A hearty thanks to all the customers, developers, engineers, and technical writers with whom I have worked during my years at Cisco. I am appreciative of the constant help and the knowledge that they have imparted to me throughout my very enjoyable career in networking. Special thanks to Christina Hattingh, Wenqing Jin, Jeff Johnson, Nandita Shenoy, and Avery Till for their direct assistance with various technical issues in this book.

This Book Is Safari Enabled

The Safari® Enabled icon on the cover of your favorite technology book means the book is available through Safari Bookshelf. When you buy this book, you get free access to the online edition for 45 days.

Safari Bookshelf is an electronic reference library that lets you easily search thousands of technical books, find code samples, download chapters, and access technical information whenever and wherever you need it.

To gain 45-day Safari Enabled access to this book:

- Go to http://www.informit.com/onlineedition
- Complete the brief registration form
- Enter the coupon code 44J1-F9SK-9Y71-5DKG-K6QX

If you have difficulty registering on Safari Bookshelf or accessing the online edition, please e-mail customer-service@safaribooksonline.com.

Contents at a Glance

Introduction xxiii

Part I Laying the Groundwork 3

Chapter 1 How Modems Work 5

Chapter 2 How Fax Works 53

Chapter 3 How Text Telephony Works 107

Part II IP Solutions and Design 127

Chapter 4 Passthrough 129

Chapter 5 Relay 151

Chapter 6 T.37 Store-and-Forward Fax 189

Chapter 7 Design Guide for Fax, Modem, and Text 207

Chapter 8 Fax Servers 263

Part III Configuration 285

Chapter 9 Configuring Passthrough 287

Chapter 10 Configuring Relay 311

Chapter 11 Configuring T.37 Store-and-Forward Fax 351

Part IV Troubleshooting 377

Chapter 12 Troubleshooting Passthrough and Relay 379

Chapter 13 Troubleshooting T.37 Store-and-Forward Fax 525

Index 562

Contents

Introduction xxiii

Part I Laying the Groundwork 3

Chapter 1 How Modems Work 5

A Brief History of Modems 5

Modem Architecture 8

Modem Types 10
- External Versus Internal Modems 10
- Hardware Versus Software Modems 10
- Fax Modems 12

Terminal-to-Modem Communication 14
- DTE and DCE 15
- RS-232 Signaling 15
- Asynchronous Framing 19
- User Interface 20

Modem-to-Modem Communication 26
- Modulation 26
  - Frequency Shift Keying (FSK) 28
  - Phase Shift Keying (PSK) 29
  - Amplitude Modulation (AM) 30
  - Quadrature Amplitude Modulation (QAM) 31
  - Trellis Coded Modulation (TCM) 32
- Modulation Standards 33

Modem Call Analysis 34
- Call Setup 35
  - Phase I: Network Interaction 36
  - Phase II: Probing/Ranging 38
  - Phase III: Equalizer and Echo Canceller Training 40
  - Phase IV: Final Training 41
- Data Mode 42
  - Retrains and Speedshifts 42
  - Error Control 45
  - Data Compression 48
- Call Disconnect 49

Summary 51

Chapter 2 How Fax Works 53

A Brief History of Fax 54

Fax Components 56

Group Classifications 57

Specifications and Standards 58

Fax Modulations 59

Fax Messaging 61
Phases of a Fax Call 62
Message Format Overview 63
Analyzing a Basic Fax Call 65
CNG Tone 66
CED Tone 67
DIS, NSF, and CSI Messages 68
DCS and TSI Messages 71
TCF, CFR, and FTT Messages 73
MPS, EOP, EOM, MCF, RTP, RTN, and DCN Messages 75
Other T.30 Messages 77
Understanding Error Correction Mode 81
ECM Call Analysis 82
PPS and PPR 84
Important G3 Timers 86
Super G3 Faxing 88
Comparison of SG3 and G3 89
Super G3 Call Analysis 89

Page Encoding 91
Modified Huffman 92
Modified READ 97
Modified Modified READ 103

Summary 105

Chapter 3 How Text Telephony Works 107

A Brief History of Text Telephony 107

Text Telephone Terminology 110

Standards and Specifications 110
Carrier Based Versus Carrierless Protocols 111
ITU-T Recommendation V.18 112

Text Telephone Operation 112
Acoustic Coupling Versus Direct Connections 113
Originating and Receiving Text Telephone Calls 114
Conversation Conventions 116
Text Telephone Relay Services 118
HCO (Hearing Carry Over) 119
VCO (Voice Carry Over) 120

Baudot Protocol 121
Baudot Character Set 121
Baudot Modulation Details 123

Summary 124

Part II IP Solutions and Design 127

Chapter 4 Passthrough 129

Passthrough Fundamentals 130

NSE-Based Passthrough 137
Fax Passthrough with NSE 139
Modem Passthrough with NSE 141

Protocol-Based Pass-Through for Fax 143
Fax Pass-Through with H.323 Signaling 144
Fax Pass-Through with SIP Signaling 145

Text over G.711 146

A Future Look at ITU-T V.152 147

Summary 148

Chapter 5 Relay 151

Relay Fundamentals 151

Fax Relay 154
T.38 Fax Relay 155
NSE-Based Switchover for T.38 167
Protocol-Based Switchover for T.38 169
Cisco Fax Relay 173

Modem Relay 175

Cisco Text Relay 181

A Future Look at ITU-T T.38, V.150.1, and V.151 185

Summary 185

Chapter 6 T.37 Store-and-Forward Fax 189

Overview of T.37 Store-and-Forward Fax 189

SMTP Overview 191
SMTP Commands and Sample Sessions 192
DSN and MDN 195

T.37 Onramp 201

T.37 Offramp 203

Summary 204

Chapter 7 Design Guide for Fax, Modem, and Text 207

General Passthrough and Relay Design Considerations 208
Bandwidth 209
Call Control Protocol 214
QoS 215
Redundancy 221
Resource Utilization 224
Secure RTP 227
Timing and Synchronization 229

Fax Design Considerations 231
Gateway Interoperability Considerations 231
Error Correction Mode 233
Super G3 235
Hairpin Calls 237
Fallback 239
T.37 Store-and-Forward Fax 241
Fax Detect Script 243
Unified CM Integration 245
Comparing Fax Passthrough and Fax Relay 249

Modem Design Considerations 251
Comparing Modem Passthrough and Cisco Modem Relay 252
Secure Modem Relay 254

Text Design Considerations 256

Summary and Best Practices 258

Chapter 8 Fax Servers 263

Fax Server Basics 264

Fax Server Integration Solutions 269
Fax Server TDM Integration with a Cisco Voice Gateway 269
Fax Server T.38 Integration with a Cisco Voice Gateway 272
Fax Server T.38 Integration with Unified CM 276

Fax Server Redundancy and Failover 281

Summary 283

Part III Configuration 285

Chapter 9 Configuring Passthrough 287

IOS Gateway Passthrough Configuration 288
IOS Gateway NSE-Based Passthrough Configuration 289
IOS Gateway NSE-Based Passthrough Configuration for H.323, SIP, and SCCP 289
IOS Gateway NSE-Based Passthrough Configuration for MGCP 292
IOS Gateway Protocol-Based Pass-Through Configuration 293
IOS Gateway Text over G.711 Configuration 295

6608 Catalyst Blade Passthrough Configuration 295

VG248 Passthrough Configuration 298

ATA Passthrough Configuration 303

Summary 308

Chapter 10 Configuring Relay 311

IOS Gateway Relay Configuration 311
Fax Relay 312
IOS Gateway Fax Relay Configuration for H.323, SIP, and SCCP 313
IOS Gateway Fax Relay Configuration for MGCP 320
Modem Relay 325
IOS Gateway Cisco Modem Relay Configuration for H.323, SIP, and SCCP 326
IOS Gateway Cisco Modem Relay Configuration for MGCP 329
Cisco Text Relay 332
IOS Example Configurations for Relay 334
Default Fax Relay Configuration for H.323 and SIP 334
Cisco Fax Relay and Modem Passthrough Configuration for H.323 and SIP 336
T.38 Fax Relay, Cisco Modem Relay, and Cisco Text Relay Configuration for H.323 and SIP 337
T.38 Fax Relay and Cisco Text Relay Configuration for SCCP 339
T.38 Fax Relay and Modem Passthrough Configuration for MGCP 340

6608 Catalyst Blade Fax Relay Configuration 342

VG248 Fax Relay Configuration 344

Summary 347

Chapter 11 Configuring T.37 Store-and-Forward Fax 351

Enabling T.37 Store-and-Forward Fax 351

Loading the TCL Scripts 352

Configuring T.37 Onramp Fax 354
- Dial-Peer Configuration for Onramp Fax 355
- Fax Receive Configuration Command for Onramp Fax 360
- MTA Configuration Commands for Onramp Fax 361
- Sample Onramp Configuration 365

Configuring T.37 Offramp Fax 367
- Dial-Peer Configuration for Offramp Fax 367
- Fax Send Configuration Commands for Offramp Fax 369
- MTA Configuration Commands for Offramp Fax 372
- Sample Offramp Configuration 373

Summary 375

Part IV Troubleshooting 377

Chapter 12 Troubleshooting Passthrough and Relay 379

Attacking the Problem 380

Fundamental Troubleshooting 382
- Checking the Condition of Originating and Terminating Devices 383
- Testing with Voice Calls 384
- Testing with PSTN Calls 385
- Confirming the Configuration 386
- Debugging Best Practices 387

Telephony and IP Troubleshooting 391
- Call Legs in IOS Gateways 392
- Viewing Call Legs 394
  - Modem Passthrough Call Legs 394
  - Fax Pass-Through Call Legs 399
  - Fax Relay Call Legs 400
  - Cisco Modem Relay Call Legs 402
  - Text Telephony Call Legs 404
  - Call Leg Troubleshooting Techniques 405

Telephony Troubleshooting 407
IP Troubleshooting 414
IP Troubleshooting for IOS Gateways 416
IP Troubleshooting for Non-IOS Gateways 419
IP Troubleshooting Using Packet Captures 424

Troubleshooting the Switchover Signaling 428
Troubleshooting NSE-Based Switchovers 430
NSE-Based Switchover for Modem Passthrough 430
NSE-Based Switchover for Cisco Modem Relay 434
NSE-Based Switchover for T.38 Fax Relay 436
Validating NSE Switchover Support 438
Troubleshooting Protocol-Based Switchovers 445
Protocol-Based Fax Pass-Through and T.38 Switchovers for H.323 446
Protocol-Based Fax Pass-Through and T.38 Switchovers for SIP 451
Protocol-Based T.38 Switchover for MGCP 455
Protocol-Based Switchovers and Unified CM 459
Troubleshooting the Cisco Fax Relay Switchover 461

Passthrough and Relay Troubleshooting 464
Troubleshooting DSP Functions 464
DSP HPI Troubleshooting 465
Loss Planning 478
Advanced Troubleshooting for Passthrough 485
Advanced Troubleshooting for Fax Relay 487
Fax Relay Data Rate 487
Dealing with Packet Loss 488
SG3 490
Debugging T.30 Fax Messaging 491
Analyzing T.38 Fax Relay Packet Captures 497
NSF/NSS 499
Handling High Delay 500
Advanced Troubleshooting for Modem Relay 503
Checking the Modem Endpoints 503
Debugging Modem Relay 505
Advanced Troubleshooting for Cisco Text Relay 506
PCM Traces for Fax and Modem 510
Capturing PCM Traces 511
Analyzing PCM Traces 515

Summary 523

Chapter 13 Troubleshooting T.37 Store-and-Forward Fax 525

Checking the Basics 525

T.37 Onramp Troubleshooting 527
Troubleshooting the Onramp Telephony Interface 532
Troubleshooting the TIFF Image Creation 537
Troubleshooting the Onramp SMTP Connection 540

T.37 Offramp Troubleshooting 545
Troubleshooting the Offramp SMTP Connection 549
Troubleshooting the Creation of the Fax Page Image 553
Troubleshooting the Offramp Telephony Interface 556

Summary 559

Index 562

Icons Used in This Book

Command Syntax Conventions

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

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

Introduction

The advent of VoIP has led to revolutionary changes in the world of telecommunications. Information that was transported on traditional telephony infrastructures such as voice, video, and modulated data is transitioning to IP backbones. However, in this transition process, modulated data such as fax, modem, and text is often overlooked. Fax, modem, and text are treated like regular voice communications in many cases when in fact they have different transport requirements and usually need unique transport protocols for communication to be reliable.

We, the authors of this book, have about 25 years of combined networking experience with the majority of it focusing on faxes, modems, and VoIP. We have seen and experienced firsthand as Cisco TAC engineers the problems that are encountered with fax and modem communications. While one of the most common problems we encounter is the failure to take into account the unique transport requirements of fax, modem, and text, we also have seen problems with the configuration of the multitude of fax-, modem-, and text-related commands in Cisco voice gateways. In addition, we have realized that many times there is just a lack in understanding of basic passthrough and relay fundamentals as they are implemented on Cisco voice products. Addressing these problems and how to troubleshoot them were our main focus while writing this book.

Therefore, you will notice that this book includes a comprehensive design guide for getting fax, modem, and text deployments working successfully from the start, a commonsense configuration section, and a thorough troubleshooting guide. Equally as important, we devoted a whole section to the fundamentals of passthrough and relay and how they are implemented on Cisco voice products. In this book, we address all the main difficulties that we have seen with the implementation of fax and modems in IP environments.

We have written this book to be the definitive resource for understanding, designing, configuring, and troubleshooting fax, modem, and text in today's IP networks. Whether you are a network designer, voice engineer, or simply someone who must support fax, modem, and text communications over IP networks, this book is practically a necessity. If you understand basic VoIP, this book will just build upon that core knowledge.

Many books and other resources are available that discuss VoIP, and some even have a casual mention of transporting fax or modem communications. However, this book is the only one that provides a comprehensive, one-stop reference for addressing all aspects of fax, modem, and text communication.

Target Release: Cisco IOS Software Version 12.4(9)T1

The examples and features explained throughout this book for Cisco IOS voice gateways target Cisco IOS Software Release 12.4(9)T1. However, other IOS versions should be applicable to the majority of this book, too. Be aware, however, that features and implementations might differ somewhat in other IOS versions. Other software versions for devices such as Cisco Unified Communications Manager, 6608, and the VG248 are noted in the text when applicable.

Goals and Methods

This book is designed to be the only resource you will ever need for handling fax, modem, and text communications in IP telephony environments. From basic theory to design solutions to configuration to troubleshooting, all aspects are covered in a clear, concise manner.

Who Should Read This Book?

Just about every IP telephony (IPT) installation has at least one fax machine, and larger installations often include modems and text telephony devices, too. If you work with IPT, your job has already required or more than likely will require in the future that you handle fax, modem, and text communications in your network. For this reason, this book is an indispensable resource that should reside beside your other books dealing with IPT.

In some areas, this book expects you to have basic IPT knowledge. You should be familiar with the Internet Protocol, possess a good grasp of voice fundamentals, and be familiar with at least one of the various call control protocols. If you work with IPT on a consistent basis, you probably already have this knowledge.

Because of this book's comprehensive coverage of fax, modem, and text, it contains relevant information for a wide variety of readers who work with IPT. For anyone who works in IPT network design, such as design engineers, network architects, or systems engineers, this book features a comprehensive design and planning section. If you deploy and install IPT networks, an easy-to-understand configuration section provides the pertinent commands and sample configurations necessary for successfully transporting fax, modem, and text communications. Lastly, for those who support IPT networks, such as customer support engineers, field engineers, network administrators, and escalation engineers, a detailed troubleshooting section equips you with the knowledge and techniques to handle any issue that arises.

If you work with IPT, you will encounter fax, modem, and text devices if you have not already. These devices have special requirements and protocols that must be addressed for successful IP integration and deployment. When it comes time to handle fax, modem, and text communications as part of your job in IPT, this is the one resource that you want by your side.

How This Book Is Organized

This book is logically laid out with critical, fundamental concepts defined at the beginning in Chapters 1 to 6. Later chapters build upon these concepts to assist you with network design, configuration, and troubleshooting. Once the initial fundamental chapters are covered in the first two sections, the remaining chapters do not have to be read in any particular order even though the listed chapter sequence is what we believe to be the most beneficial for learning the subject matter.

The chapters in this book are divided into the following sections and cover the following topics:

- **Part I Laying the Groundwork**

  Provides the fundamentals of how faxes, modems, and text telephony devices work.

  — **Chapter 1, "How Modems Work"**—Discusses modem architecture, different modem types, and the methods and modulations used by modems for communication. In addition, a basic modem call is analyzed, including the negotiation phases and data mode.

  — **Chapter 2, "How Fax Works"**—Covers the core elements of fax technology, including the common group classifications and standards, an in-depth section on fax messaging, and page encoding.

  — **Chapter 3, "How Text Telephony Works"**—Provides an introductory look at text telephony and its fundamantals. Basic text telephony operation and concepts are covered along with a technical discussion of the Baudot text telephone protocol.

- **Part II IP Solutions and Design**

  Describes the various switchover methods and transport options that are used to handle fax, modem, and text communications. Design chapters then help you determine the best solution for transporting your fax, modem, and text traffic.

  — **Chapter 4, "Passthrough"**—Shows you the fundamental methods and principles necessary for using a voice codec for transporting fax, modem, and text. The different passthrough methods on Cisco voice gateways and their various switchovers are also discussed.

  — **Chapter 5, "Relay"**—Details the intricacies of relay operation and its various transport methods and switchover types for fax, modem, and text.

  — **Chapter 6, "T.37 Store-and-Forward Fax"**—Demonstrates the workings and fundamentals of fax and e-mail integration using onramp and offramp faxing.

  — **Chapter 7, "Design Guide for Fax, Modem, and Text"**—Provides pertinent design information and best practices for integrating fax, modem, and text telephony into your IP network.

  — **Chapter 8, "Fax Servers"**—Concentrates on the design and planning aspects of integrating fax servers into your network. In addition to fax server benefits and integration models, fax server–specific configuration and troubleshooting information is also provided.

- **Part III Configuration**

  Details the configuration tasks for a variety of Cisco products that are essential for transporting fax, modem, and text successfully.

  — **Chapter 9, "Configuring Passthrough"**—Provides the configuration commands for enabling passthrough and its various features on Cisco products.

  — **Chapter 10, "Configuring Relay"**—Illustrates the numerous commands for successfully configuring the different relay transport methods and features on Cisco products. Also included are IOS voice gateways sample configurations of common deployment scenarios.

  — **Chapter 11, "Configuring T.37 Store-and-Forward Fax"**—Breaks down the somewhat confusing T.37 store-and-forward fax configuration process for onramp and offramp into simplified steps. Within each configuration step, the applicable commands are shown.

- **Part IV Troubleshooting**

  Discusses the troubleshooting techniques and procedures used by Cisco TAC engineers for resolving fax, modem, and text issues.

  — **Chapter 12**, **"Troubleshooting Passthrough and Relay"**—Details a fax, modem, and text troubleshooting methodology that efficiently resolves passthrough and relay problems. Each step of this troubleshooting methodology correlates directly to a section within the chapter that shows you the key commands, debugs, and troubleshooting steps to execute for rapidly resolving issues from the most basic to the complex.

  — **Chapter 13**, **"Troubleshooting T.37 Store-and-Forward Fax"**—Highlights graphical troubleshooting models for onramp and offramp faxing that allow you to zero in on problems quickly. In-depth debugging techniques and procedures for the different processes within the graphical model are also provided.

Comments for the Authors

The authors are interested in your comments and suggestions about this book. Please send feedback to the following e-mail address:

faxmodemtextbook@external.cisco.com

Further Reading

The authors recommend the following resources for more information.

Cisco.com

The Cisco website is one of the best resources for additional documents related to fax, modem, and text technologies and IP telephony in general. Usually the easiest way to find a document is to use the web page's search feature. Other useful links on Cisco.com include the following:

- For design related documents, see http://www.cisco.com/go/srnd.
- For Unified Communications product information, refer to http://www.cisco.com/go/unified.
- For a listing of support information links, including command references, design and troubleshooting documents, and configuration guides, go to http://www.cisco.com/go/support.

The following technical books are also recommended for supplementing the information in this book and for increasing your overall IP telephony knowledge. These books can be examined at a local technical bookseller or by entering the title in the search box at http://www.ciscopress.com.

Voice over IP Fundamentals, Second Edition

The book *Voice over IP Fundamentals* (ISBN 1-58705-257-1) is a good place to start for those making a move into the IP telephony world, and it is also a handy reference for those already familiar with VoIP.

Troubleshooting Cisco IP Telephony

You can find comprehensive troubleshooting information for all the major components of a Unified Communications network in the book *Troubleshooting Cisco IP Telephony* (ISBN 1-58705-075-7).

PART I

Laying the Groundwork

Chapter 1 How Modems Work

Chapter 2 How Fax Works

Chapter 3 How Text Telephony Works

CHAPTER 1

How Modems Work

Although analog modem technology stood on its own for many years in public switched telephone network (PSTN) environments, the rapid evolution of IP telephony (IPT) is now requiring that modem communications work successfully over IP networks. However, before discussing this complicated convergence of modems and IP, it is important to first attain a solid foundation in basic analog modem operation and communication.

This chapter addresses the basics of analog modem technology and prepares you for working with modems in IP networks. Specifically, this chapter covers the following topics:

- **A Brief History of Modems**: Highlights important developments and achievements since the modem's inception
- **Architecture of a Modem**: Details important modem components
- **Modem Types**: Covers different modem classifications and highlights important differences
- **Terminal-to-Modem Communication**: Discusses DTE/DCE interaction, RS-232, asynchronous framing, and the modem user interface
- **Modem-to-Modem Communication**: Illustrates the concepts of modulation and the various schemes that are used
- **Modem Call Analysis**: Provides a detailed analysis of all phases of a modem call

This chapter aims to be as comprehensive as possible, but because of the complicated nature of the topic in conjunction with the large number of specifications addressing modem operation, only the most important aspects of modem technology as it relates to IPT are covered.

A Brief History of Modems

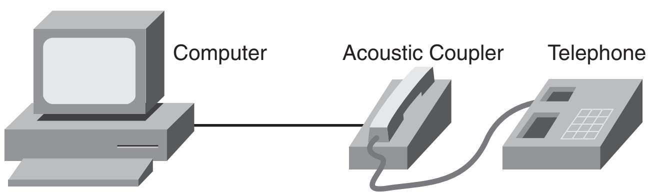
Like several other core Internet and computer technologies, the modem was first developed in the 1950s for the Semi-Automatic Ground Environment (SAGE) air defense system. The modems were used to transmit military data over dedicated telephone lines between terminals at the various participating sites.

As commercial computer use increased, so did the need for communications between them. AT&T manufactured the first commercial modem, known as the Bell 103, in 1962. The Bell 103 allowed full-duplex transmission and employed Frequency Shift Keying (FSK) modulation with a data rate of up to 300 bits per second.

NOTE The section "Modulation," later in this chapter, covers FSK and the other modulation schemes in detail.

Early modems by law were not allowed to connect directly to the telephone network. Usually they had an integrated acoustic coupler that allowed for a standard telephone handset to rest on the microphone/speaker cradle to convert between audio signals and digital data. Figure 1-1 illustrates an acoustic coupler connection. A major drawback is that the remote telephone number must be manually dialed before the handset is placed into the acoustic coupler for the modem training and connect sequence.

Figure 1-1 *Acoustically Coupled Modem*



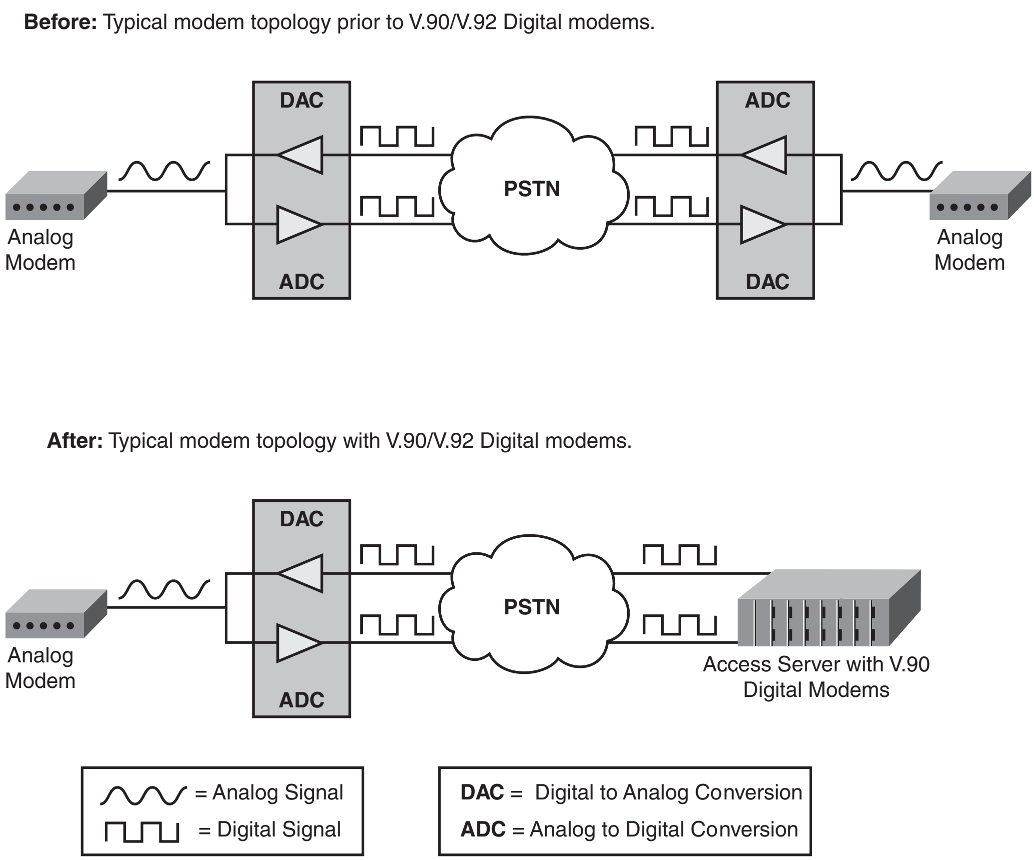
A landmark event in modem development was the introduction of the Hayes command set in 1977. Developed by Hayes Microcomputer Products for their Smartmodem product, this set of machine instructions allows the computer to control the modem's functions. Due to its popularity, the Hayes command set became the de facto standard, and most manufacturers still support it or one of its variants today. This development, along with changes in the telecommunication laws that allowed direct connection and dialing to the PSTN, spurred enormous growth in the modem industry.

Throughout the 1970s and 1980s, there were continuous improvements in the data rates of modems. These advancements were largely due to more sophisticated modulation techniques, improvements in telephony infrastructure, introduction of echo canceling methods, and the integration of error correction and data compression algorithms. A culmination of these advances was the release of the V.34 specification in 1994 by the international standards body known as the International Telecommunications Union Standardization Sector (ITU-T). You can find all the pertinent ITU-T Recommendations that are mentioned throughout this book at http://www.itu.int/ITU-T/.

The maximum speed of a V.34 connection is 33.6 Kbps. Unlike older specifications, V.34 employs multiple modulation schemes and multiple impairment compensation techniques to robustly adapt to poor line quality.

Despite the fact that it was thought that V.34 rates achieved the maximum throughput possible for a telephone line, it was only a few years later that 56-Kbps-capable modems became available. By taking advantage of pulse code modulation (PCM) and reducing the number of analog-to-digital (A/D) conversions from two to one, a data rate of 56 Kbps was achieved. Figure 1-2 shows how the typical modem topology has changed with the advent of 56K modem technology. Only a single A/D conversion is necessary because the central/ISP (Internet service provider) side modem is digitally terminated, typically with a digital T1 or E1 connected to an access server with onboard modems.

Figure 1-2 *Analog and Digital Modem Topologies*



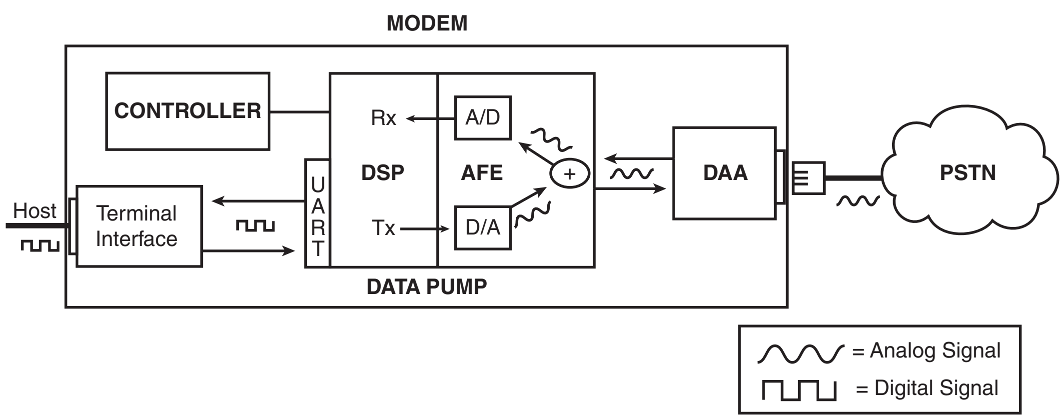
K56Flex, developed by Lucent and Rockwell, and X2, created by US Robotics (now known as 3Com), were the two early releases of the 56K protocol. These two protocols did not interoperate and this caused many problems and added unnecessary overhead. Consequently, V.90 was released as the international 56K modem standard for interoperability between different vendors. V.90 allows for data signaling rates of 56 Kbps downstream speed and 33.6 Kbps (V.34) upstream

The most recent widespread improvement to modem performance was the release of the V.92 standard. The ITU-T Recommendation V.92 made several minor incremental improvements over V.90. These included a drastic decrease in the amount of time it takes a modem to train and slight throughput upgrades by using an improved data compression scheme (V.44). In addition, V.92 provided advanced new features such as modem-on-hold (MoH). This feature allows an Internet user to suspend his data connection and accept an incoming voice call. Upon completion of the voice call, the data connection can be resumed.

Modem Architecture

Modems allow for communication between computers in much the same way a telephone allows for communication between humans. Fundamentally, an analog modem converts the digital signals from a computer to analog signals that are transmitted over voice-grade access to the PSTN. Figure 1-3 shows a high-level view of the architecture of an analog modem.

Figure 1-3 *Analog Modem Architecture*



Modem architecture can generally be broken down into four main functional units:

1. **Data pump**: Responsible for carrying out the two primary functions of a modem (that is, the ones that give it its name):
   - **mo**dulation: Conversion of the digital bit stream received from the terminal into an analog signal that is sent over the telephone line.
   - **dem**odulation: Conversion of the analog waveform received over the telephone line into binary data that is sent to the terminal.

   Therefore, the data pump is commonly viewed as the engine of the modem. A typical data pump is comprised of two main functional subunits:

   — **Analog front end (AFE)**: Comprised principally of analog-to-digital (A/D) converters and digital-to-analog (D/A) converters. The A/D converters convert the voltages on the phone line to discrete binary values for the digital signal processor to process. Likewise, the D/A converters convert the binary data from the DSP and smoothes the output to form an analog signal.

   — **Digital signal processor (DSP)**: A specialized processor that is optimized for various signal-processing functions. Most important, it executes in real time the mathematically intensive operations involved in executing the different modulation/demodulation algorithms for the various modem protocols. It also handles echo cancellation, tone generation, and other specialized functions.
2. **Controller**: Handles the command interface to the terminal, AT command interpretation and execution, performs error correction and data compression algorithms, handles flow control between terminal and the data pump, and various other supervisory and miscellaneous functions. In this context, the controller is often referred to as the CPU of the modem.
3. **Data access arrangement (DAA)**: Contains the analog circuitry that electrically isolates the modem from the telephone network and also provides the physical interface (that is, line impedance, hybrid circuitry, and so on) to connect to a plain old telephone system (POTS) line.
4. **Terminal interface**: The asynchronous serial interface between the modem and the terminal. The section "Terminal to Modem Communication," later in this chapter, covers the transmission protocol (RS232) and asynchronous character framing that occurs on this connection in greater detail.

This is obviously a broad and simplified overview, and many modems exist that have variations on this general architecture, depending on the amount of integration and the type of modem that they are. For example, some modems have no controller at all, whereas others can have more than one processor. Primary function, modem type, and particular manufacturer largely dictate these various hybrid schemes.

Modem Types

Several methods of classifying modems exist based on the way they are connected to a computer, their general architecture, and their capabilities. The sections that follow address the most common classifications along with their pros and cons.

External Versus Internal Modems

An external modem physically resides outside the computer and has its own chassis, power supply, front-panel indicator LEDs, and so on. Also, it is connected to the computer with a cable that generally connects to the serial interface on a COM port. Internal modems reside in the computer, typically in a PCI or ISA slot, and usually create a virtual COM port. Table 1-1 provides a quick comparison of external versus internal modems.

Table 1-1 *External and Internal Modem Comparison*

Modem Type	Advantages
External	Easier to troubleshoot
	Viewable modem status LEDs
	Ease of installation
	Richer feature set
Internal	Less expensive
	Integrated with computer system for added mobility and convenience

Hardware Versus Software Modems

The architecture of a modem, discussed earlier, determines whether a modem is classified as a hardware modem or a software modem. A hardware modem is one that has hardware that handles all the data pump and controller functions on its own. On the other hand, a software modem is one that offloads one or both of those responsibilities to the host computer.

NOTE Although "WinModem" is a USR brand of modem, it has become a popular term used to refer to software modems.

The two major types of software modems are as follows:

- **Controllerless modems**: A controllerless modem does not have its own on-board controller hardware. Rather, it offloads the controller's function to the computer's processor. The controllerless modem does have its own DSP hardware carrying out the data pump functions, which are generally the most processor intensive.
- **Host signal processor (HSP) modems**: HSP modems have neither controller nor data pump hardware of their own. Instead, they run software that offloads both of those functions to the host computer's CPU. From a hardware standpoint, an HSP modem is not much more than a DAA, because all of its other functions are carried out entirely in software. This is why it is commonly referred to as a softmodem. Handling these processor-intensive tasks takes away from the computer's processor and memory resources and could potentially cause a noticeable degradation in performance, especially on slower computers.

NOTE External modems are always hardware modems. However, internal modems can be either hardware or software, although recently, most internal modems are software based.

Table 1-2 lists some of the advantages and disadvantages of hardware modems versus software modems.

Table 1-2 *Hardware and Software Modem Comparison*

Modem Type	Advantages
Hardware	Contains own processing resources, so host computer's performance is not degraded
	Typically more robust connections and better performance
	Generally better compatibility with different operating systems
Software	Less expensive
	Easier to upgrade firmware
	Smaller and easier to integrate into laptop computers

Fax Modems

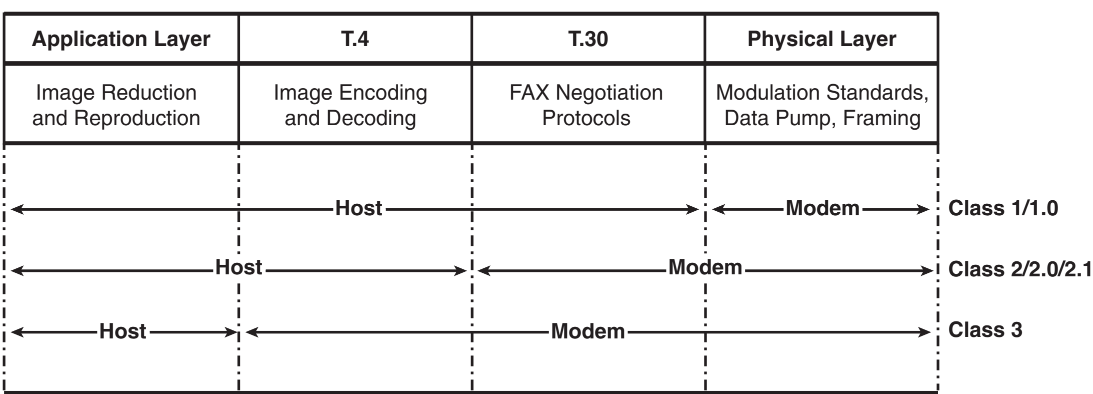
Another type of modem relevant to the discussions in this book is the fax modem. This type of modem is nothing more than a modem that runs software that enables it to transmit documents to a fax machine or another fax modem. Most modems sold since the early 1990s contain fax modem functionality. Fax modems, like regular modems, can be either internal or external.

Fax modems have become popular because of certain advantages they offer over regular fax machines. One advantage is that fax modems are less expensive and require less maintenance.

Another is the convenience of directly sending documents in electronic format without the need to print them out. In addition, maintaining the document in electronic form ensures consistent image quality and efficient storage of the fax pages.

A series of standards, known as fax classes, were developed to differentiate and define the responsibilities of the computer versus those of the fax modem. Figure 1-4 shows how those responsibilities vary with class designation.

Figure 1-4 *Modem Versus Host Responsibilities for Different Fax Classes*



NOTE The class of a modem only defines the way in which the computer's fax software controls/interfaces with the fax modem. It has nothing to do with negotiation between two fax modems or a fax modem and a fax machine.

Table 1-3 defines the classes for determining a modem's ability to conduct a fax session.

Table 1-3 *Fax Class Designations for Modems*

Fax Modem Type	Standard	Description
Class 0	N/A	The modem has no fax capabilities and functions only in data mode.
Class 1	EIA/TIA-578 and ITU-T T.31	The computer fax software manages virtually the entire fax session. It is responsible for the fax negotiation (T.30 protocol), and the image encoding/decoding (T.4 protocol). The modem, on the other hand, provides the minimum services for a fax session. It is responsible for modulation/demodulation, fax command/response interface, and the conversion from the asynchronous data from the computer to the synchronous High-Level Data Link Control (HDLC) packets required for fax communication.
Class 1.0	ITU-T T.31 Annex B	Much like Class 1, but with V.34-fax (Super G3) capability added.
Class 2	EIA/TIA SP-2388A (now obsolete)	The modem has more intelligence regarding the fax session than Class 1. In this case, the modem handles much of the fax negotiation (T.30 protocol), whereas the computer fax application deals with the image generation and page data (T.4 protocol). The Class 2 standard was in draft status for a long time. Therefore, modem manufacturers made modems that adhered to this draft rather than the final ratified standard. Thus, the Class 2 standard is now obsolete, but is still supported by various vendors.
Class 2.0	EIA/TIA-592 and ITU-T T.32	Modems adhering to the first Class 2 draft are said to be Class 2 compliant, and those adhering to the final approved standard are said to be Class 2.0 compliant. There were improvements between Class 2 and Class 2.0, such as implementing Error Correction Mode (ECM) support on the modem, resolving flow-control problems, and fixing data underrun/overrun issues.

continues

Table 1-3 *Fax Class Designations for Modems (Continued)*

Fax Modem Type	Standard	Description
Class 2.1	ITU-T T.32	Similar to Class 2.0, but with V.34-fax (Super G3) capability added. This is defined in Annex C of specification T.32
Class 3	N/A	The computer fax software offloads even more of the faxing responsibilities to the modem. For this class, the modem handles the bulk of both the fax negotiation (that is, T.30 protocol) and the image data conversion (that is, T.4 protocol) responsibilities. Class 3 is not an official standard yet, so it is not commonly seen in practice.

NOTE T.30 and T.4 fax protocols are discussed in great detail when fax is covered in Chapter 2, "How Fax Works."

Table 1-4 highlights some of the advantages of Class 1 and Class 2 fax modems.

Table 1-4 *Class 1 and Class 2 Fax Modem Comparison*

Fax Modem Type	Advantage
Class 1/1.0	Provides greater flexibility because there is no need to upgrade the modem firmware or wait for modem manufacturer to support a new feature because faxing is done almost wholly by the computer software.
Class 2/2.0/2.1	Because the modem does most of the T.30 fax negotiation, this relieves the host computer of processing resources that can be used for something else. This could be beneficial for slow or overtaxed systems.

Although many vendors support all these variants of Class 2, there is no guarantee of compatibility. Also, Class 2 is a closed standard, so any changes to T.30 would require a modem firmware upgrade.

Terminal-to-Modem Communication

This section deals with the protocols typically used on the asynchronous serial link between the host and the modem. First, you are introduced to the concept of data terminal equipment (DTE) and data circuit-terminating equipment (DCE). Then, the communications link between terminal and modem is divided into three layers. From the bottom up, they are as follows:

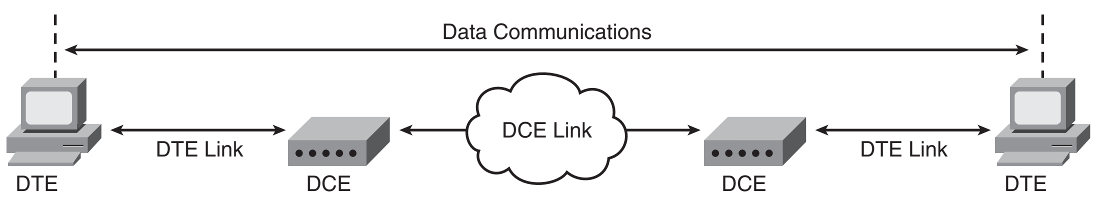
- **RS-232 physical layer**: This defines the mechanical, electrical, and hardware signaling used on the terminal-to-modem cable.
- **Async framing layer**: This specifies the format used to frame characters on an asynchronous serial link.
- **AT command layer**: This is a command language used by the host to configure and control the modem.

DTE and DCE

Various international standards bodies agreed on specifications that detail how to facilitate the connection of data communications equipment. These standards discuss the interface between DTE and DCE. The specifications describe the physical and electrical interface between a DTE and a particular type of DCE. As an example, the ITU-T V-series recommendations deal with the connection of a DTE to a modem (the DCE).

DTE is equipment that acts as a data source/sink from the point of view that it converts user information into signals to be transmitted by the DCE. The most frequently used example of a DTE is a computer. Correspondingly, DCE is the equipment that establishes and provides access to a communications link over a channel connecting the source and destination DTEs. Therefore, a DCE provides a data link service for DTEs to communicate over. In this chapter, the DCE will always be a modem. Figure 1-5 shows the logical location and function of both DTE and DCE. The practical significance in distinguishing between these two types of equipment is that they are pinned and cabled differently.

Figure 1-5 *DTE and DCE Topology*



RS-232 Signaling

RS-232 is a serial transmission system designed to support communications for short distances between a DTE and a low-speed DCE. It has evolved through several generations of standards (EIA-232C, EIA-232D, EIA-232E, and a variant has separately been standardized by the ITU as V.24). RS-232 supports a variety of applications, including synchronous and asynchronous transmission. This discussion focuses on full-duplex async DTE links to contemporary modems and uses the term RS-232 in its generic sense; for more precise details, consult the standards.

A standard RS-232 link will use the DB-25 connector. Normally (but not necessarily), the DTE port is male, and the DCE port is female. PC DTE ports often use a DB-9 connector, whereas Cisco normally uses a nonstandard 8-pin modular (RJ-45) connector for its async ports. Table 1-5 summarizes the pinouts for all three of these interface types. (Pinouts are from the plug side. Jack side pinouts are rolled.)

NOTE Technically PC DTE ports use a DE-9 connector. The misnomer "DB-9" is not a connector that exists in practice, but it is mistakenly used so frequently that it has become the de facto term for a PC DTE interface. Consequently, this book will henceforth use the commonly used DB-9 nomenclature when referring to the connector of a PC DTE port.

Table 1-5 *Pinouts for Different RS-232 Interfaces*

DB-25	DB-9	RJ-45	Name	From	Description
1			GND	gnd	Protective (shield) Ground
7	5	4, 5	SG	gnd	Signal Ground
2	3	6	TxD	DTE	Transmitted Data
3	2	3	RxD	DCE	Received Data
4	7	8	RTS	DTE	Request to Send (hw flow control)
5	8	1	CTS	DCE	Clear to Send (hw flow control)
6	6	2	DSR	DCE	Data Set Ready (DCE ready)
20	4	7	DTR	DTE	Data Terminal Ready
22	9		RI	DCE	Ring Indicator
8	1	(2)	CD	DCE	Data Carrier Detect
21			RL	DCE	Remote Loop / sig quality
23			CH/CI	DTE/DCE	Signal Rate Selector
24			DA	DTE	DTE Tx Timing
15			DB	DCE	DCE Tx Timing
17			DD	DCE	Rx Timing
14			SBA	DTE	Secondary TxD
16			SBB	DCE	Secondary RxD

Table 1-5 *Pinouts for Different RS-232 Interfaces (Continued)*

DB-25	DB-9	RJ-45	Name	From	Description
19			SCA	DTE	Secondary RTS
13			SCB	DCE	Secondary CTS
12			SCF	DCE	Secondary DCD

Not all 25 conductors are used; for async applications, typically from 3 to 9 conductors will be used, depending on whether hardware (hw) flow control or modem control signaling is required.

RS-232 does not specify bit rates per se. However, for async transmission, the following rates have been typically seen: 50, 75, 110, 134.5, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200, 230400 bps. As specified in the standards, RS-232 is officially considered to be suitable only for data rates of up to 20 Kbps and distances of up to 50 feet. In practice, RS-232 is often run at 115200 bps for distances of up to 20 feet, and at 9600 bps for distances of as much as 500 feet.

The RS-232 protocol defines nine electrical circuits to handle all the handshaking between a DTE and DCE. These electrical circuits (also referred to as leads, pins, or signals) are grouped into three categories: data interchange circuits, control interchange circuits, and the ground circuit.

Data leads are used to signal the exchange of data. Control leads govern the call signaling states between a DTE and a DCE and manage the flow control between them. As its name suggests, the ground lead is the reference ground for the DTE and the DCE. Table 1-6 details each of the nine RS-232 pins and their individual function.

NOTE Terminology, such as "raise" or "assert," with regard to the RS-232 pins implies putting a particular voltage on it. Likewise, the terms "lower" and "drop" imply changing the polarity of that voltage.

Table 1-6 *RS-232 Circuits and Their Function*

Circuit Type	Circuit Name	Circuit Function
Ground	**SG** Signal Ground	Reference Ground.
Data	**TxD** Transmit Data	Data transmitted by the DTE to the DCE.

continues

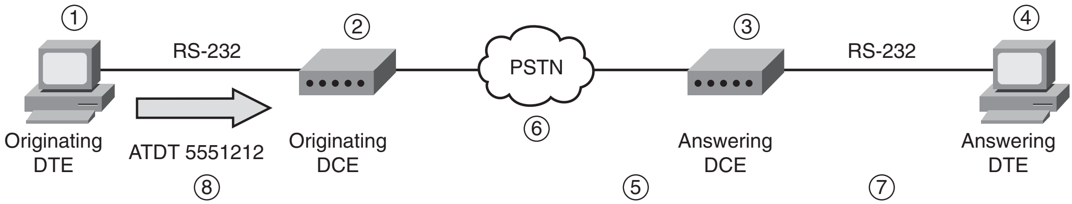
Table 1-6 *RS-232 Circuits and Their Function (Continued)*

Circuit Type	Circuit Name	Circuit Function
Data	**RxD** Receive Data	Data received by the DTE from the DCE.
Control - Modem Control	**DTR** Data Terminal Ready	Raised by the DTE when it is ready to get access to the DCE link. The modem will not dial unless it sees DTR asserted by the host.
Control - Modem Control	**DSR** Data Set Ready	Raised by the DCE when it is powered up and in such a state where the communications channel is available for transmission/reception. The DTE will not request the modem to dial unless it sees DSR high from the modem.
Control - Modem Control	**CD** Carrier Detect	Raised by the local DCE when it detects a carrier signal from the remote DCE.
Control - Modem Control	**RI** Ring Indicator	Raised by the DCE to signal to the DTE that there is an incoming call. RI is asserted in accordance with the incoming ring cadence on the phone line.
Control - Flow Control	**CTS** Clear To Send	Raised by the DCE to signal it is ready to receive data from the DTE. If the modem temporarily lowers CTS, it backpressures the DTE link.
Control - Flow Control	**RTS** Request To Send	Raised by the DTE to signal it has data to transmit to the DCE. If the host temporarily lowers RTS, it backpressures the DCE link.

Electrically, the RS-232 data interchange circuit's (for example, TxD and RxD) "mark" state (logical 1) is signaled as a voltage level less than –3V, and a "space" state (logical 0) is signaled as a voltage level greater than +3V. For control interchange circuits, an OFF state is signaled as a voltage level less than –3V, and an ON state is signaled as a voltage level greater than +3V. The signal ground lead must be connected to the equipment on each side of the link to provide a voltage reference.

Now that you know the definitions of all the RS-232 circuits, Figure 1-6 puts them into practice by tracing through all the RS-232 signaling involved in placing a modem call from one host to another. This example illustrates the transitions of the various control pins. Note that when the call is up, the actual data transmission and reception will be signaled by the data pins (TxD, RxD).

Figure 1-6 *RS-232 Signaling for a Modem Call Setup*



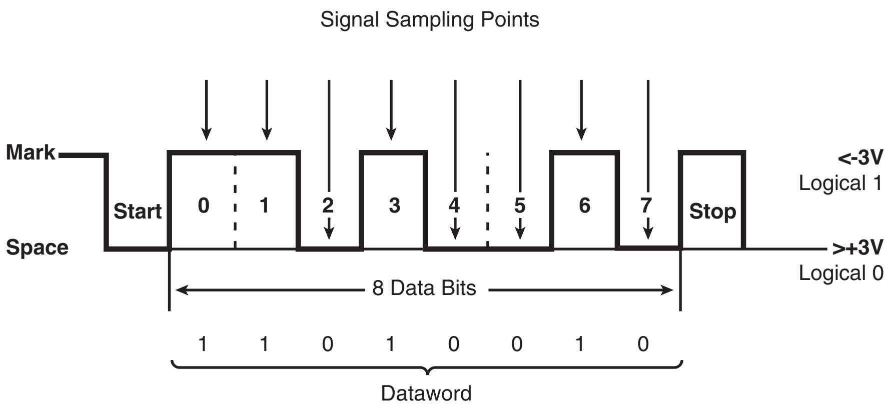
1. Originating DTE raises DTR and transmits AT dial string (DSR is high at this point)
2. Originating DCE places a call to the Answering DCE
3. Answering DCE raises RI to signal to the Answering DTE that there is an incoming call
4. Answering DTE raises DTR
5. Answering DCE goes off-hook
6. Answering DCE sends answer back tone and modems train
7. Answering DCE signals CONNECT and raises DCD
8. Originating DCE signals CONNECT and raises DCD

Asynchronous Framing

All data transmission requires that the receiver somehow synchronize with the transmitter to know when to detect symbol state changes. In synchronous framing, the receiver maintains a clock that is kept in sync with the transmitter's clock. This synchronization can be maintained either by some external hardware signal (for example, a timing circuit in sync RS-232) or by some recurring framing pattern in the received signal (for example, the framing bits in a T1 frame). In asynchronous framing, the receiver synchronizes anew with some pattern seen at the front of each frame. Examples are Ethernet (where the receiver syncs to the frame's preamble) and async character framing.

Async character framing is used on both async RS-232 links and in modem links where an error control framing protocol isn't used. In this scheme, each character (of 5, 6, 7 or 8 databits) is encapsulated in a separate frame, which is composed of a start bit (a space bit), the payload containing the databits and an optional parity bit, and 1, 1.5, or 2 stop bits (mark bits). While the async link is idle, the transmitter sends mark bits. The receiver, which must be preconfigured knowing the payload length, will synchronize on the start bit for each frame. Figure 1-7 shows an async frame and all its components.

Figure 1-7 *Asynchronous Framing*



Very old equipment might have required more than 1 stop bit, but such equipment is rarely seen now. However, a transmitter being configured for excess stop bits won't cause communications problems, only reduced payload transfer rate, as the extra stop bits are interpreted by the receiver as idle bits.

By far the most prevalent async character frame formats encountered now are these:

- 7 databits, 1 parity bit (usually even), 1 stop bit (7E1)
- 8 databits, no parity, 1 stop bit (8N1)

In both of these cases, the payload size is 8 bits, which meshes nicely with the standard byte size used on contemporary computers and in octet-oriented transmissions protocols.

With 1 start bit, 8 payload bits, and 1 stop bit, async framing has 20 percent overhead. Thus, an 115200 bps 8N1 async framed link will have a payload throughput rate of 92160 bps. Relative to sync transmission, this is rather high overhead. For RS-232 DTE links, where the DTE speed is significantly higher than the DCE rate (for example, when using an 115200 bps DTE link for a 28800 bps V.34 link), this overhead is not especially costly. However, for the relatively precious bandwidth on the DCE link, this overhead can be considered to be excessive, which is one of the motivations for using error control (EC) framing on the DCE link instead (discussed later in this chapter).

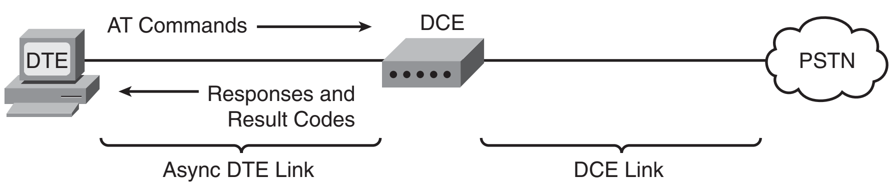
User Interface

The standard method used for an async DTE to control its DCE (for example, modem) is through a command-line interface (CLI) protocol called the AT interface. AT stands for ATtention; each command line sent by the DTE is prefixed with AT, which serves to get the CLI's attention.

As discussed earlier in the history section, the AT interface was introduced by Hayes Microcomputer Products (now a brand of Zoom Telephonics) in 1981. It has existed as an evolving de facto standard, with many vendor-specific oddities and extensions, since then. The ITU attempted to codify the standard in 1995 with V.25ter, although by then it was probably too late to impose order on the menagerie of existing command sets. Still, there is a core set of AT commands, honored by almost all modem manufacturers, that have been standardized by the V.25ter syntax.

The AT command interface is a simple CLI implemented in the DCE's controller. Figure 1-8 depicts how the CLI reads AT commands from the async DTE link, executes them as needed, and returns responses to the DTE. These responses are sent by the DCE, in the form of result codes, in reply to AT commands from the DTE and activity on the line.

Figure 1-8 *Asynchronous DTE Link Communication*



One of the main uses of the AT interface is to provide a method for supervisory and address signaling between the DTE and DCE. This includes allowing the DTE to control call setup, training, and teardown, and to allow the DCE to communicate status to the DTE. Table 1-7 shows some sample AT commands the DTE sends the DCE to establish a call. This table also highlights some typical result codes from the DCE to the DTE in response to such AT commands.

Table 1-7 *Sample AT Commands and Result Codes*

AT Command	Description
ATD*number*	Dial *number* then start training in originate mode.
ATD*number;*	Dial *number* then return to AT command mode without training.
ATDT*number*	Dial *number* using DTMF address signaling.
ATDP*number*	Dial *number* using pulse signaling.
ATDL	Redial the last number dialed.
ATA	Go offhook and begin training in answer mode.

continues

Table 1-7 *Sample AT Commands and Result Codes (Continued)*

Result Code	Description
CONNECT	Modems have trained and have gone into data mode.
CONNECT 2400	Modems have trained at 2400 bps.
CONNECT 26400/REL – MNP	Modems have trained at 26400 bps and have negotiated a reliable link with MNP error control.
RING	An incoming call is arriving from the circuit network (sent by the answer modem).
NO DIALTONE	The originate modem went offhook but did not hear dial tone.
BUSY	A busy signal was detected.
NO CARRIER	If in call setup mode, this indicates that the modems failed to train. If in data mode, this indicates that carrier was lost and that the call has disconnected.

A modem has two primary communication modes:

- **Data mode**: For data to be transferred between two hosts, the modems must be in data mode.
- **Command mode**: All the call control functions (dialing, hang up, auto answer, and so on) are handled in command mode.

Before a call is established, the DTE link is used for the AT interface; as soon as the modem sends the DTE a "CONNECT" result code with the connection speed baud rate displayed, the modem switches from command mode to data mode, and user data can begin to be transmitted between DTEs. The AT interface is inoperative in data mode.

On traditional modems, the AT interface operates in-band on the transmit/receive data path in the DTE link. This is the same path used to transmit data while in data mode. There is a significant problem associated with using in-band control: when the DTE links are in data mode, there is no guaranteed method to distinguish in-band signals from user data. Thus, if available, an out-of-band signaling path such as the RS-232 DTR, DCD and RI leads are preferable. However, in-band controls have the advantage of being cheap and easy to use. Therefore, in-band signals are what are commonly used in practice.

Because the AT interface uses the data path between the DTE and the DCE for both application data and for commands, it would be useful to have a method whereby, while the AT interface is in data mode, the DTE can tell the DCE to enter command mode, while remaining connected to the peer DTE. The standard method of escaping data mode uses this key sequence: <pause>+++<pause>.

The mode that is entered from data mode after the escape sequence has been entered is commonly referred to as online command mode. In this mode, the communication link remains established, but data transmission is suspended. The modem does accept commands like it does in regular command mode, when there is no call up.

Table 1-8 illustrates a sample modem session that will serve to highlight some aspects of how this works. The originate modem session is on the left, the answer modem session on the right. AT commands entered by the DTE are in **bold** text, and the result codes from the DCE are in *italics*. Application data is in normal text and is shown on the transmitter's side.

Table 1-8 *Sample User Interface Session*

Originate-Side Session	Answer-Side Session
AT	
OK	
! The **OK** response signals the originate DCE's AT parser's ability to accept command input.	
ATD1234	
! Modem goes offhook, hears dial tone, transmits DTMF, and waits to hear answerback tone ! (ABT). The modem gets a fast busy.	
NO CARRIER	
ATD5703933	
! This time the call goes through and the PSTN presents ring voltage to the answer modem.	
	RING
! The answer DCE transmits this on the AT interface; it also toggles the RS-232 RI signal.	
	ATA
! Normally an answer modem will automatically answer upon incoming ring, but in this case ! the answering application sends an explicit **ATA** command due to the *RING* or *RI*, which ! causes the answer modem to go offhook, then starts transmitting ABT. The modems train ! successfully.	
CONNECT 26400/REL – LAPM	*CONNECT 26400 /V.42/V.42bis*
! The modems output their *CONNECT* strings; note the differing but equivalent formats. Then ! the modems raise DCD. At this point the DTE links are in data mode and are out of AT ! command mode.	

continues

Table 1-8 *Sample User Interface Session (Continued)*

Originate-Side Session	Answer-Side Session
	Welcome! Please login with username CISCO, password cisco. User Access Verification Username:
CISCO	
	CISCO
! This text is echoed by the answer DTE.	
	Password:
cisco	
! This text is not echoed.	
	access-3>
+++	
	+++
! The originate DTE transmits **+++**, which causes the client DTE link to switch out of data ! mode and into AT command mode (while leaving the DCE link intact).	
OK	
AT@e1	
CONNECTION STATUS *Modulation Type: V.34* *TX/RX Speed: 26400 26400 BPS* *TX/RX Symbol Rate: 3200 3200 Hz* *TX/RX Carrier Frequency: 1920 1829 Hz* *OK*	
! The user enters a (nonstandard) AT command to display some technical information on the ! modem connection (output edited for brevity).	
ATO	
CONNECT 26400/REL – LAPM	
! The **ATO** command tells the DCE to put the DTE link back into data mode (that is, go back ! "online").	

Generally, AT commands can be divided into two groups: control commands and configuration commands. Control commands cause the modem to perform call control functions, such as call setup, dialing, and teardown. Table 1-8 contains a sample of AT control commands and explains how they work. In addition to control commands, AT commands can be used to configure various modem settings. Except for the small subset of AT commands specified in V.25ter, there are enormous variations in the AT command set used by the different modem manufacturers.

Another way to configure a modem is through the status registers, usually referred to as S-registers. Register is a term used to describe a specific physical location in memory. In the case of modems, the S-registers are memory locations containing configuration information that can have their stored values read or (in most cases) altered via ATS commands. Some S-registers cannot be changed; these are known as read-only registers. Those registers that can be written to (that is, altered) are used to configure/change many of the modem's functions.

S-registers can have all the bits in that memory location represent a single configuration option, or they can be bitmapped registers with a single value representing multiple configuration options. Figure 1-9 shows both types of registers. The purpose of using bitmapped registers is to pack lots of configuration information into a small space.

Figure 1-9 *Standard and Bitmapped S-Register Comparison*

Standard S-Register

Register Name: S1
Register Function: Ring Counter
Default: 0

Range: 0-255 rings

Bit-Mapped S-Register

Register Name: S51
Register Function: Multifunction
Default: 0

Bit	Value	Result
0	1	MNP/V.42 disabled in V.22.
1	2	MNP/V.42 disabled in V.22bis.
2	4	MNP/V.42 disabled in V.32bis.
3-7	-	Reserved.

NOTE A typical modem has dozens of S-registers. The first few are standard across most modem manufacturers, but the rest are all different depending on modem manufacturer and model.

An S-register is configured according to ATS$n = x$, where n is the register number and x is the new value assigned to the register. So ATS0 = 3 sets register 0 to a value of 3 (that is, auto-answer in 3 rings). Writing to an S-register changes the total value of the register. Therefore, if altering a bitmapped register, you must ensure that cumulative total is such that it properly configures the group of bit values for each configuration option.

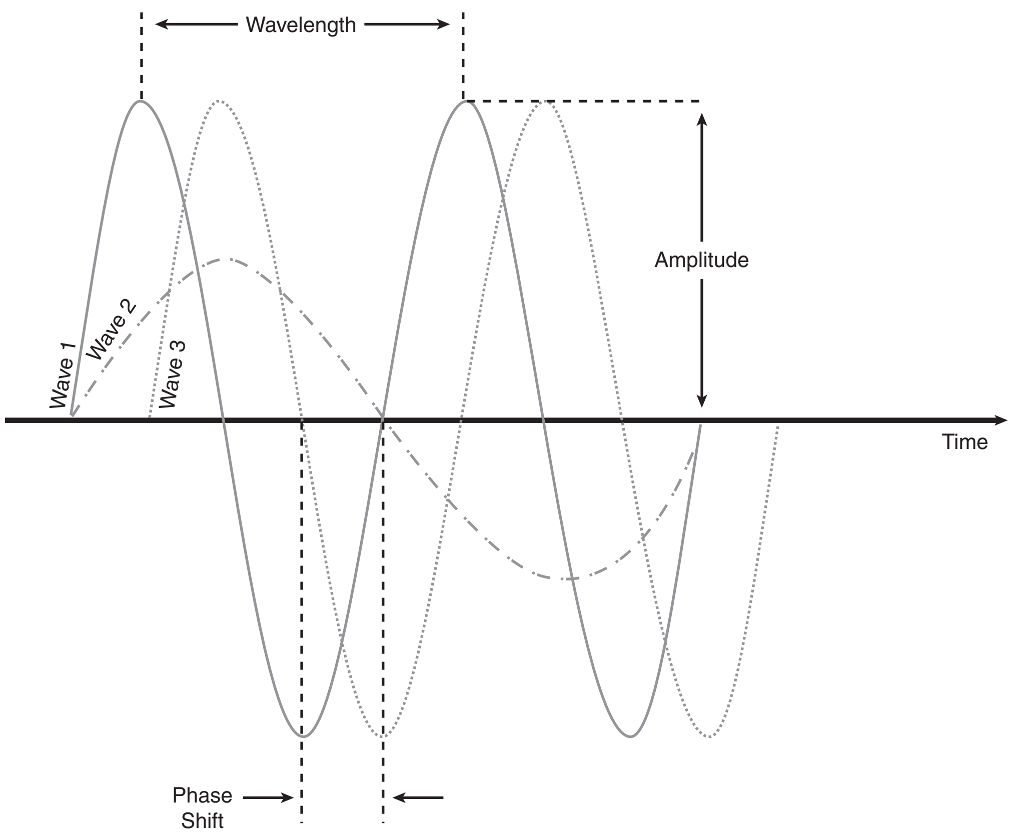
Modem-to-Modem Communication

Modulation is the most fundamental aspect of modem communications. It the means by which the binary digital data from the DTE link is encoded onto an analog signal that is sent over the PSTN. Different protocols describe the different types of modulations, and one of these protocols must be successfully negotiated between each modem on a point-to-point link for any communication to be possible. In this section, the various modulation schemes and how they correlate with the various ITU-T V-series modulation standards are discussed.

Modulation

The analog signals used in the transmission of modulated data are simply sinusoidal waveforms, and the primary components of any waveform are the amplitude, the frequency, and the phase. Figure 1-10 helps explain these concepts visually.

Figure 1-10 *Components of a Wave*



The amplitude is the magnitude of the wave. In the diagram, Wave 1 has twice the amplitude of Wave 2. The frequency is the inverse of the wavelength and thus is the number of oscillations occurring in a period. Note that in Figure 1-10 Wave 1 has half the wavelength of Wave 2, and hence it has twice the frequency. The phase is the position of the wave in its cycle period. Figure 1-10 shows Wave 3 to be 90 degrees out of phase with Wave 1.

There is a continuous analog signal between two modems during a call that is always present, known as the carrier signal. A pure sinusoidal carrier with no change in amplitude, frequency, phase, or some combination of these is unable to convey any information. To remedy this, modems change (or modulate) one or more of the components of the carrier wave to encode information onto it. The method in which the modem modifies one or more of these three wave characteristics is known as the modulation scheme.

The exact method of encoding the binary user data from the RS-232 DTE link onto the carrier is laid out in the specific modulation scheme that is negotiated. Each modulation scheme uses varying means of manipulating different combinations of the wave characteristics of the carrier signal. These differences produce diverse efficiencies in the amount of data per second that each modulation scheme can transmit over the data channel.

When talking about transmission rates for modems, two terms are commonly used. One is the familiar data rate in units of bits per second. The other is the more cryptic symbol rate in units of symbols per second. Sometimes you will hear the term baud rate, which is essentially equivalent to the symbol rate, but is no longer used in newer modulation standards because it as an antiquated term that is often misused.

A symbol represents a unique value assigned to each distinct state on a channel. In the context of modulation schemes, a symbol represents the possible states for encoding binary data. For example, assume a modulation scheme that varies the amplitude of the carrier at two distinct values. This is two unique states, which is the information carried by 1 bit of information in a binary system. If the example is extended by using a different modulation technique, whereby both the amplitude and phase of the carrier are changed, it will produce four distinct values. In this case, there are four unique states, which is the information carried by 2 bits in a binary system. Therefore, the more sophisticated a modulation scheme is, the more bits that are sent per symbol. Based on this explanation, you can clearly see that the symbol rate is equal to the data rate only when a symbol represents 1 bit of information (two states).

There are some practical applications of this concept of symbols. For example, when analyzing a modulation scheme that changes the phase or that changes both the phase and amplitude, a useful diagram, called a constellation, is used. A constellation diagram maps out points representing the various symbol states possible for a given modulation scheme. Effectively, each point in the constellation defines a sequence of bits to be transferred. Constellation diagrams are used in subsequent sections when discussing different modulation schemes.

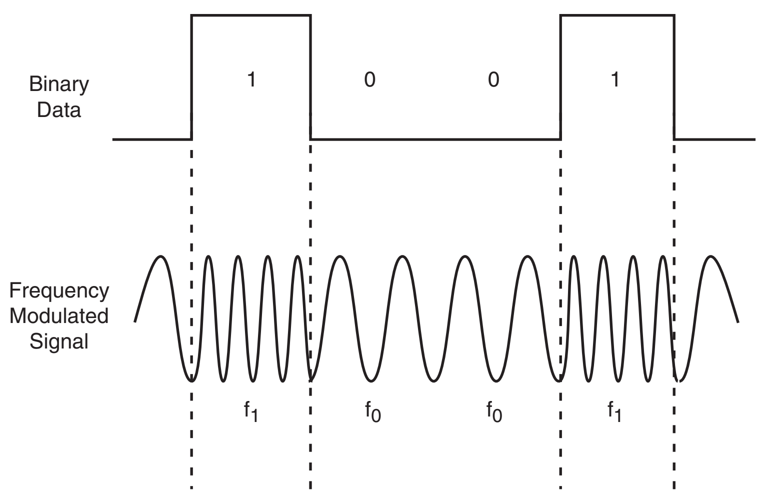
Many different modulation schemes covering various aspects of data communications exist, but for the discussion on modems in this section coverage is limited to the following:

- Frequency Shift Keying (FSK)
- Phase Shift Keying (PSK)
- Amplitude Modulation (AM)
- Quadrature Amplitude Modulation (QAM)
- Trellis Coded Modulation (TCM)

Frequency Shift Keying (FSK)

FSK simply uses one frequency tone to represent a 1 and another different frequency tone to represent a 0. Thus, it is able to produce an analog representation of the two logical states of binary digital data. For example, the digital base signal in Figure 1-11 is represented as a modulated carrier signal made up of two distinct frequencies. These two distinct frequencies are shown as f0 and f1, where f0 represents a space (or binary 0), and f1 represents a mark (or binary 1). The amplitude of the carrier is constant.

Figure 1-11 *FSK Modulation*

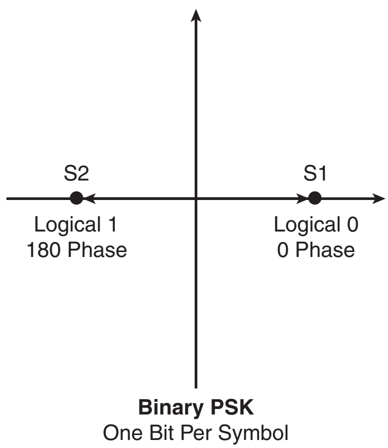


Because two states can be represented with binary FSK, there is only 1 bit per symbol. There is a single frequency per symbol and direction. Therefore, for full-duplex transmission, a set of four distinct frequencies is needed. The two most common standards that use FSK are Bell 103 and V.21 (used in fax transmissions).

Phase Shift Keying (PSK)

PSK uses a different phase to represent a binary state. For example, in binary PSK (BPSK), the phase of a constant amplitude and constant frequency carrier signal moves between 0 and 180 degrees to represent a logical 0 and a logical 1. Figure 1-12 illustrates the symbol states of binary PSK in a constellation diagram.

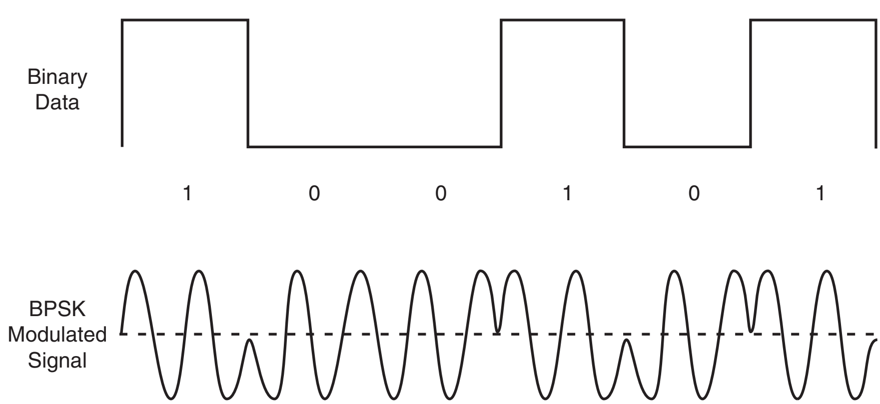
Figure 1-12 *Symbol States for BPSK Modulation*



Input Bit	Phase of Modulated Signal	Symbol State
0	0°	S1
1	180°	S2

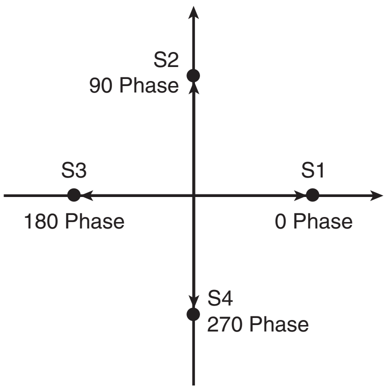
In Figure 1-13, the BPSK modulated carrier varies its phase between 0 and 180 degrees based on whether it receives a 0 or a 1 from the digital binary signal.

Figure 1-13 *BPSK Modulation*



One way to increase the transmission rate without changing the bandwidth requirements is to increase the number of bits represented by each phase change (that is, symbol). If the phase of the carrier is now varied between 0, 90, 180, and 270 degrees, you now have the ability to represent four different states (or 2 bits worth of information). This higher order of PSK is known as Quadrature PSK (QPSK) or 4-PSK and is shown graphically in Figure 1-14.

Figure 1-14 *Symbol States for QPSK Modulation*



Quadrature PSK
Two Bits Per Symbol

Input Bit	Phase of Modulated Signal	Symbol State
00	0°	S1
01	90°	S2
10	180°	S3
11	270°	S4

Clearly, the symbol rate in QPSK is double that of binary PSK and, consequently, so is the transmission rate. Higher orders of PSK are used, such as 8-PSK, which has eight states and is thus able to encode 3 bits per symbol.

Fax modulation V.27ter at 4800 bps uses 8-PSK. As expected, 8-PSK is 3 times faster than binary PSK and 1.5 times faster than QPSK, but it is more susceptible to link degradation.

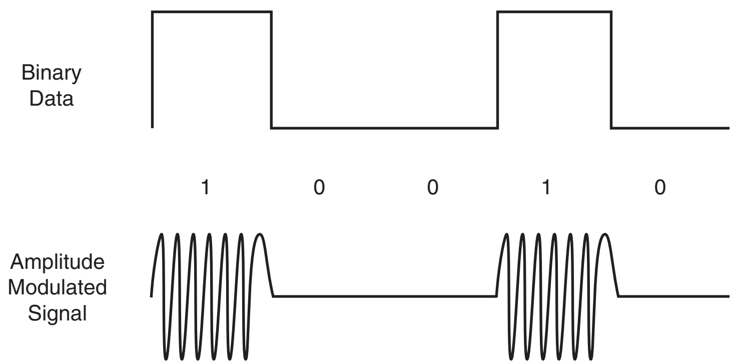
Another modulation technique that is a variation of PSK is Differential Phase Shift Keying (DPSK). As the name suggests, differential PSK encodes using changes in the phase of the carrier signal, rather than the carrier's absolute phase (as used in regular PSK). So, DPSK doesn't represent the binary signal; instead, it records changes in the binary stream.

Amplitude Modulation (AM)

AM occurs when the originating signal's variable voltage is applied to a carrier, causing the carrier's amplitude to change according to the originating signal. The digital form of AM, known as Amplitude Shift Keying (ASK), has only two logical states to re-create in the binary case. Therefore, ASK represents the digital data by using two amplitude levels, one of which is typically 0.

For example, a binary signal such as the one shown in Figure 1-15 would have an ASK modulated signal that appears as a burst of sinusoidal waves when there is a mark to transmit.

Figure 1-15 *Amplitude Modulation*

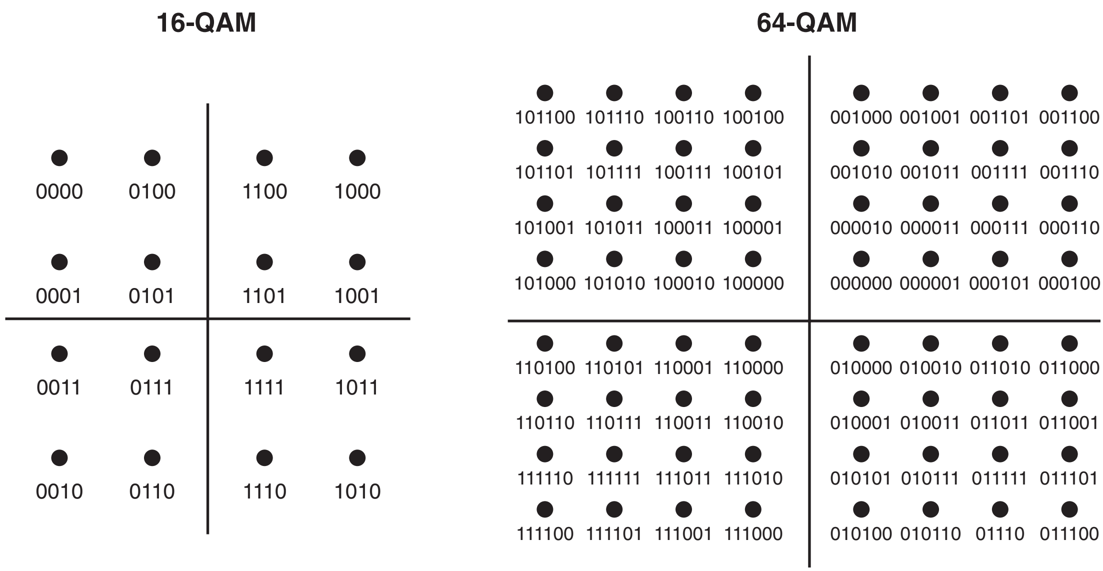


As with other schemes, ASK can have more sophisticated encoding schemes with additional amplitude levels (that is, four levels to represent 2 bits, eight levels to represent 3 bits, and so on). ASK is the simplest of the modulation techniques, but it has the drawback of being more susceptible to error, because amplitude is affected more by noise than frequency or phase.

Quadrature Amplitude Modulation (QAM)

The concepts of ASK and PSK can be combined to form QAM, where both phase and amplitude deviations can be used to encode the digital data. The dual nature of QAM allows for an increased number of unique states since many different phase shifts and amplitude level combinations can be used. Figure 1-16 illustrates how this increase in the number of symbols in the constellation pattern allows for more bits per symbol and therefore a greater data rate.

Figure 1-16 *Constellation Patterns for 16-QAM Versus 64-QAM*



There are limitations on bandwidth and signal-to-noise ratio (SNR) on an analog circuit that put an upper bound on the amount of data that can be transmitted per second. With regard to the constellation pattern of higher-order QAM, the problem manifests itself when the constellation points are close enough together that the receiving end is unable to distinguish one symbol from the next. This is due to the quantization and other noise that will invariably exist on the channel, limiting the overall throughput. Therefore, higher-order QAM is clearly more bandwidth efficient, but is much more susceptible to noise and distortion.

Trellis Coded Modulation (TCM)

TCM should be thought of as QAM with Trellis coding applied to it. Trellis coding is a mathematical algorithm that on the encoding side takes a certain number of bits (n-bits) as an input and produces a larger number of bits (m-bits) as an output. On the decoding side, an algorithm is used to find out the most likely n-bit sequence that would have produced the larger m-bit sequence, even if some of the bits were altered due to noise on the line.

For example, if a symbol is shifted by noise and falls close to a boundary, the modem uses the algorithm to examine the extra data (that is, the m-bit sequence) from the previous symbol to check the accuracy of the current symbol. Essentially, Trellis coding adds a form of error correction (known as Forward Error Correction [FEC]) to help the decoder deal with the effects of line noise.

NOTE The error correction done by Trellis coding is used at the modulation layer, so it can be done in addition to standard error correction protocols done at the data layer (that is, MNP4 and LAP-M, which are discussed in detail later in this chapter).

Therefore, TCM adds redundancy to the data and in return allows that data to be decoded with a lower error rate than plain QAM. V.32 and V.34 standards use TCM as their modulation scheme.

Modulation Standards

The preceding discussion focused on the theoretical nature of the different methods of modulation. Over time, public and proprietary standards have been defined based upon the different modulation schemes. These modulation standards can generically be broken down into two categories: analog modem modulation and digital modem modulation. The primary difference between the two is the carrier used. Analog modulation uses an analog carrier, whereas digital modulation uses a digital carrier.

Table 1-9 summarizes the primary modem modulation standards by order of chronology, which also correlates to increasing carrier rates. Also noted is the range of speeds that the protocol is usable over, and the speed increments that dictate how the protocol steps through the range of available speeds to find the most optimum bandwidth.

Table 1-9 *Modulation Standards Comparison*

Protocol	Carrier Rate (bps)	Carrier Increment	Carrier Type	Modulation Scheme
Bell103	300	N/A	Analog	FSK
V.21	300	N/A	Analog	FSK
Bell212A	1200	N/A	Analog	DPSK
V.22	1200	N/A	Analog	DPSK
V.22bis	1200 or 2400	N/A	Analog	QAM
V.23	600 or 1200 with optional 75 bps back channel	N/A	Analog	FSK
V.32	2400 to 9600	2400	Analog	QAM/TCM
V.32bis	4800 to 14400	2400	Analog	QAM/TCM

continues

Table 1-9 *Modulation Standards Comparison (Continued)*

Protocol	Carrier Rate (bps)	Carrier Increment	Carrier Type	Modulation Scheme
V.32Terbo	4800 to 19200	2400	Analog	QAM/TCM
V.FC	24000, 26400, 28800	N/A	Analog	TCM
V.34	2400 to 28800	2400	Analog	TCM
V.34+	2400 to 33600	2400	Analog	TCM
X2	28000 to 56000	1333	Digital	PCM/TCM
K56Flex	28000 to 56000	1333	Digital	PCM/TCM
V.90	28000 to 56000	1333	Digital	PCM/TCM

Modem Call Analysis

This section analyzes a call in its entirety. For the purpose of this discussion, the modem call is broken down into three component parts:

- The first part deals with the call setup and training sequence between modems.
- The second part covers the protocols and procedures associated with data transmission, including speedshifts, retrains, error control, and data compression.
- The third, and final part, discusses the call disconnect sequence.

Keep in mind that this subdivision of a modem call into three parts, as shown in Figure 1-17, is not based on any specification; instead, it is a logical division for this particular call analysis.

There are certain features and aspects common to all modulations, but for an in-depth analysis, these broad generalities become limiting. For a detailed treatment such as this, there are simply too many modulations to discuss individually, and there are too many differences between them to group them effectively. Thus, for this discussion, only one has been selected, V.34, as the backbone of the analysis. Wherever possible, the discussion tries to illustrate how the newer modulations, such as V.90, differ from the V.34 behavior being explained.