CISCO SYSTEMS

Deploying Cisco Voice over IP Solutions

Learn real-world voice-over-IP deployment solutions and strategies from the Cisco experts

Edited by **Jonathan Davidson**, CCIE® No. 2560
and **Tina Fox**

ciscopress.com

Deploying Cisco Voice over IP Solutions

Edited by: Jonathan Davidson and Tina Fox

Contributing Authors:

Phil Bailey
Rommel Bajamundi
Wayne Cheung
Thu Dao
Jonathan Davidson
Tina Fox
Sachin Gupta
Christina Hattingh
Ted Huff
Stephen Liu
Curt Mah
Greg Mercurio
Jeremy Pollock
Jim Rushton
Ravindar Shankar

Cisco Press

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

Deploying Cisco Voice over IP Solutions

Jonathan Davidson, Tina Fox, et. al.

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

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

Library of Congress Cataloging-in-Publication Number: 2001086622

ISBN:1-58705-030-7

Fourth Printing September 2004

Warning and Disclaimer

This book is designed to provide information about Voice over IP. 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.

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

Trademark Acknowledgments

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

Feedback Information

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

Corporate and Government Sales

Cisco Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact:

U.S. Corporate and Government Sales 1-800-382-3419 corpsales@pearsontechgroup.com

For sales outside of the U.S. please contact:

International Sales international@pearsontechgroup.com

Publisher	John Wait
Editor-In-Chief	John Kane
Cisco Representative	Anthony Wolfenden
Cisco Press Program Manager	Sonia Torres Chavez
Cisco Marketing Communications Manager	Scott Miller
Cisco Marketing Program Manager	Edie Quiroz
Production Manager	Patrick Kanouse
Development Editor	Andrew Cupp
Senior Editor	Sheri Cain
Copy Editor	Doug Lloyd
Technical Editor	Martin Walshaw
Team Coordinator	Tammi Barnett
Cover Designer	Louisa Adair
Composition	Argosy Publishing
Indexers	Tim Wright Larry Sweazy

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

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

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

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

Printed in the USA

About the Editors and Authors

Jonathan Davidson (CCIE #2560) is the Manager of Service Provider Technical Marketing for Packet Voice at Cisco Systems. He focuses on working with service provider and enterprise customers to develop solutions that are deployable in the new infrastructure of data and voice convergence. This includes designing customer networks and assisting with product direction.

Jonathan has been working on packet voice technologies for three years. During his seven years in the data-networking industry, he worked in various capacities, including network design, configuring, troubleshooting, and deploying data and voice networks.

Tina Fox is currently the Integration Solutions program manager for the Knowledge Management and Delivery group (IOS Technologies Division) at Cisco Systems and has been with Cisco Systems for 5 years. She attended the University of California at Los Angeles for both her undergraduate degree and graduate studies and completed a Certificate in Data and Telecommunications at the University of California at Irvine.

Phil Bailey is currently the technical documentation program manager for early deployment IOS releases for the Knowledge Management and Delivery group at Cisco Systems. He has a BSE in astronautical engineering, an M.Ed. in secondary education, and 15 years experience writing technical documentation for voice and data communications technologies, including Voice over IP, Voice over Frame Relay, and Voice over ATM.

Rommel Bajamundi is a Technical Marketing Engineer in Service Provider Technical Marketing at Cisco Systems. He has worked on various voice technologies at Cisco Systems for over 5 years and has focused on Voice over IP technologies for over three years.

Wayne Cheung is a manager in Service Provider Technical Marketing at Cisco Systems.

Thu Dao has been employed in the Voice over IP group at Cisco Systems for 4 years, mainly developing voice applications. She received her MS in computer science from Stanford University in December 1990.

Sachin Gupta (CCIE #3682) is currently the Manager of the Cisco IOS Technical Marketing group. Sachin worked in customer support at Cisco for the WAN and Multiservice teams for two years, and then as a Technical Marketing Engineer in Cisco IOS Technologies Division focusing on QoS and MPLS for two more years. Sachin holds a BS in electrical engineering from Purdue University and is completing his MS in electrical engineering at Stanford University.

Christina Hattingh is a member of the Technical Marketing organization at Cisco Systems. In this role, she works closely with product management and engineering, and focuses on assisting Cisco Sales Engineers, partners, and customers to design and tune enterprise and service provider Voice over X networks. Prior to this, she was a software engineer and engineering manager of PBX Call Center products at Nortel Networks. Earlier software development experience in X.25 and network management systems provide background to the issues involved today in migrating customers' traditional data and voice networks to packet-based technologies. Christina has a graduate degree in computer science and mathematical statistics.

Ted Huff has been Technical Marketing Engineer in the Service Provider TME group at Cisco Systems for almost 5 years. After graduating from California State University Chico in 1994 with a bachelor of science degree in computer engineering, Ted found his way to Cisco via Lockheed Martin.

All of Ted's time at Cisco has been in the same TME organization, where he has been involved in various aspects of Voice over IP technology including billing and accounting, interactive voice response programming, store and forward fax, and network management.

When not at his desk or in the lab, Ted enjoys working around the garden and spending time with his wife Catherine and his two lovely daughters, Anna and Jessi.

Stephen Liu is currently Manager of Service Provider Technical Marketing at Cisco Systems, where he has been employed for the past 6 years. He received his BSEE in Communications Systems from the University of California at San Diego and is CCIE certified (CCIE #2430). Stephen has served as Cisco's H.323 VoIP representative to IMTC/ETSI TIPHON and co-chaired the IMTC iNOW! H.323 Interoperability Forum. In his spare time, Stephen volunteers as a youth soccer coach.

Curt Mah (CCIE #3856) is a Technical Marketing Engineer for Cisco Systems, working on Voice over IP and wholesale voice networks for the service provider line of business. Curt joined Cisco Systems in 1996 and has worked assisting customers in various data networking capacities, including network design, training, implementation, and troubleshooting. Curt has a BSE in electronic engineering technology from Cal Poly, San Luis Obispo.

Greg Mercurio is a Senior Software Engineer at Cisco Systems. "Empowering the customer through advanced Internet media!"

Jeremy Pollock is a Technical Writer for Cisco Systems, working on Cisco IOS software documentation. He writes feature and solution documentation, specializing in VoIP and access VPDN documentation. He received a B.A. in physics from the University of California in 1997 and a Certificate in Technical Communications from San Jose State University in 1998.

Jim Rushton has worked with enterprise networking technology in various capacities for more than 15 years. He is currently a Cisco Systems Technical Writer in the Irvine, California office, assigned to the IOS Technologies Division.

Ravindar Shankar is a senior technical marketing engineer with a focus on voice solutions in service provider marketing. He has been with Cisco for 8 years and has served in the capacities of customer support, development engineering, and most recently in voice technical marketing. He has a master's degree, holds a patent in network management, is CCIE certified (CCIE #1303), and has a good understanding of both data and voice technologies.

About the Technical Reviewer

Martin Walshaw, (CCIE #5629), CCNP, CCDP, is a Systems Engineer working for Cisco Systems in the Enterprise Line of Business in South Africa. His areas of specialty include Convergence (voice and video) and security, which keeps him busy both night and day. During the last 12 years Martin has dabbled in many aspects of the IT industry, ranging from programming in RPG III and Cobol to PC sales. When Martin is not working, he likes to spend all of his available time with his expectant wife Val and his son Joshua. Without their patience, understanding, and support, projects such as this would not be possible. “To Infinity and beyond . . .”

Dedications

To Tina, whose dedication and determination made this project possible, and whose constant encouragement persuaded me to become a part of it.—Phil Bailey

Many thanks to my family for always being there. Thanks to all the folks on my team who help me on a daily basis: Jon Davidson, Stephen Liu, Brian Gracely, Wayne Cheung, Conrad Price, Ravi Shankhar, Anand Ramachandran, Edmund Lam, David Morgan, Oscar Thomas, Ted Huff, Shyam Kota, Curt Mah, Aseem Srivasta, and Wei Wang. Special thanks to my wife, Josie, who keeps me going.—Rommel Bajamundi

To my mother, Uma Gupta, for her strength, love, and support, always.—Sachin Gupta

My contribution to this book was made possible in large part by the technical guidance and mentorship of a number of Cisco colleagues—with particular thanks to Jonathan Davidson, Brian Gracely, Conrad Price, and Chris Spain. I dedicate this book to Robert Verkroost who never fails to support and encourage me in my various forays into the world of publishing.—Christina Hattingh

To Mom and Dad: Thanks for all of the wisdom and guidance over the years. To Tammy: Thanks for always providing love and encouragement to chase my dreams, and for reminding me of the power of "Hakuna Matata."—Curt Mah

To my family, for allowing me to spend the extra time to make a difference.—Greg Mercurio

To my parents for all of their love and support as I stumble along towards the life I'm trying to live. And to Tammy for keeping me in touch with life beyond networking and for showing me the power of the written word to make the real world a better place.—Jeremy Pollock

Acknowledgments

The writing of this book was a group effort and could not have been completed without the dedicated leadership and sacrifice of Tina Fox. She helped keep the book on track and quite literally was the brains and driving force behind this endeavor. A special thanks goes to her. I would also like to thank the writers who smoothed out the material submitted by the subject matter experts (SMEs) and created a single voice for the book. This includes Tina Fox, Phil Bailey, Jeremy Pollock, and Jim Rushton.

The SMEs wrote the initial draft of each of the chapters and provided expertise in specific areas. This book is truly a consolidation of some of the brightest minds in packet telephony today. These SMEs have real-world, in-depth knowledge of how the protocols work in live networks. Thus, the reader will receive up-to-date knowledge of the latest techniques and technologies in the packet voice arena. These SMEs include Christina Hattingh, Sachin Gupta, Rommel Bajamundi, Kevin Connor, Stephen Liu, Thu Dao, Curt Mah, Ted Huff, Wayne Cheung, Greg Mercurio, Ravi Shankar, and Massimiliano Caranza.

—Jonathan Davidson

Contents at a Glance

Contents

Introduction

This book is a sequel to *Voice over IP Fundamentals*, published by Cisco Press in 2000. Since the publication of that book, there has been a fundamental change in the assumptions made by those in the telecommunications industry. Instead of incumbent telecommunications service providers attempting to determine whether packet voice will be a viable technology, they are attempting to determine the right opportunity to begin deploying this technology. These providers are either actively researching the technology, conducting lab trials, or deploying it. In addition, existing TDM equipment providers have determined that they must provide packet voice equipment in addition to their TDM equipment. These equipment providers are forced down this path due to the fact that their customers want and need to purchase this type of equipment.

The next phase of packet voice will focus not just on lowering equipment costs (capital expenditures), but lowering operating expenditures. This phase will be completed over time by integrating voice technology into Network Management Systems (NMS) owned by incumbent carriers as well as integrating IP data NMS technology and packet voice network management systems.

Although service providers realize that there are many benefits to packet voice technology, they also recognize that there are potential downsides to this technology. The largest of the potential caveats is multi-vendor equipment interoperability. Although there are many standards defining how devices should communicate with each other, there are few standards defining how these independent standards should communicate with each other. One example is how many existing networks utilize H.323 to signal voice calls over IP. There are several newer protocols, however, that appear to have momentum in this space—MGCP, MEGACO, and SIP, for example.

The good news about *protocol interworking* is that there is much work being done in this space. Each major protocol has its own interoperability event at least yearly. There is a good analogy that can be drawn from the data networking industry. There are dozens of routing protocols currently in use across the world for routing IP across heterogeneous networks (OSPF, IS-IS, BGP, EIGRP, and so on), and all of these protocols must interoperate with one another in order for IP networks to be truly ubiquitous. This interoperability has, of course, been accomplished. Another comparison that can be drawn between packet voice signaling protocols and IP routing protocols is that there is a definite need for each of these protocols in certain types of networks, and one cannot expect to erase the need for another. In the packet voice space, a newer protocol such as MEGACO may be better for certain applications, but it doesn't solve the same problem that protocols such as H.323 solve. Therefore, they are both necessary, and interoperability between the two is required.

The interoperability between equipment vendors will be solved, and then the next level of interoperability will bubble to the surface—that of service interoperability, or how users can utilize a similar application across an entire service area in a similar manner.

Purpose of This Book

The purpose of this book is to provide you with a basic understanding of some of the advanced topics associated with designing and implementing a VoIP network. As such, this book is meant to accomplish the following goals:

- Provide an introduction to some of the more important preliminary design elements that need to be considered before implementing VoIP, such as echo and traffic analysis, quality of service (QoS), and call admission control (CAC).
- Introduce the basic tasks involved in designing an effective service provider-based VoIP network.
- Provide information on some of the more popular and widely requested VoIP services, such as prepaid services, fax services, and virtual private networks (VPNs).

Although this book contains plenty of technical information and suggestions for ways in which you can build a VoIP network, it is not meant to be used as a cookie cutter design and implementation guide. Examples shown in this book are included only to clarify concepts and design issues.

Audience

Although this book is written for anyone interested in understanding the design considerations and strategies necessary to deploy VoIP, its target audience is service provider voice and networking experts who are already familiar with VoIP and networking fundamentals. We strongly suggest that you first read *Voice over IP Fundamentals* before tackling the topics presented in this book.

Chapter Organization

Deploying Cisco Voice over IP Solutions is separated into four parts:

- Network Design Considerations
- Network Design Strategies
- Network Services
- Appendixes

Part I, "Network Design Considerations," discusses some of the preliminary topics you should take into account before designing a VoIP network:

- Chapter 1, "Understanding Traffic Analysis," describes different techniques to engineer and properly size traffic-sensitive voice networks, provides examples of several different kinds of traffic models, and explains how to use traffic probability (distribution) tables to engineer robust and efficient voice networks.

- Chapter 2, “Understanding Echo Analysis,” describes basic concepts applicable to echo analysis, explains echo cancellation, and provides a method for locating and eliminating echoes.
- Chapter 3, “Understanding Quality of Service for Voice over IP,” describes various QoS features applicable to voice and provides high-level examples showing how to deploy these features in different voice network environments.
- Chapter 4, “Understanding Call Admission Control,” describes call admission control (CAC), when the CAC decision is made, how the information is gathered to support the CAC decision, what resources are needed for the voice call and how they are determined, and what happens to calls denied by CAC.

Part II, “Network Design Strategies,” describes how to design a service provider-based voice network:

- Chapter 5, “Designing Static Dial Plans for Large VoIP Networks,” describes dial plan configuration recommendations on Cisco H.323 gateways and gatekeepers used to support large dial plans.
- Chapter 6, “Designing a Long-Distance VoIP Network,” describes the basic tasks of designing a long-distance VoIP network.

Part III, “Network Services” describes some of the more commonly requested services that service providers can offer through a voice network:

- Chapter 7, “Managed Multiservice Networks and Packet Voice VPNs,” discusses two classes of hosted voice networks: Managed MultiService (MMS) networks and packet voice virtual private networks (VPNs).
- Chapter 8, “Fax Services,” discusses store and forward and real-time relay fax services.
- Chapter 9, “Unified Messaging,” discusses various unified messaging concepts and features that apply to Cisco’s uOne unified messaging (UM) solution.
- Chapter 10, “Prepaid Services,” discusses how to design and implement a prepaid services solution managed either through an internal network infrastructure or through an OSP clearinghouse.

The appendixes are as follows:

- Appendix A, “Erlang B Traffic Model,” provides an explanation and example of an Erlang B Traffic Distribution Table. This information is supplementary to Chapter 1, “Understanding Traffic Analysis.”
- Appendix B, “Extended Erlang B Traffic Model,” provides an explanation and example of an Extended Erlang B Traffic Distribution Table. This information also is supplementary information for Chapter 1, “Understanding Traffic Analysis.”

- Appendix C, "TCL IVR Scripts," provides an overview of Interactive Voice Response (IVR) Tool Command Language (TCL) scripts and examples of some of the more common IVR TCL scripts used with prepaid services. This information is supplementary information for Chapter 10, "Prepaid Services."

Features and Text Conventions

Text design and content features used in this book are intended to make the complexities of VoIP clearer and more accessible.

Key terms are italicized the first time they are used and defined. In addition, key terms are spelled out and followed with their acronym in parentheses, where applicable.

Chapter summaries provide a chance for you to review and reflect upon the information discussed in each chapter. You might also use these summaries to determine whether a particular chapter is appropriate for your situation.

Command Syntax Conventions

Command syntax in this book conforms to the following conventions:

- Commands, keywords, and actual values for arguments are **bold**.
- Arguments (which need to be supplied with an actual value) are *italic*.
- Optional keywords and arguments are in brackets [].
- A choice of mandatory keywords and arguments is in braces { }.

Note that these conventions are for syntax only.

Timeliness

As of the writing of this book, many new protocols concerning VoIP were still being designed and worked out by the standards bodies. Also, legal aspects of VoIP constantly arise in different parts of the world. Therefore, this book is meant as a guide in that it provides foundational voice network design information.

The Road Ahead . . .

Packet voice technology is here to stay. There are potential deployments of this technology in many applications, whether residential, transit, or managed service. The predominant consensus of potential migration paths is as follows:

- Migration for the Enterprise
  - Enterprise customers will follow the path of attaching voice gateways to their PBXs to allow inter-PBX communication via VoIP. Then they will replace their PBXs with IP PBXs that can offer greater efficiency and additional applications.
- Migration for the Service Provider
  - Service provider customers will use packet voice to replace or grow their services without having to grow their TDM networks. This will start with Tandem Class 4 type networking and interconnecting with other service providers via IP instead of TDM. It will then move to Business Local services and finally the consumer.
  - Wireless voice will follow a similar path as enterprise and service provider. It will start by having a separate data network and then move to having all of the services, including voice, run over the data network.

PART I

Network Design Considerations

CHAPTER 1

Understanding Traffic Analysis

Networks, whether voice or data, are designed around many different variables. Two of the most important factors that you need to consider in network design are service and cost. Service is essential for maintaining customer satisfaction. Cost is always a factor in maintaining profitability. One way you can maintain quality service and rein in cost in network design is to optimize circuit utilization.

This chapter describes the different techniques you can use to engineer and properly size traffic-sensitive voice networks. You'll see several different traffic models and explanations of how to use traffic probability tables to help you engineer robust and efficient voice networks.

Traffic Theory Basics

Network designers need a way to properly size network capacity, especially as networks grow. Traffic theory enables network designers to make assumptions about their networks based on past experience.

Traffic is defined as either the amount of activity over a circuit or the number of messages handled by a communications switch during a given period of time. Traffic also includes the relationship between call attempts on traffic-sensitive equipment and the speed with which the calls are completed. Traffic analysis enables you to determine the amount of bandwidth you need in your circuits for both data and voice calls. Traffic engineering addresses service issues by enabling you to define a grade of service or blocking factor. A properly engineered network has low blocking and high circuit utilization, which means that service is increased and costs are reduced.

You need to take many different factors into account when analyzing traffic. The most important factors are the following:

- Traffic load
- Grade of service
- Traffic types
- Sampling methods

Of course, other factors might affect the results of traffic analysis calculations, but these are the main ones.

Traffic Load Measurement

In traffic theory, you measure traffic load. *Traffic load* is defined as the ratio of call arrivals in a specified period of time to the average amount of time it takes to service each call during that period. These measurement units are based on *Average Hold Time (AHT)*. AHT is defined as the total amount of time of all calls in a specified period divided by the number of calls in that period. For example:

3976 total call seconds / 23 calls = 172.87 sec per call = AHT of 172.87 seconds

The two main measurement units used today to measure traffic load are the following:

- Erlangs
- Centum Call Seconds (CCS)

In 1918, A.K. Erlang developed formulas that could be used to make predictions about randomly arriving telephone traffic. The Erlang—a measurement of telephone traffic—was named in honor of him. One Erlang is defined as 3600 seconds of calls on the same circuit, or enough traffic load to keep one circuit busy for 1 hour.

Traffic in Erlangs = (number of calls × AHT) / 3600
Example: (23 calls × 172.87 AHT) / 3600 = 1.104 Erlangs

CCS is based on 100 seconds of calls on the same circuit. Voice switches generally measure the amount of traffic in CCS.

Traffic in CCS = (number of calls × AHT) / 100
Example: (23 calls × 172.87 AHT) / 100 = 39.76 CCS

Which unit you use depends on the equipment you use and the unit of measurement it records in. Many switches use CCS because it is easier to work with increments of 100 rather than 3600. Both units are recognized standards in the field. The following is how the two relate:

1 Erlang = 36 CCS

Although you can take the total call seconds in an hour and divide that amount by 3600 seconds to determine traffic in Erlangs, you can also use averages of various time periods. These averages allow you to utilize more sample periods and determine the proper traffic.

Busy Hour Traffic

You commonly measure traffic load during your network's busiest hour because this represents the maximum traffic load that your network must support. The result gives you a traffic load measurement commonly referred to as the *Busy Hour Traffic* (BHT). Times can arise when you can't do a thorough sampling or you have only an estimate of how many calls you are handling daily. When that happens, you can usually make assumptions about your environment, such as the average number of calls per day and the AHT. In the standard business environment, the busy hour of any given day holds approximately 15 to 20 percent of that day's traffic. You generally use 17 percent of the day's traffic to represent the peak hour in your computations. In many business environments, an acceptable AHT is generally assumed to be 180 to 210 seconds. You can use these estimates if you ever need to determine trunking requirements without having more complete data.

Network Capacity Measurements

Many measurements can be used to discuss a network's capacity. For example:

- Busy Hour Call Attempts (BHCA)
- Busy Hour Call Completions (BHCC)
- Calls per second (CPS)

All these measurements are based on the number of calls. These measurements describe a network's capacity but they are fairly meaningless for traffic analysis because they do not consider the hold time of the call. You need to use these measurements in conjunction with an AHT to derive a BHT that you can use for traffic analysis.

Grade of Service

Grade of service (GoS) is defined as the probability that calls will be blocked while attempting to seize circuits. It is written as P.*xx* blocking factor or blockage, where *xx* is the percentage of calls that are blocked for a traffic system. For example, traffic facilities requiring P.01 GoS define a 1 percent probability of callers being blocked to the facilities. A GoS of P.00 is rarely requested and will seldom happen. This is because, to be 100 percent sure that there is no blocking, you would have to design a network where the caller-to-circuit ratio is 1:1. Also, most traffic formulas assume that an infinite number of callers exists.

Traffic Types

You can use the telecommunications equipment offering the traffic to record the previously mentioned data. Unfortunately, most of the samples received are based on the carried traffic on the system and not the offered traffic load.

Carried traffic is the traffic that is actually serviced by telecommunications equipment.

Offered traffic is the actual amount of traffic attempts on a system. The difference in the two can cause some inaccuracies in your calculations.

The greater the amount of blockage you have, the greater the difference between carried and offered load. You can use the following formula to calculate offered load from carried load:

Offered load = carried load / (1–blocking factor)

Unfortunately, this formula does not take into account any retries that might happen when a caller is blocked. You can use the following formula to take retry rate into account:

Offered load = carried load × Offered Load Adjustment Factors (OAF)
OAF = [1.0 – (x × blocking factor)] / (1.0 – blocking factor)
where x is defined as a percentage of retry probability ($x = 0.6$ for a 60% retry rate)

Sampling Methods

The accuracy of your traffic analysis will also depend on the accuracy of your sampling methods. The following parameters will change the represented traffic load:

- Weekdays versus weekends
- Holidays
- Type of traffic (modem versus traditional voice)
- Apparent versus offered load
- Sample period
- Total number of samples taken
- Stability of the sample period

Probability theory states that to accurately assess voice network traffic, you need to have at least 30 of the busiest hours of a voice network in the sampling period. Although this is a good starting point, other variables can skew the accuracy of this sample. You cannot take the top 30 out of 32 samples and expect that to be an accurate picture of the network's traffic. To get the most accurate results, you need to take as many samples of the offered load as possible. Alternatively, if you take samples throughout the year, your results can be skewed as your year-to-year traffic load increases or decreases. The ITU-T makes recommendations on how you can accurately sample a network to dimension it properly.

The ITU-T recommends that Public Switched Telephone Network (PSTN) connections measurement or read-out periods be 60 minutes and/or 15 minute intervals. These intervals are important because they let you summarize the traffic intensity over a period of time. If you take measurements throughout the day, you can find the peak hour of traffic in any given day. There are two recommendations on how to arrive at the peak daily traffic:

- **Daily Peak Period (DPP)**—Records the highest traffic volume measured during a day. This method requires continuous measurement and is typically used in environments where the peak hour might be different from day to day.
- **Fixed Daily Measurement Interval (FDMI)**—Used when traffic patterns are somewhat predictable and peak periods occur at regular intervals (i.e., business traffic usually peaks around 10:00 a.m. to 11:00 a.m. and 2:00 p.m. to 3:00 p.m.). FDMI requires measurements only during the predetermined peak periods.

In Table 1-1, by using FDMI sampling, you see that the hour with the highest total traffic load is 10 a.m., with a total traffic load of 60.6 Erlangs.

Table 1-1 *Daily Peak Period Measurement Table*

	Monday	Tuesday	Wednesday	Thursday	Friday	Total Load
9:00 a.m.	12.7	11.5	10.8	11.0	8.6	54.6
10:00 a.m.	12.6	11.8	12.5	12.2	11.5	60.6
11:00 a.m.	11.1	11.3	11.6	12.0	12.3	58.3
12:00 p.m.	9.2	8.4	8.9	9.3	9.4	45.2
1:00 p.m.	10.1	10.3	10.2	10.6	9.8	51.0
2:00 p.m.	12.4	12.2	11.7	11.9	11.0	59.2
3:00 p.m.	9.8	11.2	12.6	10.5	11.6	55.7
4:00 p.m.	10.1	11.1	10.8	10.5	10.2	52.7

The example in Table 1-2 uses DPP to calculate total traffic load.

Table 1-2 *Using DPP to Calculate Total Traffic Load*

	Monday	Tuesday	Wednesday	Thursday	Friday	Total Load
Peak Traffic	12.7	12.2	12.6	12.2	12.3	62.0
Peak Traffic Time	9 a.m.	2 p.m.	3 p.m.	10 a.m.	11 a.m.	

You also need to divide the daily measurements into groups that have the same statistical behavior. The ITU-T defines these groups as workdays, weekend days, and yearly exceptional days. Grouping measurements with the same statistical behavior becomes important because exceptional call volume days (such as Christmas Day and Mother's Day) might skew the results.

ITU-T Recommendation E.492 includes recommendations for determining the normal and high load traffic intensities for the month. Per ITU recommendation E.492, the normal load traffic intensity for the month is defined as the fourth highest daily peak traffic. If you select the second highest measurement for the month, it will result in the high load traffic intensity for the month. The result allows you to define the expected monthly traffic load.

Traffic Models

Now that you know what measurements are needed, you need to figure out how to use the measurements. You need to pick the appropriate model. The following are the key elements to picking the appropriate model:

- Call arrival patterns
- Blocked calls
- Number of sources
- Holding times

Call Arrival Patterns

Determining the call arrival pattern is the first step to designating the proper traffic model to choose. Call arrival patterns are important in choosing a model because arrival patterns affect traffic facilities differently.

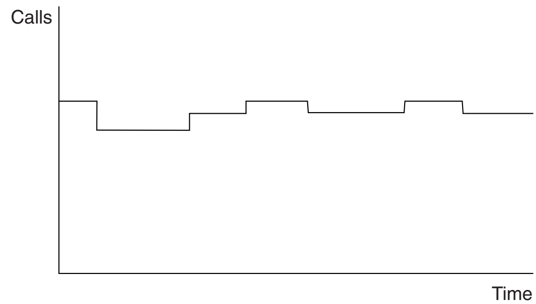
The three main call arrival patterns are the following:

- Smooth
- Peaked
- Random

Smooth Call Arrival Pattern

A smooth or hypo-exponential traffic pattern occurs when there is not a large amount of variation in traffic. Call hold time and call inter-arrival times are predictable, which allows you to predict traffic in any given instance when a finite number of sources exist. For example, suppose you are designing a voice network for an outbound telemarketing company in which a few agents spend all day on the phone. Suppose that, in a 1-hour period, you expect 30 calls of 2 minutes each, with calls coming one after the other. You then need to allocate one trunk to handle the calls for the hour. Figure 1-1 provides a graph of what calls versus time might look like in a smooth call arrival pattern.

Figure 1-1 *Smooth call arrival pattern.*

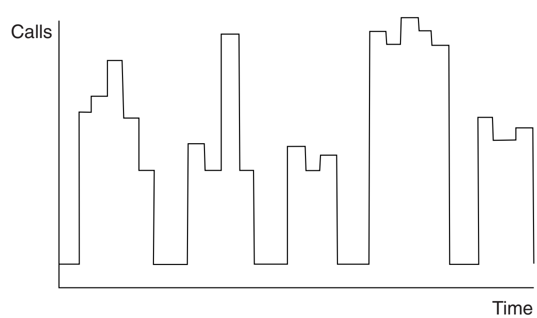


Peaked Call Arrival Pattern

A peaked traffic pattern has big spikes in traffic from the mean. This call arrival pattern is also known as a hyper-exponential arrival pattern. Peaked traffic patterns demonstrate why it might not be a good idea to include Mother's Day and Christmas Day in a traffic study. Times might arise when you would want to engineer rollover trunk groups to handle this kind of traffic pattern. In general, however, to handle this kind of traffic pattern, you need to allocate enough resources to handle the peak traffic. For example, to handle 30 calls all at once, you would need 30 trunks.

Figure 1-2 provides a graph of what calls versus time for a peaked call arrival pattern might look like.

Figure 1-2 *Peaked call arrival pattern.*

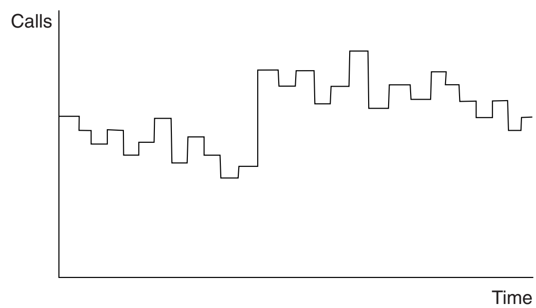


Random Call Arrival Pattern

Random traffic patterns are exactly that—random. They are also known as Poisson or exponential distribution. Poisson was the mathematician who originally defined this type of distribution. Random traffic patterns occur in instances where there are many callers, each one generating a little bit of traffic. You generally see this kind of random traffic pattern in PBX environments. The number of circuits that you would need in this situation would vary between 1 and 30.

Figure 1-3 illustrates what a graph of calls versus time for a random call arrival pattern might look like.

Figure 1-3 *Random call arrival pattern.*



Blocked Calls

A *blocked call* is a call that is not serviced immediately. Calls are considered blocked if they are rerouted to another trunk group, placed in a queue, or played back a tone or announcement. The nature of the blocked call determines the model you select, because blocked calls result in differences in the traffic load.

The following are the main types of blocked calls:

- **Lost Calls Held (LCH)**—These blocked calls are lost, never to come back again. Originally, LCH was based on the theory that all calls introduced to a traffic system were held for a finite amount of time. All calls include any of the calls that were blocked, which meant the calls were still held until time ran out for the call.
- **Lost Calls Cleared (LCC)**—These blocked calls are cleared from the system—meaning that the call goes somewhere else (mainly to other traffic-sensitive facilities).
- **Lost Calls Delayed (LCD)**—These blocked calls remain on the system until facilities are available to service the call. This is used mainly in call center environments or with data circuits, since the key factors for LCD would be delay in conjunction with traffic load.
- **Lost Calls Retried (LCR)**—This assumes that once a call is blocked, a percentage of the blocked calls are lost and all other blocked calls retry until they are serviced. This is actually a derivative of the LCC model and is used in the Extended Erlang B model.

Number of Sources

The number of sources of calls also has bearing on what traffic model you choose. For example, if there is only one source and one trunk, the probability of blocking the call is zero. As the number of sources increases, the probability of blocking gets higher. The number of sources comes into play when sizing a small PBX or key system, where you can use a smaller number of trunks and still arrive at the designated GoS.

Holding Times

Some traffic models take into account the holding times of the call. Most models do not take holding time into account because call-holding times are assumed to be exponential. Generally, calls have short rather than long hold times, meaning that call-holding times will have a negative exponential distribution.

Selecting Traffic Models

After you determine the call arrival patterns and determine the blocked calls, number of sources, and holding times of the calls, you are ready to select the traffic model that most closely fits your environment. Although no traffic model can exactly match real-life situations, these models assume the average in each situation. Many different traffic models exist. The key is to find the model that best suits your environment. Table 1-3 compares some common traffic models.

Table 1-3 *Traffic Model Comparison*

Traffic Model	Sources	Arrival Pattern	Blocked Call Disposition	Holding Times
Poisson	Infinite	Random	Held	Exponential
Erlang B	Infinite	Random	Cleared	Exponential
Extended Erlang B	Infinite	Random	Retried	Exponential
Erlang C	Infinite	Random	Delayed	Exponential
Engset	Finite	Smooth	Cleared	Exponential
EART/EARC	Infinite	Peaked	Cleared	Exponential
Neal-Wilkerson	Infinite	Peaked	Held	Exponential
Crommelin	Infinite	Random	Delayed	Constant
Binomial	Finite	Random	Held	Exponential
Delay	Finite	Random	Delayed	Exponential

The traffic models that have the widest adoption are Erlang B, Extended Erlang B, and Erlang C. Other commonly adopted traffic models are Engset, Poisson, EART/EARC, and Neal-Wilkerson.

Erlang B Traffic Model

The Erlang B model is based on the following assumptions:

- An infinite number of sources
- Random traffic arrival pattern
- Blocked calls are cleared
- Hold times are exponentially distributed

The Erlang B model is used when blocked calls are rerouted, never to come back to the original trunk group. This model assumes a random call arrival pattern. The caller makes only one attempt and if the call is blocked, the call is then rerouted. The Erlang B model is commonly used for first-attempt trunk groups where you do not need to take into consideration the retry rate because calls are rerouted, or you expect to see very little blockage.

Equation 1-1 provides the formula used to derive the Erlang B traffic model.

Equation 1-1

$$B(c, a) = \frac{\frac{a^c}{c!}}{\sum_{k=0}^{c} \frac{a^k}{k!}}$$

where:

- $B(c,a)$ is the probability of blocking the call.
- c is the number of circuits.
- a is the traffic load.

Example: Using the Erlang B Traffic Model

Problem: You need to redesign your outbound long-distance trunk groups, which are currently experiencing some blocking during the busy hour. The switch reports state that the trunk group is offered 17 Erlangs of traffic during the busy hour. You want to have low blockage so you want to design this for less than 1 percent blockage.

Solution: When you look at the Erlang B Tables (see Appendix A, "Erlang B Traffic Model"), you see that for 17 Erlangs of traffic with a Grade of Service of 0.64 percent, you need 27 circuits to handle this traffic load.

You can also check the blocking factor using the Erlang B equation, given the preceding information. Another way to check the blocking factor is to use Microsoft Excel's Poisson function in the following format:

> =(POISSON(<circuits>,<traffic load>,FALSE)) / (POISSON(<circuits>,<traffic load>,TRUE))

There is a very handy Erlang B, Extended Erlang B, and Erlang C calculator at the following URL: www.erlang.com/calculator/index.htm.

Extended Erlang B Traffic Model

The Extended Erlang B model is based on the following assumptions:

- An infinite number of sources.
- Random traffic arrival pattern.
- Blocked calls are cleared.
- Hold times are exponentially distributed.

The Extended Erlang B model is designed to take into account calls that are retried at a certain rate. This model assumes a random call arrival pattern; blocked callers make multiple attempts to complete their calls and no overflow is allowed. The Extended Erlang B model is commonly used for standalone trunk groups with a retry probability (for example, a modem pool).

Example: Using the Extended Erlang B Traffic Model

Problem: You want to determine how many circuits you need for your dial access server. You know that you receive about 28 Erlangs of traffic during the busy hour and that 5 percent blocking during that period is acceptable. You also expect that 50 percent of the users will retry immediately.

Solution: When you look at the Extended Erlang B Tables (see Appendix B, "Extended Erlang B Traffic Model") you see that for 28 Erlangs of traffic with a retry probability of 50 percent and 4.05 percent blockage, you need 35 circuits to handle this traffic load.

Again, there is a handy Erlang B, Extended Erlang B, and Erlang C calculator at the following URL: www.erlang.com/calculator/index.htm.

Erlang C Traffic Model

The Erlang C model is based on the following assumptions:

- An infinite number of sources.
- Random traffic arrival pattern.
- Blocked calls are delayed.
- Hold times are exponentially distributed.

The Erlang C model is designed around queuing theory. This model assumes a random call arrival pattern; the caller makes one call and is held in a queue until the call is answered. The Erlang C model is more commonly used for conservative automatic call distributor (ACD) design to determine the number of agents needed. It can also be used for determining bandwidth on data transmission circuits, but it is not the best model to use for that purpose.

In the Erlang C model, you need to know the number of calls or packets in the busy hour, the average call length or packet size, and the expected amount of delay in seconds.

Equation 1-2 provides the formula used to derive the Erlang C traffic model.

Equation 1-2

$$C(c, a) = \frac{\dfrac{a^{c}c}{c!(c-a)}}{\displaystyle\sum_{k=0}^{c-1} \frac{a^{k}}{k!} + \frac{a^{c}c}{c!(c-a)}}$$

where:

- C(c,a) is the probability of delaying.
- c is the number of circuits.
- a is the traffic load.

Example: Using the Erlang C Traffic Model for Voice

Problem: You expect the call center to have approximately 600 calls lasting approximately 3 minutes each and that each agent has an after-call work time of 20 seconds. You would like the average time in the queue to be approximately 10 seconds.

Solution: Calculate the amount of expected traffic load. You know that you have approximately 600 calls of 3 minutes duration. To that number, you must add 20 seconds because each agent is not answering a call for approximately 20 seconds. The additional 20 seconds is part of the amount of time it takes to service a call:

(600 calls × 200 seconds AHT) / 3600 = 33.33 Erlangs of traffic

Compute the delay factor by dividing the expected delay time by AHT:

10 sec delay / 200 seconds = 0.05 delay factor

Example: Using the Erlang C Traffic Model for Data

Problem: You are designing your backbone connection between two routers. You know that you will generally see about 600 packets per second and 200 bytes per packet or 1600 bits per packet. Multiplying 600 pps by 1600 bits per packet gives the amount of bandwidth you will need to support—960,000 bps. You know that you can buy circuits in increments of 64,000 bps, the amount of data necessary to keep the circuit busy for 1 second. How many circuits will you need to keep the delay under 10 milliseconds?

Solution: Calculate the traffic load as follows:

960,000 bps / 64,000 bps = 15 Erlangs of traffic load

To get the average transmission time, you need to multiply the number of bytes per packet by 8 to get the number of bits per packet, then divide that by 64,000 bps (circuit speed) to get the average transmission time per packet:

200 bytes / packet × 8 bits = 1600 bits per packet / 64,000 bps =
0.025 seconds to transmit, or 25 milliseconds
Delay factor 10 ms / 25 ms = 0.4 delay factor

With a delay factor of 0.4 and a traffic load of 15.47 Erlangs, the number of circuits you need is 17. This calculation is based on the assumption that the circuits are clear of any packet loss.

Again, there is a handy Erlang B, Extended Erlang B, and Erlang C calculator at the following URL: www.erlang.com/calculator/index.htm.

Engset Traffic Model

The Engset model is based on the following assumptions:

- A finite number of sources.
- Smooth traffic arrival pattern.
- Blocked calls are cleared from the system.
- Hold times are exponentially distributed.

The Engset formula is generally used for environments where it is easy to assume that a finite number of sources are using a trunk group. By knowing the number of sources, you can maintain a high grade of service. You would use the Engset formula in applications such as global system for mobile communication (GSM) cells and subscriber loop concentrators. Because the Engset traffic model is covered in many books dedicated to traffic analysis, it is not covered here.

Poisson Traffic Model

The Poisson model is based on the following assumptions:

- An infinite number of sources.
- Random traffic arrival pattern.
- Blocked calls are held.
- Hold times are exponentially distributed.

In the Poisson model, blocked calls are held until a circuit becomes available. This model assumes a random call arrival pattern; the caller makes only one attempt to place the call and blocked calls are lost. The Poisson model is commonly used for over-engineering standalone trunk groups.

Equation 1-3 provides the formula used to derive the Poisson traffic model.

Equation 1-3

$$P(c, a) = \left(1 - e^{-a} \sum_{k=0}^{c-1} \frac{a^k}{k!} \right)$$

where:

- P(c,a) is the probability of blocking the call.
- e is the natural log base.
- c is the number of circuits.
- a is the traffic load.

Example: Using the Poisson Traffic Model

Problem: You are creating a new trunk group to be utilized only by your new office and you need to figure out how many lines are needed. You expect them to make and receive approximately 300 calls per day with an AHT of about 4 minutes or 240 seconds. The goal is a P.01 Grade of Service or a 1 percent blocking rate. To be conservative, assume that approximately 20 percent of the calls happen during the busy hour.

300 calls × 20% = 60 calls during the busy hour.
(60 calls × 240 AHT) / 3600 = 4 Erlangs during the busy hour.

Solution: With 4 Erlangs of traffic and a blocking rate of 0.81 percent (close enough to 1 percent), you need 10 trunks to handle this traffic load. You can check this number by plugging the variables into the Poisson formula, as demonstrated in Equation 1-4.

Equation 1-4

$$P(10,4) = 1 - e^{-4}\sum_{k=0}^{10-1}\frac{4^k}{k!} = 1 - e^{-4}\left(1 + 4 + \frac{16}{2} + \frac{64}{6} + \frac{256}{24} + \ldots\right) \approx 0.00813$$

Another easy way to find blocking is by using Microsoft Excel's Poisson function with the following format:

= 1 – POISSON(<circuits>–1,<traffic load>,TRUE)

EART/EARC and Neal-Wilkerson Traffic Model

These models are used for peaked traffic patterns. Most telephone companies use these models for rollover trunk groups that have peaked arrival patterns. The EART/EARC model treats blocked calls as cleared and the Neal-Wilkinson model treats them as held. Because the EART/EARC and Neal-Wilkerson traffic models are covered in many books dedicated to traffic analysis, they are not covered here.

Applying Traffic Analysis to VoIP Networks

Because Voice over IP (VoIP) traffic uses Real-Time Transport Protocol (RTP) to transport voice traffic, you can use the same principles to define your bandwidth on your WAN links.

Some challenges exist in defining the bandwidth. The following considerations will affect the bandwidth of voice networks:

- Voice codecs
- Samples
- Voice activity detection (VAD)
- RTP header compression
- Point-to-point versus point-to-multipoint

Voice Codecs

Many voice codecs are used in IP telephony today. These codecs all have different bit rates and complexities. Some of the standard voice codecs are G.711, G.729, G.726, G.723.1, and G.728. All Cisco voice-enabled routers and access servers support some or all of these codecs.

Codecs impact bandwidth because they determine the payload size of the packets transferred over the IP leg of a call. In Cisco voice gateways, you can configure the payload size to control bandwidth. By increasing payload size, you reduce the total number of packets sent, thus decreasing the bandwidth needed by reducing the number of headers required for the call.

Samples

The number of samples per packet is another factor in determining the bandwidth of a voice call. The codec defines the size of the sample, but the total number of samples placed in a packet affects how many packets are sent per second. Therefore, the number of samples included in a packet affects the overall bandwidth of a call.

For example, a G.711 10-ms sample is 80 bytes per sample. A call with only one sample per packet would yield the following:

80 bytes + 20 bytes IP + 12 UDP + 8 RTP = 120 bytes/packet
120 bytes/packet × 100 pps = 12,000 × 8 bits / 1000 = 96 kbps per call

The same call using two 10-ms samples per packet would yield the following:

(80 bytes × 2 samples) + 20 bytes IP + 12 UDP + 8 RTP = 200 bytes/packet
200 bytes/packet × 50 pps = 10,000 × 8 bits / 1000 = 80 kbps per call

Layer 2 headers are not included in the preceding calculations.

The results show that a 16-kbps difference exists between the two calls. By changing the number of samples per packet, you definitely can change the amount of bandwidth a call uses, but there is a trade-off. When you increase the number of samples per packet, you also increase the amount of delay on each call. DSP resources, which handle each call, must buffer the samples for a longer period of time. You should keep this in mind when you design a voice network.

Voice Activity Detection

Typical voice conversations can contain up to 50 percent silence. With traditional, circuit-based voice networks, all voice calls use a fixed bandwidth of 64 kbps, regardless of how much of the conversation is speech and how much is silence. With VoIP networks, all conversation and silence is packetized. Voice Activity Detection (VAD) enables you to send RTP packets only when voice is detected. For VoIP bandwidth planning, assume that VAD reduces bandwidth by 35 percent. Although this value might be less than the actual reduction, it provides a conservative estimate that takes into consideration different dialects and language patterns.

The G.729 Annex-B and G.723.1 Annex-A codecs include an integrated VAD function, but otherwise have identical performance to G.729 and G.723.1, respectively.

RTP Header Compression

All VoIP packets are made up of two components: voice samples and IP/UDP/RTP headers. Although the voice samples are compressed by the digital signal processor (DSP) and vary in size based on the codec used, the headers are always a constant 40 bytes. When compared to the 20 bytes of voice samples in a default G.729 call, these headers make up a considerable amount of overhead. Using RTP Header Compression (cRTP), which is used on a link-by-link basis, these headers can be compressed to 2 or 4 bytes. This compression can offer significant VoIP bandwidth savings. For example, a default G.729 VoIP call consumes 24 kbps without cRTP, but only 12 kbps with cRTP enabled. Codec type, samples per packet, VAD, and cRTP affect, in one way or another, the bandwidth of a call. In each case, there is a trade-off between voice quality and bandwidth. Table 1-4 shows the bandwidth utilization for various scenarios. VAD efficiency in the graph is assumed to be 50 percent.

Table 1-4 *Voice Codec Characteristics*

Algorithm	Voice BW (kbps)	FRAME SIZE (Bytes)	Cisco Payload (Bytes)	Packets Per Second (PPS)	IP/UDP/RTP Header (Bytes)	CRTP Header (Bytes)	L2	Layer2 header (Bytes)	Total Bandwidth (kbps) no VAD	Total Bandwidth (kbps) with VAD
G.711	64	80	160	50	40		Ether	14	85.6	42.8
G.711	64	80	160	50		2	Ether	14	70.4	35.2
G.711	64	80	160	50	40		PPP	6	82.4	41.2
G.711	64	80	160	50		2	PPP	6	67.2	33.6
G.711	64	80	160	50	40		FR	4	81.6	40.8
G.711	64	80	160	50		2	FR	4	66.4	33.2
G.711	64	80	80	100	40		Ether	14	107.2	53.6
G.711	64	80	80	100		2	Ether	14	76.8	38.4
G.711	64	80	80	100	40		PPP	6	100.8	50.4
G.711	64	80	80	100		2	PPP	6	70.4	35.2
G.711	64	80	80	100	40		FR	4	99.2	49.6
G.711	64	80	80	100		2	FR	4	68.8	34.4
G.729	8	10	20	50	40		Ether	14	29.6	14.8
G.729	8	10	20	50		2	Ether	14	14.4	7.2
G.729	8	10	20	50	40		PPP	6	26.4	13.2
G.729	8	10	20	50		2	PPP	6	11.2	5.6
G.729	8	10	20	50	40		FR	4	25.6	12.8
G.729	8	10	20	50		2	FR	4	10.4	5.2
G.729	8	10	30	33	40		Ether	14	22.4	11.2
G.729	8	10	30	33		2	Ether	14	12.3	6.1
G.729	8	10	30	33	40		PPP	6	20.3	10.1
G.729	8	10	30	33		2	PPP	6	10.1	5.1

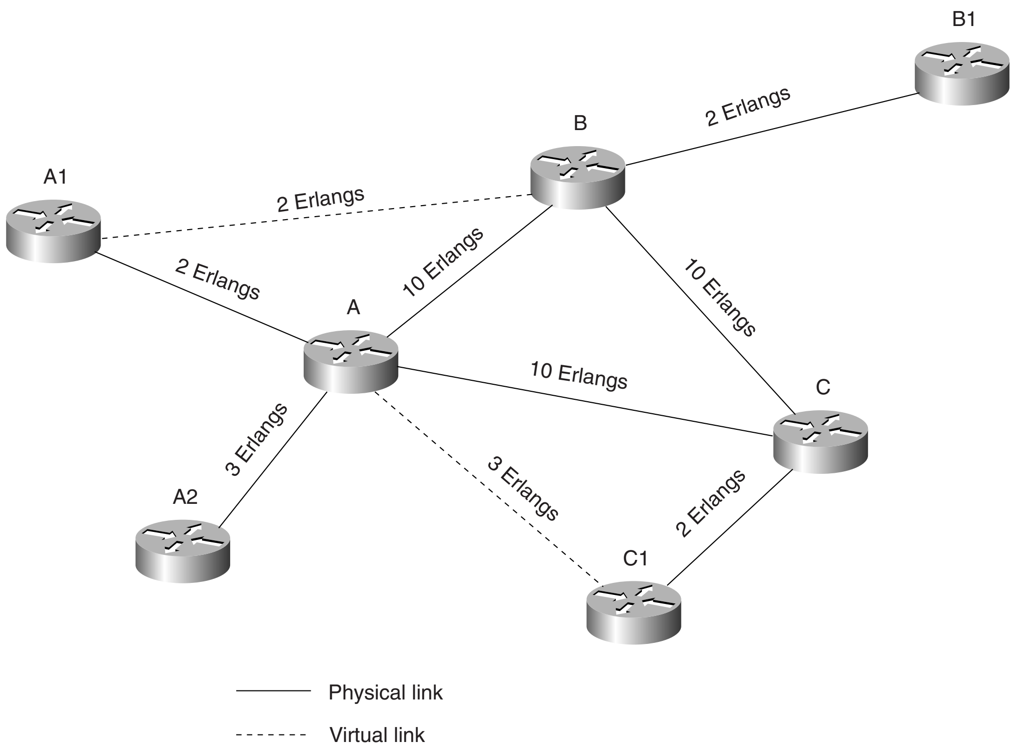
Table 1-4 *Voice Codec Characteristics (Continued)*

Algorithm	Voice BW (kbps)	FRAME SIZE (Bytes)	Cisco Payload (Bytes)	Packets Per Second (PPS)	IP/UDP/ RTP Header (Bytes)	CRTP Header (Bytes)	L2	Layer2 header (Bytes)	Total Bandwidth (kbps) no VAD	Total Bandwidth (kbps) with VAD
G.729	8	10	30	33	40		FR	4	19.7	9.9
G.729	8	10	30	33		2	FR	4	9.6	4.8
G.723.1	6.3	30	30	26	40		Ether	14	17.6	8.8
G.723.1	6.3	30	30	26		2	Ether	14	9.7	4.8
G.723.1	6.3	30	30	26	40		PPP	6	16.0	8.0
G.723.1	6.3	30	30	26		2	PPP	6	8.0	4.0
G.723.1	6.3	30	30	26	40		FR	4	15.5	7.8
G.723.1	6.3	30	30	26		2	FR	4	7.6	3.8
G.723.1	5.3	30	30	22	40		Ether	14	14.8	7.4
G.723.1	5.3	30	30	22		2	Ether	14	8.1	4.1
G.723.1	5.3	30	30	22	40		PPP	6	13.4	6.7
G.723.1	5.3	30	30	22		2	PPP	6	6.7	3.4
G.723.1	5.3	30	30	22	40		FR	4	13.1	6.5
G.723.1	5.3	30	30	22		2	FR	4	6.4	3.2

Point-to-Point Versus Point-to-Multipoint

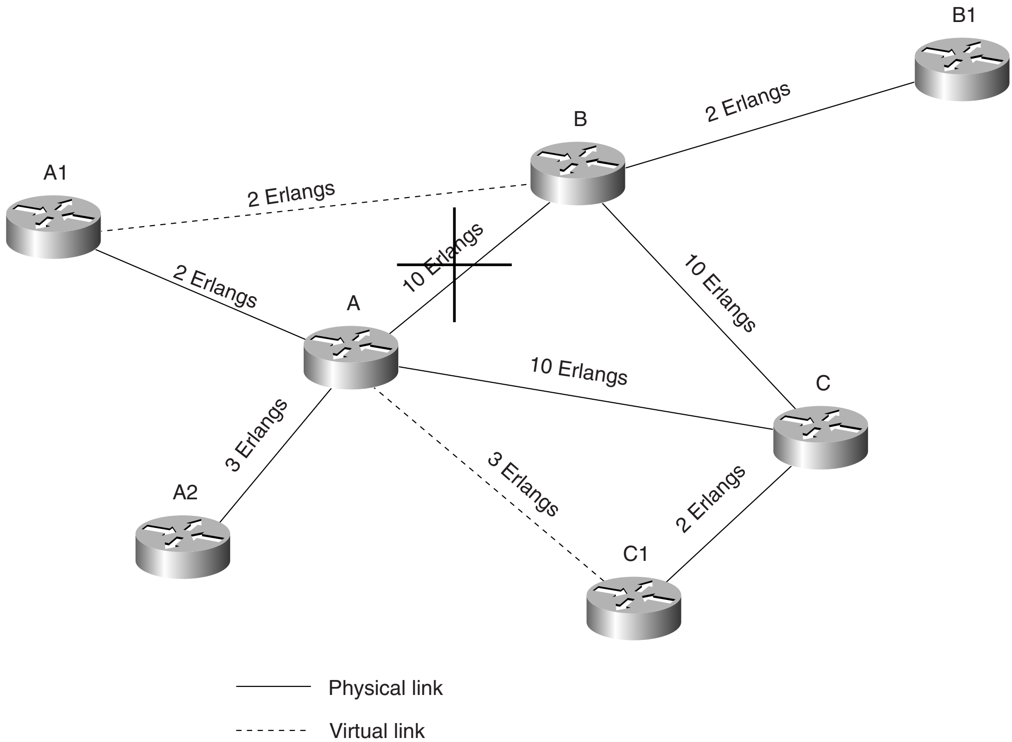
Because PSTN circuits are built as point-to-point links, and VoIP networks are basically point-to-multipoint, you must take into account where your traffic is going and group it accordingly. This becomes more of a factor when deciding bandwidth on fail-over links. Figure 1-4 shows the topology of a properly functioning voice network.

Figure 1-4 *Properly functioning topology.*



Point-to-point links will not need more bandwidth than the number of voice calls being introduced to and from the PSTN links, although as you approach link speed, voice quality may suffer. If one of those links is lost, you need to ensure that your fail-over links have the capacity to handle the increased traffic. In Figure 1-5, the WAN link between nodes A and B is down. Traffic would then increase between nodes A and C, and between C and B. This additional traffic would require that those links be engineered to handle the additional load.

Figure 1-5 *Topology with broken connection.*



End-to-End Traffic Analysis Example

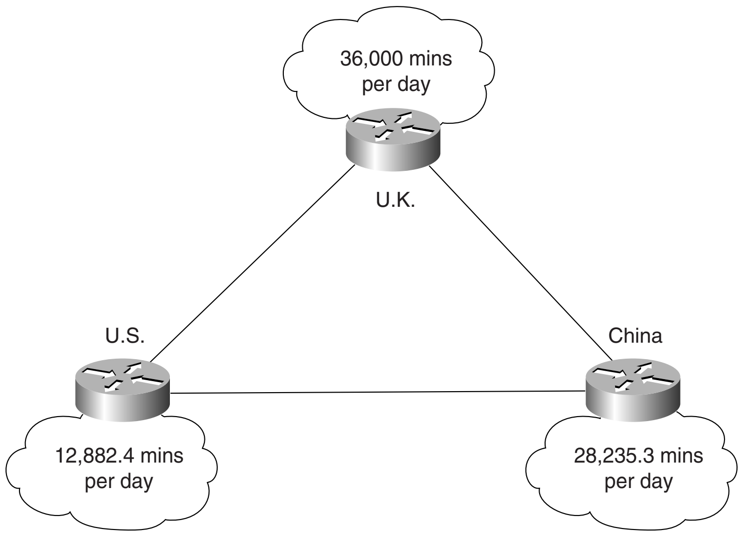
With the proper traffic tables, defining the number of circuits needed to handle calls becomes fairly simple. By defining the number of calls on the TDM side, you can also define the amount of bandwidth needed on the IP leg of the call. Unfortunately, putting them together can be an issue.

End-to-End Traffic Analysis: Problem

As illustrated in Figure 1-6, you have offices in the U.S., China, and the U.K. Because your main office is in the U.K., you will purchase leased lines from the U.K. to the U.S. and to China. Most of your traffic goes from the U.K. to the U.S. or China, with a little traffic going between China and the U.S. Your call detail records show:

- U.K. 36,000 minutes/day
- U.S. 12,882.4 minutes/day
- China 28,235.3 minutes/day

Figure 1-6 *End-to-end traffic analysis example topology.*



In this network, you are making the following assumptions:

- Each node's traffic has a random arrival pattern.
- Hold times are exponential.
- Blocked calls are cleared from the system.
- Infinite number of callers.

These assumptions tell you that you can use the Erlang B model for sizing your trunk groups to the PSTN. You want to have a GoS of P.01 on each of your trunk groups.

End-to-End Traffic Analysis: Solution

Compute the traffic load for the PSTN links at each node:

U.K. = 36,000 mins per day × 17% = 6120 mins per busy hour / 60 = 102 BHT
U.S. = 12,882.4 mins per day × 17% = 2190 mins per busy hour / 60 = 36.5 BHT
China = 28,235.3 mins per day × 17% = 4800 mins per busy hour / 60 = 80 BHT

These numbers will effectively give you the number of circuits needed for your PSTN connections in each of the nodes. Now that you have a usable traffic number, look in your tables to find the closest number that matches.

For the U.K., a 102 BHT with P.01 GoS indicates the need for a total of 120 DS-0s to support this load.

U.S. traffic shows that for P.01 blocking with a traffic load of 36.108, you need 48 circuits. Because your BHT is 36.5 Erlangs, you might experience a slightly higher rate of blocking than P.01. By using the Erlang B formula, you can see that you will experience a blocking rate of ~0.01139.

At 80 Erlangs of BHT with P.01 GoS, the Erlang B table (see Appendix A) shows you that you can use one of two numbers. At P.01 blocking you can see that 80.303 Erlangs of traffic requires 96 circuits. Because circuits are ordered in blocks of 24 or 30 when working with digital carriers, you must choose either 4 T1s or 96 DS-0s, or 4 E1s or 120 DS-0s. Four E1s is excessive for the amount of traffic you will be experiencing, but you know you will meet your blocking numbers. This gives you the number of circuits you will need.

Now that you know how many PSTN circuits you need, you must determine how much bandwidth you will have on your point-to-point circuits. Because the amount of traffic you need on the IP leg is determined by the amount of traffic you have on the TDM leg, you can directly relate DS-0s to the amount of bandwidth needed.

You must first choose a codec that you are going to use between PoPs. The G.729 codec is the most popular because it has high voice quality for the amount of compression it provides.

A G.729 call uses the following bandwidth:

- 26.4 kbps per call full rate with headers
- 11.2 kbps per call with VAD
- 9.6 kbps per call with cRTP
- 6.3 kbps per call with VAD and cRTP

Table 1-5 lists the bandwidth needed on the link between the U.K. and the U.S.

Table 1-5 *Bandwidth Requirements for U.K.–U.S. Link*

Bandwidth Consideration	Full Rate	VAD	cRTP	VAD/cRTP
Bandwidth Required	96 DS0s × 26.4 kbps = 2.534 Mbps	96 DS0s × 11.2 kbps = 1.075 Mbps	96 DS0s × 17.2 kbps = 1.651 Mbps	96 DS0s × 7.3 kbps = 700.8 kbps

Table 1-6 lists the bandwidth needed on the link between the UK and China.

Table 1-6 *Bandwidth Requirements for U.K.–China Link*

Bandwidth Consideration	Full Rate	VAD	cRTP	VAD/cRTP
Bandwidth Required	72 DS0s × 26.4 kbps = 1.9 Mbps	72 DS0s × 11.2 kbps = 806.4 kbps	72 DS0s × 17.2 kbps = 1.238 kbps	72 DS0s × 7.3 kbps = 525.6 kbps

As you can see, VAD and cRTP have a significant impact on the bandwidth needed on the WAN link.

Summary

This chapter covered the various traffic measurement techniques and sampling methods you can use to select the appropriate traffic model to help you engineer and properly size a traffic-sensitive voice network. The chapter explained how to calculate traffic load in Erlangs and in CCS. The chapter discussed the key voice network characteristics that determine which traffic model is appropriate for a particular network. Finally, you saw a description of the Erlang B, Extended Erlang B, Erlang C, and Poisson traffic models. This chapter included examples of specific network design problems that can be solved using these models.

For additional information about traffic analysis, see the following:

Martine, Roberta R., *Basic Traffic Analysis.* Englewood Cliffs, NJ:
Prentice Hall, Inc.; 1994

Harder, J., Alan Wand, and Pat J. Richards, Jr. *The Complete Traffic Engineering Handbook.* New York, NY: Telecom Library, Inc.

Newton, H. *Newton's Telecom Directory.* New York, NY Miller Freeman, Inc.

Sizing Trunk Groups, Crawley, West Sussex RH10 7JR, United Kingdom: Westbay Engineers Ltd., 1999. http://www. erlang.com/link_traffic.html

Chapter 2

Understanding Echo Analysis

In a voice call, an echo occurs when you hear your own voice repeated. An echo is the audible leak-through of your own voice into your own receive (return) path. This chapter discusses basic concepts applicable to echo analysis, explains echo cancellation, and provides a process for locating and eliminating echoes.

Echo Analysis Basics

Every voice conversation has at least two participants. From each participant's perspective, every call contains two voice paths:

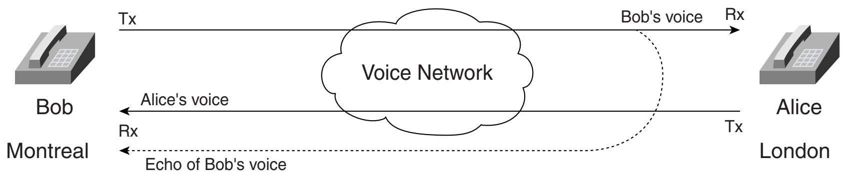
- **Transmit path**—The transmit path is also called the send or Tx path. In a conversation, the transmit path is created when a person speaks. The sound is transmitted from the speaker's mouth to the listener's ear.
- **Receive path**—The receive path is also called the return or Rx path. In a conversation, the receive path is created when a person hears the conversation. The sound is received by the listener's ear from the speaker's mouth.

Figure 2-1 shows a simple voice call between Bob and Alice. From Bob's perspective, the transmit path carries his voice to Alice's ear, and the receive path carries Alice's voice to his ear. Naturally, from Alice's side these paths have the opposite naming convention: The transmit path carries her voice to Bob's ear, and the receive path carries Bob's voice to her ear.

Figure 2-1 *Simple telephone call.*

As previously mentioned, an echo is the audible leak-through of your own voice into your own receive (return) path. Figure 2-2 shows the same simple telephone call where Bob hears an echo.

Figure 2-2 *Simple telephone call with an echo.*



Bob hears a delayed and somewhat attenuated version of his own voice in the earpiece of his handset. Initially, the source and mechanism of the leak are undefined.

One of the key factors in echo analysis is the round-trip delay of the voice network. The round-trip delay of the network is the length of time it takes for an utterance to go from Bob's mouth, across the network on the transmit path to the source of the leak, and then back across the network on the receive path to Bob's ear.

Two basic characteristics of echo are the following:

- The louder the echo (the greater the echo amplitude), the more annoying it is.
- The later the echo (the longer the round-trip voice delay), the more annoying it is.

Locating an Echo

In Figure 2-2, Bob experiences the echo problem, which means that a signal is leaking from his transmit path into his receive path. This illustrates one of the basic properties of echo: Whenever you hear echo, the problem is at the other end. The problem that's producing the echo that Bob hears—the leakage source—is somewhere on Alice's side of the network (London). If Alice were the person experiencing the echo, the problem would be on Bob's side (Montreal).

The echo leak is always in the terminating side of the network because of the following:

- Leak-through happens only in analog circuits. Voice traffic in the digital portions of the network doesn't leak from one path into another.

  Analog signals can leak from one path to another, either electrically from one wire to another, or acoustically through the air from a loudspeaker to a microphone. When these analog signals have been converted to digital bits, they don't leak.

  It is true that all digital bits are represented by analog signals at the physical layer and these analog signals are subject to leakage. The analog signals that represent bits can tolerate a good deal of distortion before they become too distorted to be properly decoded. If such distortion occurred in the physical layer, the problem wouldn't be echo. If you had connectivity at all, you would hear digital noise instead of a voice echo.

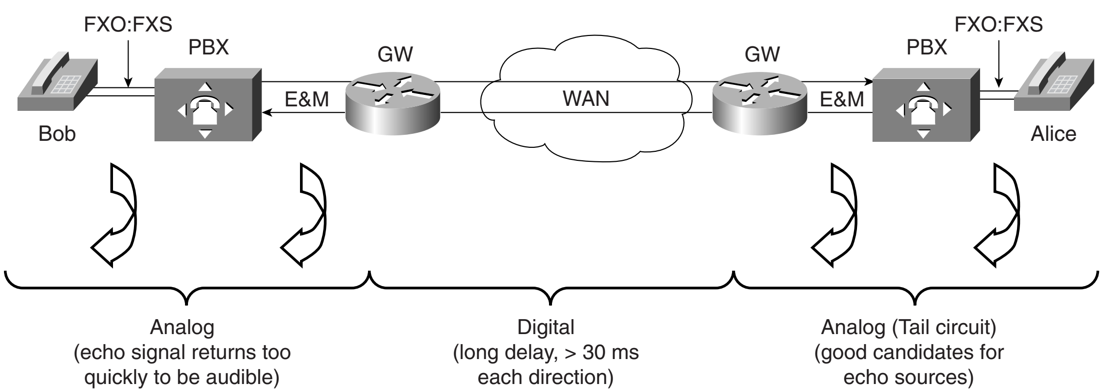
- Echoes arriving after short delays (about 20 ms) are generally imperceptible because they're masked by the physical and electrical sidetone signal.

  This point is a corollary to the previous assertion that echoes become increasingly annoying with increasing mouth-to-ear delay. A certain minimum delay is needed for an echo to become perceptible. In almost every telephone device, some of the Tx signal is fed back into the earpiece so that you can hear yourself speaking. This is known as *sidetone*. The delay between the actual mouth signal and the sidetone signal is negligible, and sidetone is not perceived as an echo.

  Also, your skull resonates during speech (an acoustic sidetone source) and the human auditory system has a certain integration period that determines the minimum time difference between events that will be perceived as separate events rather than a single one. Together, these phenomena create a minimum mouth-to-ear delay of about 20 ms for an echo signal to be perceivable.

Given these two premises—that echoes must be delayed by at least 20 ms to be audible and that leaks occur only in the analog portion of the network—you can deduce much about the location of the echo source. Figure 2-3 shows possible sources of echo in a simple VoIP network.

Figure 2-3 *Potential echo paths in a network with both analog and digital segments.*

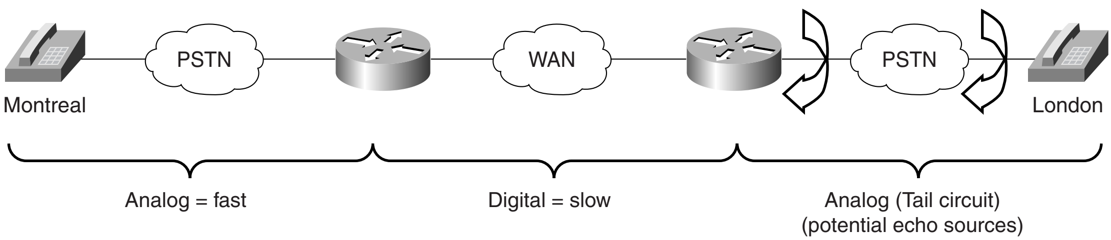


In this typical VoIP network, the digital packet portion of the network is sandwiched between two analog transmission segments. Bob in Montreal is connected by FXS (2-wire analog) to a local PBX, which is connected to a local VoIP gateway by E&M (4-wire analog). The Montreal gateway communicates with the London gateway through an IP network. As you will see later in this section, this packet transmission segment has an end-to-end latency greater than 30 ms. At the London end of the call, the gateway is connected in the same fashion to Alice's telephone (by E&M to the PBX and by FXS to the terminal).

The analog circuit in London is known as the *tail circuit*. It forms the tail or termination of the call from the user experiencing the echo, which in this case, is Bob.

Suppose that you want to locate potential sources of echo in the network in Figure 2-3. You know that bits don't leak, so you can disqualify the digital segment of the system. Therefore, the leak causing Bob's echo must be located in either the tail circuit in Montreal or the tail circuit in London. Any leak in the Montreal tail circuit would not have a long enough delay to be perceptible; echoes there would be masked by Bob's sidetone. So the source of the echo must be the London tail circuit, as shown in Figure 2-4.

Figure 2-4 *Simplified version of the VoIP network.*



Remember that an echo problem has three ingredients:

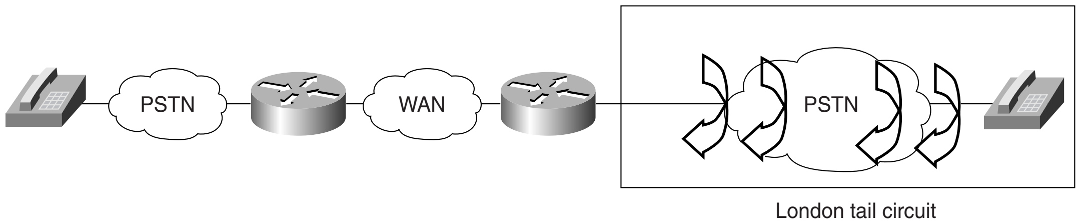
- An analog leakage path between analog Tx and Rx paths
- Sufficient delay in echo return for echo to be perceived as annoying
- Sufficient echo amplitude to be perceived as annoying

The packet link in Figures 2-3 and 2-4 is called *slow* because it takes a relatively long time for analog signals entering this link to exit from the other side: the end-to-end delay of the link. This delay occurs because packet transmission fundamentally imposes a packetization and buffering delay of *at least* two to three packet sizes, and packet sizes of 20 ms are typical for VoIP. Assuming for the moment that the WAN link imposes an end-to-end delay of 50 ms, you can see that Bob's voice takes 50 ms to cross the transmit path to Alice in London. The echo that leaks from the transmit path to the receive path in the London tail circuit takes another 50 ms to make it back to Bob's ear. Therefore, the echo that Bob hears is delayed at least 100 ms, well into the range of audibility.

Tail Circuits

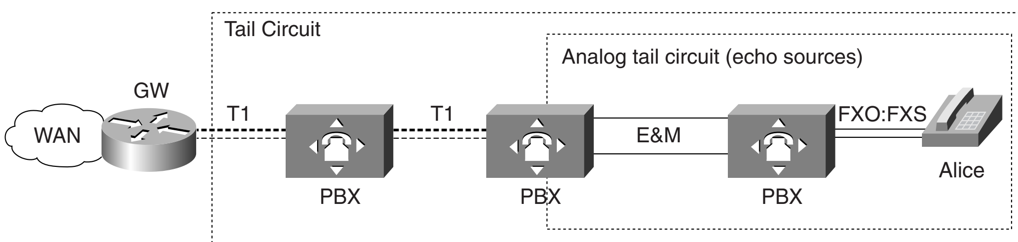
A packet voice gateway is a gateway between a digital packet network and a PSTN network. It can include both digital (TDM) and analog links. The tail circuit is everything connected to the PSTN side of a packet voice gateway—all the switches, multiplexers, cabling, PBXs—everything between the voice gateway and the telephone as demonstrated in Figure 2-5. The PSTN can contain many components and links, all of which are potential echo sources.

Figure 2-5 *Tail circuit in a VoIP network.*



Gateways have two types of PSTN interfaces: digital (ISDN BRI, T1/E1) or analog (E&M, FXO, FXS). Recalling that bits don't leak, further refine your search for echo sources to the analog elements of the tail circuit. You can extend the echo-free digital zone out from the gateway to the point of digital-to-analog (D/L) conversion in the PSTN, as shown in Figure 2-6.

Figure 2-6 *Tail circuit with both analog and digital links.*



Effects of Network Elements on Echo

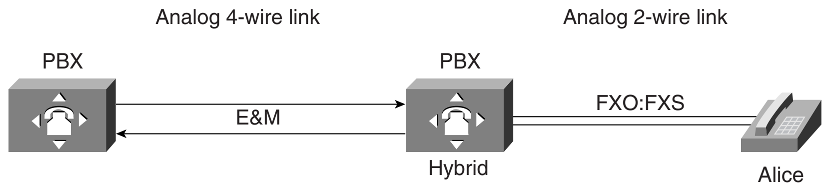
The following network elements in a VoIP network can have an effect on echo:

- Hybrid transformers
- Telephones
- Routers
- Quality of service (QoS)

Effect of Hybrid Transformers on Echo

Echo sources are points of signal leakage between analog transmit and receive paths. Hybrid transformers are often prime culprits for this signal leakage. Figure 2-7 shows an analog tail circuit with a hybrid transformer.

Figure 2-7 *Detail of analog tail circuit with a hybrid transformer.*



The analog telephone terminal is a 2-wire device, with a single pair of conductors used to carry both the Tx and Rx signals. For analog trunk connections, known as 4-wire transmission, two pairs of conductors carry separate Tx and Rx signals. Digital trunks (T1/E1) can be considered virtual 4-wire links because they also carry separate Tx and Rx signals.

A hybrid is a transformer that is used to interface 4-wire links to 2-wire links. It is a non-ideal physical device, and a certain fraction of the 4-wire incoming (Rx) signal will be reflected back into the 4-wire outgoing (Tx) signal. A typical fraction for a properly terminated hybrid is about –25 dB (ERL = +25 dB). This means that the reflected signal (the echo) will be a version of the Rx signal attenuated by about 25 dB. Remember, an echo must have both sufficient amplitude and sufficient delay to be perceived. Echo strength of –25 dB relative to the talker's speech level is generally quiet enough to not be annoying, even for relatively long delays of 100 ms.

Echo strength is expressed in decibels (dB) as a measurement called *echo return loss* (ERL). The relation between the original source and the ERL is as follows:

Original source amplitude = Echo amplitude + ERL

Therefore, an ERL of 0 dB indicates that the echo is the same amplitude as the original source. A large ERL indicates a negligible echo.

The ERL is not a property of the hybrid alone, however. It depends on the load presented by the terminating device, which might be a telephone or another PBX. The hybrid has a certain output impedance that must be balanced by the input impedance of the terminating device. If the impedances are not matched, the returning echo fraction will be larger (the ERL will be smaller) and the echo will be louder.

You can expect a certain amount of impedance mismatch (a few tens of ohms) because a normal hybrid connection will yield ERLs in the range of 20 to 30 dB. However, it is possible that one device could be provisioned for an output impedance of 900 ohms, and the terminating device provisioned with an input impedance of 600 ohms, which would yield a large echo, and would be expressed by a small ERL.

The main point to remember about hybrids is this: Ensure that output and input impedances are matched between the hybrid and the terminating device.

Effects of Telephones on Echo

Once again, the analog tail circuit is the portion of the PSTN circuit between the point of digital-to-analog conversion and the telephone terminal. By using digital telephones, this point of D/A conversion occurs inside the terminal itself. As a general rule, extending the digital transmission segments closer to the actual telephone will decrease the potential for echo.

The analog telephone terminal itself presents a load to the PBX. This load should be matched to the output impedance of the source device (FXS port). Some (inexpensive) telephones are not matched to the output impedance of the FXS port and are sources of echo. Headsets are particularly notorious for poor echo performance.

Acoustic echo is a major concern for hands-free speakerphone terminals. The air (and the terminal plastics) provide mechanical or acoustical coupling between the loudspeaker and the microphone. Speakerphone manufacturers combat this with good acoustic design of terminals, directional microphones, and acoustic echo cancellers/suppressors in the terminal. However, this is a very difficult problem, and speakerphones are inherently good echo sources. If you are hunting for an echo problem and the terminating tail circuit involves a speakerphone, eliminate the speakerphone.

Effects of Routers on Echo

The belief that adding routers to a voice network creates echoes is a common misconception. Digital segments of the network do not cause leaks; so technically, routers cannot be the source of echoes. Adding routers to the network, though, adds delays to the network—delays that can make a previously imperceptible echo perceptible. The gateway itself doesn't add echo unless you are using an analog interface to the PSTN and the output impedance is incorrectly provisioned with respect to the PBX. It is more likely that the echo was already in the analog tail circuit but was imperceptible because the round-trip delay was less than 20 ms.

For example, suppose that you are visiting London and you want to call a friend who lives on the other side of town. This call is echo free. But when you call the same friend (whose telephone is on the same tail circuit) from the U.S. over a satellite link with a round-trip delay of several hundred milliseconds, the echo is obvious and annoying. The only change has been the insertion of delay.

VoIP technologies impose a fundamental transmission delay due to packetization and the buffering of received packets before playout at the receiving endpoint. This delay is generally much smaller than the delay associated with satellite links, but it is usually sufficient to make a previously unnoticeable echo objectionable.

End-to-End Voice Call Delays

Analog transmission is very fast, limited only by the propagation speed of electrons in a wire (which is much lower than the speed of light, but still very fast) or photons in a fiber-optic link. TDM transmission is similarly very quick. A transcontinental PSTN call in the U.S. has a typical round-trip delay of about 10 to 20 ms. A local PSTN call has a typical round-trip delay of only a few milliseconds. Such short delays mean that even relatively loud echoes in the PSTN remain imperceptible as echo because they are masked by sidetone.

Imagine a call between Bob and Alice over a VoIP transmission link as in Figure 2-3. Consider the path Bob's voice takes from Montreal to London. Bob speaks into his mouthpiece and the analog signal arrives at the Montreal PBX within 1 ms. At the PBX, his analog voice signal is converted to a digital PCM stream and arrives at the Montreal IP gateway after only 1 ms more of delay. So it takes 2 ms for Bob's voice to go from his mouth to the voice gateway. The gateway sends out packets every 20 ms, which means each packet contains 20 ms of voice payload. Therefore, the voice gateway must wait to collect 20 ms of Bob's voice before it can fill the first packet. The first packet leaves the Montreal gateway 22 ms after Bob starts talking. Assuming that the WAN is very quick and uncongested, this packet arrives at the London voice gateway after only 5 ms of transit. So the London gateway gets the packet 27 ms after Bob starts speaking.

This packet is not played out from the London gateway to Alice immediately upon receipt, however. The Montreal gateway delivers new packets at 20 ms intervals, but the vagaries of packet transmission mean that packets arrive in London at non-constant intervals: Packet 2 might be 1 ms late, packet 3 might be 4 ms late, and so on. If the London gateway played out packet 1 immediately, it would be caught short 20 ms later when packet 2 was due but had not yet arrived—and Bob's voice would be interrupted.

The London gateway puts incoming packets into a buffer. The deeper the playout buffer, the longer packets wait before being played. The minimum buffer depth you can safely use is one packet, or 20 ms in this case. So packet 1 arrives at time 27 ms and is played out to the London PSTN tail 20 ms later at time 47 ms. It takes two more milliseconds to go from the London gateway across the PSTN to Alice's earpiece, for a total of 49 ms for Bob's words to go from Bob's mouth to Alice's ear. This is the end-to-end delay of the voice transmission system: 45 ms in the WAN and 4 ms in the PSTN.

You could increase the packet transmission rate to reduce the end-to-end delay, but this would increase the bandwidth necessary for the call because it would increase the ratio of header size (which is a constant) to payload size (which you would reduce).

As a general rule, the end-to-end latency for a packet transmission link has a fundamental minimum of about two to three packet sizes (in milliseconds). Even if the packet transit time was instantaneous, it still takes one packet size of time to fill the first packet. Even an unrealistically ideal, "fast-as-light" gateway and network face this fundamental, minimum delay.