IP Design for Mobile Networks

Revolutionizing the architecture and implementation of mobile networks

Mark Grayson
Kevin Shatzkamer
Scott Wainner

ciscopress.com

IP Design for Mobile Networks

Mark Grayson
Kevin Shatzkamer
Scott Wainner

Cisco Press
800 East 96th Street
Indianapolis, IN 46240

IP Design for Mobile Networks

Mark Grayson, Kevin Shatzkamer, Scott Wainner

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

Printed in the United States of America

Second Printing April 2012

Library of Congress Cataloging-in-Publication Data

Grayson, Mark, 1965-
IP design for mobile networks / Mark Grayson, Kevin Shatzkamer, Scott Wainner.
p. cm.
ISBN-13: 978-1-58705-826-4 (pbk.)
ISBN-10: 1-58705-826-X (pbk.)
1. Wireless LANs. 2. Wireless Internet. 3. Mobile computing. 4. TCP/IP (Computer network protocol) I. Shatzkamer, Kevin, 1978- II. Wainner, Scott. III. Title.

TK5105.78.G73 2009
621.382'12--dc22

2009020541

ISBN-13: 978-1-58705-826-4

ISBN-10: 1-58705-826-X

Warning and Disclaimer

This book is designed to provide information about the evolution of mobile technologies and networks to the All-IP architecture. 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

Corporate and Government Sales

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

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

Feedback Information

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

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

We greatly appreciate your assistance.

Publisher: Paul Boger

Associate Publisher: Dave Dusthimer

Business Operation Manager, Cisco Press: Anand Sundaram

Manager Global Certification: Erik Ullanderson

Executive Editor: Mary Beth Ray

Managing Editor: Patrick Kanouse

Development Editor: Kimberley Debus

Project Editors: Jennifer Gallant, Seth Kerney

Copy Editor: Water Crest Publishing, Inc.

Technical Editors: Eric Hamel, Kirk McBean, Rajesh Pazhyannur

Editorial Assistant: Vanessa Evans

Book Designer: Louisa Adair

Composition: Bronkella Publishing LLC

Indexer: Ken Johnson

Proofreaders: Jennifer Gallant, Seth Kerney

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

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

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

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

About the Authors

Mark Grayson is a Distinguished Consulting Engineer at Cisco with responsibility for leading its mobile architecture strategy. He has more than 15 years of experience in the wireless industry. He holds a first-class honors degree in electronics and communications engineering from the University of Birmingham (England) together with a Ph.D in radio communications and has been granted more than 50 patents in the area of mobile communications.

Kevin Shatzkamer is a Customer Solutions Architect at Cisco with responsibility for long-term strategy and architectural evolution of mobile wireless networks. He has worked at Cisco and in the mobile wireless industry for nine years, focusing on various technologies ranging from GSM/UMTS to CDMA networks, packet gateway, network-based services and security, video distribution, Quality of Service, and end-to-end design theory. Kevin has 16 pending patents related to all areas of work. Kevin holds a bachelor's in engineering from the University of Florida and a master's of business administration from Indiana University.

Scott Wainner is a Distinguished Systems Engineer at Cisco, providing design and consulting support for the major U.S. service providers and enterprises. He joined Cisco in 1999 and has led the architectural development of next-generation IP/MPLS services while applying his innovative efforts toward the development VPN technologies and virtualized services. Scott consults with providers in the development of managed services while guiding Cisco in the development of critical IP technologies, such as session-border controllers, virtual private networks, and multimedia distribution systems. His latest development efforts have focused on group-encrypted security architectures for broadcast/multicast content and network virtualization for the enterprise VPN. Scott has most recently started architectural development of integrated services using IP Multimedia Systems. Prior to joining Cisco, Scott was the Director of Product Development at Qwest and Engineering Manager of Global IP Services at Sprint/GlobalOne. He has been active in the networking industry for more than 23 years, including participation in the IETF and IEEE.

About the Technical Reviewers

Eric Hamel is currently working as a system architect in the Service Provider Systems Unit (SPSU) at Cisco, specifically on femtocell technology. He has been at Cisco for eight years, with previous Cisco experience as a Consulting Systems Engineer for service providers covering mobile network architecture (GSM, UMTS, and LTE) with specific focus on voice and signaling, packet data gateway infrastructure, and policy control solutions. Prior to Cisco, Eric worked at France Télécom for four years on mobile core network and was actively involved in standardization activities (ETSI, 3GPP). Eric has filed multiple U.S. patents in his area of expertise (mobile charging, policy, and so on).

Kirk McBean is currently working as an architect for mobile wireless networks at Cisco. He is actively involved in providing input to the Cisco strategy for mobility. His current focus is mobile network architecture, product definition, and product development for mobile technologies such as 4G/LTE and femtocell. Kirk has been with Cisco for 10 years and over the years has supported large customer deployments of IP/MPLS technologies ranging from MPLS L2/L3 VPN to planning complex ATM to MPLS migration strategies. Kirk holds an engineering degree from Pennsylvania State University and enjoys spending time with his wife Evalee and their two children, Sydney and Kira.

Rajesh Pazhyannur, Ph.D, is currently working in the CTO group of the Wireless Technology Group at Cisco. His current activities include indoor cellular solutions such as femtocell and distributed antenna systems, focusing on systems architecture. He has more than 15 years experience in telecommunications industry. He holds nine issued patents and has been published in numerous publications. He is an avid long distance runner and lives with his wife Ishita and daughter Saniya in northern California.

Dedications

Mark Grayson: I dedicate this book to my amazing family, who were unfortunately abandoned on many nights and weekends in order to provide the time for this manuscript to be completed. To my wonderful wife, Sharon, for all the support and encouragement, and to my two sons, Charlie and Harry, who provide the welcomed diversions from a self-absorbed focus on mobile architectures. I would also like to thank the many friends, co-workers, and mentors who, over the last 20-odd years, have helped me achieve so much.

Kevin Shatzkamer: This book is dedicated to my entire family—Jerusha, Emma, Benjamin, and Nathaniel—without whom I would never have had the constant distractions, but also the overwhelming support, during my work on this book. It is for you that I do everything that I do. To my brother, Jason, who pushes me to excel in everything that I do and makes life one big competition. To all my colleagues at Cisco—who constantly make my work environment challenging and a constant learning experience.

Scott Wainner: This book is dedicated to my children—Craig, Brett, Natalie, and Caroline. You are cherished more than you will ever know, and I thank God that He has entrusted you to me. To Julie, you are the love of my life. God blessed me with your presence, smiled, and said "Now watch him soar." To my dad, Tom Wainner, you are an awesome role model—one I aspire to mimic every day. In memory of my mother, Zenith Wainner, I feel your comforting presence with me all the time. What joy you brought to my life. To my family, friends, brothers and sisters in Christ, and colleagues, thank you all.

Acknowledgments

We'd like to give special recognition for the tremendous support we received inside Cisco in writing this book—especially our management team and co-workers. Their expert knowledge was invaluable in the development from the very onset of the book's outline.

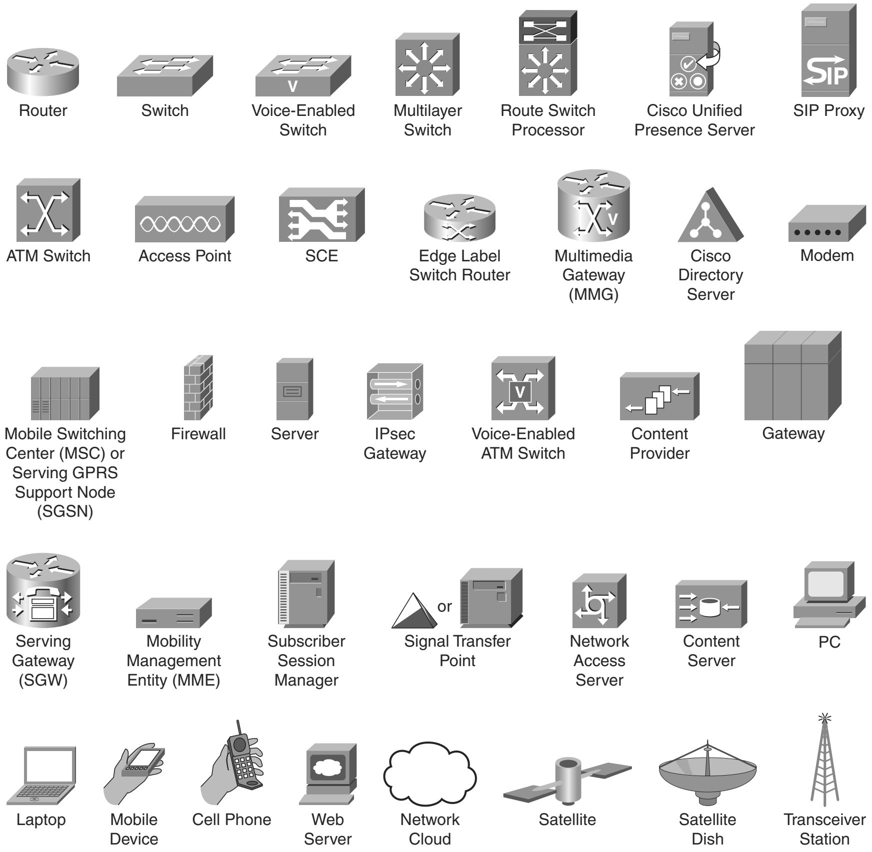
Writing a book takes a tremendous amount of patience, discipline, and of course, time. We would like to extend our gratitude to Mary Beth Ray, Cisco Press Executive Editor, for her patience with missed deadlines and rework required to bring this book to fruition. Her assistance through the entire process has made the authoring of this book a rewarding experience.

Most of all, we would like to thank the many standards organizations, technologists, and mobile experts that have contributed to mobile network evolution. Without your hard work in striving for the best of the best, we would not have had a book to write.

Contents at a Glance

Contents

Icons Used in This Book



Command Syntax Conventions

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

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

Introduction

The cellular world, for much of its history, has focused on circuit-switched voice and simple text messaging as its two primary applications. Cellular technology is tremendously successful, with over half the world's population being mobile telephony subscribers. At the same time, the Internet revolution has had a profound impact on the diversity of services accessible over IP-enabled networks, with IP now recognized as the fundamental building block for all next-generation communication networks.

The next step in the evolution of the Internet will be to make it available anytime and anywhere. This will require the convergence of the cellular world and the Internet. This convergence is being driven by a host of powerful new mobile devices, high-speed mobile networks, compelling applications, and flat-rate all-you-can-eat billing plans.

IP is now impacting all aspects of the mobile operator's network, from radio bearer support through transmission and service delivery capability. Indeed, the various definitions for the next generation of mobile networks all align around an "all-IP" vision, providing purely packet-switched capabilities and solely supporting IP services.

End-to-end IP provides the flexibility to cost-effectively deliver services and applications that meet users' changing needs.

As today's mobile networks migrate toward "All-IP," with various interim steps along the way, it is important to educate those who are focused on the evolving mobile technologies on proper IP design theory and the fundamental role IP has in their next-generation mobile networks. Tomorrow's RF engineers, mobile network designers, and system architects will be expected to have an understanding of IP fundamentals and how their role in delivering the end-to-end system is crucial for delivering the all-IP vision.

This book seeks to focus on the transition of the mobile network from today's circuit-switched technologies toward a future where IP is the fundamental building block integrated into all aspects of the network. This IP transition begins with function-specific migrations of specific network domains and ends with an end-to-end IP network for radio, transport, and service delivery. This book looks at the transition from both the standards and design theory perspective.

Who Should Read This Book?

This book is not designed to provide an all-inclusive reference for evolving mobile networks to Internet Protocol (IP). This book is intended to increase the reader's understanding of the current and target state of mobile networks, and the technology enablers that assist mobile operators' migration.

This book assumes at least a basic understanding of standard networking technologies, including the Internet Protocol itself. Many concepts are introduced in order to give the reader exposure to the key technology trends and decision points impacting today's mobile operators. The book does not give recommendations on which of these technologies should be deployed, nor does it provide a transition plan for a mobile operator. Each mobile operator is expected to evaluate the technologies and make decisions based on their own criteria.

This book is written for many levels of technical expertise, from network design engineers and network planning engineers looking to design and implement mobile network migrations toward an all-IP future, networking consultants interested in understanding the technology trends that affect their mobile service provider customers, students preparing for a career in the mobile environment, and Chief Technology Officers (CTOs) seeking further understanding of the value IP technology brings to the mobile network.

How This Book Is Organized

Depending on the level of technical depth required, this book may be read cover-to-cover or be used as a reference manual for IP's role in mobile network evolution. The book is designed to be flexible and enable you to move between chapters and sections of chapters to cover just the material that you need more work with.

The book is divided into three parts.

Part I, "Cellular Networks and Standards," provides an overview of how IP is being integrated into mobile systems, including RF, radio systems, and ceullular networks. Part I includes the following chapters:

- **Chapter 1, "Introduction to Radio Systems":** This chapter provides an introduction to various radio technologies, and wireless technologies used to transport IP over radio bearers, an important foundation for expanding into IP design theory for mobile networks.
- **Chapter 2, "Cellular Access Systems":** This chapter provides an overview of legacy mobile radio systems, including GSM, UMTS, and cdma2000, presenting details of how IP services have been overlaid on top of circuit-switched architectures.
- **Chapter 3, "All-IP Access Systems":** This chapter provides an overview of the "All-IP" Access systems and standards. IP as a fundamental technology for future mobile access systems is discussed.

Part II, "IP and Today's Cellular Network," provides an overview of IP, the technologies used for transport and connectivity of today's cellular networks, and how the mobile core is evolving to encompass IP technologies. Part II includes the following chapters:

- **Chapter 4, "An IP Refresher":** This chapter is intended to level set understanding of IP technology and design theories in order to provide a foundation for expanding into IP design theory for mobile networks.
- **Chapter 5, "Connectivity and Transport":** This chapter discusses the technologies involved in connectivity and transport for mobile networks over various media.

- **Chapter 6, "Mobile Core Evolution":** This chapter provides details on how the mobile core network is evolving, describing how IP connectivity is provided over mobile networks, as well as how IP is being used to transport the circuit-switched core network.
- **Chapter 7, "Offloading Traditional Networks with IP":** This chapter discusses the evolution of today's TDM-based technologies to IP through offload scenarios for the mobile backhaul network.

Part III, "The End-to-End Services Network," provides an overview of the end-to-end services network based on IP, including context awareness and services. Part III includes the following chapters:

- **Chapter 8, "End-to-End Context Awareness":** This chapter discusses the concept of Intelligent IP Networks to extend core functionality and provide intelligent delivery of traffic to mobile subscribers.
- **Chapter 9, "Content and Services":** This chapter discusses the evolution of content and services from circuit-switched technologies to IP-based technologies, and the evolution of the service framework from the Intelligent Network (IN) to service delivery platforms and the Intelligent Multimedia Subsystem (IMS).

Chapter 7

Offloading Traditional Networks with IP

Traditional mobile networks, such as today's 2G (GSM, CDMA 1x) and 3G (UMTS/HSPA and EVDO) networks, are based on Time Division Multiplexing (TDM) for transmission. These TDM networks comprise the majority of the backhaul networks for transport of voice and data traffic. Figure 7-1 shows backhaul penetration worldwide by technology.[1]

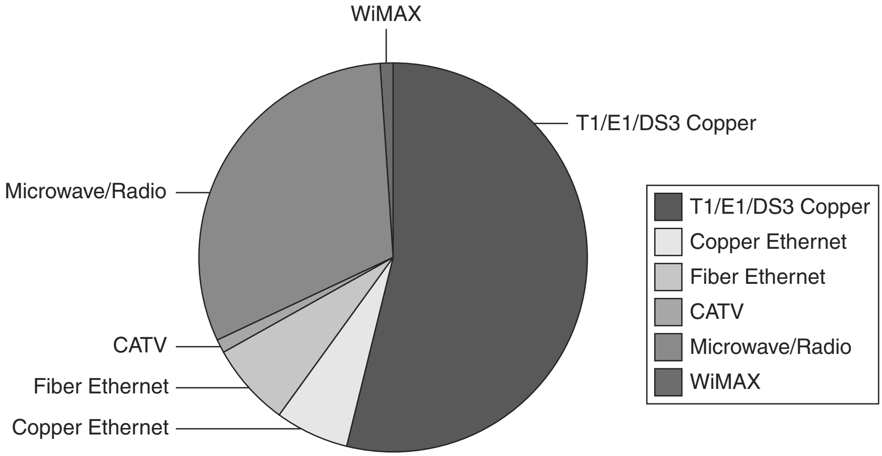


Figure 7-1 *Backhaul Network Penetration*

As mobile network data traffic grows, and user demand and dependency on the mobile operator as a data access provider increases, mobile operators are exploring various offload mechanisms to migrate legacy TDM networks to modern Ethernet and IP. Figure 7-2 demonstrates the increased bandwidth requirements per base station, to support next-generation mobile services.[2]

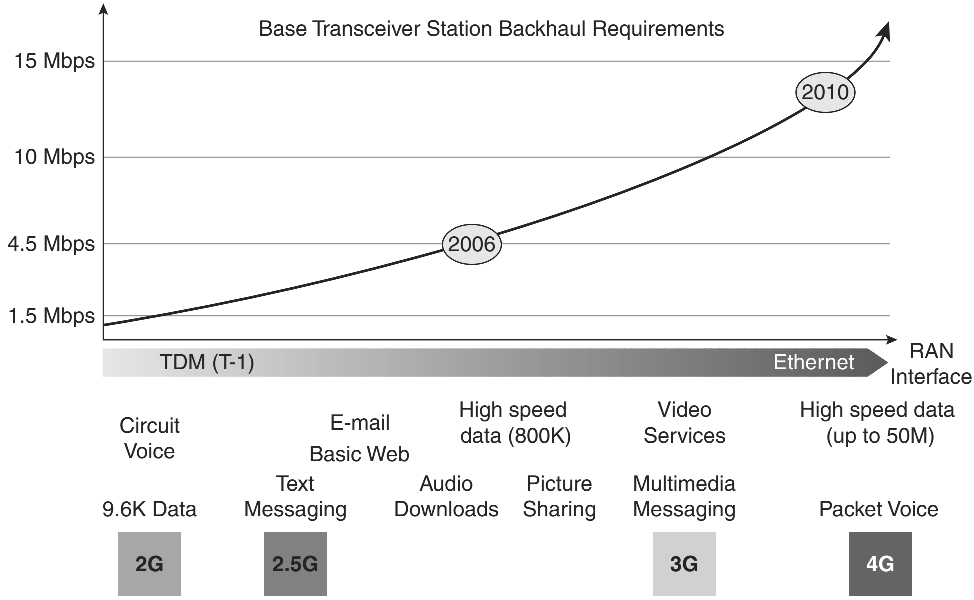


Figure 7-2 *Backhaul Bandwidth Requirements*

This migration allows a mobile operator to shed excess Operating Expenditures (OPEX) associated with TDM transport. However, during this migration, supporting legacy TDM interfaces and network elements is critical for continuing operations.

Various IP-based offload mechanisms may be employed to allow for this migration without the high Capital Expenditure (CAPEX) outlay for new equipment (BTS, BSC, and MSC infrastructure).

These IP-based offload mechanisms can be largely categorized as follows:

- **Backhaul offload** involves encapsulation of standard TDM protocol communications between the Base Transceiver Station (BTS) and the Base Station Controller (BSC), the BSC and the Mobile Switching Center (MSC), or inter-BSC/MSC, into IP packets.
- **Signaling protocol offload** involves protocol conversion of signaling packets. An example of signaling protocol offload is SS7/SIGTRAN.
- **Bearer protocol offload** involves protocol conversion of bearer packets. Examples of bearer protocol offload include Transcoder-Free Operations (TrFO) mechanisms and IP Soft-Handoff mechanisms.

Backhaul Offload with Pseudowires

Pseudowires allow for the emulation of point-to-point or point-to-multipoint links over a Packet-Switched Network (PSN). Pseudowire technology provides a migration path, allowing an operator to deploy packet-switched networks without immediately replacing legacy end-user equipment.

Each pseudowire presents a single, unshared "circuit" for carrying "native" services, such as ATM, SONET/SDH, TDM, Ethernet, or Frame Relay, over the PSN. The PSN may either be Layer 2 Tunneling Protocol Version 3 (L2TPv3), MPLS, or generic IP.

Many standards organizations, including the Internet Engineering Task Force (IETF), the Metro Ethernet Forum (MEF), and the International Telecommunications Union Telecommunications Standards Sector (ITU-T), have defined the encapsulation techniques for transport of the relevant protocols in mobile networks today, as follows:

- **IEEE RFC3985:** Pseudowire Emulation Edge-to-Edge (PWE3).
- **IEEE RFC5087 and ITU-T Y.1453:** Time Division Multiplexing over IP (TDMoIP).
- **IEEE RFC4553:** Structure-Agnostic Time Division Multiplexing over IP.
- **IEEE RFC5086:** Circuit Emulation Services over Packet-Switched Networks (CESoPSN).
- **IEEE RFC4717 and ITU-T Y.1411:** ATM Pseudowires.
- **IEEE RFC4842:** Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Circuit Emulation over Packet (CEP).

Pseudowire Use-Cases

Prior to discussing pseudowire technology itself, the following examples should help to clarify various uses for pseudowire technology in mobile networks. The examples discussed may not be applicable to all mobile operators or all mobile infrastructure vendors, but are representative of some of the many deployment scenarios where pseudowires have been successfully deployed as an offload mechanism. The examples cover four scenarios, as follows:

- TDMoIP Pseudowire for CDMA/EVDO or GSM Backhaul Networks
- CESoPSN Pseudowire for Inter-BSC/MSC Connectivity
- ATM Pseudowires for UMTS R4 Connectivity
- Pseudowires for Multi-RAN Environments

Details of each pseudowire technology and implementation follow.

TDMoIP Pseudowires for EVDO or GSM Backhaul Networks

As discussed in Chapter 4, "An IP Refresher," the traditional mobile backhaul network for a CDMA or GSM network consists of TDM interfaces on both the Base Transceiver Station (BTS) and the Base Station Controller (BSC). These TDM interfaces connect to a backhaul provider's T1/E1 circuits for transport. Figure 7-3 illustrates a mobile backhaul network with standard TDM backhaul.

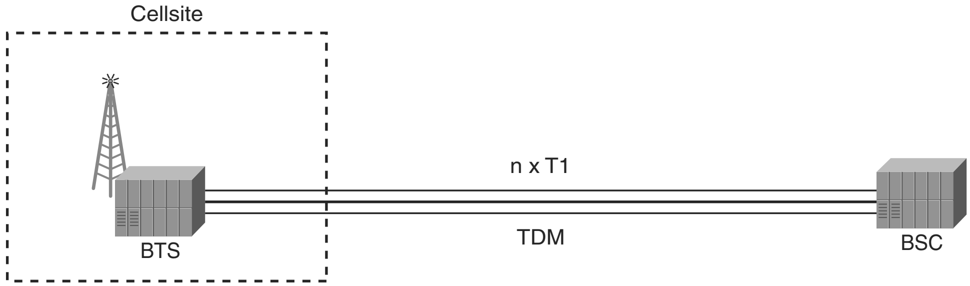


Figure 7-3 *Traditional TDM Mobile Backhaul Network*

TDM pseudowire technology plays a key role in allowing mobile operators to migrate their backhaul networks between the BTS, or cell site, and BSC or MSC location. The pseudowire provides a "transparent wire" between these locations and preserves the integrity of the TDM framing as it is transmitted across the PSN. Figure 7-4 illustrates a mobile backhaul network that uses TDMoIP pseudowires for transport.

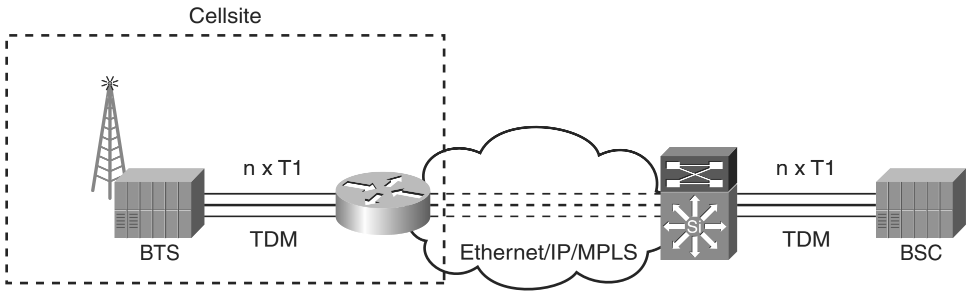


Figure 7-4 *Mobile Backhaul Network with TDM Pseudowires*

CESoPSN Pseudowires for Inter-MSC/BSC Connectivity

As discussed in Chapter 4, the traditional MSC and BSC functionality and connectivity is typically TDM-based. Interconnectivity between all BSCs/MSCs is essential for handling mobility of a voice session in a circuit-switched voice (GSM, CDMA 1x) environment. However, in order to support such an environment, typical mobile deployments rely on a combination of point-to-point TDM circuits between BSC and MSC, and fully-meshed or star configurations of TDM circuits from the MSC toward the core network. Figure 7-5 illustrates one such topology.

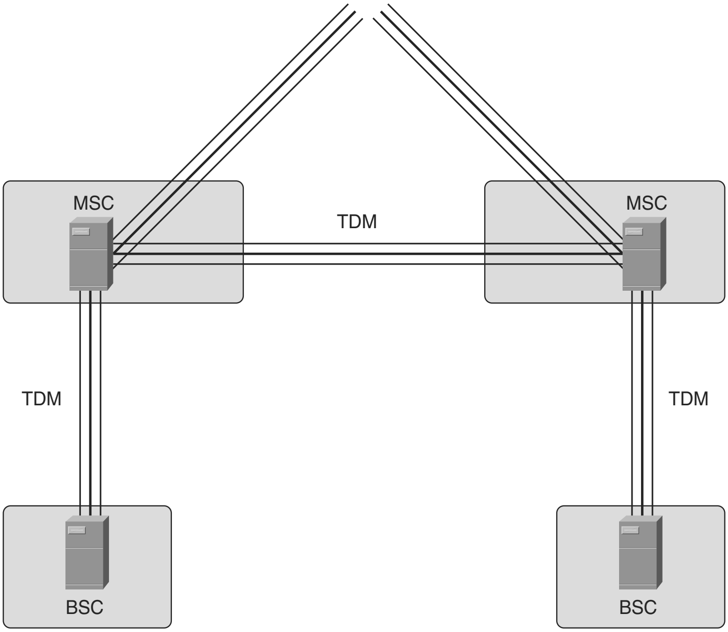


Figure 7-5 *Inter-MSC/BSC Connectivity*

The overall cost of maintaining a fully-meshed, point-to-point TDM architecture is significant from an OPEX perspective. By reducing the number of TDM circuits required from the Local Exchange Carrier (LEC), a mobile operator may immediately see impact to operating margins. One such way to reduce the number of circuits is to leverage CESoPSN pseudowires for interconnecting MSC and BSCs, as illustrated in Figure 7-6.

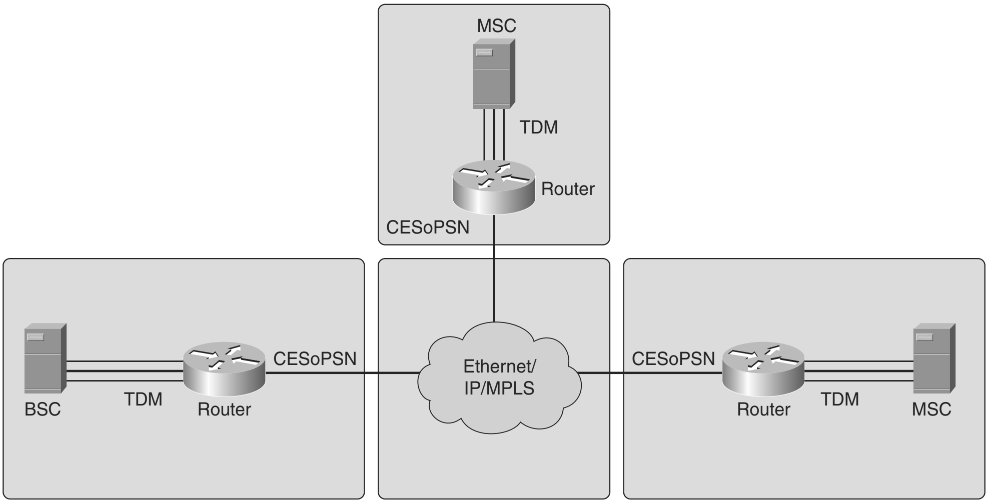


Figure 7-6 *Inter-MSC/BSC Connectivity with CESoPSN Pseudowires*

Inter-MSC/BSC connectivity with CESoPSN pseudowires allows a mobile operator to use existing infrastructure, namely their IP core network, for transport of voice traffic.

ATM Pseudowires for UMTS R4 Backhaul Networks

UMTS Release 4 networks rely heavily on ATM as a transport mechanism for data traffic. Similar to the model previously discussed for transport of TDM backhaul traffic in CDMA and GSM environments, fixed circuits must be deployed to allow for mobility. These fixed ATM circuits, known as Permanent Virtual Circuits (PVCs), are discussed in more detail in Chapter 4. Figure 7-7 depicts a UMTS R4 backhaul network, from Node B to RNC and from RNC to MSC/SGSN.

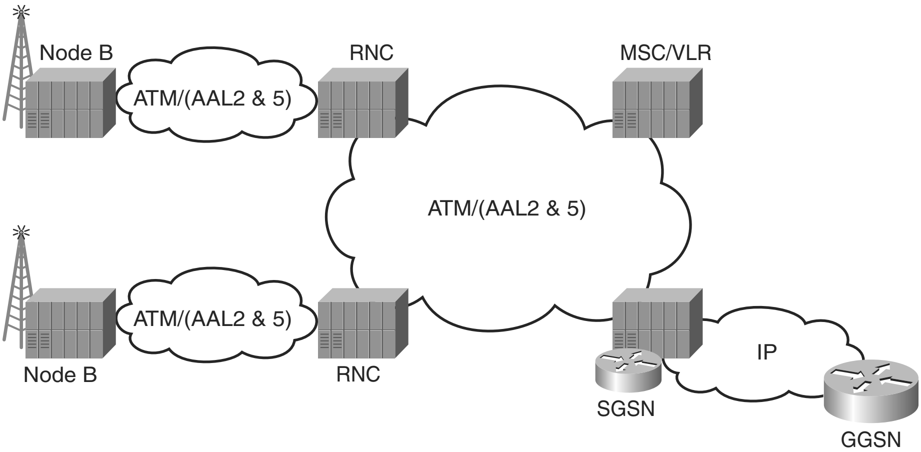


Figure 7-7 *UMTS R4 Backhaul with ATM*

By migrating to ATM pseudowires and leveraging IP core assets, mobile operators can simplify their architecture, reduce costs, and begin preparing for fourth-generation mobile technology deployment, such as 3GPP Long-Term Evolution (LTE), discussed in Chapter 3, "All-IP Access Systems." Figure 7-8 illustrates one potential solution with ATM pseudowires.

It is also possible that IP backhaul and IP core networks may converge over a common IP/MPLS network.

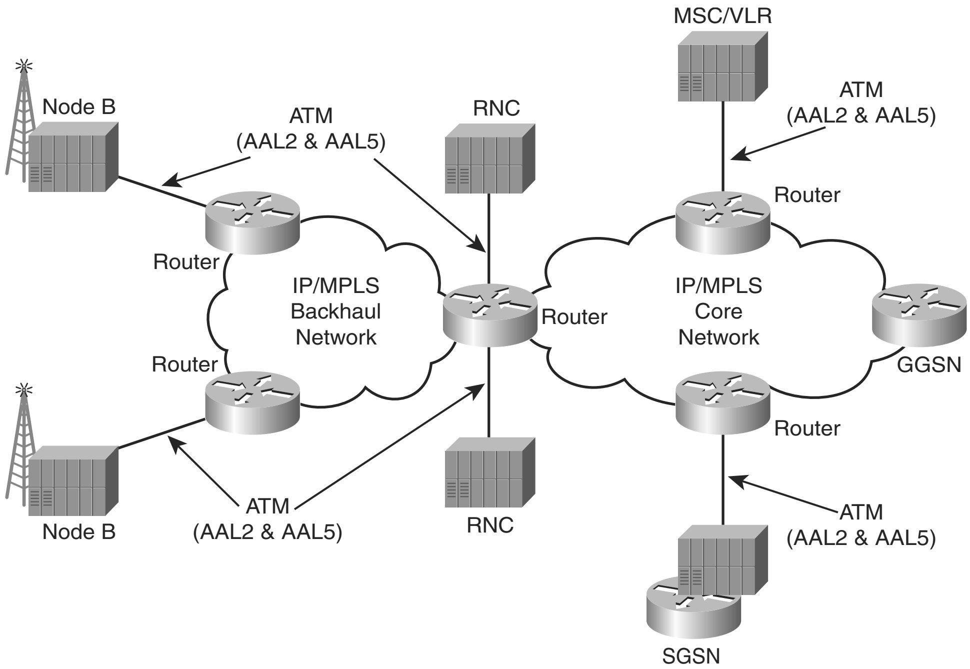


Figure 7-8 *UMTS R4 Backhaul with ATM Pseudowire*

Converging Multiple RAN Technologies over Common Pseudowire

As mobile operators complete their transition from solely circuit-switched voice networks to voice and data networks, mobile networks begin to become an overlay of multiple radio technologies. With all these multiple overlays requiring unique circuits (TDM or ATM), mobile operators incur large OPEX for maintaining multiple different backhaul networks. For instance, a CDMA operator maintains a CDMA 1x voice network and EVDO data network simultaneously. Even if the radio access cards reside in the same physical element, mobile operators use unique circuits for voice and data traffic in order to facilitate troubleshooting and problem isolation.

Pseudowires present an opportunity for mobile operators to deploy a unified backhaul architecture while still managing each circuit individually.

Example 1, illustrated in Figure 7-9, highlights a converged RAN architecture for a CDMA operator.

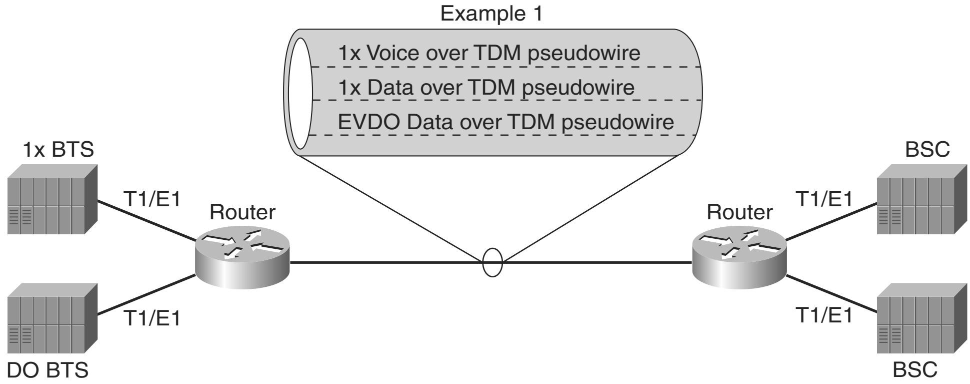


Figure 7-9 *Converged RAN Architecture for CDMA*

Example 2, illustrated in Figure 7-10, highlights a converged RAN architecture for a GSM/UMTS operator.

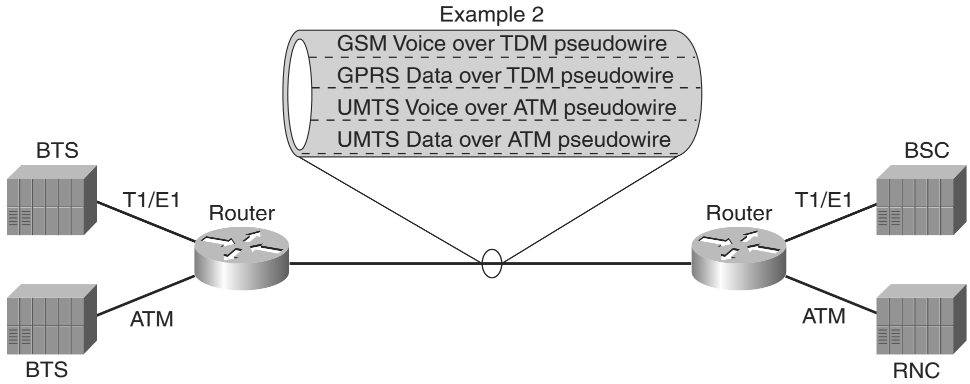


Figure 7-10 *Converged RAN Architecture for UMTS*

Note With the initial 3G release, there is no differentiation between voice and data traffic on the link between the Node B and the RNC (contrary to what is shown in the figure). The whole traffic is encapsulated in a Frame Protocol and send to/from the RNC. The differentiation is done later. This is changed in later releases of UMTS.

Pseudowire Emulation Edge-to-Edge (PWE3)

Pseudowire Emulation Edge-to-Edge RFC 3985 provides the structure and architecture for emulation of Frame Relay, ATM, Ethernet, TDM, and SONET over packet-switched networks using IP or MPLS.

Pseudowires for Time Division Multiplexing (TDM)

At the most basic level, TDMoIP pseudowires segment T1/E1 frames, encapsulate these frames in Ethernet, and fragment the frames into IP packets for transport across the PSN. At the destination, the IP header is stripped, the Ethernet frame is decapsulated, and the original bit stream is reconstructed, including regeneration of clock information. Figure 7-11 illustrates a high-level view of a TDMoIP pseudowire.

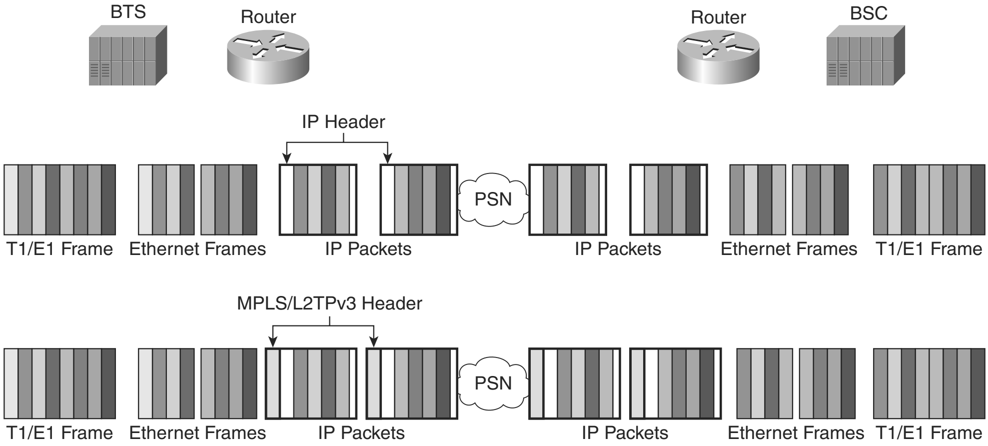


Figure 7-11 *High-Level View of TDMoIP Pseudowire*

Structure-Awareness of TDM Pseudowires

TDM over IP pseudowires can be categorized into two classes, as follows:

- **Structure-Agnostic Transport over Packet (SAToP):** With structure-agnostic transport, the protocol may disregard all structures imposed on TDM signaling or framing. Therefore, this transport method is simply bit-by-bit transport. Structure-agnostic TDM over IP is standardized in RFC4553. The PE devices in SAToP transport network do not participate in TDM signaling and do not interpret the TDM data. This implies that there are no assurances that network degradation does not impact the TDM structure.
- **Structure-Aware Transport over Packet:** With structure-aware transport, such as TDMoIP and CESoPSN, the integrity of the TDM structure is ensured, even in cases of network degradation. Because PE devices have exposure to the TDM signaling, individual channels are exposed, allowing the network to utilize Packet Loss Concealment (PLC) and bandwidth conservation mechanisms on a per-channel basis.

TDM Structures

A frame structure refers to the way a single communications channel is multiplexed in several individual channels. By multiplexing the underlying channel, more than one data stream may be simultaneously transmitted at a time. Because TDM is based on the time domain, a single frame is actually a constant-length time interval. Within this time interval, fixed-length timeslots, each representing a single circuit-switched channel, are transmitted.

A multiplexer is responsible for assigning data, or bytes, from a bitstream to each timeslot, and a demultiplexer is responsible for re-assembling the bitstream. Although every timeslot may not be used, the entire frame is always transmitted in order to ensure that frames remain synchronized.

A T1 frame consists of (24) 8-bit (1-byte) timeslots plus a synchronization bit, allowing for 193 bits. An E1 frame consists of 32 timeslots, each containing 8 bits, or a total of 256 bits per frame, including a synchronization bit. In both cases, frames are transmitted 8,000 times per second. With this framing information, it is easy to calculate the total available bandwidth for both T1 and E1 circuits:

> T1 Circuit Bandwidth = (24 timeslots * 8 bits + 1 synch bit) * 8,000 frames per second / 1*10^6 bits/Megabit = 1.544 Megabits per second

> E1 Circuit Bandwidth = 32 timeslots * 8 bits * 8,000 frames per second / 1*10^6 bits/Megabit = 2.048 Megabits per second

Multiple channels, each containing 8000 8-bit samples per second, are multiplexed together using TDM framing, as illustrated in Figure 7-12.

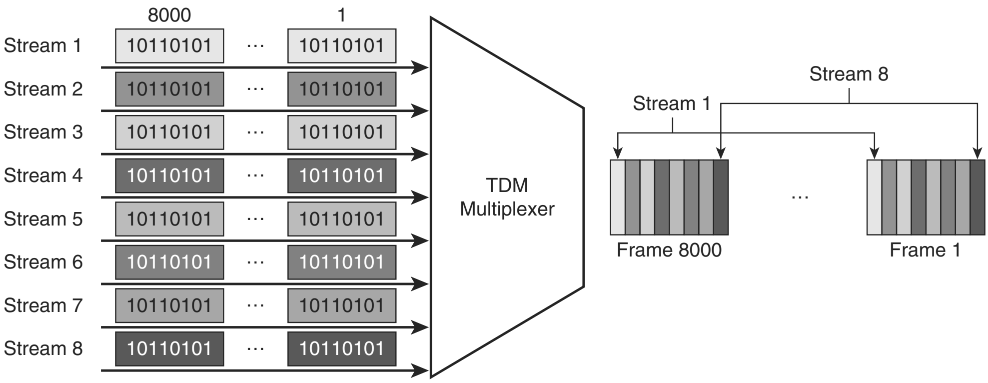


Figure 7-12 *TDM Frame Multiplexing*

Structure-Aware Transport

Structure-aware emulation assumes that the TDM structure itself, including the framing and control information, are available to the pseudowire edge device. With this information available, pseudowire encapsulation can be done in a more intelligent manner, with the edge device selecting specific channel samples from the TDM bitstream. Structure-aware transport may ensure the integrity of the original TDM structure via three distinct adaptation algorithms, as follows:

- **Structure-Locking:** Structure-locking ensures that each packet on the pseudowire contains an entire TDM structure, or multiple/fragments of TDM structures. The exact number of frames included is locked for all packets, in both directions. The order of the frames in the PSN is the same as those within the TDM frame sequence. When a TDM bitstream arrives, consecutive bits from the bitstream, most significant first, fill each payload octet. Structure-locking is not used in TDMoIP.
- **Structure-Indication:** The structure-indication method is derived from ATM Adaptation Layer 1 (AAL1), described in Chapter 4. Unlike structure locking, structure indication allows for pseudowire packets to contain arbitrary-length fragments of the underlying TDM frames. These fragments are taken from the bitstream in-sequence, from the most-significant bit first. The pseudowire packets also include pointers to indicate where a new structure begins. Because the bitstream sequence is identical to the sequence contained in the PSN, this method is commonly known as "circuit emulation."
- **Structure-Reassembly:** The structure-reassembly method allows for specific components of the TDM structure to be extracted and reorganized within the pseudowire packet structure by the ingress pseudowire edge, with enough information such that the other edge of the pseudowire may reassemble the original TDM structure. The structure-reassembly method allows for bandwidth conservation by only transporting frames/timeslots that are active. This method is commonly known as "loop emulation."

TDMoIP uses the structure-indication algorithm for constant-rate, real-time traffic and the structure-reassembly algorithm for variable-rate, real-time traffic. CESoPSN uses the structure-locking algorithm.

Packet Loss Concealment (PLC)

TDM networks are inherently lossless. Because TDM data is always delivered over a dedicated channel at a constant bitrate, TDM bitstreams may arrive with bit errors, but are never out of order and never get lost in transit.

The behavior of a TDM network is not replicable in a cost-efficient manner over an IP network. Implementation of Quality of Service (QoS) and traffic-engineering mechanisms may be used to reduce traffic loss, but there is no guarantee that packets will not arrive out of order, or arrive at all. Packet-Switched Networks are inherently unreliable, and leverage higher-layer protocols to provide for sequencing, retransmission, and reliability.

Because TDM pseudowires carry real-time bitstreams, it is not possible to rely on retransmission mechanisms. Packet Loss Concealment (PLC) masks the impacts of these out-of-order or lost packets. In the case of lost packets, arbitrary packets are inserted into the bitstream to ensure that the timing is preserved. Because a TDM pseudowire packet is considered lost when the next packet arrives, out-of-order packets are not tolerated. TDM pseudowires use different types of arbitrary packets to conceal packet loss, as follows:

- **Zero Insertion:** Insertion of a constant value, or zero, in place of any lost packets. For voice, this may result in some choppiness.
- **Previous Insertion:** Insertion of the previous frame value in place of any lost packets. This method tends to be more beneficial for voice traffic, because voice tends to have a stationarity aspect. This stationarity means that the missing frame should have characteristics similar to the previous frame.
- **Interpolation:** Because a TDM pseudowire is considered lost when the next packet in sequence arrives, the receiver has both the previous and next packets upon which to base the missing frame value. Interpolation algorithms ranging from linear (straight-line interpolation of missing frame value) to more predictive (statistical calculations of missing frame value) may be used; however, there is no standard method for TDM pseudowire frame interpolation.

Time Division Multiplexing over IP (TDMoIP)

TDM over IP was first developed by RAD Data Communications in 1998, and first deployed in 1999 by Utfors, a Swedish broadband communications operator later acquired by Telenor.

Generic Encapsulation

The basic structure of a TDMoIP packet is depicted in Figure 7-13.

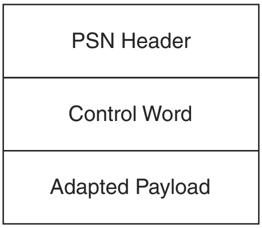


Figure 7-13 *TDMoIP Packet Structure*

TDMoIP packets are composed of three main parts, as follows:

- **PSN Headers:** PSN headers contain IP, MPLS, L2TPv3, or Ethernet information required to send the packet from the pseudowire ingress device toward the destination device, or pseudowire egress device. For example:
  - IP transport requires that the source/destination IP address and port number be included in the header.
  - MPLS transport requires that the MPLS tunnel label be included in the header.
  - L2TPv3 transport requires that the L2TPv3 Session Identifier (pseudowire label) be included in the header.
  - Ethernet transport requires that the Ethernet source/destination MAC address, VLAN header, and Ethertype be included in the header.
- **Control Word:** The Control Word is included in every TDMoIP packet. The Control Word includes information on TDM physical layer failures/defects (local or remote), length of the packet (to indicate if the packet is padded to meet PSN minimum transmission unit size), and sequence number (for detection of lost or misordered packets).
- **Adapted Payload:** The pseudowire ingress device uses either structure-indication or structure-reassembly in order to fill the packet payload.

OAM

Defects in a TDMoIP network may occur in multiple different locations. Depending on the location of the defect, standard TDM OAM mechanisms or TDMoIP mechanisms may be used to alert the TDM peer. Figure 7-14 illustrates the multiple defect locations in a TDMoIP network.

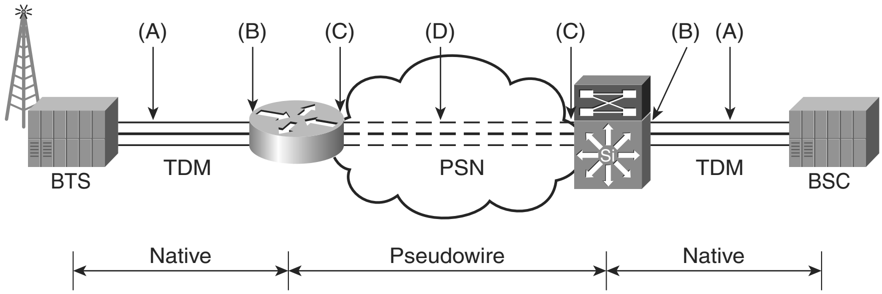


Figure 7-14 *TDMoIP Network Defect Locations*

Table 7-1 includes information about the reference points illustrated in Figure 7-14 and correlated OAM mechanisms, if available.

Table 7-1 *Reference Points and OAM Mechanisms*

Reference Point	Description	OAM Mechanism
(A)	Defect in the L2 TDM network that impacts any number of circuits terminating on the pseudowire edge devices.	The defect is communicated to the pseudowire edge devices and the remote TDM peer via native TDM OAM mechanisms.
(B)	Defect on the pseudowire edge TDM interface.	
(C)	Defect on the pseudowire edge PSN interface.	
(D)	Defect on the PSN that impacts any number of pseudowires terminating on the pseudowire edge devices.	The defect is communicated to the pseudowire edge devices via PSN or pseudowire OAM mechanisms.

Each pseudowire edge device is responsible for maintaining the state of both forward- and reverse-path traffic. Information on the forwarding paths is communicated to the pseudowire edge devices via Forward- or Reverse-Path indication notifications. Table 7-2 discusses the traffic impacts of the received messages.

Table 7-2 *Indication Notifications*

Indication	Source	Impact
Forward-Path Indication	TDM Peer	Impacts ability of the pseudowire edge device to receive traffic over the TDM circuit from the local TDM device. Note: The pseudowire edge device may be able to detect this directly if the failure occurs in the local port or link.
Forward-Path Indication	PSN Peer	Impacts the ability of the pseudowire edge device to receive traffic from the remote TDM device Note: A Forward-Path indication on the PSN does not necessarily imply that the PSN is working improperly, because the defect may be in the remote TDM circuit.
Reverse-Path Indication	TDM Peer	Impacts the ability of the pseudowire edge device to send traffic to the local TDM device.
Reverse-Path Indication	PSN Peer	Impacts the ability of the pseudowire edge device to send traffic to the remote TDM device. Note: This indication may be indicative of either a PSN fault or a remote TDM fault.

TDMoIP includes its own Operations and Maintenance (OAM) signaling path for reporting of bundle status and performance statistics. OAM signaling provides increased reliability to a protocol stack (TDMoIP pseudowires) that is inherently not reliable. The messages are similar to ICMP messages for the IP network.

Connectivity messages are sent periodically from pseudowire edge to pseudowire edge. A response from the remote pseudowire edge device indicates connectivity. Because forward and receive paths may be different, connectivity messages must be sent in both directions.

Performance messages are sent either periodically or on-demand between pseudowire edge devices. Metrics pertinent to pseudowire performance, such as one-way and round-trip delay, jitter, and packet loss, may be measured.

In addition, standard PSN mechanisms, such as Bidirectional Forwarding Detection (BFD) and MPLS Label Switch Path Ping (LSP-Ping), or other protocol-specific detection mechanisms (L2TP mechanisms described in RFC 3931) may be used over each individual pseudowire, as well as the tunnel itself. These mechanisms may be used continually (proactive notification of defects) or on-demand (reactive notification of diagnostics).

Circuit Emulation Services over Packet-Switched Networks (CESoPSN)

Circuit Emulation Services over Packet-Switched Networks (CESoPSN) is defined in RFC 5086, which was first drafted in January 2004.

Packet Structure

Packet structure of CESoPSN is very similar to that of TDMoIP, except for the inclusion of an optional fixed-length RTP header. This packet structure is illustrated in Figure 7-15.

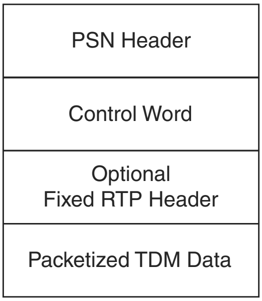


Figure 7-15 *CESoPSN Packet Structure*

RTP

CESoPSN may use an optional RTP header for the transport of timing information. Timing is further discussed later in the chapter in the section, "Timing." The RTP header includes specific timestamp information that can be retrieved in the following two manners:

- **Absolute Mode:** In Absolute Mode, the edge pseudowire device recovers the clock information from the incoming TDM circuit. In this mode, the timestamps are closely correlated with sequence numbers.
- **Differential Mode:** In Differential Mode, the edge pseudowire device has access to a high-quality synchronization source. In this mode, timestamps represent the difference between the synchronization source and the TDM circuit.

CESoPSN Versus TDMoIP

Although both CESoPSN and TDMoIP provide for transport of TDM frames over PSNs using pseudowires, there are numerous differences between the two protocols themselves. These differences include the following:

- TDMoIP uses the structure-indication and structure-reassembly mechanisms, whereas CESoPSN uses the structure-locking algorithm. Therefore, CESoPSN transmits consistent, fixed-length packets, whereas TDMoIP has several payload lengths (minimum of 48 bytes) depending on the type of traffic being transmitted.
  - This allows for CESoPSN to have a lower packetization delay in instances where the pseudowire is carrying multiple timeslots.
  - By the same token, using structure-locking creates inefficiencies when transporting unstructured T1 streams. CESoPSN payload is required to begin at a frame boundary. This means that T1 frames must be padded to create the consistent packet size.
- CESoPSN mandates use of RTP.
- By transporting entire frames, CESoPSN simplifies packet loss compensation.
  - CESoPSN does not need to look at individual timeslots. Instead, CESoPSN inserts a packet of all 1's, simulating TDM fault mechanisms.
- TDMoIP must look for structure pointers, jump to the beginning of the next structure, and insert interpolated data.

ATM Pseudowires

An ATM pseudowire uses an MPLS network for the transport of ATM cells.

Note ATM pseudowires follow the PWE-3 architecture, and therefore only ATM-specific information is included in this section.

Defined in RFC 4717, ATM pseudowires provide many of the same benefits as TDMoIP and CESoPSN:

- Simplification of network architecture and reduction of number of core networks supported
- Preserving existing legacy services during migration to next-generation IP services
- Using a common PSN to provide both legacy and next-generation services

The generic architecture of an ATM pseudowire service is illustrated in Figure 7-16.

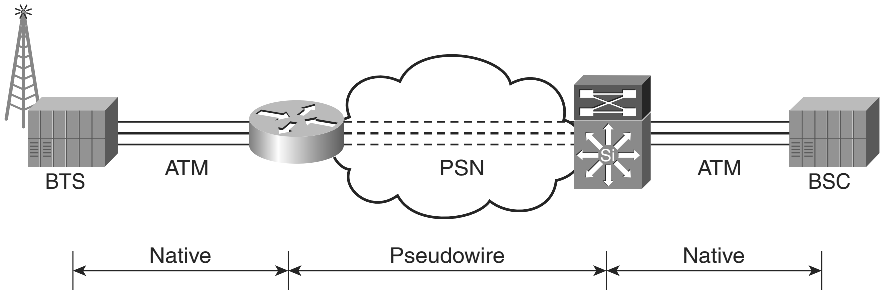


Figure 7-16 *ATM Pseudowire Architecture*

As with all pseudowire services, the intent of an ATM pseudowire is not to perfectly emulate the traditional service, but instead to provide a transport mechanism for the service. This means there are distinct differences between the traditional ATM service and an ATM pseudowire, namely the following:

- ATM cell ordering is optional.
- ATM QoS model can be emulated, but is application-specific in nature.
- ATM flow control mechanisms are not understood by the MPLS network, and therefore cannot reflect the status of the PSN.
- Control plane support for ATM Switched Virtual Circuits (SVCs), Switched Virtual Paths (SVPs), Soft Permanent Virtual Circuits (SPVCs), and Soft Permanent Virtual Paths (SPVPs) are supported only through vendor-proprietary solutions.

Generic Encapsulation

Figure 7-17 illustrates the general encapsulation method for ATM pseudowires.

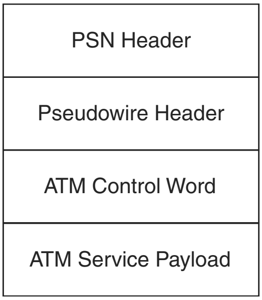


Figure 7-17 *ATM Generic Encapsulation Method*

The PSN Transport header is used to transport the encapsulated ATM information across the network. The structure of this header depends on the type of transport protocol being used.

The pseudowire header maps an ATM service to a particular tunnel. If MPLS is being used, for instance, the pseudowire header would be an MPLS label.

The ATM Control Word contains the length of the ATM service payload, sequence number, and other relevant control bits. There are two types of control words that can be used, as follows:

- **Generic Control Word:** This control word is used for ATM One-to-One cell mode and ATM Adaptation Layer (AAL) 5 Protocol Data Unit (PDU) frame mode.
- **Preferred Control Word:** This control word is used for ATM N-to-One cell mode and AAL5 Service Data Unit (SDU) frame mode.

Cell Mode Modes

There are two methods for encapsulation of ATM cells: N-to-One mode and One-to-One mode.

N-to-One Mode

N-to-One mode is the only required mode for ATM pseudowires. This encapsulation method maps one or more ATM Virtual Circuit Connections (VCCs) or Virtual Path Connection (VPC) to a single pseudowire. The N-to-One mode allows a service provider to offer an ATM PVC- or SVC-based service across a PSN.

With N-to-One mode, the ATM header is unaltered during this encapsulation, so ATM Virtual Path Identifier (VPI) and Virtual Circuit Identifier (VCI) are present. This information is required to be preserved since concatenation of cells from multiple VCCs may occur.

N-to-One mode has the following limitations:

- Explicit Forward Congestion Indication (EFCI) cannot be translated to a PSN congestion mechanism. Conversely, PSN congestion mechanisms cannot be translated to EFCI.
- Cell header detection/correction that exists in ATM cannot be replicated in the PSN.
- Cell encapsulation only functions for point-to-point MPLS Label Switched Paths (LSPs). Point-to-multipoint and multipoint-to-point are not supported.

One-to-One Mode

One-to-One mode is an optional encapsulation method that maps a single VCC/VPC to a single pseudowire. Because only one VPI/VCI is transported on a pseudowire, the pseudowire context (MPLS Label, for example) is used to derive the corresponding VPI/VCI value. The One-to-One mode also allows a service provider to offer an ATM PVC- or SVC-based service across a PSN.

The same limitations as N-to-One mode apply for One-to-One mode.

AAL5 Frame Encapsulation

There are different optional encapsulation methods that exist specifically for AAL5—one for SDUs and one for PDUs.

AAL5 SDU frame encapsulation is more efficient than using either N-to-One or One-to-One for AAL5. Because the pseudowire edge needs to understand the AAL5 SDU in order to transport it, the device must support segmentation and reassembly.

AAL5 PDU frame encapsulation allows for the entire AAL5 PDU to be encapsulated and transported. Because of this, all necessary ATM parameters are transported as part of the payload. This simplifies the fragmentation operation because all fragments occur at cell boundaries, and the Cyclical Redundancy Check (CRC) from the AAL5 PDU can be used to verify cell integrity.

Defect Handling

Figure 7-18 illustrates the four possible locations for defects on the ATM pseudowire service. These four locations are as follows:

- **(A):** ATM connection from ATM device to pseudowire edge device.
- **(B):** ATM interface on the pseudowire edge device.
- **(C):** PSN interface on the pseudowire edge device.
- **(D):** PSN network.

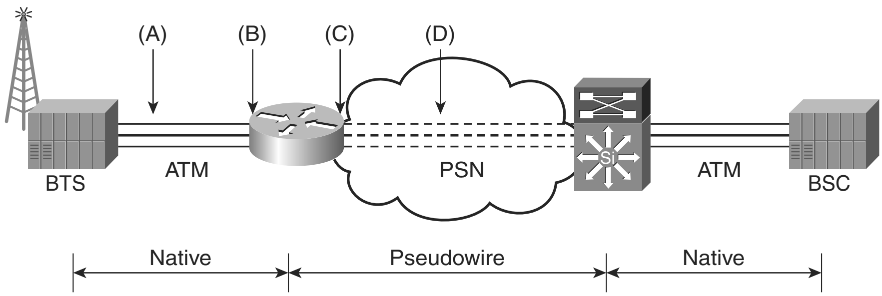


Figure 7-18 *ATM Defect Locations*

In all cases, the pseudowire edge device uses standard ATM signaling methods to notify the receiver of cell loss. This information is transported across the PSN to the receiver.

SONET/SDH Circuit Emulation over Packet

Note SONET/SDH Circuit Emulation over Packet (CEP) follows similar premise and structure to all other PWE3 standards, and therefore only SONET/SDH-specific information is included in this section.

To transport SONET/SDH over packet, the Synchronous Payload Envelope (SPE) or virtual tributary (VT) is fragments, prepended with a pseudowire header, and optionally a RTP header. The basic CEP header is illustrated in Figure 7-19.

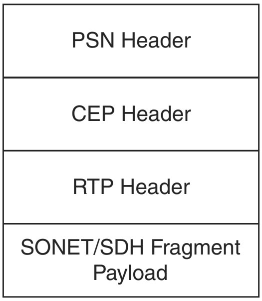


Figure 7-19 *Basic CEP Header*

The CEP header supports both a basic mode, which contains the minimum functionality necessary to perform SONET/SDH CEP, and an extended mode, which contains additional capabilities for some optional SONET/SDH fragment formats. These options fall into two categories, as follows:

- Dynamic Bandwidth Allocation (DBA) is an optional mechanism for SPE transmission suppression on a channel-by-channel basis when one of two trigger conditions are met—that the SONET/SDH path or VT is not transmitting valid end-user data or that the circuit has been de-provisioned, or unequipped.
- Service-Specific Payload Formats are special encapsulations that provide different levels of compression depending on the type and amount of user data traffic. The payload compression options are provided for asynchronous T3/E3 Synchronous Transport Signal 1 (STS-1), fractional VC-4, fractional STS-1, and others.

Fragments

When fragmented, the SONET/SDH fragments must be byte-aligned with the SONET/SDH SPE or VT. That is, the SONET/SDH byte cannot be fragmented, and the first bit in the SONET/SDH must be the most significant bit in the SONET/SDH fragment. In addition, bytes are placed into the fragment in the order in which they are received.

SONET/SDH CEP lies above the physical layer, and assumes that native transport functions, such as physical layer scrambling/unscrambling that SONET/SDH optical interfaces perform as part of their binary coding, occurs as part of the native service. However, CEP does not assume that scrambling has occurred, and fragments are constructed without consideration of this.

Abis/Iub Optimization for GSM Networks

Chapter 2 discusses GSM RAN Abis interface and UMTS RAN Iub interface. GSM RAN Optimization is a method for optimizing and encapsulating structured (NxDS0) TDM signals between the BTS and BSC into IP packets. The optimization is performed by removing nonessential traffic on the GSM Abis interface. Such nonessential traffic includes idle subrates that have a repeating pattern every 20 msec, idle TRAU frames used to keep subrates in-sync for GPRS, and speech TRAU frames with silence used to provide white noise that lets the other party know that the call has not been dropped. In addition, High-Level Data Link Control (HDLC) signaling data flows, which are part of the GSM Radio Link Protocol (RLP), can be optimized by suppressing inter-frame flags.

Note Chapter 2 describes the GSM Abis interface and UMTS Iub interface, including the protocols, functions, and capabilities of these interfaces.

Optimization is done at the bit level, resulting in no impact to voice quality or data throughput. This bit level optimization makes GSM Optimization radio-vendor independent and radio software version independent. Figure 7-20 illustrates GSM Abis optimization.

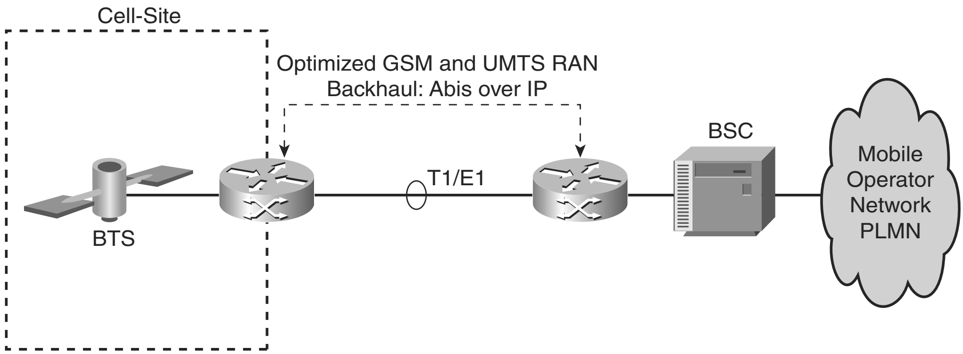


Figure 7-20 *GSM Abis Optimization*

Timing

Today's mobile networks are reliant on accurate timing, or accurate distribution and synchronization of precise clock information, in order to accurately transport voice and data traffic. In existing radio architectures, frequency synchronization is typically achieved through the backhaul network itself. These legacy architectures are based on TDM backhaul. Because TDM carries time inherently, the radio architecture itself was designed with frequency synchronization embedded in the physical layer.

Radio Access Network and Synchronization

The need for synchronization has always been inherent in Radio Access Networks. As discussed in Chapter 1, "Introduction to Radio Systems," radio networks fall into two categories:

- **Frequency Division Duplexing (FDD)**, in which two sets of frequencies are used for transmit/receive. These networks require frequency synchronization in order to accurately send and receive traffic.
- **Time Division Duplexing (TDD)**, in which a single frequency is used for transmit/receive and a demarcation based on timeslots is identified for both transmission and reception. These networks require time synchronization in order to accurately send and receive traffic.

Table 7-3 provides a reference for today's wireless technologies and their synchronization requirements.

Table 7-3 *Wireless Technologies and Synchronization Requirements*

Application	Service
TDM Support	Frequency/Timing
3GPP (GSM, WCDMA FDD)	Frequency
3GPP (LTE, eMBMS)	Time (TDD Mode) Frequency (FDD Mode)
WiMAX (IEEE 802.16d/e)	Frequency Time
DVB-T/DVB-H	Time
TD-SCDMA 3GPP2 CDMA	Time Frequency

In mobile networks, high-quality frequency and time/phase synchronization are useful and in some cases required. The accuracy of these services differs based on the radio technology and standards organization. The synchronization service accuracy based on application (radio technology) is referenced in Table 7-4.

Table 7-4 *Synchronization Service Requirements*

Synch Service	Application	Expected Quality
Frequency	TDM Support	Primary Reference Source (PRS) Traceable
	3GPP/3GPP2 BS	Frequency assignment shall be less than ± $5x10^{-8}$ (± 0.05 parts-per-million [ppm])
	WiMAX (IEEE 802.16)	.16D: Reference frequency accuracy shall be better than $\pm 8*10^{-6}$ ($\pm 2*10^{-6}$) .16e: Reference Frequency Tolerance at BS: ≤ $\pm 1*10^{-6}$
	DVB-T/H/SH/T2	Frequency shall provide a traceable Primary Reference Clock (PRC) source for 10MHz signal
Time	802.16D/e TDD	Better than 5µs
	DVB-T/H	Within 1µs accuracy
	3GPP LTE	Better than or equal to 3µs
	3GPP2 CDMA BS	*Should* be less than 3 µs *Shall* be less than 10 µs
	3GPP eMBMS	TBD

Network Synchronization Options

In order to achieve the stringent quality requirements identified in Table 7-4, there are multiple network synchronization options, as follows:

- **Free-running oscillator:** A free-running oscillator is one that has never been synchronized to a reference clock. This oscillator's accuracy is based on the technology within the oscillator. In this model, each network element would either contain or be connected directly to a free-running oscillator and rely on the local clock for all synchronization. Table 7-5 highlights the different oscillator technologies and accuracy.

Table 7-5 *Oscillator Technology and Accuracy*

Technology	Stratum Level	Accuracy
Hydrogen Maser		$1x10^{-15}$
Cesium	1	$1x10^{-11}$
Rubidium	2	$5x10^{-11}$
Crystal	3/4	$4.6x10^{-6}$

- **Global Positioning System (GPS):** GPS synchronization relies on a GPS satellite to provide the clock source. All GPS satellites contain a Cesium standard clock. Because GPS satellites circle the globe twice per day, any device relying on GPS for synchronization must also calculate geographic location in order to determine from which satellite it can receive signals.
- **Physical layer:** Physical layer synchronization has long been used for transporting clock information. SONET/SDH and T1/E1 are well-known examples of physical layer synchronization. More recently, Synchronous Ethernet (SyncE) uses the Ethernet physical layer interface to pass timing from node to node in much the same way. SyncE is discussed later in this chapter in "Packet-Based Timing."
- **Higher layer:** Higher-layer synchronization relies on a packet-based protocol to distribute clocking information. IEEE 1588v2 and Network Time Protocol (NTP) are discussed later in this chapter in "Packet-Based Timing."

Introduction to Timing

This section provides an overview of timing, including definitions, clock hierarchies, and reference clock architectures. These hierarchies and architectures are leveraged repeatedly in many different circuit-switched and packet-switched timing protocols, and understanding these architectures provides the foundation knowledge for the remainder of this chapter.

Understanding Timing Definitions

Before defining architectures, it is important to understand some of the basic definitions that will be used continually throughout the remainder of this chapter. This section provides some of these basic definitions.

Precision, Accuracy, and Stability

Precision, accuracy, and stability are used to measure the reliability of a clock signal.

- **Precision** is defined as the ability of a measurement to be consistently reproduced. When referencing timing, precision refers to the amount of variation of a set of measurements.
- **Accuracy** is defined as the ability of a set of measurements to consistently match the exact value being measured. In the case of timing, the value being measured is a predefined reference time.
- **Stability** is defined as the amount a measurement changes as a function of time or environment (temperature, shock, and so on).

The goal of every clock is to be highly precise, highly accurate, and highly stable. In practice, however, every clock signal has unique characteristics of precision, accuracy, and stability. These clock signals can fall into six broad categories, as follows:

- **Accurate, precise, stable:** This clock source produces a consistent measurement as a function of time and environment that is representative of the predefined reference time. Figure 7-21 illustrates this type of clock source.

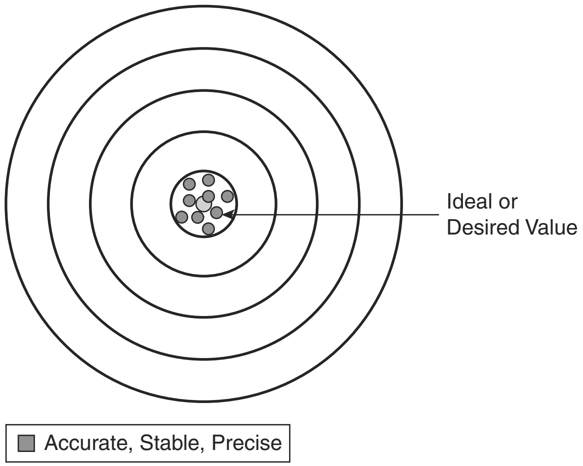


Figure 7-21 *Accurate, Precise, Stable*

- **Accurate, imprecise, stable:** This clock source produces a large variety of measurements, consistent as a function of time and environment, which are representative of the predefined reference time. Figure 7-22 illustrates this type of clock source.

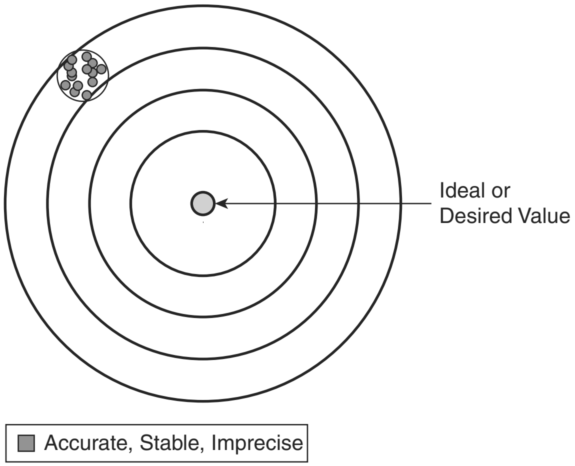


Figure 7-22 *Accurate, Imprecise, Stable*

- **Precise, accurate, unstable:** This clock source produces a small variety of measurement, inconsistent as a function of time and environment, which are representative of the predefined reference time. Figure 7-23 illustrates this type of clock source.

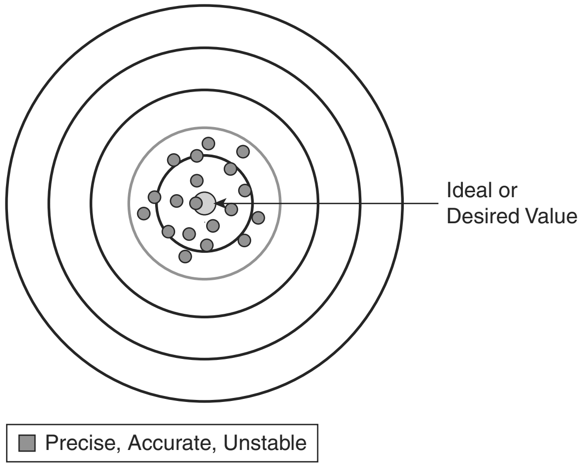


Figure 7-23 *Precise, Accurate, Unstable*

- **Inaccurate, precise, stable:** This clock source produces a small variety of measurements, consistent as a function of time and environment, which are not representative of the predefined reference time. Figure 7-24 illustrates this type of clock source.

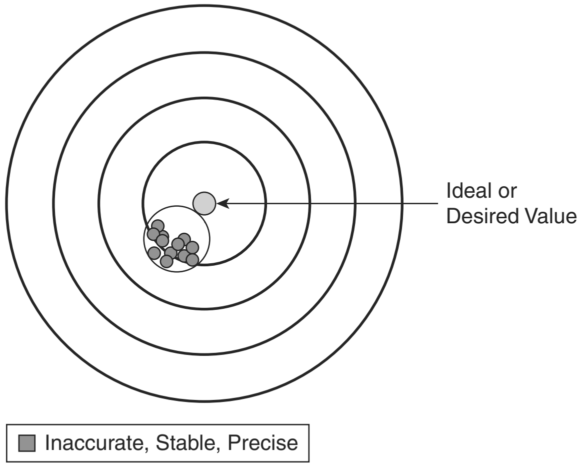


Figure 7-24 *Inaccurate, Precise, Stable*

- **Accurate, imprecise, unstable:** This clock source produces a large variety of measurements, inconsistent as a function of time and environment, which are representative of the predefined reference time. Figure 7-25 illustrates this type of clock source.

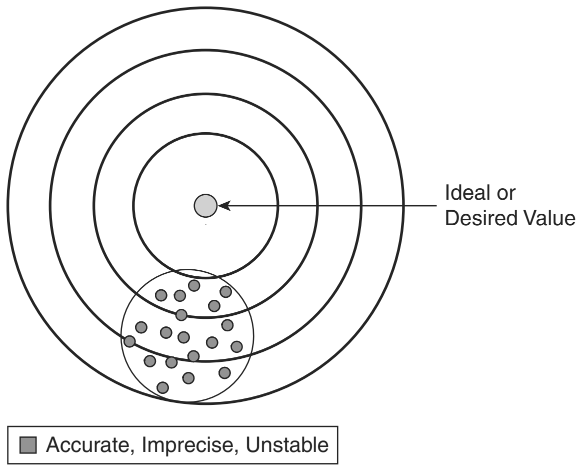


Figure 7-25 *Accurate, Imprecise, Unstable*

- **Inaccurate, imprecise, stable:** This clock source produces a large variety of measurements, consistent as a function of time and environment, which are not representative of the predefined reference time. Figure 7-26 illustrates this type of clock source.

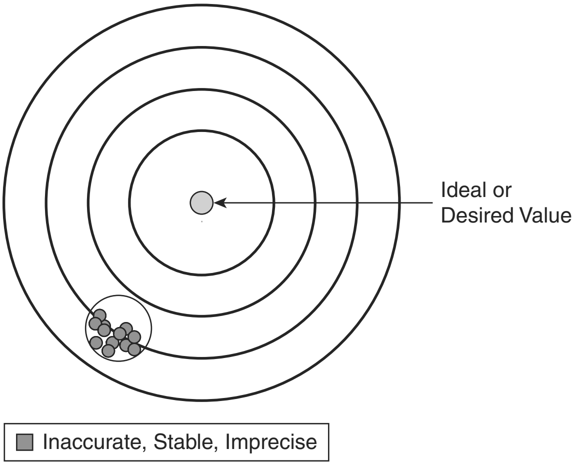


Figure 7-26 *Inaccurate, Imprecise, Stable*

- **Inaccurate, precise, unstable:** This clock source produces a small variety of measurements, inconsistent as a function of time and environment, which are not representative of the predefined reference time. Figure 7-27 illustrates this type of clock source.

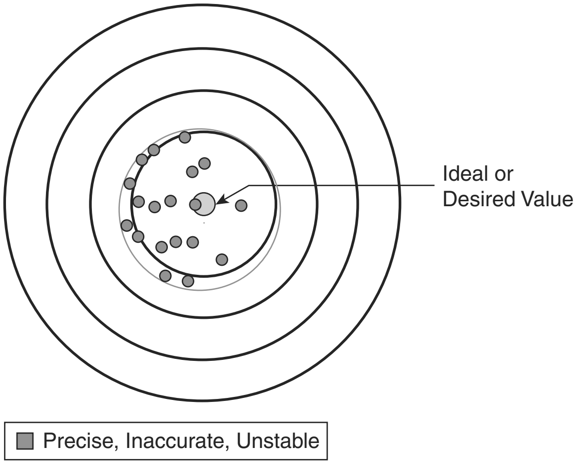


Figure 7-27 *Inaccurate, Precise, Unstable*

- **Inaccurate, imprecise, unstable:** This clock source produces a large variety of measurements, inconsistent as a function of time and environment, which are not representative of the predefined reference time. Figure 7-28 illustrates this type of clock source.

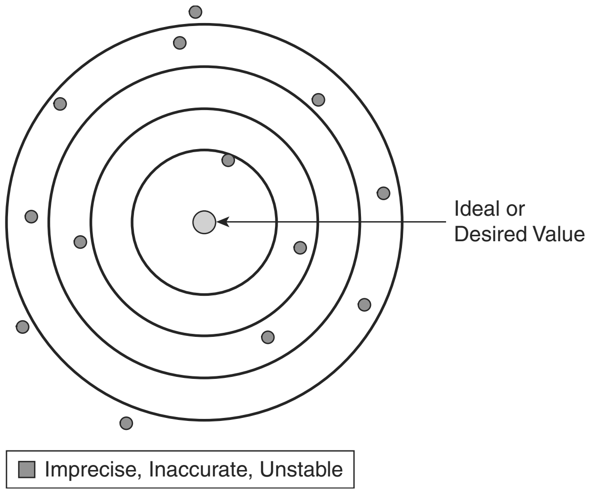


Figure 7-28 *Inaccurate, Imprecise, Instable*

Synchronization

Synchronization refers to timing that requires multiple devices to operate as part of a system at the exact same time. In transporting time-based data traffic, synchronization of all network elements can be achieved in multiple ways. These elements can be synchronized in the following two key ways:

- Frequency synchronization refers to the need for two network elements (transmitter and receiver) to operate at the same rate—that is, both network elements need to operate at the same rate.
- Phase/Time synchronization refers to the need for two network elements to be able to accurately identify the end of a frame or byte. Phase/time synchronization first requires frequency synchronization.

Jitter

Jitter refers to the short-term fluctuations of a timing signal from their ideal positions in time (variations greater than or equal to 10 Hz). Jitter, which is constant over time, makes a clock source unstable. Figure 7-29 illustrates the effects of jitter.

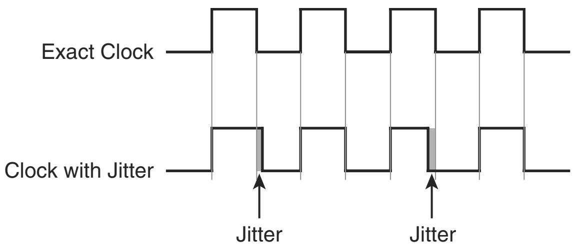


Figure 7-29 *Jitter*

Wander

Wander refers to the long-term fluctuations of a timing signal from their ideal positions in time (variations less than 10 Hz). Unlike jitter, wander is not constant over time, and accumulates in a network. This accumulation leads to either incorrect synchronization or loss of synchronization. Figure 7-30 illustrates the effects of wander. Frequency Drift is a specific type of wander where a constant accumulation occurs.

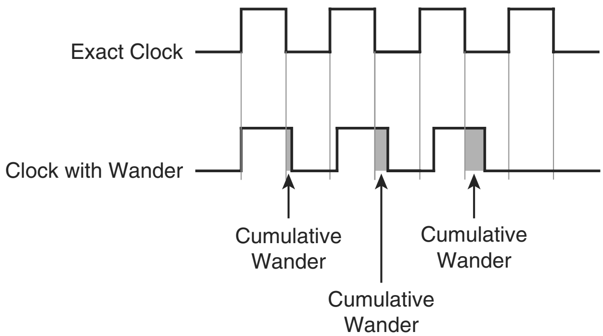


Figure 7-30 *Frequency Drift*

Timing Architectures

Many different timing architectures have been deployed to all accomplish the same end-goal—maximum accuracy, stability, and precision of timing information to all nodes within the network. These timing architectures are dictated by standards-based hierarchies, with each tier of the hierarchy representing a different level of precision. Once the hierarchy is established, operators have varied in their deployment models for distribution and synchronization of this clock information. This section looks at the various clock hierarchy considerations and network architectures that have been deployed.

Clock Hierarchy

Clock standards have a hierarchy defined by the International Telecommunications Union (ITU) Telecommunications Standardization Sector (ITU-T) and the American National Standards Institute (ANSI). For simplicity, ANSI clock hierarchy is used within this chapter. The hierarchy defines the relationship between every clock within a synchronization domain and the model for distribution across the domain. The hierarchy is based on five quality metrics, as follows:

- **Accuracy**.
- **Holdover Stability**, or the ability to continue to preserve accurate time when a clock's reference signal is lost.
- **Pull-In/Hold-In Range**, or the largest offset/differential between the reference frequency and nominal frequency for which the clock can still acquire "lock."
- **Wander**.
- **Time to First Frame (193 bits) Slip**, or the length of time that the clock can remain accurate.

The ANSI and ITU-T clock standards are summarized in Table 7-6.

Table 7-6 *ANSI/ITU-T Clock Standards*

ANSI Stratum	ITU-T Clock Level	Accuracy	Holdover Stability	Pull-In Range	Wander	Time to First Frame Slip
1	PRC	$1x10^{-11}$	None	None	None	72 days
2	Type II	+/-0.016 ppm	$+/-1*10^{-10}$/day	0.016 ppm	0.001 Hz	7 days
-	Type I	Not Defined	$+/-2.7*10^{-9}$/day	0.01 ppm	0.003 Hz	
3E	Type III	+/-4.6 ppm	$+/-1.2*10^{-8}$/day	4.6 ppm	0.001 Hz	3.5 hours
3	Type IV	+/-4.6 ppm	$+/-3.9*10^{-7}$/day	4.6 ppm	3 Hz	6 minutes
-	Option I	+/-4.6 ppm	$+/-2*10^{-6}$/day	4.6 ppm	1–10 Hz	
SMC	Option 2	+/-20 ppm	$+/-4.6*10^{-6}$/day	20 ppm	0.1 Hz	
4	4	+/-32 ppm	None	32 ppm	None	

The network is controlled by a Primary Reference Clock (PRC), or Stratum 1 clock, which is accurate to $1x10^{-11}$. Synchronization requires the distribution of the reference signal from the PRC to all network elements. The master-slave method is used for this propagation. The synchronization between hierarchies is unidirectional; that is, synchronization is always transferred from a higher layer to a lower layer. The Stratum 1 clock receives information from any number of Stratum 0 clocks. The Stratum 1 clock provides the reference clock for multiple Stratum 2 clocks, and each Stratum 2 clock provided the reference clock for multiple Stratum 3 clocks. This hierarchy is illustrated in Figure 7-31.

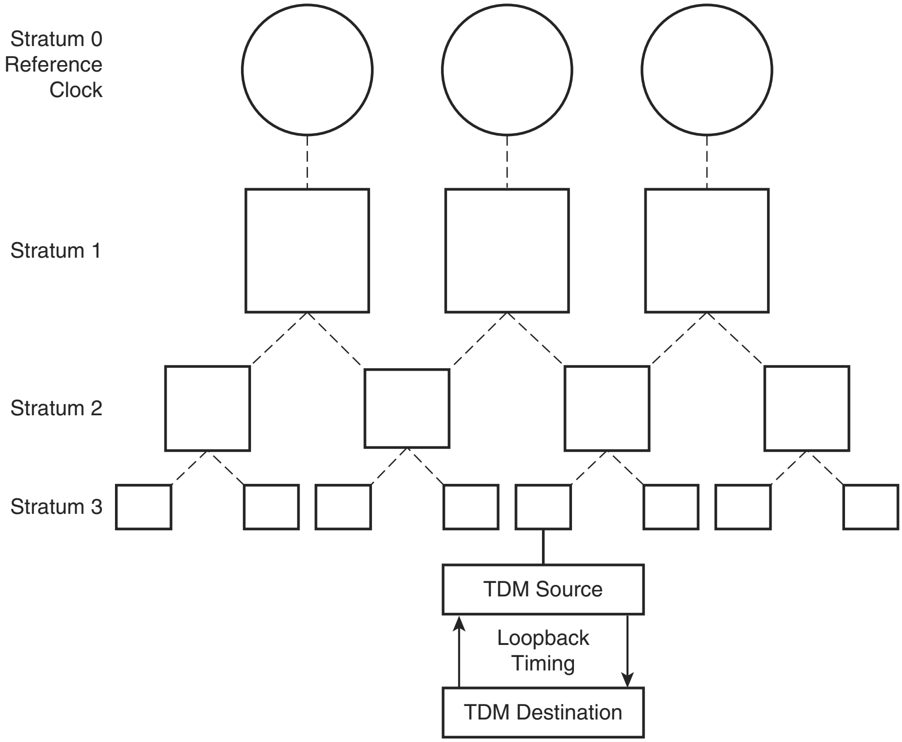


Figure 7-31 *Clock Hierarchy*

PRC Architectures

A PRC is designed to provide highly-accurate time, and therefore tends to rely on more than one Primary Reference Source (PRS). A Cesium-beam tube is always used in the generation of a PRC signal because of their accuracy in ensuring no aging or frequency drift. Three types of PRCs have emerged (and have been identified by The European Telecommunications Standards Institute [ETSI]), as follows:

- Autonomous PRCs with up to three local Cesium tubes incorporated within the PRC and used as the PRS.
- Radio-controlled PRCs, which use remote Cesium tubes in the radio infrastructure (either satellite-based, like GPS, or land-based, like Long Range Aid to Navigation [LORAN]-C) as the PRS.
- PRCs that use a combination of local Cesium tubes and radio-based Cesium tubes.

In the event of a failure of one of the PRS, the PRC can use one of the other PRS as the reference; however, the failover time must be within the Maximum Time Interval Error (MTIE) defined by ITU-T.

Table 7-7 depicts the MTIE defined by ITU-T for each clock level.

Table 7-7 *MTIE by Stratum*

ANSI Stratum	ITU-T Clock Level	Phase Transient
1	PRC	-
2	Type II	MTIE < 150ns
-	Type I	MTIE < 1µs
3E	Type III	MTIE < 150ns
3	Type IV	MTIE < 1µs
-	Option I	MTIE < 1µs
SMC	Option 2	MTIE < 1µs
4	4	No Requirement

Figure 7-32 illustrates these three types of PRC architectures.

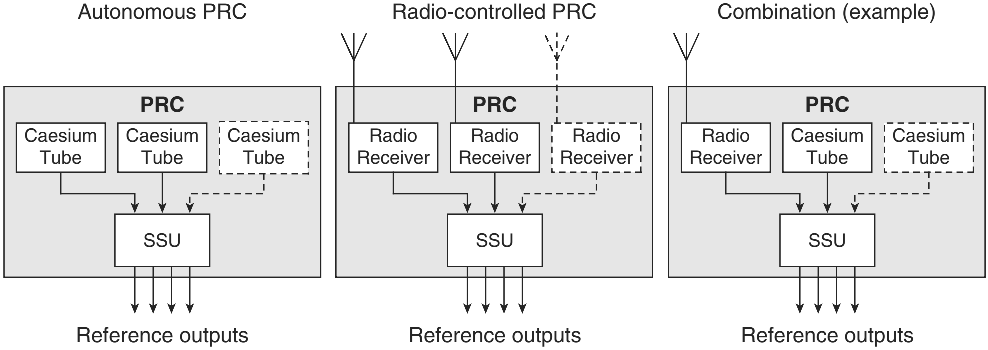


Figure 7-32 *PRS/PRC Architectures*

PRCs can be deployed in multiple different architectures. These architectures provide different levels of resiliency, complexity, and cost. In general, the following principles are adhered to in all PRC architectures:

- The synchronization distribution is tree-shaped.
- The synchronization network can be decomposed into multiple synchronization chains.

- Several stratum of slave clocks with different properties/roles exist.
- A higher-quality level is never slaved to a reference signal of a lower-quality.
- The SSU provides timing to a portion of the network. If the SSU's reference signal is lost, the SSU supplies timing to the network downstream.
- Radio-controlled PRCs use remote Cesium tubes in the radio infrastructure (either satellite-based, like GPS, or land-based, like Long Range Aid to Navigation [LORAN]-C) as the PRS.
- PRCs use a combination of local Cesium tubes and radio-based Cesium tubes.

Figure 7-33 provides an example of a synchronization network, including the Synchronization Supply Unit (SSU).

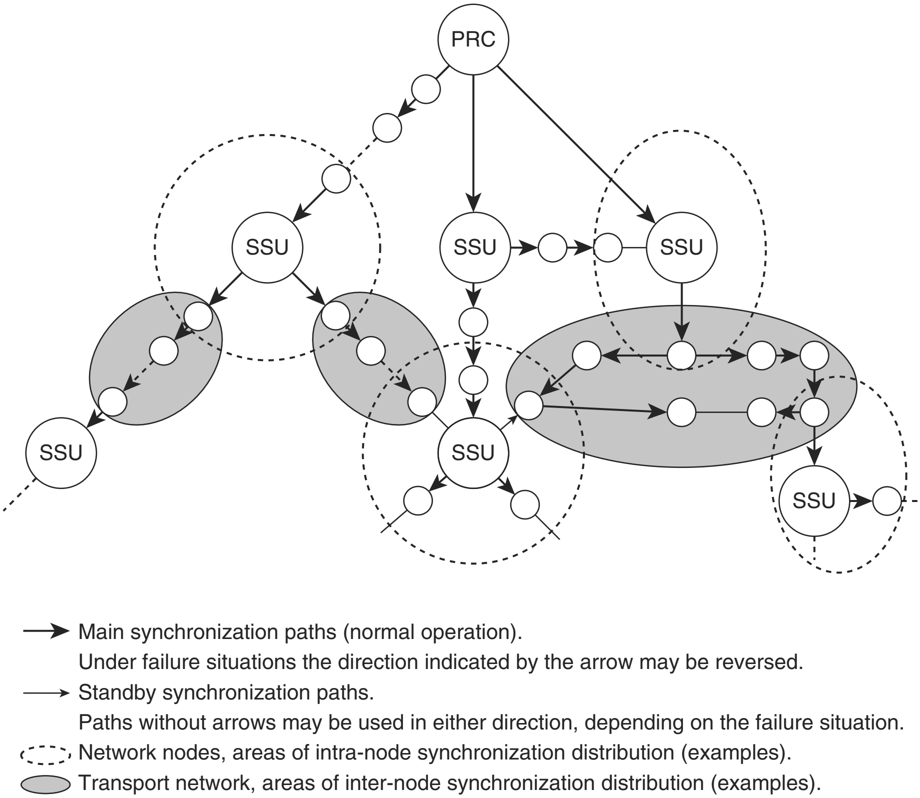


Figure 7-33 *Synchronization Network Example*

SSUs are used to provide reliable distribution of clock information. SSUs are part of every synchronization domain, including the PRC. SSUs receive clocking information from higher-layer clocks and distribute the clock information to all local equipment. SSUs also have the ability to provide accurate holdover mode, in the event that their clock source is lost.

The SSU does not belong to the transport network, but only provides the timing for the transport network elements within its synchronization domain.

There are two primary methods for providing clock synchronization, as follows:

- Master-slave synchronization, which has a single PRC from which all other clocks are synchronized. Synchronization in this method is achieved by sending timing signals from one clock to the next, in a hierarchical fashion. Figure 7-34 illustrates this master-slave synchronization network architecture.

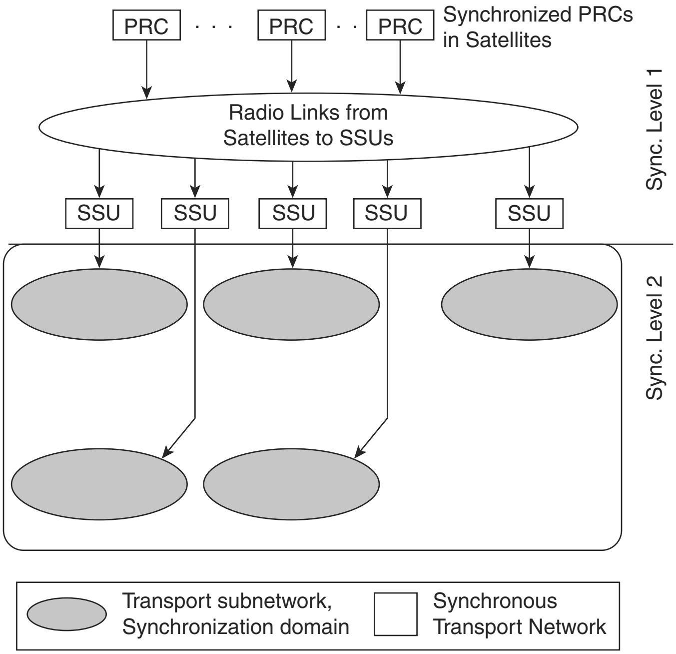


Figure 7-34 *Master-Slave Synchronization Network*

- Mutual synchronization, in which all clocks are interconnected. In this method, there is no unique PRC or hierarchical structure defined. Figure 7-35 illustrates this mutual synchronization network architecture.

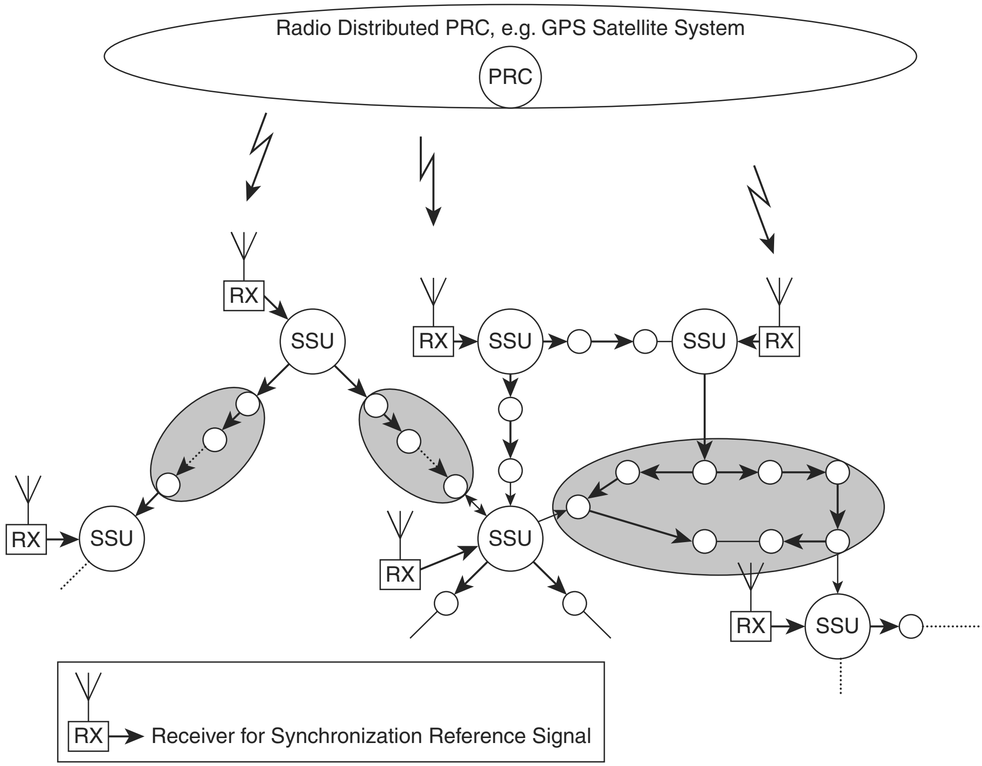


Figure 7-35 *Mutual Synchronization Network*

In practice, master-slave and mutual synchronization methods may be deployed in combination. In this architecture, the main PRC is usually an autonomous or combined PRC (see Figure 7-32). Synchronization from the main PRC is done in standard master-slave hierarchical fashion. At Level 2, the SSU is connected to both the PRC (primary) and an off-air PRC (backup).

Figure 7-36 illustrates this combined network architecture. Priorities for clock source and synchronization are identified in the figure.

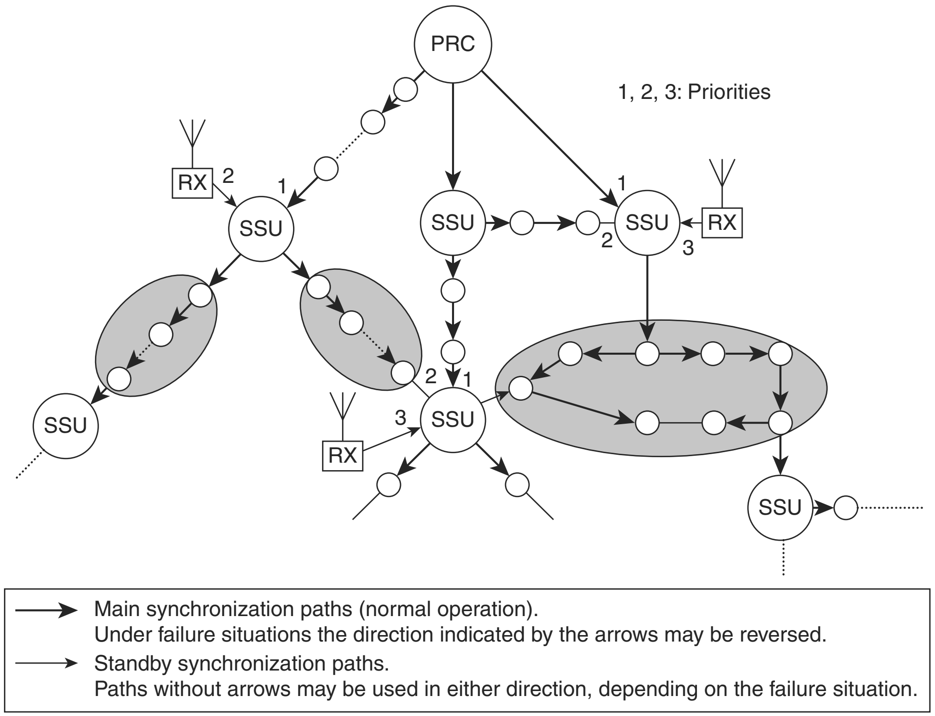


Figure 7-36 *Combined Master-Slave and Mutual Synchronization Network*

Timing Modes

Timing modes define what a clock is referenced to. Network elements may operate in four different timing modes, as follows:

- External timing, where the reference source signal is received via a local timing interface directly.
- Line timing, where the reference source signal is received from one or more data interfaces that also carries timing information.
- Loop timing, where the reference source signal is received from only one data interface as part of a ring topology.
- Through timing, where the reference source signal is transported transparently across the network element.

These timing modes map to four network architectures—synchronous networks, asynchronous networks, pseudo-synchronous networks, and plesiochronous networks.

Synchronous

A *synchronous network* is one where all clocks within the network have identical long-term accuracy. These networks require synchronization to avoid jitter and wander. Synchronous networks have a single active PRC source signal and rely on line timing to distribute clock information across the network. Figure 7-37 depicts a synchronous network that relies on line timing for clock source.

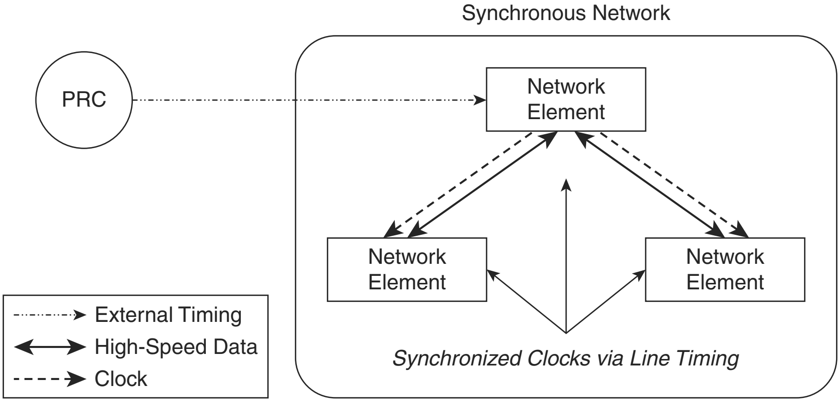


Figure 7-37 *Synchronous Network*

Asynchronous

An *asynchronous network* is one where not all clocks within the network have identical long-term accuracy due to multiple clock sources. In an asynchronous network, clocks are operating in free-running mode. These networks do not require that all clocks be synchronized to operate properly. Figure 7-38 depicts an asynchronous network.

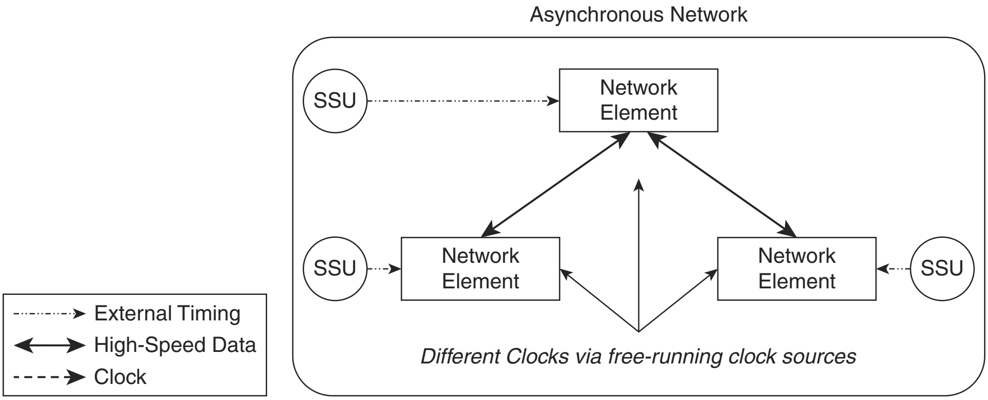


Figure 7-38 *Asynchronous Network*

Pseudo-Synchronous

A *pseudo-synchronous network* is one where not all clocks use the same PRC, but all rely on PRC-level accuracy for their reference source. These networks require synchronization to work properly. Figure 7-39 depicts a pseudo-synchronous network.

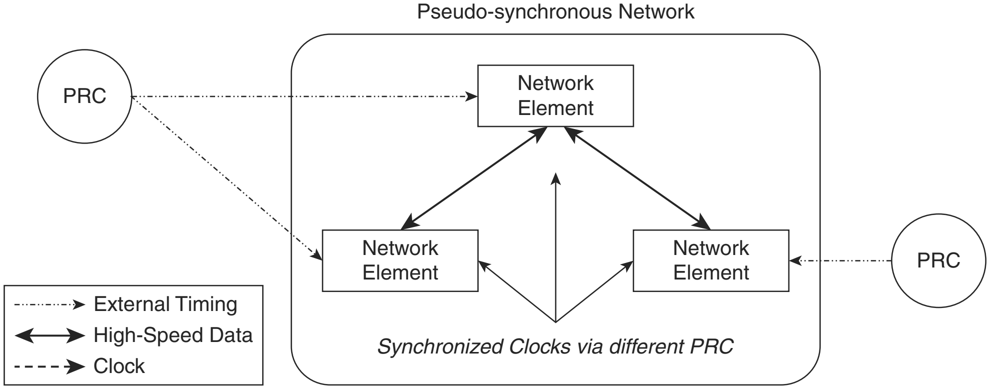


Figure 7-39 *Pseudo-Synchronous Network*

Plesiochronous

A *plesiochronous network* is one where different parts of the network are not perfectly synchronized with each other. Plesiochronous networks operate within a threshold of acceptable asynchronization; that is, two network elements act as if they are synchronized, but must accept and cope with time slips. A plesiochronous network is depicted in Figure 7-40.

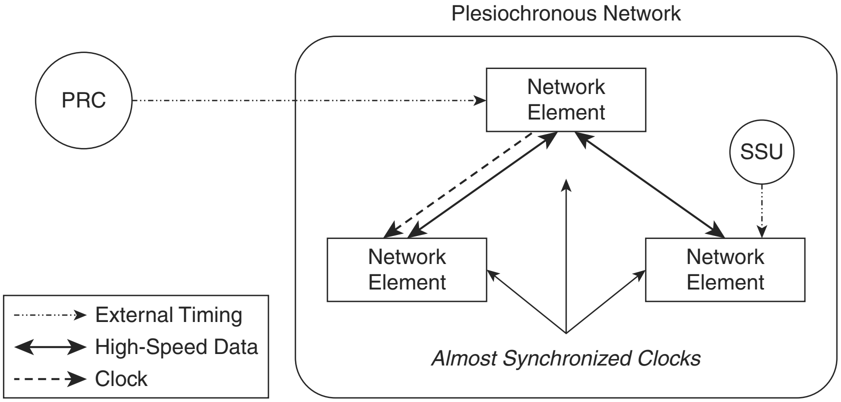


Figure 7-40 *Plesiochronous Network*

Packet-Based Timing

TDM networks are inherently synchronous. All network components must be synchronized with each other to ensure that data is not lost. In a native TDM network, clock synchronization is performed at the physical layer, and clocking information is carried along with data traffic. Clock slips occur when the receiver and transmitter have clocks that either run faster or slower than the other. These clock slips result in frames being either added or lost from the data stream.

IP networks, by nature, are asynchronous, and therefore cannot provide a constant bitrate. Packets reach their destination with random delay, known as jitter or Packet Delay Variation (PDV), already inserted. It is possible to remove random delay with a "jitter buffer," which temporarily stores all incoming packets and then forwards them at evenly spaced intervals; however, the original reference time is not available to determine what those evenly spaced intervals should be. Due to this, it is not possible to use the physical layer clock synchronization information from the native TDM frame for accurate clocking over pseudowires.

Although pseudowire endpoints do not need the clock synchronization information directly to implement the packet-switching functions, the constant bitrate applications that leverage the pseudowire transport must receive accurate timing information. This requires that the packet-switched network—that is, the pseudowire itself—provide this information to the applications. In such architecture, the reference clock may be connected directly to the synchronous network elements on each side of the pseudowire (see Figure 7-41) or to the pseudowire interworking function (see Figure 7-42).

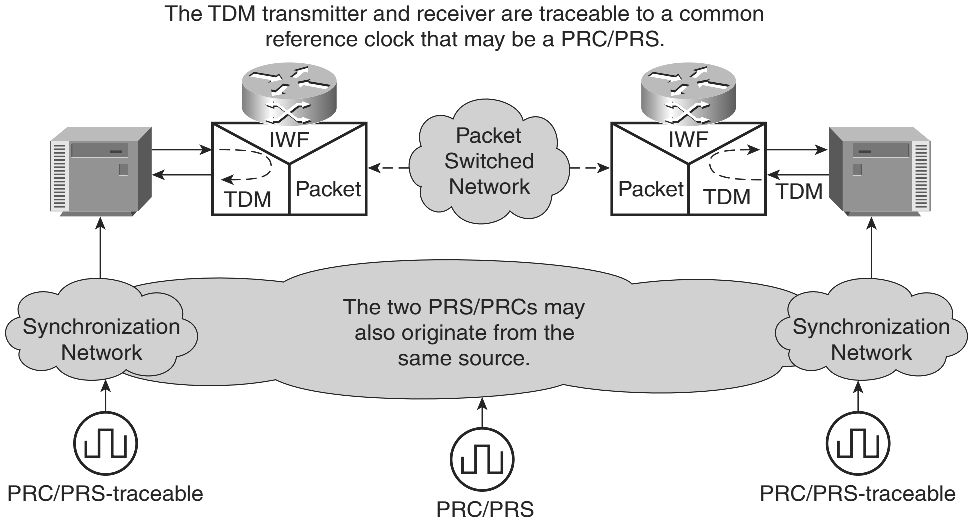


Figure 7-41 *Pseudowire Network Synchronization—Reference Clock Connected to Sync Network Elements*

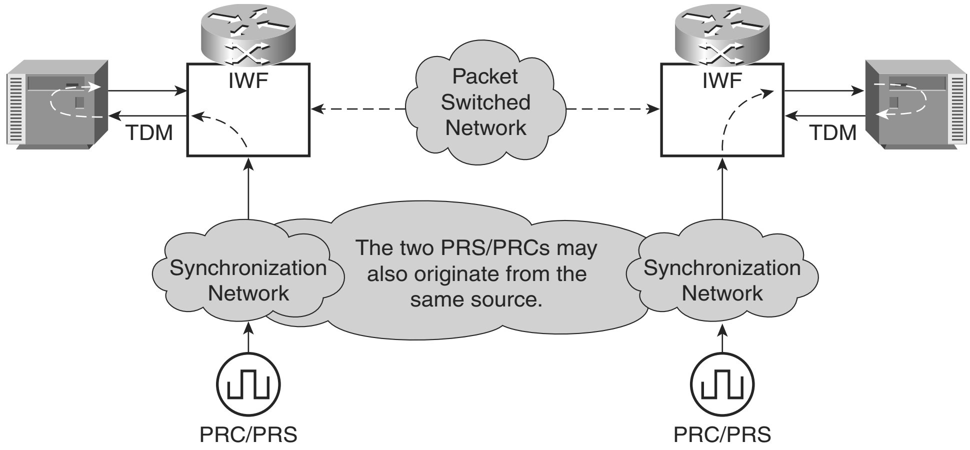


Figure 7-42 *Pseudowire Network Synchronization—Reference Clock Connected to Pseudowire IWF*

Although there are a large number of solutions for providing synchronization information, the same models presented previously in the "Timing Modes" section apply to packet-based networks, namely external timing, line timing, and loop timing.

Clock Recovery over Packet

Clock recovery is an important consideration when providing circuit emulation services over a PSN. The receiving Interworking Functions (IWF) must accurately recover the clock source from the sending IWF. There are two methods to provide clock recovery over packet, as follows:

- **Differential Clock Recovery** involves having a reference clock available at both sides of the pseudowire. Only the difference between the reference clock and the IWF service clock is transmitted across the pseudowire. Although this solution provides accurate frequency information and is tolerant to network delay, delay variation (jitter), and packet loss, the differential clock recovery solutions are expensive because they require multiple reference clocks. CESoPSN optionally may use differential clock recovery. Figure 7-43 illustrates differential clock recovery.
- **Adaptive Clock Recovery** involves having a reference clock available only at one side of the pseudowire. A timestamp is applied to all outbound packets by the sending IWF. The receiving IWF uses the information in the timestamp to recover the original reference clock information. Although this solution is less expensive (only a single reference clock is required), adaptive clock recovery is more susceptible to delay variation. TDMoIP uses adaptive clock recovery. Figure 7-44 illustrates adaptive clock recovery.

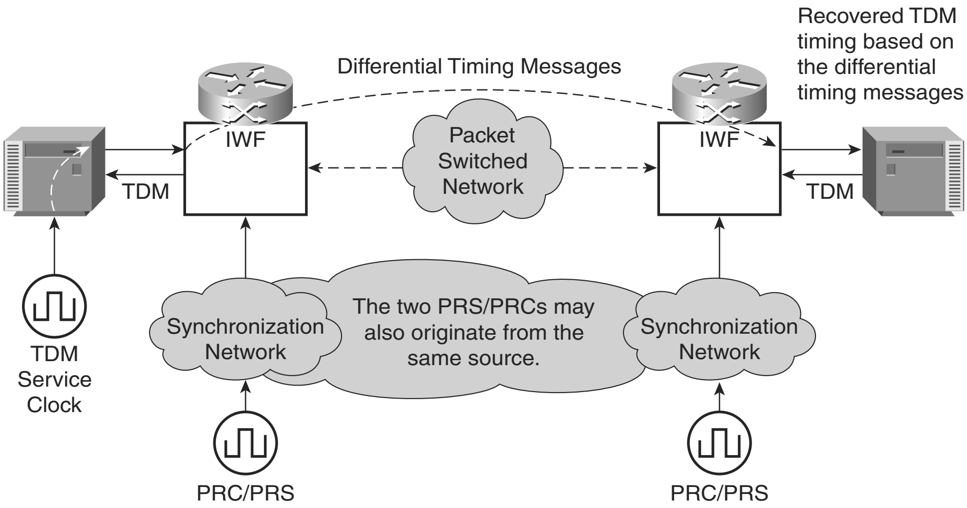


Figure 7-43 *Differential Clock Recovery*

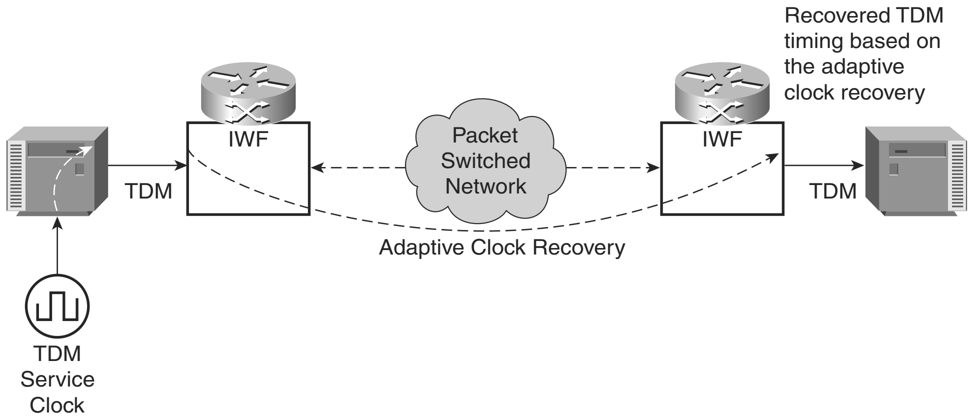


Figure 7-44 *Adaptive Clock Recovery*

Timing over Packet Solutions

There are four technologies for addressing synchronization over a packet network: Synchronous Ethernet (SyncE), Precision Time Protocol (PTP), Network Time Protocol (NTP), and Timing over IP Connection and Transfer of Clock BOF (TICTOC).

Any timing protocol should operate over the generic Internet with little or no intervention or management. Due to the unpredictable nature of the Internet, however, the accuracy of the protocol is greatly diminished. The accuracy of the frequency and time distribution is improved when operated over a managed network.

SyncE

Synchronous Ethernet (SyncE) is a line-timing method for transporting timing information over the Ethernet physical layer. Built on a Layer 1 model similar to SONET/SDH, SyncE provides accurate frequency synchronization, but does not provide time/phase synchronization.

SyncE specifications and requirements rely on four primary standards, as follows:

- **ITU-T G.8261:** Timing and synchronization aspects in packet network
- **ITU-T G.8262:** Timing characteristics of Synchronous Ethernet equipment slave clock
- **ITU-T G.8264:** Distribution of timing through packet networks
- **ITU-T G.781:** Synchronization layer functions

SyncE standards provide additional functionality to the 802.3 Ethernet standards while maintaining interworking between existing asynchronous Ethernet nodes and synchronous Ethernet nodes.

SyncE uses Synchronization Status Messages (SSMs) to transport timing information. Downstream clocks use the SSM for troubleshooting purposes, as the SSM will communicate if the clock source is a synchronized signal or derived from a free-running oscillator. These SSMs are transmitted using the Ethernet OAM protocol (ITU-T Y.1731 standard).

PTP

IEEE 1588v2, Precision Time Protocol, defined a protocol for precise, real-time, network-wide synchronization accuracy in the sub-millisecond range.

Each PTP domain consists of a number of clocks that synchronize with one another using the PTP protocol. Clocks within a PTP domain may not necessarily be synchronized with clocks within a different PTP domain.

Four types of clocks are defined within PTP, as follows:

- **An Ordinary Clock (OC)** has a single interface in a single PTP domain. The OC may be a master or slave, and may be responsible for providing time to an end node or application.
- **A Boundary Clock (BC)** has multiple interfaces in a single PTP domain. These interfaces may consist of multiple master interfaces, but only a single slave interface. The BC transfers all timing on the slave interface to the master interfaces. The BC can only be responsible for providing time to an application, not an end node.
- **A Transparent Clock (TC)** provides information on the time taken for a PTP message to transit the device and provides this information to all clocks receiving the PTP message. There are two types of transparent clocks:

- **A Peer-to-Peer Transparent Clock (P2P TC)** also provides corrections for any propagation delay on the link connected to the port receiving PTP messages.
- **An End-to-End Transparent Clock (E2E TC)** provides only the time taken for a PTP message to transit the device.

The PTP establishes a communications path across the network between all OCs and BCs. TCs may lie within the communications path, but, in general, P2P TCs and E2E TCs cannot be mixed in the same path.

Prior to synchronization, the clocks are organized into a master-slave hierarchy through a series of PTP Announce messages. The hierarchy contains a grandmaster, or PRC, multiple masters, and multiple slaves. This selection process is the Best Master Clock Algorithm (BMCA), which includes a clock class, based on where the clock has synchronized its timing from; clock accuracy, based on maximum accuracy threshold; and time source, based on the type of clock from which the advertising clock has received its timing (Atomic, GPS, Terrestrial Radio, PTP, Internal oscillator, and so on).

Synchronization in PTP

Once the hierarchy is established, each slave then synchronizes with its master using either a Delay Request-Response mechanism or a Peer Delay mechanism. These mechanisms cannot be mixed over the same communications path.

Delay Request-Response Mechanism The Delay Request-Response mechanism consists of four messages: Sync, Follow_Up (optional), Delay_Req, and Delay_Resp. Sync and Follow_Up messages are typically multicast, but may be unicast. Delay_Req and Delay_Resp are typically unicast messages between master and specific slave. Figure 7-45 illustrates this Request-Response mechanism.

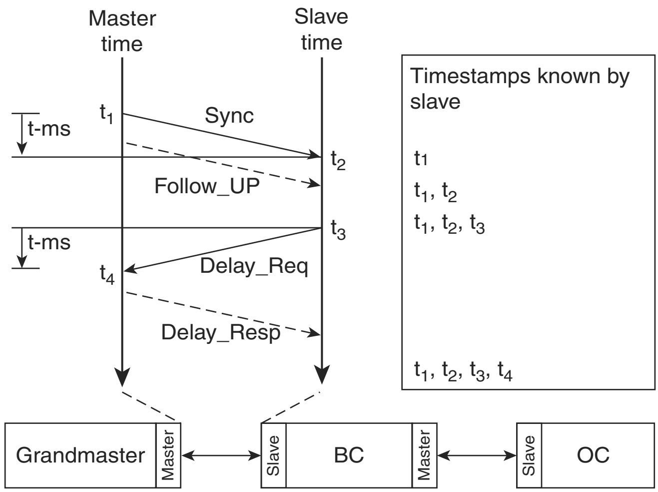


Figure 7-45 *PTP Delay Request-Response Mechanism*

Propagation time, or transit time, and an *offset*, or processing time, are calculated during this process.

Assuming a symmetrical link:

- The propagation time is $[(t_2-t_1)+(t_4-t_3)]/2$.
- The offset is t_2-t_1-(propagation time).

Assuming an asymmetrical link:

- The propagation time is the average of the slave-to-master and master-to-slave propagation times.
- The offset is the difference between the actual master-to-slave time and the average propagation times.

Peer Delay Mechanism The Peer Delay mechanism is limited to point-to-point communications paths between two OC, BC, or P2P TC. The Peer Delay mechanism is also symmetric; that is, it operates separately in both directions.

The Peer Delay mechanism consists of five messages: Sync, Follow_Up (optional), Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up (optional). Figure 7-46 illustrates this Request-Response mechanism.

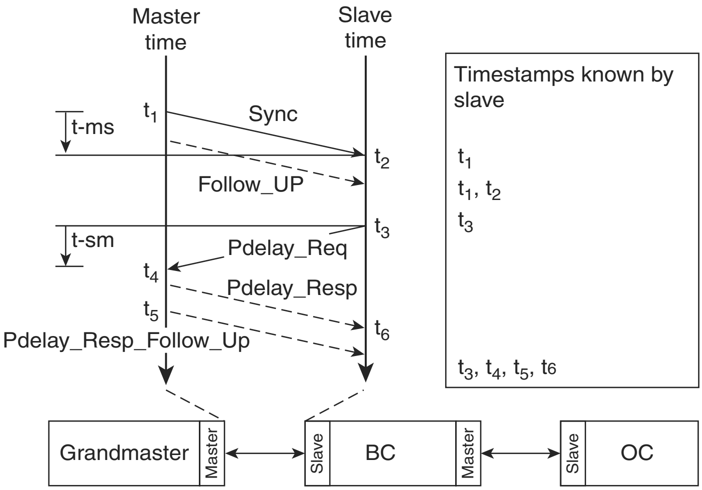


Figure 7-46 *PTP Peer Delay Mechanism*

A propagation time, or the transit time, and an offset, or the processing time, are calculated during this process.

Assuming a symmetrical link:

- The propagation time is $[(t_4-t_3)+(t_6-t_5)]/2$.
- The offset is t2-t1–(propagation time).

Assuming an asymmetrical link:

- The propagation time is the average of the slave-to-master and master-to-slave propagation times.
- The offset is the difference between the actual master-to-slave time and the average propagation times.

When using an E2E TC in the network, the E2E TC is not synchronized at all. Instead, the E2E TC timestamps the Sync message or Follow_Up message on both ingress and egress, and computes the time taken for the message to traverse the node from these timestamps. This time is the residence time, and is included in the message as a Correction field, such that OC and BC may account for this processing time. Each E2E TC in the chain adds its own residence time to the value already contained in the Correction field.

PTP Profiles and Conformance

PTP supports extensible profiles that allow for transport of optional features and attribute values, including interworking and desired performance levels required for a particular application. These profiles are created by numerous third parties, such as standards or industry organizations and vendors.

Network nodes are required to conform to the normative sections of the IEEE 1588 standards and at least one PTP profile. IEEE 1588 defines two default profiles: Delay Request-Response Default PTP Profile and Peer-to-Peer Default PTP Profile. In addition, a network node may comply with certain optional sections of the standards but must implement the optional section in its entirety.

NTP

The Network Time Protocol (NTP) is the most predominant method of synchronizing clocks on the Internet. The National Institute of Standards and Technology (NIST) estimates over 10 million NTP servers and clients deployed in the Internet.

The most recent version, NTPv4, extends upon previous versions (NTPv3–RFC 1305) by introducing accuracy to the tens of microseconds (with a precision time source, such as a Cesium oscillator or GPS receiver), dynamic discovery of servers, and includes an extensibility mechanism via options.

A NTP node operates as either a Primary (Stratum 1) server, a Secondary (Stratum 2) server, or a client. Primary servers synchronize to national time standards via radio (terrestrial or satellite). A client synchronizes to one or more upstream servers, but does not provide any synchronization services to downstream nodes. A Secondary server synchronizes to one or more Primary servers and also provides synchronization services to one or more downstream servers or clients.

NTP Protocol Modes

There are three NTP protocol modes: client/server, symmetric, and broadcast.

In client/server mode, clients and servers send unicast packets to each other. Servers provide synchronization services to the clients, but do not accept synchronization from them. In client/server mode, clients are responsible for pulling synchronization from the server.

In symmetric mode, a peer functions as both a client and server. Peers provide synchronization services and accept synchronization from other peers. In symmetric mode, peers push and pull synchronization from each other.

In broadcast mode, a server sends periodic broadcast messages to multiple clients simultaneously. On instantiation of communication, unicast messages are sent between client and server such that the client can accurately calculate propagation delay. Following this unicast exchange, the client listens for broadcast messages generated by the server. In broadcast mode, the broadcast server pushes synchronization to clients.

Offset

The basic operation of NTP synchronization involves determining the offset in clock from one network node to another. This works as follows:

1. The NTP client sends a packet to a specified NTP server. In this packet, it stores the time the packet left as defined by its clock (t_1).
2. The NTP server receives the packet and notes the time it received the packet, according to its clock (t_2), and the time the client sent the packet (t_1).
3. The NTP server sends a packet back to the client and includes what time it was sent according to its clock (t_3). The packet sent back contains three timestamps: t_1, t_2, and t_3.
4. The client receives the packet from the server and notes what time it receives the packet according to its clock (t_4).

Figure 7-47 illustrates this synchronization flow.

Assuming a symmetrical link:

- The propagation delay is $(t_4\text{-}t_1)–(t_3\text{-}t_2)$.
- The clock offset is $[(t_2\text{-}t_1)\text{-}(t_4\text{-}t_3)]/2$.

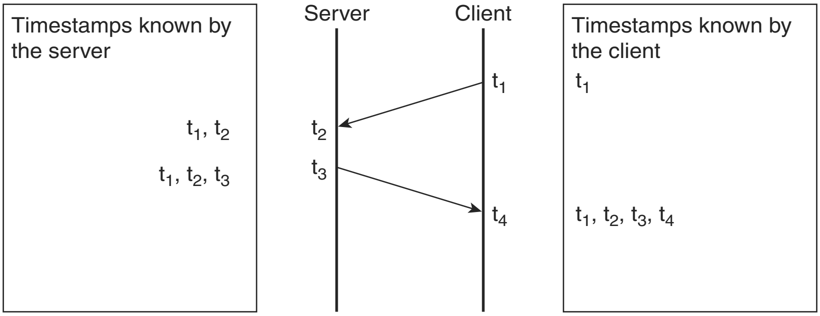


Figure 7-47 *NTP Offset Calculation*

Although the basic algorithm seems simple, the NTP architecture is more complex than expected. Once a server sends information to a client, the client uses a combination of clock/data filter, selection, clustering, and combining algorithms to determine its local offset. Figure 7-48 depicts a typical NTP architecture.

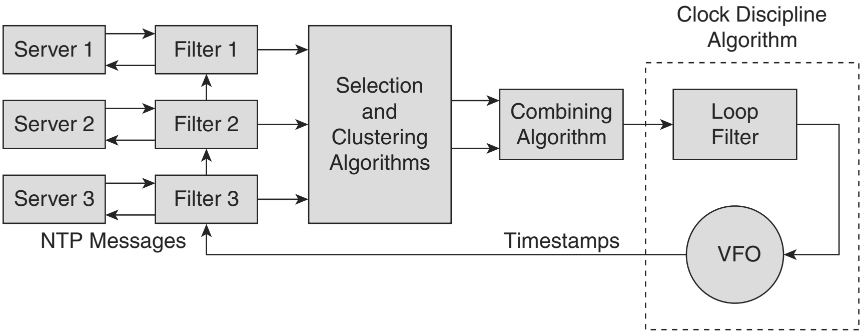


Figure 7-48 *NTP Architecture*

NTP Clock Filter Algorithm

The NTP clock filter algorithm analyzes the stream of NTP data received to determine which samples are most likely to represent accurate time. The algorithm produces the offset, delay, dispersion (maximum error in measurement), jitter, and time of arrival information that is used to calculate the final offset for the system clock. These values are also used to determine if the server is functioning properly and whether it can be used as a reference source. The NTP clock filter algorithm actually consists of four other algorithms, as follows:

- The NTP selection algorithm scans the stream of NTP data and discards samples that are clearly incorrect, known as *falsetickers*, and keeps only those that appear to be accurate, known as *truechimers*. Falsetickers may be caused by the long-tail effect of Packet-Delay Variation (PDV) or network degradations, such as congestion and reroutes caused by node failures.

- The NTP cluster algorithm then discards those samples that are statistically furthest from the mean until a minimum number of samples remain.
- The NTP combine algorithm produces the final values based on a weighted average calculation from the samples remaining.
- The NTP clock discipline algorithm takes the final values output from the combine algorithm and uses these to discipline the local clock.

NTP Poll Interval

The NTP poll interval is the term used to define how often a new calculation of offset should be made. The poll interval is determined dynamically by the clock discipline algorithm based on the observed clock offset measurements. The poll interval will increase if the internal oscillator frequency stays constant. If the oscillator frequency changes, the poll interval will decrease in order to track these changes.

NTP Security Considerations

Because NTP broadcast clients are vulnerable to broadcast storms from spoofed or misbehaving NTP broadcast servers, NTP includes an optional authentication field. This optional authentication field supports MD5 encryption. This encryption can be negotiated between a broadcast client and server during instantiation.

TICTOC

The Timing over IP Connection and Transfer of Clock BOF (TICTOC) draft standard was written to provide a robust IP/MPLS-based time and frequency distribution architecture. TICTOC, like other protocols, can be decomposed into two layers corresponding to time and frequency. Implementations may vary depending on the exact need of the application. For example, if an application or network node only needs time synchronization and not frequency synchronization, only the time layer may be present.

Figure 7-49 illustrates the TICTOC layers.

TICTOC Clients

TICTOC clients are comprised of up to four modules (illustrated in Figure 7-50)—the frequency acquisition module, the frequency presentation module, the time acquisition module, and the time presentation module.

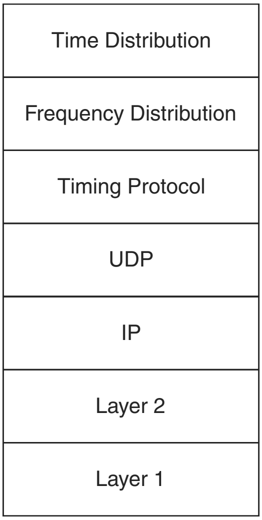


Figure 7-49 *TICTOC Layers*

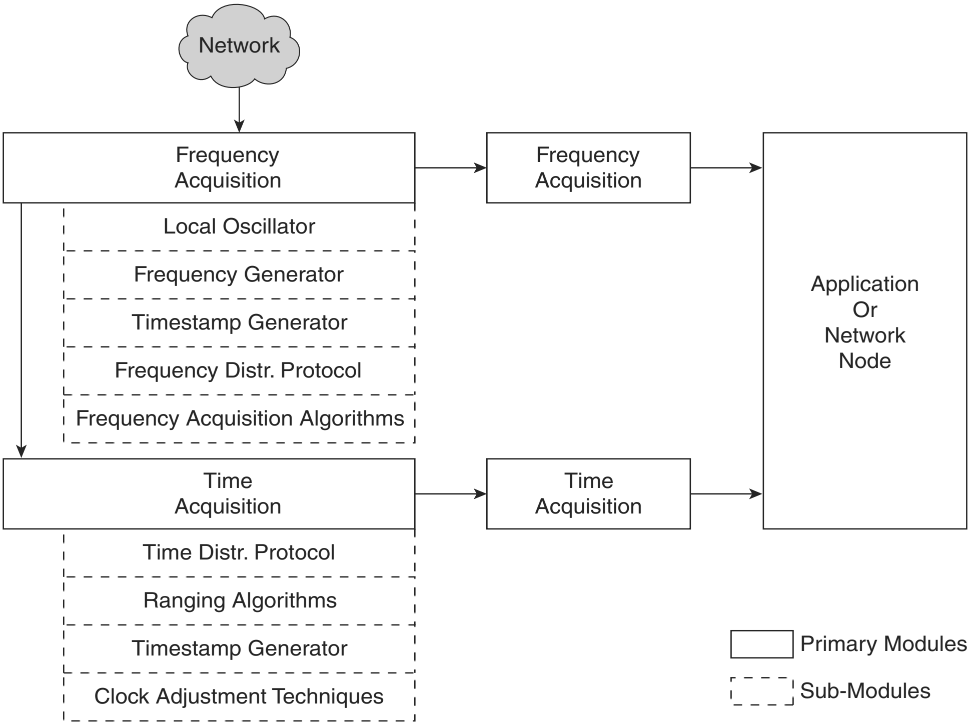


Figure 7-50 *TICTOC Client Modules*

Frequency Acquisition Module

The *frequency acquisition module* retrieves the frequency information distributed over the network. The frequency acquisition module may be divided into the following sub-modules. These sub-modules may not all be present depending on requirements and implementation:

- Local oscillator. Both master and clients in the TICTOC architecture need a local oscillator. The master uses a Cesium clock, whereas the clients may use a lower-accuracy oscillator, such as quartz crystal. The client's local oscillator must be adjusted to match the master's oscillator to ensure synchronization. Disciplining the local oscillator is based on arrival time and information received in packets from the frequency distribution protocol.
- Frequency generator.
- Timestamp generator.
- Frequency distribution protocol. The frequency distribution protocol is used to distribute frequency across the network. This protocol may be the same or different than the protocol used to distribute time. Frequency distribution protocols in TICTOC are one-way, and may be unicast, multicast, or broadcast. Although multicast distribution is supported, there is the inherent risk that a multicast replication operation may add variable delay.
- The frequency acquisition algorithms are used to re-acquire the source time from the received packets. As with all packets in a PSN, the packet distribution protocol packets are subject to Packet Delay Variation (PDV). The frequency acquisition algorithms are used to filter out the PDV through a two-step process, as follows:
  - **Packet Selection and Discard Algorithms:** These algorithms are similar to those used in NTP to eliminate those packets that would lead to accuracy degradation of the recovered frequency. This packet selection algorithm works by selecting a series of packets that are all similar in result, as long as the sample still represents a relatively high percentage of packets.
  - **Filtering and control servos:** With linear averaging, the time to calculate and eliminate frequency drift effects would be so high that the frequency difference calculated would no longer be relevant. "Control loops" are used to accurately model the frequency drift effects on sampled packets in a non-linear manner.

Frequency Presentation Module

If the frequency is needed by the application, the *frequency presentation module* formats this information into the application-specific requirements. Presentation methods may include graphical display, clock discipline, and so on.

Time Acquisition Module

The *time acquisition module* requires a stable frequency reference. Even if frequency is not needed, the time acquisition module may rely on the frequency acquisition module to retrieve information. The time acquisition module may also retrieve information from an external source, such as a GPS receiver. This module allows multiple TICTOC clients to share a common offset. The time acquisition module may be divided into the following sub-modules. These sub-modules may not all be present depending on requirements and implementation:

- **Time distribution protocol:** The time distribution protocol is used to accurately synchronize a clock based on measured offset between client and master oscillators. Ranging algorithms are used to estimate this offset. Time distribution protocol, unlike frequency distribution protocol, is typically bi-directional, requiring both client and master to send and receive packets.
- **Ranging algorithms:** Ranging is the estimation of propagation delay within a network. This is done in a manner similar to PTP, where a packet is sent from the master with a timestamp, followed by a second packet with a timestamp. These timestamps on the packets indicate the time that the master injected the packet into the network. The client then has sufficient timestamp information to calculate the propagation delay.
- Clock adjustment techniques
- Timestamp generator

Time Presentation Module

The *time presentation module* formats information from the time acquisition module into a format that is relevant to the application requiring synchronization.

Generic Modules

TICTOC supports various generic modules that may be applied to frequency distribution, time distribution, or both. These modules include enhanced security (certificate-based), auto-discovery of masters, master clock selection algorithms, OAM, performance monitoring, and network management.

Combining Protocols

SyncE, PTP, NTP, and TICTOC need not be mutually exclusive within a network. A combination of these protocols, along with external means of synchronization, may be leveraged. For instance, because SyncE provides a highly accurate frequency source and TICTOC provides a highly accurate time source, a network may use the SyncE's physical layer frequency synchronization as the source for TICTOC's IP layer time synchronization input.

Summary

This chapter discussed one of the predominant IP migration methods for today's mobile networks. Whether driven by technological or financial decisions, backhaul network evolution to IP-based mechanisms is a clear operator strategy, and pseudowire transport mechanisms provide a bridge between the legacy TDM systems presented in Chapter 2 and the All IP systems presented in Chapter 3. Although not without their share of complexity, including time and frequency synchronization, IP backhaul networks are an obvious value to mobile operators. With such a large number of solutions for both pseudowire transport and synchronization, mobile operators need to understand the technologies themselves and determine which best meets their requirements.

Endnotes

[1]Source: ABI Research

[2]Source: ABI Research

This page intentionally left blank

Index

Numbers

A

B

C

D

E

F

G

H

I

J-K

L

M

N

O

P

Q

R

S

T

U

V

W-X-Y-Z