WILEY

Second Edition

Internet Communications Using SIP

Delivering VoIP and Multimedia Services with Session Initiation Protocol

Henry Sinnreich
Alan B. Johnston

Internet Communications Using SIP

Delivering VoIP and Multimedia Services with Session Initiation Protocol Second Edition

Henry Sinnreich
Alan B. Johnston

WILEY
Wiley Publishing, Inc.

Internet Communications Using SIP
Second Edition

Internet Communications Using SIP

Delivering VoIP and Multimedia Services with Session Initiation Protocol Second Edition

Henry Sinnreich
Alan B. Johnston

WILEY
Wiley Publishing, Inc.

Internet Communications Using SIP: Delivering VoIP and Multimedia Services with Session Initiation Protocol, Second Edition
Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published simultaneously in Canada
ISBN-13: 978-0-471-77657-4
ISBN-10: 0-471-77657-2
Manufactured in the United States of America
10 9 8 7 6 5 4 3 2 1
2MA/QW/QX/QW/IN

Library of Congress Cataloging-in-Publication Data
Sinnreich, Henry.
Internet communications using SIP : delivering VoIP and multimedia services with Section Initiation Protocol / Henry Sinnreich, Alan B. Johnston. — 2nd ed.
p. cm.
Includes index.
ISBN-13: 978-0-471-77657-4 (cloth)
ISBN-10: 0-471-77657-2 (cloth)
1. Computer network protocols. 2. Internet telephony. 3. Multimedia systems. I. Title.
TK5105.55.S56 2006
621.3850285'4678—dc22
2006009325

We could not have written this book without the support of our forgiving spouses, Fabienne and Lisa, who held the fort while we were working on SIP. And to both our family members shouting, "Your SIP phone is ringing."

About the Authors

Dr. Henry Sinnreich (Richardson, TX) is Chief Technology Officer at Pulver.com, a leading media company for VoIP and Internet communication services. Dr. Sinnreich has held engineering and executive positions at MCI where he was an MCI fellow and has been involved in Internet and multimedia services for more than 12 years, including the development of the flagship MCI Advantage service based on SIP. Henry Sinnreich is also a contributor to IETF standards for Internet communications in such areas as SIP telephony devices and using RTP extensions for voice quality monitoring. He was awarded the title Pioneer for VoIP in 2000 at the VON Europe conference. Henry Sinnreich has been a cofounder and board member of the International SIP Forum based in Stockholm. He is a frequent speaker and is known as the leading evangelist, worldwide, for SIP based VoIP, presence, IM, multimedia, and integration of applications with communications. Dr. Sinnreich is also a guest lecturer at the Engineering School of the Southern Methodist University in Dallas, TX.

Alan B. Johnston (St. Louis, MO) is a Consulting Member of Technical Staff at Avaya, Inc. He has coauthored the core Internet SIP standard RFC 3261 and four other SIP related RFCs. He is the co-chair of the IETF Centralized Conferencing Working Group and is on the board of directors of the International SIP Forum. His current areas of interest include peer-to-peer SIP and security. Dr. Johnston is a frequent speaker and lecturer on SIP and contributor to various publications, and is an adjunct professor at Washington University in St. Louis, MO.

Credits

Executive Editor
Carol Long

Development Editor
Kevin Shafer

Production Editor
Pamela Hanley

Copy Editor
Foxxe Editorial Services

Editorial Manager
Mary Beth Wakefield

Production Manager
Tim Tate

Vice President and Executive Group Publisher
Richard Swadley

Vice President and Executive Publisher
Joseph B. Wikert

Project Coordinator
Ryan Steffen

Graphics and Production Specialists
Stephanie D. Jumper
Heather Ryan
Alicia B. South

Quality Control Technician
Laura Albert

Proofreading and Indexing
Joe Niesen
Palmer Publishing Services

Contents

Foreword

About 10 years ago, the first drafts describing the Session Initiation Protocol (1996) were published, with the rather modest ambition of setting up multicast groups for multimedia conferences. In the intervening decade, a draft of about 20 pages has turned into an ecosystem of dozens of RFCs, hundreds of Internet drafts—and several books, conferences, and a magazine. It has become difficult to get a feel for the overall landscape, to distinguish the important core concepts from the niche applications. This book offers a detailed, technically informed, yet accessible, introduction to the overall SIP ecosystem, suitable both for someone who needs to understand the technology to make strategic decisions and implementers who need to build new components.

SIP is part of the second wave of Internet application protocol. While the first wave largely focused on asynchronous communications (such as e-mail, and data transfer), this second wave introduces the notion of interactive, human-to-human communication that allows integration with any media, not just voice. As SIP and interactive communications have matured, the goal for human-to-human communication has shifted. Initially, cell phones promised voice communication at any time, at any place. Multimedia communications, on PCs and maybe emerging cellular networks, allow us to add "any media." However, the "any time, any place, any media" can also turn us into slaves of our communications devices, interrupting our ability to think, to eat in peace, and to meet in person. Thus, our goal has to be to design communications technology that offers the right media, at the right place, and at the right time. With some of the advanced functionality of SIP, such as presence, location-based services, user-created services, and caller preferences, we can get closer to creating communication systems that support our work and enhance our personal life.

With new communications technologies, there is always the temptation to mimic the old. E-mail inherited aspects of the interoffice memo and fax; web pages attempted to look like newsprint and brochures. However, in VoIP, there is the particular temptation to recreate old technology features, as interoperability with the old PSTN will remain important for at least another decade. Fax-to-email gateways were never quite as important as VoIP-to-PSTN gateways. This emphasis on interoperability with 100-year-old technology has provided a financial motivation—provide the same service more cheaply. However, this may also hold back the promise offered by Internet-based multimedia communications, such as the integration of presence, the ability not just to communicate by voice and maybe video but also to share any application, or the ability to customize the user experience and integrate interactive communications with existing Internet tools and applications. Just as most microprocessors are embedded in household appliances and cars, not desktop PCs and laptops, we might find that Internet-based voice and multimedia communications will be integrated into games, appliances, and cameras, or be hidden behind a link on a web page, rather than dialed by name or number. As for many of the most innovative applications, users will likely not even consider them phone services at all, but extensions that make some other application more productive or more fun.

This book is like a good tour guide to a foreign country. It doesn't just describe the major sites and tourist attractions; it lets the reader share in the history, spirit, language, and culture of the place. Natives write the best tour guides, and the authors have been living and working in SIP land since it was a small outpost in one large country called the IETF. The authors have served as ambassadors in lands near and far, but have also made major contributions to the development of this part of the Internet landscape, always reminding others of the original goals of the first inhabitants. After taking the tour, the reader will be ready not just to show off a stamp on a passport or certificate but also to contribute to new modes of communications. SIP land is still young and needs lots of pioneers who can push the frontiers of Internet-enabled communications. There might not always be gold in those hills, but enriching human communications will always be its own reward.

Henning Schulzrinne
Professor, Columbia University

Acknowledgments

We have enjoyed the benefit of early and significant support from colleagues and management in MCI. Vint Cerf was, as mentioned, one of the early supporters, and so were Teresa Hastings, John Gallant, Bob Spry, and Robert Oliver who first took the responsibility for developing and deploying SIP in their respective engineering departments. John Truetken, Lance Lockhart, and many other engineers in MCI also had critical contributions to the implementation of SIP. Fred Briggs, Patrice Carroll, Barry Zip, and Leo Cyr from MCI helped with the challenge to develop marketable services based on SIP. We were fortunate to work jointly in the development and deployment of SIP services with Steve Donovan, Diana Rawlins, Dean Willis, Robert Sparks, Ben Campbell, Chris Cunningham, Kevin Summers, and many other engineers from MCI and elsewhere in the industry engaged in the development of SIP in the Internet Engineering Task Force (IETF).

Most ideas and inspirations driving SIP are due to Prof. Henning Schulzrinne from Columbia University and to Jonathan Rosenberg from DynamicSoft and are reflected in this book. Among the many industry contributors, we gratefully acknowledge discussions and guidance from Rohan Mahy from Cisco Corporation, Gonzalo Camarillo and Adam Roach from L.M. Ericsson. Jiri Kuthan from GMD Focus, Berlin, was helpful with SIP tutorial charts and with discussions in transatlantic calls using SIP phones—again, calls of crystal clear clarity to our surprise. The authors are grateful to Richard Shockey from NeuStar, Inc. and Douglas Ranalli from NetNumber, Inc. for numerous discussions regarding ENUM. Theodore Havinis has contributed to the SIP-QoS-AAA aspect for mobile users.

We acknowledge countless helpful discussions and insight from many participants in the IETF and especially to Scott Bradner for holding the authors and others in the IETF SIP community in line to the true conceptual, technical, and procedural spirit of the Internet.

Jeff Pulver has played a special role in providing a platform and leading exhibition of products for what was initially an obscure and unknown protocol in the Voice ON the Net (VON) and other conferences held in America, Europe, and Asia.

Carrol Long, Kevin Shafer, and Adoabi Obi Tulton from John Wiley & Sons have been instrumental in editing this book.

Introduction

The second edition of *Internet Communications Using SIP* had to be rewritten almost from the ground up, because of the dramatic changes in the industry in the five years that have passed since the first edition. Some of the developments had been envisaged in the first edition, but naturally, some have not.

The Internet Has Replaced the Telephone System and the Telecommunication Networks

Since the publication in 2001 of the first edition of this book, *Internet Communications Using SIP*, Voice over IP (VoIP) has developed from an emerging technology to the recognized replacement of existing global telephone systems based on Time Division Multiplex (TDM) circuit switching. The Internet has also replaced the proposed connection-oriented offsprings of TDM, such as the Integrated Services Digital Network (ISDN) and the Asynchronous Transfer Multiplex (ATM) based broadband version BISDN, envisaged for the telecommunications industry by the International Telecommunications Union ITU-T standards body. TDM, ATM, ISDN, and BISDN are now history.

All wired and wireless communications are instead migrating to the Internet standards developed by the Internet Engineering Task Force (IETF). The legacy telecommunication networks, while still dominant, are recognized as a present-day cash cow only and are scheduled for replacement by IP networks.

The end-to-end nature of the Internet that places intelligence in the applications running in the endpoints and gives control to the user at the endpoints has indeed replaced TDM-based telephony with central control. The Internet

has also proven to be the home network for other types of communications, information, entertainment, and data applications. To quote Jon Peterson, area director of the IETF:

"The Internet is the service."

The Session Initiation Protocol Is the Standard for VoIP and Multimedia Communications

Another change from the first edition of this book is the Session Initiation Protocol (SIP), which has been adopted by practically all public VoIP service providers for wired and wireless communications. The discussions about SIP versus H.323 standardized by the ITU-T are over as well. The installed base of H.323 is considered a liability and planned for replacement by SIP sooner or later.

A global industry has emerged to take advantage of SIP and its associated IETF standards for real-time communications. More than 560 VoIP service providers have been reported [1] in early 2006, most of them using SIP-based networks. The list of SIP-based equipment (such as SIP phones, software for PCs, and mobile devices, servers, gateways, and so on) is now large and still growing. Actually, all equipment and system vendors are now supporting SIP.

Presence and Instant Messaging Are Mainstream Communications

Presence and instant messaging (IM) are now mainstream with consumers and, in the enterprise, complementing or sometimes replacing voice communications in specific situations (such as in circumstances where silence is required). Even for VoIP, presence has emerged not only as a valuable enhancement, but presence may be the dial tone of the twenty-first century.

Presence and event-based communications have enabled the integration of communications with applications. Presence and IM are discussed in Chapter 13, "Presence and Instant Messaging."

The so-called IM services provided by large Internet companies, such as AOL, Apple, Google, IBM, Microsoft, Skype (not SIP-based), and Yahoo!, actually carry at present most of the public VoIP traffic between end users around the globe.

It is not far-fetched to see the IM Internet companies replacing the former telephone companies in the voice communication business. Many legacy telecommunication companies are also using VoIP to replace the internal TDM voice networks, but their VoIP services may not survive the advanced technologies deployed by the IM Internet companies and the challenge posed by peer-to-peer (P2P) communications.

Redefining Communications: Mobility, Emergency and Equal Access for the Disabled

Internet communications have been known not to be dependent on the location on the Internet. Application-level mobility based on SIP is a key component to seamless mobile communications, as discussed in Chapter 15, "SIP Application Level Mobility."

Emergency calling services by users in distress using the Internet (such as 911 in the United States or 112 in Europe) are far more powerful and cost less than the Public Switched Telephone Network (PSTN) based emergency services. Internet-based emergency calling is indeed in the design stage in a number of countries. Chapter 16, "Emergency and Preemption Communication Services," discusses Internet-based emergency services.

The multimedia nature of Internet communications gives hearing- and speech-impaired people the opportunity to fully participate in rich communications for work and in personal life. Chapter 17, "Accessibility for the Disabled," discusses access to communications for disabled people.

The Rise of Peer-to-Peer Communications

P2P traffic has risen in the Internet since around 2000 and became the dominant part of Internet traffic by 2004. Since 2004, Skype (which is based on P2P VoIP, IM, and presence) has also become by far the dominant VoIP provider worldwide. Since P2P SIP standards work is just emerging as of this writing, Skype can be considered a prestandard P2P Internet communication service.

The reasons for the emergence of overlay networks and P2P applications and their nature are discussed in Chapter 20, "Peer-to-Peer SIP," and also in Chapter 6, "SIP Overview." Though the present VoIP industry is built on client-server (CS) SIP, this may significantly change. To quote David Bryan from `p2p.org`:

> *"P2P SIP may change VoIP to the same extent that VoIP has changed telecommunications."*

VoIP and Multimedia Communications Services Are Still Fragmented

In spite of all the technological progress, VoIP, IM, presence, and multimedia services are still a highly fragmented industry:

- Telephone services based on VoIP operate as islands and can interconnect (as of this writing) using mostly the legacy Public Switched Telephone Network (PSTN). The service model is giving broadband users

access to the legacy telephone system, actually a voice gateway service between the Internet and TDM. The business model of most VoIP service providers is just lower cost for legacy-style telephone service, also called *PSTN over IP*. The PSTN gateway services are using IP inside their networks, but users are not exposed to the rich IP services, except when all parties are on the same network.

- The most successful public voice, IM, and presence service is Skype, which is not standards-based.
- Walled gardens: The fragmentation of communications is still actively pursued by most mobile service providers by deploying systems where their users can get rich IP multimedia services only on their own networks. The fees to communicate between mobile service providers are a significant part of the business model, and open connectivity to the Internet ("Internet neutrality") is still a hotly debated issue. Internet neutrality is also still debated by many broadband Internet access providers (such as DSL and cable companies), although we believe that enlightened government regulators in the developed countries will weigh in favor of users and open network access in general.

The proliferation of islands for communications makes them less useful the more there are, since this proliferation is in denial of Metcalf's law that the value of a network increases with the square of the number of points attached to the network. The Internet with more than *1 billion* attached endpoints has thus the highest value for communications. By contrast, the mobile phone industry boasts *3 billion* users, but in many fragmented networks.

Past Obsessions and Present Dangers: QoS and Security

Network-based quality of service (QoS) for voice and the reliability of the legacy telephone network have long been used by telephone industry marketers to scare users away from VoIP. In the meantime, all public VoIP services have proven that Internet best-effort QoS works just fine, as long network congestion is avoided. Internet-based voice can actually be much better than the 3.1 kHz voice over the PSTN. As for reliability, all recent major man-made and natural disasters have proven the Internet and VoIP to be more resilient than the existing wireline and wireless telephone networks.

Chapter 18, "Quality of Service for Real-Time Internet Communications," is aimed at a balanced approach for QoS, and Chapter 16, "Emergency and Preemption Communication Services," discusses the Emergency Services based on SIP.

The security threats on the Internet have provided well-justified concerns about the security of VoIP, and even more, the security of IM. As a result, a new industry niche, that of VoIP and IM security, has sprung up and, as usual, marketers are first drumming up the vulnerabilities of Internet communications to prepare the sell for all kinds of security products. Though no significant security breaks have been reported so far for Internet communications, security for VoIP and IM is still work in progress. Chapter 9, "SIP Security," deals with SIP security.

References

[1] A list of VoIP companies is provided at `www.myvoipprovider.com`.

CHAPTER 1

Introduction

The telecommunications, television, and information technology (IT) network industries are all transformed by the Internet. The transformation is driven by the need for growth based on new services, more complete global coverage, and consolidation. In this chapter, we will explore some of the problems and solutions for end users and every type of business because of the profound disruptions caused by the Internet.

Problem: Too Many Public Networks

Before the emergence of the Internet, users and service providers were generally accustomed to thinking in terms of four distinct network types: Networks for IT (data), networks for voice, mobile networks, and networks for television. Each of these dedicated network types could, in turn, be divided into many incompatible regional and even country-specific flavors with different protocol variants.

Thus, we find many types of telephony numbering plans, signaling, and audio encodings; several TV standards; and various types and flavors of what the telecom industry calls *data networks*—all of them incompatible and impossible to integrate into one single global network.

The mobile telephone networks have converged on a smaller number of standards in the second generation (2G) networks and in the emerging third generation (3G) mobile networks. It may turn out, however, that with the proliferation of new radio technologies for the so-called 4th generation (4G), such as Wi-Fi and WiMAX, all modern mobile networks will become just a wireless access mechanism to the Internet, where all public communications, entertainment, and applications will reside anyhow.

Data networks that originated in the telecom industry came in many forms, such as digital private lines, X.25, Integrated Services Digital Network (ISDN), Switched Multimegabit Data Service (SMDS), Frame Relay, and Asynchronous Transfer Mode (ATM) networks. These so-called data networks were mostly inspired by circuit-switched telephony concepts. Their names are meant to suggest that they were not designed primarily to carry voice.

Voice networks are still used for data and fax because of their general availability, though less and less so. However, these networks have come to the end of their evolution, since they are fundamentally optimized for voice only. TV networks were designed and optimized for the distribution of entertainment video streams.

Needless to say, all network types (data, voice, TV, and mobile) have specific end-user devices that cannot be ported to other service providers or network types, and most often cannot be globally deployed.

The impact of the Internet has made the wired and wireless phone companies and the TV cable companies look for new business models that can take advantage of Internet technologies and protocols, among them the Session Initiation Protocol (SIP) for real-time communications, such as Voice over IP (VoIP), instant messaging (IM), video, conferencing/collaboration, and others. Examples of the various categories and their business models are illustrated in Table 1.1. We assume that most readers are familiar with the acronyms used in the table, and we also explain these acronyms and terms in the book. They can also be found in the index.

Table 1.1 Internet Communications in 2005 with Examples from North America

CATEGORY	WHO	PROTOCOLS	STRENGTHS	WEAKNESSES
Open IM services with VoIP voice (competing islands)	Pulver FWD, Gizmo/ SIPphone, Damaka, Ineen	Standard SIP	Internet Presence Video User gets SIP URI On Net is free	Limited financing

CATEGORY	WHO	PROTOCOLS	STRENGTHS	WEAKNESSES
Closed IM islands with VoIP	Yahoo, MSN, Google, AOL, Skype (the most innovative)	SIP or other	Internet Presence Video On Net is free PSTN gateways	Nonstandard Walled gardens
PSTN over IP	Most "VoIP" companies	SIP	Internet Anywhere Video (Packet8) On net is free	Low-cost PSTN No new services Compete on price Costly infrastructure
Telephony over cable	TV cable companies	Everything from PSTN to MGCP to SIP with "P-" extensions	Broadband Internet Access to 80%+ households	Large investments in PSTN and older VoIP flavors
Wireless walled gardens	3G mobile operators	SIP for IMS with "P-" extensions	Strong financing	Central control inhibits innovation IP network cost
Wireline emulation of IMS: TISPAN	Wireline phone companies "NGN"	SIP with "P-" extensions		Duplicate IMS & NGN services

The proliferation of isolated communication islands as shown in Table 1.1 makes them less useful as their number keeps increasing (think of many more communication islands all over the world). Building communication islands (also called "walled gardens") is in conflict with Metcalfe's law that the value of the network increases by the square of the number of connected endpoints. Last, but not least, in case of an emergency, having many networks that cannot communicate directly is not very helpful.

Closed networks are an impediment for innovation, since innovators must work (technology and legal agreements) with every closed network separately to bring a new service or product to market. By contrast, the Internet extends the reach for new applications and services instantly to the whole world.

Another observation from Table 1.1 is that the strongest financing available is at present for closed networks (walled gardens), the ones that are most limited in reach and usefulness. This raises business issues and regulatory questions (what are the public interest obligations, if any?) that are beyond the scope of this book.

Incompatible Enterprise Communications

Enterprise communication systems are often an even greater mix of incompatible and disjoint systems and devices:

- Proprietary PBX and their phones. Phones from one PBX cannot be used by another.
- Instant messaging is a separate system from the PBX.
- Various IM systems don't talk to each other.
- Voice conferencing and web-based collaboration use yet other systems.

Maintaining various incompatible and nonintegrated proprietary enterprise systems is quite costly and reduces the overall productivity of the workforce.

Network Consolidation: The Internet

The Internet has benefited from a number of different fundamentals compared to legacy networks, such as the tremendous progress of computing technology and the open standard Internet protocols that define it. This progress can be attributed to the expertise of the research, academic, and engineering communities whose dedication to excellence and open collaboration on a global basis have surpassed the usual commercial pressure for time-to-market and competitive secrecy.

The result is an Internet that uses consistent protocols on a global basis, and is equally well suited to carry data, transactions, and real-time communications, such as instant messaging (IM), voice, video, and conferencing/collaboration. Actually, the Internet is the "dumb network," designed for any application, even those not yet invented. This is in stark contrast to the isolated "walled gardens" with central control of all services illustrated in Table 1.1.

Voice over IP

Although the Internet has quickly established itself as the preeminent network for data, commercial transactions, and audio-video distribution, the use of voice over the Internet has been slower to develop. This has less to do with the capability of the Internet to carry voice with equal or higher quality than the telephone network but rather with the complex nature of signaling in voice services, as you will see in Chapter 6, "SIP Overview."

There are various approaches for voice services over the Internet, based on different signaling and control design. Some examples include the following:

- Use *signaling* concepts from the telephone industry—H.323, MGCP, MEGACO/H.248.
- Use *control* concepts from the telephone industry—central control and softswitches.
- Use the Internet-centric *protocol*—Session Initiation Protocol (SIP), the topic of this book.

The movement from such concepts as telephony call models to discovery/rendezvous and session setup between any processes on any platform anywhere on the Internet is opening up completely new types of communication services.

The use of SIP for establishing voice, video, and data sessions places telephony as just another application on the Internet, using similar addressing, data types, software, protocols, and security as found, for example, on the World Wide Web or e-mail.

Separate networks for voice are no longer necessary, and this is of great consequence for all wired and wireless telephone companies.

Complete integration of voice with all other Internet services and applications probably provides the greatest opportunity for innovation. The open and distributed nature of this service and the "dumb" network model will empower many innovators, similar to what has happened with other industries on the Internet and the resulting online economy.

Most IM systems on the Internet already have voice and telephony capability as well, though if it is proprietary, they cannot intercommunicate without IM gateways, although IM gateways inevitably cannot translate all the features from one system to another. IM gateways are also transitory in nature,

since any changes to a proprietary IM protocol may render the gateway close to useless. By contrast, SIP-based communications offer a global standards-based approach for interoperability for presence, IM, voice, and video, as we will show in the following chapters.

Presence—The Dial Tone for the Twenty-First Century?

Unsuccessful telephone calls are a serious drag on productivity and a source of frustration, since both parties waste time and talk to voicemail instead to each other. Also, the timing of the phone call may not be appropriate or not reach the called party in a suitable location. The advent of presence, so well-known from IM systems, can provide much more rich information before trying to make a call in the first place, compared to just hearing the dial tone. Another convenience of SIP and presence is that many contact addresses may reside beneath a buddy icon, so the caller need not to know or worry about picking the right phone number or URI. Presence may, therefore, replace the dial tone used in telephony for well over 100 years.

The Value Proposition of SIP

SIP is not just another protocol. SIP redefines communications and is impacting the telecom industry to a similar or greater degree than other industries. This has been recognized by all telecom service providers and their vendors for wired and wireless services, as well as by all IT vendors. Chapter 2 will provide an overview of how the Internet and SIP are redefining communications.

SIP Is Not a Miracle Protocol

As discussed in Chapter 2, "Internet Communications Enabled by SIP," SIP is not a miracle protocol and is not designed to do more than discover remote users and establish interactive communication sessions. SIP is not meant to ensure quality of service (QoS) all by itself or to transfer large amounts of data. It is not applicable for conference floor control. Neither is it meant to replace all known telephony features, many of which are caused by the limitations of circuit-switched voice or to the regulation of voice services. And such a list can go on.

Various other Internet protocols are better suited for other functions. As for legacy telephony, not all telephone network features lend themselves to replication on the Internet.

The Short History of SIP [1]

By 1996, the Internet Engineering Task Force (IETF) had already developed the basics for multimedia on the Internet (see Chapter 14, "SIP Conferencing") in the Multi-Party, Multimedia Working Group. Two proposals, the Simple Conference Invitation Protocol (SCIP) by Henning Schulzrinne and the Session Initiation Protocol (SIP) by Mark Handley, were announced and later merged to form Session Initiation Protocol. The new protocol also preserved the HTTP orientation from the initial SCIP proposal that later proved to be crucial to the merging of IP communications on the Internet.

Schulzrinne focused on the continuing development of SIP with the objective of "re-engineering the telephone system from ground up," an "opportunity that appears only once in 100 years," as we heard him argue at a time when few believed this was practical.

SIP was initially approved as RFC [2] number 2543 in the IETF in March 1999. Because of the tremendous interest and the increasing number of contributions to SIP, a separate SIP Working Group (WG) was formed in September 1999. The SIP for Instant Messaging and Presence Leveraging (SIMPLE) was formed in March 2001, followed by SIPPING for applications and their extensions in 2002. The specific needs of SIP developers and service providers have led to an increasing number of new working groups. This very large body of work attests both to the creativity of the Internet communications engineering community, and also to the vigor of the newly created industry.

We will shorten the narrative on the history of SIP by listing the related working groups (WG) in chronological order in Table 1.2. We have listed for simplicity the year of the first RFC published by the WG, though the WG was sometimes formed one to two years earlier. Years denote a new WG that has not yet produced any RFC.

Table 1.2 History of SIP-Related Working Groups

NAME	FIRST RFC	CHARTER
avt	1996	Real-time transmission of audio and video over UDP/IP: RTP
mmusic	1998	Internet conferencing and multimedia communications: SIP, SDP, RTSP
iptel	2000	Routing and call processing for IP telephony: TRIP, CPL, tel URI
sip	2000	Development of the SIP protocol: SIP methods, messages, events, URI

(continued)

Table 1-2 *(continued)*

NAME	FIRST RFC	CHARTER
enum	2000	DNS-based use of ITU-T E.164 telephone numbers
sipping	2002	Applications and extensions to SIP
simple	2004	Use of SIP for Instant Messaging (IM) and Presence
xcon	2005	Centralized conferences
behave	(2005)	Behavior for Network Address Translation (NAT) for use with SIP, RTP
ecrit	(2005)	Emergency communications (such as 911, 112)
p2psip	(2005)	Peer-to-peer SIP (not yet a formal WG)

The growth of SIP-related standards in the IETF is illustrated and discussed in Chapter 21, "Conclusions and Future Directions."

References in This Book

Because of the multiple developments on the Internet, SIP is being used in ever-more services, user software, and various user devices (such as in SIP phones, PCs, laptops, PDAs, and mobile phones). This is, in effect, a new industry and its participants keep making new contributions to the core SIP standards, mainly in the area of new services and new applications. This book reflects SIP developments up to and including the 64th IETF in November 2005.

We have included, by necessity, many Internet drafts that are designated *work in progress*, since they are the only reference source for this particular information. Some of these drafts may become standards by the time you are ready to use them; some may be a work in progress and have a higher version number than quoted as of this writing; and still others may be found only in an archive for *expired drafts*.

The SIP WG drafts that are work in progress can be found online at the IETF web site:

```
http://ietf.org/html.charters/sip-charter.html
```

Additional individual submissions and Internet drafts from other working groups can be found at the following site:

```
http://ietf.org/ID.html
```

SIP-related drafts that have *expired* (older than six months) can be found on several archives. As of this writing, following are some of the sites:

```
www.cs.columbia.edu/sip/drafts
www.softarmor.com/sipwg
```

Readers may also perform a web search, such as Google, for any IETF SIP-related topic or for any Internet draft or RFC.

Several books have been published on Internet multimedia, Voice over IP, and SIP, some of which are listed here. [3], [4], [5] They focus mainly on how SIP works. This book is less about explaining how SIP works, but rather what it does and the new communications and services it enables.

We have reproduced some of the exciting services and features discussed in the IETF SIP WG and its main offsprings, the SIPPING and SIMPLE Working Groups. Also included in our discussion are some drafts from Bird of Feather (BOF) sessions that have not even made it to an accepted WG charter, such as the peer-to-peer (P2P) SIP group. [6]

Many of these expired proposals may not develop into IETF standards for various reasons, but represent good work, often backed up by running code. The references to such expired Internet drafts are intended to make you aware of these ideas that may otherwise remain buried in an archive. Such references are clearly marked as expired, so as to distinguish them from accepted work in progress items of IETF WGs that are on the path toward acceptance as standards.

SIP Open Source Code and SIP Products

There is an ever-increasing amount of open source code for SIP, and it is increasing in quality. Most or many commercial SIP products are actually based on open source SIP code. An authoritative list of SIP open source code is available from the SIP Forum:

```
www.sipforum.org
```

The SIP Forum is also an excellent source for finding commercial SIP software and products for the enterprise, for consumer products, for service providers, tools, and others.

Excellent lists of SIP products are also maintained on the web sites of `Pulver.com` and `Ubiquity.com`:

```
www.pulver.com/products/sip
www.sipcenter.com
```

References for Telephony

We assume throughout this book some understanding of telephone services and of telecommunication protocols. There is a vast literature pool available on telephony and telecommunications. We refer you to *Newton's Telecommunications Dictionary* [7] to brush up on various terms that will be used in the following chapters.

Summary

This chapter has discussed some of the problems and solutions to the communications industry by the Internet, and also a brief history of the SIP protocol.

During the migration from circuit-switched telephony to IP-based communications, there are too many isolated wired and wireless communication networks, even though most (but not all) are converging on SIP. SIP has undergone a 10-year development as a standard and in implementations in the marketplace.

By adopting the Internet as *The Network* with wired and wireless access, and SIP as the standard protocol, rich global communications are taking shape.

The old dial-tone in telephony may well be replaced by *presence* information, and rich multimedia will replace the narrowband voice communications used in circuit-switched telephony.

References

[1] The authors would like to thank Professor Dr. Jörg Ott, co-chair of the SIP WG and early contributor to the MMUSIC WG for helping with data on SIP history.

[2] RFC stands for Request for Comments and many of them are Internet standards.

[3] *SIP: Understanding the Session Initiation Protocol*, 2nd Edition, by Alan B. Johnston, Artech House, 2003.

[4] *SIP Demystified* by Gonzalo Camarillo, McGraw-Hill, 2001.

[5] *SIP Beyond VoIP* by Henry Sinnreich, Alan B. Johnston, and Robert J. Sparks, VON Publishing, 2005. `www.vonmag.com/books`.

[6] See the web site for P2P SIP at `www.p2psip.org`.

[7] *Newton's Telecommunications Dictionary*, 17th edition by Harry Newton, CMP Books, March 2001.

CHAPTER

2

Internet Communications Enabled by SIP

This chapter provides a short overview of the topics that are discussed in more detail in Chapters 4–20.

The Internet challenges and transforms the more than one-trillion-dollar-per-year business of telecommunications. A renaissance in communications is taking place on the Internet. At its source are new communication protocols that would be impractical on the centralized control systems of ITU-T type networks used in telecommunications. Internet communications can benefit from the IP soft state and connectionless nature of the Net, and at the application layer of the IP protocol stack from its associated addressing and data representations. Users and Internet service providers (ISPs) are reaping the benefit from standards that allow interoperability with all connected parties on a global scale. The end-to-end (e2e) nature of the Internet avoids the friction of having intermediaries between the communicating parties, and also avoids the breaking of applications and security by intermediaries in the network.

While it is not possible to forecast technology and services, it is already apparent that the Internet and web technology have created an unprecedented toolkit for new applications. However, these new applications are hard to predict, just as presence and instant messaging were not predicted in the telecom world. What can be shown, however, are some of the capabilities of the technology that are presently well understood in already established services. New Internet communication services may create new revenue opportunities for Internet service providers and their suppliers of applications.

This chapter refers to many legacy telephony services.

Readers may consult *Newton's Telecommunications Dictionary* [1] for definitions of the telephony and telecom services mentioned here. Chapter 11, "SIP Telephony," also discusses in detail many enhanced telephone services.

The overview of SIP services provided here reflects current thinking in the community of SIP service and technology developers. Most (but not all) of them have been actually tested and implemented. Some proposed Internet drafts on SIP will make it to the level of IETF standards; some will not. It also is likely that new technologies and services will emerge that have not been made public or envisaged as of this writing.

Internet Multimedia Protocols

Networks are defined by their protocols. The global telephone network uses its own signaling and communication protocols, as do other telecom networks such as the Public Switched Telephone Network (PSTN), X.25, Integrated Services Digital Network (ISDN), Switched Multimegabit Data Services (SMDS), frame relay, Asynchronous Transfer Mode (ATM), mobile circuit-switched networks, and the (seemingly always) proposed ITU-T Next Generation public Networks (NGN). Besides legacy network protocols, there are also application-level protocols, such as those used between fax machines.

Though started with much smaller resources than the previously dominant telecom and non-IP data networks (SNA, DECnet, Novell), the Internet's success is solely due to its well-designed architecture and protocols. The architectural principles of the Internet (covered in Chapter 3, "Architectural Principles of the Internet") have made it the most effective network for any type of application, including real-time communications.

Internet telephony and the wider family of Internet communications are defined by several key application level protocols. The list of Internet protocols used for interactive communications is shown in Table 2.1.

Table 2.1 Key Standard Internet Multimedia Protocols

FUNCTION OF THE PROTOCOL	STANDARD [2]	DESCRIPTION
Real Time Transfer (RTP)	RFC 3550	End-to-end real-time transport for audio, video, and data, without quality of service (QoS).
Audio/Video Profiles RTP/AVP	RFC 3551	Defines protocol fields for audio and video and lists some basic standard encodings.

Table 2.1 *(continued)*

FUNCTION OF THE PROTOCOL	STANDARD [2]	DESCRIPTION
Session Description (SDP)	RFC 2327	The description required for initiating multimedia sessions. Has many related RFCs.
Session Initiation (SIP)	RFC 3261	Application layer protocol for creating, modifying, and terminating sessions.
Instant Messaging (IM)	RFC 3428	SIP extension for instant messaging.
Presence	RFC 3856	Presence event package for SIP.
Real Time Streaming	RFC 2326	Control of the delivery for real-time data such as audio and video.

The nature of interactive communications and the type of service are determined by the signaling used for establishing the communication, hence the name *value of signaling.*

The Value of Signaling

Signaling in telephone systems is the key mechanism by which telephone calls are set up and terminated. For example, signaling from a desktop business phone tells the PBX to forward the call to another phone. In the public telephone network, signaling instructs the switching systems to forward an 800 call to a specific call center where an agent will answer the call.

An example of the value of signaling is the comparison between a telephone chat between two residences and an 800-number call to a customer-support center. Such calls are also priced differently. In the end, both phone calls sound the same, except that signaling has enabled the adding of commercial value to the 800 number call for a possible business transaction.

Signaling defines the desired service for the user, such as point-to-point calls, multipoint conferencing, Centrex services, text, voice, and video, and others (see Table 2.2).

Table 2.2 Value-Added Telephony Services Based on Signaling

Intelligent Network (IN) services
PBX features

(continued)

Table 2.2 *(continued)*

PSTN Class services
Mobile cellular roaming
Desktop call manager
Replacement for Computer Telephony Integration (CTI)
Group calling
Click-to-connect
Internet call waiting
"Dialing" an e-mail address or URL

The signaling protocol for Internet multimedia real-time communications is the Session Initiation Protocol (SIP).

Protocols for Media Description, Media Transport, and other Multimedia Delivery

The Internet has established itself as the most adequate platform for multimedia communications and for the delivery of streaming multimedia content. Though signaling (the topic of this book) is most critical, it is not the only protocol required for multimedia communications. Internet standards feature a pretty clean breakdown of functionality, and each function is performed by one single protocol. Duplication of functions on two or more protocols is carefully avoided so that different implementations can have only one single way of being standard-compliant. Table 2.1 shows the key standard Internet multimedia protocols. The complete list is, however, bigger that what is shown in the table.

The wise design decision of the Internet architects to make it equally suitable for any application has, however, led to the surprise of multimedia file sharing and real-time communications that are not based on standards, mostly using peer-to-peer protocols (P2P), some of which are described in Chapter 20. A useful observation is that P2P traffic is dominant on the Internet, according to several sources. An interesting source on Internet usage statistics that include P2P traffic are given in [3] and [4]. P2P applications (such as replacements of centralized IP PBXs) are also being deployed in the enterprise.

Addressing

IP communications use SIP Uniform Resource Identifiers (URIs) for addressing similar to e-mail, where the form of the URI resembles an e-mail address in `mailto:`, such as `user@domain`. A more detailed discussion of URIs is provided in Chapter 4, "DNS and ENUM."

SIP URIs can have various forms and include telephone numbers. For example,

```
sip:henry@pulver.com
```

Example: "Dialing" an address refers to the PC of Henry in the domain `pulver.com`. (See Chapter 4, "DNS and ENUM," for more about domain names.)

Here are some examples of using URIs for SIP:

```
sip:+1-972-555-1234@pulver.com; user=phone
```

is a phone number that can be reached via a gateway (note that visual separators within a telephone number, such as dashes and dots, are optional and are ignored by the protocol);

```
sip:123-4567@pulver.com; user=phone; phone-context=VNET
```

is a phone number in the internal network "`VNET`" of `pulver.com` and

```
sip:guest314@pulver.com
```

is the address of the laptop of a guest plugged into the LAN of a conference room in the `pulver.com` domain.

The support for both telephony and web-type addressing enables Internet communications to bridge in a seamless way the telephone network and the Internet. Users on either network can reach any point either on the PSTN or on the Internet without giving up existing devices or the accustomed conveniences of either. For example, a user of the telephone network can make a call to a device on the Internet or to any other device on any other network (mobile voice, paging, data networks) just by dialing a number, as will be explained further. The ENUM technology allows users to have a single URl or phone number, if they so prefer, on their business card for contact information.

SIP in a Nutshell

Table 2.3 explains the core SIP functions in a nutshell and Figure 2.1 shows the flow diagram for the message exchange between to SIP endpoints.

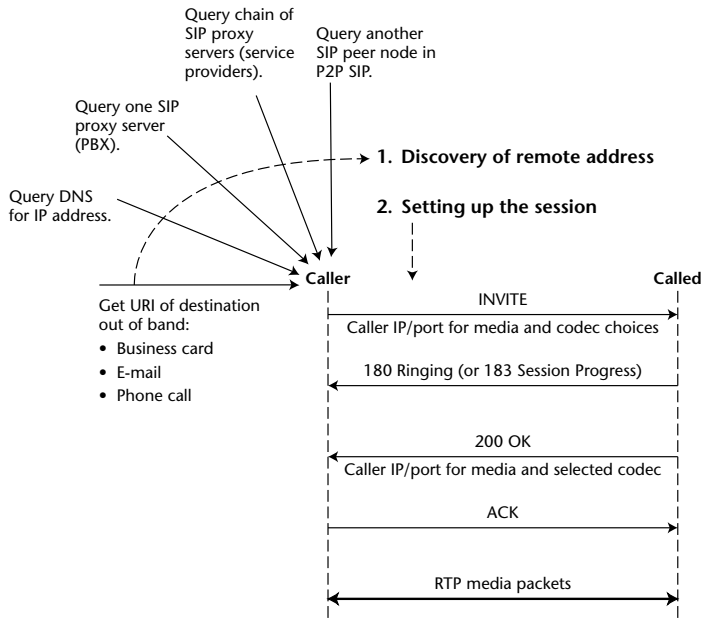


Figure 2.1 Basic SIP

Table 2.3 SIP Operations in a Nutshell

FUNCTION	DESCRIPTION	COMMENTS
Discovery of destination address	Get SIP URI out of band: address book, business card, e-mail, phone call. The URI requires several DNS lookups.	Basic SIP requires no SIP servers, as shown in Figure 2.1.
	Query the DNS for the IP address. This allows the IP address to change.	RFC 3261 specifies SIP servers are optional.
	Single domain: Query a central SIP proxy server to determine the destination IP address. This triangle model was first used in RFC 2543.	This is the model for the IP PBX or most public VoIP service providers (VISP).
	Inter-domain: The query passes to an outgoing SIP proxy and a remote incoming SIP proxy. The trapezoid model is based on RFC 3261.	The trapezoid model will support session setup between different Internet domains.

Table 2.3 *(continued)*

FUNCTION	DESCRIPTION	COMMENTS
	Query using a peer-to-peer procedure such as Chord to determine the IP address of the destination peer.	This is the model for basic P2P SIP and also requires no central SIP proxy servers.
Session setup	The `INVITE` message informs the called party of the IP address and ports of the caller and offers a choice of audio/video codecs.	
	A provisional response message, such as 100 Trying or 180 Ringing, informs the caller of the progress.	
	The message 200 OK confirms the called party is ready to receive media at its specified IP address and port(s) and the selected codec from the choice.	
	The `ACK` message from the caller acknowledges is readiness as well.	
	RTP media packets containing audio, video or IM will now flow between the SIP endpoints.	

SIP Capabilities

SIP-enabled IP devices can call each other directly, if they know each other's URL. Thus, an IP phone call can be placed directly between two or more SIP phones or PCs.

Small conferences can be held by several users connecting to one device acting as the conference bridge, where one of the SIP phones can act as both conference participant and conference bridge.

Besides SIP devices such as phones, PCs, IP telephony gateways, and mobile devices, service providers also deploy SIP servers for a variety of additional services.

Figure 2.2 illustrates how SIP servers perform a routing service that puts the caller in contact with the called party in a step-by-step fashion, taking into account the desired service and user preferences. We will show in the following sections that the SIP service model provides users with all services known from the circuit-switched telephone network, as well as new services that result from taking advantage of the Internet.

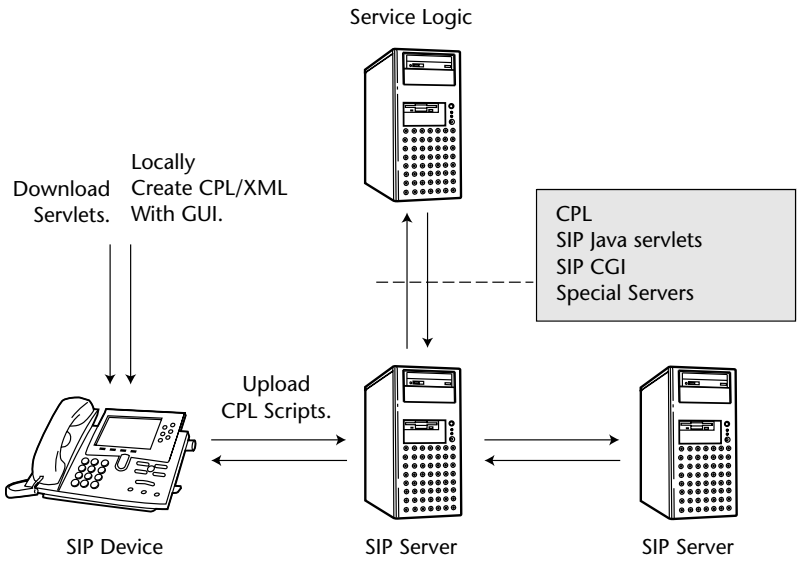


Figure 2.2 Call routing performed by a SIP proxy server

Overview of Services Provided by SIP Servers

Multimedia conferencing on the Internet was well developed by the research and academic community by 1997. This has been reflected in the explosion of commercial ventures for Internet multimedia during the past decade. Work started at the same time to extend the Internet multimedia architecture for use in telephony. Because of the enormous complexity and richness of services on the PSTN, this work has taken much longer to develop, and only at the end of 2000 had it reached a critical mass where true reengineering of the telephone system for the Internet was well understood.

In the history of science and technology, many new technologies have found applications that were not envisaged by their inventors. With this limitation in mind, the following sections will provide an overview of services that are supported by SIP servers, such as those used by public VoIP service providers and in enterprise PBX networks.

The prevalent business model of VoIP service providers in early 2006, however, is to not support any features that require going outside the walled garden. The assumption of practically all VoIP service providers is that all services are provided in-house. This may change, however, as Internet-wide VoIP will mature.

Recent work has shown that all or most services performed by SIP proxy servers in the network can also be performed by server-less P2P SIP.

Peer-to-Peer SIP (P2PSIP)

P2P SIP is a recent development that follows the general P2P trend on the Internet.

The end-to-end nature of the Internet (the dumb network) has led to an amazing growth of various peer-to-peer applications and to the predominance of P2P traffic on the Internet, estimated in 2004 and 2005 to be between 60 percent and more than 80 percent of all Internet traffic.

In private networks, commercial P2P IP PBX functions have been implemented that allow advanced enterprise desktop phones to work without any central IP PBX server. See, for example, the references [5] and [6]. Small SIP P2P networks (such as for communities of interest) may not require any servers or other service infrastructure in the network. Very large P2P SIP networks may use a hierarchical structure for the self-organizing P2P cloud where the "supernodes" are the equivalent of SIP servers.

Since P2P VoIP networks do not require any VoIP infrastructure, the ever-increasing costs to buy and to operate complex VoIP networks are eliminated. There are also other general benefits to P2P that will be discussed in Chapter 20. P2P is, thus, a considerable disruption of the VoIP industry.

Caller Preferences

Caller preference allows a user to specify how a call should be handled in the network (for example, if the call should be queued or forked to several destinations), and what features should be supported, such as media types, language, and mobility. Additional preferences are also supported (for example, prevent people from the office disturbing someone at their residential number).

Called party preferences include accepting or rejecting calls (from unlisted numbers), depending on time of day, location of the called party, origin of the call, and other criteria.

SIP caller preferences and called party capabilities reveal unprecedented service capabilities under control of either the caller with the consent of the called party, with or without the involvement by the service provider. Services can be customized with ease on a dynamic basis, depending on a large set of criteria such as presence, time of day, caller or called party identity, call urgency, personal caller preferences, network status, and the content of external databases [2]. User preferences are presented in more detail in Chapter 8, "User Preferences."

Mobility in the Wider Concept

This section examines mobility in the wider context of the Internet, the PSTN, mobile networks, and the application-layer mobility based on SIP.

SIP can support application-level mobility across different networks where the network and the device cannot be changed. Application-level mobility that SIP can support is described in Chapter 15, "SIP Application Level Mobility."

Global Telephone Number Portability

Local number portability allows telephone subscribers to keep their phone number when changing service providers, but only in the same local calling area. Telephony number portability on a national scale poses an implementation challenge on the PSTN, and global telephone number portability seems just not to be possible with the various national telephone networks of today. Global number portability is, however, a trivial application for SIP services if users have a domain name and a SIP URI in their own domain, such as the following:

```
sip:alice@mydomain.net
```

A SIP URI is globally valid by contrast to telephone numbers that have a local significance, except for the E.164 number format, though even 1-800-. . . numbers have a local significance only.

Public VoIP providers with gateways in multiple cities and multiple countries are offering phone numbers from everywhere their SIP to PSTN gateway service can reach. For example, a resident in Karachi, Pakistan can have a phone number in Dallas, Texas, USA (this is a real service known to the authors). The use of DNS is described in Chapter 4, "DNS and ENUM."

SIP Application-Level Mobility

You can make the distinction between the following:

- *Terminal mobility*—Terminal moves between subnets.
- *Personal mobility*—Different terminals, same address.
- *Service mobility*—Keep same services while mobile.

SIP has been chosen for call control for the third generation (3G) wireless networks by the Third-Generation [wireless mobile] Partnership Program (3GPP and 3GPP2) initiatives.

Mobility for IP has been defined in the IETF by RFC 2002 with the basic concept that a mobile host maintains its IP address while changing the point of attachment to the IP network. Mobile IP is, therefore, valid for any application,

be it file transfer, web browsing, or communications. For example, mobile IP is a useful feature when moving with a wireless-connected laptop to another office in the same building or campus.

Although there is no agreement yet in the IETF about application-level mobility, many SIP developers feel that terminal mobility, personal mobility, and service mobility (where users can change devices, networks, and the IP address used for communications) are valid extensions of the more limited notion of mobile IP. SIP mobility will allow users to communicate while on the move, with a short handover, but an uninterrupted file transfer or web browsing would not be possible with changing IP addresses.

Mobile IP and SIP mobility are, therefore, complementary capabilities with different areas of application.

We believe SIP mobility is a wide-open field where many interesting developments are possible. SIP mobility is presented in Chapter 15, "SIP Application-Level Mobility."

Context-Aware Communications: Presence and IM

Presence and instant messaging (IM) are usually perceived as parts of a single service, but, in fact, represent different communications service components. We will discuss them here separately.

It is important to notice that presence and instant messaging based on SIP use similar message flows to SIP and the same "infrastructure" that is used for SIP-based voice and video communications: Endpoints, servers in the network, message exchanges, software, and data. Presence and IM come virtually at no (or little) extra infrastructure cost for SIP service providers, but enable very innovative new services, such as the following:

- Push-to-talk for mobile networks
- Integration of applications with communications

These are the reasons not to deploy disparate voice systems (such as a PBX) and separate IM systems.

SIP Presence

The presence information conveys the ability and willingness of a user to communicate [7]. Initially, presence has been limited to "online" and "offline" indicators, but later work has added emoticons about the state and mood of the remote party, as well as some other useful information, such as "is typing" [8]. Table 2.4 shows some examples of rich presence information.

Table 2.4 Examples of Rich Presence Information

On the phone
Away
Appointment
Holiday
Meal
Meeting
Driving
In transit
Travel
Vacation
Busy
Permanent absence

Rich presence information as described in Chapter 13 is not only very valuable for communications between humans but can also support the understanding of presence by machines.

Presence requires the messages `SUBSCRIBE` and `NOTIFY` between the parties that display the presence (the "presentity") and the party that is interested in this information (the "watcher") SIP presence can also be explained in a nutshell with the help of the message flow shown in Figure 2.1 by noting the analogies between the SIP and presence messages shown in Table 2.5.

The `SUBSCRIBE` and `NOTIFY` messages can be extended from human users to any type of applications, and this can support the integration of communications and applications, as we will be described in Chapter 13, "Presence and Instant Messaging."

Table 2.5 SIP Messages for Presence and IM vs. Voice and Video

PRESENCE AND IM	VOICE/VIDEO
`SUBSCRIBE`	`INVITE`
`200 OK`	`200 OK`
`NOTIFY`	`ACK`
`MESSAGE`	`RTP Media`

Instant Messaging

Instant messaging is the exchange of text messages between users in real time. The text messages need not to be short and, as we will show, actually quite large files can be transmitted using suitably designed IM applications. Instant messaging using SIP is based on the `MESSAGE` method [9]. Table 2.5 shows the analogies with SIP voice and video signaling and is also illustrated in Figure 2.1.

Presence and instant communication clients can have a rich graphic user interface (GUI) for PC displays and also for display phones, palm computers, mobile phones, and other devices.

It is probable that future communication interfaces will resemble an instant messaging GUI rather than the present telephone keypad.

The Integration of Communications with Applications

The SIP Event Architecture using messages such as `SUBSCRIBE` and `NOTIFY` is the basis for the integration of communications and applications. A money transfer can, for example, trigger a communication between the interested parties, or a call with an important customer can trigger, for example, a notifiction to other employees to join the call or to prepare relevant data for the customer.

E-Commerce: Customer Relations Management

Traditional voice call centers for customer support are migrating to web-based support centers where the focus is shifting from pure voice (800 numbers) to e-mail support, text chat, and voice with click-to-connect service. Besides the shift in functionality, the voice call center part also can be rearchitected from the ground up. The following aspects are changed from conventional call centers:

- The user experience includes web-based choices, IM, and voice, instead of navigating irritating interactive voice response systems (IVR).
- The call center PBX and automatic call distribution systems are replaced by SIP-based voice and presence-based internal communications.
- Customer calls for support can be routed more effectively and at low cost compared to 800-based PSTN routing.
- Call routing inside the contact center can be accomplished by having the SIP proxy routing database use the following data and criteria:
  - Caller ID to find the attached data for the customer
  - The service or product that is supported
  - The agent with most appropriate skill set
  - Presence of agents

- Day of week, holidays, and time of day
- Promotions
- Service priorities dictated by business decisions
- Measuring the agents (response time, handling time, successful calls, abandoned calls)
- Routing to legacy Time Division Multiplex (TDM) systems

We believe that Internet-enabled customer contact centers have a far greater commercial potential than just offering "PSTN over IP" services (that is VoIP that emulates the PSTN).

Traditional telephony call centers can be redesigned from ground up using the following opportunities:

- Replacing the PBX with SIP call routing
- Replacing the automatic call distributor (ACD) with SIP presence
- Replacing the interactive voice response (IVR) with web choices and VoiceXML
- Having the agent workstation becomes a SIP-enabled user agent, plus applications
- Using web-style search in the customer database for relevant information

Conferencing and Collaboration

SIP has its roots in academic collaboration and conferencing over the Internet during the mid-1990s. Since then, most conferencing and collaboration vendors have been using SIP-based technology for the following activities:

- Displaying who is attending and which participant is speaking
- Text chatting, including private sidebar conversations
- Multimedia, voice and video conferencing
- Web page sharing/web page pushing
- Desktop displays and whiteboard sharing
- File transfer between participants

As of this writing, though, some public Internet conferencing services are either not SIP standards–compliant or do not offer the full complement of services mentioned here.

Advanced SIP PC software can support the hosting of rich multimedia conferencing on a user's PC with up to six other participants, depending on the speed of the PC and the speed of Internet service as implemented commercially [10]. This is illustrated in Figure 2.3.

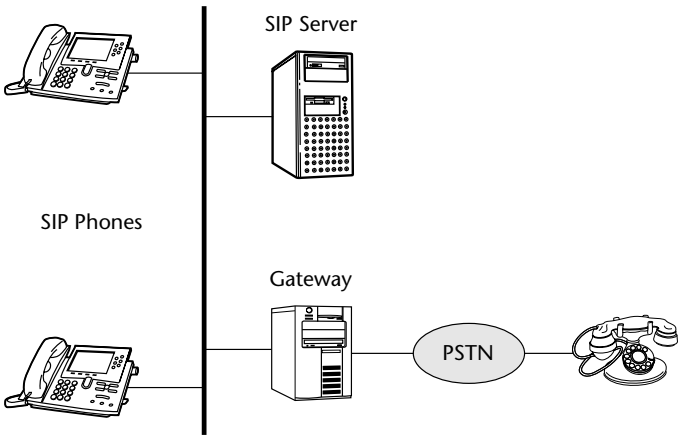


Figure 2.3 Conference call hosted on a SIP endpoint, such as a SIP phone

Telephony Call Control Services

Telephony call control has been developed to an advanced degree of maturity and complete standards documents, as well as many commercial implementations, exist for the emulation of Intelligent Network (IN) services, as well as of PBX-and Centrex-style telephony features. The SIP standards for telephony include the following:

- Basic SIP call flows [11]
- SIP-PSTN call flows [12]
- SIP service examples for PBX/Centrex style functions [13]

Intelligent Network Services Using SIP: ITU Services CS-1 and CS-2

Though the PSTN business has declined more rapidly than forecast in 2001 when the first edition of this book was published, it is still of interest to show the capabilities of SIP for the control of ITU-T style legacy TDM switched networks. The extensive capabilities provided by the PSTN and ISDN Intelligent Network services can be supported by SIP. Authors from Columbia University and Lucent Bell Labs have published a detailed paper on Intelligent Network services that can be provided by SIP [1, 4]. The capabilities also include mobile telephony features.

Service Examples for PBX and Centrex-style IP Systems are given in Table 2.6.

Table 2.6 IP PBX and Centrex Service Examples Using SIP

Call Hold
Consultation Hold
Unattended Transfer
Call Forwarding, Unconditional
Call Forwarding on Busy
Call Forwarding on No Answer
Three-Way Conference
Single Line Extension
Find-Me
Incoming Call Screening
Outgoing Call Screening
Secondary Number - In
Secondary Number - Out
Do Not Disturb
Call Waiting

PBX systems have additional, application area–specific features that we do not list here for brevity, and also because most people rarely make use of them. Such features are mostly proprietary.

SIP Service Creation—Telephony-Style

The wide range of possibilities for new services is matched by the ease of service creation and deployment [14]. Simple services can even be developed by end users.

- *Call Processing Language (CPL)* [15] is mainly intended to be used by nontrusted end users to upload their services on SIP servers. XML scripts created by end users can be uploaded to SIP servers for call setup in a secure execution environment.
- *SIP Common Gateway Interface (CGI)* [16] is analogous to the Common Gateway Interface (CGI) used for web server access to databases. Complex services can be programmed under control of network administrators using SIP CGI.

- *Java platforms* have been extended for the development of SIP [17]. There is extensive literature available on this topic as well.

ENUM

ENUM was initially meant to stand for ***E.***164 ***NUM***ber translation to IP, but now correctly stands for E.164 to URI Dynamic Delegation Discovery Systems (DDDS) Application [18] and is a service that allows users to have only one single phone number on their business card. The ENUM user may have multiple PSTN, mobile and PBX phone and fax numbers, at home, at work, and in autos or boats, as well as several IP devices such as PCs, laptops, and palm computers, and SMS. ENUM can use the Domain Name System (DNS) in combination with SIP user preferences, so if someone uses the single number on a business card, the call, SMS page, voicemail, or e-mail can be directed to the device of preference of the called party.

Using a single telephone number to be reached anywhere is a valid concept at present, since many phone calls originate on circuit-switched networks using PSTN or PBX-type telephones. However, telephone numbers need not be the preferred contact address everywhere and for all times. As communications over the Internet and in 3G/4G mobile networks gain more and more user acceptance, the single contact address in the form of a URI (such as an e-mail address) may become the more practical choice.

ENUM allows callers from circuit-switched networks that are predominant at present to reach any device on either on another circuit-switched network or on the Internet. ENUM service with SIP is described in Chapter 4, "DNS and ENUM."

SIP Interworking with ITU-T Protocols

Much work has been dedicated by the Internet Engineering Task Force (IETF), the International Telecommunications Union Telecommunications Standardization Sector (ITU-T), and the European Telecommunications Standards Institute (ETSI) for interworking of SIP with other protocols, such as those shown in Table 2.7.

Table 2.7 SIP Interworking with ITU-T Protocols

ENUM: E.164 to IP address mapping using DNS
SIP-H.323 [18] Interworking

(continued)

Table 2.7 *(continued)*

Accessing ISDN and PSTN IN services from SIP networks [20]
SIP and QSIG for circuit-switched PBX interworking [20]
SIP for Telephones (SIP-T), for transport of telephony signaling across IP [22]
In addition to the preceding protocols interworking with ITU-style networks, interworking with or making use of new protocols also is being investigated. See, for example, SIP and SOAP [23].

Figure 2.4 shows an example of SIP and PSTN interworking that will be referred to in the following and SIP-PSTN gateway service. The example also shows a SIP-PBX gateway.

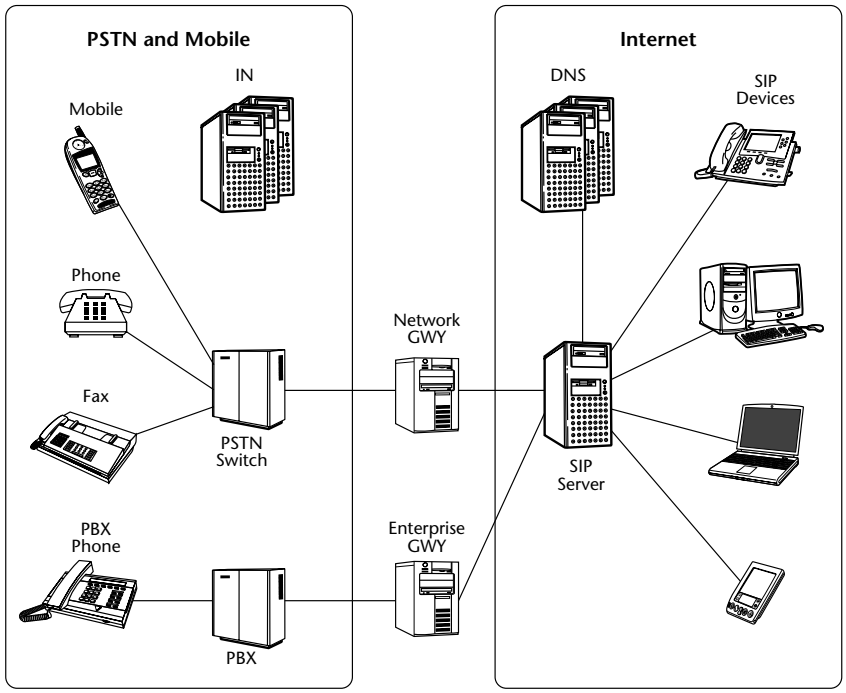


Figure 2.4 IP-PSTN gateway service and SIP-PBX gateway

Mixed Internet-PSTN Services

There are a wide range of interworking modes between the PSTN- and SIP-based IP communication networks. Early work on SIP was mainly focused on interworking with the PSTN and with PBXs, and you might suppose this was the price the SIP developers' community had to pay for making SIP acceptable to telephone companies as the signaling standard for VoIP. In retrospective, given the other innovative services enabled by SIP and the faster-than-expected diminishing importance of wireline telephony compared to mobile telephony and the Internet, we don't believe this work is now as important as it was thought to be initially.

PSTN and INTerworking (PINT)

***P**STN and **I**nternet **INT**erworking* (PINT) [24] is a service where an action from the Internet (such as a click on a web page) invokes a PSTN service, such as setting up a call between two phones (RFC 2848) or between two fax machines, or connects a fax machine to an information service that can send a fax on demand. Applications are click-to-connect, click-to-fax, click for information, and various others.

SPIRITS

***S**ervers in the **P**STN **I**nitiating **Re**quests to **I**n**T**ernet **S**ervers* (SPIRITS) [25] is the name of a family of IN services on the PSTN that can be implemented using SIP. It also applies to such services as Internet call waiting, where an event (calling a busy phone line) on the PSTN can generate an action on the Internet (call waiting pop-up panel on the PC that is using the called line for Internet access).

TRIP

The ***T**elephony **R**outing over **IP*** (*TRIP*) protocol [26] is designed to find the desired gateway to terminate a call on the PSTN. Given the increasing number of IP telephony gateways, it may not be practical to maintain huge SIP routing tables. It also may be desirable to route calls to gateways that meet certain criteria. The Telephony Routing Protocol is modeled after the IP Border Gateway Protocol (BGP) routing protocol and inherits its scalability.

Figure 2.5 shows an enterprise network connected to an Internet service provider (ISP) with SIP servers and various other SIP devices such as SIP phones and also the gateway to the PSTN.

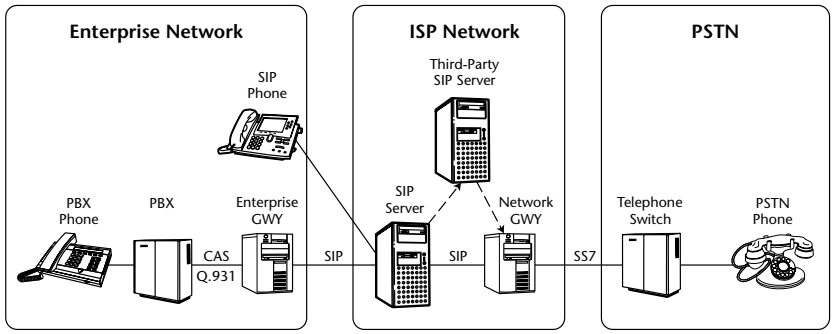


Figure 2.5 Mixed enterprise voice systems using legacy PBX, the PSTN, and VoIP.

Firewalls and network address translators (NATs) are not shown here for simplicity, but are discussed in Chapter 9, "SIP Security" and in Chapter 10, "NAT and Firewall Traversal." SIP servers placed in both public IP network domains of ISPs and in private enterprise networks can, however, perform many functions for end users, as shown in Table 2.8.

Large enterprises must often have a long and smooth transition from the installed base of PBX systems and new SIP phones, as well as existing connectivity to the PSTN. This is also illustrated in Figure 2.5.

Table 2.8 Functions Performed by SIP Servers

Register SIP phones and other SIP devices
Register individual end users for access to their services
Register end-user preferences
Perform Authentication, Authorization, and Accounting (AAA) for end users
Look up the address of the other endpoint
Route call requests to the appropriate server
Route to devices according to user preferences
Support user mobility across networks and devices
Register, filter, and publish information about presence
Inform users of call progress, success, or failure
Communicate requests for QoS to other network elements

While the approach in Figure 2.5 avoids sudden disruptions to the enterprise voice services, voice-only users cannot benefit from presence, IM, and multimedia. The challenge in such mixed environments is to integrate presence, IM, and multimedia on desktop PCs and on laptops (not shown in the diagram) with the legacy voice services. SIP-based systems can support this integration.

SIP Security

The advantages of Internet communications with the world can also have unfortunately similar security vulnerabilities as found in e-mail and on the web, unless security is built into SIP implementations right from the beginning.

Spam, or the unsolicited transmission of bulk messages in e-mail, can also happen for VoIP and IM in various ways as described in reference [27]. There is, however, an ample solution space to protect from SPAM in SIP.

There is an arsenal available to implement SIP security that includes many facets, such as the protection of the `REGISTER` method, denial-of-service (DOS) prevention, and Transport Level Security (TLS). There is no single security mechanism that can address all security threats for SIP, but a minimal approach has been documented in reference [28].

An interesting debate with regard to SIP security is how much effort should be put into perimeter security only, such as firewalls, versus security in the endpoints, or in a mix of the two. This debate is illustrated in new security approaches that do not depend on the security of the perimeter [29].

Nothing can be more dangerous than the expectation that SIP and communication security can be purchased in a box, though regrettably, there are many such products that promote this idea marketed at present. Other common security pitfalls come from assuming that there is safety in any specific closed IP network, without considering the wide variety of vulnerabilities from inside the closed networks or from infected applications that may be imported in various ways.

Similar to security in general, good SIP security is based on the quality of the software and on security procedures and practices. Useful directions can be found in white papers from the security community involved in SIP, such as reference [30].

SIP Accessibility to Communications for the Hearing and Speech Disabled

Hearing- and speech-disabled people use text such as Text over IP (ToIP) or IM and video to communicate with other users or among themselves. The selection of the media types in SIP enable the automatic insertion of relay services

that can, for example, transcode speech to text and text to speech, or insert video for using a sign language. SIP-based communications for the disabled are described in Chapter 17, "Accessibility for the Disabled."

SIP Orphans

Not all SIP-related work has as yet found its way into Internet standards, but some of it is quite interesting.

SIP has been considered for other applications as well, besides VoIP, IM, and multimedia. Significant work has been done, for example, to use SIP for the control of home appliances [31]. P2P SIP has also been discussed for communication between home entertainment devices [32].

Commercial SIP Products

At the time of writing the first edition of this book, published in 2001, we still tracked the vendors for SIP products and service providers using SIP. This is no longer possible, since practically everyone in the IT communications industry, Internet products, and telecom services, wired and wireless are now using SIP. It would be easier to count the exceptions, services that do not use SIP, though there are a few significant ones, like Skype and Google Talk in 2005. Even such closed services use SIP to connect to the rest of the world. A well-ordered list of SIP products can be found on the `pulver.com` web site `www.pulver.com/products/sip`. The SIP products and services can be classified in many ways from a market segment or from a technical perspective. Following are categories on the `pulver.com` web site:

- SIP-aware firewalls and NATs
- SIP-PSTN and SIP-PBX gateways
- SIP servers
- SIP services
- SIP software components
- SIP software tools
- SIP user agents for the PC/laptop and for PDAs and mobile phones

Figure 2.6 shows two examples of popular desktop SIP phones.

a b

Figure 2.6 Desktop SIP phones: (a) Business SIP phone; (b) Consumer SIP videophone
(a) Courtesy: snom Technology; AG (b) Courtesy Grandstream Networks, Inc.

What SIP Does Not Do

The preceding list of communications services that can be provided by SIP should not leave the impression that SIP is a "miracle protocol" that can solve all communications problems [33].

As will be discussed in Chapter 6, "SIP Overview," SIP is a very powerful, yet simple and general protocol for establishing interactive communication sessions across the Internet. SIP is a protocol for initiating, modifying, and terminating interactive sessions. This process involves the discovery of a user, wherever he or she may be located, so that a description of the session can be delivered to the user. There are quite a number of features and services that SIP was *not* designed to support, such as the following:

- SIP is not meant to replace all known telephony features and services from circuit-switched networks with identical services. There are many telephony services that have their rationale because of the limitations of circuit-switched technology and in legacy telecommunications regulation, rather than in objective needs for communication. The majority of the countless Class 5 telephone switch features make no sense on the Internet. Local telephone number portability is another example of a service that makes no sense on the Internet. While SIP can support local

number portability, on the Internet such a service is not required in the first place, since URIs have no geographic significance. Caller ID is another paid "service" that makes no sense for SIP, since just like in e-mail, the `To:` and `From:` headers are always there without extra cost.

- SIP is not a transfer protocol such as HTTP, designed to carry large amounts of data. It is designed to transport only small amounts of data required to set up interactive communications. Small amounts of data not related to call setup (such as short text messages for instant messages) are well suited for SIP, as will be shown in Chapter 13, "Presence and Instant Messaging," but large amounts of general data are not suited for carrying by SIP.
- SIP is not a resource reservation or prioritization protocol, so it cannot ensure QoS but can only interwork with other protocols designed to support QoS, as will be discussed in Chapter 18, "Quality of Service for Real-Time Internet Communications."

Divergent Views on the Network

No book on SIP would be complete without mentioning the fact there are completely divergent views on the network and how SIP will be used [34].

The Internet view on the network is as follows:

- The Internet is *the* network. The next generation network (NGN) is IPv6.
- The Internet is transparent e2e or just "dumb;" it is application-unaware.
- User consent and control resides in the endpoints.
- Service availability is what matters to users and not QoS. QoS is good as long as network congestion is avoided and, if so, voice quality is an endpoint capability.
- The Internet is the result of a continuous evolution, and the architecture changes constantly over time [35].

The ITU-T view on the network is as follows:

- The NGN will be derived form the PSTN, but using IP technology; the IP Multimedia System (IMS).
- The NGN is application-aware.
- Control resides in the network.
- The NGN has ample QoS definitions and guarantees for the network service.

- All ITU-T NG networks (such as ISDN, BISDN/ATM, IMS/NGN/IP) are based on grand designs and are not based on a continuous evolution. The changes from TDM to ATM to IP are significant discontinuities in the ITU-T architectures.

The authors of this book fully share the IETF view that NGN and IMS are technically a protocol layer violation (application-aware networks). Such networks are, therefore, not scalable and may collapse under their own weight or from sheer complexity or for various technical reasons.

Moreover, we believe that the central control in NGN and IMS takes the control away from users and will fail in the open market where users have a choice. For example, someone using Google, Yahoo!, or MSN to search for a product and then going to Amazon.com or eBay to buy it, may want to use the voice services of those companies and not be restrained in any way by the constraints imposed by IMS or NGN services.

Summary

SIP has all the marks of a thoroughly disruptive technology. It will fundamentally change communication services as we know them today and also the communication habits of users [36]. The complete integration of communications with the web and e-mail has thus started and much innovation and the resulting new services are still ahead. SIP and its related protocols prove to be the enabling ingredients for new communications, much like its model protocol HTTP 1.1 was to the World Wide Web.

Chapter 21, "Conclusions and Future Directions," summarizes the information on future work and current directions for IP communications.

References

[1] *Newtons' Telecomm Dictionary*. 21st Edition. CMP Books, 2005.
[2] The search index page for IETF RFCs is `www.rfc-editor.org/rfcsearch.html`.
[3] The True Picture of Peer-to-Peer Filesharing: `www.cachelogic.com/research/slide1.php`.
[4] Internet usage statistics: `www.sims.berkeley.edu/research/projects/how-much-info-2003/internet.htm`.
[5] Nimcat Networks at: `www.nimcatnetworks.com/Products.aspx`.
[6] Peerio ("Because people don't need servers") at `http://peerio.com`.
[7] "A Presence Event Package for SIP" by J. Rosenberg. RFC 3856, IETF, August 2004.
[8] "Indication of Message Composition for Instant Messaging" by H. Schulzrinne, RFC 3994, January 2005.

[9] "SIP Extension for Instant Messaging" by B. Campbell et al. RFC 3428, IETF, December 2002.
[10] See for example www.xten.com.
[11] "SIP Basic Call Flow Examples" by A. Johnston et al. IETF, RFC 3665, December 2005.
[12] "SIP-PSTN Call Flows" by A. Johnston et al. IETF, RFC 3666, December 2005.
[13] "SIP Service Examples" by A. Johnston et al. Internet draft, work in progress, IETF 2005. http://www1.ietf.org/internet-drafts/draft-ietf-sipping-service-examples-09.txt.
[14] "Programming Internet Telephony Services" by J. Rosenberg, J. Lennox, and H. Schulzrinne. Columbia University Tech/Report CUCS-010-99, 1999.
[15] "CPL: A Language for User Control of Internet Telephony Services" by J. Lennox and H. Schulzrinne. IETF, October 2004.
[16] "Common Gateway Interface for SIP" by J. Lennox et al. RFC 3050. IETF, January 2001.
[17] "SIP Specifications and the Java Platforms" P. O'Doherty et al. SUN Microsystems, 2003.
[18] "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)" by P. Fallstrom et al. RFC 3761, IETF, April 2004.
[19] "SIP-H.323 Interworking Requirements" RFC 4123 by H. Schulzrinne and C. Agboh. IETF, July 2005.
[20] "Interworking SIP and IN Applications", RFC 3976 by V. K. Gurbani et al. IETF, January 2005.
[21] "MIME media types for ISUP and QSIG Objects", RFC 3204 by E. Zimmerer et al. IETF, December 2001.
[22] "SIP for Telephones: Context and Architectures", RFC 3372 by A. Vemuri, IETF, December 2002.
[23] "SIP and SOAP". White paper by N. Deason, www.ubiquitysoftware.com/pdf/SIP_and_SOAP_1-3.pdf.
[24] "Toward the PSTN/Internet Inter-Networking—Pre-PINT Implementations" by H. Lu et al. RFC 2458, IETF, November 1998.
[25] "The SPIRITS (Services in PSTN requesting Internet Services) Protocol" by V. Gurbani et al. RFC 3910, IETF, October 2004.
[26] "Telephony Routing over IP (TRIP)" by J. Rosenberg et al., January 2002.
[27] "SIP and SPAM" by J. Rosenberg. Internet draft, IETF, February 2005.
[28] "A Minimalist Security Framework for SIP" by H. Schulzrinne. Internet draft, IETF, November 2001.
[29] See the Jericho Forum at www.opengroup.org/jericho.
[30] "The Art of SIP fuzzing and the Vulnerabilities found in VoIP" by M. Varpiola. The Blackhat Briefings Conference, 2005. www.codenomicon.com/media/white-papers/bh-us-05-nuwere-varpiola.pdf.

[31] "Framework Draft for Networked Appliances using SIP" by S. Moyer et al. www.softarmor.com/wgdb/docs/draft-moyer-sip-appliances-framework-01.txt.

[32] See www.p2psip.org/ietf.php on the ad-hoc meeting at the 63 IETF.

[33] "Guidelines for Authors of Extensions to SIP" by J. Rosenberg. Internet draft, work in progress, IETF, February 2005. www.ietf.org/internet-drafts/draft-ietf-sip-guidelines-09.txt.

[34] Groups try to chart the future of IP nets, Network World 09/26/05, www.networkworld.com/news/2005/092605-ngn.html.

[35] RFC 1958: "Architectural Principles of the Internet" by B. Carpenter, IETF, June 1956.

[36] *The Economist* magazine. "Almost-free Internet phone calls herald the slow death of traditional telephony," September 15, 2005.

CHAPTER 3

Architectural Principles of the Internet

After the overview on SIP-based IP communication services presented in Chapter 2, we will provide here a brief summary of the Internet architecture (from the perspective of real-time communications) by contrasting it to the telecom-style circuit-switched networks. This review will facilitate an understanding of the chapters that follow.

Telecom Architecture

We refer readers to the numerous references available on ITU-T telecommunications networks. The complete, original, and up-to-date documents are available to be purchased from the ITU-T at the following address:

```
http://itu.int/home/index.html
```

Other relevant documents (such as for Frame Relay, ATM, and Multiprotocol Label Switching (MPLS) networks) can be obtained at no cost at the following address:

```
www.mfaforum.org
```

Figure 3.1 shows the architecture of all ITU-T-style circuit-switched networks.

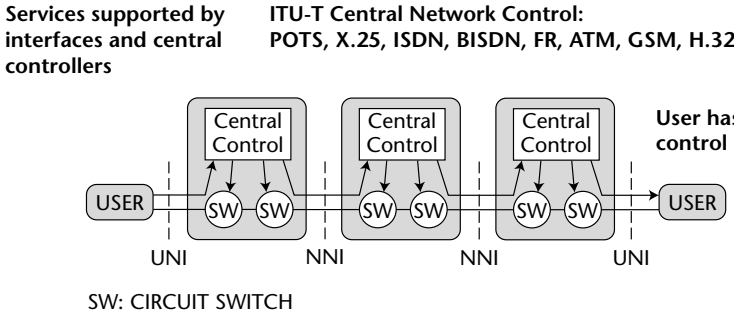


Figure 3.1 Circuit-switched network model

The main features of circuit switched-architectures such as Time Division Multiplex (TDM), Frame Relay, and ATM are:

- Central control, such as the Intelligent Network (IN) for TDM in each network for setting up the paths across each of the respective networks.
- Telecom standards are focused on interfaces. The most common interfaces are standard user-to-network interfaces (UNI) and network-to-network interfaces (NNI).
  - Services must be supported by the features in all NNIs, UNIs, and in all the central control units.
  - New services require new standards support in all UNIs, NNIs, and central controllers.
- No service intelligence is presumed in the user devices. Users have no control over the applications or the choice of services, except for the services made available by subscription by the service provider, such as the local telephone company or the mobile phone company.

These features are an impediment for the richness and time-to-market for new services, as well as for user choice of services. Experience has shown that this type of network is not favorable for innovation, since independent developers cannot get easy access to the central control platforms of the service providers or influence the interface standards in a timely manner.

Following are some features that are more of an engineering concern and are also reflected in the high cost of service:

- The circuit-switched nature requires the keeping of state for a connection or call in every switch, and in several switch subsystems in every switch, as well as in every central control unit. State is also kept in all networks in the path between the parties. State requires expensive processing and memory in all network elements and components where state is kept.

- There are single points of failure. Protection against network failures requires carrier-grade equipment (the so-called 99.999 percent, or "five nines" reliability for equipment), standby equipment, and entire standby network paths.
- Telephone network standards such as Signaling System 7 (SS7) are not global. Countless regional variants, profiles, and various options are permitted. As a consequence, interoperability of telecom networks is a hard problem and is usually achieved only for the least-common denominator of standard features.

In spite of these comments made in hindsight, the global telecom networks amount to close to a trillion-dollar industry that is still robust because of mobile telephony. Most of the Internet traffic is also still carried on telecom-type transmission systems, such as on SONET or SDH links.

The growth of telecom transmission systems in developed countries is, however, predominantly because of the Internet, and this indicates the probable near-term end of the life cycle for most telecom networks (except for mobile phone networks). The end of the life cycle for telecom networks can probably be explained by the absence of new services that has been observed for some time.

Figure 3.2 shows an example of the architectural concepts in the telecom industry. The IP Multimedia Subsystem (IMS) is chosen as the reference architecture for third generation (3G) mobile telephone networks in the 3GPP organization (see `www.3gpp.org`). As of this writing, similar concepts are also under development in the telecom standards organizations ETSI and ITU-T (NGN groups).

Explaining this architecture is beyond the purpose of this book, and explanations can be found in many online resources, for example in magazine articles such as in reference [1]. The IMS architecture diagram is shown here only to illustrate the following points:

- A detailed diagram describes the reference architecture.
- The various functions are called out as boxes in the diagram.
- The links for signaling and media are specified between functions.
- The protocols are specified between the functional entities.
- The applications reside in application servers and the only endpoints shown are black telephones.
- Last, but not least, while the PSTN/ISDN networks are specified, the Internet is presumably included in the box labeled "Other IP Networks" but is not shown.

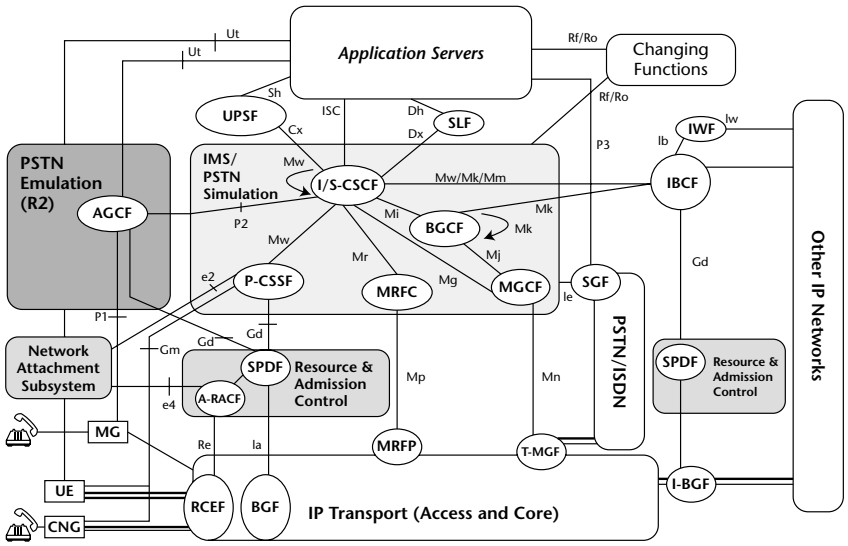


Figure 3.2 IMS reference architecture
Courtesy R. Stastny, ÖFEG

Besides such features illustrated by the architecture diagram for the IMS in Figure 3.2, the following are several other telecom standards procedures:

- Detailed standards documents are developed before research and trials of experimental systems.
- Telecom standards documents usually do not bear the names of their authors and are considered to have been issued by organizations, often driven by marketing departments.
- Telecom standards are driven by commercial business models and the constraints of time-to-market. As we will see later in this chapter, Internet standards have very different drivers.

Internet Architecture

The engineering of Internet communications differs in many ways from telecommunications engineering. We will quote the relevant passages from RFC 1958, "Architectural Principles of the Internet," [2] by Brian Carpenter and reproduce some paragraphs, since we find it impossible to articulate the issues in any better way.

The end-to-end argument is discussed in depth in Saltzer [3]. The basic argument is that, as a first principle, certain required end-to-end functions can only be performed correctly by the end systems themselves. A specific case is that any network, however carefully designed, will be subject to failures of transmission at some statistically determined rate. The best way to cope with this is to accept it, and give responsibility for the integrity of communication to the end systems. Another specific case is end-to-end security. To quote from Saltzer:

"The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the endpoints of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.)

This principle has important consequences if we require applications to survive partial network failures. An end-to-end protocol design should not rely on the maintenance of state (that is, information about the state of the end-to-end communication) inside the network. Such state should be maintained only in the endpoints, in such a way that the state can only be destroyed when the endpoint itself breaks (known as fate-sharing). An immediate consequence of this is that datagrams are better than classical virtual circuits. The network's job is to transmit datagrams as efficiently and flexibly as possible. Everything else should be done at the fringes.

To perform its services, the network maintains some state information: routes, QoS guarantees that it makes, session information where that is used in header compression, compression histories for data compression, and the like. This state must be self-healing; adaptive procedures or protocols must exist to derive and maintain that state, and change it when the topology or activity of the network changes. The volume of this state must be minimized, and the loss of the state must not result in more than a temporary denial of service given that connectivity exists. Manually configured state must be kept to an absolute minimum."

RFC 1958 on the architectural principles of the Internet deals with many other issues, such as focus on the network layer protocol, scalability, heterogeneity, security, simplicity, internationalization, standards proven by interoperable implementations, and others. Those issues are, however, beyond the scope of this book, and you should read this important document separately.

Following is another important observation by B. Carpenter:

The current exponential growth of the network seems to show that connectivity is its own reward, and is more valuable than any individual application such as mail or the World-Wide Web.

Figure 3.3 shows a summary graphic representation of the end-to-end control model of the Internet.

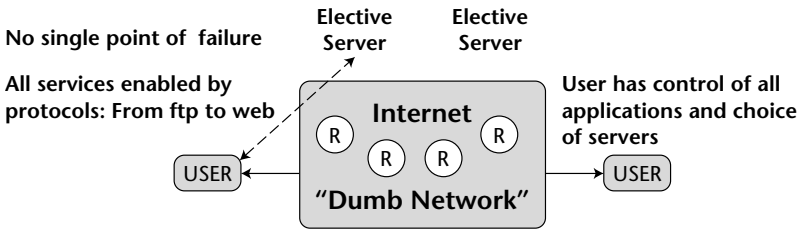


Figure 3.3 Internet model

The Internet is characterized mainly by the following:

- Datagram-oriented instead of circuits.
- No single point of failure.
- End-to-end transparency for applications (see below for how this has degraded).
- End-to-end control.
- End-to-end security.
- Complete control by users over the applications and selection of services.

Internet standards are focused on protocols and not on interfaces, specifying only how the devices communicate across the Net.

The end-to-end control design of the Internet cannot be always maintained, because of the loss of Internet transparency from various developments such as NAT and other devices. This problem is discussed in RFC 2775 [4]. Besides NAT, for SIP, the so-called Session Border Controllers (SBC) or back-to-back user agents (B2BUA) discussed in Chapter 10, "NAT and Firewall Traversal" are also destroying Internet transparency.

The Internet Backbone Architecture

One of the key advantages of the Internet is complete independence of the applications from the network, except for such performance metrics as bandwidth and availability of service. Packet loss, delay, and jitter on the Internet have improved to such a degree as to be negligible for voice in most parts of the world [5]. This is shown in Figure 3.4.

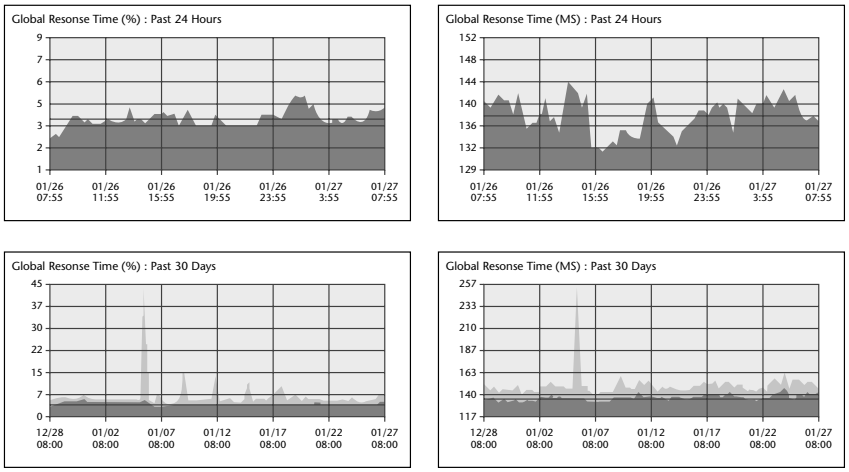


Figure 3.4 Packet loss and delay on the Internet

Some critical knowledge about the network part of the Internet is still useful for making decisions about applications and communications. Table 3.1 provides a short summary of the Internet architecture, guidelines, and philosophy [6].

Table 3.1 Internet Architecture, Guidelines, and Principles

GUIDELINES AND PRINCIPLES	DESCRIPTION
The Simplicity Principle	Application knowledge and state does not belong in the network, although the Internet core performs complex routing and maintains routing state. The complexity of the Internet belongs at the edges and any additional complexity is added above the IP layer. The IP layer of the Internet should remain as simple as possible. Complex systems have nonlinear catastrophic behavior that is thus avoided here. Loose coupling between components is desirable to reduce the probability of failures. (Note that complexity is also expensive, but, unfortunately, designing complex solutions is easier than designing simple ones.)

(continued)

Table 3.1 *(continued)*

GUIDELINES AND PRINCIPLES	DESCRIPTION
Layering Considered Harmful	The optimization for each layer must be done separately. Introducing extra layers can lead to violation of the simplicity principle and add inefficiency caused by duplication of functions, such as error control. See the section, "Middle-Age Symptoms of the Internet," later in this chapter.
Optimization Considered Harmful	Optimization introduces complexity and tighter coupling between components and layers, and leads to less reliable systems.
Feature Richness Considered Harmful	Feature richness in the network increases system complexity and, therefore, has a higher risk for failures and the certainty for higher cost.
Transport Efficiency for IP	IP, Frame Relay, ATM, SONET and WDM are examples of inefficiency often found in carrier networks, without even mentioning cases of double-layering (such as IP, ATM, and SONET).
Avoiding Interworking Functions	Interworking functions between networks increase complexity and degrade performance. Avoid control plane interworking, such as that between ATM and MPLS. The payload of IP packets should (where possible) be transmitted without modification.
IP is Quite Complex	Packet switching may not be simpler than circuit switching, and IP routers can be quite expensive. Where possible, lower-layer switching (Ethernet) and optical switching may be preferable.
The Myth of Overprovisioning	Overprovisioning of IP networks that keeps utilization under 50 percent is not much worse than the efficiency of TDM networks, assures good network performance, and can be considered a 1:1 protection service at the IP layer.
Quality of Service	QoS is deployed when there is insufficient bandwidth to support traffic. Packet drop may have a more severe impact on TCP traffic than on voice. QoS is better assured by avoiding congestion than by complex software.

Table 3.1 *(continued)*

GUIDELINES AND PRINCIPLES	DESCRIPTION
The Myth of Five Nines	The Internet can provide better than 99.999 percent availability without depending on equipment with "five nines" reliability. DNS, for example, has never failed since its introduction in 1986, though individual DNS servers may have quite modest reliability. The Internet has proven to provide better than PSTN availability in all recent disasters in North America, Europe, and Asia. Outages are only partial and are mostly caused by people or by disasters.
Simple Delivery Paths	The quality of a path between users, or between users and a service (server), is very sensitive to the number and complexity of the elements in that path.
Security is Enhanced by Simplicity	The simplicity principle is helpful for security, since security implications are easier to understand.

You should compare the principles of the Internet architecture shown in Table 3.1 with the telecom architectures exemplified with the IMS shown in Figure 3.2. The differences are too many to be summarized by a simple bulleted list or table. Note the following, however:

- None of the Internet architecture documents referenced here have any diagrams, although some Internet standards have some diagrams that are simple enough to be drawn as stick drawings using simple text symbols and lines.
- All Internet document authors are listed as individual contributors, and many of them are well-known personalities from research and academia.
- Internet standards are driven by the desire for excellence in engineering by its authors.
- Internet standards are not driven by marketing departments or by time-to-market commercial interests, though the marketplace is considered the ultimate arbiter for technology.

The Internet Standards Process

How are the so-very-successful Internet standards produced?

The success of the Internet is attributable to its excellent standards, and this excellence is the result of a standards process that has been carefully developed over the years, starting with Request for Comments (RFC) number 1, "Host Software," by Steve Crocker published on April 7, 1969.

One notable distinction of the Internet standards process is its complete openness—everything is available online at all times and without cost to everyone who may be interested. Online participation in the standards' development is open to anyone who feels qualified to contribute.

Another notable distinction is that standards are proposed and developed by individual contributors and not by organizations or corporate entities. To quote freely Fred Baker from Cisco, a former chair of the IETF:

> *Make sure you represent your own opinion, so that you don't have to change it if you change jobs.*

We also observe the predominance of computer scientists from research and academia in the Internet standards community, people who are less process-oriented, but have a keen sense and hands-on knowledge of networks and computing.

The Internet standards process at present is best described in reference [7]. In a nutshell, the goals of the Internet standards process are as follows:

- Technical excellence
- Prior implementation and testing
- Clear documentation
- Openness and fairness with regards to intellectual property
- Timelines from birth to obsolescence of Internet standards

Internet standards fall into the categories of technical specifications and associated applicability statements. There are several requirement levels, such as Required, Recommended, Elective, Limited Use (such as Experimental), and Not Recommended [any more for various reasons].

Of special interest is the Internet standards track that is defined by several maturity levels:

1. *Proposed Standards*—Further experience may result in changes or even retraction of the specification.
2. *Draft Standards*—Specifications for which at least two independent and interoperable implementations from different code bases have been developed and tested for interoperability.

3. *Internet Standards*—Specifications that have significant implementations and for which a successful operational experience has been obtained.

For various reasons, not every specification is on the standards track, and such specifications have the maturity levels of "Experimental," "Informational," or "Historic."

Last, but not least, there are carefully designed procedures, called *Standards Actions* on how documents of the IETF are published, discussed, and processed. Standards Actions include (but are not limited to) advancements on the standards track, revisions, and retiring of standards.

Internet standards may incorporate (by reference) other open standards, such as the American National Standards Institute (ANSI) standard for the ASCII character set.

The IETF has very detailed rules on intellectual property rights so that the benefits to the Internet community are not in conflict with the legitimate rights of others. For this reason, contributions that are subject of confidentiality or have any other restrictions are not acceptable. Submissions that may be subject to copyright grant the Internet Society (ISOC) the license to the contribution, and must take into account quite complex other legal requirements that go beyond the scope of this book.

Protocols and Application Programming Interfaces

The reliance on protocols in Internet engineering is another significant difference from the practice in the software industry to use application programming interfaces (APIs).

Internet protocols specify only how processes running on different computing devices on the Internet communicate "across the wire" and do not impose any restriction on how the applications and protocol machines are implemented (this is best left to the creativity and competitiveness of the software developers).

By contrast *APIs* are commonly used to program most applications by developers. APIs are, however, most often owned and under the control of the software vendor. In addition, they are often written for a specific operating system only.

Users and developers of telecommunication equipment are informed of the "open APIs" for the product so as to develop or customize services. Remember, however, that "open APIs" introduce a certain level of dependency on (1) the software vendor and (2) the operating system vendor, because they both have intellectual property rights and change control.

Protocols are preferred on the Internet for these reasons, since any Internet standard or practice must be completely vendor- and computing-platform-independent. Moreover, a core design principle for the Internet and the World

Wide Web is that the various parts should be designed and implemented independently of each other, and yet still interoperate flawlessly. This is experienced daily by everyone using file transfer, e-mail, the web, or any other standards-compliant applications over the Internet, on a truly global scale.

Is XML the Presentation Layer of the Internet Protocol Architecture?

As will be seen in Chapter 13, "Presence and Instant Messaging," the new preferred data format for most SIP related protocols is XML [8]. XML has become the design choice by default for Internet application data and seems to be the equivalent of a presentation layer in the Internet protocol architecture.

By similar arguments, it appears that SIP and Real Time Streaming Protocol (RTSP) are the de facto session layer for the Internet.

Middle-Age Symptoms of the Internet

The classical model of the Internet protocol stack has an hourglass architecture with the following three parts:

1. *The base of the hourglass*—Various Layer 2 link protocols such as Ethernet, SONET, cable, and DSL link layers, as well as many types of radio links.
2. *The slim middle (waistline)*—One single Layer 3 protocol (IP).
3. *The upper part of the hourglass*—Several Layer 4 transport protocols (such as User Datagram Protocol [UDP] and Transmission Control Protocol [TCP]) and many application Layer 5 protocols (such as FTP, SMTP, HTTP, and SIP).

The maturity of the Internet has led to some symptoms of middle-age [9], such as the thickening of the waistline and other symptoms, including the following:

- *More functionality*—The push for QoS requires more functionality from the underlying Layer 2 networks resulting in additional complexity.
- *Layer splitting*—A new Layer 2.5 is emerging with the use of MPLS and L2TP used for certain virtual private networks (VPN).
- *Layer violations* — Putting various functions into the IP layer, such as packet inspection for security.

The authors take a dim view of all these and believe that the market will prove the original Internet design to be the most effective network alternative, in spite of the seemingly endless commercial initiatives to "add value to the network" by carriers and their suppliers.

The transition from IPv4 to IPv6 will also require doubling the number of interfaces below and above the IP Layer 3, but this seems to be a normal penalty for such a key transition.

Fighting Complexity

Internet engineering guidelines have always stressed the importance of finding the simplest possible solution to a problem. As the complexity of Internet technology continues to grow, this topic tends to surface in email lists and at various conferences have been held on the topic of avoiding complexity, such as in "IMS 101"[1].

Figure 3.5 shows the self perpetuating circle of complexity as presented at the North American Network Operators Group (NANOG) in October 2002.

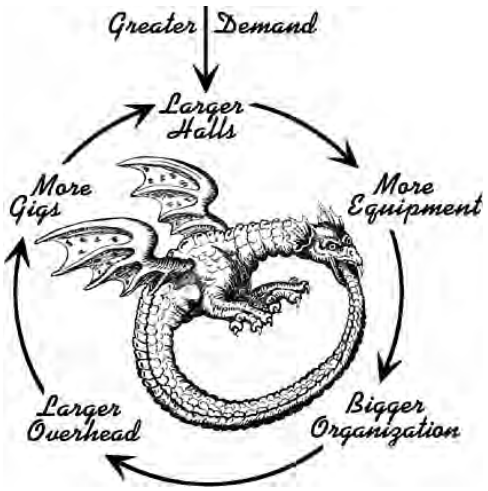


Figure 3.5 The circle of complexity and its components.
Courtesy David Meyer.

Summary

The emergence of the Internet as both *The Network* and *The Service* has occurred because of its technical excellence with deep roots in research and academia. The Internet standards are based on interoperable implementations and operational experience. The resulting global connectivity between machines and humans at the edge of the network has proven to be a true historical engine for innovation and a driver for the global economy.

References

[1] "IMS 101: What You Need To Know" by J. Waclawsky, *Business Communication Review*, June 2005. `www.bcr.com/bcrmag/2005/06/p18.php`.

[2] "Architectural Principles of the Internet" by B. Carpenter, RFC 1958, IETF, June 1986. Updated by RFC 3439.

[3] "End-To-End Arguments in System Design" by J. H. Saltzer, D. P. Reed, D. D. Clark, ACM TOCS, Vol. 2, No. 4, November 1984, pages 277–288.

[4] "Internet Transparency" by B. Carpenter, RFC 2775, IETF, February 2000.

[5] "ICFA SCIC Network Monitoring Report," updated February 2005. `www.slac.stanford.edu/xorg/icfa/icfa-net-paper-jan05`.

[6] "Some Internet Architectural Guidelines and Philosophy" by R. Bush and D. Meyer, RFC 3439 IETF, December 2002.

[7] "The Internet Standards Process - Rev. 3" by S. Bradner, RFC 2026, IETF, October 1996.

[8] Extensible Markup Language (XML) of the World Wide Web Consortium, see: `www.w3.org/XML`.

[9] "Modern Internet Architecture and Technology" by H. Schulzrinne, Dept. of Computer Science, Columbia University class, Fall 2003.

CHAPTER

4

DNS and ENUM

This chapter introduces the Domain Name System (DNS), and ENUM, a telephone number mapping system for Internet resources. Readers interested in a more thorough understanding of the DNS and ENUM can find books on these topics or a few high-quality free online tutorials. (Unfortunately, many DNS online tutorials have not been updated for quite some time.) A good online resource for understanding DNS can be found for example at `http://technet.microsoft.com/default.aspx` [1]. For an in-depth study of DNS we recommend the authoritative online tutorial available from the IETF and further reading of the references provided there [2].

Introduction

Readers will easily recall how their web browser finds the IP address for a web site. For example, when you enter for the first time (the web site address and the web pages were not yet cached by the browser) the web site name:

```
ietf.org
```

the browser will first display (usually in the bottom-left panel) a message such as:

```
Finding site: ietf.org
```

Since the built-in DNS client will talk to one of the configured DNS servers in the local IP protocol stack (DHCP or static configuration), when the DNS returns an IP address, the browser will then display:

```
Contacting: ietf.org
```

followed by

```
Web site found. Waiting for reply...
```

The web page content is transferred to the browser after the web server has responded. The speed of the response may sometimes make it difficult to follow this message sequence. The high speed of the this process, given reasonably fast Internet access, shows the power of the global Internet DNS system.

SIP is closely related to web technology and can make similar use of DNS-based lookup.

The preceding steps in finding a web site illustrate very roughly how the Internet DNS is used by the browser by performing the following operations:

1. The local DNS resolver sends a DNS query for `ietf.org` (`Finding site...`)
2. The DNS returns the IP address for the site.
3. The HTTP request from the browser is sent there (`Contacting...`)
4. An initial reply comes from the web site (`Web site found...`), but this is not a DNS operation any more.

The DNS operations will be described summarily further in this chapter, before proceeding further with the SIP topics. All SIP topics presume some understanding of the underlying DNS operations, as well as ENUM [3], which is of particular interest to VoIP.

The *DNS client* located in the application is also called a *DNS resolver*, and we will use here these names interchangeably.

Addressing on the Internet

Understanding the DNS and directories requires a brief review of Internet and web addressing that we provide here. Readers familiar with this topic can proceed directly to the section on DNS.

The Universal Resource Identifier (URI)

The *Universal Resource Identifier (URI)* [4] is a name associated with a universal set of names in a registered naming space, such as Internet domain names

registered with the Internet Assigned Number Authority (IANA) [5] or host names registered for a specific domain. URIs are independent of the location (a specific host) of the named object. For example:

```
mailto:firstname.lastname@example.com
```

is a URI associated with e-mail. Note the mail URI does not specify any specific computer.

mailto:

The `mailto:` URI schema [6] designates an Internet mailing address of an individual or service. It does not represent an actual Internet location, but serves only to route Internet mail. For example:

```
mailto:firstname.lastname@example.com
```

The `mailto:` URI appears frequently in web pages for providing e-mail feedback. In addition, the values of certain Simple Mail Transfer Protocol (SMTP) headers can be prepopulated by the URL. For example, clicking on the URI that follows in a web browser will bring up a blank e-mail message addressed `To: webmaster@example.com` and with the `Subject:` header set to "`Feedback`":

```
mailto:webmaster@example.com?Subject=Feedback
```

A mailto: URI can appear in a SIP message as part of a list of Contact headers.

URI schemas are associated with various protocols and services, such as: FTP, HTTP, Mail, News, SIP, Telnet, and others.

In SIP, a Request-URI is defined in RFC 2543 as a type of URI used to indicate the name of the destination for the SIP Request (`INVITE`, `REGISTER`, `SUBSCRIBE`, and so on). As a SIP Request is forwarded by proxies, the Request-URI can be changed as database lookups and feature invocations change the final destination of the request.

The Universal Resource Locator (URL)

The *Universal Resource Locator (URL)* [7] describes the location of a resource available on the Internet. Contrary to URI schemas, the schemas for URLs are associated with specific hosts, protocols, and services. For example:

```
http://ws1.example.com
```

is web server #1 for the domain `example.com`.

Tel URI

No Internet device needs to have knowledge of telephone numbers and their context-specific meanings, but needs only to understand URIs such as the tel URI [8]. The *Telephony URI* schema specify the "tel" name of a terminal in the phone network as seen from the Internet, and the connection types that can be used to connect to that entity. Telephony URIs can be used for fixed and mobile phone calls and for fax. For example:

```
tel:+1-201-555-0123
```

points to a number in the United States, while:

```
tel:1234;phone-context=munich.example.com
```

indicates that this phone number (1234) may only be used by a SIP-proxy recognizing the phone-context (that is, the SIP server "knowing what to do" within the domain `munich.example.com`).

Digit separators can be "(", ")", "-", or "." (or none). The separators are removed by parsers and are ignored. The grouping of +1-... as the country code for the US in the example is done for human readability.

SIP URIs can also handle telephone numbers. In fact, the entire set of parameters specified for the tel URI can be used in the username portion of a SIP URI.

An example of a parameter in a tel URI is described for the `phone-context`.

The phone-context

Not all tel URIs are as simple as the one shown earlier. The `phone-context` specifies under what circumstances a phone number can be used. For example:

```
tel:1-800-123-4567;phone-context=+1-972
```

refers to an 800 number valid only for the North American numbers in the 972 calling area. The expression 1-800-123-4567 is called a *dial string*—something used by humans.

Note the absence of a + in the URI. Any digits in a tel URI that do not begin with a + are assumed to be a locally valid but not global valid numbers. Using this phone number outside the calling area is not permitted and would generate an error announcement, or result in a misrouted call.

If a `phone-context` tag is not present in the URI, then some other context (typically geographic) must be used to interpret the digits. Since the Internet does not have the kind of geographic isolation typically present in the PSTN, this is a difficult thing to do.

NOTE **In the PSTN, a Class 5 telephone switch in a North American Rate Center covered by only a single numbering plan area code (NPA) may safely assume that any 7 digit number uses the default area code and country code. This is possible because of the structure of the PSTN and the limited connectivity that this switch has to other switches in the PSTN. In a SIP network on the Internet, a proxy is theoretically accessible to any SIP user agent in the world, and such assumptions about global validity of a phone number must not be made.**

SIP URI

SIP URIs [9] used are very similar to e-mail addresses within SIP messages to indicate the originator (`From`) and the final recipient (`To`). SIP also has several new headers such as current destination (Request-URI), intended recipient (`To`), and the direct route address (`Contact`), among others. This will be explained in more detail in Chapter 5, "Real-Time Internet Multimedia."

When used with a hyperlink, the SIP URI indicates the use of the `INVITE` method. SIP URI hyperlinks allow the embedding of links that when opened can initiate a phone call, for example:

```
sip:firstname.lastname@example.com
```

may be used to send a call to a voice mailbox. Or:

```
sip:+1-214-555-1212@gateway.com;user=phone
```

indicates how to address a call from the Internet to the PSTN E.164 phone number +1-214-555-1212 via the IP telephony gateway having the domain name `gateway.com`. The `user=phone` tag is a hint to parsers that a telephone number is present in the username portion of the URI and is not just a numerical name.

The host portion of a SIP URI containing a telephone number does not always indicate a gateway. This is because the creator of the URL may not know the location of the gateway and may instead be relying on a proxy to

locate an appropriate gateway. In this case, the URL for the location of the SIP proxy would look like:

```
sip:+1-214-555-1212@proxy.gateway.com;user=phone
```

where the SIP proxy at `proxy.gateway.com` will determine the gateway and forward the request.

IANA ENUM Service Registrations

ENUM may also be used for other services besides SIP. Several ENUM services have been registered with the IANA, as shown in Table 4.1.

The Domain Name System

The Internet Domain Name System (DNS) [10] is a scalable namespace used to refer to resources on the Internet and in private networks. DNS names avoid specifics such as IP addresses and port numbers.

Scalability for the huge size of the DNS is because of its distributed nature and the use of caching.

Table 4.1 ENUM Services

ENUM SERVICE NAME	SERVICE TYPE/SUBTYPE	URI SCHEME
h323	h323	h323:
sip	sip	sip:, sips:
ifax	ifax:mailto	mailto:
pres	pres	pres:
web	web:http	http:
web	web:https	https:
ft	ft:ftp	ftp:
email	email:mailto	mailto:
fax	fax:tel	tel:
sms	sms:tel	tel:
sms	sms mailto	mailto:
ems	ems:tel	tel:
ems	ems:mailto	mailto:
mms	mms:tel	tel:
mms	mms:mailto	mailto:

Delegation

The actual DNS records for host names and servers, and so on, for a domain are the responsibility of, and under the complete control of, the administrator of that domain.

For example, the Address records (or A records) that contain the IP addresses of servers within the domain "`example.com`" (such as `mail.example.com`, `www.example.com`, `ftp.example.com`, and so on) are maintained there. The domain "`example.com`" is said to be authoritative for the IP addresses, names, local resources, and delegations further down within its own domain. The creation and deletion of names is fully distributed and delegated to lower levels of the DNS in this manner.

Figure 4.1 shows an example with several leaves of the "golden tree" of the DNS. The name "golden tree" is because of the clear structure of delegation for authority within the DNS so as to avoid any errors caused by incorrect duplication of data. This clear structure of the DNS is a key architectural principle of the Internet and is the technical explanation why the Internet has one and only authority for assigning names and numbers—the Internet Authority for Assigned Names and Numbers (IANA).

Caching

Without caching, every DNS query would have to begin at the root DNS server, continue downward to the top-level domain server, and then end at the authoritative DNS server. However, very efficient caching schemes employed in DNS make this the rare exception rather than the rule. Most DNS queries only traverse one or two DNS servers. The price paid for this efficient caching is that DNS changes (updates) do not happen in real time but take significant time to propagate throughout the Internet. As a result, DNS is not suited for roaming and other mobility services where the IP address may change rapidly.

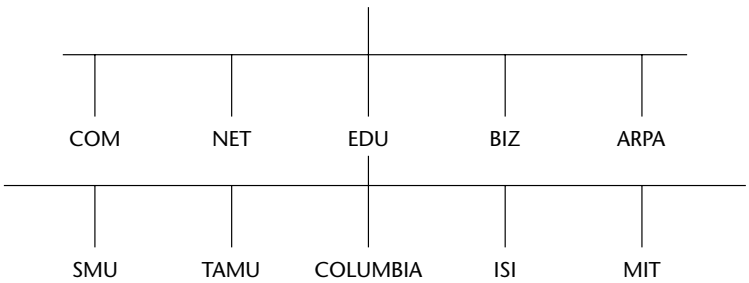


Figure 4.1 Example of the DNS "golden tree" leaves

The DNS is designed to support many applications, such as referring to host addresses or mailbox data, and, as shown here, is also used to locate SIP endpoints and SIP proxies in the network. Address formats differ for various protocols, and the DNS is designed to support various protocols with their own notion of an address (such as the addresses for FTP, HTTP, mail, or SIP).

RFC 2219 [11] specifies the protocols and services. Following are some of the ones of interest:

- *File Transfer Protocol*—FTP (RFC 959)
- *Lightweight Directory Access Protocol*—LDAP (RFC 1777)
- *Network Time Protocol*—NTP (RFC 1305)
- *Post Office Protocol*—POP (RFC 1939)
- *Session Initiation Protocol*—SIP (RFC 3261)
- *SMTP mail* (RFC 821)—SMTP
- *World Wide Web, HyperText Transport Protocol*—HTTP (RFC 1945).

Other real-time communication services are shown in Table 4.1.

The DNS is independent of the underlying transport system for the data and can work equally well with datagrams or with virtual circuits as originally designed, although it is almost universally used with IP datagrams. The DNS is also designed to be used by computers of all sizes, from small personal devices to large computers. RFC 1034 was the initial basis for the DNS system design. Other RFCs have introduced various improvements. Here, we limit the discussion of the DNS to the information required for understanding SIP services that use DNS.

Most data in the DNS system is assumed to change very slowly (for example, mailbox addresses or host name–address bindings). Current IETF work aims at providing faster, dynamic updates. Lower levels of the DNS may also accommodate faster changes, for example on the order of minutes or even seconds.

A Partial DNS Glossary

Table 4.2 shows a summary overview of some DNS terminology, with special emphasis on the use of DNS with SIP, where real-time communications reside.

Table 4.2 DNS and ENUM Terminology

Domain Name	The Domain Name is a list of labels on the path from the node to the root of the tree. For example: `ipcom.example.com` is the name of the host `ipcom` in the example network registered in the `.com` top-level domain (TLD).

Resource Record (RR)	Resource records associated with a particular name, such as for hosts, mail exchanges, web servers, SIP servers. For example,
	`ns.example.com IN A 166.15.21.14`
	provides the IP address (`A`) on the Internet (`IN`) for the host called `ns` in the `example.com` domain. Or,
	`mail2.example.com MX example.com`
	gives the name of the mail server `mail2` for `example.com`. And finally,
	`SIP3.example.com SIP example.com`
	refers to a SIP server (Number 3) in the domain `example.com`.

A Record (A)	An RR that designates a host address. A records are the most commonly used DNS records for translating a domain name into an IP address.

Service Record (SRV RR) [12]	An RR that contains the locations of servers for a specific protocol and domain. For example, the lookup for SIP servers that support the TCP protocol would be:
	`_sip._tcp.example.com`
	The underscore "_" prevents name collisions with other DNS labels. The response could be:
	`sip1.example.com`
	`sip2.example.com`
	`sip3.example.com`
	The response also includes priorities and weights (not shown here) for the target host, indicating to the client how to select the target server. Clients can use a selection mechanism to distribute the load among servers based on priorities and weights.

(continued)

Table 4.2 *(continued)*

	The *failure of one or more servers should not cause a service failure* at all, as long as there are other servers left. If a server fails during a call setup, that call setup will fail, however, and a new call setup request must be made. The use of SRV records to locate a SIP proxy server for a particular domain is described in RFC 3263 [13].
Mail Exchange Record (MX)	Identifies a mail exchange in the domain.
Name Server (NS)	Authoritative name server for the domain.
Pointer (PTR)	Pointer to another part of the domain space. These records are used for backward DNS lookups—resolving an IP address into a domain name.

Naming Authority Pointer (NAPTR) used for ENUM	The Naming Authority Pointer (NAPTR) [3] is an RR written according to specific rules, so that the address lookup by the DNS client can be continued for a new target that is computed from the response.
	
	For example, the E.164 phone number
	
	`+1-770-555-1212`
	
	can be converted for lookup in the DNS of the `e.164.arpa` domain to
	
	`2.1.2.1.5.5.5.0.7.7.1.e164.arpa`
	
	A query sent to `164.arpa` may produce the NAPTR records for the hosts that can further process this address:
	
	`sip:information@example.com`
	`mailto:information@example.com.`
	
	Thus, when the phone number +1-770-555-1212 is dialed, the call will go to the information service of `example.com`.

DNS and ENUM Usage Example

We will illustrate the DNS and ENUM terminology with an example for real-time communications based on SIP. Since SIP will be explained in more detail in the following chapters, the required SIP message exchanges have been summarized in Figure 4.2 as "SIP Transactions," because they are outside the topic of DNS.

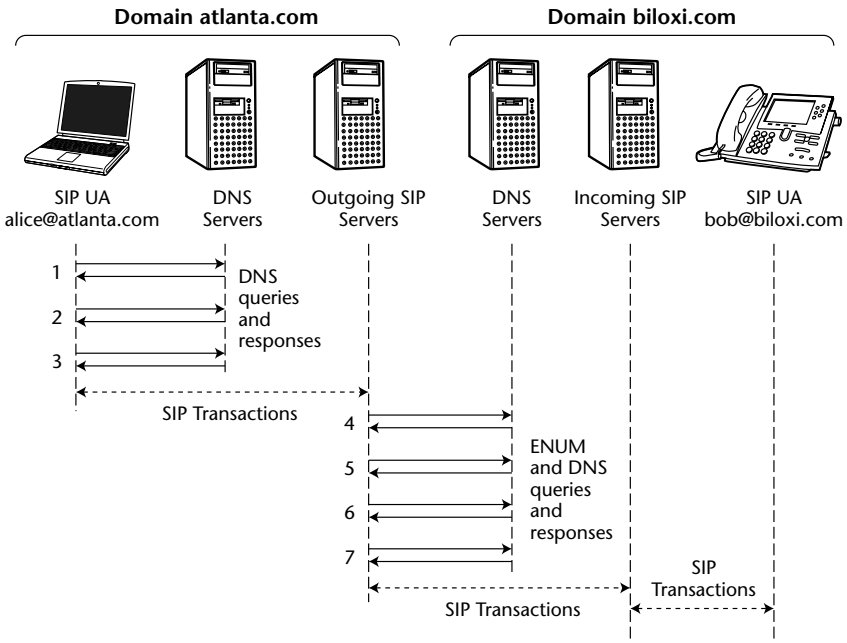


Figure 4.2 High-level DNS and ENUM service example for SIP

Finding an Outgoing SIP Server

We assume the user Alice has a SIP service subscription in the domain `Atlanta.com`. To make use of the SIP service, the SIP user agent (UA) Alice must first register with the outgoing SIP server in the domain `Atlanta.com`. To register, the SIP UA must first:

1. Chose a transport protocol for with adequate security to register; that is to discover Transport Layer Security (TLS) if it is available. SIP signaling using TLS transport is called SIPS in the following.
2. Discover the outgoing SIP servers.

The DNS resolver in the application that also contains the SIP UA will first decide to use SIPS with TLS transport and then do a NAPTR query (query #1 in Figure 4.3) for the SIP servers in the domain `example1.com`.

```
IN NAPTR 50   50   "s"   "SIPS+D2T"
```

Following is the legend for the items in the NAPTR response:

- `IN`—Internet class
- `NAPTR`—Query type
- `50`—Ordering for targeting by the client, RFC3403
- `50`—Preference (is a server selection mechanism)
- `"s"`—Flag to indicate the next lookup will be for a service record
- `"SIPS+D2T"`—DNS service field for TLS (D2T), required for SIPS

The result for the NAPTR query performed in the domain `altlanta.com` is as follows:

```
_sips._tls. altlanta.com  (the underscore "_" prevents name collisions)
```

The SRV query is now (query #2):

```
_sips._tls.-altlanta.com
```

The DNS with return the domain name and IP address of the server that provides the answer (for example, `dns3.example1.com`, `1.2.3.4`) and also SRV locations:

```
sip1.-altlanta.com

sip2.-altlanta.com

sip3.-altlanta.com
```

The DNS resolver will now make the query #3 for an A record, that is to find the IP address of one of the three SIP servers. The pseudorandom mechanism in the client for load sharing could select `sip3.example1.com` for the A type DNS query. The returned IP address could be `1.2.3.14`. Now the SIP UA has the IP address of an outgoing SIP proxy to register for service, and the resulting SIP transaction is shown by the dotted double arrow on the left in Figure 4.2.

Finding an Incoming SIP Server in the ENUM Case

Suppose that `alice@atlanta.com` does not know the SIP URI for Bob, but has only Bob's telephone number: +1-404-555-1234. This is where ENUM comes into play.

Either the SIP UA of Alice or the outgoing SIP server for the domain `atlanta.com` needs to find the URI of Bob when it has only the telephone number for Bob. The queries shown in this example can be performed either by the SIP UA of Alice or by the outgoing SIP proxy for the domain `atlanta.com` shown in Figure 4.3. We would prefer to give Alice the

control and have the DNS client in the SIP endpoint, although for backward-compatibility with many legacy SIP UA implementations, the outgoing SIP proxy can perform these queries as well. Here is an example of how a client application processes a phone number to make a DNS query:

1. Start with the complete E.164 phone number:

   ```
   +1(404)555-1234
   ```

2. Remove all characters that are not digits:

   ```
   14045551234
   ```

3. Reverse the order of the phone number:

   ```
   43215554041
   ```

4. Insert dots between digits and at the end:

   ```
   4.3.2.1.5.5.5.4.0.4.1
   ```

5. Append the DNS top-level domain (`e164.arpa`):

   ```
   4.3.2.1.5.5.5.4.0.4.1.164.arpa
   ```

The client is now ready to make a DNS query using the result from step 5.

The DNS client will make several DNS queries, as shown in Figure 4.3 for `biloxi.com`, and retrieve the IP address of an incoming SIP server in the domain `biloxi.com` where Bob is registered. As a result, the necessary SIP transactions can proceed to set up a session between Alice and Bob.

Of special interest for understanding ENUM are the first query and response in Figure 4.3. The query from the client after step 5 is:

```
4.3.2.1.5.5.5.4.0.4.1.164.arpa
```

The DNS ENUM response will show several services registered for this phone number—for example, SIP and e-mail (RFC 3761):

```
IN NAPTR 100 10 "u" "E2U+sip" "!^.*$!sip:user@biloxi.com!"
IN NAPTR 100 10 "u" "E2U+msg" "!^.*$!mailto:user@biloxi.com!"
```

The legend for the items in the NAPTR response is similar to the legend for the previous example, although there are some new items:

- `"u"` *flag*—Output is a URI for information on the respective service
- `"E2U+sip"`—Service field for ENUM yielding a SIP URI
- `!^.*$!`—`'regular expression for greedy search'` (it matches all) starting with the `"!"` separator
- `sip:user@biloxi.com`—Replacement value for the next query

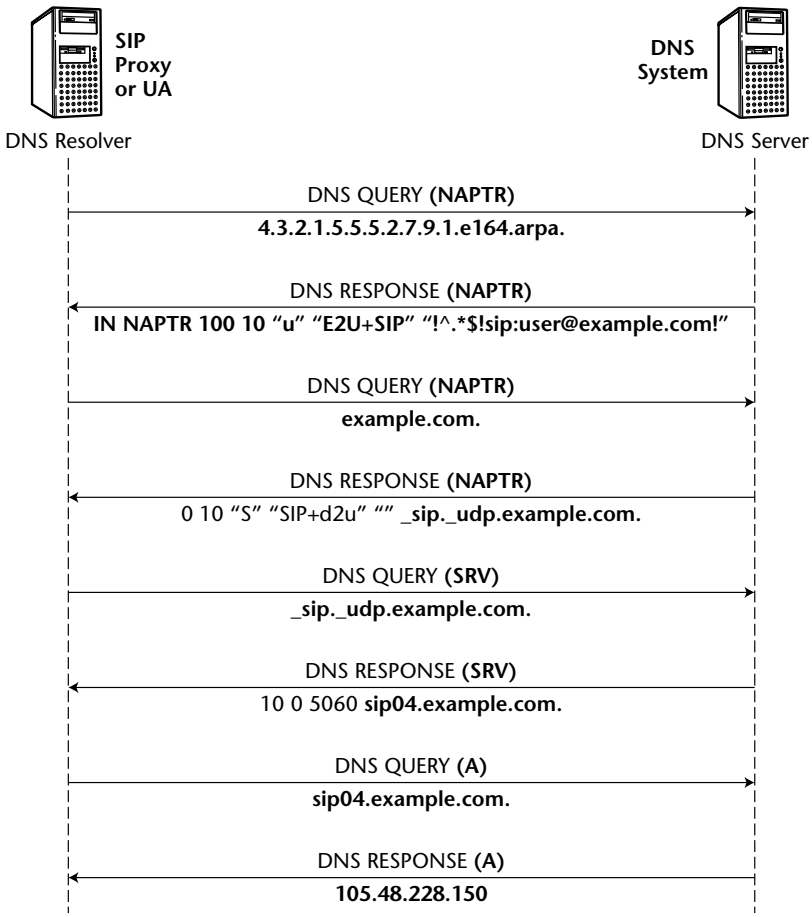


Figure 4.3 DNS and NAPTR service examples

The regular expression in the NAPTR response is a powerful technology to control the search process used in ENUM. It allows you, for example, to search not only for a single phone numbers but also for blocks of phone numbers. For example:

```
^123(.*)567$
```

will search for all numbers starting with 123 and ending with `567`. The following apply:

- "^"—Start of the string
- ".*"—Wildcard
- "$"—End of the string

Blocks of phone numbers may be allocated to specific SIP service providers or to IP PBXs accessible from the Internet. This may have wide-ranging regulatory and business implications for the deployment of ENUM that are, however, beyond the scope of this book.

Call Setup Delay

Note the large number of DNS lookups required (1) for registration and (2) for finding the remote SIP server. The DNS lookup can be an important factor in long call setup delays.

DNS-Based Routing Service Using SIP

There are some problems with personal addresses. With the proliferation of communication services provided by the PSTN, wireless phone systems, and various Internet services, it becomes quite difficult to track the increasing number of addresses for contacts of people we are interested in. The frequent changes of IP addresses by many Internet access service providers make this problem even harder.

Contacting a called party is a difficult problem because of the number of communication devices (such as: home PCs and home faxes, palmtop computers, laptops, office PCs and faxes, pagers, cell phones, IP printers, vehicles, boats, and so on). It is thus possible to require a choice among up to 10 devices or more, to reach the called party.

Users may also want to indicate temporary contact addresses, such as the phone number of a secretary, restaurant, or a hotel when traveling. Besides the contact address, users may also want to indicate the mode by which they are to be contacted. A user attending a meeting may indicate, for example, that text chat is appropriate to convey urgent messages in a nonintrusive manner.

SIP URI or Telephone Number?

A SIP URI in the form of `sip:alice@wonderland.com` is a very powerful single address, since it can support user mobility (see Chapter 15) as well as user preferences.

In the transition period where most users still have PSTN and/or mobile telephone service, another solution is to provide an E.164 phone number as the single contact address on a business card. When a PSTN telephone number includes a country code and conforms to certain rules, it is known as an E.164 number, which refers to the ITU-T document that describes telephone number's structure. Phone numbers, especially the home phone number or the mobile phone number, change less frequently and are, thus, well suited to be stable contact addresses.

DO YOU REALLY HAVE VOIP?

Users may want to check commercial claims for what is a VoIP service. If they get only a PSTN phone number and can call only PSTN phone numbers, that is really a "PSTN over IP" service, since no Internet applications can be used and no Internet user experience is provided. Using IP inside the plumbing of the "PSTN over IP" network of that service provider does not justify the term "VoIP" in our opinion, even though some added features are available that go beyond what the PSTN can support (new features such as global mobility and good quality video phones).

There are two tests to apply for any VoIP service claim:

1. **Does the subscriber get a URI so as to be reachable from the Internet?**
2. **Can the caller "dial" a URI to reach anyone on the Internet?**

Only if the answer is "yes" to both questions is the service true VoIP.

If the service provides only phone numbers, then it would be correctly called "PSTN over IP" instead of VoIP.

These criteria also apply, in our opinion, to the many VoIP solutions marketed to enterprises, especially to the so-called IP PBX that that is voice-centric, without the benefits of presence, IM, video, and the integration of communications and applications. It all comes down to the power of the DNS and URIs being available to consumers and to the enterprise/institution information technology (IT) organization in an integrated fashion, and not having the IT organization maintain a distinctive voice-centric infrastructure.

The predominant availability of phones with 12-digit keypads (10 numerical digits, plus the special characters "#" and "*") makes this the most practical near-term option for address entry.

If there is a desire not to give out one's home phone number, another phone number (such as a work number) could be given as the single contact address. Thus, by knowing just one telephone number, all the other communication addresses can be found.

An example of a contact list for SIP is:

```
Contact: <sip:henry@wcompulver.com
;service=IP,voice mail

;media=>;audio ;duplex="full ;q=0.7"

 ;actor="msg-taker" ;automata ;q=0.7
Contact: phone: +1-972-555-1212; service=ISDN
<tel:+19725551212>;mobility="fixed; ";language="en,es," ;q=0.5
```

```
Contact: phone: <tel:+1-214-555-1212; service=pager
>;mobility="mobile
";duplex=send-only ;media=
 ;text;  ;q=0.1;  ;priority="urgent
" ;description="For emergency only"

Contact: <mailto:henry@mcipulver.com>
```

This contact list for Henry shows the most preferred contact to be voicemail, followed by an ISDN phone number to call, pager, and e-mail in that order.

The callers having only a telephone, PBX, or PSTN, must only dial a single phone number that will be used in the tel URI on the Internet side.

- DNS and ENUM resolution will route the call to the SIP server.
- The SIP server can map the address of record (AOR) to all the Contact addresses of the called party.
- The called party's contact addresses are kept confidential.
- The called party has full control if and where to receive the call.
- If the PBX or the PSTN telephone switch has no support for Internet telephony, the call will be routed over the PSTN, usually at higher cost.

The ENUM Functional Architecture

The ENUM application for DNS presented in the previous sections is quite straightforward for DNS developers and administrators but raises significant issues regarding the registration of users and their telephone phone numbers. The mapping of telephone numbers to IP addresses falls at the intersection of two very different networks and standards bodies, as shown in Table 4.3.

Besides the technical standards, there are huge global industries dependent on the addressing on both the PSTN and the Internet, and the registration of users is big business indeed. For these reasons, the schema for registration of telephone users in the DNS databases has required good collaboration between the IETF and IUT-T and, as of the writing of this book, a reasonable agreement has been reached.

Table 4.3 Networks and Standards Bodies

NETWORK	STANDARDS	ADDRESS ADMINISTRATION
Internet	IETF	ICANN
Fixed & mobile phones	ITU-T, 3GPP/3GPP2	ITU-T

We will describe here the essence of the schema for user registration in ENUM, though various business interests and the resulting political pressures may lead to further evolution of ENUM registration. The ENUM functional architecture discussed here is also know under the name "Public ENUM," and there are other schemas as well, such as "Private ENUM" that may apply to consortiums of telephone companies (where the relevant DNS data is visible only to consortium members).

We will illustrate the public ENUM architecture with an example for North America, as designed by the North American ENUM Corporation with Limited Liability (ENUM LLC). This design has been developed for trials of ENUM in North America [14]. Similar work is being performed in Europe by RIPE in the ENUM WG [15]. ENUM work in the Asia-Pacific region is described in reference [16].

The ENUM functional architecture as seen by the ENUM LCC during the 2005 trial stage is shown in Figure 4.4.

The elements of the ENUM functional architecture are:

- The DNS root "." and the first leaf ".arpa" that is also shown in Figure 4.4.
- The ENUM Tier 0 for the domain "`e164.arpa`". Tier 0 contains the delegations for the telephony country codes and is, therefore, international in scope.
- ENUM Tier 1 that contains the delegations for North America: Canada, the United States, and the Caribbean countries. Tier 1 is split into two layers: Tier 1A for the country code 1 for North America and second layers, such as Tier 1B, which is the trial Tier for the United States. Tier 1 contains the registries for the DNS name servers with ENUM RR for the countries with country code 1. The DNS registry is populated with ENUM RR by the DNS registrar that, in turn, takes its data from the registrants in Tier 2.
- ENUM Tier 2 contains the service providers that, interestingly enough, are called in the ENUM trial documents the application service provider (ASP). The ASP provide applications directly to end users, such as telephony based on SIP, although other applications are also envisioned, as shown in Table 4.1. Tier 2 service providers manage the ENUM domain names associated with an E.164 telephone number or block of numbers, and act as the registrars for the NAPTR RR.

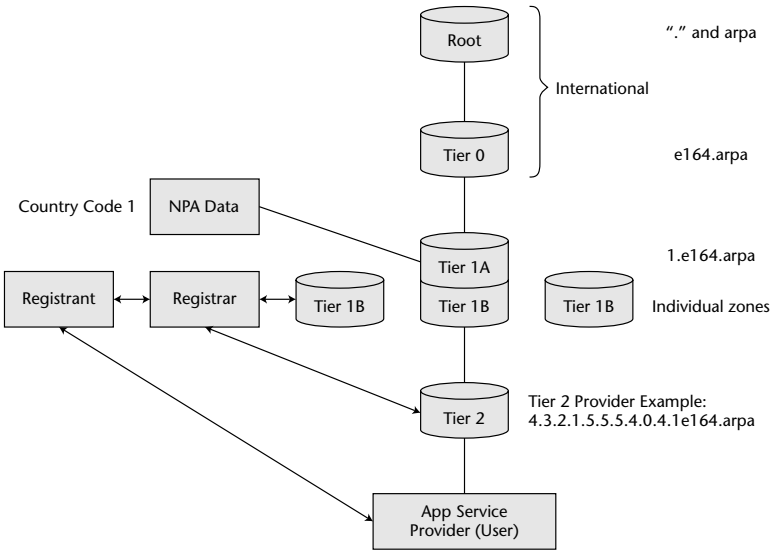


Figure 4.4 The ENUM functional architecture

ENUM and Number Portability

The notion of registering blocks of numbers for service providers is complicated by number portability (NP), which allows end users to keep their telephone numbers if they change service providers or move from one location to another within a calling area. The Tel URI can have extensions to inform the PSTN switch that this is a ported number and where it has been ported. To perform this function, a numbering plan database dip has to be performed as described in [17].

Implementation Issues

We have so far considered the Tier 2 address resolution using the ENUM NAPTR approach. Several considerations have to be taken into account when choosing an implementation alternative:

- *Client complexity*—The NAPTR solution requires more complex regular expression processing and parsing to obtain a result, but a single DNS client is all that is required.

- *Real-time updates*—Present DNS technology may introduce minute-long update delays. Current DNS work aims, however, to reduce the update delay.
- *Provisioning complexity*—The implementation of secure provisioning of NAPTR records required for ENUM is quite complex and beyond the scope of this introduction to SIP.
- *Other considerations*—These may include the storage of supplementary information, such as security data and spoken names (audio files), or the improved flexibility of queries.

More detailed implementation issues go beyond the scope of this book. The IETF, in keeping with its tradition of choosing scalability, simplicity, and state-of-the art technology, together with allowing for only one option, has decided to use DNS NAPTR RR for the Tier 2 address resolution using ENUM.

DNS and SIP User Preferences

Contact addresses for phone, e-mail, and other addresses can also be hosted in the DNS, though we believe this to be a less scalable approach, since user preferences will slow down such servers.

Besides contact preferences, SIP also allows users to specify preferences so as to route incoming calls according to who the caller is, the time of day, the location of the user, and so on. These preferences can be updated or changed in real time by the user.

DNS servers are less suited for frequent real-time updates. There are, at present, no standard facilities to register and authorize users and to provide standard contact data formats as specified in SIP. By contrast, SIP servers can accommodate fast-changing user preferences. SIP adds the following value-added features to ENUM:

- Confidentiality (DNS, by contrast, contains public data)
- User preferences
- Personal, service, and precall mobility (see Chapter 15)
- Frequent and secure access by end users.

These features simply may not make sense for ENUM lookup.

Implementers of SIP clients can use the preceding features to considerably enhance the value and indeed "stickiness" of their products. Feature-rich SIP phones with adequate displays and PC clients, for example, can display the presence information about the called party after being allowed to be a watcher (see Chapter 12), and use it in combination with other applications.

A number of interesting issues for implementing SIP call routing using ENUM for number portability are discussed in [18]. New parameters are proposed for the tel URI to carry number portability (NP) related information that can be passed to the next hop after the NP database dip has been performed

It is important to note that SIP provides the ability of a UA without access to ENUM to place SIP calls. The configuration of a default SIP outgoing proxy server allows an extremely simple UA to simply take a telephone number or URL input from a user and forward the SIP Request to the outgoing proxy. The proxy then performs DNS and ENUM described in this chapter, and the call completes in the same way as if the SIP UA had made the ENUM query.

It is a system design choice if the ENUM resolver is placed in the SIP UA, such as a SIP phone, or in the outgoing SIP proxy. For the sake of end-user control, we believe that well-designed SIP UAs must also have an integrated ENUM resolver as part of the DNS resolver module.

The default SIP proxy can be statically configured in a SIP user agent client (UAC), or automatically configured using Dynamic Host Configuration Protocol (DHCP) at the same time the device is assigned an IP address. Work is in progress for the complete automatic configuration of SIP UA, including with the address of the outgoing SIP proxy [19].

Outgoing calls can be handled by the SIP UAC several ways using ENUM:

1. UAC takes phone number and performs ENUM DNS query to get the URI, or UAC performs DNS SRV query on the domain in the URI to get the IP address.
2. UAC puts phone number in a tel URL and forwards it to a gateway or proxy, or UAC puts phone number in a SIP URL and forwards it to a gateway or proxy.
3. UAC forwards the URL to the default proxy, which performs either steps 1 and 2, or queries a location database. This last option may be preferable, because it relieves SIP devices of DNS transactions required for ENUM and, thus, avoids extra call setup delay.

Application Scenarios for SIP Service Using ENUM

Here, we will provide high-level examples for an end user in an enterprise network (such as a broker in a financial institution) trying the reach a customer who may be accessible on either the PSTN/mobile telephone network, Internet, or paging network.

PBX Enterprise Voice Network

The caller in the enterprise network on the left of Figure 4.5 tries to reach a client with the phone number 214-123-4589. Suppose that there is some urgency and using voicemail or e-mail is not desirable. There are several scenarios:

- The enterprise PBX of the caller has not been upgraded for ENUM service. The called party can only be reached using a phone number. In this case, the call will be routed over the PSTN.
- The PBX has been upgraded for access to ENUM service by using a service provider with an outgoing SIP proxy that is ENUM-enabled. In this case, when dialing the phone number:
  - The called party can be reached at any PSTN/mobile phone, without the calling party having to notice the difference. If the called party has also the benefit of SIP service, the caller may be notified where the call has been redirected. This option is, however, under control of the called party and can be made dependent on who the caller is, location, time of day, and so on.
  - The called party is not available. A voice announcement can inform the caller of the alternative contact options. If paging is an option, the called party can be paged and a IM text message sent by the caller. If both the caller and called party have IM, they can start communicating, even if the called party is on another call.

Enterprise System with IP Communications

The PBX has been replaced with a SIP-based IP multimedia communication system (Figure 4.6). The callee can be reached anywhere on the PSTN or mobile phone network or on the Internet, without the need for other systems, such as a separate paging system. The ENUM system will provide a URI for routing the SIP call to any destination.

The SIP phone user will be connected to the called party for a voice call, if the called party is reachable on any PSTN/mobile phone or by an IP phone.

If the called party is in a meeting and has a laptop computer connected to the Internet, the call will be redirected to the instant messaging client, and both parties can use text chat in a nonintrusive manner. The caller in the enterprise network can also push web pages or transmit other documents to the client during the conversation.

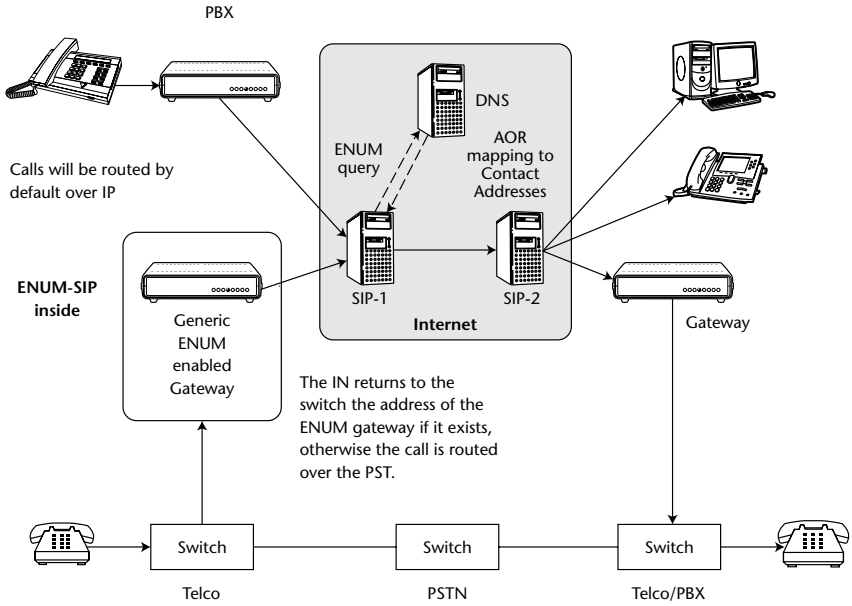


Figure 4.5. Originating calls from a PBX of from the PSTN using ENUM

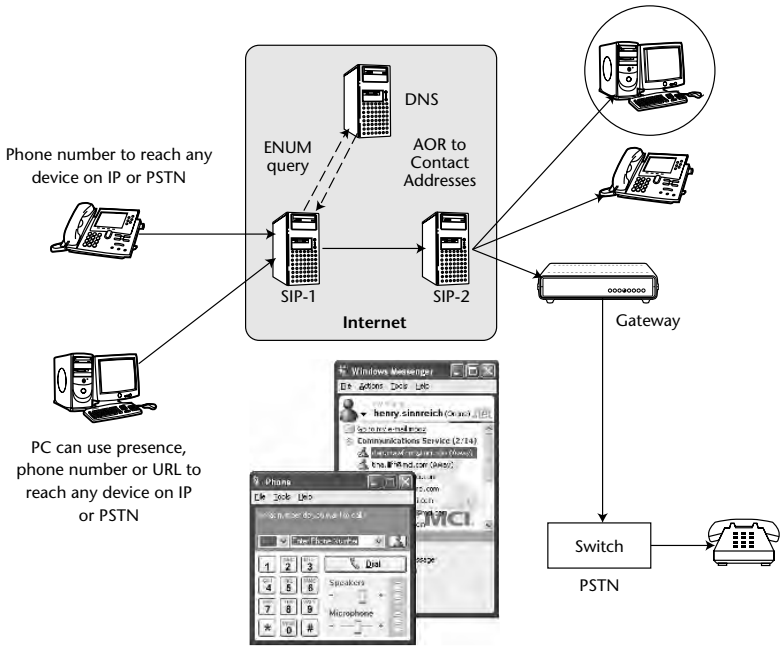


Figure 4.6 Using ENUM for IP-IP communications

We notice here that the caller can also have the benefit of the SIP presence service. In this case, the presence service would not only notify the caller of the availability of the called party anywhere on the PSTN/mobile or Internet, but could also convey additional information about the willingness to accept calls and other information, such as being in a quiet place (meeting) or already in a call, and even who the other party in the conversation might be. Displaying such information is also subject to the preferences of the called party.

All these communication capabilities argue against the use of a voice-centric PBX, be it TDM- or IP-based.

Residential User with ENUM Service

A residential user who wants to print only one residential phone number (for example, the user may have separate phone numbers for the home and office) on a business card has the following options:

- Request ENUM service from the local phone company. If the local telephone company has ENUM service, the incoming call will be routed by the *intelligent network (IN)* control system. The IN will use ENUM and SIP to route the call, depending on the script for SIP called party preferences.
- If the local phone company does not offer ENUM service, users can request to have the telephone service moved to another service provider that offers ENUM service. *Local number portability (LNP)* will ensure the user keeps the same phone number. Incoming phone calls to the user will now be diverted to the other service provider (the ENUM service provider), using the *Local routing number (LRN)* to designate the alternate terminating switch of the ENUM service provider. The ENUM service provider will route the call as described previously.

Notice that if the called PSTN device happens to be outside the local switch calling area (the called user is traveling and wishes to get incoming phone or fax calls directed to some remote place), the ENUM service provider can divert the call from the PSTN network and route the call over the Internet, thus reducing the cost of the call. This operation, known as *PSTN call diversion* will, therefore, also benefit from ENUM.

Miscellaneous: ENUM Lookup of the Display Name

The profitable use of Internet technology in the transition from the PSTN to the Internet is illustrated by many innovative applications, such as using ENUM to map a phone number to a display name that can be rendered by the display on the SIP UA for the benefit of the called party [20]. This application avoids the use of the existing PSTN databases and their associated higher cost for new, IP-based service providers.

DNS and Security

Users of the public DNS data stored in both the Tier 1 and Tier 2 of the ENUM service must be assured that they will receive valid information. Hence, the core underlying security considerations for the DNS and ENUM service focuses on add, change, and delete security at both the first and second levels of the solution.

Clients who have authority to add, change, and delete entries in the ENUM system must be assured that they:

- Are updating data in the correct server and the correct DNS zone
- Have uninterrupted access to the data
- Are allowed to update the data based on presenting valid credentials

Service administrators for both the first and second tiers of the ENUM service have the responsibility to protect their physical and network resources as well as to ensure the validity of the DNS data entered in the system.

Tier 2 of the ENUM architecture needs to have secure communications between the PSTN telephone service provider that owns the phone numbers and its subscribers. If, for example, a phone is disconnected or the number is changed, a secure update must be made in the DNS.

When preparing to prevent security breaches, the following types of attacks must be considered.

Impersonation

Clients attempting to add and update entries in an ENUM service must be able to unequivocally prove their identity to the DNS system. Spoofing or misrepresentation of the identity of the originator of the information could allow unauthorized updates to the database. Invalid or missing data could, in turn, cause malicious redirection and denial of service, which are discussed later. The update facility of each ENUM system is responsible for preventing impersonation attacks.

Eavesdropping

If the privacy of the information that is being transmitted between a client application and the ENUM service (first or second level) is compromised, then registrant-sensitive information such as the registrant's username and password, could be obtained by a malicious intruder. The DNS system must be able to prevent eavesdropping attacks.

Data Tampering

During the transmission of directory records, valid URIs could be replaced by invalid URIs, in turn causing malicious redirection as discussed later. Because a high percentage of security breaches (such as data tampering) can be caused by "insiders," physical and network security must be addressed. The widest range of network and physical security features must protect servers. DNS Security (DNSSEC) [21] may provide integrity and authentication of DNS records in future deployments.

Malicious Redirection

Malicious entries into the database will cause clients to retrieve wrong URIs that point to fraudulent or damaging content. This can be accomplished in two ways: first, by data tampering as discussed previously, and second, through server impersonation whereby a malicious server is masquerading as a valid ENUM server.

Denial of Service

There are several ways that a client could be denied access to the desired network resources, which may include access to the DNS data, as well as access to physical DNS servers.

First, a malicious intruder could remove the URIs from the database, using the data-tampering methods discussed previously, thus making it impossible for the client to access the correct information.

Another way to cause a denial of service to customers is to flood the DNS servers with enough data to prevent further communication with that server. This is done by either downloading gigabytes of information from the server all at once or by maliciously flooding the server with bogus requests.

And finally, by breaching the physical security of the servers by, for example, cutting off electricity to the facility, clients would be denied access.

The security of the DNS responses as they route through the public Internet must be considered. A third party could intercept and modify a DNS SRV record by deleting or modifying URIs.

Extensive work on DNS security has been done, and more work is in progress. Interested readers are referred to the IETF Working Group documents on DNS extensions [22].

- *Example of a secure implementation for a DNS server*—Employs two-factor authentication that requires username and password, as well as a client certificate, utilizing public-key cryptography along with the Secure Socket Layer protocol (SSL). This approach addresses each of the security concerns described earlier. First, it reduces the possibility of impersonation by the parties who are attempting to update the DNS. The

identity of the sender of the data, as well as the identity of the Authoritative Directory Service operator, is ensured using SSL and mutual authentication. Further security of the database is assured by giving users access only to their own data. After the identity of a user is established through two-factor authentication, users cannot change or enter data that does not belong to them. Finally, the use of SSL eliminates the risk of data tampering, as well as providing privacy through encryption of sensitive information.

- *Physical security*—The facilities in which the DNS servers reside are protected by physical security measures, including 24×7 secured access, video camera surveillance, security breach alarms, and secured equipment cabinets. State-of-the-art firewall appliances, VPN equipment, and hardened server operating systems provide network security.

Summary

The Internet Domain Name System is an essential component for Internet communications, since it allows routing e-mails, telephone calls, and other communication requests to the right services and to the desired party.

ENUM service is part of the DNS and is extremely valuable for users of the existing switched telephone systems. ENUM remains valuable even in an end-to-end IP communications environment. Since there will be a long transition time to universal IP communications, if and when this happens, ENUM remains a powerful service that is also application-independent.

DNS and ENUM security are critical to Internet communications and strong security is used to protect the DNS and ENUM data.

References

[1] "Understanding DNS," Microsoft Technet, 2005, `http://technet.microsoft.com/default.aspx`.

[2] "DNS Tutorial at the 63 IETF," `http://edu.ietf.org/node/view/48`.

[3] "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)" by P. Fallstrom and M. Mealing. RFC 3761, IETF, April 2004.

[4] "Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web" by T. Berners-Lee. RFC 1630, IETF, June 1994.

[5] See the IANA home page at `www.iana.org`.

[6] "The mailto URL scheme" by P. Hoffman et al. RFC 2368. IETF, July 1998.

[7] "Uniform Resource Locators" by T. Berners-Lee et al. RFC 1738, IETF, December 1994. This RFC has the level of a proposed standard, and there are several other RFCs with updates.
[8] "The tel URI for Telephone Numbers" by H. Schulzrinne. RFC 3966. IETF, December 2004.
[9] "SIP: Session Initiation Protocol" by J. Rosenberg et al. RFC 3261, IETF, June 2002.
[10] "Domain Names—Concepts and Facilities" by P. Mockapetris. RFC 1034, IETF, November 1987.
[11] "Use of DNS Aliases for Network Services" by M. Hamilton and R. Wright. RFC 2219, IETF, October 1997.
[12] "A DNS RR for specifying the location of services (DNS SRV)" by A. Gulbrandsen et al. RFC 2782, IETF, February 2000.
[13] "Session Initiation Protocol (SIP): Locating SIP Servers" by J. Rosenberg and H. Schulzrinne. RFC 3263, IETF, June 2002.
[14] The Web site for the North American ENUM trial is `http://enumllc.com`.
[15] The Web site for the European ENUM working group is `www.ripe.net/ripe/wg`.
[16] `www.apenum.org/archive/APEET-ENUM-BoF-KL-04-Collobration.pdf`.
[17] "New Parameters for the "tel" URI to Support Number Portability" by J. Yu, Internet draft, IETF, November 2004, work in progress.
[18] "New Parameters for the 'tel' URI to Support Number Portability" by J. Yu. Internet Draft, work in progress, IETF, July 2005.
[19] "A Framework for Session Initiation Protocol User Agent Profile Delivery" by D. Petri. Internet Draft, work in progress. IETF, July 2005.
[20] "IANA Registration for an Enumservice and 'tel' Parameter for Calling Name Delivery (CNAM) Information" by R. Shockey and J. Livingood. Internet Draft, IETF, January 2006.
[21] "DNS Security Introduction and Requirements" by R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, RFC 4033, March 2005.
[22] The IETF WG on DNS extensions, see `http://ietf.org/html.charters/dnsext-charter.html`.

CHAPTER 5

Real-Time Internet Multimedia

This chapter provides an overview of the Internet protocol architecture for real-time multimedia. Other key Internet transport and multimedia protocols besides SIP, such as IP unicast and multicast transport protocols, RTP/RTCP, SDP, and RTSP, are briefly introduced.

Introduction

Though the Internet was not created primarily for real-type communications, and its initial growth has been driven mainly by file transfer, e-mail, data, and the World Wide Web, multimedia on the Internet has also seen tremendous growth for various applications, including (but not limited) to telephony. Indeed, many online magazines routinely carry links to streaming audio or streaming audio/video news clips, movie trailers, or online tutorials and presentations. Video services on the Internet are emerging strongly. There are an ever-increasing number of radio and video stations worldwide on the Internet, rivaling the number of stations on shortwave radio. The Internet has shown its capability of:

- Consolidating all types of media and data on one single network
- Integrating all services at the application layer for information, communications, entertainment, and transactions

- Scaling from point-to-point voice calls to conferences and network broadcasts encompassing millions of users
- Empowering end users to choose both applications and content on a global basis
- Revolutionizing the software industry by forcing the redevelopment of practically all software applications so that they are Internet-centric and the providing of software such as office productivity applications that are communication-aware (as is the case for the Microsoft Office suite)

For any type of communications and media, these features of the Internet are leading to a migration to the Internet of both the telecommunication services (such as telephony) and broadcast services (such as TV and radio). We share the belief found in the Internet community that the Internet will lead to services and social structures that do not exist today, similar to e-commerce that has emerged with profound implications for all commerce.

Such an impact, however, does not come without a price, and as is very appropriate for the Internet, the price is not primarily in a new physical infrastructure (huge in itself by any measure) but in what we believe to be a large and ever-expanding mandatory knowledge base and skill sets.

We provide here a short overview of the main protocols required for Internet multimedia and conferencing.

An overview of the Internet protocol stack for multimedia [1] is shown in Figure 5.1.

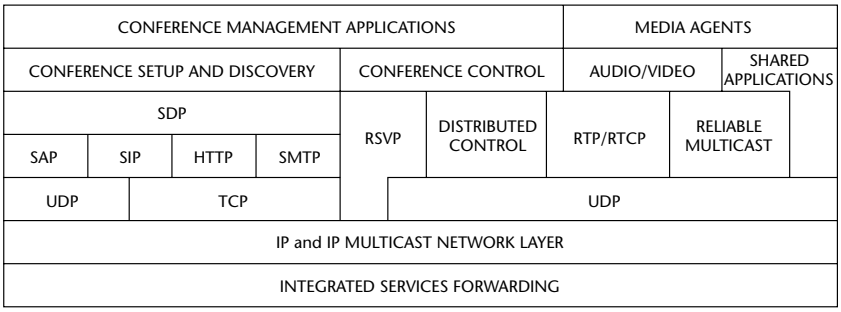


Figure 5.1 Internet multimedia protocol stack

Freshening Up on IP

Though most readers are fairly familiar with IP, those interested in the original work may want to review RFC 791, published in 1981. In RFC 791, Jon Postel expresses, in standard form, the concepts of interworking and IP, first introduced by Vint Cerf and Robert Kahn in 1974 [2].

An excellent summary of recent items of interest for IP, most notably IP address allocation of the 32-bit IP version 4 address space is provided in RFC 2102 [3].

IP multicasting has always played a large part in the Internet concepts on conferencing and multimedia, although its deployment has been rather sparse up to the present. A good overview on IP multicast applications is given in RFC 3170 [4].

The relevant protocols for Internet multimedia and conferencing are summarized in Table 5.1. We will provide in the following discussion a short list of the main topics for Internet multimedia and refer the reader to Table 5.1 for the applicable protocols. The headers in Table 5.1 refer to the protocols grouped under the respective cross-header.

Table 5.1 Network Protocols for Internet Multimedia and Conferencing

NAME	DOCUMENT	SUBJECT
IP Unicast		
Internet Protocol	RFC 791	DARPA Internet Protocol
IP Policies	RFC 2008	IP Address Allocation Policies
IP Multicast Protocols		
SSM	RFC 3569	Overview of Source Specific Multicast
IGMP version 2 Protocol	RFC 2236	Internet Group Management
CBT version 2	RFC 2189	Core Based Tree Multicast Routing
PIM-DM	RFC 3973	Protocol Independent Multicast-Dense Mode
Multicast Address Allocation		
MADCAP	RFC 2907	MC Addressing Dynamic Client Allocation
MASC	RFC 2909	The Multicast Address-Set Claim Protocol
BGMP	RFC 3913	Border Gateway Multicast Protocol

(continued)

Table 5.1 *(continued)*

NAME	DOCUMENT	SUBJECT
Differentiated Services		
DiffServ Field	RFC 2474	Definition of the DiffServ Field in IP Header
DiffServ Arch	RFC 2475	An Architecture for Differentiated Services
Resource Reservation		
IETF Integrated Services and Resource Reservation	RFC 2205	Resource Reservation Protocol (RSVP)
	RFC 2210	IETF Integrated Services using RSVP
	RFC 2211	Controlled Load Network Element for RSVP
	RFC 2212	Guaranteed Quality of Service for RSVP
Data Formats for Real Time Communications (`w3c.org`)		
XML	XML Schema	Extensible Markup Language
VoiceXML	VoiceXML 1.0	Voice eXtensible Markup Language
SMIL	SMIL 2.1	Synchronized Multimedia Integration Language
Multimedia Server Playback Control		
RTSP	RFC 2326	Real-Time Streaming Protocol
Media Transport and Codec Profiles		
RTP, RTCP	RFC 3550	Transport Protocol for Real-Time Apps
RTP AV Profile	RFC3551	RTP Profiles for A/V Conferences
RTP Payloads for Video	RFC 2032, RFC 2435	RTP payloads video codecs. There are many other A/V payloads documented in the IETF.
Session Description		
SDP	RFC 2327	Session Description Protocol

Table 5.1 *(continued)*

NAME	DOCUMENT	SUBJECT
Session Announcement		
SAP	RFC 2974	Session Announcement Protocol
Session Invitation		
SIP	RFC 3261	Session Initiation Protocol
Security Mechanisms for the Internet		
Security	RFC 3631	Internet Security Threats and Protection

Multicast Protocols

IP multicast is not yet implemented in most public IP networks, but it is an important concept to discuss with regard to Internet multimedia. The Internet multimedia conferencing architecture [5] was initially developed on experimental Internet multicast networks in the early 1990s. The initial application was the audiovisual transmission of IETF meetings to online attendees around the world, an application that is continued without interruption to the present.

IP multicast is the most efficient procedure to distribute data and multimedia to large groups of users by locating the distribution function in the IP network layer and, thus, making it available to any type of application. IP multicast also absolves applications from establishing any type of communications between senders and receivers, since joining a multicast group is all that is required. The IP multicast address is the unique identifier for senders and receivers to join a multicast session. Multicast is also highly scalable, since the data replication is delegated to the edge of the network as required by traffic patterns.

Multicast Address Allocation

Multicast addresses are allocated from a pool of Class D IP addresses that are reserved for multicast. Most multicast address allocations are implemented at present in a static manner with manual configuration. However, work is in progress to dynamically allocate multicast group addresses and to provide directory services for multicast groups.

We will use Table 5.1 as guide to provide a short overview on Internet multimedia and real-time communications.

IP UNICAST AND MULTICAST

In unicast IP packet forwarding, a packet stream is delivered to a single destination. Multicast IP delivers a packet stream to a set of destinations. Note that this is different from packet broadcast, which can be done on an Ethernet LAN. Packet broadcast delivers the packet to every destination on the LAN. As a result, packet broadcast does not scale outside the LAN, because it can generate huge traffic loads. Multicast IP is different in that an endpoint must join a multicast group before any of the multicast packets will be forwarded by routers serving that user. This multicast group is identified by a multicast IP address. This scalable architecture limits the distribution of multicast packets only to users that are participating in the session.

IP multicast is generally not enabled on the public Internet. However, it is available using an overlay network called the MBONE (Multicast Backbone). As mentioned, some audiovisual IETF conference sessions were distributed over the MBONE [6] starting in 1992 using multicast.

Application-Level Multicast

IP network-level multicast has not found a wide deployment on the Internet for a number of technical and commercial reasons. To satisfy the need for multicast, various other techniques have emerged, such as:

- The distribution of applications on servers, such as reviewed in [7], [8].
- Application-level multicast on overlay networks [9], [10], and content addressable networks (CAN) [11]. Overlay networks for real-time communications are discussed in Chapter 20, "Peer-to-Peer SIP."

Application-level multicast has not yet applied for real-time communications, and the technology is still in the research stage. For this reason, we will not present it here, though we believe application-level multicast will significantly disrupt the content distribution industry.

Transport Protocols

Media streams (such as voice and video for real-time communications) use UDP packet transport, since it makes no sense to wait for delayed media packets or for the retransmission for packets that were lost, as is done using TCP. Lost media packets are discarded in favor of getting the shortest delivery time possible. Media delivery using UDP over IP is sensitive to packet delay and packet loss. Quality of IP service is, therefore, an important part for real-time Internet multimedia communications.

As we will show in Chapter 18, "Quality of Service for Real-Time Internet Communications," packet loss on the Internet has been reduced in the last 10

years to the 0.1 percent to 1.0 percent range, and the delay is close to the speed of light. We believe, therefore, the Internet "as is" to be adequate for high-quality real-time communications, as long as congestion in the access networks is avoided.

Besides delay and packet loss that are relevant to transmission impairments, occasional *route flapping* on the Internet can also impair real-time communications, though proper router maintenance procedures can reduce route flapping, since human interventions for maintenance are quoted to produce 80 percent of the network problems. BGP route flap damping is described in [12].

IP Network Layer Services

Though the terms "Internet service" or "IP service" have many marketing connotations, IP-level services in the technical sense really refer to the level of quality of service (QoS) provided. Figure 5.2 shows the spectrum of IP services.

Most Internet and IP users are familiar only with the best-effort type of service. Best-effort service is shown at the far left of the spectrum in Figure 5.2. Best-effort service can provide adequate QoS for interactive communications as long as there is no traffic congestion on any of the links between the respective endpoints. Best-effort IP service, however, cannot provide assurance that QoS will be maintained at all times during a session, since congestion may affect packet delay and packet loss in an unpredictable manner for media packets. Rare route flapping events as mentioned previously can also be a source of impairments.

Readers should not assume that best-effort service might not be adequate for IP communications. Daily use of public VoIP services and of SIP telephones on the public Internet by the authors have convinced us that, with adequate broadband access (such as cable or DSL for home use), best-effort service provides high-quality telephony equal or better than that on the PSTN, though no guarantees can be provided. Emerging fiber to the home and broadband wireless will also support better than PSTN voice services.

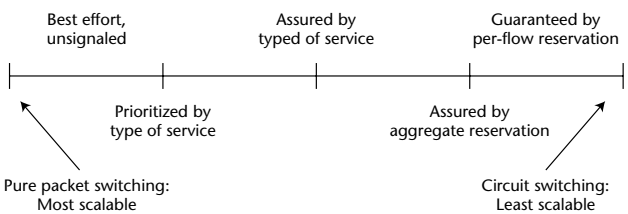


Figure 5.2 Range of IP and Internet services

The description here of the Internet services from a QoS perspective will be useful for reading Chapter 18, "Quality of Service for Real-Time Internet Communications."

Differentiated Services

The *Differentiated Services (DS)* [13] model allows only certain classes of service that are defined in the network. The main properties of differentiated services are therefore:

- DS require no state in the network.
- Applications can be fit only within certain classes of service.
- The network is not aware of the individual IP packet streams, but only of classes of service. Accounting for individual users is therefore not possible.
- DS are highly scalable and, therefore, well suited to be used in the core of the Internet.
- Because of their simplicity, DS are also most useful in Internet access networks.

Resource Reservation

At the other end of the IP services spectrum from best-effort service is the guaranteed-by-per-flow reservation service based on the Resource Reservation Protocol (RSVP) [14], shown at the right in Figure 5.2. RSVP resembles the bandwidth and delay qualities of TDM circuits, either for guaranteed service or to the degree to which TDM circuit properties can be emulated over an unloaded IP network. RSVP reserves the resources across the IP network associated with individual flows. The IP addresses and port numbers of the IP endpoints, and also the transport layer protocol (such as UDP or TCP) characterize an IP packet flow. RSVP is a form of virtual circuit setup over a packet switched network.

Following are the main properties of RSVP:

- Applications in endpoints can communicate their requirements for QoS to the network directly and in a flexible manner.
- The ownership of the RSVP-supported QoS flows can be clearly distinguished. This allows accounting for the use of network resources by individual users.
- Routers in the network have to keep state for each RSVP reservation, and, as a consequence, RSVP is not scalable to large networks. The use of RSVP is, therefore, practical in the Internet access part, in the periphery of the network, or in private IP networks of limited size.

Two types of services have been standardized for RSVP: *Controlled load*, (which offers service with QoS equal to that of the unloaded network), and *Guaranteed* (with hard QoS limits).

An industry-wide effort in the Integrated Services over Specific Link Layers (ISSLL) working group of the IETF has produced detailed specifications on how to map IP QoS in the integrated services architecture onto most link-layer technologies (such as slow links, Point-to-Point Protocol (PPP), Ethernet 802.3-style LANs, and legacy technologies, such as Frame Relay and ATM).

Integrated Services and DiffServ Networks

Using differentiated services at the edge of the network in the access portion and differentiated services in the core is a good match for end-to-end QoS.

Several intermediate QoS models are possible between the extremes of RSVP and differentiated services, as shown in Figure 5.2. For example, so-called RSVP aggregators can have another, aggregated RSVP as the output, or certain classes of services can be associated with guaranteed delivery. The latter raises the intriguing possibility of a standard class of service for voice across the Internet and private IP networks. Since voice seems to require only a small part of the overall bandwidth compared to data, setting a standard QoS for telephony might simplify considerably end-to-end IP network design.

Multiprotocol Label Switching

Multiprotocol Label Switching (MPLS) is a controversial protocol and an often mentioned reason for its deployment is VoIP. Recent marketing push by vendors and traditional carriers for MPLS [15], and the resulting standards activities require, therefore, some clarifications here.

On the positive side, MPLS may be useful in the internal plumbing of large Internet service providers for *traffic engineering* [16], for steering traffic along certain paths (for example, between a north and a south route crossing the Atlantic Ocean).

On the negative side, MPLS has been pushed to market not necessarily for the benefit of Internet users, but as a revenue enhancer for equipment vendors and their traditional carrier customers. Facility-based Internet domain owners may also see MPLS as a tool to discriminate against emerging VoIP services provided over the open Internet.

This is not an advantage for users who may want to reach anyone one the Internet.

Other MPLS issues include the following:

- MPLS does not provide QoS, but can only invoke DS in networks where QoS using DS is already implemented.

- MPLS is only applicable in the core of the network where no QoS services are required due to the ample bandwidth, but is not applicable on congestion-prone access links where most QoS problems arise. MPLS is also of no help for occasional route flapping events in the Internet.
- MPLS does not necessarily enhance security. MPLS can provide certain isolation for networks using MPLS for interconnection, but this isolation does not protect such networks from security threats at the application level, where most security threats reside. MPLS can actually be a vulnerability, since it requires central control that can be targeted for attack.
- The scalability of MPLS is limited by the capability of the network equipment to process large numbers of MPLS paths across the network. This is in contrast to the scalability of the Internet that has no known limits.
- MPLS resembles ATM (it can actually be considered a reincarnation of ATM) and carries the disadvantages of legacy telecom into the IP realm:
  - Central control
  - MPLS paths resemble circuit switching

Facility-based domain owners may certainly establish MPLS paths for peering for VoIP, but this has not happened at present for reasons explained in Chapter 18, "Quality of Service for Real-Time Internet Communications."

Media and Data Formats

The predominant media types for conferencing are text, audio, and video, although other real-time media application such as games may also be considered a conference session. The XML-based documents for presence information are also quite large and can be a source of significant traffic, especially for frugal mobile networks. Various types of data also are exchanged during a conference (such as Web pages and desktop applications). The protocol for media transport is the Real-Time Transport Protocol (RTP). In addition, during a conference, presentations are made. The protocol for synchronizing various media and data types during a presentation is the Synchronized Multimedia Integration Language (SMIL—pronounced like "smile"). Finally, recorded media streams of complete presentations can be uploaded, downloaded, and replayed using Real-Time Streaming Protocol (RTSP).

Media Transport Using RTP

The RTP document, RFC 3550 [17] consists of the following:

- RTP for media packet transport
- The RTCP to monitor the quality of service and generate reports to the network

RTP uses UDP for transport over IP.

A complete treatment of RTP would require a book by itself. RTP has many capabilities, including minimal control for multimedia conferences.

We will provide here in a nutshell only the most relevant aspects required to understand the environment for SIP. We refer readers to the Web site of the IETF AVT working group (WG) at `http://ietf.org/html.charters/avt-charter.html` for more information.

Audio and video packets are encapsulated in RTP packets that provide the following information carried in the RTP header:

- *Packet sequence number*—Allows the user to reorder packets on arrival and to detect the loss of packets.
- *Timestamps*—Allows jitter to be detected (that is, packet arrival time variations across the network).
- *Synchronization (media) source*—Allows the identification of the sources of the packet streams (such as specific microphones or specific video cameras).
- *Contributing media source*—Allows the identification of a specific media source from several others that have been mixed together (see the following discussion), for example in a centralized multiparty conference.

It is important to realize that RTP is an application layer protocol and does not provide any QoS guarantees at all. However, it does allow transmission impairments such as packet loss or jitter to be detected.

RTCP uses data at the receiver to convey back to senders in the network that monitor QoS to perform fault diagnosis, and report long-term statistic data. Information conveyed by RTCP includes the following:

- The Network Time Protocol (NTP) timestamps can be used to assess absolute round-trip delay.
- The RTP timestamps can be used in conjunction with NTP timestamps, (for example, to assess the local RTP clock rate).
- Synchronization (media) source identifier (SSRC).
- Packet and byte counts.

- Lost packets reported as a fraction of the total and as a cumulative number.
- Highest sequence number received.
- Inter-arrival jitter and other parameters.

Listening-only participants will send Receiver Reports (RR) to applications that monitor the quality of service, while speaking participants will also send Sender Reports (SR).

A special Source Description RTCP packet (SDES) conveys information about the user:

- Canonical Name (CNAME) to identify the participants in the conference
- Username (such "John Doe, Bit Recycler, Megacorp")
- Phone number
- Geographic user location
- Name of the application that is using RTP/RTCP

RTP also uses protocol-specific devices such as translators and mixers. An *RTP translator* connects two different transport networks, such as IP v.4 and IP v.6 networks. Translators also may change the media encoding as required, allowing two endpoints that have no common codecs to be able to communicate.

An *RTP mixer* receives media streams from several sources, combines them, and forwards the combined stream. An RTP mixer will add its own SSRC identifier to the existing identifiers in the component streams.

RTP Payloads and Payload Format Specifications

The Audio/Visual Profile (AVP) for RTP [18] specifies payloads registered with the Internet Assigned Numbers Authority (IANA) and specifies such items as the name, clock rate, or frame size of audio codecs and encoding independent parameters (such as the audio left, right, center, surround, front, and rear).

RTP payloads are grouped for specific applications (such as for audio/video conferencing). Payload types specify specific codecs, such as for MPEG-4 streams, DV format video, or Enhanced Variable Rate Codec (EVRC) Speech [19].

The highly structured and open approach of the RTP payload and format specifications has led to a rich portfolio of standard payloads for the most-used audio and video codecs.

RTP also allows dynamic payloads, which are defined at the initiation of a session.

Multimedia Server Recording and Playback Control

The Real-Time Streaming Protocol [20] is an application level protocol for the control of the delivery of data with real-time properties (such as audio/visual media using RTP). Readers may think of recording video sessions and replaying them over the Internet with playback controls, such as found in consumer sound and video players. The protocol is similar in syntax and operation to HTTP/1.1, so that extension mechanisms to HTTP also can be added, in most cases, to RTSP. However, RTSP differs in a number of aspects, such as the following:

- RTSP introduces new methods.
- RTSP servers maintain state, contrary to Web servers.
- Data is carried out-of-band, such as in RTP packets.
- RTSP has the notion of the request URI pointing to the desired service.

RTSP is similar in many ways to SIP in its approach to protocol design. Also in common with SIP, RTSP uses web security mechanisms and can use different transport mechanisms such as UDP and TCP.

Session Description

The Session Description Protocol (SDP) [21] is rather a session description format than a protocol. SDP is also a quite complex topic because of the many capabilities and the issues related to NAT traversal. The description of the session parameters is used by SIP Internet multimedia and conferencing for session initiation. SDP will be covered in more detail in Chapter 6, "SIP Overview."

Session Announcements

The Session Announcement Protocol (SAP) [22] is a multicast session announcement protocol. SAP advertises multicast sessions by stating the specific multicast address and time information, and it carries a payload that describes the session. In some ways, SAP is analogous to the *TV Guide* where information about the channel, time, and program is provided.

Session Invitation

The Session Initiation Protocol (SIP) is the subject of this entire book.

Authentication and Key Distribution

Messages used by the protocols in the Internet multimedia conferencing architecture can be signed and encrypted using S/MIME [23]. See also Chapter 9, "SIP Security."

The distribution and management of public and private cryptographic keys for real time communications will be treated in more detail in Chapter 9, "SIP Security." A good overview on this topic can be found in several RFCs, such as in [24].

Summary

The architecture and protocols of the Internet multimedia conferencing architecture have been discussed in this chapter. Internet multimedia leverages the entire suite of protocols that include network protocols and application-level protocols for real-time Internet multimedia communications.

Internet multimedia makes use of various network and transport layer protocols, such as IP multicast and protocols for quality of service such as Differentiated Services. RSVP may be used for QoS in private IP networks of limited size, while MPLS is a controversial protocol when it comes to interdomain communications.

The family of Internet application layer protocols: RTP/RTCP, SAP, SDP, SIP, and others were developed for multimedia in this architecture.

References

[1] "Internetworking Multimedia" by Jon Crowcroft, Mark Handley, Ian Wakeman. Morgan Kaufmann Publishers; ISBN: 1558605843, New York, 1999.

[2] "A Protocol for Packet Network Intercommunication" by V. Cerf and R. Kahn. IEEE Transactions for Communications, Vol. Com-22, May 1974. This paper is available online at several Web sites, see for example `www.cse.ucsc.edu/research/ccrg/CMPE252/Papers/1974.pdf`.

[3] "IPv4 Address Behavior Today" by B. Carpenter et al. RFC 2101. IETF, Feb 1997.

[4] "IP Multicast Applications: Challenges and Solutions" by B. Quinn and K. Almeroth. RFC 3170. IETF, September 2002.

[5] "The Internet Multimedia Conferencing Architecture" by M. Handley et al. Internet-Draft, February 1996.

[6] "MBone: Multicasting Tomorrow's Internet" by K. Savetz et al. IDG, 1998. The text of this book is available online at `www.savetz.com/mbone`.

[7] "An Analysis of Live Streaming Workloads on the Internet" by K. Sripanidkulchai, B. Maggs, and H. Zhang. Proceedings of the Internet Measurement Conference 2004 (IMC), Taormina, Sicily, Italy, October 2004. www.akamai.com.

[8] "What Is Web 2.0?" by T. O'Riley, September 2005. www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-Web-20.html.

[9] "Scribe: A large-scale and decentralized application level multicast infrastructure" by M. Castro et al. *IEEE Journal on Selected Areas in Communications*, October 2002.

[10] "An evaluation of Scalable Application-level Multicast Built Using Peer-to-peer Overlays" by M. Castro et al. IEEE Infocom 2003.

[11] "Application Level Multicast using Content-Addressable Networks" by S. Ratnasamy et al. University of California, Berkeley, 2001. http://berkeley.intel-research.net/sylvia/can-mcast.pdf.

[12] "BGP Route Flap Damping" by C. Villamizar et al. RFC 2439. IETF, Nov. 1998.

[13] "New Terminology and Clarifications for Diffserv" by D. Grossman. RFC 3260. IETF, April 2002.

[14] "The Use of RSVP with IETF Integrated Services" by J. Wroklawski. RFC 2210, IETF, September 1997.

[15] "Multiprotocol Label Switching Architecture" by E. Rosen et al. RFC 3031, IETF, January 2001.

[16] "Applicability Statement for Traffic Engineering with MPLS" by J. Boyle et al. RFC 3346, IETF, August 2002.

[17] "RTP: A Transport Protocol for Real-Time Applications" by H. Schulzrinne et al. RFC 3550, IETF, July 2003.

[18] "RTP Profile for Audio and Video Conferences with Minimal Control" by H. Schulzrinne et al. IETF, July 2003.

[19] "RTP Payload Format for Enhanced Variable Rate Codecs (EVRC) and Selectable Mode Vocoders (SMV)" by A. Li. RFC 3558, IETF, July 2003.

[20] "Real Time Streaming Protocol (RTSP)" by H. Schulzrinne et al. RFC 2326, IETF, April 1998.

[21] "SDP: Session Description Protocol" by M. Handley and V. Jacobson. RFC 2327, IETF, April 1998.

[22] "Session Announcement Protocol" by M. Handley et al. RFC 2974, IETF, October 2000.

[23] "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification" by Ramsdell, B. RFC 3851, July 2004.

[24] "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)"by C. Adams et al. RFC 4210, IETF, September 2005.

CHAPTER 6

SIP Overview

In this chapter, an overview of the operation of the SIP protocol will be given, followed by a discussion of the basic functions of the SIP protocol. Example call flow diagrams are used throughout to illustrate the protocol. The references at the end make a reading list for the details of the protocol.

What Makes SIP Special

We will try here to provide a summary for readers who have no special interest in protocols, why SIP has the capabilities to redefine communications, as they migrate from the telephone network to the Internet. As you will see, SIP combines the features of the Advanced Intelligent Network (AIN) from the telecom world for fixed and mobile telephony, with Internet features for e-mail, web, transactions, and entertainment. To illustrate SIP concepts, we will introduce the notion of a SIP-enabled IP network.

SIP Enabled Network

Figure 6.1 shows the elements of a SIP-enabled IP communication network. The network is composed of the following:

- *SIP endpoints, such as phones, gateways, and various types of computers*—SIP endpoints are fully qualified Internet hosts, as defined in RFCs 1122 and 1123. Internet hosts are very different from telecommunication devices, such as phones, fax machines, or mobile phones in the sense that (1) they may use the services of any other host on the IP network and (2) they may run any and all applications the user may desire. The user can direct communications via any service provider and can load any application, similar to other services on the Internet. Depending on the service and user preferences, most communication services can also be controlled from end to end without support from the network for call setup. There are two types of SIP endpoints:
  - User devices, such as phones and personal computers.
  - Gateways to other networks, especially IP telephony gateways that use CAS, Q.931 or SS7 signaling. Other gateways can connect to H.323 or other legacy VoIP networks and device control networks, such as found in certain IP PBXs and so-called *softswitches*, using MGCP, MEGACO, or H.248 master-slave protocols.
- *SIP servers*—Most users have no desire to understand and manage the services they use, and there are also security and technical reasons to place services on dedicated servers in the network, where they can be accessed from anywhere and used with various communication devices. SIP servers accomplish the functions found in the telecom AIN, in e-mail systems, and in web servers, as well as new functions, specific to SIP. SIP servers can be stateless, similar to other Internet devices. SIP servers can be deployed in geographically distributed clusters to avoid service failures. All this ensures very fast response time and avoids failures in the network to disable calls, since the call state is kept at the periphery of the network and not in the core. Users do not depend on any potential central points of failure in the network and can communicate as long as they have working end devices.

A caller can send an `INVITE` message to establish a session to the called party, without knowing exactly where the other endpoint may be, and the SIP servers will route the call to the destination. The route to the destination can be forked in the network so as to find the other endpoint. The same infrastructure can also serve for an instant message and presence service. A watcher can subscribe to a presentity and receive `NOTIFY` messages from the presentity. The watcher and presentity can exchange short text messages using SIP itself, or RTP packets for any other communication media: audio, various data applications, video, or games for instant communications.

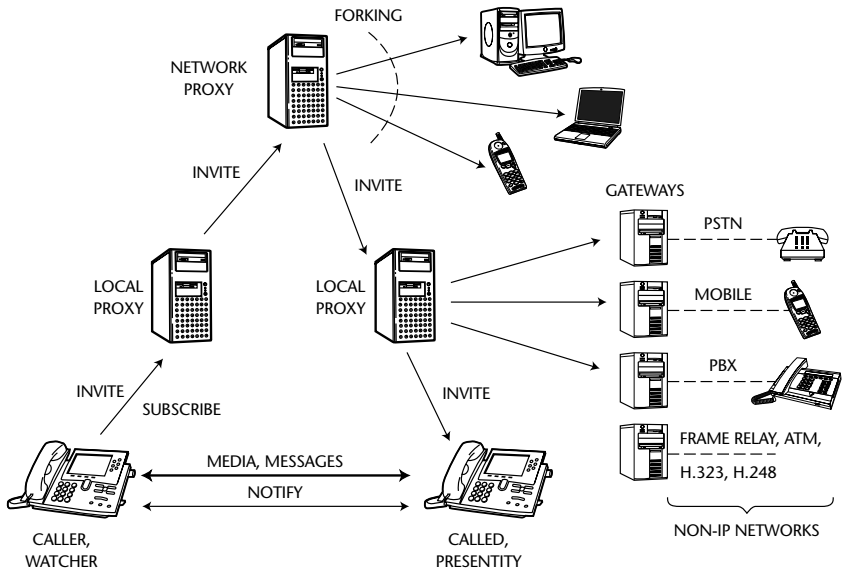


Figure 6.1 SIP-enabled IP communication network

Endpoints and servers benefit from a long list of protocol features of SIP:

- *Web-style and telephony-type addressing*—SIP devices can use URIs that are location-independent and URLs that point to a specific host. Addresses can take the form of e-mail addresses or telephone numbers, with clearly defined options for E.164 public telephone numbers and private numbering plans.
- *Registration*—Devices connected to the network are registered so as to route calls to and from the device. Users may register themselves using their ID to get access to their particular information and services, independent from the device registration. This is similar to e-mail access from web kiosks or Internet cafes. Such dynamic routing to/from the user is accomplished without needing "switch translations" or other static routing tables to be managed.
- *Security*—SIP is designed to use the Internet security mechanisms protect sensitive signaling information from various types of attack. User location and traffic patterns can be kept confidential. SIP security can be quite complex and uses the advances in all generic IP security mechanisms.
- *Redirect*—A SIP server can redirect a request to another address, similar to the core function of the AIN.

- *Proxy*—SIP proxy servers will forward the request of the user to another server that can provide the requested service, such as voicemail, conferencing, or presence information.
- *Forking*—A request from a user can be forwarded in several directions simultaneously, as, for example, when trying various locations where the called party may be found.
- *Rendezvous and presence*—The active form of rendezvous consists of routing a request for call setup to another server or endpoint where the desired service may be performed, such as communication with an individual, or with a machine. The passive form of rendezvous consists of presence information (that is, letting someone know that a party of interest is connected to the network and its communication state, such as available or busy).
- *Mobility*—Users may have many communication devices such as phones, fax machines, computers, palm computers, and pagers, at home, at work, and while traveling. User devices can be attached to various types of networks, if proper gateways are provided: IP, PSTN, mobile telephony, wireless mobile data, or paging. SIP call setup can proceed without regard to the type of network or type of device the parties may use at a certain instance.
- *User preferences*—Callers can specify how servers and the network should handle their requests, and also specify what type of service is desired or acceptable, whom they would like to reach, and whom they would like to avoid (for example, to avoid making calls to busy lines or to speak to machines). Called parties can specify how to handle incoming calls, depending on a very large set of criteria, such as who the caller is, from where the call is coming from, time of day, the communication device, and others.
- *Routing control*—The route taken by SIP messages can be specified and recorded for various services.

Some (but not all) of the preceding features are known from the AIN and other from e-mail and the web. It is the combination of all these features that makes SIP unique. Last, but not least, SIP has unique features of interest to developers and service providers, features not available in telecommunication networks.

The similarity of SIP to HTTP facilitates easy service creation by a very large community of software developers that may be familiar with web site development.

SIP is text-based and easy to debug without using specialized test equipment. SIP messages are seen on standard data analyzers in the very form shown in this introduction.

Watching How Sausages Are Being Made

We believe the style in which SIP has been developed is another feature, equal in importance to what gets actually written in the standard.

- *Open*—The development of SIP in the IETF mirrors many other developments that have contributed to the success of the Internet. The SIP protocol development is a completely open process, where everyone (from anywhere) can follow the online postings and discussions, and make technical contributions. There are no fees for participating, except a moderate attendance fee to cover the cost of IETF meeting logistics. Nothing in SIP involves intellectual property rights claimed by organizations or individuals, and open source code and testing facilities are available on the Internet, as well as the ample technical information.
- *Contributors*—Technical discussions conducted by e-mail and concluded in face-to-face IETF meetings are moderated by the some of most recognized academics and industry experts in the field, regardless of the size or origin of the organization they are from. The authors of the base SIP protocol standard and of extensions to SIP are clearly identified and can be contacted by anyone for discussions regarding their contributions. The SIP discussion mail is mostly populated by hands-on developers exchanging notes on issues with running code on their machines.
- *Surge of creativity*—The completely open and collaborative environment for SIP has generated the largest number of technical contributions experienced in any area of Internet technology, from many individuals, working for various organizations, small and large, from all over the world. The top problem facing the chairs of the IETF SIP working group (WG) is managing the very large number of technical contributions. Various subgroups have been created within the SIP WG to cope with this problem.
- *The SIP standard is based on running code*—Contributors to the SIP WG bring to the table experience from building SIP products and SIP services. Numerous interoperability tests are conducted on a regular basis as SIP matures, so as to prove features before declaring them part of the standard.

The abundance of SIP implementations across the industry is the result of this working style in the development of SIP.

What SIP Is Not

The virtual explosion in proposals to extend the SIP protocol to solve various problems has resulted in much discussion about whether a particular application is well suited to SIP or not. Some of the results of this discussion are summarized here. A more detailed discussion of this is found in the SIP extension guidelines document [1].

SIP is a protocol for initiating, modifying, and terminating interactive sessions. SIP is not a protocol for device control or remote procedure calls (RPCs). It is not a transport protocol—it can carry small message body attachments, but not large chunks or streams of data. SIP is not a resource reservation protocol, since the path of SIP messages does not generally reflect the path of the resulting media. SIP is also not a PSTN replacement protocol—its approach is very different from telecommunications call models and telecommunication signaling protocols. SIP can interwork with the PSTN through gateways, but this is not the primary function of SIP. SIP is also not a session-management protocol, but only a session-setup protocol.

SIP is also not a VoIP protocol, although VoIP is one possible service that can be implemented over a SIP-enabled network. SIP is purely a signaling protocol and makes no specification on media types, descriptions, services, and so on. This is in comparison to a VoIP umbrella protocol such as H.323, which specifies all aspects of signaling, media, features, services, and session control, similar to the other ISDN family of protocols from which it is derived.

Introduction to SIP

SIP is a text-encoded protocol based on elements from the HyperText Transport Protocol (HTTP) [2], which is used for web browsing, and also the Simple Mail Transport Protocol (SMTP) [3], which is used for e-mail on the Internet. SIP was developed by the IETF Multiparty Multimedia Session Control (MMUSIC) WG as part of the Internet Multimedia Conferencing Architecture [5] but has since gained its own SIP WG within the IETF. As the name implies, the primary function of SIP is session initiation (setup), but it also has other important uses and functions, such as notification for presence and short messaging. SIP is used for peer-to-peer communications—that is, those in which both parties in the call are considered equals, there is no master or slave. However, SIP uses a client-server transaction model similar to HTTP, as described in the next section. A SIP client generates a SIP request. A SIP server responds to the request by generating a response.

The growing set of SIP request types (known as *methods*) are shown in Table 6.1. The first six are defined in RFC 3261 [6], the base SIP specification. The rest are extensions to SIP and are defined in separate RFCs or Internet drafts. New methods are continually being proposed to add additional functionality to the protocol.

Table 6.1 SIP Methods

METHOD	DESCRIPTION
INVITE	Session setup
ACK	Acknowledgment of final response to INVITE
BYE	Session termination
CANCEL	Pending session cancellation
REGISTER	Registration of a user's URI
OPTIONS	Query of options and capabilities
INFO	Mid-call signaling transport
PRACK	Provisional response acknowledgment
UPDATE	Update session information
REFER	Transfer user to a URI
SUBSCRIBE	Request notification of an event
NOTIFY	Transport of subscribed event notification
MESSAGE	Transport of an instant message body
PUBLISH	Upload presence state to a server

Responses in SIP are numerical. Many response codes have been borrowed from HTTP as well as new ones created. SIP response codes are divided into six classes, identified by the first digit of the code, as shown in Table 6.2.

Table 6.2 SIP Response Code Classes

CLASS	DESCRIPTION
1xx	Provisional or Informational — Request is progressing but not yet complete
2xx	Success — Request has been completed successfully
3xx	Redirection — Request should be tried at another location
4xx	Client Error — Request was not completed because of an error in the request, can be retried when corrected
5xx	Server Error — Request was not completed because of an error in the recipient, can be retried at another location
6xx	Global Failure — Request has failed and should not be retried again

The response codes are a good illustration of the resemblance of SIP to HTTP. The response code 404 Not Found is reminiscent of web browser error codes.

SIP requests and responses are composed of either the request method or response code, then a list of fields (called *headers*), which are similar to the headers in an e-mail message. In fact some (such as `To`, `From`, `Subject`, and `Date`) have an identical meaning.

An example SIP Request message is shown in Table 6.3, along with the minimum required set of headers and a line-by-line description.

Table 6.3 SIP Example with Line-by-Line Description

LINE	DESCRIPTION
`INVITE sip:userb@there.com SIP/2.0`	The first line of a SIP request does not contain headers, but starts with the name of the method (`INVITE`), followed by a space, the Request-URI, (in this case, `sip:userb@there.com`, which is the destination address of the request), a space, then the current version of SIP (`2.0`). Each line ends with a CRLF (Carriage Return and Line Feed). Note that both RFC 2543 and RFC 3261 SIP implementations are both version 2.0.
`Via: SIP/2.0/UDP 4.3.2.1:5060 ;branch=z9hG4bK765d`	The `Via` header contains the version of SIP (`2.0`) and the transport protocol (`UDP`) followed by the IP Address (`4.3.2.1`) or host name of the originator of the request and the port number (`5060`, the well-known SIP port number). Any server that forwards the request adds a `Via` header with its own address to the message and the port number at which it wants to receive responses. The `branch` parameter is a transaction identifier, indicated to be unique by the first seven characters being the cookie `z9hG4bK`.
`To: User B <sip:userb@there.com>`	The `To` header contains a display name (`User B`) followed by the URI of the request originator enclosed in angle brackets `< >` (`sip:userb@there.com`).

Table 6.3 *(continued)*

LINE	DESCRIPTION
`From: User A <sip:usera@here.com> ;tag=34kd92kfs`	The `From` header contains a display name (`User A`) followed by the URI of the request recipient enclosed in `< >` (`sip:usera@here.com`). The `tag` parameter is a pseudo random string generated uniquely for each dialog.
`Call-ID: 4r59899D8g10c3413`	The `Call-ID` header contains a unique identifier for this call (session). It is usually made up of a locally unique pseudorandom string. All requests and responses during the call will contain this same `Call-ID`.
`Max-Forwards: 70`	The `Max-Forwards` header field is a hop count that is decremented by each proxy server that forwards a request. When the count goes to zero, a `483 Too Many Hops` response is returned.
`CSeq: 1 INVITE`	`CSeq` is the Command Sequence number, which contains an integer (`1`) a space, then the request method (`INVITE`). Each successive request (command) during the call will have a higher `CSeq` number. The caller and called parties each maintain their own separate `CSeq` counts.
`Contact: sip:usera@4.3.2.1`	`Contact` contains one or more SIP URIs that provide information for the other party in the session to contact User A.
`Content-Length: 126`	`Content-Length` is the octet (byte) count of the message body (`126`) that follows the list of SIP headers and is separated from the headers by a single CRLF. A `Content-Length` of 0 indicates no message body.

The details of SIP headers will be discussed as needed in the explanations that follow. For a full description and examples of all SIP headers, see [7].

Elements of a SIP Network

There are three main elements in a SIP Network: user agents, servers, and location services.

User Agents

User agents are the end devices in a SIP network. They originate SIP requests to establish media sessions, and send and receive media. A user agent can be a SIP phone or SIP client software running on a PC or palmtop. Alternatively, a user agent can be a gateway to another network, such as a PSTN gateway, which allows a SIP phone to receive and make calls to the PSTN.

A *user agent client* (*UAC*) is the part of the user agent that initiates requests, while the *user agent server* (*UAS*) is the part of the user agent that generates responses to received requests. Every SIP user agent contains both a UAC and a UAS. During the course of a session, both parts are typically used. This is different from most other client-server architectures, such as web browsing. During a web browsing session, a PC is always the HTTP client (web browser software), and the web server is always the HTTP server.

SIP user agents are usually assumed to be intelligent, in the sense of being part of a fully qualified Internet host as defined in RFC 1121 and RFC 1122 [8], [7], and support many other basic Internet protocols including DHCP, DNS, IMCP, and so on.

Servers

Servers are intermediary devices that are located within the SIP-enabled network and assist user agents in session establishment and other functions. There are three types of SIP servers defined in RFC 3261:

- A SIP *proxy* receives SIP requests from a user agent or another proxy and forwards or proxies the request to another location.
- A *redirect server* receives a request from a user agent or proxy and returns a redirection response (`3xx`), indicating where the request should be retried.
- A *registrar server* receives SIP registration requests and updates the user agent's information into a location service or other database.

SIP proxy, redirect, and registrar servers are purely signaling relay elements. They have no media capabilities and do not initiate requests except on behalf of a user agent.

SIP servers are optional for SIP-based communications, as will be discussed in Chapter 20, "Peer-to-Peer SIP."

LOCATING SIP SERVERS

SIP servers can be located using a number of schemes. User agents are typically configured with IP addresses of a primary and secondary SIP proxy server, in much the same way that a web browser has a default web page that it loads upon initialization. This proxy server is sometimes referred to as the *outbound proxy*, since a user agent will route outgoing messages to that proxy.

Proxies can also be located using a DNS lookup, in which the domain name from a SIP URI is extracted and the IP address of the proxy server supporting that domain is found. This proxy is sometimes called an *incoming proxy*, since it is used to route incoming calls for that particular domain.

A SIP registrar server can be hand-configured in the device or can be located using IP multicasting. Registrar servers listen at the well-known SIP multicast address (such as `sip.mcast.net`) and can receive registrations. A SIP registrar server can often be located by sending a registration request to an outbound proxy, which then proxies the request to a registrar server. In this way, SIP servers can be located by sending requests to other SIP Servers, as part of the address resolution process described for SIP in the next section and governed by RFC 3263.

Location Services

A *location service* is a general term used in RFC 2543 for a database. The database may contain information about users such as URIs, IP addresses, scripts, features, and other preferences. It also may contain routing information about the SIP-enabled network, including the locations of proxies, gateways, and other location services. User agents generally do not interact directly with a location service, but go through a proxy, redirect, or registrar server. SIP servers use a non-SIP protocol to query, update, and retrieve records from the location service in the course of routing a SIP message.

The role of these elements will be discussed in terms of SIP functions in the next section.

SIP Functions

The SIP protocol will be introduced in terms of some of the basic functions of a communications network: address resolution, session-related functions (including session setup, media negotiation, session modification, session termination, and cancellation), mid-call signaling, call control, QoS call setup, and nonsession-related functions (such as mobility, message transport, event subscription and notification, authentication, and extensibility). Each of these will be discussed and explained in turn.

Address Resolution

Address resolution is one of the most important functions of the SIP protocol. The SIP address resolution process usually begins with a URI and ends with a username at an IP address. This resolution from a general name to an actual user at a host is extremely powerful in that various types of mobility and portability are automatically implemented. Address resolution can be performed by both user agents and servers.

The address resolution process can involve the following steps:

- DNS NAPTR lookup to determine transport protocol (UDP, TCP, SCTP) as described in RFC 3263
- DNS SRV [9] lookup to determine the server host name and port number as described in RFC 3263
- DNS A lookup to determine the IP address of the host
- ENUM [10] lookup if a telephone number
- When routed to a proxy server in the domain of the user, a location service lookup, as described in RFC 3261

While it is possible that a SIP user agent may have access to a location service, this lookup is usually performed by a proxy or redirect server on behalf of a user agent.

In general, the address resolution process involves multiple steps and multiple SIP message hops. This allows user agents and proxies to perform request routing on a hop-by-hop basis. Each proxy consults DNS or a routing table, then forwards the request to the next hop. This process continues until the request is delivered to the destination. Note that routing of the responses in SIP does not involve address resolution; all responses are routed back through the same set of proxies as the request. This is possible because of the `Via` header chain in the request message.

Consider the request routing example of Figure 6.2. This example does not show outgoing and incoming proxy servers, but just one proxy server in the middle. Such a simple network configuration may apply for routing calls within a small private IP network. The SIP user agent A wishes to send a general SIP request to another user agent B identified by the SIP URI `sip:userb@there.com`. The SIP Telephone A first performs a DNS Naming Authority Pointer (NAPTR) and then a Service Record (SRV) query to determine the transport protocol and to locate the proxy server for the `there.com` domain (which is TCP and `sipproxy.there.com` using port 5060 in steps 1 and 2). The SIP request 3 is then sent to the IP address of `sipproxy.there.com`. This proxy then consults a location service in step 5, which locates the current registration URI for user B, which is `tel:+65123456789`. The proxy then

sends a set of ENUM queries in step 7 to DNS to find the corresponding URI address, which is returned and used as `sip:userb@100.101.102.103` in Step 9. The request is then routed to user B at that IP address, who returns a successful SIP response `200 OK` in step 10 to the proxy server. The proxy server forwards the success response `200 OK` in step 11 back to caller A.

The address resolution process in SIP is dynamic—a proxy can use *any* header present in a request and many other factors in routing decisions, including the following:

- Time of day
- `From` header
- Various request header fields for load sharing or automatic call distributor (ACD) applications

Usually, this process of address resolution only occurs once at the start of a session. The results of the initial address resolution are cached and used in future requests between user agents.

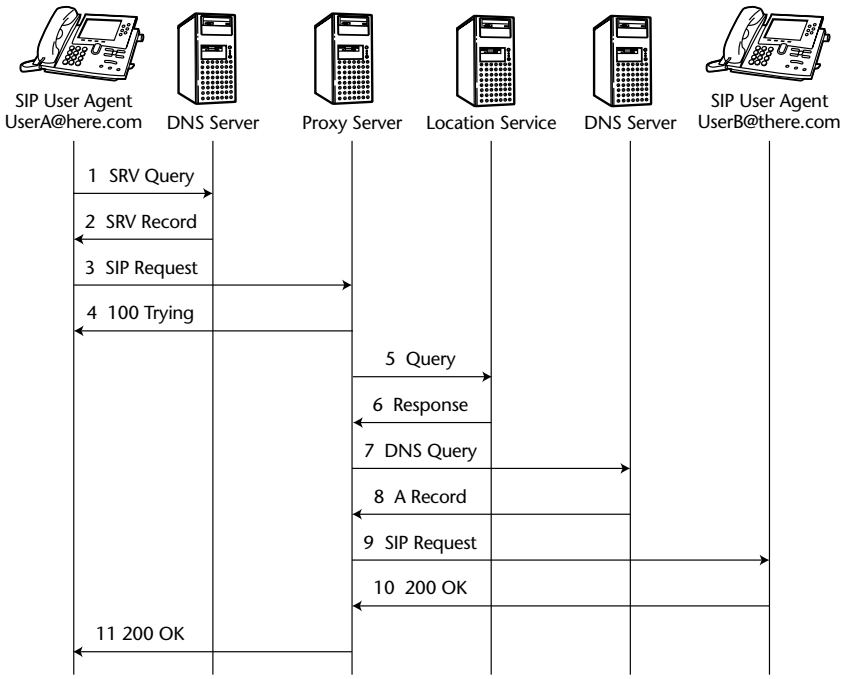


Figure 6.2 Request address resolution example using location service and DNS

Session-Related Functions

Most SIP functions involve setting up sessions or occur during an established session. Although some applications of SIP do not make any use of session-related functions, most useful applications of SIP make use of these powerful functions.

Session Setup

As the name of the protocol implies, *session setup* is the primary function of SIP. Being a polite protocol, SIP uses an `INVITE` request to setup a session between two user agents. The `INVITE` message usually contains a message body that describes the type of session the user agent wishes to establish.

A SIP user agent client initializes the `To`, `From` with a tag parameter, and `Call-ID` headers at the start of the session. Each user agent that generates a response adds a tag to the `To` header field. The combination of the `To` tag, `From` tag, and `Call-ID` are then used to uniquely identify this session, referred to as a "dialog" in SIP. These headers are never modified during a session. This information, plus any required media information, represents the minimum amount of "call state" that a user agent must maintain.

In the event of a user agent "crash" or reboot, the state information must be recovered somehow for the call to continue; otherwise, the call will have to be reinitiated. Note here that in harmony with the Internet architecture, the call state can be maintained in the SIP endpoints, without any call state being kept in the servers in the networks, if so desired. SIP proxy servers may, however, keep transaction state during the call setup phase. Keeping the state in SIP endpoints makes the call setup independent of transient failures in the network, since the endpoints can use the state to retransmit messages for call setup.

The SIP session setup is a three-way handshake—`INVITE/200/ACK` for a success, and `INVITE/4xx` or `5xx` or `6xx/ACK` for a failure. `INVITE` is the only method in SIP in which there is this three-way handshake involving `ACK`. All other SIP requests are of the form `REQUEST/200` or `REQUEST/4xx` or `5xx` or `6xx` for a failure. Figure 6.3 shows a successful session setup between two SIP phones involving an `INVITE`, two provisional responses (`100 Trying` and `180 Ringing`), and a final response (`200 OK`), which receives an `ACK`. Zero or more provisional (`1xx`) responses can be sent prior to a final response.

Once established, a media session continues indefinitely without requiring a further SIP signaling message exchange. A SIP Session Timer can, however, be used to terminate excessively long SIP sessions [4]. If one party to the session wishes to modify or terminate the session, a new exchange of SIP signaling messages ensues.

This three-way handshake allows for *forking*, which is a parallel search initiated by a proxy, in which multiple successful responses can be returned for a single `INVITE` in a reliable way, as will be discussed in a later chapter.

TRANSPORT OF SIP MESSAGES OVER IP

SIP messages can be carried by transport-layer IP protocols, including Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Stream Control Transport Protocol (SCTP), and Transport Layer Security (TLS). TLS is also known by the name of its predecessor protocol, Secure Socket Layer (SSL), that uses TCP transport. Datagram TLS (DTLS) uses UDP transport.

SIP has built in reliability mechanisms so that it can use a "best effort" unreliable transport protocol such as UDP. When UDP is used, one SIP message is carried per UDP datagram. When TCP is used, a TCP connection is first opened between the user agent and the next hop (which could be directly to the other user agent or to a server). SIP messages are then streamed in the connection. The `Content-Length` header is mandatory for stream transports as it provides a way to parse separate messages. Responses are sent in a second TCP connection opened in the reverse direction using the information in the `Via` header field. A TCP connection does not have to be kept open for the duration of a session. If it has closed, a new TCP connection would have to be opened to send a re-`INVITE` or a `BYE` to close the session.

Note that a SIP message path with multiple hops can use UDP for some hops and TCP for other hops. The transport protocol used for a hop is recorded in the `Via` header along with the IP Address and port number for sending responses. SIP messages can also be carried using other transport protocols such as Stream Control Transport Control (SCTP) developed by the IETF SIGTRAN Working Group [11]. SCTP provides a reliable connection and additional functionality such as multi-homing. *Multi-homing* allows a host to connect to two or more servers at the same time. Should one of these servers become unreachable, traffic can be instantly routed to the other server, minimizing the outage time.

The choice of transport protocol is determined be the application. Most simple SIP user agents such as SIP phones and PC clients use UDP for transport because of the simplicity of managing a UDP session compared to other transport protocols. Also, there is no setup delay in opening up a connection (as with TCP transport) before the SIP message exchange can begin. TCP is sometimes used between proxies, or in other applications where a more permanent SIP connection is useful. SCTP has been proposed for use in connections between proxies or between proxies and large PSTN Gateways where a high throughput and low-latency connection is needed.

Media Negotiation

Media negotiation is part of the `INVITE/200/ACK` sequence used to establish a SIP session between two endpoints. SIP itself does not provide the media negotiation, but it enables media negotiation to occur between the user agents using the Session Description Protocol (SDP). SDP is not a true protocol, but is rather a text-based description language, which is defined by RFC 2327 [12]. It has required and optional fields. Some of the required fields are included in a SIP message body but are not used as will be shown here.

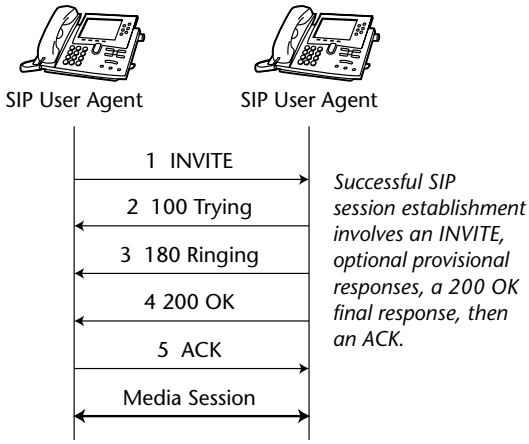


Figure 6.3 Successful session establishment example using `INVITE`

SDP was initially developed in the framework of the Internet multimedia architecture as a sort of "*TV Guide*" for multicast multimedia sessions over the Internet. Some of the capabilities of SDP are, therefore, not used in SIP, such as advertising the origin of the session advertisement, the subject of the session and the scheduling function based on starting and end time, with repeat features (the `t=`... line).

The negotiation is an offer answer model defined in RFC 3264 in which one user agent proposes one or more media types, and the other user agent either accepts or declines each media session in a response. Referencing Figure 6.3, usually, the offer is made in the initial `INVITE` by the caller, and the response is carried in the `200 OK`. However, the caller can allow the called party to select the media session type by not sending SDP in the `INVITE`. In this case, the called party makes the offer in the `200 OK` (or in a reliable provisional response), and the caller responds in the `ACK`. In the SDP body attached to the SIP header, the user agents specify the media type, codec, IP address, and port number for each media stream. More than one codec can be specified for each media type. Once an offered codec has been accepted, user agents must be prepared to receive media with that codec for the duration of the session. For examples of offer/answer SDP exchanges, see RFC 4317.

The example SDP offer shown in Table 6.4 contains two media lines: one for video and one for audio. Each media line has two possible alternative codecs that the calling user agent supports.

Table 6.4 SDP Offer Example with Line-by-Line Description

LINE	DESCRIPTION
`v=0`	Version — Current version number of SDP (`0`) — not used by SIP.
`o=usera 2890844526 2890844526 IN IP4 client.example.com`	Origin — Only the version (`2890844526`) is used by SIP.
`s=Subject`	Subject
`c=IN IP4 128.2.3.1`	Connection — network (`IN` for Internet), address type (`IP4` for IP Version 4) and address (`128.2.3.1`).
`t=0 0`	Time — start and stop time — not used by SIP.
`m=video 51372 RTP/AVP 34 98`	Media — Media type (`video`), port number (`51372`), type (`RTP/AVP Profile`), and number (Profiles `34` or `98`).
`a=rtpmap:34 H263/90000`	Attribute — `rtpmap` lists attributes of RTP/AVP video profile `34`, including codec (`H.263`) and sampling rate (`90000` Hz).
`a=rtpmap:98 H264/90000`	Attribute — `rtpmap` lists attributes of RTP/AVP video profile `98` (dynamic payload) including codec (`H.264`) and sampling rate (`90000` Hz).
`m=audio 4006 RTP/AVP 0 97`	Media — Second media type (`audio`), port number (`4006`), type (`RTP/AVP` Profile), and number (Profiles `0` or `97`).
`a=rtpmap:0 PCMU/8000`	Attribute — `rtpmap` lists attributes of RTP/AVP audio profile `0`, including codec (`PCMU` — PCM μ-Law) and sampling sate (`8000` Hz).
`a=rtpmap:97 iLBC/8000`	Attribute — `rtpmap` lists attributes of RTP/AVP audio profile `97` (dynamic payload) including codec (`iLBC`) and sampling rate (`8000` Hz).

In the response to this offer, the other party declines the video media session by setting the port number to 0, and accepts the audio session by selecting the `iLBC` codec and returning a nonzero port number, as shown in Table 6.5.

Further negotiation and changes to the media can be accomplished using a re-`INVITE` once the session is established, as described in the next section.

This type of limited media negotiation capability is supported by SDP and, hence, in SIP. Currently work is underway to develop a successor to SDP,

tentatively called "SDPng" for Next Generation [13]. This new protocol will have more advanced media negotiation and description capabilities. It is likely that support of SDP will remain in the base SIP specification, with successors to SDP being optional to support.

Session Modification

Once a session has been established using the INVITE/200/ACK sequence, it can be modified by another INVITE/200/ACK sequence, sometimes referred to as a re-INVITE. Since there can only be one pending SIP request at a time, a re-INVITE cannot be sent until the initial INVITE has been completed with an ACK. The re-INVITE can be done by either party and uses the same To, From (including tags), and Call-ID as the INVITE. However, the SDP in the re-INVITE is assumed to replace the initial INVITE SDP, if the re-INVITE is successful. If the re-INVITE fails in any way or is refused, the original SDP and the original media session will continue until a BYE is sent by either party.

Table 6.5 SDP Response Example with Line-by-Line Description

LINE	DESCRIPTION
v=0	Version – Current version number of SDP (0) – not used by SIP.
o=userb 2890844342 2890844543 IN IP4 client.example.net	Origin – Not used by SIP.
s=-	Subject.
c=IN IP4 16.22.3.1	Connection – Network (IN for Internet), address type (IP4 for IP version 4) and address (16.22.3.1).
t=0 0	Time – Start and stop time – not used by SIP.
m=video 0 RTP/AVP 34 98	Media – media type (video), port number is set to zero, which indicates that the video session has been declined.
m=audio 6002 RTP/AVP 97	Media – Media type (audio), port number (6002), type (RTP/AVP profile), and number (profile 4). By specifying a nonzero port number, the audio session has been accepted.
a=rtpmap:97 iLBC/8000	Attribute – rtpmap lists attributes of RTP/AVP audio profile 97, including codec (iLBC) and sampling rate (8000 Hz).

In the example of Figure 6.4, a call is set up between two user agents using the media description `sdp1`, carried in the initial `INVITE` and `200 OK` response. The called party tries to change the session parameters by sending another `INVITE` with a new message body `sdp2'`. However, this is not acceptable to the other party, and the re-`INVITE` fails with a `405 Not Acceptable` response in Message 6. The media session continues using the initial media parameters. The called party tries one more time and this time the re-`INVITE` succeeds, and the old media session is terminated and a new one using `sdp2''` and `sdp1''` is established with different values in each direction. Note that the re-`INVITE`s do not usually generate provisional responses (such as `180 Ringing`), since the two parties are already communicating with each other.

Note that a re-`INVITE` may change any of the media characteristics, including the session type, codec used, even the source IP addresses and port number.

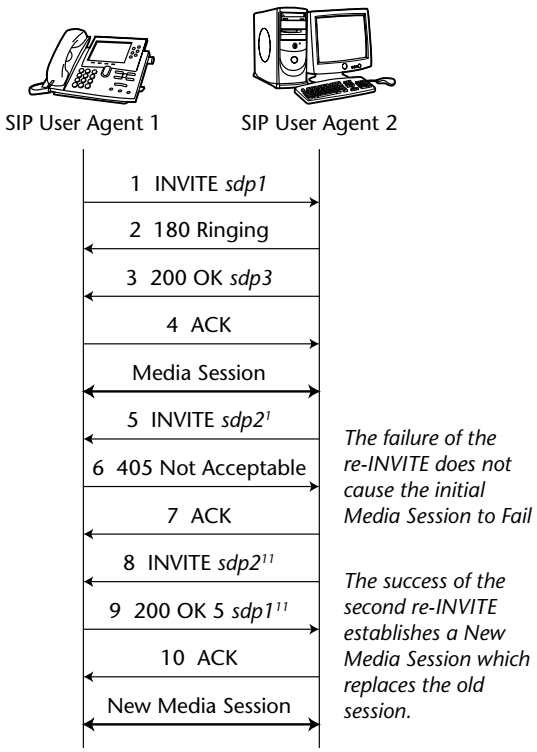


Figure 6.4 Session modification example using `INVITE`

Session Termination and Cancellation

Session termination and cancellation are two separate operations in SIP but are often confused. Session termination occurs when either user agent sends a `BYE` referencing an existing call leg (that is, a session successfully established using the `INVITE/200/ACK` exchange). This is shown in the example of Figure 6.5

Session cancellation occurs when a user agent ends a call prior to the call setup completing and the call being established. The reader can think of the analogy to the action of the cancel button on the browser. In this scenario, a user agent that has sent an `INVITE`, but has not yet received a final response (`2xx`, `3xx`, `4xx`, `5xx`, or `6xx`), sends a `CANCEL` request. A `CANCEL` can also be originated by a proxy to cancel individual legs in a forking proxy or parallel search.

While `INVITE` and `BYE` are end-to-end methods, `CANCEL` is an example of a SIP request that is a hop-by-hop request. A proxy receiving a `CANCEL` request immediately responds with a `200 OK` response, then proxies the `CANCEL` on to the same set of destinations to which the original `INVITE` was sent.

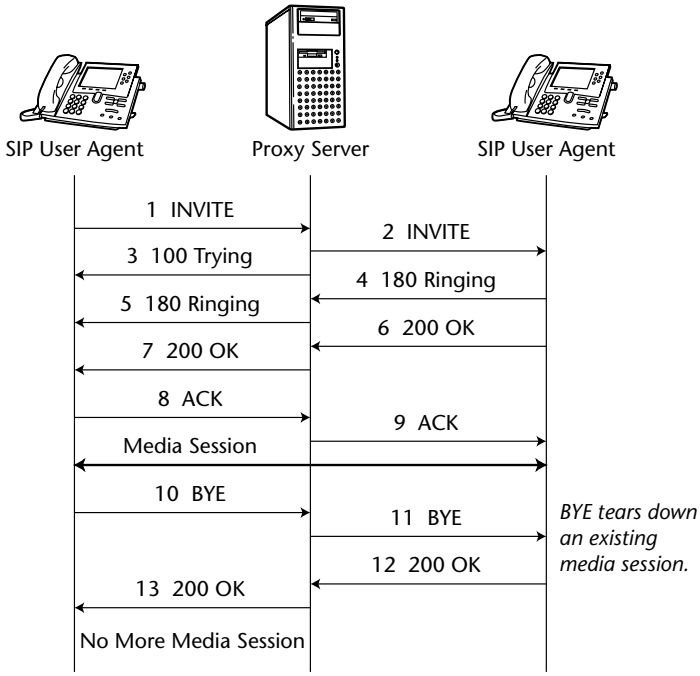


Figure 6.5 Session termination example using `BYE`

A user agent receiving a CANCEL replies with a 200 OK if a final response has not yet been sent, or a 481 Transaction Unknown response if a final response has been sent. The latter corresponds to the "race" condition, where the CANCEL and final response "cross on the wire." In this condition, the user agent may have to send a BYE to cancel the call [6].

In the example of Figure 6.6, a user agent sends an INVITE request, and then a CANCEL request. The INVITE is forwarded through two proxies to reach the destination user agent. Notice that the CANCEL request sent to the first proxy results in a 200 OK response to the CANCEL, and the CANCEL being forwarded to the next proxy. The second proxy immediately sends a 200 OK to the first proxy and forwards the CANCEL to the destination user agent. Finally, the user agent responds with a 200 OK to the CANCEL and a 487 Request Cancelled response to the INVITE. The 487 response is acknowledged by the second proxy with an ACK, and then forwards the 487 to the first proxy, which eventually is received by the calling user agent, which then knows that the pending session was successfully cancelled. (Non-success final responses such as 3xx, 4xx, 5xx, or 6xx are always acknowledged on a hop-by-hop basis. Only a 200 OK receives an end-to-end ACK.) The user agent then has completed two transactions: a CANCEL/200 and an INVITE/487/ACK transaction.

Since it is possible that a CANCEL may be sent at the same time as a 200 OK response, the user agent must be prepared to send an ACK and a BYE to the 200 OK even after sending the CANCEL.

Note that a CANCEL request is unique in that it can not be challenged for authentication as all other SIP requests can be, as described later in this chapter.

Mid-Call Signaling

Mid-call signaling is a signaling message exchange between two user agents that does not change the session parameters between them. If a mid-call signaling event did change the session parameters (that is, the SDP), then a re-INVITE would be used. Otherwise, the SIP INFO method [14] is used to transport the information between the two user agents. The information is carried in the message body of the INFO request. For example, mid-call signaling information contained in an ISDN USR (User to User Message) message can be transported using the INFO method in a network where ISDN User Part (ISUP) encapsulation is being used. An example of this is shown in Figure 6.7, where basic SIP-to-ISUP mapping is performed by two gateways. Following are the ISDN messages in Figure 6.7:

- *IAM*—Initial address message
- *ANM*—Answer message
- *USR*—User to user message

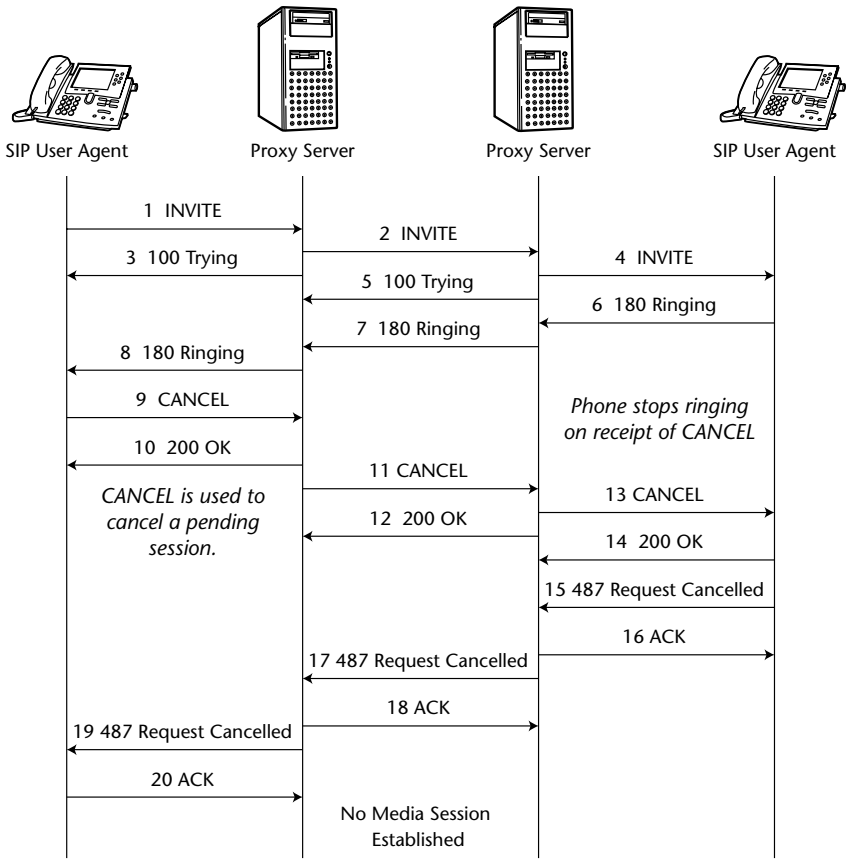


Figure 6.6 Session cancellation example using `CANCEL`

Call Control

The SIP architecture is one of peer-to-peer communication and end-to-end control. For example, a proxy may not issue a `BYE` request terminating a call. It can only be issued by one of the user agents (end devices) participating in the call.

However, the ability for a third party to direct or control a call between two other parties can be extremely useful in various service implementations. For example, an embedded SIP URI in a web page, when clicked, could cause a desktop SIP phone to place a call to the desired URI. Or, third-party call control could be used to implement a web call center or ACD feature, which is useful for handling calls to customer service numbers, where the controller receives the call and routes it based on a number of factors such as available agents, time of day, and other factors.

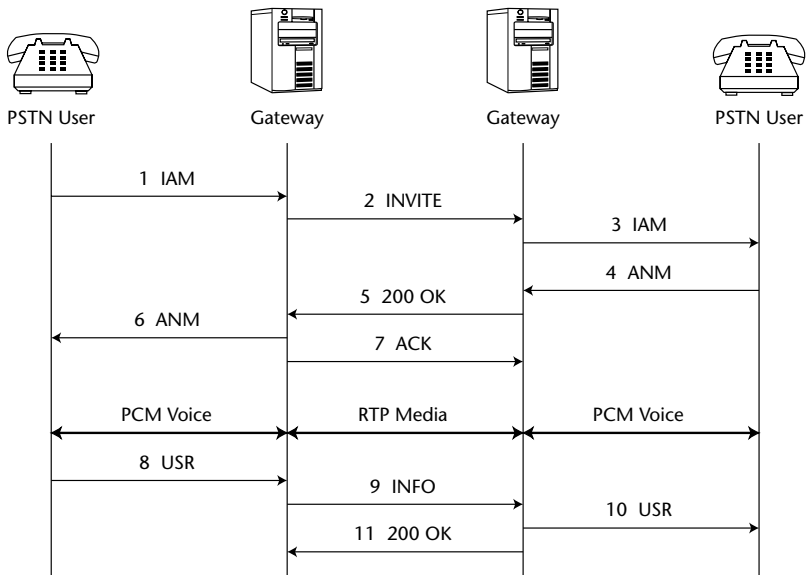


Figure 6.7 Mid-call signaling example using `INFO`

There are two ways of implementing third-party control. The first uses a controller that receives the SIP `INVITE` request, answers it, then proxies the `INVITE` to a third party. The controller then stays in the signaling path, swapping SDP from one leg to another, and transparently controlling the call. The second way uses the `REFER` method [15] to initiate the third-party control.

In the example of Figure 6.8, A and B establish a session. A then refers B to initiate a session with C using a `REFER` request. A then terminates the session with B, while B establishes a new session with C.

The `REFER` request in Message 6 has the following form:

```
REFER sip:userb@there.com SIP/2.0
Via: SIP/2.0/TCP pc.there.com:5060;branch=z9hG4bK765d
To: User B <sip:userb@there.com>
From: User A <sip:usera@here.com>
Call-ID: a5-32-43-12-77@4.3.2.1
CSeq: 2 REFER
Refer-To: <sip:UserC@anywhere.com>
Referred-By: <sip:usera@here.com>
Content-Length: 0
```

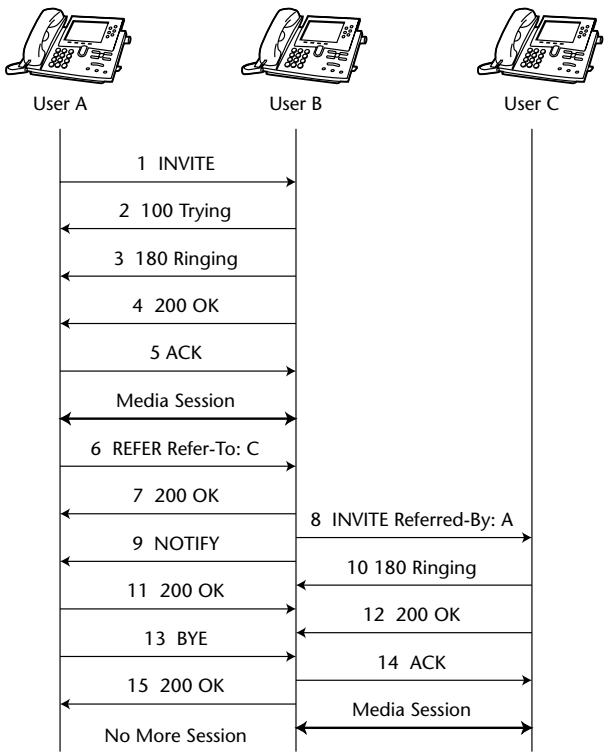


Figure 6.8 Call control example using `REFER`

The resulting `INVITE` message (Message 8) would then have the following form:

```
INVITE sip:UserC@anywhere.com SIP/2.0
Via: SIP/2.0/UDP 100.101.102.103:5060
To: <sip:UserC@anywhere.com>
From: User B <sip:userb@there.com>
Call-ID: 383874109476@there.com
CSeq: 67 INVITE
Contact: sip:userb@here.com
Referred-By: <sip:usera@here.com>
Content-Length: ...
```

The `Refer-To` header in the `REFER` contains the URI to whom A is referring, while the `Referred-By` header identifies A as the referrer, and is passed to C in the `INVITE` so that C knows that B has been referred by A in initiating this session.

Preconditions For Call Setup

SIP has extensions to require preconditions.

Quality of service (QoS) can be supported in the network layer (IP layer 3) and in the link layer below (layer 2). QoS in IP networks is independent of any specific application and the network, therefore, need not be aware of the specifics of the applications (be they telephony, multimedia, financial transactions, or games). SIP is orthogonal to QoS.

Setting up an application such as a commercial-grade phone call with QoS requires the support of valuable network resources (for example, giving priority to a media flow having a data rate of 100 kb/s for 30 minutes over a distance of 5,000 km). The authorization required to provide the network resources for the SIP-initiated session involves complex procedures for authentication, authorization, and accounting (AAA) that go beyond the topics discussed here [16], [17], [18]. We will, therefore, limit the discussion of QoS for SIP only for the simple case where the AAA issues can be ignored.

SIP enables user agents to establish sessions using the `INVITE/200/ACK` exchange. However, in order to establish an IP session with QoS, a more complicated message exchange is required. The Integrated Services QoS protocol assumed in these examples is the Resource Reservation Protocol (RSVP) [19]. However, the approach described here for SIP will also work with other QoS approaches, such as setting the type of service (TOS) bits in the IP header used in DiffServ [20].

A simplified approach to QoS would be to first establish a "best-effort" session between user agents, then use a re-`INVITE` to set up the new QoS session. However, since the SIP messaging is completely independent of the media, it is entirely possible to successfully set up a session using SIP, only to have the session fail because of lack of bandwidth for the media, in which case this approach will fail. Also, there was a desire to mimic the behavior in the PSTN, where the called party's phone will not ring if there are not sufficient resources (that is, trunks) to complete the call if answered. The approach described here was developed by the PacketCable consortium [21] for the Voice over Cable Modem project. The call flow is shown in Figure 6.9.

This call flow makes use of three extensions to SIP. The first is Early Media [22], which allows SDP to be present in the provisional `183 Session Progress` response. This allows an addition media (SDP) handshake between the user agents necessary to establish the QoS prior to the call being answered. The second is the Reliable Provisional Responses extension to SIP, which allows a lost provisional response such as a `183` to be detected and retransmitted (see the sidebar "Message Retransmissions in SIP"). The receipt of the `183` response is indicated by the `PRACK` (Provisional Response `ACK`nowledgement) [23] message. The third extension is the use of the `COMET` (preCOnditions MET) [24] method, which allows the UAS to indicate the QoS preconditions

have been met and that the user may now be alerted, and the `180 Ringing` response sent. The call then continues as per normal. Note that the need for QoS was indicated in the last line of the SDP of the initial `INVITE` request, as shown here with the attribute `qos:mandatory`:

```
   INVITE with mandatory QoS request
INVITE sip:userb@there.com SIP/2.0
Via: SIP/2.0/UDP 100.101.102.103:5060;branch=z9hG4bK765d
To: <sip:userb@there.com>
From: User A <sip:usera@here.com>
Call-ID: 5448kewl13981304oierek
Max-Forwards: 0
CSeq: 1 INVITE
Contact: sip:usera@here.com
Content-Length: ...

v=0
c=IN IP4 100.101.102.102
m=audio 47172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=qos:mandatory
```

MESSAGE RETRANSMISSIONS IN SIP

The base SIP specification allows almost any lost request or response method to be automatically retransmitted. The sender of a SIP request using nonreliable transport starts a timer, called T1 (default value is 500ms). If a response is not received before the expiration of the timer, the request is retransmitted. If a provisional (`1xx`) response is received, the sender switches to a second longer timer, called T2 (default value 4 seconds). If the request itself is lost, the recipient will not have received it and will never generate a response. After the expiration of T1, the sender will resend the request. If the response to the request is lost, the sender will again resend the request. The recipient will recognize the request as a retransmission and retransmit its own response.

Handling of `INVITE` requests is slightly different than all other request types, since it may take a long time for the call to be answered by a person. The receipt of a provisional response to an `INVITE` does not switch to timer T2 but stops all retransmissions of the `INVITE`. A responder to an `INVITE` starts timer T1 when it sends a final response. If an `ACK` is not received, the responder resends the final response. This allows a lost `INVITE`, final response, or `ACK` to be detected and retransmitted.

The exceptions to this retransmission rule are provisional responses. Since provisional responses do not receive an `ACK`, there is no way for either party to know if one has been lost. The Reliable Provisional Response [23] extension to SIP was developed to allow a provisional response to be acknowledged with a `PRACK`, thus providing reliability for all requests and responses in SIP.

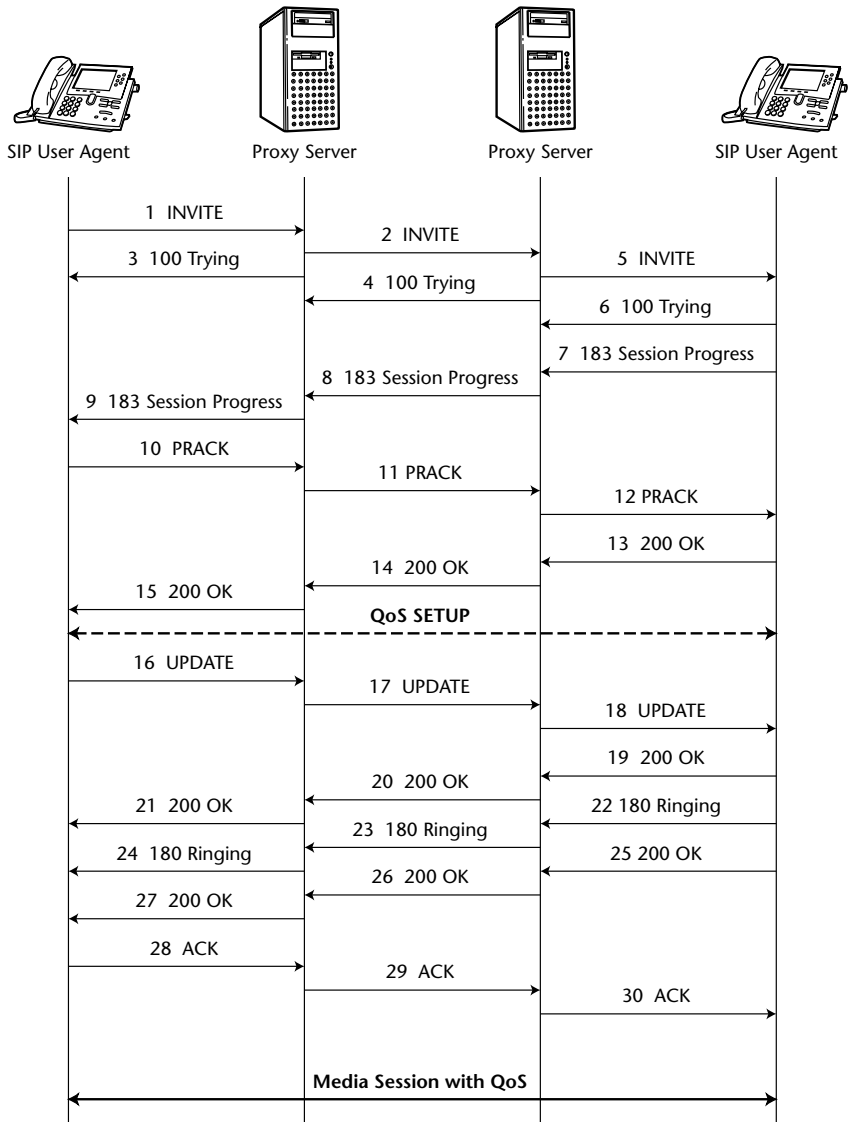


Figure 6.9 Preconditions of call setup using `UPDATE` and `PRACK`

Nonsession-Related Functions

Some SIP functions do not relate directly to session setup. These functions can occur outside of a session established using SIP.

Mobility

The registration function of SIP is very similar to registration in cell phones. In a registration message, a user sends a proxy server the URI for which it wishes to receive calls. This built-in support of mobility is an extremely useful feature of SIP and is one of the most often cited benefits of the protocol over others. It is also this support of mobility that has lead to the application of the protocol in many new applications and its proposed used for call control in third generation wireless networks.

The SIP `REGISTER` request is used to accomplish this function. The request contains `Contact` headers, which are the URIs being registered by the user. For example, a successful user agent registration is shown in Figure 6.10. The user initially registers his office SIP phone by sending a `REGISTER` message to the Registrar server. The Registrar updates the user's record in the location service and returns a `200 OK` confirmation of the registration. Later in the day, the user leaves his office for home where he cancels his office phone registration and registers his SIP home phone. (A mobile phone registration during the commute could also be envisaged). Note that the protocol used to upload the registration to the location service or other database is not SIP. Incoming calls to the users URI will now be routed to the IP address of his SIP home phone. Also note that the home phone need not be SIP for this pre-call mobility. The user could also register a PSTN phone using web access, e-mail, or having the registration preprogrammed for certain times.

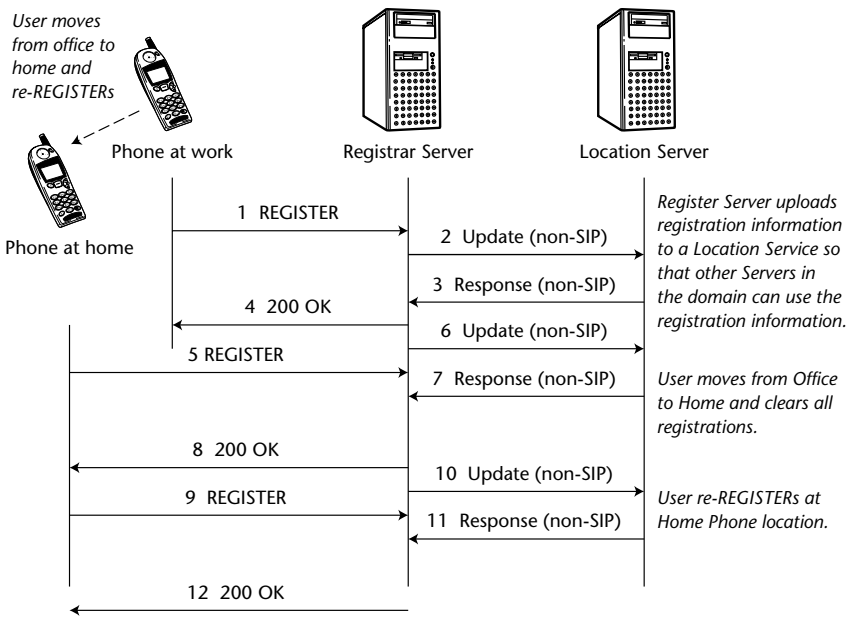


Figure 6.10 Mobility Example using `REGISTER`

A user agent can be configured to automatically register upon initialization, at preset intervals, or whenever a new user signs on to the particular device.

Registration is not limited to a single URI. Multiple URIs can be used to list a number of alternative locations in a preferred order, or may be used to list multiple possible services such as SIP, PSTN, and e-mail. For example, consider the following example REGISTER message:

```
  SIP client to Registrar
REGISTER sip:registrar.here.com SIP/2.0
Via: SIP/2.0/UDP 4.3.2.1:5060;branch=z9hG4bK87ds
To: User A <sip:usera@here.com>
From: User A <sip:usera@here.com>;tag=34323kl12d
Call-ID: a532431277gfhd43gsfg3awrsad
Max-Forwards: 70
CSeq: 16 REGISTER
Contact: sip:usera@4.3.2.1;class=personal
Contact: tel:+1-314-555-1212
Contact: mailto:usera@here.com
Content-Length: 0


   Registrar to SIP client
SIP/2.0 200 OK
Via: SIP/2.0/UDP 4.3.2.1:5060;branch=z9hG4bK87ds
To: User A <sip:usera@here.com>;tag=9128394
From: User A <sip:usera@here.com>;tag=34323kl12d
Call-ID: a532431277gfhd43gsfg3awrsad
CSeq: 16 REGISTER
Contact: sip:usera@4.3.2.1;class=personal
Contact: tel:+1-314-555-1212
Contact: mailto:usera@here.com
Content-Length: 0
```

The 200 OK response to a REGISTER echoes the three Contact URIs that have been successfully registered. In this case, a query to the Location Service for the SIP URI sip:usera@here.com would return the three Contact URIs that were registered. The first is a SIP URIs that can be used to reach user A. The second URI represents user A's telephone number, which could be reached via the PSTN (or through SIP and a gateway), and the e-mail address of user A.

The SIP URIs in this example may also contain parameter extensions not shown here, such as Contact that are defined in the Caller Preferences document [25], which allows a user agent to identify information about the type of device identified by the URI. For example, the first URI is identified as a personal URI, the second as voicemail, the third is for business, and the fourth is a cell phone.

Normally, a SIP server would process a list of URIs by trying the first Contact header URI first, then moving to the second, and so on, assuming a sequential search. The Reject-Contact header works in a similar way, but

with the reverse result. In this way, SIP allows user preferences to be carried with a request message. For example, a SIP request sent to a user's URI could be routed to any number of devices, depending on where the user is currently registered, and what features and scripts are activated in the called party's SIP network. A SIP request could also be sent containing a `Reject-Contact` [25] header, indicating that the requestor does not want to reach voicemail, for example.

When a SIP message is processed by servers, it is usually up to the server as to whether to proxy or redirect the request, and whether to invoke a serial or parallel search (forking). However, the use of the `Request-Disposition` header allows the requestor to have some input. For example, a request containing a `Request-Dispositon: proxy, sequential` indicates that the requestor wishes the request to be proxied instead of redirected, and to have a serial search as opposed to a parallel search. The Caller Preferences document [25] describes all the options. Note that a proxy that does not implement a particular feature may simply ignore the header. The Caller Preferences draft includes pseudocode describing the exact URI and parameter matching, and the interaction of the `q` preference values, if present.

Note that the use of the caller preferences defined `Contact` header extensions is useful in SIP CGI and CPL scripting for SIP service creation.

The use of the `Requires: prefs` header allows a user agent to require that a registrar support caller preferences and will act accordingly.

Message Transport

The `MESSAGE` method [26] simply transports the message body to the destination URI within or outside an established session. For example, consider the following instant message (IM) transported using SIP:

```
  SIP message
MESSAGE im:userb@there.com SIP/2.0
Via: SIP/2.0/UDP pc.here.com;branch=z9hG4bK343g
To: User B <im:userb@there.com>
From: User A <im:usera@here.com>;tag=4541232ds
Max-Forwards: 70
Call-ID: a532431277432513
CSeq: 15 MESSAGE
Content-Type: text/plain
Content-Length: 15

Hi, how are you?
```

Notice that the URIs in the example are IM URIs instead of SIP URIs. When user B receives the message, a `200 OK` response would be generated. Unlike the `INFO` method, which can only be sent when there is an established session

between two user agents, a MESSAGE request can be sent at any time. SIP support for presence and instant messaging includes SIP messages as in this example. The other methods in SIP to support instant communications are event subscription and notification for presence.

Event Subscription and Notification

The ability to request and receive notification when a certain event occurs is supported in SIP by the SUBSCRIBE and NOTIFY request types [26], [27]. For example, the automatic callback feature in telephony can be used when the called party is busy (off hook) and the caller wishes to be notified as soon as the called party is available [29]. In Figure 6.11, user A sends an INVITE request and receives a 486 Busy Here response from user B's user agent. User A then sends a SUBSCRIBE request to user B requesting notification when user B is available to establish a session. When user B sends a NOTIFY request indicating that the user is now available, User A immediately establishes the session.

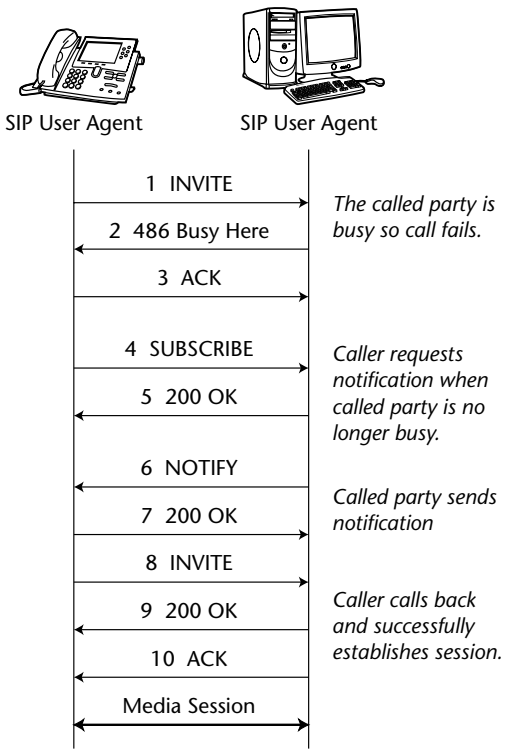


Figure 6.11 Automatic callback feature example using SUBSCRIBE and NOTIFY

The subscription request has the following form:

```
SUBSCRIBE sip:userb@there.com SIP/2.0
Via: SIP/2.0/UDP 4.3.2.1;branch=z9hG4bK343d
To: User B <sip:userb@there.com>
From: User A <sip:usera@here.com>;tag=h34s341
Max-Forwards: 70
Call-ID: a5f2d43127767eh54wfd
CSeq: 23 SUBSCRIBE
Contact: <sip:usera@client.there.com>
Event: dialog
Expires: 60
Content-Length: 0
```

The notification request has the form:

```
NOTIFY sip:usera@here.com SIP/2.0
Via: SIP/2.0/UDP pc.here.com:5060;branch=z9hG4bK343d
To: User A <sip:usera@here.com>;tag=9839421323
From: User B <sip:userb@there.com>;tag=h34s341
Max-Forwards: 70
Call-ID: a5f2d43127767eh54wfd
Subscription-State: active;expires=55
CSeq: 5 NOTIFY
Event: dialog
Content-Length: ...
```

The `Event` header indicates which event notification is being requested. If User B's user agent was not willing to provide the notification of this event, a `603 Decline` response could be sent.

A service network can be built in a serverless design using `SUBSCRIBE` and `NOTIFY`. SIP also supports a server-based approach using the `PUBLISH` method.

Presence Publication

The SIP `PUBLISH` method allows a user agent to publish or upload presence information to a presence server. The presence server can then distribute this information. Figure 6.12 shows an example of this.

Authentication Challenges

SIP supports two types of authentication challenges: user agent to user agent, and user agent to server. It does not currently support server to server authentication challenges, although this could be accomplished using a non-SIP scheme such as IPSec [30]. SIP supports a number of authentication schemes borrowed from HTTP. SIP Digest authentication is the most commonly used scheme today, which relies on a challenge/response and a shared secret

between the user agent requestor and the proxy or user agent requiring the authentication. Any SIP request can be challenged for authentication.

The shared secret usually will be an encrypted username and password. A typical authentication SIP message exchange between user agents has the form `INVITE/401 Authentication Required/ACK` in which the user agent discovers that the request requires authentication, and also learns the nature of the authentication challenge from the `401` response. Then, a new `INVITE` containing an `Authorization` header is resent. If it contains the correct credentials, the call will proceed as normal. Otherwise, another `401` response will be received.

A proxy server can also request authentication using the `407 Proxy Authentication Required` response. However, there is no support for one proxy to authenticate another proxy in SIP. Instead, a proxy can establish a secure connection to another proxy using IPSec.

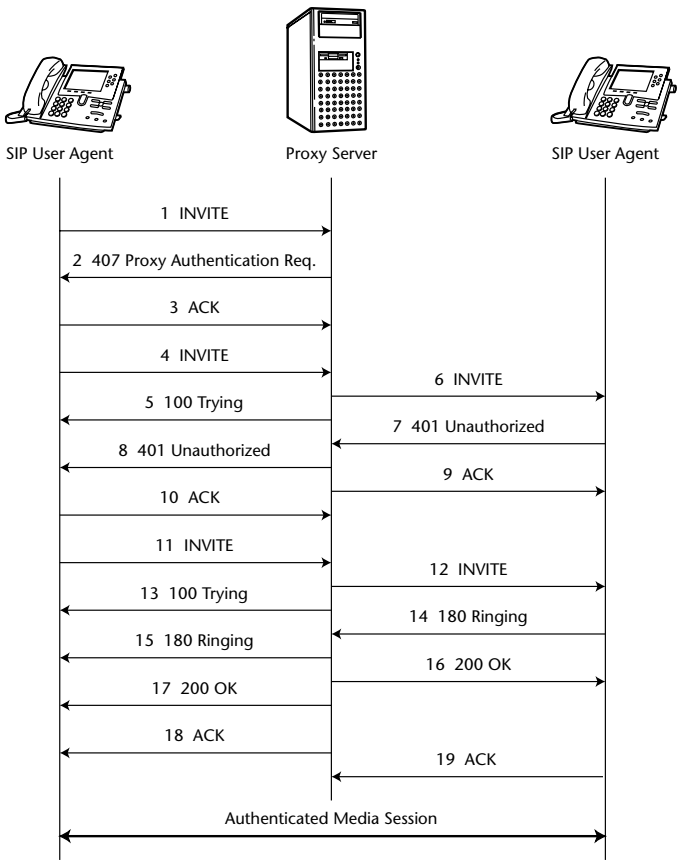


Figure 6.12 Presence publication

An example SIP digest authentication exchange is shown in Figure 6.13. The initial `INVITE` message has no authorization credentials and has received a `407 Proxy Authorization Required` response from the proxy, which contains a `Proxy-Authorization` header describing the nature of the challenge. After sending an `ACK` to the proxy, the user agent then resends the `INVITE` with an `Authorization` header containing the encrypted username and password of the user. The proxy then accepts the credentials, sends a `100 Trying` response, and forwards the request to the destination user agent. The user agent then launches its own authentication challenge with a `401 Unauthorized` response. This response is proxied back to the calling user agent. The SIP user agent then finally does the right thing and resends the `INVITE` request containing both the `Proxy-Authorization` with the credentials for the proxy and `Authentication` header with the credentials for the other user agent.

Following are the details of Message 11, which contains both sets of credentials:

```
INVITE sip:userb@there.com SIP/2.0
Via: SIP/2.0/UDP 4.3.2.1
To: User B <sip:userb@there.com>
From: User A <sip:usera@here.com>
Call-ID: a5-32-43-12-77@4.3.2.1
CSeq: 3 INVITE
Proxy-Authorization: Digest username="usera",
 realm="SIP Telephone Company", nonce="814f12cec4341a34e6e5a35549"
 opaque="", uri="sip:proxy.sip.com", response="6131d1854834593984587ecc"
Authorization: Digest username="A", realm="userb",
 nonce="e288df84f1cec4341ade6e5a359", opaque="",
 uri="sip:userb@there.com", response="1d19580cd833064324a787ecc"
Contact: sip:usera@here.com
Content-Length: ...
```

In this way, SIP supports both network (server) and user (user agent) authentication within a call.

Extensibility

The SIP protocol was designed to be extensible. As a consequence, the protocol was designed so that user agents could implement new extensions using new headers and message bodies without requiring intermediate servers such as proxies to also support the extensions. By default, a proxy forwards unchanged unknown request types and headers. The use of the `Supported` header allows a requestor to inform the network and the other user agent of which extensions and features it supports, allowing them the option of activating the feature. If it is required that the feature be understood or activated, there is a `Require` header [30], which is included in a request. A user agent receiving such a request must return an error if it does not understand or support the feature.

There is also a `Proxy-Require` header that lists features that any proxies in the path must support. However, the use of this header is discouraged, since its overuse will lead to call failures and interoperability problems.

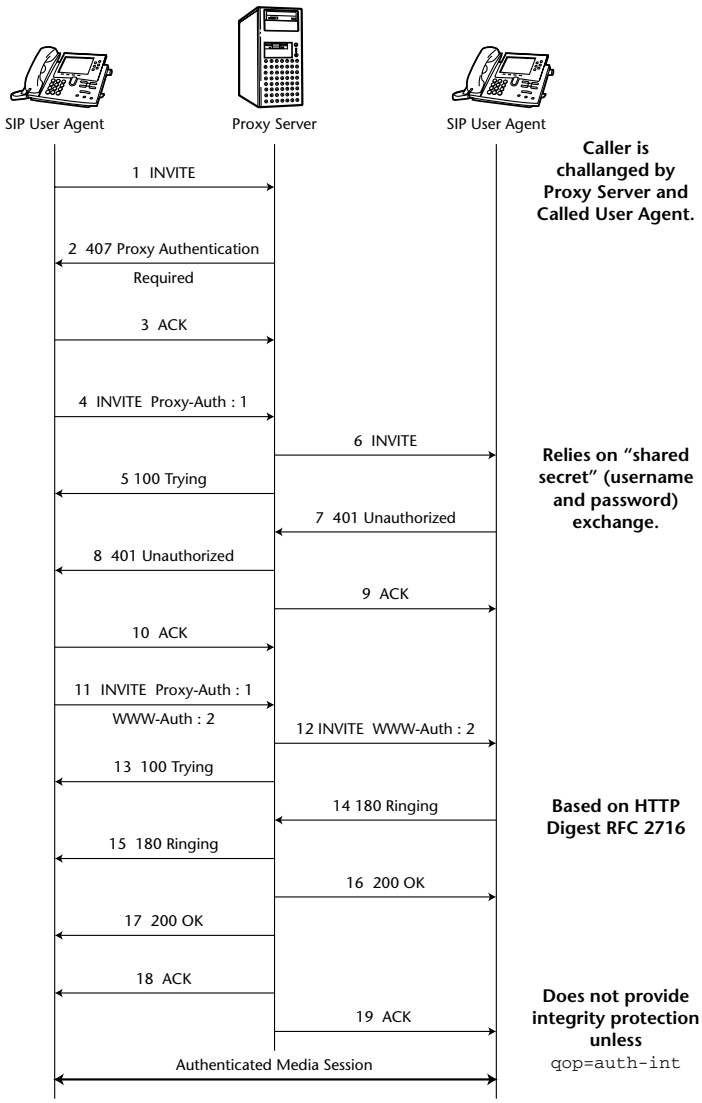


Figure 6.13 Proxy and user authentication example using SIP Digest

SIP user agents should also indicate which methods and features they support using the `Allow`, `Supported`, `Allow-Events`, and `Accept-Content` header fields.

Summary

The basic operation and functions of the SIP protocol have been covered in this overview chapter. The following chapters will use these basic functions of SIP to build networks and implement services and features.

References

[1] "Guidelines for Authors of Extensions to SIP" by J. Rosenberg. Internet Draft, IETF, November 2002.
[2] "Hypertext Transfer Protocol — HTTP/1.1" R. Fielding, et al. IETF RFC 2616, 1999.
[3] "Simple Mail Transfer Protocol" by J. Postel. IETF RFC 821, 1982.
[4] "Session Timers in SIP" by S. Donovan and J. Rosenberg. Internet Draft, IETF, February 2004.
[5] "The Internet Multimedia Conferencing Architecture" by M. Handley, J. Crowcroft, C. Borman, and J. Ott. IETF Internet-Draft, Work in Progress, July 2000.
[6] "SIP: Session Initiation Protocol" by J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. RFC 3261, June 2002.
[7] *SIP: Understanding the Session Initiation Protocol* by A. Johnston, 2nd Edition, Artech House: Boston, 2004.
[8] "RFC 1121: Requirements for Internet Hosts — Communication Layers" by R. Braden, IETF, 1989.
[9] "A DNS RR for specifying the location of services (DNS SRV)" by A. Gulbrandsen. IETF RFC 2782, 2000.
[10] "E.164 number and DNS," P. Faltstrom. IETF RFC 2916, 2000.
[11] "SCTP as a Transport for SIP" by J. Rosenberg and H. Schulzrinne. IETF Internet-Draft, Work in Progress, November 2004.
[12] "SDP: Session Description Protocol" by M. Handley and V. Jacobson. IETF RFC 2327, 1998.
[13] "Requirements for Session Description and Capability Negotiation" by D. Kutscher, et al. IETF Internet Draft, Work in Progress.
[14] "The SIP `INFO` Method" by S. Donovan. IETF RFC 2976, 2000.
[15] "The Session Initiation Protocol (SIP) Refer Method" by R. Sparks. RFC 3515, April 2003.

[16] "AAA Usage for IP Telephony with QoS" by H. Sinnreich, D. Rawlins, A. Johnston, S. Donovan, and S. Thomas. Internet Draft, Internet Engineering Task Force, 2000, http://www.aaaarch.org/pittsburgh/sinnreich/index.htm.

[17] "OSP Authorization Token Header for SIP" by A. Johnston, D. Rawlins, H. Sinnreich, and S. Thomas. Internet Draft, Internet Engineering Task Force, 2000.

[18] "QoS and AAA Usage with SIP Based IP Communications" by G. Gross, H. Sinnreich, and D. Rawlins. Internet Draft, Internet Engineering Task Force, 2000, http://www.iptel.org/ietf/aaa/draft-gross-cops-sip.01.txt.

[19] "RSVP Extensions for Policy Control" by S. Herzog. IETF RFC 2750, 2000.

[20] "An Architecture for Differentiated Services" by S. Blake., D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. IETF RFC 2475, 1998.

[21] Information about PacketCable is available at: www.packetcable.com.

[22] "SIP: Session Initiation Protocol" by J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. RFC 3261, June 2002.

[23] "Reliability of Provisional Responses in Session Initiation Protocol (SIP)" by J. Rosenberg and H. Schulzrinne. RFC 3262, June 2002.

[24] "Architectural Considerations for Providing Carrier Class Telephony Services Utilizing SIP-based Distributed Call Control Mechanisms" by W. Marshall, et al. IETF Internet-Draft, Work in Progress.

[25] "Caller Preferences for the Session Initiation Protocol (SIP)" by J. Rosenberg, J., Schulzrinne, H., and P. Kyzivat. RFC 3841, August 2004.

[26] "Session Initiation Protocol (SIP) Extension for Instant Messaging" by B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, and D. Gurle. RFC 3428, December 2002.

[27] "Session Initiation Protocol (SIP)-Specific Event Notification" by A. Roach. RFC 3265, June 2002.

[28] "The PINT Service Protocol: Extensions to SIP and SDP for IP Access to Telephone Call Services" by S. Petrack and L. Conroy. IETF RFC 2848, 2000.

[29] "Automatic Call Back Service in SIP" by A. Roach. IETF Internet Draft, Work in Progress.

[30] "Security Architecture for the Internet Protocol" by S. Kent and R. Atkinson. IETF RFC 4301, 2006

CHAPTER 7

SIP Service Creation

A major driver for many service providers adopting SIP is the advantages and flexibility of service creation using the protocol. Some of the typical approaches will be discussed in this chapter, including server implementation, called user agent implementation, and calling user agent implementation. Call Processing Language (CPL) and SIP Common Gateway Interface (CGI) will also be introduced in this chapter. The various options for service creation, such as CPL, CGI, SIP Java Servlets, Java Integrated Network (JAIN), and Voice Extended Markup Language (VoiceXML) will also be discussed.

Services in SIP

The basic functions of the SIP protocol involved in establishing sessions between two endpoints over the Internet was discussed in Chapter 6, "SIP Overview." This chapter discusses implementations of additional functionality in relation to session establishment, henceforth referred to generically as "services." A classic example of a telephony service is call forwarding, which results in an endpoint being contacted that is different from the one that was dialed.

More advanced services can be implemented using SIP than can be implemented in the PSTN because of the increased amount of signaling information available during a call setup in SIP. Many of these advanced features and

services will include integration with the World Wide Web or other databases of information. However, the first set of services implemented using SIP will be PSTN telephony features, which are used as examples throughout this chapter.

Services can reside in a number of locations in SIP. For example, many services can reside exclusively in user agents, requiring no support or servers in the network. Intelligent phones such as those shown in Figure 2.6 can well support a variety of SIP services. Other services can reside in proxy or redirect servers. The following simple example illustrates the implementation options in SIP.

Service Example

Consider a call-forward, no-answer service, in which a user wants an unanswered call to his or her SIP phone to automatically forward to a voicemail server after a certain period of time or a certain number of "rings." This service could be implemented in either a proxy server, called user agent, or calling user agent.

Server Implementation

This service could be implemented in the proxy server that handles registrations for the called party. The resulting flow is shown in Figure 7.1. The proxy starts a timer when the `INVITE` is proxied to the latest registered address for the SIP phone:

```
sips:alan@office51.example.com
```

Since the call is not answered (no `200 OK` is sent by the phone), the proxy sends a `CANCEL` to stop the phone from ringing, then forks the `INVITE` to the voicemail server, rewriting the `Request-URI` to the following:

```
sips:alan-msg-deposit-external@voicemail.example.com
```

The voicemail system answers, plays a prompt, and records a message on behalf of the called party.

If a SIP server wishes to provide services beyond the initial call setup (`INVITE/200 OK/ACK` exchange), the proxy must insert a `Record-Route` header into the `INVITE` request. This ensures that all future requests, such as re-`INVITE`s and other methods, will be routed through the proxy, giving the proxy an opportunity to invoke a service.

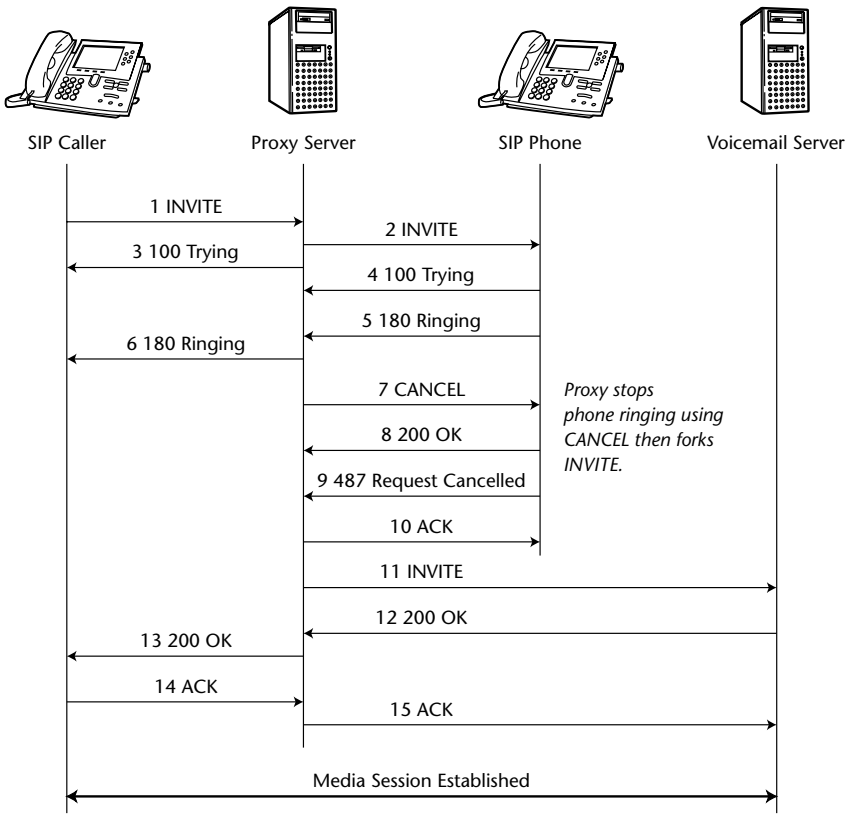


Figure 7.1 Call-forward, no-answer service implemented by the proxy server

Called User Agent Implementation

Figure 7.2 shows how the same feature can be implemented in the called SIP phone. In this case, the ring-no-answer timer is started in the called SIP user agent. When the timer expires, the phone sends a redirection response:

```
302 Moved Temporarily
Contact: <sips:alan-msg-deposit-external@voicemail.example.com>
```

This causes the calling SIP phone to generate an `ACK` to the called SIP phone then generate a new `INVITE` directly to the voicemail server, which then answers, plays a prompt, and records a message.

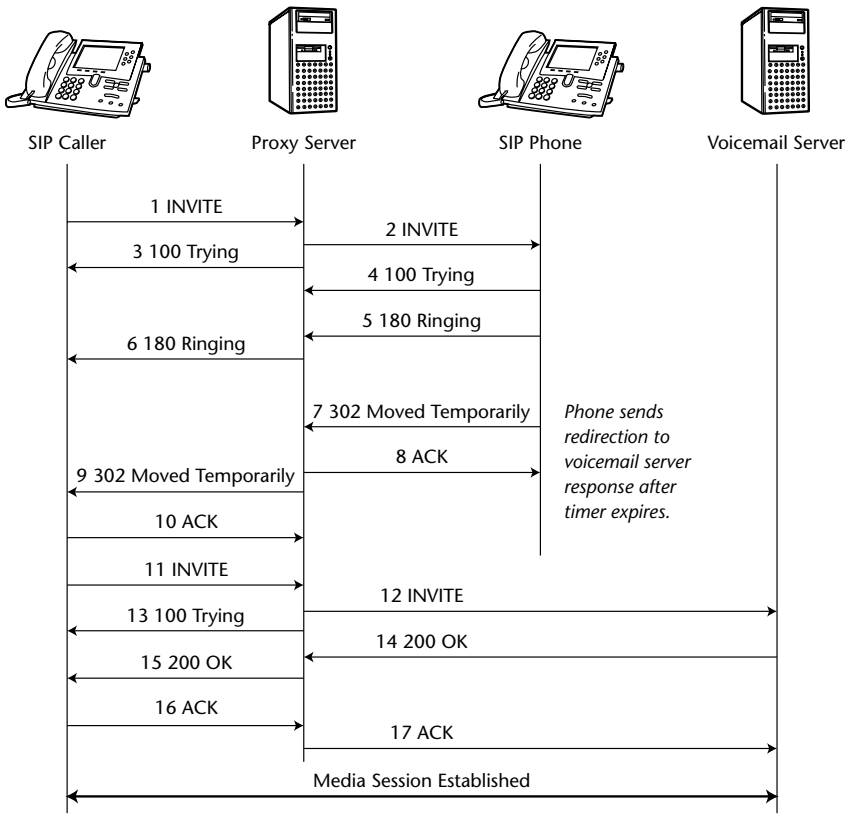


Figure 7.2 Call-forward, no-answer service implemented by the called user agent

Calling User Agent Implementation

Finally, Figure 7.3 shows how this same feature can be implemented in the calling SIP phone. In this case, the SIP server redirects, instead of proxying the `INVITE`:

```
302 Multiple Choices
Contact: <sips:alan@office51.example.com>
Contact: <sips:alan-msg-deposit-external@voicemail.example.com>
 ;actor=msg-taker;automata
```

The caller then sends an `INVITE` to the first URI. After the ring-no-answer timer expires in the calling user agent, the caller sends a `CANCEL`, then sends a new `INVITE` to the voicemail server. This second `Contact` header shows the use of feature tags (covered in Chapter 8, "User Preferences") in specifying that the URI is that of a voicemail server.

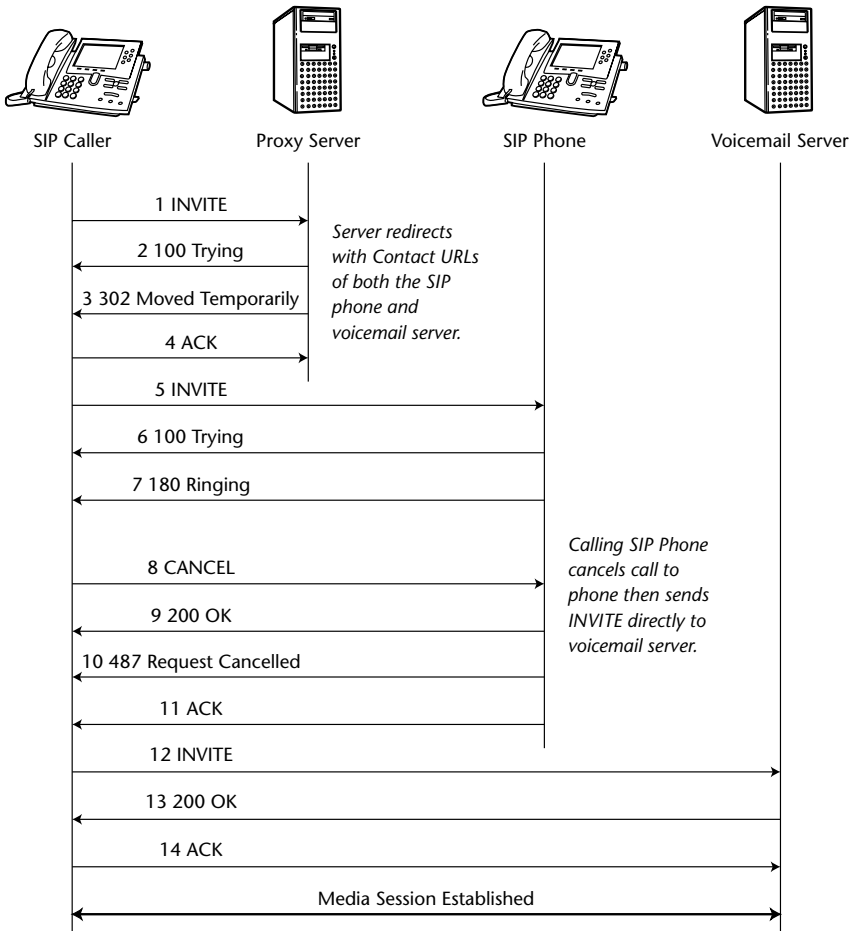


Figure 7.3 Call-forward, no-answer service implemented by the calling user agent

The construction of the voicemail URI also shows the method of using an "opaque" URI [1] to indicate to the voicemail server the intention of the `INVITE`. In this example, the user portion of the URI contains the username `alan` and also the keyword `msg-deposit-external`, indicating that this is a message deposit session. This indicates to the voicemail server to play an external greeting and record a message. Another URI possibility would be `alan-msg-retrieval`, which could indicate message retrieval. In this case, the voicemail server would authenticate the caller for appropriate credentials, and then play back messages to the caller. This is shown in Figures 7.1, 7.2, and 7.3 using different design options for comparison.

Comparison

The advantages and disadvantages of these three implementations are summarized in Table 7.1. In summary, each implementation shifts the location of the service logic (such as the ring-no-answer timer and the recursive retries).

The implementation of a particular service will depend on many factors and may be influenced by economies of scale.

In addition to these three common methods, there is a fourth method of service implementation that involves third-party call control. This approach has been generalized to an architecture of special proxies that modify SIP messages (headers and message bodies), and generate and respond to requests. This is covered in detail in Chapter 19, "SIP Component Services."

Table 7.1 Comparison of Service Implementation

SERVICE IMPLEMENTATION	ADVANTAGES	DISADVANTAGES
Server	Called user agent does not need to be registered. Neither user agent requires any provisioning or special logic.	User must change proxy logic in order to change nature of service. As a result, the service logic is not under the direct control of the user.
Called user agent	Service logic is under control of user in phone configuration.	Feature logic must be in called user agent. User agent must be registered. This means effectively that the called user agent must be "on" or else the service will fail. This type of "24x7" reliability is more difficult to achieve on a customer's premises, as compared to a service provider's centralized location.

Table 7.1 *(continued)*

SERVICE IMPLEMENTATION	ADVANTAGES	DISADVANTAGES
Calling user agent	No logic or provisioning in called user agent, which does not need to be registered. Caller has the choice of connecting to voicemail.	Requires feature logic in calling user agent. Only works if SIP server redirects instead of proxying. Since this is not under the control of the caller, this service will not always work in a reliable way.

New Methods and Headers

New features and services can be implemented in SIP by defining new methods or headers. The basic set of methods and headers are defined in the SIP base specification, which covers basic session establishment and some features and services. New headers and methods can be proposed in the IETF through a process of writing and submitting an Internet draft document. If this document fits the chartered scope of the working group and gathers sufficient support, it may be adopted by the working group as an official work item. The status of the Internet draft is then tracked on the working group charter page as it is discussed and reviewed. Eventually, the document may become an RFC and an official extension to SIP. Nearly all the SIP extensions referenced in this book are RFCs or are official work items of the SIP, SIPPING, or SIMPLE working groups, and are likely to become RFCs in the near future.

It is important to note that new methods or headers do not need support by SIP servers. For example, a SIP proxy receiving a request with an unknown method will proxy the request, treating it as if it were an `OPTIONS` request. A SIP proxy that receives a request with an unknown header will simply proxy the request, making no change to the header. Only the presence of a `Proxy-Require` header will force a proxy to understand and take action based on a particular header or method.

This allows new services to be created in user agents and deployed without any changes in the SIP network. Note that this is essentially what has happened with SIP-enabled telephony networks that can provide SIP instant message and presence transport without any changes to the SIP infrastructure.

Many new methods can be defined without having to use the `Supported` header. For example, a user agent sending an `INFO` request to a user agent that does not support this extension to SIP will receive a `405 Method Not Allowed` (if it recognizes the method but does not support it) or a `500 Bad Request` (if it does not recognize the method) response with an `Allow` header

listing supported methods. However, other extensions, such as reliable provisional responses, need the `Require` header that lists required features of the UAS. If the UAS does not support the feature, the request must be rejected with a `420 Bad Extension` response.

Any extension to SIP that can be referenced using the `Require` or `Supported` header must be fully documented in an RFC, even if it is an informational and not a standards track document. This should prevent vendor proprietary headers and methods from causing interoperability problems in the SIP protocol. The use of headers and methods in SIP that have not been standardized by the IETF is extremely dangerous to interoperability, because these extensions may not be fully documented or may have been rejected by the working group for good reasons. All standardized extensions to SIP must describe how the extensions interact with elements that do not understand the extension.

The next section will describe how the service or feature logic can be scripted or programmed into SIP devices.

Service Creation Options

Just as there are a number of options where service logic can reside in a SIP network, there are many options for the form of the service logic. These scripting and programming options include Call Processing Language (CPL), SIP Common Gateway Interface (CGI), and SIP Servlets.

Call Processing Language

Call Processing Language (CPL) [2] was developed to allow nontrusted end users to upload their services to SIP servers. CPL will be briefly introduced in the following sections, which include some examples of services created using CPL.

Introduction to CPL

CPL was adopted by the IETF IP telephony (IPTEL) working group (WG) as executable code to be run on a SIP proxy server to implement services. CPL is an official work item of the IPTEL WG. CPL is based on Extensible Markup Language (XML) [3], which is a form of Standard Generalized Markup Language (SGML) [4] developed by the W3C.

Readers familiar with Hypertext Markup Language (HTML) [5] who are used to formatting web documents will recognize a similar structure. XML tags have the form `<tag>`, which opens the tag, and then `</tag>`, which

closes the tag. There are, however, some important differences between XML and HTML. In XML, there are strict parsing rules that are defined by the document type definition (DTD) (in this case, `cpl.dtd`) defined in the `<!DOCTYPE>` header. Any discrepancies between the script and the schema must produce an error. In HTML, parsing rules are forgiving. Unknown tags may be silently ignored, while missing required tags may be added. In HTML, some tags do not need to be closed, while in XML every opened tag must be closed.

CPL defines behavior for SIP URIs, tel URIs, and also H.323 URIs. Each action has a specific result for each of these signaling protocols. CPL, like SIP, is a text-based protocol.

Some tags have attributes, in which case they are written as `<tag attribute="value">`. Tags also can have multiple attributes. Tags without any attributes, or nested tags, can be opened and closed in a single tag using `<tag />`, which is equivalent to `<tag></tag>`.

An example CPL script from the RFC 3880 to screen calls from anonymous callers is shown here:

```
<?xml version="1.0" encoding="UTF-8"?>
 <cpl xmlns="urn:ietf:params:xml:ns:cpl"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="urn:ietf:params:xml:ns:cpl cpl.xsd ">
  <incoming>
   <address-switch field="origin" subfield="user">
    <address is="anonymous">
     <reject status="reject" reason="I reject anonymous calls"/>
    </address>
   </address-switch>
  </incoming>
 </cpl>
```

In this example, the first tag indicates the version of XML. The second tag begins the CPL script, lists XML namespace (`urn:ietf:params:xml:ns:cpl`), and defines the schema, which supplies the parsing rules for the document. Everything until the `</cpl>` tag is the CPL script itself. The next tag `<incoming>` indicates that this defines behavior for incoming calls, not outgoing ones. The next tag is `<address-switch>`, which is a type of switch or decision point. This switch specifies that the username part of the origin address (`From` header) is the value being tested. The `<address>` tag with the attribute `is="anonymous"` means that the username portion is anonymous. The `<reject>` tag with the attributes `status` and `reason` indicates that a call that matches this switch (`user = "anonymous"`) should be rejected by the server. The rest of the tags simply close the opened tags. The complete set of CPL tags is listed in Table 7.2.

Table 7.2 CPL Tag Summary

TAG	DESCRIPTION
`cpl`	Begins the CPL script
`incoming`	Defines server operation for an incoming call
`outgoing`	Defines server operation for an outgoing call
`location`	Defines a URI location
`lookup`	Defines action based on result of lookup
`remove-location`	Removes a URI location from a set
`proxy`	Causes call to be forwarded (proxied) to the set of locations specified
`redirect`	Causes call to be redirected to the set of locations specified
`reject`	Causes call to be rejected
`mail`	Causes an e-mail notification to be sent to the specified e-mail address
`log`	Causes the server to log the specified information about the call
`subaction`	Defines a subaction, which can then be referenced in the script using the `sub` tag
`sub`	Causes server to execute the defined subaction script
`language-switch`	Choices or decision points based on language
`address-switch`	Choices or decision points based on address (`From` header)
`string-switch`	Choices or decision points based on a string
`time-switch`	Choices or decision points based on time of day
`priority-switch`	Choices or decision points based on priority of request (`Priority` header)
`ancillary`	Unused—available for future extensions

CPL has switches defined for address, string, time, language, and priority. Each of these has a number of attributes, including fields and subfields. The matching rules include `is`, `contains`, and `subdomain-of`. The complete set of switches listing the matching conditions, fields, and subfields is shown in Table 7.3.

Table 7.3 CPL Switch Types

SWITCH TYPE	MATCHES	FIELDS	SUBFIELDS
address	is	origin	address-type
	contains	destination	user
	subdomain-of	original-	host
		destination	port
			display
string	is	subject	
	contains	organization	
		user-agent	
		language	
		display	
time	dtstart		
	dtend		
	duration		
	freq		
	interval		
	until	tzid	
	byday	tzurl	
	bymonthday		
	byyearday		
	byweekno		
	bymonth		
	wkst		
priority	less		
	greater		
	equal		

Example of CPL Scripts

The following example from the CPL RFC 3880 shows a CPL script implementing a call-forward, no-answer, and busy to voicemail.

```
<?xml version="1.0" encoding="UTF-8"?>
   <cpl xmlns="urn:ietf:params:xml:ns:cpl"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="urn:ietf:params:xml:ns:cpl cpl.xsd ">
     <subaction id="voicemail">
       <location url="sip:jones@voicemail.example.com">
         <redirect />
       </location>
     </subaction>
     <incoming>
       <location url="sip:jones@phone.example.com">
         <proxy timeout="8">
           <busy>
             <sub ref="voicemail" />
           </busy>
           <noanswer>
             <address-switch field="origin">
               <address is="sip:boss@example.com">
                 <location url="tel:+19175551212">
                   <proxy />
                 </location>
               </address>
               <otherwise>
                 <sub ref="voicemail" />
               </otherwise>
             </address-switch>
           </noanswer>
         </proxy>
       </location>
     </incoming>
   </cpl>
```

In this script, two subactions are defined at the start. The first defines the voicemail `voicemail` subaction, in which the server redirects the call to the voicemail server with the `Request-URI` `sip:jones@voicemail.example.com`.

The script operation begins with the `<incoming>` tag. The `<address-switch>` tag checks to see if the caller is part of the `example.com` domain or a different domain. If the caller is internal to the `example.com` domain, the call is proxied to the URI `sip:boss@example.com`. If the result of that proxy is busy, failure, or no answer, the call is then processed by the `voice mail-internal` subaction, which proxies the call to voicemail. For external callers, the call is immediately sent to voicemail.

Note that CPL also can be used for service creation for outgoing calls. Consider the following example, also taken from the CPL RFC 3880:

```
<?xml version="1.0" encoding="UTF-8"?>
   <cpl xmlns="urn:ietf:params:xml:ns:cpl"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="urn:ietf:params:xml:ns:cpl cpl.xsd ">
     <outgoing>
       <address-switch field="original-destination" subfield="tel">
         <address subdomain-of="1900">
           <reject status="reject"
               reason="Not allowed to make 1-900 calls."/>
         </address>
       </address-switch>
     </outgoing>
   </cpl>
```

In this example, any telephony URIs that begin with 1-900- ... are rejected by the server because they might be 900-number toll calls.

SIP Common Gateway Interface

The SIP CGI is analogous to HTTP CGI used for web server service creation. SIP CGI is defined by an informational RFC [6], which means that it is not a standards track protocol. For example, web page forms are usually implemented using HTTP CGI scripts. In a similar way, complex services can be programmed under control of network administrators using SIP CGI. SIP CGI runs on a SIP server that interacts with a program containing the service logic using the CGI interface. This arrangement is shown in Figure 7.4.

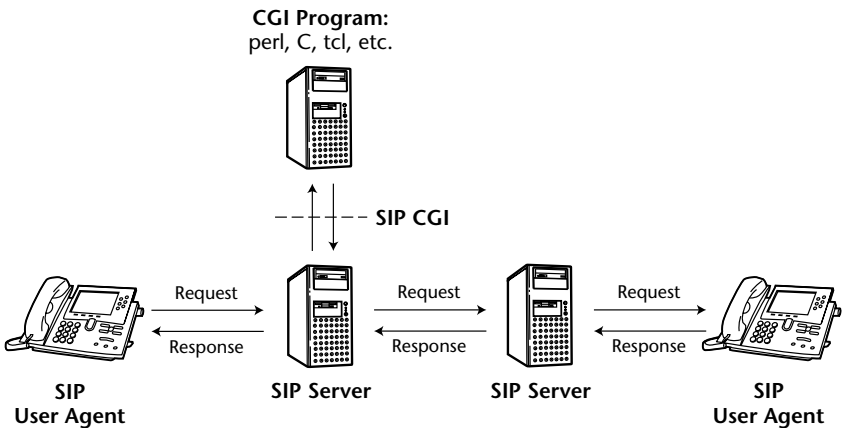


Figure 7.4 SIP CGI model

SIP CGI is an interface, not a programming language. It allows services to be developed in familiar languages such as Perl, C, Tolol Command Language (Tcl), and so on. Because of the similarities, SIP CGI can reuse most HTTP CGI codes. Unlike HTTP CGI that deals exclusively with generating responses to requests, SIP CGI can be used to generate responses and also can cause the server to proxy requests to other locations. SIP CGI scripts are *call-stateful*, in that they can correlate multiple requests corresponding to the same SIP session. This allows a wide spectrum of SIP services to be developed using SIP CGI. So that the server does not have to execute the script for every SIP request, SIP CGI scripts allow the specification of a "default" action, and the conditions under which this default action is executed.

RFC 3050 lists more than 20 metavariables that can be used by the CGI script. In addition, the CGI script has complete access to the request SIP headers and message bodies.

For example, an `INVITE` request could generate the following SIP CGI response to the SIP server:

```
CGI-PROXY-REQUEST sip:j.customer@carrier.com SIP/2.0
Organization: MegaCarrier

SIP/2.0 100 Trying

CGI-SCRIPT-COOKIE hfkelwoeih SIP/2.0
```

The first line tells the server to proxy the request to the specified URI using the metavariable `CGI-PROXY-REQUEST`. The second line tells the proxy to insert the `Organization` header into the request. (The use of uppercase for SIP CGI metavariables allows them to be easily distinguished from SIP headers.) Note that the proxy knows to add a `Via` header and to do the normal operations associated with proxying a SIP request—the script does not need to tell the server this. The third line tells the proxy to send a `100 Trying` response back to the caller. The fourth line tells the SIP server to store a cookie associated with this call at the SIP server using the `CGI-SCRIPT-COOKIE` metavariable. If the script is reactivated for this call, this cookie would be returned to the CGI program, allowing it to operate statelessly but still track the progress of the session.

SIP Application Programming Interfaces

A number of SIP application programming interfaces (APIs) have been developed, including SIP servlets and JAIN. The use of APIs offers the possibility of lower overhead than CPL and SIP CGI, since an external process does not need to be spawned each time. API capabilities for storing state and timers are also simpler than SIP CGI. However, a major disadvantage is that this approach is language-dependent.

SIP Servlets

SIP Java servlets [7, 8] are a powerful tool for extending the functionality of a SIP client by allowing it to pass received messages to SIP servlets. SIP servlets can then process the message and even interact with the SIP client to generate new messages (if the security settings allow it). This API can be used on both SIP servers and user agents. It is not a general-purpose SIP API but rather an API for service extensions. SIP servlets are currently not a work item of any IETF working group, but may become an informational RFC in the future.

JAIN

The Java Integrated Network (JAIN) SIP specification [9] is part of an effort to create a set of APIs for various telephony and Internet protocols for service development. The JAIN SIP specification provides a standard interface to proprietary vendor SIP stacks. JAIN is defined by various Java documents that are not related to the IETF in any way.

SIP and VoiceXML

Voice Extensible Markup Language (VoiceXML) [10] has been developed to enable simple voice-enabled services and features to be developed. VoiceXML is defined by documents at the VoiceXML consortium, which is not related to the IETF in any way. Like CPL, VoiceXML is based on XML and has a similar structure. VoiceXML scripts play prompts (either using prerecorded or synthesized speech), collect input (via DTMF tones or speech recognition), and take specified actions based on results. While VoiceXML does not relate directly to SIP, a VoiceXML script can be run in conjunction with a SIP CPL or CGI script to implement a complete interactive service. An example VoiceXML script to prompt a caller as to whether he or she wishes to be connected to a voicemail server is shown here:

```
<vxml>
  <form id="message">
    <field name="choice">
      <prompt>
        <audio>Do you want to be connected to voicemail?
               Say yes or no.</audio>
      </prompt>
      <grammar>
      <![CDATA[
      [
        [yes] {<option "yes">}
        [no] {<option "no">}
      ]
      ]]>
```

```
        </grammar>
      </field>

      <filled>
        <result name="yes">
          <goto next="#proxy_voicemail"/>
        </result>
        <result name="no">
          <goto next="#disconnect"/>
        </result>
      </filled>
    </form>
  </vxml>
```

In this example, the system will play the audio prompt, "Do you want to leave a message? Say yes or no." The grammar is defined to be "yes" or "no," and this is used by the speech-recognition system to make a decision. The SIP service logic will then perform the routing to the voicemail server or disconnect the call, depending on the outcome of the VoiceXML script.

Summary

SIP provides an extremely flexible set of tools for service creation and implementation. The architectures and tools described in this chapter should allow the development of many different services in a SIP-enabled network. The large portfolio of available development tools and the open nature of these tools will enable the development of many domain-specific communication services by third-party developers. For example a courier or transportation company may develop a Presence-based communication application that allows tracking and contacting any of the fleet workforce according to certain criteria that are business-specific for the company. Future developments of SIP are discussed in Chapter 21.

References

[1] "Framework for SIP Call Control Extensions" by B. Campbell. IETF Internet draft, work in progress, March 2001.

[2] "CPL: A Language for User Control of Internet Telephony Services" by Lennox and H. Schulzrinne. IETF RFC 2824, 2001.

[3] "Extensible Markup Language (XML) 1.0 (2nd edition)" by T. Bray, J. Paoli, and C.M. Sperberg-McQueen.W3C Recommendation REC-xml-20001006, World Wide Web Consortium (W3C), October 2000.

[4] "Information Processing —Text and Office Systems —Standard Generalized Markup Language (SGML)" by ISO (International Organization for Standardization), ISO Standard ISO 8879:1986(E), International Organization for Standardization, Geneva, Switzerland, Oct. 1986.
[5] HTML information is available at the W3C (World Wide Web Consortium) Web page at: `http://w3.org/MarkUp`.
[6] "Common Gateway Interface for SIP" by J. Lennox, H. Schulzrinne, and J. Rosenberg. IETF RFC 3050, 2001.
[7] "The SIP Servlet API" by A. Kristensen and A. Byttner. Internet Draft, Internet Engineering Task Force, September 1999, expired.
[8] "SIP Servlet API Extensions" by K. Peterbauer, et al. IETF Internet Draft, 2000, work in progress.
[9] JSR-000032 JAIN(TM) SIP Specification is available at `http://java.sun.com/products/jain`.
[10] VoiceXML information is available at the VoiceXML Forum web page at: `http://voicexml.org`.

CHAPTER

8

User Preferences

Any advanced network must be flexible enough to take into consideration the preferences and desires of users. In this chapter, we will show how SIP can use the preferences of both the caller and the called party in call routing, features, and services.

Introduction

Telephony services based on the intelligent network architecture for public networks and private circuit-switched networks (PBXs) give the users, in general, little or no control over the preferences of how calls should be handled. Whatever call features are possible can only be subscribed to, but cannot be exercised as individual preferences on a call-by-call basis. There are many reasons, one of them being the frugal user interface of user devices (called *terminals* in ITU standards language) and the general concept of user devices not being the location for intelligence. Another reason is scalability. It is more difficult to store a page full of user preferences for millions of users in central servers of the Intelligent Network (IN) in the PSTN, and also have the data changed by users on a dynamic basis, as compared to having such data and access to it handled at the periphery of the network.

IP communications, by contrast, consider the intelligence, and control resides primarily in user devices. As a consequence, dynamic user preferences can be fully enabled on a scalable basis, no matter how many users are on a network. User preferences are well documented for SIP [1]. The methods of specifying caller preferences are documented in [2].

Following are some examples of caller and called party preferences:

- Call someone, but speak only to voicemail, so as to shorten the call as much as possible.
- Receive calls only from certain parties at certain times (such as accepting calls during lunch hour only from the spouse or the boss) and sending all other calls to voicemail.
- Specify certain times/dates to be accessible only on the mobile phone or at hotel phone numbers when traveling.
- Specify instant text messaging only when in a meeting or at the theater.

There are three parties to user preferences, each having very different roles and using different technologies:

1. *Caller preferences*—Since the caller is the active party, it can express clearly the preferences for the call at call setup by three SIP headers, as will be shown later.
2. *Called party preferences*—The called party is passive, since it has to wait for incoming calls and it cannot anticipate all possible preferences of callers, but can formulate clear rules how incoming calls should be handled. Such rules can be expressed in CPL scripts that reside either in the designated proxy server that handles the calls or in the user agent.
3. *Server support for user preferences*—SIP servers can be designed to understand and process caller preferences, and also to execute scripts with rules for incoming calls. SIP servers can, however, also enforce policy rules for communications on behalf of the network administrator.

Preferences of Caller

The caller can request servers proxy or redirect a call, and also specify how to search for the destination. The instructions are carried in the `Request-Disposition` header. For example:

```
Request-Disposition: proxy, parallel, queue
```

The instructions in the `Request-Disposition` header are explained here. The caller can request the server to:

- *Proxy*—Proxy or redirect the call.
- *Cancel*—Handle `CANCEL` requests on behalf of the caller or let the caller do it.
- *Fork*—Fork call requests to different URIs.
- *Recurse*—When receiving addresses to redirect the call to, the server should try the new addresses, or return them to the caller to make the decision to try again.
- *Parallel*—Try multiple addresses in parallel, or in sequential order, and wait for a response before trying the next address.
- *Queue*—Queue the call if the called party is busy and return the provisional response `182 Queued`. Waiting in the queue can be terminated by `CANCEL` or `BYE` requests.

The caller can also express preferences for how certain URIs should be handled by using the headers `Contact`, `Accept-Contact`, and `Reject-Contact`.

User preferences that relate to the same URI can have a range of classes to specify such preferences as:

- *Audio*—To specify that the UA supports audio sessions.
- *Application*—To specify that the UA supports application sessions.
- *Data*—To specify that the UA supports data sessions.
- *Control*—To specify that the UA supports control sessions.
- *Video*—To specify that the UA supports video sessions.
- *Text*—To specify that the UA supports text sessions.
- *Automata*—To specify that the UA is an automata (such as a voicemail server).
- *Class*—To specify the class of the UA (business or residential class). So the caller can avoid calling someone at home.
- *Duplex*—To specify for certain types of lectures or conferences.
- *Mobility*—To set preference for mobile phone or fixed phone.
- *Description*—To specify a description of the UA.
- *Event Packages*—To specify which SIP event packages the UA supports.

- *Priority*—To specify the priority of the request.
- *Schemes*—To specify which URI schemes the UA supports.
- *Extensions*—To specify which SIP extensions the UA supports.
- *Methods*—To specify the capabilities of the UA (such as voice or IM).
- *Actor*—To express the role a UA performs. Example roles that are defined in RFC 3840 are `principal` (direct communication with the person), `attendant` (indirect communication through a third person), `msg-taker` (a message will be taken and delivered to the principal), and `information` (information about the principal is available).
- *Is Focus*—To specify that the UA is a conferencing server or focus.

The following are examples of these classes.

Example for Contact

The caller would specify in the `REGISTER` message or an `INVITE` message:

```
Contact: HenryS <sip:henry@pulver.com>;audio;video
 ;text;duplex="full";priority="urgent"
```

The preferences include audio, video, and text chat in full duplex with a specification for urgency.

Example for Accept-Contact

The caller would like to speak to a UA that supports the SIP `MESSAGE` method for page mode instant messaging, and is a business device. The degree of preference is indicated by the weight factor `q`.

```
Accept-Contact: *;methods="MESSAGE";class="business";q=1.0
```

Example for Reject-Contact

The caller would not like to communicate with a voicemail server or a device with video.

```
Reject-Contact: *;actor="msg-taker";video
```

The `Reject-Contact` field can also contain a list of URIs for which no call setup is desired.

For more examples and use cases of caller prefs, see [3].

Preferences of the Called Party

The preferences of the called party are generally invoked by an incoming proxy server that handles incoming calls for the called party. For example, calls to `sip:henry@pulver.com` will be routed through the SIP proxy server specified in the DNS SRV records for the `pulver.com` domain.

Since this server will be making decisions on the called party's behalf, a mechanism has been developed in SIP for a user to upload preferences and services into a SIP proxy server. This mechanism is the `REGISTER` message. It is the means for specifying preferences and services using CPL [3], as introduced in Chapter 7, "SIP Service Creation."

Many of the switches in CPL use the Caller Preferences parameters in the `Contact` headers of the caller's `INVITE` and the called party's `REGISTER`.

Server Support for User Preferences and for Policies

Servers can use the `Contact`, `Accept-Contact`, and `Reject-Contact` headers to make the following decisions:

- Should it proxy or redirect the request?
- Which URIs to proxy or redirect to.
- Should it fork the request?
- How to search (recursively or not), or to search in parallel or sequentially.

Administrative policies can also be exercised at the server to exclude, for example, certain URIs or to exclude video for certain callers to conserve bandwidth.

Summary

This chapter has shown how the combination of SIP caller preferences and CPL scripting provide a powerful capability for processing calls and designing services.

References

[1] "Caller Preferences for the Session Initiation Protocol (SIP)" by J. Rosenberg, H. Schulzrinne, and P. Kizivat. RFC 3841, August 2004.

[2] "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)" by J. Rosenberg, H. Schulzrinne, and P. Kizivat. RFC 3840, August 2004.

[3] "Guidelines for Usage of the Session Initiation Protocol (SIP) Caller Preferences Extension" by J. Rosenberg, and P. Kizivat. Internet Draft, October 2005.

[4] "CPL: A Language for User Control of Internet Telephony Services" J. Lennox and H. Schulzrinne. IETF Internet Draft, work in progress, May 2001.

CHAPTER 9

SIP Security

The Security Considerations Section of RFC 3261 begins with the following:

> *"SIP is not an easy protocol to secure. Its use of intermediaries, its multifaceted trust relationships, its expected usage between elements with no trust at all, and its user-to-user operation make security far from trivial." [1]*

SIP security is tricky, and there are many pitfalls for implementers and service providers. This chapter will summarize some of the risks and threats and point to the various mechanisms that can be used to protect against them. For a more detailed coverage of these points, including an introduction to cryptography and security concepts, see Johnston and Piscitello [2].

Threats

This section will summarize the basic threats to SIP, by looking at two common applications of SIP: session setup, and presence and IM. The following sections will discuss security mechanisms to protect against problems involving them.

Session Setup

The main threats to session setup are described in Table 9.1. They are described in terms of their impact on a single SIP user. A similar set of threats could be listed as threats against a server or service provider.

Table 9.1 Threats on SIP Session Setup

THREAT	DESCRIPTION	PROTECTION	MECHANISM
Call hijacking	A user "dials" a SIP URI but establishes a session with different user.	Authentication of signaling; identity	Digest, Enhanced Identity
Registration hijacking	Incoming calls to a user are diverted to a third party.	Integrity protection of registration.	auth-int Digest or TLS
Impersonation	A third party impersonates another user in a session.	Identity	Enhanced Identity
Eavesdropping on signaling	A third party tracks and records whom a user is communicating with by monitoring SIP messages.	Confidentiality of SIP	TLS
Eavesdropping on media	A third party tracks and records media sessions by a user.	Confidentiality of RTP	SRTP
Denial of Service	Calls to or from a user are prevented.	IP, SIP, and RTP layer traffic management	Variety of mechanisms
Session disruption	Calls to or from a user are disrupted after they are established.	Integrity	Secure SIP
Bid-down attack	Calls to or from a user are forced to use a lower level of security by an attacker.	Integrity protection; not supporting low-security modes of communication	Secure SIP

Presence and IM

The main threats to presence and IM are described in Table 9.2. They are described in terms of their impact on a single SIP user. A similar set of threats could be listed as threats against a server or service provider.

Table 9.2 Threats on SIP Presence and IM

THREAT	DESCRIPTION	PROTECTION	MECHANISM
Instant message session hijacking	An instant messaging session intended for one user is redirected to a third party.	Authentication of signaling; identity	Digest, Enhanced Identity
Presence publication hijacking	An attacker modifies presence publication data or injects false data for a user.	Authentication of signaling; integrity protection of publication.	auth-int Digest or TLS
Presence notification impersonation	An attacker sends false presence notifications about another user.	Authentication of signaling; integrity protection of publication.	Enhanced Identity
Eavesdropping on Presence	A third party tracks and records the presence of a user.	Confidentiality of SIP	TLS
Eavesdropping on Instant Messages	A third party tracks and records IM exchanges between two parties.	Confidentiality of SIP or MSRP	Secured MSRP
Denial of Service	IMs to or from a user or presence publications or notifications are prevented.	IP, SIP, and RTP layer traffic management	Variety of Mechanisms
IM session disruption	Instant messages to or from a user are blocked or deleted.	Integrity	Secure SIP
Bid-down attack	IM sessions to or from a user are forced to use a lower level of security by an attacker.	Integrity protection; not supporting low security modes of communication	Secure SIP

Security Mechanisms

This section will discuss the security mechanisms that can be used to counter against a number of threats.

Authentication

SIP can use a number of Internet *authentication* mechanisms. HTTP Digest authentication, defined in RFC 2617 [3] and described for SIP in Section 22 of RFC 3261, provides a simple way for a server or UA to challenge another UA to produce a shared secret such as a username and password. The use of the Message Digest 5 (MD5) hash algorithm means that the credential (password) is never sent in the clear. Also, if each SIP request is challenged with a unique *nonce* (a one time string used in the MD5 hash calculation), Digest responses cannot be cut from one request and pasted into another request. As such, Digest is a lightweight mechanism that can be used without encryption or confidentiality. An example HTTP Digest exchange is shown in Figure 9.1.

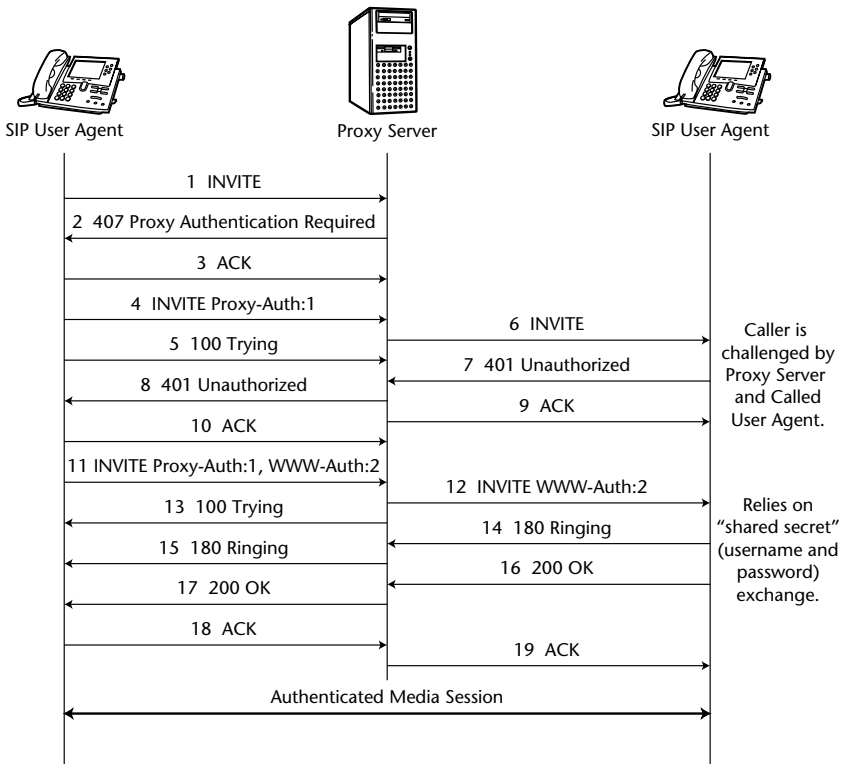


Figure 9.1 Authentication using HTTP Digest

SIP can also use certificates for authentication in the same way that web browsers and servers use them. A *certificate* is a digital document that is issued by a third party, known as a *certificate authority* (*CA*), which makes assertions about a user. For example, a proxy server for the `example.com` domain could use a certificate to assert that it is a valid proxy server for the `example.com` domain. If TLS (Transport Layer Security) [4] is used by SIP, the client can request the certificate of the server. If the certificate received during the TLS Handshake protocol exchange matches the server the UA wishes to talk to, the connection has been authenticated.

Self-signed certificates can also be useful in certain situations. For example, the use of the SIP certificate service [5] allows a UA to generate a self-signed certificate and upload it to a certificate server. A `PUBLISH` with `Event: credential` is used to upload the certificate, while a `SUBSCRIBE` with `Event: credential` is used to retrieve the certificate. Another UA can retrieve the public key of the UA in a `NOTIFY`.

Another use of self-signed certificates is when the fingerprint of a self-signed certificate is exchanged over a secured SIP connection. An SDP attribute extension `a=fingerprint` to do this is defined in [6]. Following is an example from the specification showing a SHA-1 hash of a self-signed certificate:

```
m=image 54111 TCP/TLS t38
c=IN IP4 192.0.2.2
a=setup:passive
a=connection:new
a=fingerprint:SHA-1 \
       4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
```

The TLS connection established with this SDP message will be authenticated using a self-signed certificate that matches the SHA-1 hash in the fingerprint.

Confidentiality

Confidentiality makes a message or communications session private. Encryption can be used to implement confidentiality. If two parties know a secret key, they can use this key to encrypt messages between them so that any third party that does not know the key cannot read the message.

SIP can utilize encryption at any layer. For example, a SIP session over an 802.11 wireless LAN employing Wireless Protected Access (WPA) [7] is confidential. However, if the SIP session extends beyond the LAN, then confidentiality may no longer be ensured.

Encryption at the IP layer with IPSec [8] can also be utilized. IPSec can be established between any two Internet hosts. When used in ESP mode, IPSec provides confidentiality. IPSec is typically performed by hosts at the operating system/kernel level. As a result, it is difficult for an application such as SIP to know if IPSec is in place or not.

Encryption at the transport layer using TLS is visible at the application. Therefore, a SIP UA that attempts to open a TLS connection over TCP to another UA or server will receive a failure message if it is not established. TLS transport is recorded in the `Via` header fields, so the encryption of previous hops can be verified. The use of TLS with SIP is described in Section 26 of RFC 3261.

Secure SIP URI Scheme

However, TLS only provides a single hop of confidentiality and authentication. Since most SIP sessions involve at least one proxy server, there is typically more than one hop between the two communicating UAs. If TLS is used on the first hop but not on the second, then end-to-end confidentiality has not been provided. To solve this problem, RFC 3261 defines Secure SIP.

Secure SIP, which uses the URI scheme `sips`, is used to guarantee end-to-end confidentiality of a SIP session. At each hop, TLS transport must be used or the connection is failed back to the initiator with a `416` error response message. The only exception made is for the very last hop, which may be secured by another mechanism providing confidentiality, besides TLS. For example, IPSec or radio-layer encryption is acceptable for the last or first hop. The use of Secure SIP is shown in Figure 9.2.

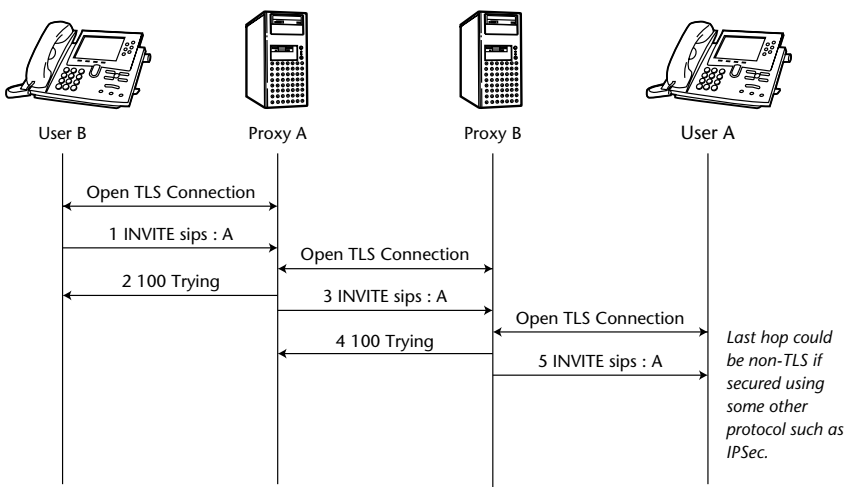


Figure 9.2 Secure SIP

Confidentiality can also be done end to end in SIP using Secure Multipurpose Internet Mail Extensions (S/MIME) [9]. Information in a SIP message body, or selected SIP header fields not required for proxy routing, can be encrypted using S/MIME and carried in the message body. For example, SDP information about the media, including a media key (discussed in the section, "Media Security," later in this chapter) could be encrypted using S/MIME.

Integrity

Integrity allows the recipient of a SIP request to know that the contents of the message have not been modified by a third party. Integrity can be ensured by using a secured hash or by using digital signatures.

Digest authentication provides integrity protection across the method type and the `Request-URI`. However, any other SIP header field, including a critical header field such as `Contact` does not have integrity protection. Digest does have an option that provides integrity protection across the SIP message body. This provides integrity protection over the message body between the UA and the challenging proxy server.

The use of TLS transport provides integrity protection. However, only Secure SIP, which requires TLS over every hop, provides integrity from end to end.

An S/MIME signature can also be used, but only if the other UA requires the presence of the S/MIME body. Otherwise, an attacker could simply modify the SIP request and remove the S/MIME signature body. S/MIME also requires the use of certificates. The UA receiving the request needs to be able to obtain the public key of the sender to verify the signature.

Identity

Identity in SIP means the SIP URI of the user. In receiving a request, a UA can look at the `From` header field and use this as the identity of the requestor. However, how do you know that a value in a `From` header field is accurate? If the UAs share a secret, an authentication challenge along the lines discussed in the earlier section, "Authentication," would serve to validate the identity. However, in most cases, users will not have a shared secret with every other user they may want to establish SIP sessions with.

One way in which identity can be ensured is by policy in an administrative domain. Let's say that all UAs within the `example.com` domain must register and authenticate with the `example.com` proxy server. Each user has a shared secret with the proxy and must produce it each time. Users do not share secrets

with each other, but they only accept requests that come through the `example.com` proxy and, hence, have been authenticated. In this domain, the `From` header can be trusted as a valid identity. Of course, having integrity protection is also required or an identity can be modified.

Within a trust domain, SIP has a mechanism for asserting identity. Known as *network asserted identity*, it uses the `P-Asserted-Identity` [10] header field. Older implementations use a nonstandard header field `Remote-Party-ID`. If a UA receives a request from a proxy server it trusts, then the UA can trust the asserted identity and display it as a calling party ID.

Enhanced SIP identity using the `Identity` header field [11] provides cryptographically verified identity in an interdomain SIP exchange. The `Identity` header field is added by a proxy server after it has authenticated a request and validated the `From` header in the request. The header field contains a cryptographic signature over a subset of SIP header fields, including the `From` URI, `To` URI, `Call-ID`, `Date`, `Contact` URI, and message body. Any proxy server of a UA downstream can validate the signature in the `Identity` header field and validate the `From` identity. Since the signature covers the message body, the `Identity` header field also provides integrity protection over key header fields and the message body, which could contain a media key.

The `Identity-Info` header field contains a URL that allows the public key of the signing proxy server to be easily retrieved. This example from the specification shows a signature and a URL for the server's public key:

```
Identity:"kjOP4YVZXmF0X3/4RUfAG6ffwbVQepNGRBz58b3dJq3prEV4h5GnS4F6udDRC
 rSK9cl+TFv45nu0Qu2d/0WPPOvvc3JWwuUmHrCwGwC+tW7fOWnC07QKgQn40uwg5
 7WaXixQev5N0JfoLXnO3UDoum89JRhXPAIp2vffJbD4="
Identity-Info: <https://atlanta.example.com/atlanta.cer>;alg=rsa-sha1
```

Media Security

Media security is a separate topic from SIP security. However, the topics are related, because SIP can help to establish a secure media session by assisting in the media key exchange.

SRTP

Confidentiality and integrity of RTP media is provided by *Secure RTP* (*SRTP*) [12]. SRTP uses the Advanced Encryption Standard (AES) encryption algorithm with 128- or 256-bit length keys. AES is a symmetric cipher and requires that the key be exchanged or derived using some other protocol. Confidentiality with SRTP is achieved by keeping the key secret. Authentication is an

optional feature with SRTP and is provided with an authenticated hash or HMAC. The use of authentication adds an additional 32 or 80 bits to each SRTP packet. Because of the design of SRTP, the same SRTP master key can be used to secure both directions of a media session. The same key can also secure multiple media streams (such as an audio and video stream or two audio streams).

MIKEY

SRTP keys can be exchanged out of band (for example, shared in a conference invitation). Or, within a small group, a single key could be shared and used for calls within the group. For general SIP use, the SRTP key is exchanged via the SIP signaling. The SDP specification [13] has a `k=` attribute for transporting a media key. However, other information must be exchanged with SRTP (such as the key length, whether an authentication tag is in use, and so on). The Multimedia Internet Keying (MIKEY) protocol [14] has been defined with a profile for SRTP. It operates in a number of modes. MIKEY messages can be carried in SDP in a `a=key-mgt` attribute [15]. MIKEY provides its own integrity and authentication mechanisms. As a result, MIKEY can be used even if the SDP does not have confidentiality. However, MIKEY has a number of possible modes of operation, and the only mandatory mode is the preshared keys mode—the least useful mode of operation. Additional MIKEY modes have also been proposed. The result of this is that the complexity and interoperability of MIKEY has been a problem. If two UAs both support MIKEY but do not support the same mode, a secure session will not be established. An example SIP message is as follows:

```
v=0
o=alice 2891092738 2891092738 IN IP4 lost.example.com
s=Secret discussion
t=0 0
c=IN IP4 lost.example.com
a=key-mgmt:mikey AQAFgM0XflABAAAAAAAAAAAAAAsAyO...
m=audio 39000 RTP/SAVP 98
a=rtpmap:98 AMR/8000
m=video 42000 RTP/SAVP 31
a=rtpmap:31 H261/90000
```

In this example, the same key is used for both audio and video sessions. The use of SRTP is indicated by the use of the Secure Audio Video Profile (SAVP).

SDP Security Descriptions

To overcome some of the complexity issues of MIKEY, the SDP Security Descriptions have been developed [16]. An `a=crypto` SDP attribute carries

both the SRTP key and the SRTP configuration parameters. However, the SDP needs to have confidentiality provided by SIP, or the key will be carried in the clear. End-to-end S/MIME offers the best confidentiality, but hop-by-hop TLS with Secure SIP provides a level of confidentiality (although the secret key will be available to each SIP proxy server in the signaling path). An example is as follows:

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.47.16.5
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf
e=j.doe@example.com (Jane Doe)
c=IN IP4 161.44.17.12/127
t=2873397496 2873404696
m=video 51372 RTP/SAVP 31
a=crypto:1 AES_CM_128_HMAC_SHA1_80
 inline:d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj|2^20|1:32
m=audio 49170 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_32
 inline:NzB4d1BINUAvLEw6UzF3WSJ+PSdFcGdUJShpX1Zj|2^20|1:32
m=application 32416 udp wb
a=orient:portrait
```

In this example, separate SRTP master keys are used for the video and audio streams. In both cases, 128-bit AES encryption is used. For the video stream, an 80-bit HMAC-SHA-1 authentication tag is used. For the audio stream, a 32-bit HMAC-SHA-1 authentication tag is used.

Both the `a=crypto` and `a=key-mgt` approaches have difficulties in falling back to RTP if SRTP is not available. This is because a given media line must be either RTP (`RTP/AVP`) or SRTP (`RTP/SAVP`). There is no way currently in SDP to group two media lines to, say, accept one or the other, but not both. As a result, a common mode will be to initially offer a SRTP session, then fall back to a RTP session after the secure session fails. This is not a very good solution to this all-to-common case in the interim when both secure and nonsecure sessions are common.

New Directions

Security for SIP and related media streams is an area that has received considerable attention over the past few years, and the security mechanisms described in this chapter are, for the most part, well defined and understood. However, there are some new areas of standardization that will likely happen in the coming few years that will be mentioned in this section.

DTLS

The *Datagram TLS* (*DTLS*) [17] transport protocol has been recently standardized in the IETF. DTLS adapts the TLS protocol to work over a datagram transport such as UDP (User Datagram Protocol). As such, it offers many of the advantages of TLS (such as confidentiality, hop-by-hop encryption, and mutual authentication using certificates but without requiring TCP transport). For many communications systems, it is desirable to continue to use UDP transport instead of TCP.

Extensions to SIP to allow the use of DTLS transport will likely be standardized. Also, the use of DTLS transport for RTP has been proposed [18]. If RTP over DTLS is combined with a way to use certificates in UAs, this could provide good authentication for media and signaling sessions, even in peer-to-peer modes.

ZRTP

ZRTP [19] is a new extension to RTP to add integrated key management and SRTP, making it a stand-alone protocol. No longer is there a need to exchange an SRTP master secret out-of-band or in the signaling path. ZRTP does this by performing a Diffie-Hellman key agreement in RTP packets, using RTP's header extension mechanism. To avoid having to utilize certificates for authentication and to prevent man-in-the-middle (MitM) attacks, ZRTP uses a retained shared secret from previous calls. This is similar to the way that the SSH protocol [20] allows a "leap of faith" mode in which the host key is accepted on the first session, then cached for future sessions. In ZRTP, endpoints authenticate each other by retaining and using a secret from a previous ZRTP session. In addition, it is possible to use a spoken voice authentication digest string to prevent a Diffie-Hellman MitM attack.

ZRTP provides better confidentiality than SRTP and SDP Session Descriptions, in which the SRTP master key is available to proxy servers in the path. It provides better interoperability than MIKEY with its many modes and reliance on certificates. It is also simpler to implement in a backward-compatible way than the other approaches. ZRTP simply falls back to RTP when Hello ZRTP messages do not receive a response.

Summary

The mechanisms in this chapter describe how to secure SIP. However, the ultimate security of a device or service is not achieved by securing a single protocol. Rather, it involves a complete system.

References

[1] "SIP: Session Initiation Protocol" by J. Rosenberg, Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, RFC 3261, June 2002.

[2] "Understanding VoIP Security" by A. Johnston and D. Piscitello. Artech House, Boston, MA, 2006.

[3] "HTTP Authentication: Basic and Digest Access Authentication" by J. Franks, P. Hallam-Baker., J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart, RFC 2617, June 1999.

[4] The TLS Protocol version 1.0. RFC 2401 by T. Dierks and C. Allen. IETF, January 1999.

[5] "Certificate Management Service for SIP" by C. Jennings and J. Peterson. Internet-Draft, draft-ietf-sipping-certs-00, (work in progress), October 2004.

[6] "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)" by J. Lennox. IETF Internet-Draft, July 2005.

[7] See the Wi-Fi Alliance at `www.wi-fi.org`.

[8] "Security Architecture for the Internet Protocol" by S. Kent and K. Seo. RFC 4301, IETF, December 2005.

[9] "Enhanced Security Services for S/MIME" by P. Hoffman et al. RFC 2634. IETF, June 1999.

[10] "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks"by C. Jennings et al. RFC 3325. IETF, November 2002.

[11] "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)" by J. Peterson. IETF Internet-Draft, draft-ietf-sip-identity-03, work in progress, September 2004.

[12] "The Secure Real-time Transport Protocol (SRTP)" by M. Baugher et al. RFC 3711. IETF, March 2004.

[13] "SDP: Session Description Protocol" by M. Handley and M. Jacobson. RFC 2327, IETF, April 1998.

[14] "MIKEY: Multimedia Internet KEYing" by J. Arkko et al. RFC 3880. IETF, August 2004.

[15] "Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP)" by J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman. Internet-Draft work in progress, November 2004.

[16] "Session Description Protocol Security Descriptions for Media Streams." F. Andreasen, M. Baugher, and D. Wing. work in progress, February 2005.

[17] "Datagram Transport Layer Security" by E. Rescorla and N. Modadugu. RFC TBD, June 2004.
[18] "Session Initiation Protocol (SIP) for Media Over Datagram Transport Layer Security (DTLS)" by J. Fischl, H. Tschofenig, and E. Rescorla. IETF Internet-Draft, February 2006. Work in Progress.
[19] "ZRTP: Extensions to RTP for Diffie-Hellman Key Agreement for SRTP" by P. Zimmermann and A. Johnston. IETF Internet-Draft, February 2006. Work in Progress.
[20] "SSH Protocol Architecture" by Ylonen, T., and C. Lonvick RFC 4251, January 2006.

CHAPTER

10

NAT and Firewall Traversal

This chapter will discuss how security devices such as firewalls and network address translators (NATs) can complicate SIP call setup signaling and the flow of Real-Time Transport Protocol (RTP) media packets. NATs are used to create private IP networks that use internal IP addresses that are not part of the public Internet address space and are not routed over the Internet [1]. Network administrators use NATs either because they may not have enough public IPv4 addresses or to avoid reconfiguring all their IP devices when they change service providers. However, this has quite a number of undesired consequences, as discussed in Hain [2]. The overall negative implications of firewalls and NATs on Internet transparency are discussed in Carpenter [3]. Since SIP signaling carries rich information, it can reveal valuable personal data of the calling and called parties such as IP addresses (location), contact lists, and traffic patterns.

Firewalls and NATs greatly complicate calls for users in enterprise or home networks that use such devices. Several approaches are possible for firewall and NAT traversal for phone and multimedia communication calls, the most prominent being the following:

- Control of firewalls and NATs from a SIP proxy acting as an Application Level Gateway (ALG)
- Modification of SIP signaling, without changing anything in existing firewalls and NATs

- Modifications to firewalls and NATs so as to make them SIP-aware
- NAT and firewall traversal using peer-to-peer (P2P) techniques such as Interactivity Connectivity Establishment (ICE), Simple Traversal of UDP through NAT (STUN), and Traversal Using Relay NAT (TURN)

Network Address Translators

Network address translators (NATs) are devices that modify the IP address and port numbers, in the case of network address and port translators (NAPTs), of IP packets as they are forwarded from one network to another. NATs are commonly used when a local network utilizes IP addresses that are not globally unique. When an IP packet that originated from this network needs to traverse the public Internet, the use of NATs is required to replace the local addresses with globally routable addresses.

The reason the private address space is not routable is that numerous entities on the public network utilize these addresses on their own internal networks. If these addresses were propagated on the public network, core routers would not know which direction to send the response because of the large number of locations that may utilize the same address space.

NATs are also used sometimes as security mechanisms to hide the internal structure of a local network from users outside the network. For example, internal network topology can be hidden with a NAT by making all internal users appear to be one external, globally unique IP address to the rest of the world. NATs typically operate transparently to the application layer, modifying network layer fields as required to provide this transparency.

Many routers designed for home and small office use incorporate NAT functionality along with a Dynamic Host Configuration Protocol (DHCP) server often bundled with an Ethernet hub in the same device. As devices are plugged into the hub, they are assigned a local IP address (typically assigned from one of the private network address ranges such as 192.168.x.x or 10.x.x.x), which allows them to communicate with other devices on the local area network (LAN). When the packets leave the router, the NAT functionality allows multiple internal PCs or devices to share a single external, globally unique IP address. When used in this fashion, these routers are sometimes called *Internet sharing hubs*.

Some network administrators also use private numbering schemes to avoid having to renumber their networks if they ever have to change Internet service providers (ISPs). Without a NAT, every IP device would need to be readdressed. With a NAT-enabled device, only the NAT device must be reconfigured with a new pool of IP addresses.

Since SIP was developed, guidelines for protocol design to make them more NAT "friendly" have been developed by the IETF [4]. Unfortunately, SIP violates most of these newer guidelines. For example, one of the major recommendations of this document is that application layer protocols should not transport IP addresses and port numbers. The next example shows why this is a major problem for routing SIP and resulting Real-time Transport Protocol (RTP) sessions through a NAT. In this `INVITE` generated from behind a NAT, the fields in bold represent IP addresses that cannot be routed across a globally addressed network such as the Internet.

```
INVITE sip:UserB@there.com SIP/2.0
Via: SIP/2.0/UDP 10.1.1.221:5060;branch=z9hG4bKhjh
From: TheBigGuy <sip:UserA@customer.com>;tag=343kdw2
To: TheLittleGuy <sip:UserB@there.com>
Max-Forwards: 70
Call-ID: 123456349fijoewr
CSeq: 1 INVITE
Subject: Wow! It Works...
Contact: <sip:UserA@10.1.1.221>
Content-Type: application/sdp
Content-Length: ...

v=0
o=UserA 2890844526 2890844526 IN IP4 UserA.customer.com
c=IN IP4 10.1.1.221
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Because of the presence of the NAT:

- The response to this request could not be routed back to the originator because of the inability to route these private network address ranges defined for use on private internal networks (based on an incorrect `Via` header).
- Future requests during this session would be misrouted (based on an incorrect `Contact` header).
- RTP packets sent by user B would be misrouted (based on an incorrect connection IP address `c=` for the media in the Session Description Protocol, or SDP).

Note also that the two port numbers contained in this `INVITE`, port 5060 and port 49170, also may be changed by the NAT and may cause signaling or media exchange to fail.

If the NAT is being used for security purposes, the amount of topology leakage shown in this `INVITE` would not be acceptable to a network administrator, as shown in Figure 10.1.

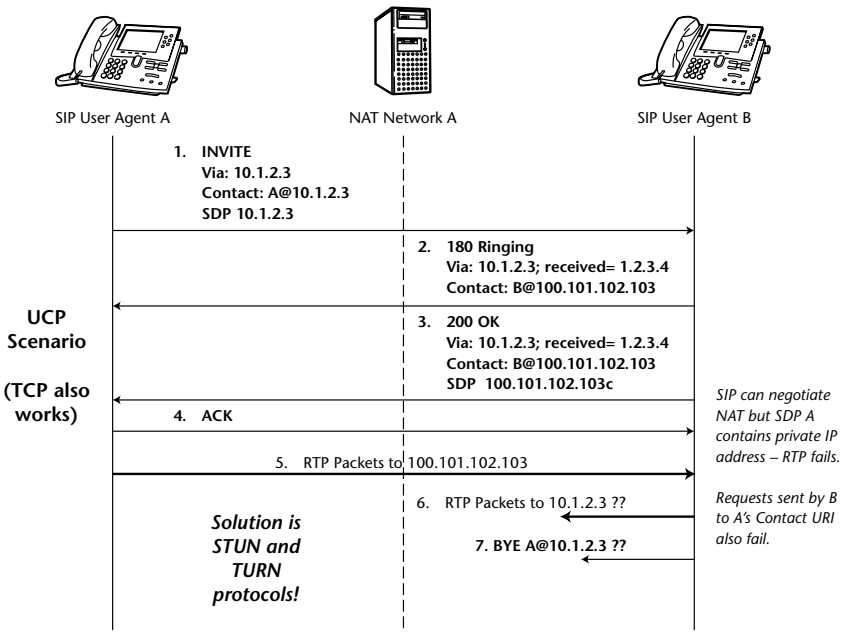


Figure 10.1 Unsuccessful session setup through NAT

Of these three problems identified, only this first one has a solution in SIP. A proxy or user agent (UA) receiving this request would compare the IP address in the `Via` header to the IP address from which the packet was received. If the two are different (as they would be if a NAT is present), the correct IP address is added to the `Via` header with a `received=` parameter listing the actual IP address. This IP address would be used to route the response successfully back to user A, provided the NAT maintains the same binding between the private IP address and public IP address. (This is not a problem if TCP is used as the transport. When a TCP connection is opened, the NAT creates the binding between the private IP address and port number and the assigned public IP address and port number. When the connection closes, the NAT removes the binding.) However, no easy solution exists for the other two problems.

The second problem could be solved by a persistent TCP connection for the duration of the session. This would mean that the `Contact` header would never be used to route future requests (such as a re-`INVITE` or `BYE`), since there would always be an open TCP connection.

A possible solution to the third problem has been proposed [5] that involves making RTP flows symmetric. For the case where only one endpoint is behind a NAT, RTP packet flow will be possible in at least one direction. This is so because the SDP of the endpoint outside the NAT will contain a correct globally routable IP address and port number. The use of symmetric RTP would

make the recipient of the successful RTP stream use the received IP address and port number to send RTP, ignoring the IP address in the SDP (which is not routable).

In addition to these SIP and RTP issues, there is the issue of the disclosure of the private IP address, information that administrators like to see blocked by the NAT. Although not significant from a signaling or media perspective, the `Call-ID` also leaks the private IP address of the UA. The complete solution to this problem will be discussed after the other major obstacle to SIP (firewalls) is discussed.

Firewalls

A *firewall* is a device typically present where a private IP network interconnects with the public Internet. A firewall acts like a one-way gate, allowing requests to go from the private network into the Internet, and allowing *only* responses to those requests to return, but blocking most requests originating in the Internet destined for the private network.

Certain types of requests from the public Internet are typically allowed. For example, HTTP requests to the corporate public web server will not be blocked by the firewall, nor SMTP e-mail transfers, nor are DNS queries for the public DNS server. These types of legitimate requests can be identified by the firewall by examination of the destination IP address in the IP header and the destination port number in the UDP or TCP headers.

For example, a valid web browsing request will contain the destination IP address of the public web server and port 80 (a well-known port number for HTTP). A particularly diligent firewall may even parse the packet to ensure that it contains a valid HTTP message.

The nature of the interaction between SIP and a firewall depends on the transport protocol. If the UA uses UDP to initiate the session, the server outside the firewall will be able to receive the SIP messages, but responses sent using UDP will be blocked by the firewall, since they are not associated with an outgoing request, because they are sent over a TCP connection. Any resulting media stream also will be one-sided only. This scenario is shown in Figure 10.2.

If TCP is used, it is possible for a SIP UA to establish a SIP session with a server on the outside of the firewall. This is because the SIP responses will be sent in the TCP connection opened by the user behind the firewall and will not be blocked. However, RTP media packets sent by the called party will be blocked by the firewall. The resulting media session will be only one-way. This is shown in Figure 10.3.

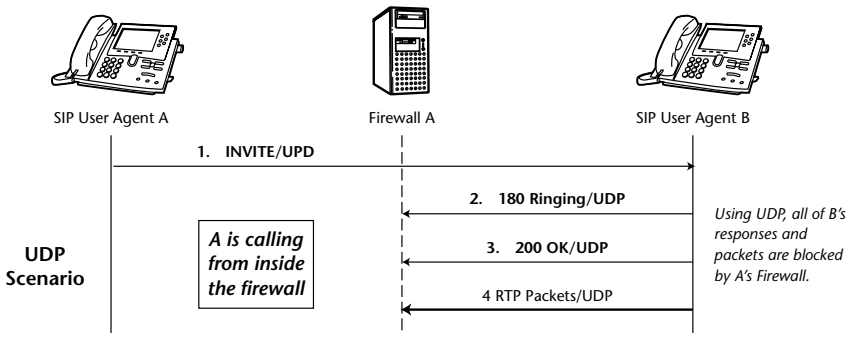


Figure 10.2 Unsuccessful call through firewall using UDP

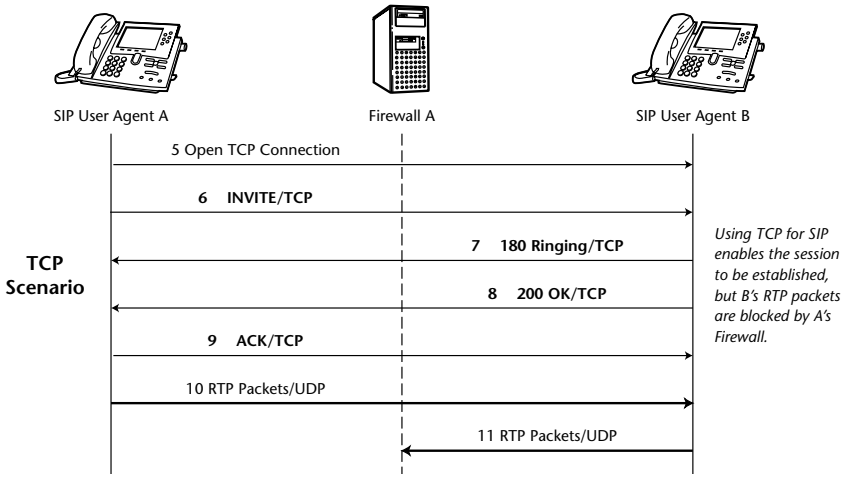


Figure 10.3 Unsuccessful call through firewall using TCP

If the a UA outside the firewall attempts to establish a session with the UA inside the firewall, all SIP and RTP packets will be blocked, regardless of transport, resulting in no session.

Note that it is possible to configure a firewall to allow SIP. However, doing so opens so many holes and weakens the protection provided by a firewall to such a degree that few network administrators would allow it. This is in contrast to NATs, which currently cannot be reconfigured to pass SIP and media.

Solutions to the firewall and NAT traversal problem will now be discussed.

STUN, TURN, and ICE

The IETF has standardized three protocols to help assist in NAT traversal. They are Simple Traversal of UDP through NAT (STUN), Traversal Using Relay NAT (TURN), and Interactive Connectivity Establishment (ICE).

STUN [6] is a simple protocol that allows a UA to discover if it is behind a NAT, and, if so, what type of NAT and what its public IP address is. STUN packets are sent by the UA to a STUN Server, which is located in the public Internet. The STUN responses tell the STUN client the public IP address and port that the STUN server received the STUN requests from. If the sent and received addresses and ports are the same, there is no NAT. If they are different, there is a NAT between them. In cases where a UA behind a NAT is trying to talk to a gateway or UA that has a public IP address, STUN allows a UA to "fix" all the parts of a SIP and SDP message with the correct public IP address. In this way, the UA manages its own NAT traversal. However, this does not work if both ends of the SIP and media session are behind NATs. For this, TURN may be required.

TURN [7] is a protocol that allows a client to obtain transport addresses from a TURN server on the public Internet. Since the TURN server is located in the public Internet, TURN addresses will always be routable. However, TURN addresses used for signaling or media are not optimal IP routes—the packets will traverse a triangular path. However, for some symmetric NAT and strict firewall traversal situations, TURN is the only way for a session to be established.

ICE [8] is a methodology for using STUN and TURN in a P2P manner that guarantees that the most efficient routing through NATs will occur. Using ICE, during the offer/answer session establishment, each UA signals all possible address candidates that it knows. For example, a UA may have three possible addresses:

- Private IP address, local to the LAN
- Public IP address discovered through STUN
- Media-relay address obtained using TURN

If a UA is multi-homed, has multiple Internet connections, or has a dual-stack IPv4/v6, these additional address candidates would be listed as well. The list is ordered by preference—direct addresses would be listed first, while relay addresses listed last.

After this exchange, the two UAs begin sending STUN packets to the candidate addresses received from the other UA. The STUN packets are sent using the same IP address and port numbers as the intended media stream. As a

result, these STUN packets can create bindings in NATs and open pinholes in firewalls, allowing packets to flow in the reverse direction. After a short period of testing, the most favorable addresses that have completed the P2P STUN exchange are selected and the RTP session begins. The ICE exchange can be reinitiated later in the session if the media flows change, or the network topology changes.

The techniques used in ICE are similar to techniques commonly used in P2P file-sharing networks today to traverse firewalls and NATs, and have been shown to be extremely effective. For a thorough discussion of the issues along with call flows, refer to [9]. An important standard under development for NAT traversal is the so called "SIP Outbound" Internet Draft [12].

Application Layer Gateways

STUN, TURN, and ICE all require support in the UA to traverse NATs and firewalls. An alternative approach that does not require special protocol support in the UA is known as an Application Layer Gateway (ALG). For firewall traversal, an ALG is a SIP and RTP proxy that is trusted by the firewall. That is, all SIP and RTP packets are directed at the ALG, which then performs authentication, validation, and so on, and enforces whatever policy the security administrator desires. ALGs are also sometimes known by their marketing name of *Session Border Controllers*. The firewall only allows SIP and RTP packets to pass, which originate or terminate on the ALG; all others are blocked. In this way, communication is possible through the firewall. This ALG works with NAT operation as well, because the IP addresses (which contain internal addresses) are modified when the SIP message is proxied. A detailed call flow is shown in the SIP Call Flow Examples [10]. The ALG may be connected to the firewall in a secure subnet sometimes called the *Demilitarized Zone* (*DMZ*).

A call flow involving a SIP ALG is shown in Figure 10.4. This example shows the ALG modifying the SDP so that the resulting RTP session is established in two legs between user agent A and the ALG, and user agent B and the ALG.

In this example, SIP Messages 2, 4, 6, and 9 (used to establish the session) are passed by the firewall, since these packets were sent to or from the IP address of the SIP ALG at port number 5060. The resulting RTP media packets also are passed by the firewall, since they originate or terminate at the IP address of the SIP ALG. In this way, the firewall needs only to open holes to allow SIP and RTP packets to the ALG. No dynamic changes in firewall policy are needed.

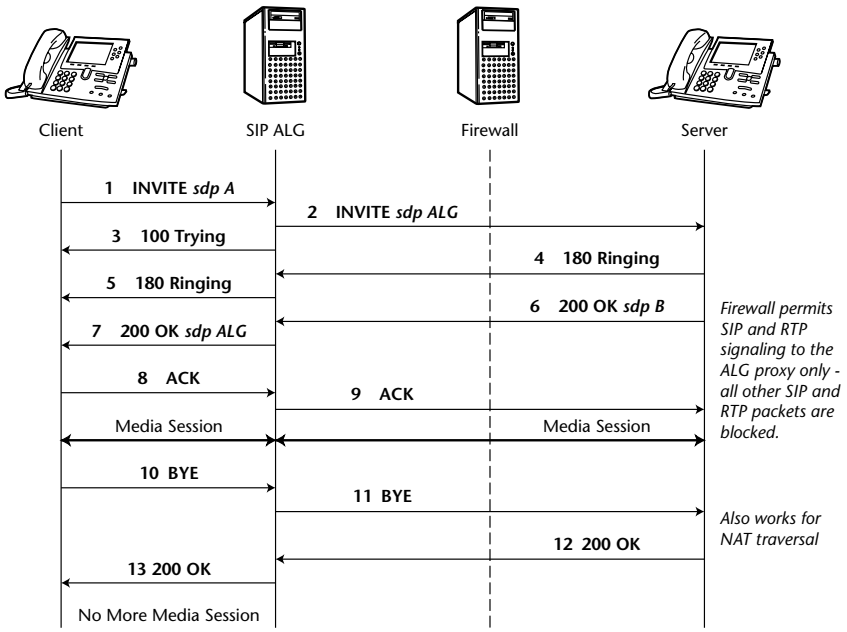


Figure 10.4 SIP ALG for firewall traversal

The other alternative to an ALG, which proxies both the signaling and media, is to use a SIP firewall proxy that communicates with the firewall or NAT. The firewall proxy performs any authentication, authorization, and so on, and then parses the SIP messages for the source and destination IP addresses and port numbers of the RTP packets. For example, the source and destination IP addresses and port numbers can be obtained from the SDP in the `INVITE` and `200 OK` messages. The firewall proxy then tells the firewall to open pinholes to let only those RTP packets pass. The firewall proxy also maintains the NAT address binding, and modifies the SDP accordingly so that the RTP packets can be sent directly between the UAs. Upon session termination with a `BYE`, the firewall proxy tells the firewall to close the pinholes and the NAT to remove the address binding. There is currently no standard protocol for communication between the SIP proxy and the firewall/NAT.

For these types of firewall traversal to work, the `Contact` header of the UA behind the firewall either must be set by the UA to resolve to the IP address of the ALG or firewall proxy, or the ALG or firewall proxy must `Record-Route`. A proxy inserts a `Record-Route` header containing an entry that resolves

back to the IP address of the proxy. This `Record-Route` header is forwarded with the request, stored by the UA server, and included in the response sent back to the UA client that originated the request. All future requests during the session must now include a `Route` header that forces the request to route through the proxy.

An example with one proxy that `Record-Route`s and another that does not is shown in Figure 10.5.

In this example, proxy A needs to be included in all future SIP messaging between the UAs, while proxy B does not. As a result, proxy A inserts a `Record-Route` header, while proxy B does not add itself to the `Record-Route` header. Therefore, the `ACK` and the `BYE` requests are routed through proxy A but bypass proxy B. Note that the `Route` header always contains information about the next hop, not the current hop. As a result, in this example, since the `ACK` of Message 13 is sent to proxy A, the `Route` header does not contain the URL of proxy A, but instead contains the URL of the next hop, which is UA2. The same is true for the `BYE` of Message 15. After the last URL in the `Route` header is used, the header is removed from the request, as it is in Messages 14 and 16. Also note that the `Route` header is never present in responses, because they are always routed back through the same set of proxies taken by the request.

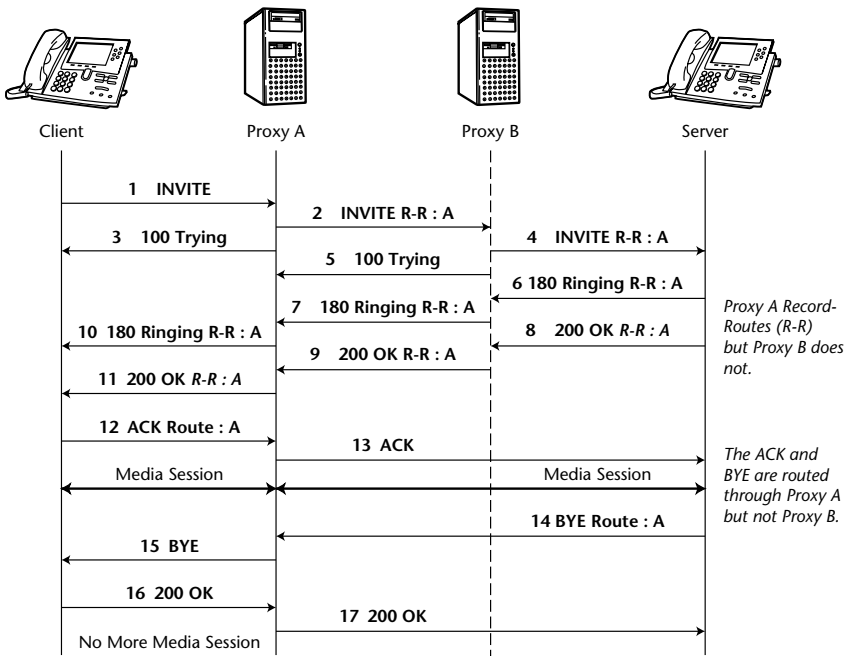


Figure 10.5 SIP proxy `Record-Route` example

For this scenario to work, the UA behind the firewall must have the ALG or firewall proxy set as the default outbound proxy for all outgoing requests.

The disadvantage of using ALGs is that they break the end-to-end nature of SIP. As a result, many of the security mechanisms described in Chapter 9, "SIP Security," are broken by ALGs. For example, An ALG which is not on the SIP signaling path will be bypassed by a UA using TLS transport. If S/MIME is used to secure a message body, an ALG will not be able to parse and modify the body. If the SIP `Identity` header field is used, an ALG may modify fields, causing the signature to become invalid. For these reasons, the use of ICE, STUN, and TURN is preferred over ALGs.

The ALG may also introduce longer media paths, similar to TURN. The ALG acts as a "media relay" and introduces delay for speech in both directions, thus reducing the quality of the conversation.

Privacy Considerations

Some aspects of privacy have been previously discussed in this chapter. However, these privacy aspects relate only to eavesdropping of a third party. Another issue is caller privacy. In the PSTN today, it is possible to block one's calling party number from being displayed to the called party. It is also possible to place a phone call anonymously by using a pay phone in which only the location (but not the identity) of the caller can be determined. In establishing a SIP session, the two parties must exchange significant information that might be considered private, including IP addresses, which can be traced to a particular subnet location or have a reverse DNS lookup performed to resolve the address back to a domain name.

In a session established directly between two UAs, there is no alternative to this information exchange. However, SIP network elements have been designed using a back-to-back UA (B2BUA) to implement an "anonymizer" service in which a caller's IP address, URL, or other identifying information can be blocked from the called party. In this application, there are actually two completely separate sessions established, with the B2BUA proxying signaling and media information from one call to the other. As a result, each party sends SIP and RTP packets to the B2BUA and not to each other. Once the call is completed, the anonymizer service can erase any logs, flush all states, and the resulting call is essentially untraceable.

The `P-Asserted-Identity` header field [11] can be used to assert identity within a trust.

Summary

NAT and firewalls break the SIP signaling and also interfere with the RTP media packet flow between SIP endpoints. The solutions developed by the IETF are the STUN, TURN, and ICE protocols that work (only if the SIP end devices can support them).

For older, existing SIP endpoints that do not support STUN, TURN, and ICE, Application Layer Gateways (ALG) or back-to-back UAs (B2BUA) can solve the problem of NAT and firewall traversal. However, ALGs break the end-to-end nature of SIP and, as a consequence, break the security mechanisms for SIP.

B2BUA can also be deployed as anonymizers to ensure caller privacy.

References

[1] "The IP Network Address Translator" by K. Egevang and P. Francis. IETF RFC 1631, May 1994.

[2] "Architectural Implications of NAT" by T. Hain. IETF RFC 2993, 2000.

[3] "Internet Transparency" by B. Carpenter. IETF RFC 2277, February 2000.

[4] "NAT Friendly Application Design Guidelines" by D. Senie. IETF Internet draft, March 2001, work in progress.

[5] "Common Local Transmit and Receive Ports (Symmetric RTP)" by D. Wing. IETF Internet Draft, work in progress, June 2005.

[6] "STUN—Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)" by J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy. RFC 3489, March 2003.

[7] "Traversal Using Relay NAT(TURN)" by J. Rosenberg, R. Mahy, and C. Huitema. IETF Internet Draft, work in progress, February 2005.

[8] "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Offer/Answer Protocols" by J. Rosenberg. IETF Internet Draft, work in progress, October 2005.

[9] "Best Current Practices for NAT Traversal for SIP" C. Boulton, J. Rosenberg, and G. Camarillo. IETF Internet Draft, work in progress, October 2005.

[10] "SIP Basic Call Flow Examples" by A. Johnston, et al. IETF RFC 3665, December 2003.

[11] "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks" by C. Jennings, J. Peterson, and M. Watson. IETF RFC 3325, November 2002.

[12] "Managing Client Initiated Connections in SIP" by C. Jennings et al. Internet Draft, IETF, March 2003.

CHAPTER 11

SIP Telephony

In this chapter, the basic telephony services and features will be discussed as implemented in a SIP-enabled network. First, basic telephony will be covered, followed by more advanced features. We will describe the basic and more advanced telephony features for which there are Internet drafts published to support inter-service-provider and intervendor product interoperability.

Basic Telephony Services

Basic telephony involves the establishment of sessions between endpoints. The basics of telephony in an all-IP environment are covered in Chapter 6, "SIP Overview." This chapter will focus on SIP and PSTN internetworking for basic telephony services.

SIP and PSTN Interworking

SIP and PSTN interworking occurs whenever a call originates in one network and terminates in another network. To accomplish this, the signaling and media transport protocols must be mapped between the two domains.

Gateways are the network elements bridging the two networks, as shown in Figure 11.1. The gateway is, as a result, part of both the PSTN and SIP network.

There are two basic approaches to building these gateways—complete protocol internetworking and protocol encapsulation. The latter approach is known as SIP Telephony (SIP-T) [1], which is not a separate protocol, but rather the SIP protocol plus a number of extensions. The gateway appears to the SIP network to be a user agent for many different users, and to the PSTN as a terminating telephone switch, known in North America as either Class 5 or Class 3, depending on the design.

Gateway Location and Routing

Since the gateway has multiple users, it does not `REGISTER` like a normal user agent. A normal registration binds a user's URI with a number of URIs. A gateway instead serves a host of users, either a corporate entity served from a PBX or Centrex group, a local Internet service provider (ISP) domain, or users associated with a particular geographic region, usually identified by a PSTN number range: country code, Numbering Plan Area (NPA), or area code or NPA-NXX (area code and local exchange). Instead of modifying SIP registration, the problem of gateway location and routing has been tackled in the IETF IP Telephony Working Group (IPTEL WG) with the development of the Telephony Routing over IP (TRIP) [2] protocol. This gateway to the location server protocol, based on Border Gateway Protocol (BGP)—used to advertise IP routes between networks—allows a gateway to advertise what PSTN number range it supports. This information is then available to proxies in routing SIP URIs containing telephone numbers and telephony URIs.

TRIP is designed for interdomain gateway location—it is not specifically designed to be used within a domain. However, the need for this same service within a domain has been identified by the IPTEL WG, which has begun work on a new version called Telephony Gateway Registration Protocol (TGREP) [3].

SIP URIS AND TELEPHONE NUMBERS

As discussed in Chapter 6, "SIP Overview," SIP URIs can contain telephone numbers such as `sip:+65234213@carrier.com;user=phone`. This URI does not need TRIP or any other protocol for routing, since the SIP request should be routed to the `carrier.com` domain, which then would locate a gateway. However, when a user dials a telephone number on a SIP device, the resulting *dial string* will first need to be interpreted based on a *dial plan* so that it can be put in global or E.164 format. This could either be represented as a tel URI or a SIP URI that has the domain of the user agent. In this case, routing information is needed, either through gateway routing tables, or automatically determined using TRIP or TGREP to locate and select an appropriate gateway. In addition, DNS ENUM queries, as described in Chapter 4, "DNS and ENUM," may be performed on this telephone number to resolve it to a URI instead of just forwarding it to the PSTN.

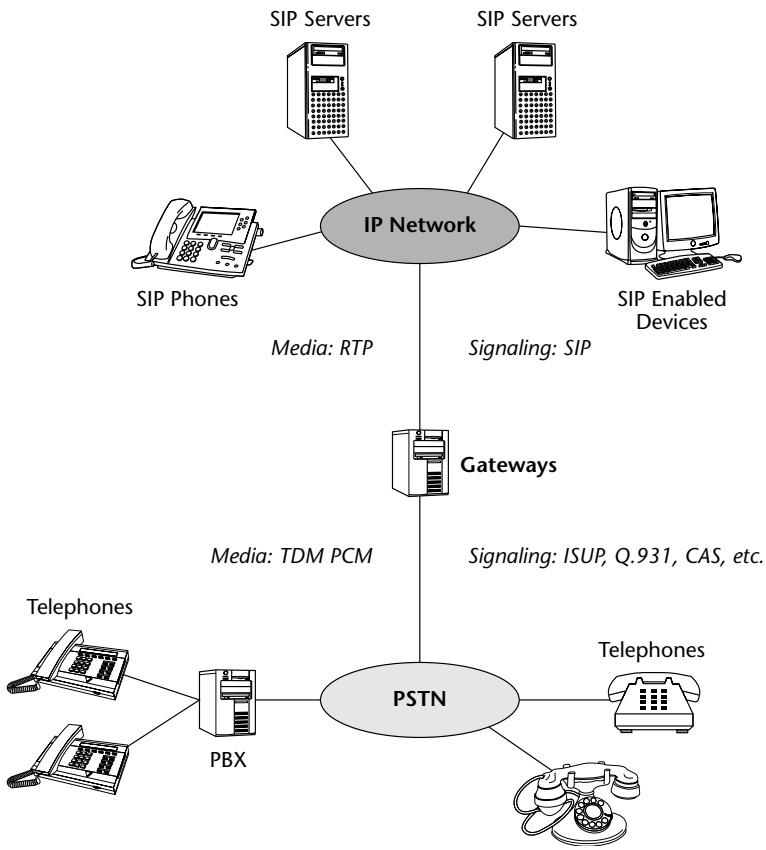


Figure 11.1 Gateways, SIP, and PSTN networks

To date, neither TRIP nor TGREP has seen any significant deployment or usage by service providers.

SIP/PSTN Protocol Interworking

SIP and PSTN protocol interworking has two levels: the media and the signaling levels.

The media interworking is quite straightforward. The PSTN generally uses a 64-kb/s pulse-coded modulation (PCM) encoded Time Division Multiplex (TDM) channel known as a *trunk* to carry the voice media. If Integrated Services Digital Network (ISDN) is used, the B channel contains the 64 kb/s PCM media stream. SIP-enabled devices generally have audio capabilities in the form of Real-Time Transport Protocol (RTP) packets. The media interworking

in a gateway involves terminating a PCM trunk on the PSTN side and bridging the media with an IP port that sends and receives RTP packets. Codec conversion between PCM and another codec is possible in the gateway, or the gateway may simply reuse the 64 kb/s PCM as RTP/Audio-Video Profile (AVP) 0, a common codec supported by nearly every SIP device capable of sending and receiving audio. The gateway refuses SIP sessions that do not contain an audio channel, and will decline all media types that it does not know how to map into PSTN telephony voice channels.

The signaling interworking is much more complex. While a SIP network is "flat" in terms of not having a different ITU-style user-to-network interface (UNI) and network-to-network interface (NNI), the PSTN uses many different signaling protocols to complete a call. For example, a PSTN call may enter the PSTN as a PBX trunk, in which a Circuit-Associated Signaling (CAS) protocol is used to out-pulse dialed digits as multifrequency (MF) tones. The telephone switch then signals to other telephone switches using ISDN User Part (ISUP) signaling, which is carried out-of-band in a dedicated packet switched network known as Signaling System 7 (SS7). Alternatively, ISDN (Q.931) D-channel signaling may be used.

Types of Gateways

The PSTN signaling protocol that a SIP/PSTN gateway will use will depend on the way it interfaces with the PSTN. We will consider two types of gateways:

- A *network gateway* is a high-port-count gateway that is typically owned by a PSTN carrier and interfaces with other PSTN switches using ISUP and ISDN as its NNI. A network gateway is typically located at a PSTN central office, where other large telephone switches are located. A network gateway is considered part of the PSTN trust domain.
- An *enterprise gateway*, on the other hand, is typically a small-port-count gateway that may be owned by a PSTN customer and interfaces with the PSTN via UNI protocols such as CAS and ISDN. This device typically will be located on a customer's premises or building. An enterprise gateway is not part of the PSTN trust domain.

SIP and Early Media

A number of Internet drafts have been written to document the basic mapping between SIP and PSTN protocols. However, the base SIP specification was found to be missing one key component of successful SIP/PSTN interworking—support of early media. In the PSTN, call progress indicators are

often provided in band in the media path (such as ring tone, busy signal, reorder tone, and so on). These indicators are carried in a one-way speech path that is established as soon as the called party is alerted but prior to the call being answered. The caller hears the ring tone or busy signal and knows how the call is progressing.

In SIP, the media path is not established until the called party answers (`200 OK`), and all call progress indicators are assumed to be carried in the SIP responses, not in any media path (`180 Ringing`, `181 Call is Being Forwarded`, `486 Busy Here`, `503 Service Unavailable`, etc.). This is not a problem in a call from the PSTN to SIP—the gateway simply takes the SIP response code and generates any tones or signals in the PSTN media path. However, for SIP-to-PSTN calling, the SIP phone's local ring-back tone generated by the receipt of a `180 Ringing` response from the gateway masks the in-band progress indicators being received by the gateway. The result is that the call may fail, and the SIP caller will never hear any indication, just the locally generated ring-back tone.

The solution is for a PSTN-to-SIP gateway to use early media and a `183 Session Progress` response, which is used to indicate that the call is progressing, but that the user agent server (PSTN gateway) is not able to determine from signaling what is occurring, but that information may be available in the media path. The gateway then sends the call progress tones or signals it is receiving in the one-way speech path in the TDM channel as RTP packets to the SIP phone. The SIP phone receiving a `183` response knows then to play those RTP packets instead of generating local alerting, as shown in Figure 11.2. In general, a SIP UA that receives early media (RTP that arrives before the `200 OK` answer) should stop playing any locally generated tones and play the early media instead.

This approach works but has an unfortunate side effect in the case of a SIP call that may have been forked to two different locations in the PSTN. The result of this is that two `183` responses will be received, and the SIP user agent client will have to decide which media stream to play, or whether to mix the two together.

Note that the gateway only sends a `183 Session Progress` response if it is unable to determine whether ringing is occurring. For example, if the PSTN connection is exclusively ISDN, then the alerting message can be mapped to a `180 Ringing`. However, in many cases (especially where some type of non-SS7 signaling path is present in the PSTN), the gateway will not be able to make this determination and will send a `183 Session Progress`.

A SIP-to-PSTN call flow is shown in Figure 11.3. In this case, the called SIP user agent sends a `180 Ringing` response to the gateway, which then seizes a trunk in the PSTN and sends an address complete message (ACM) to the PSTN. If the PSTN requires in-band alerting such as a ring tone, the gateway

would generate it. The `183 Session Progress` is not used. In this scenario, if the SIP user agent returned an error response (such as `410 The Number is No Longer in Service`), the gateway would be responsible for playing a suitable announcement for the PSTN caller. In the long term, this will involve a text-to-speech conversion in which the gateway would speak the error reason phrase. In the short term, the gateway will need to play a prerecorded announcement, or forward an `INVITE` to an announcement server, which can play the announcement.

Many examples of SIP and PSTN interworking are given in the IETF SIP PSTN Call Flows Best Current Practice (BCP) RFC 3666 [4]. Detailed information about mapping between SIP and ISUP can be found in RFC 3398 [5].

SIP Telephony and ISUP Tunneling

SIP internetworking with the PSTN at the signaling level involves a mapping of message types and parameters from one network to another. For example, consider the PSTN-to-SIP call in Figure 11.3. The ISUP Initial Address Message (IAM), or Message 1 in Figure 11.3, is shown in Table 11.1, along with a description of each field.

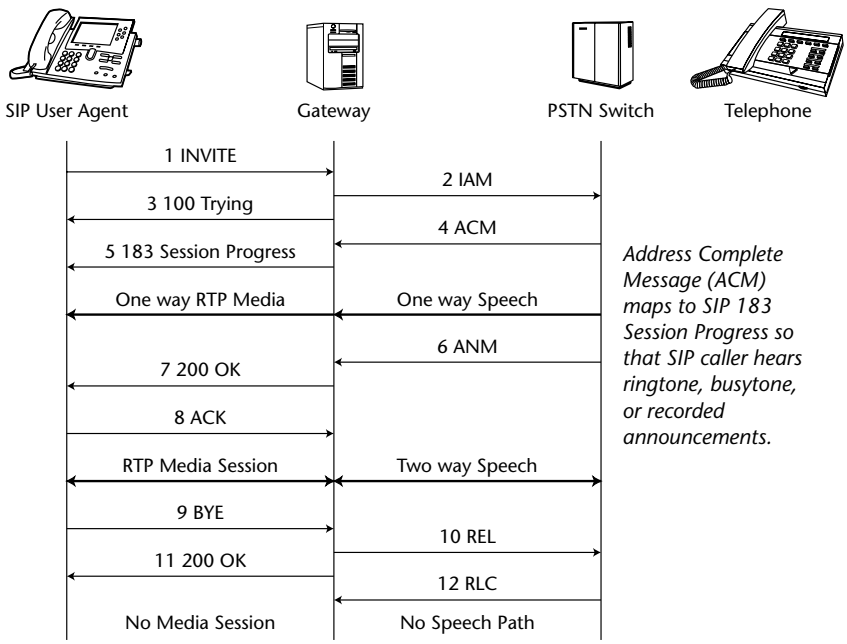


Figure 11.2 SIP-to-PSTN call with in-band call progress indicators

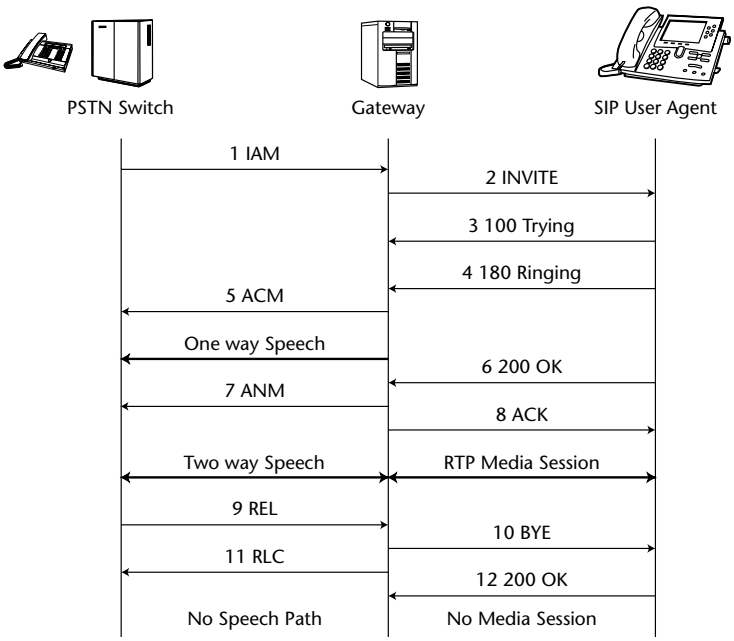


Figure 11.3 PSTN-to-SIP call

Table 11.1 ISUP IAM Message and Field Description

INITIAL ADDRESS MESSAGE (IAM)	DESCRIPTION
`CgPN=314-555-1111,NPI=E.164, NOA=National`	Calling Party Number
`CdPN=972-555-2222,NPI=E.164, NOA=National`	Called Party Number, Numbering Plan Indicator, Nature of Address
`USI=Speech`	User Service Information
`FCI=Normal`	Forward Call Indicator
`CPC=Normal`	Calling Party's Category
`CCI=Not Required`	Call Charge Indicator

The IAM can be mapped to the SIP INVITE Message 2 of Figure 11.3 as shown here:

```
INVITE sip:+19725552222@proxy.carrier.com;user=phone SIP/2.0
Via: SIP/2.0/UDP gw1.carrier.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: <sip:+13145551111@gw1.carrier.wcom.com;user=phone>tag=3342k
To: <sip:+19725552222@proxy.carrier.com;user=phone>
Call-ID: 123456028796867655
CSeq: 10 INVITE
Contact: <sip:gw1.carrier.com>
Content-Type: application/sdp
Content-Length: 156

v=0
o=GATEWAY1 2890844527 2890844527 IN IP4 gatewayone.carrier.com
s=-
c=IN IP4 gatewayone.carrier.com
t=0 0
m=audio 3456 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Some field mapping is obvious, such as calling party number to From, but others are not so obvious. In particular, there are generally many more parameters in a PSTN signaling message than can be mapped to a SIP message. (For example, how is the forward call indicator mapped to SIP?) The result is some information loss. However, if the call routes over the SIP network to the destination, there is no net effect on the call completion, since all information usable in the SIP network has been mapped. The additional parameters that are not mapped from ISUP to SIP are designed for PSTN routing, not SIP routing, and their loss has no effect.

Similarly, mapping from SIP to ISUP (shown in Figure 11.2) does not cause a loss in functionality. In this case, some ISUP parameters that have no counterpart in SIP will need to be created for the mapped IAM. These values are set to default values, typically on a trunk group basis.

However, should a call be routed from the PSTN to SIP, and then back to the PSTN, some of the lost parameters from the first PSTN leg could be useful in routing in the second PSTN leg. To solve this problem for networks designed to do this, the encapsulation of PSTN signaling messages, in addition to internetworking, was developed. This application of SIP, known as SIP Telephony (SIP-T) [1], carries the PSTN signaling information in the SIP signaling message as a MIME message body [6]. The terminating gateway then constructs the second leg PSTN signaling based on the SIP signaling parameters and the attached message body of the original PSTN signaling. The resulting network offers the possibility of making the SIP leg of the call transparent to the PSTN. Put another way, SIP-T enables the ISUP transparency across a SIP network. This call flow is shown in Figure 11.4.

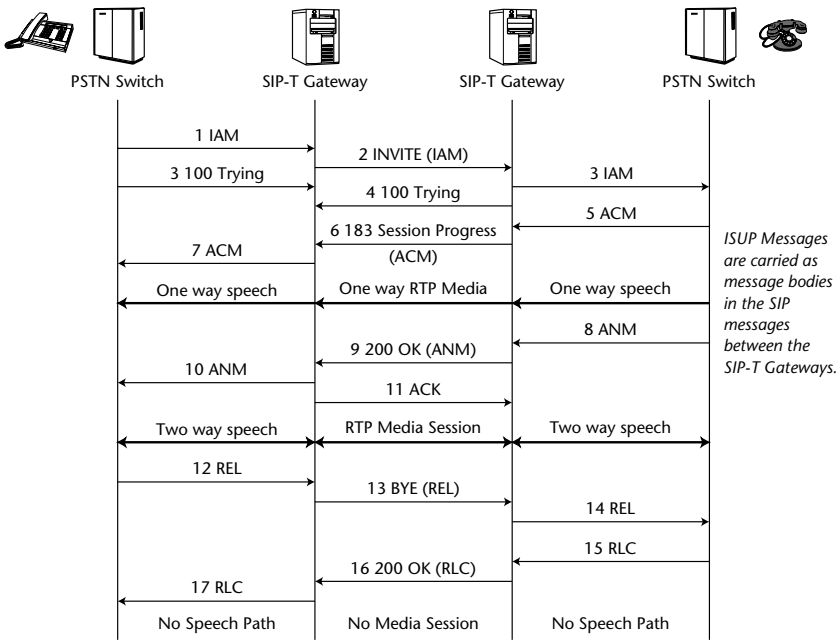


Figure 11.4 SIP-T call flow with ISUP tunneling

SIP-T also uses the `INFO` method to carry midcall signaling information, as shown in Figure 6.7.

The advantages of implementing SIP-T are obvious—it allows a carrier to build a PSTN network using a SIP IP telephony core, and provides transparency and full features. The disadvantages are not so obvious but can be seen on closer examination. For example, the complexity of gateways implementing SIP-T is much greater than a normal gateway, since a SIP-T gateway must still do all the PSTN-to-SIP mapping of a regular gateway, plus the additional encoding, decoding, and parsing, of the ISUP attachments. For example, the `INVITE` Message 2 of Figure 11.4 would be:

```
INVITE sip:+19725552222@proxy.carrier.com;user=phone SIP/2.0
Via: SIP/2.0/UDP ngw1.carrier.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: <sip:+13145551111@carrier.wcom.com;user=phone>;tag=gx3432
To: <sip:+19725552222@proxy.carrier.com;user=phone>
```

```
Call-ID: 12345602@ngw1.carrier.com
CSeq: 1 INVITE
Contact: <sip:ngw1.carrier.com;user=phone>
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="***"
Content-Length: 318

--***
Content-Type: application/sdp

v=0
o=GATEWAY1 2890844527 2890844527 IN IP4 gatewayone.carrier.com
s=-
c=IN IP4 gatewayone.carrier.com
t=0 0
m=audio 3456 RTP/AVP 0
a=rtpmap:0 PCMU/8000

--***
Content-Type: mime/isup

7452a43564a4d566736fa343503837f168a383b84f706474404568783746463ff
```

Compared to the example `INVITE` of Message 2 of Figure 11.3 shown previously, the message body is now a multipart MIME attachment. The first part contains the SDP of the gateway, while the second part contains the binary encoded IAM of Message 1 in Figure 11.4. In creating the IAM of Message 3 in Figure 11.4, the terminating SIP-T gateway uses information from both the SIP headers of the `INVITE` and the ISUP attachment.

In a SIP-T network, the complexity and intricacies of the PSTN are not absorbed in the gateways, but distributed throughout the network. All gateways must be able to parse the ISUP attachments, or true ISUP transparency will not occur. For international networks, there are many different incompatible "flavors" of ISUP. International gateway PSTN switches are extremely complicated and expensive because of the requirements that they be able to use and convert all these different incompatible protocols. In an international SIP-T network, what "flavor" of ISUP will be used? If multiple versions are allowed in the SIP network, every gateway will need to be able to deal with all versions of ISUP. If only a single version is allowed, there will be information loss as the ISUP is converted, leading to a failure in true transparency.

Finally, in a mixed SIP and SIP-T network (one involving SIP-T gateways, conventional gateways, and SIP phones and end devices), the ISUP will need

to be encrypted using S/MIME because of the sensitivity of some information present there (such as the calling party number of a private call), leading to additional complexity in gateways. Alternatively, a network of screening proxies will be needed to selectively remove ISUP attachments. (These proxies will have to decode the multipart MIME attachment composed of the SDP and the ISUP, extract only the SDP, recalculate the octet count, and then forward the message.) This adds delay and complexity to the SIP network.

The simplest argument against SIP-T is the fact that SIP phones cannot easily talk to SIP-T gateways, and there are now really two types of SIP endpoints that cannot talk to each other. As a result, T-SIP gateways can talk only directly to other SIP-T gateways, and SIP phones can talk only to other SIP phones.

Some early SIP networks will implement SIP-T, especially in so-called "softswitch" networks. However, for truly scalable and cost-effective telephony, the complexity and protocols of the PSTN must not be carried into the IP domain—a true SIP/PSTN internetworking gateway is required and ISUP tunneling or encapsulation is not required.

A summary of SIP-to-PSTN protocol mapping is shown in Table 11.2. Note that this table is greatly simplified and the actual mapping is more complex, depending on the version of ISUP used (ETSI, ANSI, and so forth).

Table 11.2 SIP-to-ISUP and ISDN Message Mapping

SIP MESSAGE OR RESPONSE	ISUP MESSAGE	ISDN MESSAGE
INVITE	IAM or SAM	Setup
INFO	USR	User
BYE	REL	Release
CANCEL	REL	Release
ACK	—	—
REGISTER	—	—
18x	ACM or CPG	Alerting
200 (to INVITE)	ANM or CON	Connect
4xx, 5xx, 6xx	REL	Release
200 (to BYE)	RLC	Release Complete

Enhanced Telephony Services

One remarkable promise of SIP-enhanced telephony services is that they can be implemented across the open Internet environment, working effectively across service provider boundaries and between equipment and software from many vendors. It remains to be seen to what extent this promise of extending PBX-like rich call features across the Internet will be fulfilled.

Enhanced services in telephony come in three possible forms:

- PBX or Centrex features
- Custom Local Area Signaling Services (CLASS) features
- Advanced Intelligent Network (AIN) services

Features have very specific names and definitions in the PSTN and PBX world. However, to discuss their analog in SIP, we will use generic names, which may or may not exactly map to the PSTN or PBX features. Although the IETF does not standardize features or services, many of these services implemented using SIP are described in Johnston [7].

Standardizing the key PBX functions across the Internet may herald a significant disruption in the PBX market, where all products are vendor-proprietary and interoperability (as with the ITU QSIG standard) is difficult to achieve. In addition, PBX phones from different vendors are not interchangeable. IP PBXs based on SIP have the potential of basic standards-based interoperability and also the potential of interchangeable SIP phones for baseline PBX features. We will take a closer look at baseline PBX features.

PBX or CLASS features generally include the following:

- *Call transfer*—There are three types of call transfer services (blind, unattended, and attended) that can be implemented using the `REFER` method [8]. In a blind transfer, the transferor sends a `REFER` and then immediately sends a `BYE` and terminates the existing session without waiting for the outcome of the transfer. In it is an unattended transfer, the transferor may keep the transferee on hold pending the outcome of the `REFER` request. Once the transferor receives notification that the transfer has succeeded, a `BYE` is sent to tear down the existing session. Finally, the attended transfer involves a temporary conference call between the three parties, in which the transferor knows the exact progress of the transfer. Once the transfer is complete, the transferor can then drop out of the call. The types of call transfer are described in Table 11.3 and in [9].
- *Call waiting*—This is a service implemented on single-line telephones. Since there is no such thing as a "line" in a SIP network, this feature does not have an exact analog. However, a SIP phone that offers multiple

"line" behavior would return a `180 Ringing` response and initiate alerting even when there is an active session established. The called party can then either place the session on hold and answer the second call, or ignore the second caller.

- *Call hold*—This feature has many forms in the PSTN, from a button on a telephone set that simply cuts the speaker and microphone to advanced features in PBX or ISDN systems. In a SIP network, a call is placed on hold by sending a re-`INVITE`, changing the media stream from bidirectional (`sendrecv`) to unidirectional (`sendonly`). Note that older SIP implementations may implement hold with a re-`INVITE` with a connection IP address of 0.0.0.0 in the SDP. The call is taken off hold when either party sends a re-`INVITE` with bidirectional media or a nonzero connection IP address.
- *Call park and pickup*—In this feature, a call is placed on hold at one location and then retrieved (picked up) at another location. There are a number of ways to implement these features in SIP. Some of them use third-party call control and a re-`INVITE`, while others use a `REFER` and then a redirect.
- *Call forwarding*—There are three options with this feature: forward on busy, don't answer, and unconditional. Forwarding can be done in SIP either in a proxy or in a user agent, as shown in Chapter 7, "SIP Service Creation." A proxy can translate one URI for another, resulting in a forward that is transparent to the calling user agent. Alternatively, a user agent or a proxy can issue a redirection response (`302 Moved`). A proxy receiving a `486 Busy Here` response can invoke a call-forward-on-busy feature by generating an `ACK` and then forwarding the `INVITE` to another URI. A proxy can also start a ring timer upon receipt of a `180 Ringing` response, and then send a `CANCEL` and proxy the `INVITE` to another URI to implement a call-forward, don't-answer service.
- *Calling line identification*—The ability to display calling line identification is a useful feature in the PSTN to aid the caller during alerting in deciding whether to answer a call or to implement automated screening services. For example, a feature could be implemented to block incoming SIP calls in which the calling party has not been identified. The basic functionality is built into SIP to accomplish this, using the `From` header. However, since the `From` header is populated by the calling user agent and not by a trusted source such as a carrier, this calling line identification is not verified or guaranteed to be accurate. The use of the `Identity` header field to provide a way to check the validity of the `From` header field is discussed in Chapter 9, "SIP Security."

- *Incoming and outgoing call screening*—Incoming and outgoing call screening can be implemented in either a proxy or a user agent. A `Request-URI` or `From` header is compared to a list of allowed or blocked URIs, and an appropriate response generated, such as `403 Forbidden`, in the event of a call being blocked. The outgoing call screening feature can only be implemented in a proxy if the user agent is configured to always use that proxy as an outgoing proxy. The incoming call-screening feature can only be implemented in a proxy if the user agent is configured to accept only requests from an incoming proxy, redirecting all other requests with a `305 Use Proxy` response.
- *Automatic callback and recall*—Automatic recall allows a PSTN user to return a missed call based on calling line identification. This is easily implemented in a user agent by caching the `From` header from the previous failed `INVITE` request. Automatic callback allows a PSTN user whose call fails because of a busy signal to have the call automatically placed, as soon as the called party becomes free. This can be implemented in a SIP network using a simple presence service, in which a `SUBSCRIBE` is sent to request notification when the called user agent is no longer busy. The `NOTIFY` response would then automatically generate a new `INVITE` to complete the call. This was shown as an example in Figure 6.11 in Chapter 6, "SIP Overview."
- *Speed dial*—Speed dial allows a user to place a call by dialing a shorter digit string, often stored in the network or in the telephone set. A SIP user agent can use any speed dial method. Alternatively, a mapping from a "nickname" to a full URI is possible to allow easier "dialing" in a similar way that nicknames are useful in e-mail.
- *Conference calling*—Conference calling is described in Chapter 14, "SIP Conferencing."
- *Voicemail*—This important service is described in Chapter 12, "Voicemail and Unified Messaging."

Table 11.3 Types of Call Transfer

TRANSFEROR ACTION	TRANSFER	CONSULT NEW PARTY	TALK TO BOTH PARTIES
Unattended	Yes	No	No
Consultation Hold	Yes	Yes	No
Attended	Yes	Yes	Yes

Besides emulating PBX features, SIP also can emulate AIN services found in the PSTN. Both capability sets CS-1 and CS-2 defined by the ITU are discussed and illustrated for SIP in Lennox, et al. [10]. AIN or advanced features in the PSTN often take one of the following forms:

- *Interactive voice response (IVR) system*—These "voice menu" or "auto-prompt" systems allow an automated attendant to answer a call, play prompts, collect information using either spoken words or DTMF digits, and then route the call to its final destination. This typically is accomplished using a third-party control mechanism discussed later in this chapter.
- *Specialized routing services*—Call routing is performed based on time of day, origin number, traffic load, and other factors. This type of routing decision is routinely made in proxy servers in a SIP network.
- *Database query and information retrieval services*—These services are extremely primitive in the PSTN because of the separation of the PSTN from the databases in which the information is stored. For a SIP-enabled network, retrieval and return of information from the biggest database of all, the Internet, is trivial using HTTP, FTP, and so on. Simple query services can be built using a SIP redirect server.

Call Control Services and Third-Party Call Control

Call control services and third-party control are important topics in telephony, because they enable many advanced services and features. For example, automated dialers and IVR systems can be built using SIP third-party control.

Problem Statement

Circuit-switched telephony services in the PSTN historically have been augmented by advanced services in private voice networks by innovative PBX vendors. Where even PBX technology failed to provide adequate solutions, such as in PC-phone interaction and especially for call center applications, the computer telephony industry (CTI) has come to the rescue. Advanced services, however, were created at significant cost and had the drawbacks of: local reach only because of lack of global standards, tremendous complexity that translated into very high cost of ownership, and long time-to-market. Finally, new telephony applications and other communications have emerged with the advent of the Internet.

A list of call scenarios in use at present would include the following:

- Managing telephony applications from the desktop PC
- Click-to-connect
- Internet call waiting
- Instant communications

More complex call control models, however, are used at present in conventional telephony systems. Call control, especially in private voice networks, can be extremely complicated. Some examples are given here, with increasing degrees of complexity:

- Pick up a call that was ringing someone else's phone.
- Monitor a call in progress such as for call center operations.
- Join a conference call (whether scheduled or spontaneous).
- Transfer a call to another party.
- Receptionist and secretary model. A caller on the PSTN calls an employee accessible via a private voice network. The call will first reach the receptionist, who will inquire about the nature of the call and forward it to the desired party. The called party may have a secretary who may screen the call before connecting it to the boss.
- Call center applications. In a call center scenario, a customer call for support to an 800 number may be routed to an enterprise call center by a public carrier, depending on the call's origin and time of day. The call reaches the call center and, depending on the interaction of the caller with an IVR, the call may be routed according to load and skill set to an appropriate agent group. The agent taking the call may refer the caller to a subject expert in another location and may stay online to make sure that the call has been routed to the customer's satisfaction. This complex scenario is accomplished at present with quite expensive call software on various carrier and private network switches, using a rather high count of circuit switch ports both on the carrier side and within the private voice network that owns the call center.

The complexity in the preceding scenarios addresses real business requirements for customer, vendor, and partner relations and were addressed by circuit-switched telephony in combination with CTI. Though Internet engineers promote simplicity as an engineering design goal, it is felt SIP-based solutions have to deal with such complexity as well.

NOTE **Examination of the preceding scenarios for call control shows that protocols and standards from conventional telephony models—IN/AIN, PBXs, and softswitch devices—control protocols cannot meet the requirements for advanced voice services.**

SIP can be applied for extremely complex call scenarios for all of the call scenarios presented here. The question is, "Is there a consistent mechanism to deal with such scenarios, with no unnecessary complexity beyond doing the job right and based on public standards for a large degree of interoperability?"

The problem was first addressed in the SIP community in the paper on third-party call control in SIP [11] and then formulated within a framework for SIP call control extensions [10]. The desire was to achieve functionality with extensions and without burdening the base SIP protocol implementations that do not require functions that are more complex. The transfer services were treated in more detail in Sparks [9], after which some interesting new applications emerged, most notably controlling a SIP phone from a desktop PC [12],[13] using SIP third-party call control. A useful method for SIP, called `REFER`, turns out to be appropriate to handle most complex call control scenarios. Readers who have some insight into the complexity of the IN/AIN, PBXs, H.323, and MEGACO/H.248 protocols may find the simplicity of the `REFER` method outlined here especially intriguing.

The REFER Method

The `REFER` method allows a third party (such as a controller) to request the caller set up a call with a resource. The resource is identified in a new SIP header called `Refer-To`. Note that the resource is a URI, not necessarily a SIP URI. For example, the `Refer-To` header could contain an HTTP URI, which would result in the web page being retrieved instead of a new SIP call being initiated. In addition, a SIP URI in a `Refer-To` header may contain the method type, which defaults to `INVITE` if not specified. Thus, a `REFER` request could be used to request a `BYE` be sent instead of an `INVITE`. A `REFER` request must contain a `Referred-By` header that identifies the referring party. In the new request generated as a result of the `REFER` request, this `Referred-By` header can be used to inform the called party by whom it was referred to make this call. Following are examples of the `Refer-To` header:

- To request a party to call John Doe, a `REFER` request is sent containing:

```
Refer-To: sip:john.doe@isp.com
```

- The same request containing the header:

```
Refer-To: sip:john.doe@isp.com
          Accept-Contact=sip:jdoe@100.101.102.103;only=true
```

  ensures that the right John Doe device (instance) is reached by the request. For example, the referred request could encounter a forking proxy, or some other service logic in which there are multiple possible `Contact` URIs for John Doe. The `Accept-Contact` header along with the `only=true` parameter ensures that the right one is reached.

Following are examples of the `Referred-By` header:

- `Referred-By: sip:manager@isp.com;ref=http://headhunters.com` provides the reference source for "headhunters," which is a web page.
- `Referred-By: sip:john.doe@isp.com; ref=<htpp://headhunters.com>; scheme=pgp;pgp-version="5.0"; signature="34a6e328d7cc710f8382"` also provides a Pretty Good Privacy (PGP) signature computed across the URI of referee and the reference URI. It is recommended that `Referred-By` headers be signed to prevent unauthorized parties from hijacking calls.

Informational reply status codes have been proposed [13] to provide information about the progress of the call. Other approaches are also being considered, including the use of `SUBSCRIBE` and `NOTIFY` requests to request and receive notification about the final outcome of a `REFER` request.

SIP Third-Party Call Control

As mentioned, by contrast to the ITU and CTI standards, the SIP third-party control presumes intelligent IP end devices instead of dumb terminals, keeping pace with the advent of highly distributed computing. We will show here how intelligent IP endpoints can be controlled for communication services using SIP only. The central idea is loose coupling between intelligent IP endpoints and using SIP call setup only to invoke the necessary functions in the devices (such as phones and desktop and Palm computers). The details of the device operation are then left to the device itself.

Third-party call control is a basic telephony function and is used for many services (such as call setup by a controller and call transfers). SIP third-party call control also can be used to control many other services and integrate communications with various other applications and transactions, as shown in the application services architecture in Chapter 19, "SIP Component Services." No other standard or proprietary protocols are necessary besides SIP.

NOTE **Readers familiar with computer telephony integration (CTI) can appreciate the complexity of the technology involved in controlling phones from computers and desktop PCs. CTI relies on complex schemes that are based on special application programming interfaces (APIs) that, in turn, depend on proprietary system implementations and proprietary operating systems.**

Any "open" API has, therefore, two claimants on intellectual property rights and change by the owners at their convenience. We will show here that computers can control phones in a very simple and completely open, standard manner using SIP. It is the opinion of the authors that CTI has been made obsolete by SIP.

Basic Third-Party Call Control

The concept for basic third-party call control is shown in Figure 11.5, where a controller will set up a call between two parties, A and B, without participating in the conversation in any way.

The operation is as follows:

1. The controller sets up a call with the first party (party A) by sending an INVITE message (Message 1 in Figure 11.5). This INVITE has an SDP message body with no media lines. Party A responds with a 200 OK also containing SDP with no media lines. The controller then sends an ACK to complete the exchange. This sets up a SIP session or dialog but does not establish a media session.
2. The controller will now set up a call with the second party (party B) by sending a second INVITE (Message 4 in Figure 11.5) without any SDP present. This time, the controller stores the SDP data received from party B in the 200 OK message (Message 5). The controller holds off sending the ACK to party B until this media information is communicated to party A in the next step.
3. Next, the controller re-INVITEs party A using the SDP connection and media data supplied by party B in the previous 200 OK message (Message 5 in Figure 11.5). Party A responds with SDP media information in the 200 OK (Message 7). This SDP information is then passed back to party B in an ACK (Message 8). An ACK without SDP to A completes the session setup. Parties A and B can now exchange Real-Time Protocol (RTP) media, since they both have the required SDP connection and media data from each other.
4. From a signaling perspective, parties A and B are still communicating with the controller and not with each other. To terminate the call, any party can send a BYE message to terminate the connection to the controller (as shown in Message 11 in Figure 11.5). The controller will follow up by sending a BYE to the other party (Message 12). Both BYE messages are followed by 200 OK messages (Messages 13 and 14), and the call is terminated.

Security for Third-Party Call Control

Third-party call control is simple to implement in well-secured IP networks where no security risks are assumed within the trusted environment. In a larger context, however, additional steps must be taken to authenticate the

controller to the controlled parties and to make the controlled SIP endpoints exchange RTP media with each other. If the controller has the same identity as one of the parties (for example, if the controller is just another device that is associated with A in Figure 11.5), then no new authorization or identity issues are caused by this. However, if the controller is a different identity, then this scenario can appear to the parties involved as a man in the middle attack (MitM). That is, B is exchanging signaling with the controller, but media with A. Note that since the controller is not actually manipulating and modifying the SDP, but is just cutting and pasting it from one message to another, it is possible that the SDP bodies could be encrypted with S/MIME. A and B could use this to securely exchange SRTP master keys to have an encrypted and authenticated media session between them.

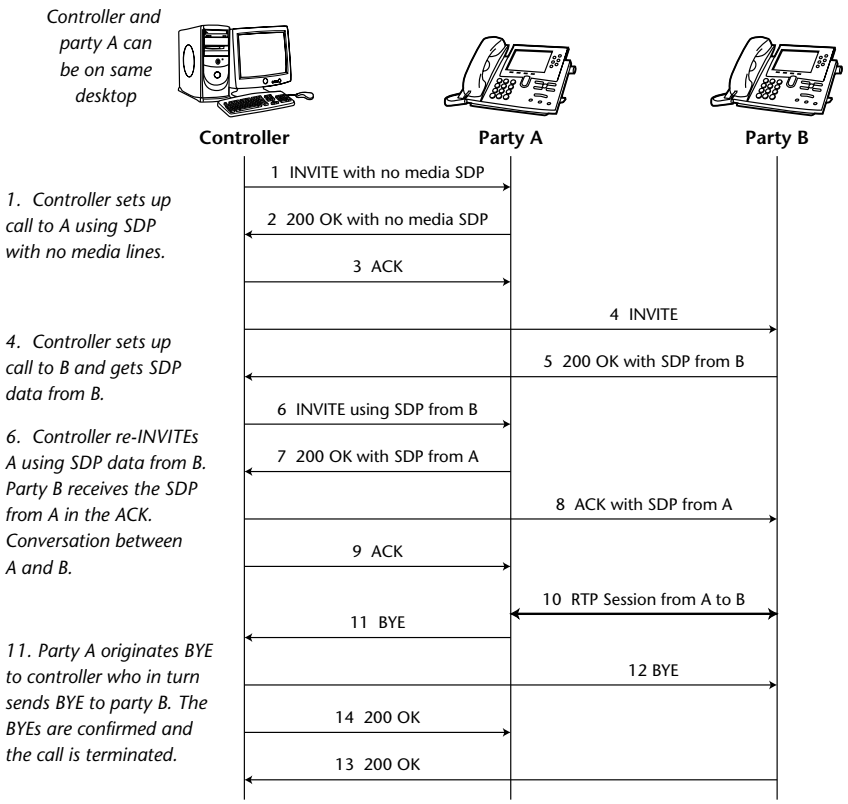


Figure 11.5 Basic third-party call control

Peer-to-Peer Third-Party Call Control

SIP can be used for complex call control applications in the peer-to-peer control model. We will discuss an example where a dialer application on a desktop computer of a secretary can control his or her own phone, and also can be used to set up calls between two other phones (such as between the boss and a customer, as shown in Figure 11.6). In this application, SIP for presence is used to display on the computer the state of the secretary's own phone. For simplicity, it is assumed that the presence publisher for the phone is located in the phone itself. This is quite doable with intelligent SIP phones and does not require more complex message exchanges with a dedicated presence server.

The framework for peer-to-peer call control with SIP is detailed in [12].

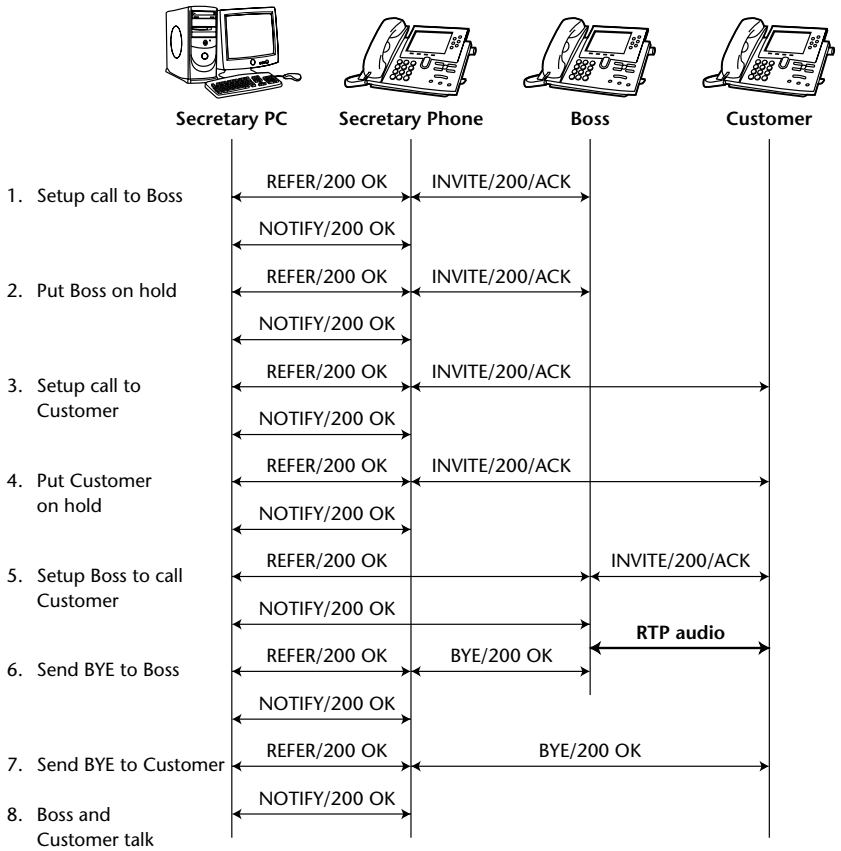


Figure 11.6 Example of peer-to-peer third-party call control

SIP servers that normally route all calls are not shown in Figure 11.6 so as to focus the example on only peer-to-peer third-party call control. In addition, complete message transaction sequences (such as `REFER/200 OK` and `INVITE/200/ACK`) are grouped and represented by single, two-headed arrows. This example is based loosely on [13]. The dialer will now use third-party call control to go through a number of steps to set up a call between the boss and a customer:

1. The dialer refers the phone to set up a call to the boss using a `REFER` request. The result of the `REFER` is sent in the `NOTIFY` response. This occurs after each `REFER` and enables the dialer to know the exact state of the call.
2. The dialer refers the phone to put the boss on hold. This is accomplished by sending a `REFER` to the phone to set the media stream to `sendonly` or `inactive`.
3. The dialer refers the phone to place a call to the customer.
4. The dialer refers the phone to place the customer on hold.
5. The dialer refers the boss's phone to call the customer. The boss and the customer now have an RTP voice "call" established. The dialer now proceeds to get out of the loop.
6. Dialer refers the phone to send a `BYE` to the boss.
7. Dialer refers the phone to send a `BYE` to the customer.
8. The boss and the customer continue to talk.

Real-life scenarios would look more complex, since we have omitted several messages in this example for the sake of clarity. There would be extra message exchanges for presence to display more extensive information about the status of the phone, and messages with SIP servers to route the calls. In addition, as discussed in Chapters 9 and 10, the existence of firewalls and NATs will complicate third-party call control.

Summary

This chapter has examined SIP telephony, internetworking with the PSTN, and feature implementations. Even though SIP uses a radically different call model and structure, basic and enhanced telephony services can easily be implemented in a SIP-enabled network.

References

[1] "Session Initiation Protocol for Telephones (SIP-T): Context and Architectures" by A. Vemuri and J. Peterson. IETF RFC 3372, September 2002.
[2] "A framework for Telephony Routing Over IP" by J. Rosenberg, et al. IETF RFC 2871, June 2000.
[3] "A Telephony Gateway REgistration Protocol (TGREP)" by M. Bangalore, R. Kumar, H. Salama, J. Rosenberg, and D. Shah. Internet Draft, work in progress, July 2005.
[4] "Session Initiation Protocol (SIP) Public Switched Telephone Network (PSTN) Call Flows" by A. Johnston, et al. IETF RFC 3666, BCP 76, December 2003.
[5] "SIP to ISUP Mapping" by G. Camarillo, A. Roach, J. Peterson, and L. Ong. IETF RFC 3398, December 2002.
[6] "MIME Media Types for ISUP and QSIG" by E. Zimmerer, et al. IETF RFC 3204, December 2001.
[7] "SIP Service Examples" by A. Johnston, et al. IETF Internet Draft, February 2006, work in progress.
[8] "The Session Initiation Protocol (SIP)" by R. Sparks. Refer Method, RFC 3515, April 2003.
[9] "SIP Call Control – Transfer" by R. Sparks, A. Johnston, and D. Petrie. IETF Internet Draft, work in progress, February 2006.
[10] "Implementing Intelligent Network Services with the Session Initiation Protocol" by J. Lennox, H. Schulzrinne, and T. La Porta. Columbia University Computer Science Technical Report CUCS-002-99, January 1999.
[11] "Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)" by J. Rosenberg, J. Peterson, H. Schulzrinne, and G. Camarillo. RFC 3725, April 2004.
[12] "A Call Control and Multi-Party Usage Framework for the Session Initiation Protocol (SIP)" by R. Mahy, et al. Work in progress, February 2005.
[13] "Remote Call Control in SIP using the REFER Method and the Session-Oriented Dialog Package" by R. Mahy and C. Jennings. IETF Internet Draft, October 2005, work in progress.

CHAPTER 12

Voicemail and Universal Messaging

We will show in this chapter how SIP-based voicemail, presence, and e-mail can support unified messaging on the same Internet infrastructure at much reduced complexity, while at the same time providing enhanced functionality. The integration of e-mail and IM is beyond our scope here, since this can be done in a text-based message gateway in a straightforward manner. Fax and video messages can be treated in a similar manner to voice as shown here.

After presenting an example for the overall messaging architecture, we will discuss the major steps in the life cycle of a voice message: Depositing a message, notification of message waiting, and message retrieval.

Problem Statement for Unified Messaging

On the positive side, rich messaging choices (such as the venerable voicemail and the various text-based messaging services), especially Internet-based e-mail, can accommodate the preferences of users for one or another mode of communications. Some users hate voicemail and prefer e-mail only, while other users prefer to have voicemail as well.

On the negative side, the proliferation of messaging services, such as voicemail on PBXs, for PSTN phones, for mobile phones, e-mail, fax, instant messaging, and paging, creates challenges for:

- *End users*—To manage and keep track of their messages on multiple devices, systems, and networks
- *Service providers and network administrators*—To manage multiple message systems

This challenge has prompted vendors and service providers to offer various solutions for unified messaging, but a closer look will reveal these systems to be proprietary. Service integration is accomplished by "brute force" with its resulting high complexity.

There are incompatible PBX voicemail systems, PSTN voicemail, mobile voicemail, fax, and pagers. There are different e-mail clients and Web clients, including multiple media (text, voice, fax, video) and individual user preferences.

A very short summary of key system properties would look like the following:

- Full user control of messaging features and personalization
- Full user control of recording and playback
- Options for receiving notifications: e-mail, IM, pager, Web browser
- Scalability for very large systems and multiple accounts
- Media-agnostic (text, voice, fax, video, whiteboard, and so on)
- Device-agnostic (PC, phones, fax, pager)
- Use existing infrastructure, including:
  - Data types and records
  - Network protocols between network elements
  - Network elements
    - IP telephony gateways
    - SIP servers: registrar, redirect, proxy, forking
    - Component servers: media servers, voice portal, or IVR
    - Directory
- Security infrastructure

The main new network element in the design of a unified message system is the universal message store, under control of a unified message server. The unified message server and store can be implemented in various ways to meet

these requirements, but it also needs to present a uniform approach for the three message store access phases:

1. User and system access for managing the "mailbox"
2. Message deposit
3. Message retrieval

Architecture and Operation

An example is appropriate at this point to illustrate the open architecture for unified messaging. This architecture is based entirely on the standard protocols SIP, SMTP, HTTP, and RTSP. An example illustrating the operation of the open architecture for unified messaging is shown in Figure 12.1.

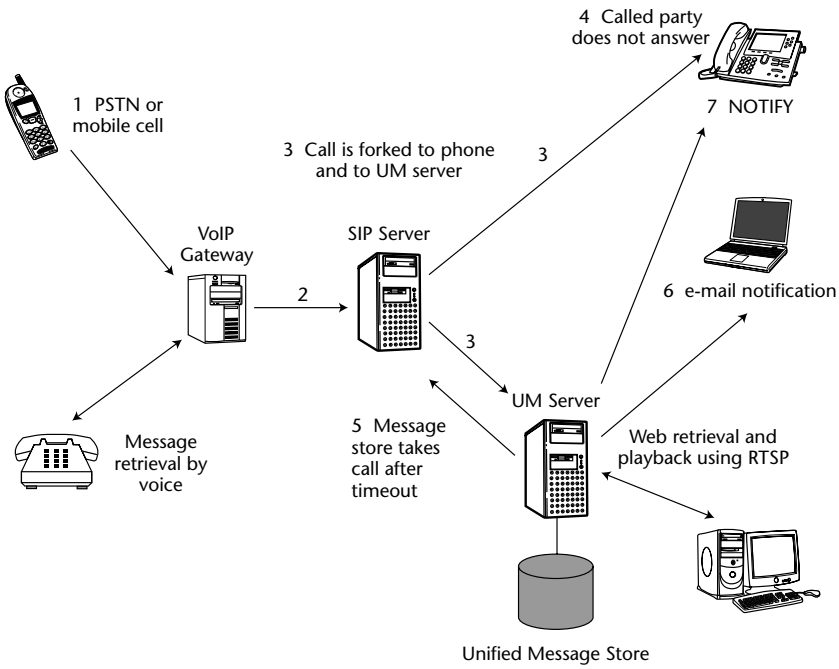


Figure 12.1 Example of open standards based unified messaging operation

The example in Figure 12.1 shows the following steps for a voicemail application [1]. The high-level message flow is as follows:

1. A caller on the PSTN or mobile network places a call to a SIP phone.
2. The VoIP gateway sends the `INVITE` message to its outgoing SIP server.
3. The SIP server forks the call to the called party and to the unified message (UM) server.
4. The called party does not answer the call.
5. The UM server takes the call after a timeout, and the caller leaves a message that is stored in the UM store.
6. The UM server sends an e-mail notification to the called party.
7. The UM server sends a `NOTIFY` message to SIP devices and while also using the other facilities for message waiting and the notification for message waiting will be delivered:
   - To the SIP phone as a `NOTIFY` message
   - To the IM client (not shown in Figure 12.1) as a `NOTIFY` message
   - Via e-mail to the mail client
   - Can be displayed on the browser by accessing the UM store via HTTP
   - Can be announced via PSTN phone when the called party calls the mailbox

The voice message can be retrieved by PSTN, PBX, or SIP phones and also by Web browsers that have an RTSP-enabled media player (native or plug-in).

For simplicity we have not shown other network elements that may be involved in this voicemail application, such as various VoIP gateways; SIP servers performing registration, rendezvous, and routing; and component servers, such as VoiceXML voice portals and IVRs.

These network elements are part of the SIP infrastructure and can be reused without modification.

RTSP-Enabled Voice Message Retrieval

The Real-Time Streaming Protocol (RTSP), defined by RFC 2326 [2], allows the remote recording and replay of various media across the Net, and provides similar functionality to the familiar players for audio/video. The most popular Internet media players (such as RealPlayer or QuickTime) support RTSP.

Users will quickly appreciate the advantage of listening to voicemail by using an RTSP-enabled media player instead of a phone. Lengthy voicemail messages can be replayed selectively, so as to listen to certain parts with relevant information, instead of having to replay all messages in sequence with

their entire content. Replay control is exercised by moving the cursor to the desired message part marked by timestamps.

PC retrieval of voicemail can be a valuable component of a package of desktop and laptop PC applications for IP communications.

Figure 12.2 shows an example of a complete unified messaging system.

The UM system in Figure 12.2 supports Web, e-mail, IM, and phone clients, and uses only the core protocols for the transport and control of these applications (HTTP, SMTP, SIP, and RTSP). Media (voice, fax, and video) is carried in RTP packets. Application programming interfaces are *not* required for interoperability in the open, standards-based unified message system. This unified message system is built entirely along the lines of the component server architecture described in more detail in Chapter 19, "SIP Component Services," where the only parameter the individual servers have to know about each other are the respective URIs. Each server can be developed independently, without any knowledge of the internal working of the other corresponding servers.

We will exemplify in the following discussions some relevant message exchanges for unified messaging.

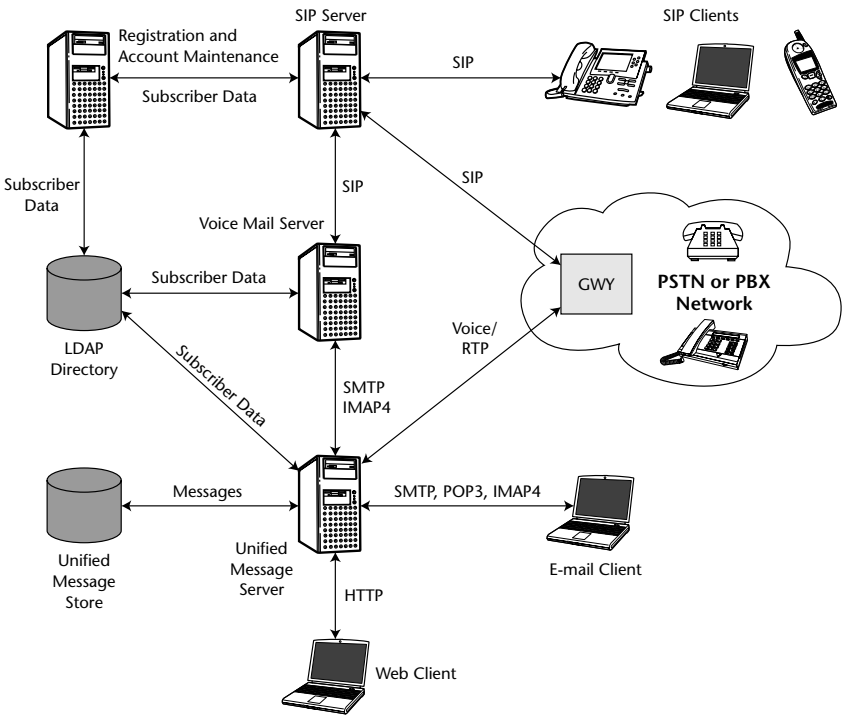


Figure 12.2 Complete unified messaging system

Depositing of Voice Messages

Voice messages can be created for various reasons, such as if the called party is not available, busy, there is no reply, or for other reasons. SIP can also support such caller preferences where the caller prefers voicemail anyway to avoid speaking to a party that would take up too much time. See the section "Prefer Voicemail" in [3].

It is quite simple in a pure Internet environment to connect as caller to voicemail by using SIP Service Control based on the design of URIs for services [4]. If, for whatever reason, the SIP proxy or the SIP UA wants to direct the call to voicemail, they can use the request URI to specify the destinations, as shown in this example:

```
sip:UserB-dep-fb@vm.mci.com
```

In the user part of the SIP URI, instead of `UserB` only, new target attributes are specified as well: `dep` for message deposit and `fb` to indicate the reason is forward on busy. Note the simplicity of this approach in a pure Internet environment.

The design challenges are, however, more complex when compatibility with TDM voicemail systems is required (such as for phone companies with legacy TDM switch systems and in enterprises with TDM-based PBX systems). Especially in environments like call centers, the reason code and the so-called history of the call are important indicators. To meet the requirements in mixed TDM and IP systems, a different encoding of the desired service of the SIP URI has been proposed [5], where the SIP error code can indicate the reason for retargeting the call. This type of usage opens more capabilities. Here is an example:

```
sip:voicemail@example.com;target=bob%40example.com;cause=486
```

In this example, `voicemail` is the user part in the domain `example.com` and the actual target is `bob@example.com` (where `@` is replaced with `%40` as per the syntax for URI). The SIP reason code is 486 (user busy), and it is called here "`cause`". The `cause` parameter can have the values as shown in Table 12.1.

Table 12.1 Reason for Redirect Values in the SIP Request URI

REASON FOR REDIRECT	VALUE
Destination Unknown/No available	404
User Busy	486
No Reply	408

Table 12.1 *(continued)*

REASON FOR REDIRECT	VALUE
Unconditional	302
Deflection During Alerting	487
Deflection Immediate Response	480
Mobile Subscriber Not Reachable	503

Even more comprehensive information (such as when the call has been retargeted multiple times) can be conveyed using the Request History Information [6]. Comprehensive retargeting information can be used for compatibility with legacy TDM systems, to preserve their complex functionality.

The limitation for using rich reason codes or the Request History Information is that most SIP proxies need to be designed to understand these extensions to SIP and must be programmed to support the behavior required for such service.

An example of forwarding to voicemail with the reason indication is given here in the scenario where the endpoint forwards on busy to voicemail. This example shows the service based on a SIP proxy server but is also of interest for peer-to-peer (P2P) SIP, since the forwarding function is performed by the target endpoint. So, for certain causes (such as busy or no answer), this example can also work without the SIP server in a pure P2P fashion. The flow diagram is shown in Figure 12.3.

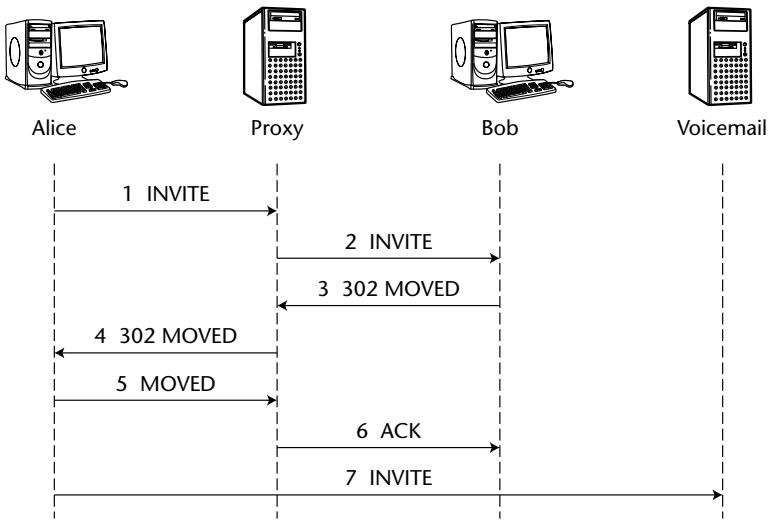


Figure 12.3 Flow diagram for voice message deposit with indication of cause

The main messages in the flow diagram in Figure 12.3 are given here. Message 3 is a redirect of the type 302 to the voicemail "`sip: voicemail@example.com`". Note the cause given in Message 7 — `cause=486` (user busy).

```
F2: INVITE proxy.example.com -> 192.0.2.2

    INVITE sip:line1@192.0.2.2 SIP/2.0
    Via: SIP/2.0/TCP 192.0.2.4:5060;branch=z9hG4bK-ik80k7g-1
    Via: SIP/2.0/TCP 192.0.2.1:5060;branch=z9hG4bK-74bf9
    From: Alice <sip:+15551001@example.com;user=phone>;tag=9fxced76sl
    To: <sip:+15555551002@example.com;user=phone>
    Call-ID: c3x842276298220188511
    CSeq: 1 INVITE
    Max-Forwards: 70
    Contact: <sip:alice@192.0.2.1;transport=tcp>
    Content-Type: application/sdp
    Content-Length: ...

    * SDP goes here *

F3: 302 192.0.2.2 -> proxy.example.com

    SIP/2.0 302 Moved Temporarily
    Via: SIP/2.0/TCP 192.0.2.4:5060;branch=z9hG4bK-ik80k7g-1
    Via: SIP/2.0/TCP 192.0.2.1:5060;branch=z9hG4bK-74bf9
    From: Alice <sip:+15551001@example.com;user=phone>;tag=9fxced76sl
    To: <sip:+15555551002@example.com;user=phone>;tag=09xde23d80
    Call-ID: c3x842276298220188511
    CSeq: 1 INVITE
    Contact: <sip:voicemail@example.com;\
           target=sip:+15555551002%40example.com;user=phone;\
           cause=486;>
    Content-Length: 0

F7: INVITE proxy.example.com -> um.example.com

    INVITE sip:voicemail@example.com;\
           target=sip:+15555551002%40example.com;user=phone;\
           cause=486  SIP/2.0
    Via: SIP/2.0/TCP 192.0.2.4:5060;branch=z9hG4bK-ik80k7g-2
    Via: SIP/2.0/TCP 192.0.2.1:5060;branch=z9hG4bK-74bf9
    From: Alice <sip:+15551001@example.com;user=phone>;tag=9fxced76sl
    To: <sip:+15555551002@example.com;user=phone>
    Call-ID: c3x842276298220188511
    CSeq: 1 INVITE
    Max-Forwards: 70
    Contact: <sip:alice@192.0.2.1;transport=tcp>
```

```
Content-Type: application/sdp
Content-Length: ...

* SDP goes here *
```

Notification for Waiting Messages

Traditional analog and PBX phones have lamps that signal when there are voice messages to be retrieved. Users content to have a similar experience with SIP-based systems will require only a minimum of information when alerted by a message-waiting signal.

The UA (typically a SIP phone, a PC client, or other type of device) will use the SIP `SUBSCRIBE` method to receive `NOTIFY` messages for changes of state in the mailbox. The UA also can explicitly fetch the status of the mailbox.

The UA can subscribe to multiple mailboxes distinguished by the URIs in the `To` headers.

Multiple UAs can subscribe to the same account. This allows the use of several devices to retrieve waiting messages (for example, using a SIP desktop phone or PC in the office, or a laptop, PDA or mobile phone while traveling).

We will illustrate in the following example the message notification using SIP UAs. As shown in Figure 12.2, message waiting notification can also be achieved through e-mail as an option, but this is not illustrated here since it is trivial, though very useful.

Simple Message Notification Format

The simple message waiting format [3] can convey only summary information about the status of the mailbox:

- Media type: e-mail, IM, voicemail, fax, and video mail
- Message status: new/old mail
- Urgent/normal messages

For example, the message-waiting summary `Voicemail: 1/3 (0/1)` conveys the information that there is one new message, three old messages, of which zero new messages are urgent and one old message was urgent. If no such details are required, the message-waiting summary could simply be `Messages-Waiting: yes`.

Figure 12.4 shows the message exchange required to subscribe the UA to the mailbox and the notification messages using SIP events [7].

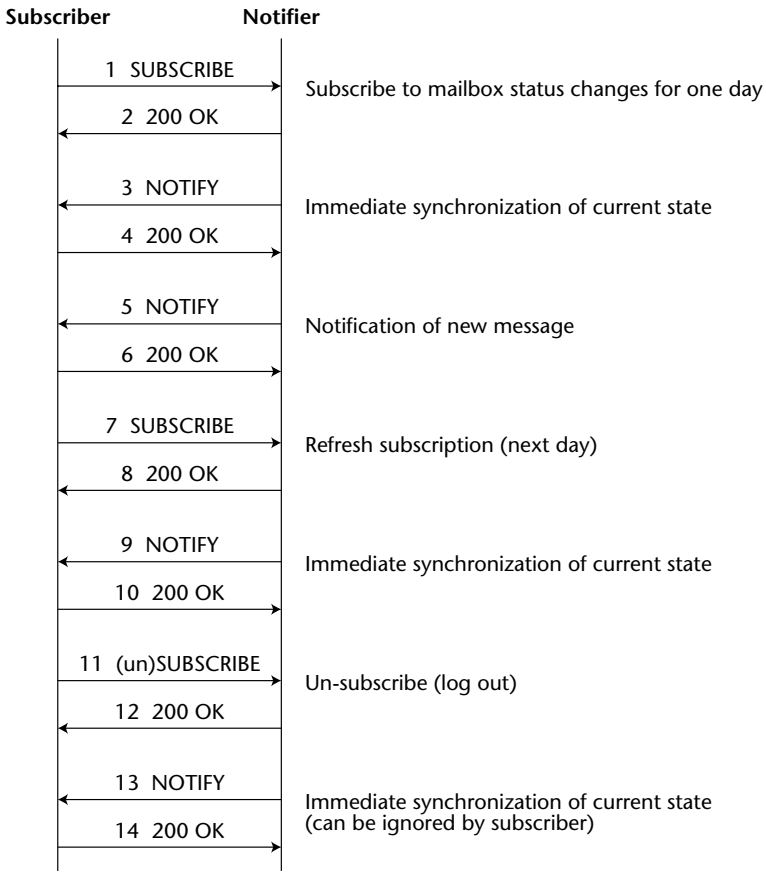


Figure 12.4 Message waiting notification using SIP events

Writing out in full the first three messages from Figure 12.4 will explain the use of SIP `SUBSCRIBE`, `200 OK`, and `NOTIFY` [8] messages. Note the use of the Secure SIP (`sips`) URI scheme in this example.

1. Subscriber (Henry's PC) to Notifier (Henry's voicemail server) Subscribe to Henry's message summary status for one day:

```
SUBSCRIBE sips:henry@mail3.mci.com SIP/2.0
Via: SIP/2.0/TLS 192.0.2.1:5060;branch=z9hG4bK83
To: <sips:henry@mail3.mci.com>
From: <sips:henry@mci.com>;tag=312
```

```
Date: Thu, 12 Apr 2005 15:30:00 GMT
Call-ID: 314@henrys-phone.mci.com
Max-Forwards: 70
CSeq: 4 SUBSCRIBE
Contact: <sips:henry@henrys-phone.mci.com>
Event: message-summary
Expires: 86400
Accept: application/simple-message-summary
Content-Length: 0
```

2. Notifier to Subscriber:

```
SIP/2.0 200 OK
Via: SIP/2.0/TLS 192.0.2.1:5060;branch=z9hG4bK83
To: <sips:henry@mail3.mci.com>;tag=221
From: <sips:henry@mci.com>;tag=312
Date: Thu, 12 Apr 2005 15:30:00 GMT
Call-ID: 314@henrys-phone.mci.com
CSeq: 4 SUBSCRIBE
Event: simple-message-summary
Expires: 86400
Content-Length: 0
```

3. Notifier to Subscriber: Immediate synchronization of current state. There are two new and eight old messages, of which two old messages are urgent:

```
NOTIFY sips:henry@henrys-phone.mci.com SIP/2.0
Via: SIP/2.0/TLS 192.0.2.1:5060;branch=z9hG4bK82
To: <sips:henry@mail3.mci.com>;tag=221
From: <sips:henry@mci.com>;tag=313
Date: Thu, 12 Apr 2005 15:30:00 GMT
Call-ID: 314@henrys-phone.mci.com
Max-Forwards: 70
CSeq: 20 NOTIFY

Contact: <sips:henry@vmail.mci.com>
Subscription-State: active; expires=86400
Event: simple-message-summary
Content-Type: application/simple-message-summary
Content-Length: 145

Messages-Waiting: yes
Voicemail: 2/8 (0/2)
```

Rich Message Notification Format

Though richer notification than in the previous example (`simple-message-summary`) has not been standardized, it can also be accomplished as shown here [9] by using XML documents that describe the message summary:

```
<!DOCTYPE message_summary SYSTEM xml_mwi.dtd>
   <MESSAGE_SUMMARY>
      <MAILBOX_IN>
        <NAME>Inbox</NAME>
        <VOICEMAIL>
                <UNTOUCHED urgent="1">2</UNTOUCHED>
                <SKIPPED>1</SKIPPED>
                <READ>3</READ>
                <DELETED>2</DELETED>
        </VOICEMAIL
        <FAX>
                <READ>1</READ>
        </FAX>
        <VIDEO/>
      </MAILBOX_IN>
      <MAILBOX_IN>
        <NAME>Inbox.Priority</NAME>
        <VOICEMAIL/>
        <EMAIL>
                <UNTOUCHED urgent="1">101</UNTOUCHED>
                <SKIPPED/>
                <FLAGGED urgent="2">4</FLAGGED>
                <READ>3</READ>
                <ANSWERED>2</ANSWERED>
                <DELETED/>
        </EMAIL>
      </MAILBOX_IN>
   </MESSAGE_SUMMARY>
```

A text-to-speech converter can read this document, providing the following voice message summary:

```
"You have reached the mailbox of <Name>"
"Inbox:"
"You have the following voice messages:"
"You have two skipped messages"
"You have three read messages"
"You have two deleted messages"
"This was the summary of your voice mail"
"You have the following fax mail:"
"You have one read fax"
"This was the summary of your fax mailbox"
"You have no video mail"
"Priority inbox of <NAME>"
"You have the following email:"
```

```
"You have 101 untouched email messages, one urgent message"
"You have no skipped email messages"
"You have four messages flagged urgent two"
"You have two answered messages"
"You have no deleted messages"
"This is the end of your email inbox"
"This was the complete message summary".
```

It is interesting to note that the same XML script can be used by a graphic application to paint the graphic user interface, showing the folders for e-mail, voice, fax, and video mail, with subfolders for normal and priority mail, and the respective icons for normal, urgent, skipped, and deleted messages.

This example shows clearly why looking at the messaging summary displayed on a Web page is preferable to listening to the same information on a phone.

Retrieval of Messages

As mentioned, voicemail can be retrieved either using a SIP UA (such as a SIP phone, a PC or some mobile device), by using e-mail or a Web page and a media player.

Summary

SIP can support all forms of messaging in universal messaging systems, as well as supporting the specific requirements for voicemail that matches the voicemail features in legacy TDM systems. The use of the Internet allows the integration of messaging for all media (such as text, voice, fax, and video) in a consistent fashion, while giving the user the options of retrieving messages on various devices, ranging from the PC to the simple phone.

References

[1] "Unified Messaging Using SIP and RTSP" by H. Schulzrinne and K. Singh. IP Telecom Services Workshop, Atlanta, Georgia, September 11, 2000.

[2] "Real-Time Streaming Protocol (RTSP)" by H. Schulzrinne, A. Rao, and R. Lanphier. RFC2326, IETF, April 1998.

[3] "Guidelines for Usage of the SIP Caller Preferences Extension" by J. Rosenberg and P. Kyzivat: Internet Draft, IETF, October 2005, work in progress.

[4] "Control of Service Context Using SIP Request-URI" by B. Campbell and R. Sparks. RFC 3087, IETF, April 2001.

[5] "SIP URIs for Applications such as Voicemail and Interactive Voice" C. Jennings et al. Internet Draft, IETF, November 2005.
[6] "An Extension to the Session Initiation Protocol (SIP) for Request History Information" by M. Barnes. RFC 4244, IETF, November 2005.
[7] "SIP-Specific Event Notification" by A. Roach. RFC 3265, IETF, June 2002.
[8] "A Message Summary and Message Waiting Indication Event Package for SIP" by R. Mahy. RFC 3842, IETF, August 2004.
[9] "SIP Extensions for Message Waiting Indication" by R. Mahy and I. Slain. Expired Internet Draft, IETF, July 2000.

CHAPTER 13

Presence and Instant Messaging

As of this writing, Instant Messaging (IM) services that also provide VoIP seem to surpass by far the VoIP services offered by telephone companies, whatever metric one may want to chose: Number of subscribers, revenue, or traffic statistics. Well-known IM services are available from large providers such as Apple, AOL, Google, IBM, ICQ, Microsoft, Skype, Yahoo!, and also from an increasing number of smaller providers.

NOTE **As mentioned in Chapter 4, "DNS and ENUM," in the sidebar "Do you really have VoIP?" we do not consider a service to be true VoIP unless the user gets a URI and can initiate calls to other URIs, besides using phone numbers.**

The dominance of IM services over VoIP services seems to be empirical proof that voice may eventually be relegated to an application within an IM service, together with video, white board, collaboration, application sharing, file transfer, and so on. From this perspective, telephony seems to be a business model that is being made obsolete by the Internet, for all users who have Internet access, wired or wireless.

URIs enable presence and IM-based communications and applications, while phone numbers and using the PSTN or PBX do not.

The Potential of SIP Presence, Events, and IM

Presence has been predicted to be the dial tone of the twenty-first century. Here are some of the new communication capabilities enabled by presence:

- Replace the dialing of phone numbers or the typing of URIs with a single click on the icon of the buddy (a frequently called party).
- The icon can replace numerous phone numbers and URIs for all the devices and network services the called party may have, since the Address of Record (AOR) in the SIP registrar can resolve the AOR to many Contact addresses.
- Using the presence icon, we can determine if the desired party is available on line, is in a good mood, is busy, is in a place where he or she must be quiet (in a meeting), and so on.
  - Presence enables polite and sensitive communications.
  - Presence avoids futile calls that may terminate in voice mail and require the called party to listen to sometimes long, ranting voice messages, and then try to call back only to get voice mail in turn.
- SIP events enable the integration of communications and applications, as we will discuss here.

IM is also an often preferred communication mode because of the following:

- IM works in emergencies when there may be network congestion.
- IM works in quiet places: People often exchange IM during a conference call with other parties (for example, to get key information or to evade boredom while in conference).
- IM enables the quick redirection to valuable information by typing a web URI.
- IM can bridge language difficulties, since the written words may be easier to understand.
- Agents in customer contact centers or in financial institutions can multiplex communications with several parties at the same time, in contrast to phone calls that can be conducted with only one party at a time. Enhanced agent productivity may be a significant reason to invest in and to use IM. IM log files can serve as legal proof for important financial transactions.

These capabilities of presence and IM have made the voice-only PBX or IP PBX obsolete for enterprise communications.

The Evolution of IM and Presence

The first widespread use of IM was ICQ and AOL's own Instant Messenger, which proved to be so popular that many non-AOL customers signed up for a free IM account. The companion "Buddy List" (which allows a user to be notified when a specified set of users is active) also represents a basic presence client. However, the first IM products used proprietary protocols and a centralized server architecture.

A large number of proprietary IM services have emerged on the Internet. Unfortunately, as the number of incompatible IM services grows, their convenience for users goes down, since users need to keep several IM applications running all the time on their PCs. The security of proprietary IM systems is also not known and, as a consequence, such systems must be considered a vulnerability.

Efforts by various IM developers to interwork using IM gateways have not been completely successful for the following reasons:

- Proprietary IM protocol updates make the gateway service transitory.
- Even technically well-working gateways between IM systems are not enough, in our opinion, since agreements between the various IM services are also required.

These problems and the ever larger number of IM services make the IM gateway solution neither durable nor scalable.

As a result, there has been a strong push in the industry to develop an open standard, interoperable, and scalable protocol for IM, similar to Internet e-mail. This has lead to the formation of the IETF Instant Messaging and Presence Protocol Working Group (IMPP WG). This group has produced two key documents on requirements and a model for presence and IM. It soon became apparent, however, that:

- IM and presence service may be used for all other communication services, beyond short text messaging.
- IM by itself can be implemented using various protocols.

The commonalities and differences were clearly articulated in the IMPP WG, and it was felt the different approaches may meet different needs and should have a common model and data exchange format for interoperability between the various protocols. The key document, the *Common Profile for Instant Messaging* (*CPIM*) [1], was the result of this agreement in the IMPP WG.

In conclusion, the internal protocols and data formats of various IM systems are a local business decision, but interoperability between IM systems should be possible via CPIM. A key step in the IT market was the technical agreement by AOL, IBM, and Microsoft on presence and IM interoperability using SIMPLE/SIP (SIMPLE stands for SIP for IM and Presence Leveraging Extensions).

The IETF Model for Presence and IM

Presence and IM are made possible by the packet nature of the Internet and may merit dedicated books on their own. We will attempt to give some basic notions that help in understanding the new SIP-based IM services and their potential. The IETF model for a presence service is shown in Figure 13.1.

The model is meant to help the understanding of the basics and is not a standard by itself. Note the distinction in the model for presence between the user agents (UA) of the principals, shown here as people, the Presence UA and the Watcher UA, and Presentity and Watcher functions that may be network-based elements or co-located with the respective UAs. The presentity is an application that manages the presence information and presents it to the presence service.

The Presence service is an abstraction that communicates with the presentity and watcher using the Presence Protocol, the SIMPLE protocol in our case.

The presentity information is displayed to the watcher on the UA as a "buddy" or "contact." We will use here the term "buddy" so as to avoid confusion with the "contact" as in the Contact address. A watcher usually has a buddy list and one can see their status at a glance.

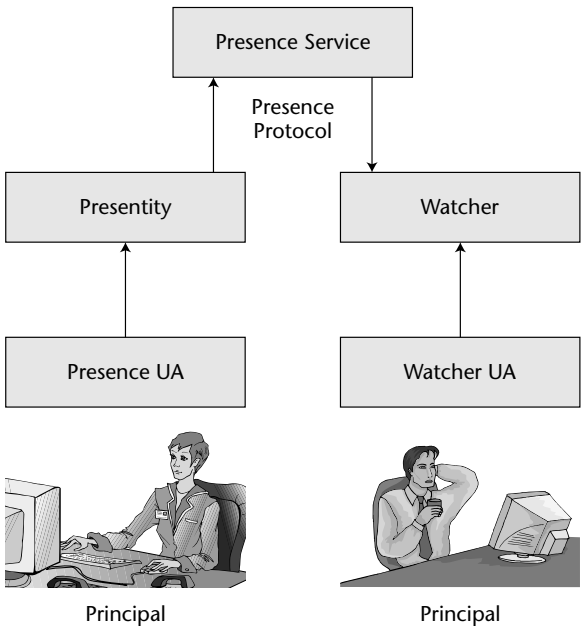


Figure 13.1 Model for presence service

As we will discuss later, a principal can also be an application. If the watcher declares an application to be just another buddy, the status of the application can be monitored at a glance on the watcher UA and an interaction with the application can be started by clicking on the buddy icon.

The model for instance IM is similar and is shown in Figure 13.2.

Both services have other similarities such as the notions of *principals*, which can be either people or software and that appear to the service as a single entity. Principals interact with the system via *user agents*. A UA is the coupling between the principal and some core entity in the system.

Both the presence and the IM services may have complex internal structures with specific servers and/or proxies. There may also be quite complex security implementations to protect the presence and IM services from various attacks, and to make sure they don't communicate with other systems that are either from competitors or that cannot be trusted for security.

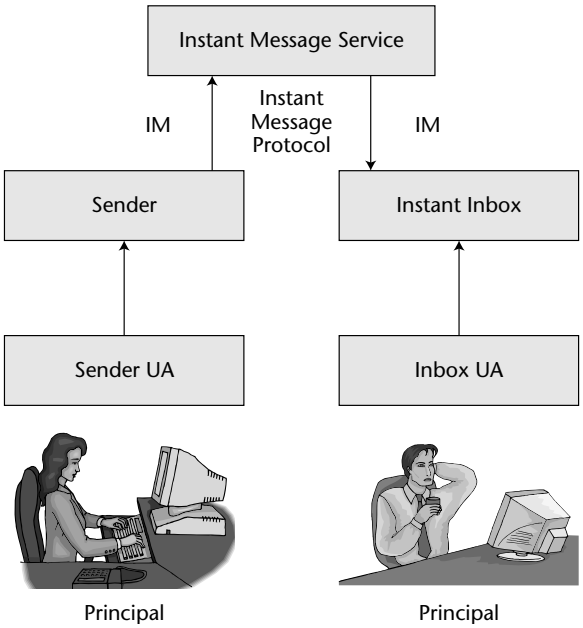


Figure 13.2 Model for IM service

Client Server and Peer-to-Peer Presence and IM

In keeping with the end-to-end control principle of the Internet, presence, and IM services can also be implemented in the endpoints, without dependence on intermediate elements in the network, as is the case with SIP. Figure 13.3 shows a client-server (CS) based system (a) and a peer-to-peer (P2P) system (b).

Several operation modes are possible:

- Pure client-server, as in Figure 13.3a
- Pure peer-to-peer to peer without any server, using a P2P protocol (see Chapter 20, "Peer-to-Peer SIP"
- Rendezvous for the SIP UAs using the SIP server, and then operating presence and IM in a peer-to-peer mode, as indicated in Figure 13.3b

Both CS SIP and P2P SIP presence and IM have been implemented in well-known commercial products, as seen in Figure 13.4a and in Figure 13.4b, respectively.

The Microsoft Messenger 4.7 integrated with Microsoft Office 2003 shown here is one of the first integration examples known to the authors for presence with an application. Figure 13.4 a shows the e-mail in the Microsoft Outlook client, as well as the presence icon for the author of the message. By clicking on the author buddy icon of an Office 2003 document, an audio/video conversation can be invoked by the reader of the document to discuss it with the author.

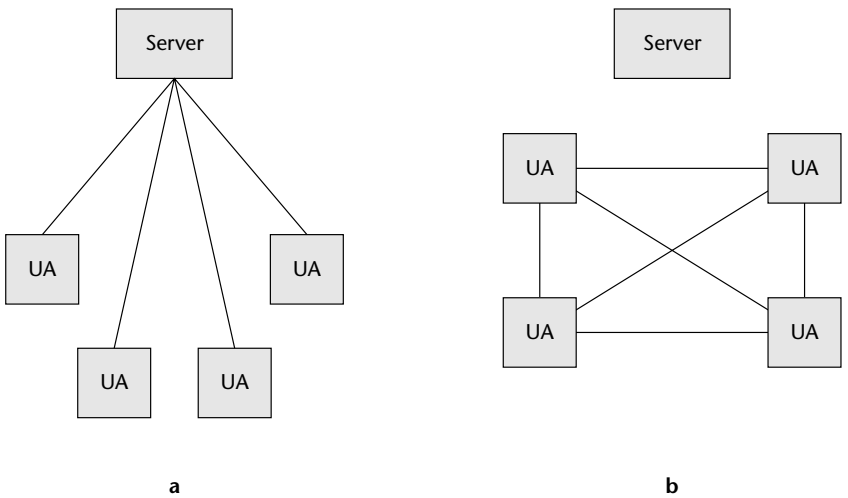
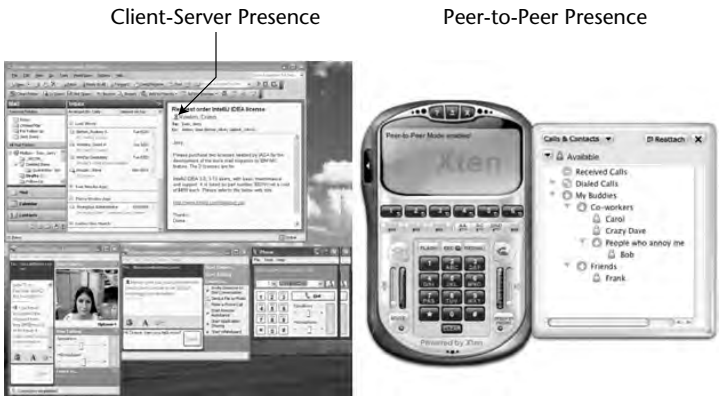


Figure 13.3 (a) Client-server and (b) P2P presence and IM systems



a b

Figure 13.4 Commercial presence and IM products: (a) Microsoft Messenger integrated with Microsoft Office 2003 and (b) the CounterPath eyeBeam SIP UA

SIP Event-Based Communications and Applications

SIP presence is a form of a SIP-specific event notification [2] also called "SIP Event Package." SIP events can support the many types of SIP services and their integration with applications. Examples of notification-based services include automatic callback based on the presence of the called party, message waiting indicators, and alerts caused by communication events (such as alerting a supervisor when a customer is calling an agent in a contact center).

Not all applications are good candidates for SIP events. Frequent notifications about the geographic location provided by a GPS location service for a moving person would probably overload the SIP system and be of little value.

The operation for SIP `NOTIFY` flow is shown in Figure 13.5.

As shown in Figure 13.5, the subscriber sends a `SUBSCRIBE` message to the notifier to request a subscription to an application state. The notifier will acknowledge the request with a `200 OK` message and also send a `NOTIFY` message with the complete current state. As soon as there is some change in state, the notifier will send either updates for the new state or complete new state information to the subscriber. Sending only updates or all the new state information depends on the type of application.

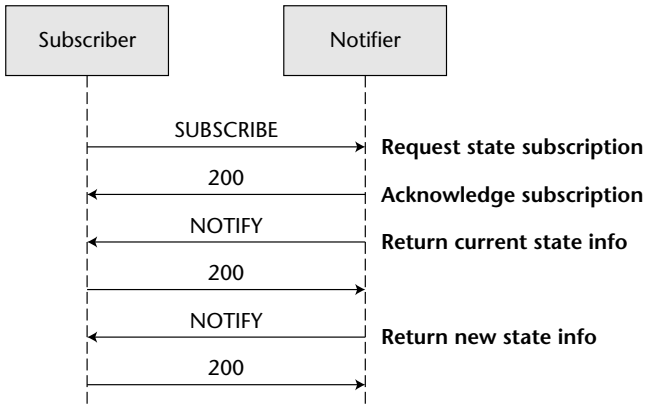


Figure13.5 SIP NOTIFY flow

SIP event packages have been proposed and developed for numerous applications, some of which have been reported in IETF Internet drafts, though not all of them have made it to the standards level. As seen from the following list (which is not complete), event-based SIP communications constitute a significant area for product innovation. We believe many such innovations are likely to proliferate in IP communications based on SIP. SIP event packages include the following:

- Automatic updates of Internet Media Guides [3]
- Registrations of SIP UA with servers to change or enforce policy [4]
- Notifications about the dialog state of `INVITE` initiated dialogs [5]
- Message waiting alerts [6]
- Conference state (such as reporting the attendance and speaker name) [7]
- Push-to-talk over cellular networks [8]
- Key events for buttons, feature keys, and so on [9]
- Key press stimulus for monitoring DTMF signals using XML documents [10]
- Location events using location filters [11]
- Remote device configuration and status information [12]
- Call control events happening in the PSTN [13]

- Network preemption events for priority calls [14]
- Last but not least, *presence* as described in the following section.

Presence Event Package

Presence information conveys the ability and willingness of a user to communicate with the watcher [15]. As mentioned, presence can also be used to communicate with applications, though not all application events are useful presence data and SIP `NOTIFY` is not indicated as a universal tool.

Presence events are best illustrated with an example, as in Figure 13.5.

The example in Figure 13.5 is client-server-based, though we should keep in mind that peer-to-peer is also possible, as shown in Figure 13.3.

The authentication and registration of the watcher by the server are not shown here for simplicity.

An example for the presence message flow is shown in Figure 13.6.

The watcher will first subscribe to the presence information from the server as shown in Messages 1 and 2. The watcher will also receive the notification of the presence information and acknowledge it in Messages 3 and 4.

An update of the presence information to the server by the presence UA will be immediately followed by a new notification shown in Message 5.

Polite blocking is possible for unwanted watchers. A `200 OK` is generated or marked pending as a response to the `SUBSCRIBE` request, even though the request has been rejected [16].

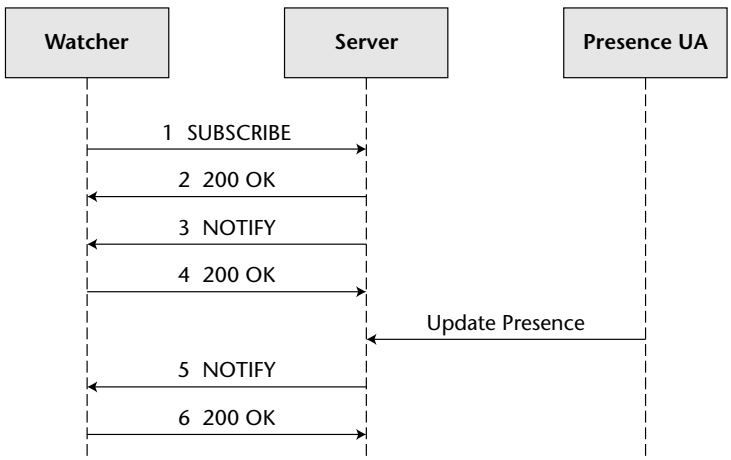


Figure 13.6 Example of SIP message flow for presence

We will show here the first three messages to better understand how presence works.

Message 1: SUBSCRIBE watcher->example.com server

```
SUBSCRIBE sip:resource@example.com SIP/2.0
Via: SIP/2.0/TCP watcherhost.example.com;branch=z9hG4bKnash4
To: <sip:resource@example.com>
From: <sip:user@example.com>;tag=xfg9
Call-ID: 2010@watcherhost.example.com
CSeq: 17766 SUBSCRIBE
Max-Forwards: 70
Event: presence
Accept: application/pidf+xml
Contact: <sip:user@watcherhost.example.com;transport=tcp>
Expires: 600
Content-Length: 0
```

Message 2: 200 OK example.com server->watcher

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP watcherhost.example.com;branch=z9hG4bKnashds7
  ;received=192.0.2.1
To: <sip:resource@example.com>;tag=ffd2
From: <sip:user@example.com>;tag=xfg9
Call-ID: 2010@watcherhost.example.com
CSeq: 17766 SUBSCRIBE
Expires: 600
Contact: <sip:server.example.com;transport=tcp>
Content-Length: 0
```

Message 3: NOTIFY example.com server-> watcher

```
NOTIFY sip:user@watcherhost.example.com SIP/2.0
Via: SIP/2.0/TCP server.example.com;branch=z9hG4bKna998
From: <sip:resource@example.com>;tag=ffd2
To: <sip:user@example.com>;tag=xfg9
Call-ID: 2010@watcherhost.example.com
Event: presence
Subscription-State: active;expires=599
Max-Forwards: 70
CSeq: 8775 NOTIFY
Contact: <sip:server.example.com;transport=tcp>
Content-Type: application/pidf+xml
Content-Length: ...

[PIDF Document]
```

Note in the last line, [PIDF Document] stands for the *Presence Information Data Format* document that will be described in the following sections.

Presence Information Data Format

The Presence Information Data Format (PIDF) is a standard [17] for minimal presence information. PIDF is based on the model for presence and IM [18] where the structure of presence information has been defined, as shown in Figure 13.7.

PIDF is best understood using an example.

In this example, the presence information says the entity someone@example.com is online (open), is busy, and home. The presence document says further that the preferred contact is someone@mobilecarrier.net, but Don't Disturb Please! and it also says it in French, Ne derangez pas, s'il vous plait. A timestamp is associated with this document. An even higher-contact priority (1.0) is to use e-mail to someone@example.com. This contact is also online. A final note says, I'll be in Tokyo next week.

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
     xmlns:im="urn:ietf:params:xml:ns:pidf:im"
     xmlns:myex="http://id.example.com/presence/"
     entity="pres:someone@example.com">
  <tuple id="bs35r9">
    <status>
      <basic>open</basic>
      <im:im>busy</im:im>
      <myex:location>home</myex:location>
    </status>
    <contact priority="0.8">im:someone@mobilecarrier.net</contact>
    <note xml:lang="en">Don't Disturb Please!</note>
    <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
    <timestamp>2001-10-27T16:49:29Z</timestamp>
  </tuple>
  <tuple id="eg92n8">
    <status>
      <basic>open</basic>
    </status>
    <contact priority="1.0">mailto:someone@example.com</contact>
  </tuple>
  <note>I'll be in Tokyo next week</note>
</presence>
```

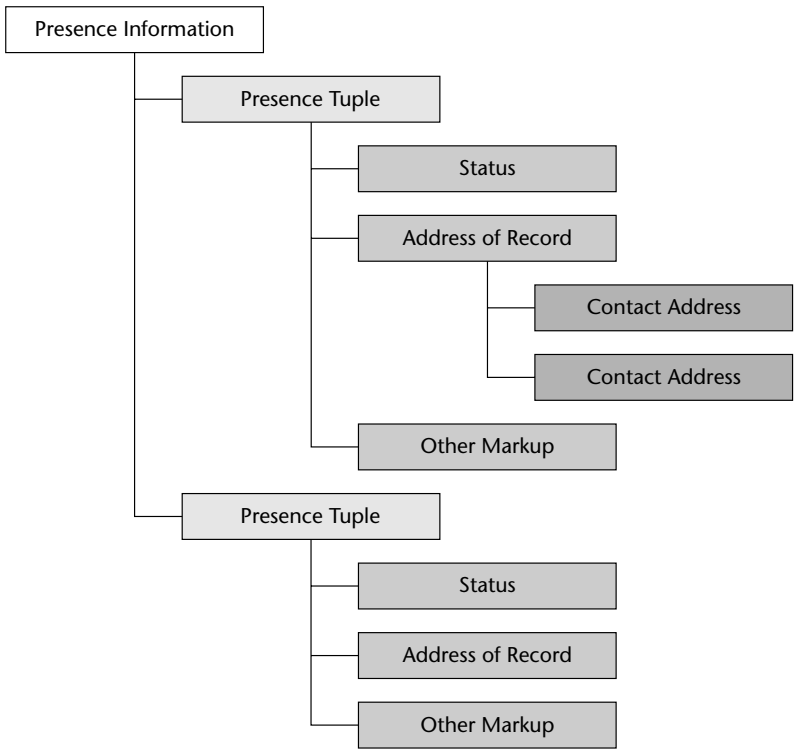


Figure 13.7 The structure of presence information

The example will now make it easier to understand some presence elements in the PIDF standard.

- The PIDF object is a well formed XML document that contains the encoding declaration `<?xml version="1.0" encoding="UTF-8"?>`.
- The `<presence>` element is associated with the XML namespace `"urn:ietf:params:xml:ns:pidf'"` and has a namespace declaration "`xmlns`" that points to the URN "'`urn:ietf:params:xml:ns:pidf`:". The namespace is identified by a URI that must be globally unique but does not necessarily represent an existing web resource.
- The `<tuple>` element consists of a mandatory `<status>` element followed by optional extension elements, such as `<contact>`, `<note>`, and `<timestamp>`.
- The optional `<note>` element is especially interesting, since it contains comments readable by humans, such as `I'll be in Tokyo next week` in the example.

The PIDF standard has great flexibility because of the XML data elements, as shown in the example, and this flexibility can lead to confusion. To avoid confusion, a data model is helpful and, as a consequence, a data presence model has been developed.

The Data Model for Presence

The data model for presence [19] introduces some useful definitions that we will mention here briefly:

- *Service*—Such as IM or telephony.
- *Device*—PC/laptop, PDA, phone.
- *Person*—The end user.
- *Occurrence*—There can be multiple occurrences for a service or multiple person occurrences in a presence document. Ambiguities are best resolved by the watcher.
- *Presentity*—The complete picture that combines the person, services, and devices for the user's presence status on the network.
- *Presentity* URI—The unique identifier for the presentity on the network.
- *Data component*—Part of the presence document that describes the person, service, or device.
- *Status*—Dynamic information about a person, service, or device.
- *Characteristics*—Static information about a person, service, or device.
- *Attribute*—A single piece of presence information (also called a *presence attribute*).
- *Composition*—Combining presence data into a coherent picture of presentity.

The main presence data elements and their relationships can be represented graphically as in Figure 13.8.

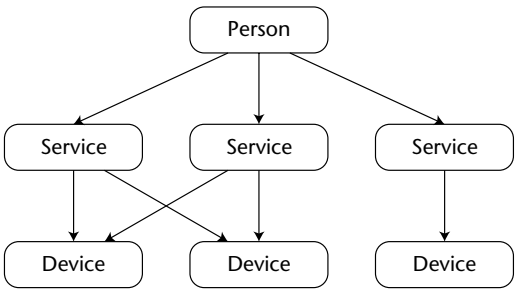


Figure 13.8 The presence data model

We refer the reader to the original document on the presence data model [15] for a detailed discussion of the meaning of the various presence data elements.

Indication of Message Composition for IM

Commercial IM systems have introduced a useful presence indication for IM users, also sometimes called `is-typing`, that alerts the remote party to wait for the message. This avoids the nuisance of crossed messages in an interactive communication. In the SIP standard for this type of presence notification, the status message is called `isComposing` and has been defined not only for text-based IM but also for voice and video exchanges that require some time to prepare [20]. A state machine has been defined that can understand the behavior of the composer with the help of activity timers and generate the `idle` and `active` states that are then communicated as presence information in an XML document with a well-defined namespace and XML schema.

Rich Presence Information

More comprehensive work on the use of presence is reflected in the Rich Presence Extensions to the Presence Information Data Format (RPID) [21]. The presence information data in RFC 3863 that is intended for humans has been extended to be generated by applications for consumption by automata, while maintaining backward compatibility with RFC 3863. Applications such as the calendar or UAs can generate such rich information presence, often without any human intervention. Rich presence information documents avoid the use of the `<note>` element that is intended for humans.

Rich presence data elements describe the person, the services, and the devices. Here are some examples for rich presence data elements:

- Activities (many can be generated by the calendar)
  - `appointment`
  - `away`
  - `breakfast`
  - `busy`
  - `dinner`
  - `holiday`
  - `in transit` (in a car)
  - `looking for work`
  - `lunch`
  - `meal`

- meeting
- on the phone
- playing
- performance (cinema, lecture, theater)
- presentation
- shopping
- travel
- vacation
- ...

- Device Identifier
- Mood Element
  - afraid
  - amazed
  - angry
  - annoyed
  - anxious
  - ashamed
  - bored
  - brave
  - calm
  - cold
  - confused
  - contented
  - ...
- Place-is (examples)
  - noisy
  - ok
  - quiet
- Place-type
  - aircraft
  - airport
  - arena
  - automobile
  - bank

- bar
- bus
- cafe
- classroom
- club
- ...

- Privacy element (the presence info may be intercepted in the vicinity)
  - audio
  - text
  - video
- Relationship element
  - associate
  - assistant
  - family
  - friend
  - supervisor
  - self
  - ...
- Service class
  - courier
  - electronic
  - freight
  - in-person
  - postal
  - ...
- Sphere
  - home
  - work
- Status icon. Example: `www.example.com/playing.gif`
- Time offset (at the person's current location)

Automata-readable rich presence information is extensible for various applications, though the actual list of elements will be tailored for each specific application or environment.

SIP Extensions for Instant Messaging

Applications that integrate IM with presence have been in use on the Internet for a long time, though not in a standard form. SIP, as initially designed, has useful mechanisms for presence but needs an extension for IM. The SIP extension for IM is `MESSAGE`, and `MESSAGE` closely resembles the `INVITE` method. `MESSAGE` relieves SIP from the session mode and can support conversations based on independent messages, initially using only text, but as we will see, multimedia conversations are also supported without requiring the prior setting up of an explicit session. IM has therefore two modes of operation:

- Short individual messages, mostly text.
- Session mode when the conversation needs to be associated with a SIP session, for such application as for secure tunneling through NAT and firewalls between domains from different enterprises or for the transfer of large audio/video files. The protocol for the session mode is called the Message Session Relay Protocol (MSRP) [22], [23].

`MESSAGE` may traverse a number of SIP proxies, fork into different branches, and have 2xx response—in short, use the existing SIP infrastructure, with the caveat that the infrastructure needs to understand the `MESSAGE` extension as well. This is not an expensive proposition and saves complete separate systems for voice and IM and presence. Besides avoiding the cost of acquiring and operating different systems, users don't need separate applications for voice, IM, and presence, and can greatly benefit from their integration.

Sharing the SIP infrastructure for signaling and IM requires congestion control because of the high-volume usage of IM. For this reason, it may be advantageous to have SIP endpoints exchange `MESSAGE` directly in a peer-to-peer fashion, similar to RTP media packets. This also has the advantage of IM text being treated in central conferences in the same manner as RTP media packets for consistent multipoint centralized conferencing architecture.

An IM inbox has an Instant Message URI in the form `im:user@domain`.

Figure 13.9 shows an example of the IM message exchange between two SIP user agents [24].

Note here two interesting properties of IM using SIP:

- IM is can be based on the common infrastructure with voice.
- The same SIP UA can support voice with one party or parties and while at the same time, use IM with another party.

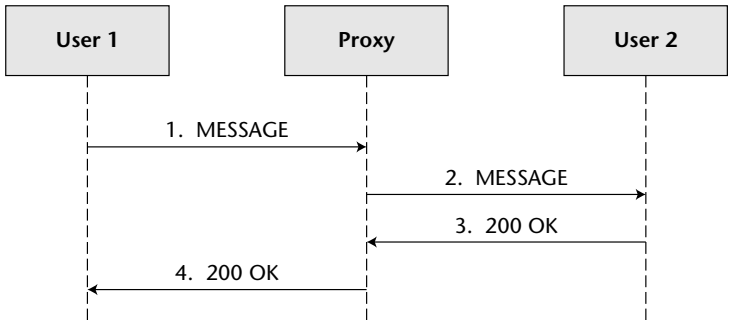


Figure 13.9 Example message flow for SIP based instant messaging

SIP `MESSAGE` is similar to the SIP `INVITE`. Here are two message examples for Figure 13.9.

Message 1

```
MESSAGE sip:user2@domain.com SIP/2.0
Via: SIP/2.0/TCPuser1pc.domain.com;branch=z9hG4bK77
Max-Forwards: 70
From: <sip:user1@domain.com>;tag=49583
To: sip:user2@domain.com
Call-ID: asd88asd77a@1.2.3.4
CSeq: 1 MESSAGE
Content-Type: text/plain
Content-Length: 18

Watson, come here.
```

Message 2

```
MESSAGE sip:user2@domain.com SIP/2.0
Via: SIP/2.0/TCP proxy.domain.com;branch=z9hG4bK123
Via: SIP/2.0/TCP ser1pc.domain.com;branch=z9hG4bK77
 ;received=1.2.3.4
Max-Forwards: 69
From: <sip:user1@domain.com>;tag=49394
To: sip:user2@domain.com
Call-ID: asd88asd77a@1.2.3.4
CSeq: 1 MESSAGE
Content-Type: text/plain
Content-Length: 18

   Watson, come here.
```

Message 3

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP proxy.domain.com;branch=z9hG4bK123
 ;received=192.0.2.1
Via: SIP/2.0/TCP user1pc.domain.com
 ;branch=z9hG4bK77; received=1.2.3.4
From: <sip:user1@domain.com>;tag=49394
To: <sip:user2@domain.com>;tag=ab8asdasd9
Call-ID: asd88asd77a@1.2.3.4
CSeq: 1 MESSAGE
Content-Length: 0
```

Message 4

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP user1pc.domain.com
 ;branch=z9hG4bK77; received=1.2.3.4
From: <sip:user1@domain.com>;tag=49394
To: <sip:user2@domain.com>;tag=ab8asdasd9
Call-ID: asd88asd77a@1.2.3.4
CSeq: 1 MESSAGE
Content-Length: 0
```

Presence and IM have serious security issues that are part of the larger security aspects for SIP discussed in Chapter 9, "SIP Security."

Summary

Presence and instant messaging are new forms of communications made possible by the advent of the Internet. SIP presence is a subset of SIP events, and there is an increasingly long list of communications and applications based on presence. The standard for the presence information data format (PIDF) has been enhanced with the Rich Presence Information Data format (RPID) that is tailored to communicate presence directly to machines.

IM can be text-based, but can also support audio visual communications. IM modes are the pager mode and the session mode using the Message Session Relay Protocol (MSRP).

IM and presence can work both in the client-server and peer-to-peer modes.

References

[1] "Common Profile for Instant Messaging" by J. Peterson:. RFC 3860, IETF, August 2004.
[2] "Session Initiation Protocol (SIP)-Specific Event Notification" by A. B. Roach. RFC 3265, IETF, June 2002.
[3] "SIP Event Notification for Internet Media Guides" by Y. Nomura and H. Schulzrinne. Internet Draft, IETF, July 2005, work in progress.
[4] "A SIP Event Package for Registrations" by J. Rosenberg. RFC 3680. IETF, March 2004.
[5] "An `INVITE` Initiated Dialog Event Package for SIP" by J. Rosenberg. RFC 4235, IETF, November 2005.
[6] "A Message Summary and Message Waiting Indication Event Package for SIP" by R. Mahy. Internet Draft. IETF, October 2002.
[7] "A SIP Event Package for Conference State" by J. Rosenberg et al. Internet Draft, July 2005.
[8] "A SIP Event Package and Data Format for Various Settings in Support for the Push-to-talk Over Cellular (PoC) Service" by A. Garcia-Martin:. Internet Draft, IETF, September 2005.
[9] "SIP Event Package for Keys" B. Culpepper et al. Internet Draft, June 2003.
[10] "A SIP Event Package for Key Press Stimulus" by E. Burger et al. Internet Draft, December 2005, work in progress.
[11] "A Location Event Package using the Session Initiation Protocol (SIP)" by R. Mahy. Internet Draft, IETF, July 2005.
[12] "A Session Initiation Protocol (SIP) Event Package for Device Information" by M. Rahman at al. Internet Draft, IETF, January 2005.
[13] "Toward the Definition of the SIP Events Package for SPIRITS Protocol" I. Faynberg et al. Internet Draft, IETF, September 2002.
[14] "Extending SIP Reason Header for Preemption Events" by J. M. Polk. Internet Draft. IETF, September 2005.
[15] "A Presence Event Package for SIP" by J. Rosenberg. RFC 3856. IETF, August 2004.
[16] "Instant Messaging/Presence Protocol Requirements" by M. Day et al. RFC 2779, IETF, February 2000.
[17] "Presence Information Data Format (PIDF)" by H. Sugano et al. RFC 3863. IETF, August 2004.
[18] "A Model for Presence and Instant Messaging" by M. Day. RFC 2778. IETF, February 2000.
[19] "A Data Model for Presence" by J. Rosenberg. Internet Draft, IETF, October 2005.
[20] "Indication of Message Composition for Instant Messaging" by H. Schulzrinne. RFC 3994. IETF, January 2005.

[21] "RPID: Rich Presence Extensions to the Presence Information Data Format (PIDF)" by H. Schulzrinne et al. Internet Draft, IETF September 2005, work in progress.
[22] "The Message Session Relay Protocol" by B. Campbell et al. Internet Draft, IETF, October 2005, work in progress.
[23] "Relay Extensions for the Message Sessions Relay Protocol (MSRP)" by C. Jennings et al. Internet Draft, IETF, July 2005, work in progress.
[24] "Session Initiation Protocol (SIP) Extension for Instant Messaging" by B. Campbell et al. RFC 3428, IETF, December 2000.

CHAPTER

14

SIP Conferencing

We will present in this chapter conferencing services based on SIP. Though the interest in SIP is at present mostly because of telephony and other IP communication services, it is useful to remember that SIP has been developed initially in the IETF MMUSIC Working Group for large-scale multimedia conferencing over the Internet within the Internet Multimedia Conferencing Architecture [1].

Introduction

Present commercial conferencing products and services are mainly of three types:

- PSTN-based telephone conferences, which are the most widely used.
- Various video conferencing products, such as based on the ITU H.3xx series of recommendations that support voice, video, and also document sharing, using the T.120 standard. Contrary to voice conferencing, most video and data conferencing products are not fully interoperable, though they may be advertised as being in compliance with the previous standards. This is because of the large number of options permitted in ITU conferencing standards and the fact that various products may support different sets of options, besides some proprietary enhancements.

- So-called Web Conferencing in which a browser window is used to present conferencing information such as a presentation or slide show, the participant roster, and so on.

The main issues we have with the H.3xx type of conferencing products is the way they are architected, based on ITU network models, and their technology, which makes them a poor fit for the Internet and the World Wide Web. Thus, integration with other communication and Web services is difficult, and there are a number of divergent approaches (such as for security and scalability).

Considerable effort is expended in both the ITU and IETF on interoperability between SIP and H.323 signaling, but this work is mainly for telephony. A summary on H.323-SIP internetworking aspects is provided in Schulzrinne and Agboh [2].

Web conferencing services are currently limited in that they require a full-featured Web browser, often with specialized plug-ins that must be installed prior to using the Web conferencing service. Smaller mobile devices are not able to be participants in Web conferences, even though they have a simple Web browser built in. Despite the use of the Web, Web conferencing systems actually represent yet another proprietary signaling channel that fails to integrate all media types and devices.

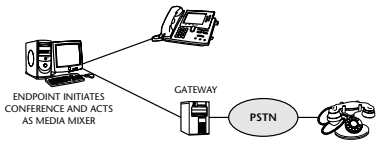
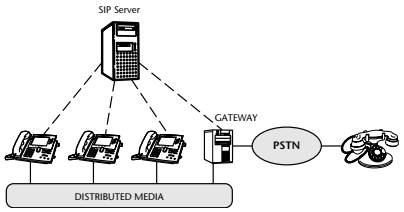
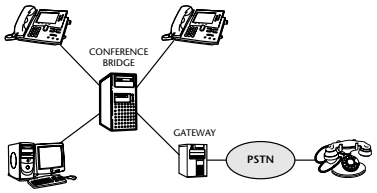
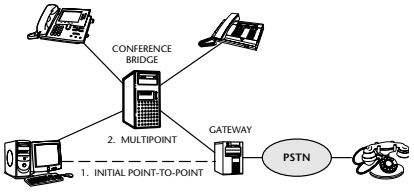
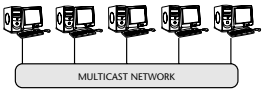
Most commercial telecom conferencing services are based on a central multipoint control unit (MCU) that serves both as the central signaling control point and media mixer. Rosenberg provides a detailed review of SIP conferencing models in the SIP Conferencing Framework [3]. The requirements for SIP conferencing are described in [4]. The Best Current Practice (BCP) document for SIP conferencing is defined in [5], which utilized the `isfocus` feature tag defined in RFC 3840 [6].

We will present in this chapter the Internet conferencing models and services based on SIP that include the MCU model but also have other flexible approaches for various other conferencing models.

SIP Conferencing Models

IP conferences may differ in many respects, depending on the signaling to set up the conference and the way media is transported and mixed for the conference participants. Table 14.1 shows the main conference models possible. The models are roughly ordered by the possible scale of the conference. We will use the generic term "conference bridge" in the table, since the conferencing network element is sometimes a SIP server only, and sometimes a SIP server combined with an RTP media mixer. Telephony conferencing network elements are sometimes called *multipoint controller units* to emphasize the control aspects for certain telephony conference types.

Table 14.1 SIP Conference Models

CONFERENCE MODELS	HOW IT WORKS
1. Endpoint mixing	
	Small conferences with three to nine participants. One endpoint handles signaling and also acts as media mixer, and is required to stay until the end of the conference. Endpoint bandwidth is often the limiting factor.
2. SIP Server and distributed media	
	The central SIP server establishes a full mesh of point-to-point RTP streams between all participants. Each participant mixes all the media it receives and plays out its own media to every participant. Media latency is minimized and end-to-end security maximized. However, media synchronization can be difficult.
3. Conference Bridge – as in PSTN conferences	
	Medium-sized conferences. Users dial in for the conference or the bridge can dial out to bring a participant into the conference. The bridge mixes media from other directions for each participant. The conference server also houses the conference applications. The bridge could support PSTN, SIP, and H.323, for example.
4. Ad hoc centralized conference	
	Two users may transition to a multiparty conference by having one user making the transition using SIP call control.
5. Large multicast conference	
	Very large-scale conferences, up to millions of users. Users join a multicast address announced on the Web, by e-mail, or Session Announcement Protocol (SAP), or are invited to join using SIP.

Each conference model shown in Table 14.1 differs from the other models by one or more of the following:

- Scale of the conference
- Call flows for users to join the conference
- How and where the media is sent and mixed
- Location of the service logic (in endpoints or in servers)

The conference models apply equally well for both audio-only (as in telephony conferences) and for mixed-media conferences (for audio, video, and text). Depending on the quality of video that users may send and receive because of bandwidth limitations, several IP addresses may be required for layered video codecs. Users with the lowest bandwidth may send and receive only the basic video layer, suitable for small images only. We will focus, for simplicity, on audio conference examples in the following.

Small-scale conferences do not require any support from network servers, since a few RTP streams may be mixed in one of the endpoints that originates the conference. Such a small-scale conference model is shown in the first row of Table 14.1.

In the second row of Table 14.1, a pure SIP service is shown, with no media mixing provided in a server. A SIP server for conferences can support conferences by setting up a full mesh of RTP streams between participants. Each participant mixes all incoming streams for individual use. Since it is unlikely to experience more than one or two speakers at the same time, the required RTP processing in the user endpoints is quite modest.

Telephony-style conferencing is shown in the third row in Table 14.1. The *conferencing bridge* (see Figure 14.1) is a conceptually simple device, consisting of a SIP user agent to handle signaling, an RTP mixer to handle the media streams, and a conference application layer for the authentication, authorization, and accounting (AAA) service, and possible conference control functions, as shown in Table 14.1. The RTP mixer will send out to each participant the mix of media streams from all other participants.

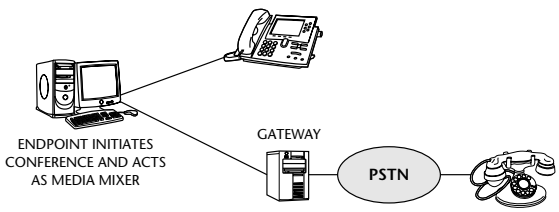


Figure 14.1 Generic conferencing bridge

An important requirement for commercial conference bridges is the capability to convey with confidence the list of all participating users in the conference to ensure participants the conference is private, with no undesirable parties listening in. The list of participants is transmitted by the RTP mixer, using the RTP capability to transmit the name of all registered participants. In a multicast conference, the CNAME in the Source Description (SDES) is transmitted to inform about participating users. It is the responsibility of the conference bridge to authenticate all participants (the AAA function) and to communicate the list on a dynamic basis using RTP. In a centralized conference, the participant list, or roster is communicated to the participants using the SIP Conferencing Event package [7]. In addition, the RTP mixer includes information about the current speakers in the Contributing SSRC (see Chapter 5, "Real-Time Internet Multimedia") or CSRC field. The SIP conference package provides a way to map a participant's URI to the SSRC in the RTP packet.

The conference model number 5 in Table 14.1 is for large multicast conferences, where IP multicast is available. Multicast conferences can scale up to millions of users and do not really require any SIP signaling. Users can join the multicast conference by addressing their RTP streams to/from the multicast addresses belonging to the particular conference. SIP may be used to inform users of the multicast conference address, though any other means to convey this information (such as Web pages, e-mail, and the SAP) are just as adequate.

Ad Hoc and Scheduled Conferences

Presence and instant messaging can support the setup of spontaneous conferences, in contrast to the more customary scheduled conferences, as used on the PSTN. It is of interest to note that SIP enables a continuous transition of conferences from ad hoc to scheduled conferences.

Participants in an ad hoc conference could agree, for example, they need more time and would like to invite other participants, so they set up a scheduled conference to discuss a topic in more depth. The list of the ad hoc conference participants can be included by the application to set up the scheduled conference. Call flows for ad hoc conferences and call control operations are described in [5].

Changing the Nature of a Conference

If users in an ad hoc conference with endpoint mixing (row 1 in Table 14.1) decide to increase the number of participants, they can move the conference to a central conference server so as to benefit from a dedicated RTP mixer, as shown in row 4 of Table 14.1. One of the parties must assume the responsibility

for moving existing participants to the centralized conference. The example provided here also is valid for moving from a two-party call to a centralized conference.

The following steps involve the move from an ad hoc to a centralized conference:

1. Discover a server that supports ad hoc centralized conferences, such as `conf.factory.example.com`.
2. Create a conference at the server by sending an `INVITE` to the Conference Factory URI [5].
3. The conference server creates the unique conference URI by assigning it a random number or string and returning it with the `isfocus` feature tag to indicate that the server is acting as a focus.
4. The application of the user in charge of migrating the ad hoc conference to the centralized server can now use the `REFER` method [8] in SIP call control to set up calls between the existing and new participants, and the conference server. See Chapter 19, "SIP Component Services," on third-party call control used in component services.

For example, John Doe initiates the call setup for Mary Higgins:

```
REFER sip:maryhiggins@example.net SIP/2.0
Via: SIP/2.0/UDP 192.0.0.4:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
From: <sip:johndoe@example.com>;tag=3412349dfa3s
To: <sip:maryhiggins@example.net>;tag=874726
Call-ID: 4gfjweroiu2aiqjnszd
CSeq: 3432 REFER
Contact: <sip:192.0.0.4>
Refer-To: <sip:32341293874@conf.example.com>
Referred-by: <sip:johndoe@example.com>
Content-Length: 0
```

As a result, Mary Higgins sets up her call to the conference server:

```
INVITE sip:32341293874@conf.example.com SIP/2.0
Via: SIP/2.0/UDP phone282.example.net:5060;branch=z9hG4bKnashds7e3
Max-Forwards: 70
From: <sip:maryhiggins@example.net>;tag=1234d9dfa3s
To: <sip:32341293874@conf.example.com>
Call-ID: 4gfjweroiu2aiqjnszd
CSeq: 54 INVITE
Contact: <sip:phone282.example.net>
Content-Length: ...
```

Other users will be migrated in the same way to the centralized conference.

Centralized Conferencing

The IETF has a new working group called the Centralized Conferencing Working Group, or XCON [9] that is extending the SIP conferencing work in the SIP and SIPPING working groups. The work is limited to conferences that have a focus—a centralized point of signaling, admission, and authentication. The media can still be full mesh, distributed, or even multicast, but the signaling must be centralized.

So far, the working group has produced a framework and data model document [10]. The actual protocols used to implement the framework are still under discussion. The working group has also published a floor control protocol known as Binary Floor Control Protocol (BFCP) [11].

Summary

This chapter has introduced a number of models of SIP conferencing such as endpoint mixing, SIP server and distributed media, dial-in conferences, ad hoc, centralized, and large scale multicast conferences. We have also shown how SIP can support changes in the nature of conference services. Protocols for advanced conferencing services are being developed in the IETF Centralized Conferencing (XCON) working group.

References

[1] "The Internet Multimedia Conferencing Architecture" by M. Handley, J. Crowcroft, and C. Borman. Internet Draft, IETF, July 2000.

[2] "Session Initiation Protocol (SIP)-H.323" by H. Schulzrinne and C. Agboh. Interworking Requirements, RFC 4123, July 2005.

[3] "A Framework for Conferencing with the Session Initiation Protocol" by J. Rosenberg. RFC 4353, IETF, February 2006.

[4] "High-Level Requirements for Tightly Coupled SIP Conferencing" by O. Levin and R. Even. IETF RFC 4245, November 2005.

[5] "Session Initiation Protocol Call Control - Conferencing for User Agents" by A. Johnston and O. Levin. IETF Internet Draft, June 2005, work in progress.

[6] "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)" by J. Rosenberg, H. Schulzrinne, and P. Kizivat RFC 3840, August 2004.

[7] "A Session Initiation Protocol (SIP) Event Package for Conference State" by J. Rosenberg, H. Schulzrinne, and O. Levin." IETF Internet Draft, July 2005, work in progress.

[8] "The Session Initiation Protocol (SIP) Refer Method, RFC 3515" by R. Sparks. April 2003.
[9] www.ietf.org/html.charters/xcon-charter.html.
[10] "A Framework and Data Model for Centralized Conferencing" by M. Barnes, C. Boulton, and O. Levin. IETF Internet Draft, October 2005, work in progress.
[11] "The Binary Floor Control Protocol (BFCP)" by G. Camarillo, J. Ott, and K. Drage. IETF Internet Draft, November 2005, work in progress.

CHAPTER 15

SIP Application Level Mobility

Mobile phone networks have shown even higher growth rates than the Internet, according to many reports from analysts. We explain this by a confluence of several factors, such as a satisfactory service for telephony (though not exactly a 100 percent replacement for QoS and Centrex features of wireline telephony), a real need for wireless services, and, most of all, mobile telephony has no equivalent in existing wireline telephony networks. Circuit-switched mobile telephone networks have surpassed wireline telephony, especially in new emergent economies where the wireline infrastructure is less developed.

All emerging mobile communications are IP-based.

NOTE **Mobile telephony has proven in the market that users will gladly trade in less than perfect QoS for the convenience of mobility. We will use this as reminder when discussing QoS in Chapter 18.**

Marketers and business planners did not take long to discover that the intersection of mobility and the Internet could be an even better combination of two already excellent ingredients. Several approaches are pursued to make this happen:

- *Add Internet-style services to present circuit-switched mobile networks, such as the Short Messaging Service (SMS) and web access*—An attempt at what was marketed as web access was the Wireless Access Protocol (WAP),

though WAP was quite different in reach, technology, and performance from the World Wide Web. The WAP Forum has consolidated into the Open Mobile Alliance (OMA) [1], and no longer exists as an independent organization. We don't count WAP as a successful technology, but rather as one of the many failed attempts in the telecom industry to provide data services without adequate bandwidth [2].

- *Design "next generation" mobile networks that can accommodate both voice and data*—One of the most prominent of such designs is the Third-Generation Partnership Project (3GPP) initiative. 3GPP is a complex project that has both a circuit switch legacy and also features the Internet Protocol Multimedia Subsystem (IMS) that uses SIP for call setup [3] and other SIP-enabled services, such as presence and IM. Much has been written about 3G wireless networks and about the IMS, so we will add our reservations here as well. There are many reasons for prudence or to be outright skeptical about IMS; see, for example, [4].
- *Commercial wireless networks for portable devices such as laptops and palm-sized computers*—Such networks extend in variety from IEEE 802.11 wireless LANs to IEEE 802.16 metropolitan networks. Surprising technology and market developments may make some of these the real winners in future wireless services.
- *Proposals for pure Internet-based mobile network designs, such as the proposal for Internet Technology Supporting Universal Mobile Operation (ITSUMO) [5]*—ITSUMO has not been implemented commercially at present, but we believe it to be one of the key blueprints for long-term next generation mobile communications based on SIP application-level mobility.

Various mobile networks have very different approaches to mobility. We will try to shed some light on the meaning of mobility, so as to provide a better understanding of the various approaches taken in the design of mobile networks.

Mobility in Different Protocol Layers

Mobility can be implemented in various layers of the protocol stack [6]:

- *Mobility at the link layer (L2), such as mobility in Wi-Fi networks or mobility in 2G and 3G mobile networks*—L2 mobility works only in the same network and using the same device. Only the point of attachment to the network can change.
- *Mobility at the IP network layer (L3)*—The protocol for mobile IP (MIP) is discussed in the following sections. Mobile IP also presumes mobility in the same network (the Internet) and using the same mobile device.

- *Mobility at the application layer (L5) provided by SIP*—As we will show, application-layer mobility can support the changing of L2 networks and the changing of devices, and significantly enlarges the dimensions of mobility.

It is possible to combine mobility at different layers, for example to combine SIP mobility with MIP or with L2 mobility to improve handover performance.

Dimensions of Mobility

Henning Schulzrinne from Columbia University in New York and his research group have introduced the concepts for SIP-based application-level mobility [7] [8]. These concepts are very innovative and disruptive by nature, and have already, in part, been proven in the market (such as the inherent mobility offered by commercial VoIP networks). Table 15.1 examines SIP-based application-level mobility. Some of the mobility modes discussed are, however, attributable to the intrinsic mobility on the Internet. These may apply equally well for other VoIP protocols, other than SIP, such as is the case for Skype (though Skype uses SIP gateways as well for PSTN termination).

Table 15.1 SIP-Based Application-Level Mobility

ROAMING USERS	LOGGING AWAY FROM OFFICE: HOME AND WHILE TRAVELING
Terminal mobility or *network level mobility*	The same endpoint moves between different attachment points to the same network. This is familiar from all 2G mobile phone services, though they are not IP-based.
Personal mobility	User is reachable under the same Address of Record on various networks and various devices (for example, at work, in the office, on a PC/laptop, office phone, office fax, mobile phone, and PDA). Personal mobility that includes legacy phones, fax, and 2G mobile services requires ENUM, as discussed in Chapter 4.
Service mobility	Users keep the same services when moving to another location. VoIP users can travel across the world and still call and be reached on the PSTN home phone number. This is a common feature found in almost all present VoIP services and is rather an Internet property.

(continued)

Table 15.1 *(continued)*

ROAMING USERS	LOGGING AWAY FROM OFFICE: HOME AND WHILE TRAVELING
	The price to pay is the more difficult determination of location for emergency calls when on travel or when connected with the home enterprise network using VPN. See Chapter 16, "Emergency and Preemption Communications services."
Session mobility	Users can move active sessions between terminals. For example a conference call participant on a mobile phone can move to a multimedia PC without dropping out from the conference and without losing any media content during the switchover. Streaming media sessions can be moved seamlessly from a mobile phone or PDA to a living room TV set.

Examples of SIP Application-Layer Mobility

The various types of mobility defined in Table 5.1 will be illustrated here with several high-level examples.

The basic scenario for roaming users is shown in Figure 15.1, where there is open Internet access, for example at professional meetings (IETF), conferences (Voice on the NET, or VON), some hotels, and in some public places.

The first step (a) consists of acquiring an IP address, gateway address, and DNS server address from the network using the Dynamic Host Configuration Protocol (DHCP). The mobile host will send out `DHCPDISCOVER` messages and one or more DHCP servers will reply with a `DHCPOFFER`, followed by `DHCPACK` to the mobile station (MS) [9]. The other DHCP messages are not shown for brevity, since this topic is out of scope here. Once the mobile host has network connectivity, the SIP UA can register with the outgoing SIP proxy. There are several methods to configure the SIP UA with the outgoing SIP proxy, one of them is specified in [10].

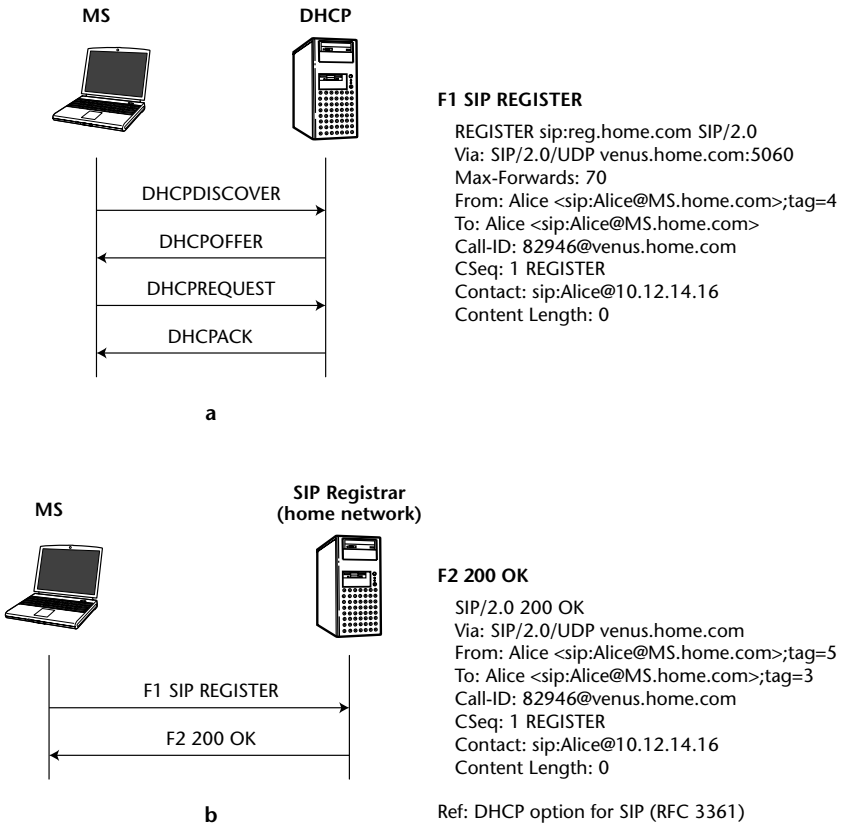
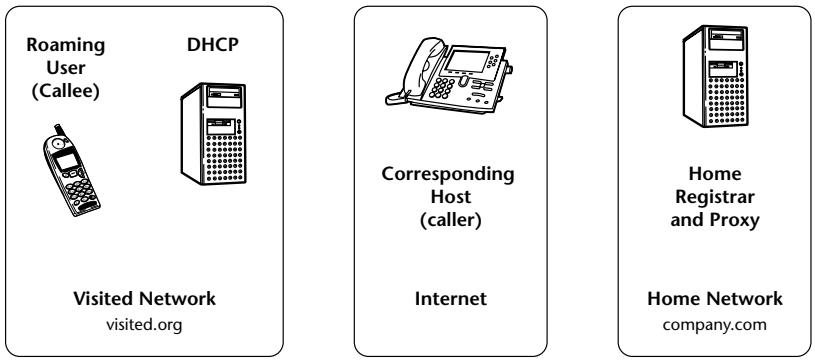


Figure 15.1 Basic scenario for roaming users: (a) DHCP network acquisition and (b) SIP registration

The terminal mobility scenario before starting a SIP session is shown in Figure 15.2. The roaming user belongs to a home network domain `company.com` and has a registered IP address in the home network domain. While traveling and roaming in a visited network, the mobile host will acquire an IP address using DHCP as previously shown, and register this new address with the home SIP registrar. The diagram shows what happens when a corresponding host elsewhere on the Internet tries to communicate with the mobile host. The initial `INVITE` (Message 1) will get a 302 response `moved temporarily` with the new IP address of the roaming user. In Message 4, the corresponding host will send another `INVITE` to the new IP address.



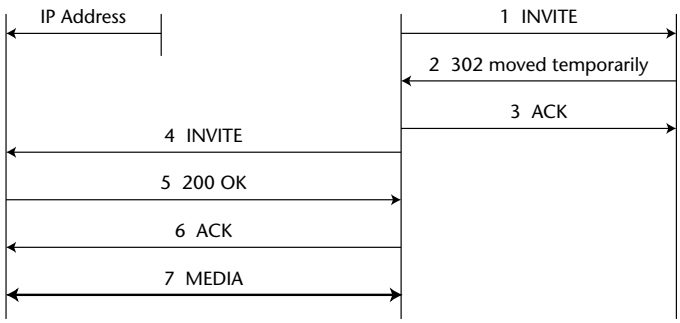


Figure 15.2 Terminal mobility: Precall scenario

Terminal mobility has to work also when the mobile host is moving between networks during a SIP session. This is shown in Figure 15.3 when the mobile user moves from Network 1 to Network 2.

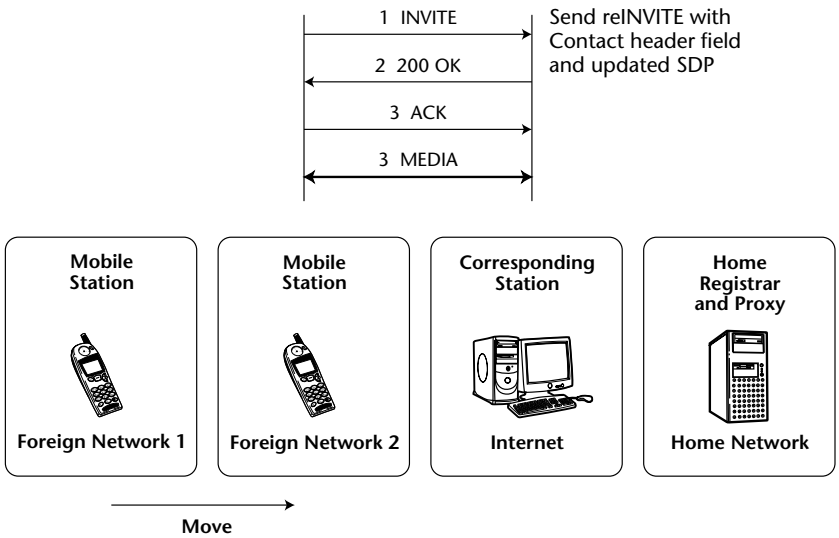


Figure 15.3 Terminal mobility: Mid-call scenario using SIP re-`INVITE`

When moving from one foreign network to another, the mobile host uses the Layer 2 link indications to acquire a new IP address and to communicate the new address to the home network. The mobile host will also send a re-`INVITE` [11] message to the corresponding host on the Internet so the SIP session can continue with the new network address.

As mentioned in Table 15.1, personal mobility enables the discovery and calling of someone on several devices, possibly at the same time. This is shown in Figure 15.4.

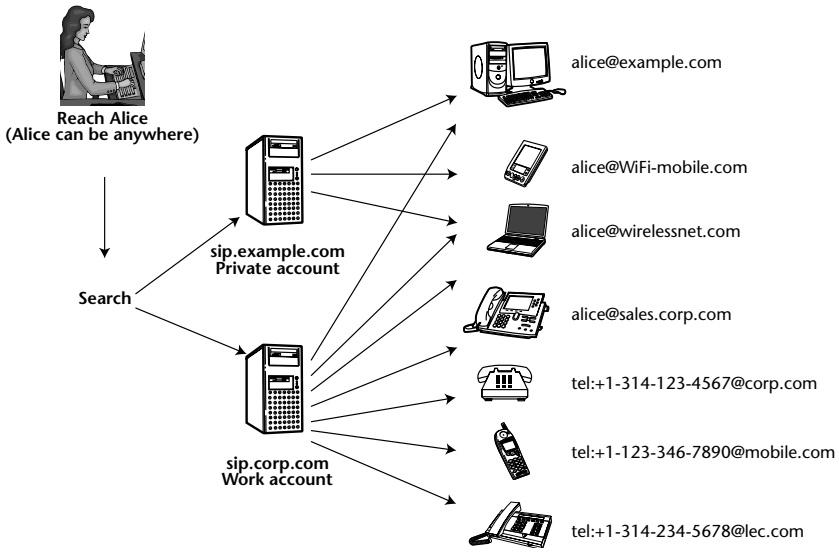


Figure 15.4 Personal mobility example

The example for personal mobility shown in Figure 15.4 also illustrates that users can have more than one SIP account for communications (for example, an account in the workplace and separate account for private use). The user can technically register the various communication devices with the account of their choice within the policy constraints of the account administrator. The enterprise administrator policy may allow, for example, registering only company-issued devices, while public service providers with restrictive policies may limit the user for one single device so that they can charge extra for each new user device. G2 mobile phones and PSTN phones and fax machines require static registrations, since they are not enabled for presence.

Session mobility to move an established session from one device to another can also be easily supported by the SIP `REFER` method [12]. The more interesting handling of session mobility is, however, when the handover from one device to another must be accomplished in a seamless way, so as not to cause any gap in the received media (missed sentences in conference, missed text in an IM session, or missed audio and images in an RTSP session). A scenario for seamless session mobility for streaming media is shown in Figure 15.5 [13].

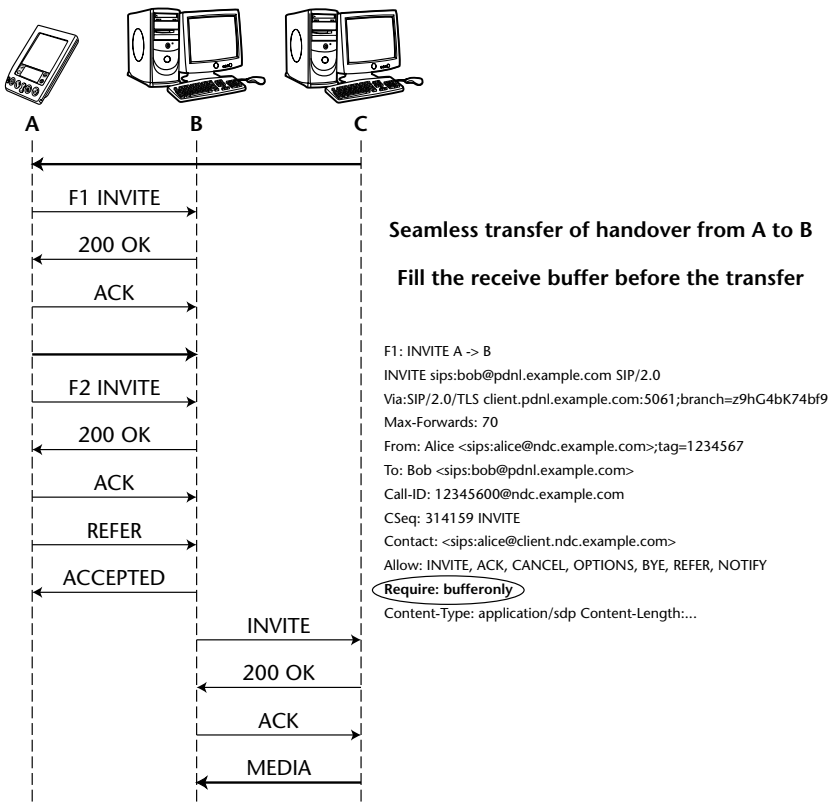


Figure 15.5 Session mobility with seamless handover

To avoid the gap for media, the media buffer must be filled before the complete handover to the new SIP UA, and this is accomplished with the header `Require: bufferonly`, as shown in the `INVITE` message to the new device where the session will be transferred to.

SIP Network-Based Fixed-Mobile Convergence

SIP-based application-level mobility is also an enabler for the convergence between mobile and fixed (wireline) networks, as shown in Figure 15.6.

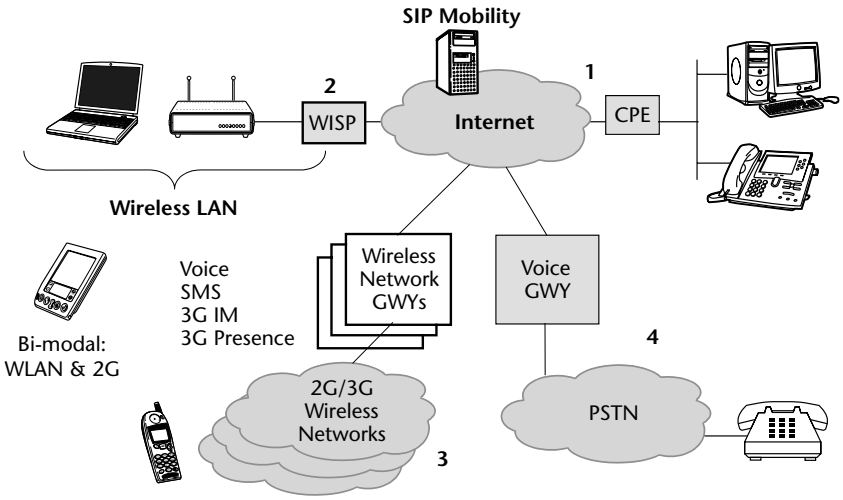


Figure 15.6 SIP network-based call control for fixed-mobile convergence

Though much as been written about fixed-mobile convergence, as shown in Figure 15.6, this is a natural application for SIP call control placed on the Internet. In this diagram, all communication devices on all possible networks are considered SIP endpoints:

1. PC/laptops and SIP phones on IP LANs
2. Mobile SIP devices on Wi-Fi (and WiMAX in the near future) networks
3. 2G and 3G devices as they appear on the Internet side of mobile-Internet gateways
4. PSTN and PBX phones as they appear on the Internet side of TDM-Internet gateways

An adequate SIP mobile device can also act as a virtual PBX phone in the enterprise with proper support for PBX-like voice services on the Internet-based SIP server and strong security precautions, such as those discussed in Chapter 9, "SIP Security."

The main challenge for network-based SIP mobility and fixed mobile integration is for the Internet-based SIP server shown in Figure 15.6 to get access to the gateway service providers to connect to 2G/3G mobile networks and to the PSTN gateways. Besides the technical issues of call control and security, bilateral business arrangements are also required, such as developed in the IETF SPEERMINT WG [14].

SIP Device-Based Fixed-Mobile Convergence

An increasing number of SIP endpoints have more than one network interface. This was initially the case with laptop computers but has now also spread to pocket computers, PDAs, and mobile phones. Following are the possible network interfaces:

- 802.3 wired LAN connection
- 802.11 a/b/g wireless LAN (Wi-Fi)
- 2G and 3G radio connectivity
- Bluetooth radio
- Emerging 802.16 wireless MAN (WIMAX)
- New standard radios such as Ultrawideband technology 802.15 (UWB) and proprietary systems such as xMAX radios

A SIP device with several network interfaces is also called a *multimodal* device. Multimodal devices could support the control for selecting the service and features to the end user, the owner of the device. This has tremendous implications for the business models of facility-based mobile services. We will not discuss here all the business and regulatory implications from the emergence of mobile SIP multimodal communication devices but will rather focus on the technical challenges in the section, on "Multimodal Mobile Device Technology and Issues," later in this chapter.

SIP Application-Layer Mobility and Mobile IP

Mobility on the Internet can be provided by Mobile IP (MIP), see [15]. SIP and Mobile IP (MIP) differ however in a fundamental way:

- Application-layer mobility based on SIP deals with the changing IP address to keep the applications working. SIP application-layer mobility works only for the applications for which it has been designed; in this case, for real-time communications only.
- Mobile IP presents to the applications the same IP address, though the actual IP address in the network may have changed. Mobile IP works, therefore, for all applications, including file transfer, e-mail, and the web, for example.

As can be seen in Figure 15.7, the triangle routing may introduce undesirable extra delays for real time media like voice.

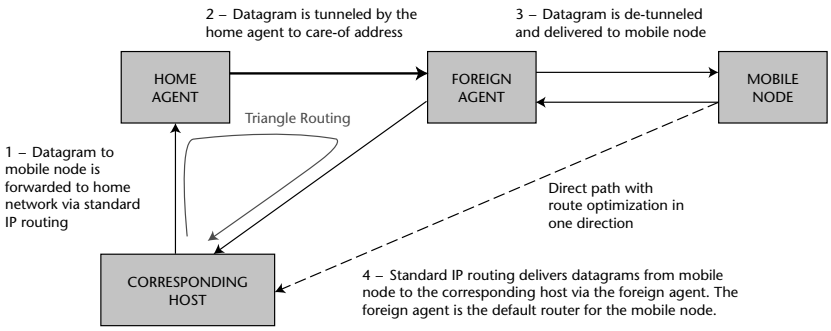


Figure 15.7 Mobile IP operation

MOBILE IP OPERATION

Mobile IP maintains the IP address of the mobile host, when away from home. It works in the following way. The mobile host has a permanent IP address assigned in the home network. A router in the home network (called the *home agent*) will route IP packets to and from the host using its IP home address, while the host is still in the home network. When away from home, the mobile host will register with a local router (called the *foreign agent*) and will receive a temporary care-of address. The foreign agent will do the following for the mobile host:

1. **The care-of address is communicated to the home agent.**
2. **An IP tunnel is used to forward IP packets (datagrams) between to the home agent and the mobile host.**

Mobile IP works with both UDP and TCP transport and keeps the applications unaware of mobility.

The flow of IP packets in both directions, between the mobile host and a corresponding host on the Internet, is shown in Figure 15.7. Note that communications between the mobile host and the corresponding host always go via the home network, although this route may not be optimal.

If, for example, the two hosts are quite near geographically, but the home network is far away, the nonoptimal routing becomes a problem. Route optimization for mobile IP is described in [16] and provides extensions to Mobile IP for the corresponding host to cache the care-of address of the mobile host and to bypass the home network, so that packets use normal IP routes to the mobile host. Packets from the mobile host to the corresponding host will, however, still take the longer route via the home network.

The nonoptimal routing, at least in one direction, may introduce an undesirable delay for interactive communications. The encapsulation in the IP tunnel shown in Figure 15.7 also adds to the overhead for RTP/UDP/IP packets.

Mobile IP requires two addresses for the mobile host and has problems with NAT transversal. NAT transversal problems are, however, not unique to Mobile IP. SIP also needs firewall transversal support, as we show in Chapter 10, "NAT and Firewall Traversal."

SIP mobility does not have the drawback of long media paths and is, therefore, better suited for real-time communications than Mobile IP.

We are not aware yet of any major deployments by Mobile IP by service providers. This may probably be because the first service providers to implement Mobile IP have little incentive to do so, since they would only serve users belonging to other networks that just happen to visit their own network. The complex payment settlement technologies between providers have also not been deployed for MIP.

Combining SIP mobility with MIP can, however, give excellent results in some application scenarios by reducing both the delay and the handover time [17], keeping, for example, the handover time at roughly 100 ms in 802.11 wireless networks independent of the delay between the mobile node and the corresponding node on the Internet.

Using MIP alone would have increased the handover time in a linear fashion with the delay between the two nodes.

Multimodal Mobile Device Technology and Issues

As mentioned, multimodal mobile communication devices have the potential to produce significant new business models, disruptions of existing mobile services, and ensuing regulatory changes required for the benefit of users and in the interest of public policy. Also as mentioned, quite a number of mobile devices, besides the laptop computer have more than one network interface. Figure 15.8 shows an example of a SIP UA for dual-mode PDAs and pocket computers.

As more multimodal mobile devices are emerging in the market, SIP-based mobile communications are becoming more widespread. The major technical, interoperability, and standards approaches are still under development. We will present in the following sections some of these technologies and issues.

Figure 15.8 SIP user agent for GSM- and Wi-Fi-enabled PDA (courtesy Cicero Networks). (a) HP PDA, (b) Qtek mobile phone

Network Control versus User Control of Mobility

Network control of mobility as shown in Figure 15.6 assumes the existence of one single network-based service and this does not apply to most users. Mobile IP makes no assumptions about network ownership, and this is not satisfactory either.

The emergence of multimodal mobile devices (as shown in Figure 15.8) places the control of mobility in the hands of the user who can now chose between several types of networks, not all necessarily owned by the same organization or service provider. We will show in the following discussion how the handover between different network types (referred to here as "media") can be accomplished, and what the issues are when selecting networks from different owners.

IEEE 802.21 Media-Independent Handover (MIH)

We will use in the this section the term "media" to refer to the Layer 2 of the IP protocol stack, such as the various IEEE 802 wired and wireless standards, wireless 3G networks, and others.

A multimodal mobile device can switch from one "media" (Layer 2 network) to another and for real-time communications, as well as for streaming multimedia. The IEEE 802.21 [18] is one of the standards organizations working to define the issues as they relate to the various L2 link layers. The other standards body working at the IP network layer (Layer 3) and the application layer (Layer 5) is the IETF. The IEEE and the IETF are consistently trying to accommodate various commercial interests, such as the 3GPP and 3GPP2 organizations for 3G wireless networks.

The reference model for the IEEE 802.21 media-independent handover (MIH) is shown in Figure 15.9, reflecting the standard proposal as of summer 2005 and it may take several years for the work standards to be completed.

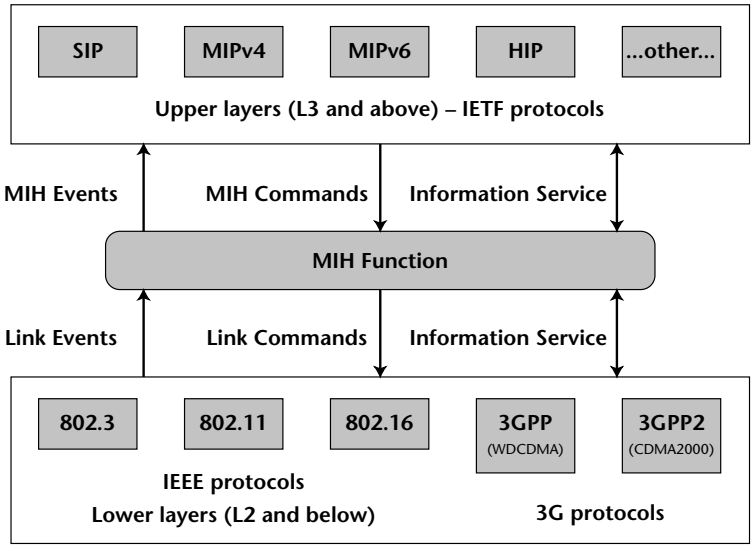


Figure 15.9 The IEEE 802.21 architecture for media-independent handover

The elements in the IEEE 802.21 architecture are:

- The various L2 media types: 802.3 (Ethernet), 802.11 (Wi-Fi), 802.16 (WiMAX), and the 3G mobile service layers 3GPP and 3GPP2.
- The service layers: SIP, MIPv4 (for IPv4), MIPv6 (for IPv6), the Host Identity Protocol (HIP), and others.
- Between the two (L2 and L3-L5, respectively), three services have been defined:
  1. Media Independent Event Service to report L2 events to the upper layers. Local L2 events happen inside the device, while remote L2 events happen in the network. Common events are Link UP, Link Down, Link Parameters Change, Link Going Down, L2 Handover Imminent, and so on.
  2. Media Independent Command Service to control the lower layers with commands to poll, scan, configure, and switch at L2 and below.
  3. The Media Independent Information Service provides rich data to both upper and lower layers for channel information, MAC address, security information, and everything else required to make correct handover decisions. The data formats can be XML or ASN1.

Joint work on a test bed reported by Telcordia, Toshiba, and Columbia University for 802.21 MIH-assisted SIP mobility has been reported in [19]. Figure 15.10 shows some of the results for the audio output at the mobile node using 802.11 networks.

Non-802.21 assisted SIP-based mobility

802.21 assisted SIP-based mobility – Optimized hand-off

Figure 15.10 Audio output at the mobile node with and without 802.21-assisted SIP mobility

The results in Figure 15.10 show a very smooth handover performance from combining SIP with IEEE 802.21 media-independent handover. Without MIH, the handover time for 802.11 test networks is about 4 ms. 3G mobile networks may introduce up to 15 seconds handover delay without MIH because of the lengthy authentication and signaling procedures in such networks. As we will see in the next section, lengthy authentication and signaling can happen in commercial 802 networks, as well to ensure payment for service, though probably not as lengthy as in 3G networks.

Network Selection Issues

Most present business models require secure (but complex) authentication procedures to ensure the payment for Internet access service in wireless networks. This is usually the cause for quite complex network selection issues. The reasons for the network selection issues include the following [20]:

- There may often be several wireless networks in the same location but with different characteristics, different payment methods, and different credentials required.
- Several service providers share the same wireless networks (for example, a hot spot in an airport is accessible via subscription from various wireless providers or aggregators).
- The path for the IP packets depends on the service provider connected to the wireless access network.
- Complex roaming relationships may result in different cost structures to the user.
- The user has to make network selection decisions and provide different credentials for different providers.
- The user may have different credentials, such as one for company use and another for personal use.

Figure 15.11 shows one of the possible scenarios for network selection.

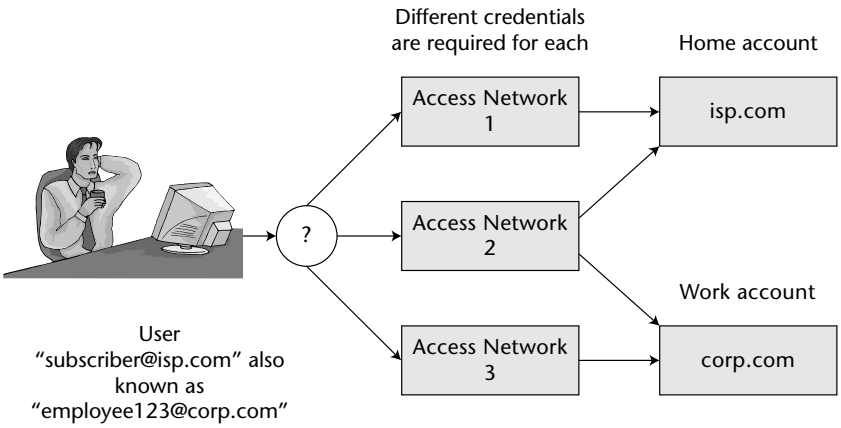


Figure 15.11 Network selection scenario: two credentials and three links

Besides the complexity caused by L2 and L5 signaling, authentication, and payments, there is also additional complexity attributable to several standards bodies (such as the IETF, IEEE, and others), as well as commercial consortia (such as 3GPP and 3GPP2), are all involved in mobility services because of the huge market potential. The penalty to users will be the very long time until mobile services across networks will be invisible when changing networks.

The authors believe for these reasons that users will prefer locations where they enjoy perceived "free" network access, such as is already the case in some hotels, airline lounges, public places, or in some municipalities.

Summary

This chapter has discussed various aspects of mobility, at all levels of the protocol stack. We have shown some of the key advantages of application layer mobility using SIP.

References

[1] See the web site for OMA at `www.openmobilealliance.org`.
[2] "W* Effect Considered Harmful" by R. Khare. IEEE Internet Computing, Vol. 3, No. 4, July–August 1999, pp. 89–92.
[3] The 3GPP web site is `www.3gpp.org`.

[4] "IMS: A Critique of the Grand Plan" by J. Waclawski. *Business Communications Review*, October 2005, pp. 54–58. A summary of this article is also available online at www.bcr.com/bcrmag/2005/10/p54.php.

[5] "Supporting Service Mobility with SIP" by F. Vakil et. al. Internet Draft, archive, IETF, December 2001.

[6] *IP for 3G: Networking Technologies for Mobile Communications* by D. Wisely et al. John Wiley & Sons, Ltd., 2002.

[7] "Application-Layer Mobility Using SIP" by H. Schulzrinne and E. Wedlund. *Mobile Computing and Communications Review*, Vol. 1, No. 2, 2001. www.cs.columbia.edu/~hgs/papers/Schu0007_Application.pdf.

[8] "SIP for Mobility" by H. Schulzrinne. Columbia University, 2001. www.cs.columbia.edu/~hgs/papers/2001/sip2001_mobility.pdf.

[9] "Dynamic Host Configuration Protocol" by R. Droms. RFC 2131, IETF, March 1997.

[10]"DHCP Option for SIP Servers" by H. Schulzrinne. RFC 3361, IETF, August 2002.

[11] "A Call Control and Multi-party usage framework for SIP" by R. Mahy et al. Internet Draft. IETF, October 2005, work in progress.

[12] "The SIP Refer Method" by R. Sparks. RFC 3515, IETF, April 2003.

[13] "Indicating Media Handling Features in SIP for Seamless Session Mobility" by Xu Mingqiang. Internet Draft, IETF, March 2005, work in progress.

[14] SPEERMINT WG in the IETF. www.ietf.org/html.charters/wg-dir.html.

[15] "IP Mobility Support for IPv4" by C. Perkins. RFC 3344, IETF, August 2002.

[16] "Route Optimization in Mobile IP" by C. Perkins and D. Johnson. Internet Draft, IETF, February 1999. http://mosquitonet.stanford.edu/mip/draft-ietf-mobileip-optim-08.txt.

[17] "Performance Evaluation of Two Layered Mobility Management using Mobile IP and Session Initiation Protocol" by Jin-Woo Jung et al. Globecom 2003 conference. http://w3.antd.nist.gov/pubs/sip-mip-jwjung-globecom2003.pdf.

[18] See the home page for IEEE 802.21 at www.ieee802.org/21/index.html.

[19] "Seamless Handover across Heterogeneous Networks - An IEEE 802.21 Centric Approach" by A. Dutta et al. Columbia University, April 2005. See also http://www1.cs.columbia.edu/~dutta/research/mpa-mobiquitous.pdf.

[20] "Network Discovery and Selection Problem" by J. Arkko and B. Aboba. Internet Draft, IETF, October 2005, work in progress.

CHAPTER

16

Emergency and Preemption Communication Services

Users and government regulators expect VoIP service providers to support emergency calls as good or better than the PSTN. Emergency calls need to convey the exact location of the caller so that assistance can be dispatched to where it is required. Note that this is in contrast to the initial deployments of mobile telephony, where even as of this writing, mobile phone services do not always support emergency calling with location information.

Making emergency calls over the Internet is not a trivial task for a variety of reasons, all Internet-related, but also because of other administrative issues, some of which will be mentioned here.

Internet-related difficulties stem from the very fact that users can be located anywhere, and the IP address bears no relation to their location. There are various mobility scenarios on the Internet (see Chapter 15); the most obvious is users of VPN tunneling. An employee of a company located in Ireland may be traveling in Australia and connect from a Wi-Fi hot spot to the home network in Dublin using VPN. The SIP UA will have an internal enterpise IP address obtained from the VPN system. If an emergency call, say, for medical purposes, is placed by the user, the call will appear on the Internet as coming from the home network in Dublin, and sending an ambulance to the office in Dublin would not be helpful.

Emergency calls are routed to a Public Safety Access Point (PSAP), where an agent takes the call and determines where to route it further to provide the required assistance (such as police, fire, ambulance, mountain rescue, and so

on). Determining the right PSAP is not trivial either, because the service boundaries for the various response centers for police, fire, ambulance, and so on may differ for various local administrative reasons. They vary by local jurisdiction, and they also vary from country to country. It is estimated there are about 6,000 PSAPs in North America alone and possibly three times as many in the rest of world [1]. Accurate dispatch of assistance to the emergency caller requires mapping the civic or geographic address to the exact street address where the caller will be found. This mapping is done using the Master Street Address Guide (MSAG). The organization of PSAPs and responders (such as fire, police, ambulance, and so on) is done differently in various parts of the world.

Requirements

The IETF working group on Emergency Context Resolution Using Internet Technology (ECRIT) has oultined the requirements for emergency calling [2], and we will provide here a short overview of these requirements.

- The caller *may not* have a VoIP service provider, since a user, a residence, or a small business may have their own domain name and SIP endpoints for their domain (such as SIP UAs or their own SIP proxy). Larger enterprises or university campuses may also provide VoIP for their users. A VoIP service provider *must,* therefore, not be assumed.
- PSAP information *must* be available even when no VoIP service provider is used.
- Internet emergency protocols and data formats *must* support consistent international deployment.
- Deployment of emergency calling over the Internet *must not* depend on any central authority.
- Internet multimedia, including IM and Text over IP (ToIP), *must* be supported for better assistance. It should be possible to convey telemetry data, such as that from crash sensors or medical vital signs from patients under remote supervision.
- Emergency calls that can be routed within a region *must not* be routed over long paths that are outside the region, and *must not* have any out-of-region dependencies, such as remote SIP proxies, or gateways to the PSTN.
- The ECRIT mapping protocol *must* return a URI that can be understood by any "legacy" SIP UA.
- Callback information *must* be provided to enable emergency assistance to call back the user.

- Credible emergency call testing mechanisms *must* be provided.
- It should be possible to invoke relay services for the hearing or speech disabled (see Chapter 17, "Accessibility for the Disabled").

Location Information

As mentioned, location information is a critical ingredient for any emergency call. At the same time, location information is very sensitive and must be carefully protected. The protection of location information has been defined in the IETF working group on Geographic Location and Privacy (geopriv) [3].

Types of Location Information

There are four types of location information as shown in Table 16.1.

Sources of Location Information

SIP UA can derive location information (LI) from a number of sources, and should indicate the source of LI that may be used for call-routing decisions. There are a variety of sources available to the SIP UA for location information, such as the following:

- Manually entered LI
- Wire databases for the Ethernet switch, line identification dabase for DSL, or cable service provider
- GPS (available only when there is a clear view of a large part of the sky)
- Third party:
  - Wireless triangulation in mobile networks
  - Location radio beacons announcing the location

Unless the SIP UA has direct access to LI, it must obtain the Location Object (LO) using DHCP.

DNS-Based Location Information

As mentioned, determining the right PSAP to route an emergency call to is difficult for several reasons.

These difficulties can be circumvented with central databases that map PSAPs to the civic address and to the data in the MSAG. Such a centralized approach is used in the PSTN and has advantages and disadvantages common to all centralized approaches. The centralized approach is described later in this chapter in the section, "Using the PSTN for VoIP Emergency Calls."

Table 16.1 Types of Location Information

TYPE	EXAMPLE
Civic	Country, state, city, street address and floor, appartment, or cube.
Postal	Similar to the civic address but the post office box (P.O.B.) or building may not reveal the actual location.
Geospacial	The longitude, latitude, and altitude information.
Cell Tower/Sector	Cell tower ID and antenna used by the mobile device. The mobile country code and mobile network code may also be provided.

An Internet-centric approach using the DNS and its design philosophy of delegation has been outlined in [1]. The emergency call information in the DNS is based on the Dynamic Discovery and Delegation System using the DNS as specified in RFC 3761. We will illustrate here the main characteristics to explain how the DNS-based Internet emergency calling information works:

- The top-level domain is defined as `sos.arpa`, where `sos` stands for emergency calls.
- The country code is added to the left and the administration is delegated to the country domain, such `us.sos.arpa`.
- Province or state domains, counties, cities, and street addresses are added further to the left and their administration is delegated further down so as to create a delegation tree, similar to those described in Chapter 4, "DNS and ENUM."

The administrator of any entity can contract the maintenance of the DNS entries with a DNS registrar of their choice. This is similar to the DNS and ENUM registrars discussed in Chapter 4.

Table 16.2 shows an example of a delegation tree for the root domain `sos.arpa`.

Table 16.2 Example for Delegation in the SOS DNS Tree for Location Information

DELEGATION TREE BRANCH	ADDRESS
Root for SOS	`sos.arpa`
Country	`us.sos.arpa`
State/Province	`pa.us.sos.arpa`
County	`allegheny.pa.us.sos.arpa`
City	`pittsburg.allegheny.pa.us.sos.arpa`

Table 16.2 *(continued)*

DELEGATION TREE BRANCH	ADDRESS
Street	`main.pittsburg.allegheny.pa.us.sos.arpa`
Street Number	`123.main.pittsburg.allegheny.pa.us.sos.arpa`
Cube No. & Floor	`235-5.123.main. pittsburg..allegheny.pa.us .sos.arpa`

Note how the delegation principle has replaced the central MSAG database. The accuracy of the information is because of the local entries in the DNS servers at each of the respective leaves in the DNS tree for SOS. The lowest-level entry is made by the network administrator for the building on 123 Main Street who has the best detailed knowldge down to the cube level on every floor.

The DNS Naming Authority Pointers Records (NAPTR) for these domain and subdomains may have pointers to several types of XML documents:

- Polygon describing the geographic boundaries of the domain
- List of subdomains to facilitate searching
- Building-related information (such as contact person)

The DNS NAPTR records for SOS can support automatic routing to the correct PSAP without human intervention and without any depencency on central databases.

Internet-Based Emergency Calling

PSTN emergency calls have evolved over more than 100 years and work quite well, except there are many numbers to remember in an emergency, depending on the location. The emergncy number to call in the United States is 911, while in other countries, differnet numbers are used. The following emergency telephone numbers are used in France [4]:

- *Emergency*—112 (used throughout most of Western Europe)
- *Ambulance*—15
- *Fire*—18
- *Police*—17

The general emergency phone number in the United Kingdom is 999 with various numbers for specific emergencies or crimes to report. Other countries in other regions of the world have widely different emergency phone numbers—about 60 service numbers across the world. To configure mobile phones for

emergency numbers, various mechanisms exist. GSM mobile phones use the SIM card inside the device, while 3G mobile phones rely on the 3GPP-specific network-based solutions. We expect emergency calling on the Internet to use URIs that have global significance and are also simple to remember in an emergency.

Identifying an Internet Emergency Call: The SOS URI

The emergency URI defined for the Internet is `sos` [5], and it uses the feature tag for the caller Feature Set Preferences based on RFC 3841. The proposed values for "`sos`" for anumber of emergency services as shown here below. The various services are invoked using a new media feature tag called "`sip.emergency-service`". The feature tag indicates the type of service requested as shown in Table 16.3.

The outbound SIP proxy will check whether an emergency phone number has been dialed and will translate it into a `sos` URI for call routing. The `sos` URI can also be used directly in the SIP `Request-URI` by the mobile Internet device.

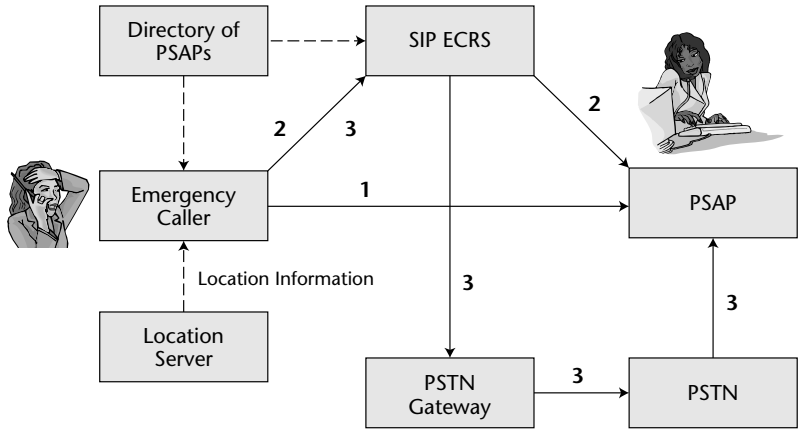
Internet Emergency Call Routing

Emergency call routing over the internet is illustrated in Figure 16.1, where the following can be highlighted:

- The location information may be made available to the device itself using one of the methods discussed in the preceding sections, for example being provided by the network infrastructure and loaded by DHCP at same time with acquiring an IP address.
- The user could also consult a directory to find the the right PSAP URI.
- The emergency call could also be routed by the VoIP service provider, if one is used, to the SIP proxy for emergency call routing support (ECRS).

Table 16.3 Internet Emergency Services for SIP

SERVICE	TAG
General emergency service	`sos`
Fire brigade	`sos.fire`
Marine guard	`sos.marine`
Mountain rescue	`sos.mountain`
Police (law enforcement)	`sos.police`
Ambulance, emergency medical service	`sos.rescue`
Testing, not a real emergency call	`sos.test`



Routes to the PSAP:
1. Direct
2. Internet Emergency Call Routing Support (ECRS)
3. PSTN gateway and the PSTN

Figure 16.1 Emergency call routing on the Internet

Once the location and the PSAP URI are available, three call routes are possible:

1. The emergency caller routes the call directly to the PSAP. This is the most reliable alternative, since there are no other dependencies of possibly remotely located SIP proxies owned by a VoIP service provider.
2. The emergency call is routed to the PSAP via the ECRS SIP proxy of the VoIP service proivder.
3. There is no Internet-connected PSAP available for the location of the user. The ECRS will route the emergency call as a last resort to the PSTN gateway to connect to the PSAP.

Routing a VoIP emergency call to the PSTN implies additional complexity, some of which is presented later in this chapter in the section, "Using the PSTN for VoIP Emergency Calls."

Security for Emergency Call Services

Attacks on the PSTN to exploit emergency services have been known to happen, mainly to steal service. Security is a much larger issue when routing emergency calls on the Internet. The possible threats have been analyzed in [6] and the requirements for secure SIP call routing are also proposed.

Possible motivations for attackers could be the following:

- Prevent the caller from receiving aid
- Gain infomation about the caller
- Bypass normal authentication to gain access to free services

The reference architecture offers many options for attacks that go beyond the scope of this chapter and, as a consequence, ECRIT security requirements are under development as of this writing.

Using the PSTN for VoIP Emergency Calls

PSTN-based emergency calling services need, in principle, only to be connected to the Internet to complete emergency calls using IP-IP from end to end. Existing emergency call services have, however, a different logical structure and legacy databases that have to be taken into account. This brings along the associated complexity that will be illustrated here.

We will use as the base reference the Interim VoIP Architecture for Enhanced 911 Services [7], published by National Emergency Number Association (NENA). This architecture leverages the best insight from the Internet-based architecture shown in Figure 16.1, while taking into account the constraints and resources of the PSTN-based emergency call services in most of North America. A simplified diagram of VoIP emergency calling using the PSTN is shown in Figure 16.2. This architecture does not use either the `sos` URI, nor the DNS-based location information for call routing discussed in the preceding sections, since they are at present not yet implemented.

The SIP UA or some other SIP endpoint (such as an IP-enabled PBX) can obtain the location object (LO) document from the location information (LI) server in the network. The LI originates from a VoIP positioning center (VPC) using an Emergency Routing database that maps civic addresses to routing numbers on the PSTN. To avoid errors in the central database for routing numbers, the operator of the validation database has access to the MSAG that is also used by the PSTN-based PSAP. The telephony call routing numbers from the VPC are used by the emergency SIP proxy to route the call to the PSTN gateway. The telephony routing number will be used by the emergency switch for 911 calls to route the call to the correct PSAP. The agent taking the call in the PSAP can consult the Automatic Line Identification (ALI) database to obtain the callback number for the caller.

Detailed definitions of the elements in the transition VoIP-PSTN architecture, functions, and call flows are provided in the NENA design that we have outlined here.

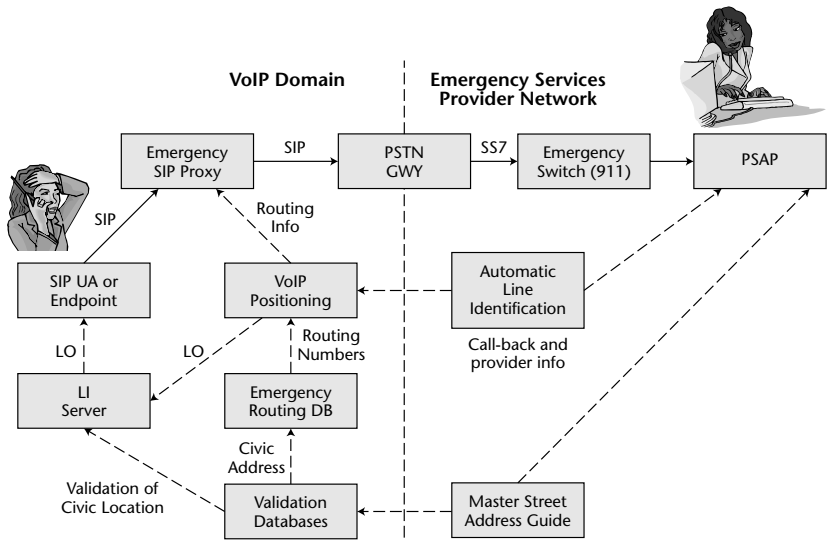


Figure 16.2 Emergency VoIP calling using the PSTN emergency infrastructure

Emergency Communication Services

Emergency calls, as described in the previous sections, are made to assist individual users under normal circumstances (that is, when no general emergencies occur, such as natural or man-made disasters).

During an emergency, the network resources may not be able to handle the surge of communications, and preference must be given to rescue, government, and military callers. Limited network resources could be the ports on PSTN gateways, and bandwidth limitations on access links or on bandwidth constrained global satellite links, for example. The Emergency Telecommunications Service (ETS) has been implemented in the PSTN and an excellent overview of implementing ETS in IP telephony is provided in RFC 4190 [8]. The work on emergency services using IP and the Internet has focused on a few critical topics, most notably:

- Requirements for resource priority mechanisms for SIP [9]
- The solutions for communications resource priority for SIP [10]
- SIP reason header for preemption events [11]
- Preemption of network resources used by SIP in government and military networks, and in single administrative domains in general

The IETF working group on Internet Emergency Preparedness (ieprep) [12] is working on various Internet standards for emergency preparedness.

NOTE **Internet Emergency Peparedness is complex topic given several facts of Internet traffic, such as:**

- **Wide disparity in bandwidth between the Internet core and access links to the Internet.**
- **Reports indicate the majority of the Internet traffic in 2004–2005 was due to peer-to-peer (P2P) applications of proprietary nature, some of them purposefully designed not to be easily blocked.**
- **Multimedia traffic such as streaming video (IP TV) is expected to produce a further surge in Internet traffic as well as in private domains.**
- **Voice traffic is only a negligible fraction of Internet traffic and as a consequence, applying preemption measures only for voice make sense only in single administrative domains where traffic from all applications can be strictly controlled.**
- **Last, but not least, should voice communications deteriorate due to traffic peaks in an emergency, users can fall back on IM and still get through for vital communications.**

For the reasons outlined here, preemption mechanisms for SIP make sense where SIP is controlling all or most of the IP traffic:

- For gateways to the PSTN
- On access links in single domains where the traffic can be controlled, especially on satellite links to remote or isolated locations on the globe

Emergency Call Preemption Using SIP

In a mixed PSTN-Internet environment, either the circuits on the PSTN or the VoIP-PSTN gateways may be the most constrained resources. Also, ETS is already implemented on the PSTN side, as mentioned. A scenario for this mixed environment is shown in Figure 16.3, where User 1 on the PSTN (left side) needs to send a preemption message to User 2 on the Internet (right side).

User 2 may have to drop any other sessions in progress not shown here and take the call. The preemption message on the PSTN side will also free TDM circuits using the ITU-T specification Q.850, and will also free a gateway port in case this is necessary.

The rich capabilities of SIP can support informing the preempted user of the reason, thus avoiding confusion and further unwanted callback attempts. This is illustrated in Figure 16.4.

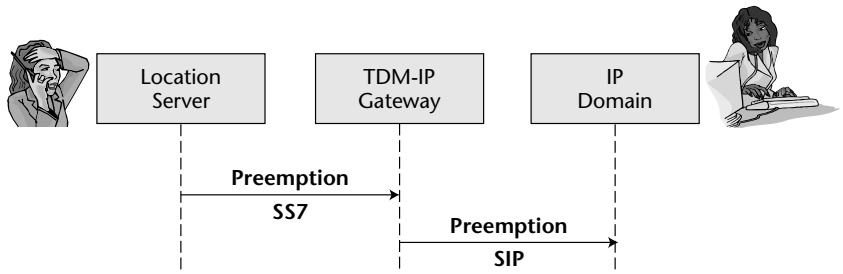


Figure 16.3 TDM-IP preemption event

The scenario in Figure 16.4 starts with UA1 initiating a call to UA2 using an `INVITE` message with resource priority `R-P:3` at level 3. In the success scenario shown here, an RTP media stream is established.

During the conversation, an emergency caller, UA3, initiates a new `INVITE`, but with priority level 2 (`R-P:2`), which is higher than for the existing call.

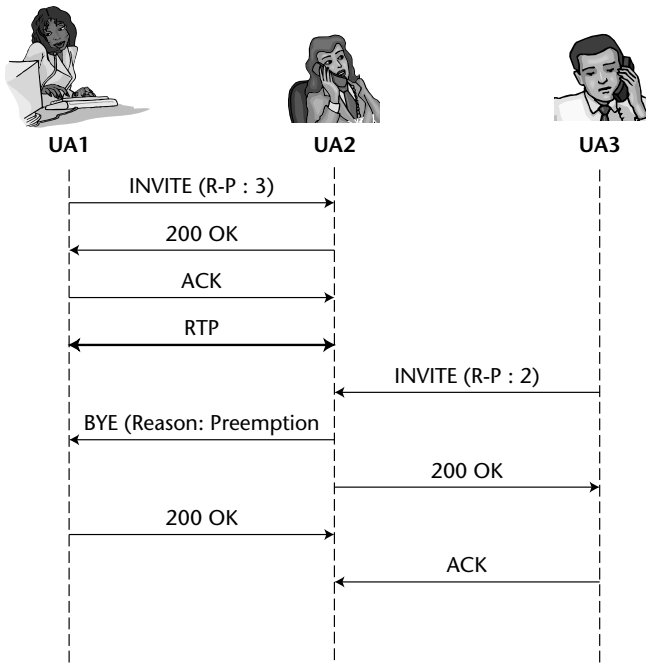


Figure 16.4 SIP access preemption with reason header

As a consequence, UA2 will send a BYE message to UA1 that contains the reason header:

```
Reason: Preemption; cause 1, text="UA Preemption"
```

The first call is terminated, and UA2 can send a 200 OK message to UA3 to take the call.

Several reason codes have been proposed for the SIP reason header for preemption as shown in Table 16.4.

Linking SIP Preemption to IP Network and Link Layer Preemption

As mentioned, in some special private networks using satellite links or on frugal access links, preemption may also be required at the IP layers. Proposals have been advanced to use various technologies at the IP layer such as Differentiated Services, RSVP, and MPLS [13]. The elements required as preconditions for SIP [14] can then link the preemption technology at the network layer, such as RSVP, with the preemption technology at the SIP application layer.

A comprehensive framework that describes in detail the options for the IP layer and also for various link layers (such as 802 networks, mobile networks, and cable links) is provided in [15].

Preemption at the IP network layer has many of the same limitations and complexity as QoS for IP that is discussed in Chapter 18, "Quality of Service for Real-Time Internet Communications."

Table 16.4 SIP Reason Codes for Preemption

CAUSE	DEFAULT TEXT	DESCRIPTION
1	UA Preemption	The session has been preempted by a UA.
2	Reserved Resources Preempted	The session has been preempted by an IP occurence, such as RSVP preemption and not by a link error.
3	Generic Preemption	Designed to be used on the final leg to the preempted UA to generalize the event.
4	Non-IP Preemption	The preemption has occured in the non-IP infrasructure, and the reason code is inserted by the gateway.

Summary

Internet communications based on SIP can provide rich and reliable emergency call services for end users, such as "911" or "112," often with better availability because of the resilience of the Internet compared to the PSTN.

The emergency call services over the Internet require the emergency Public Safety Access Points (PSAP) to be connected to the Internet, and also be IP enabled. The emergency calling services based on IP can be most effectively implemented and automated by using the DNS and delegating the correct registrations down the DNS delegation tree. The Interim "911" emergency calling architecture for VoIP makes use of the PSTN and its legacy database infrastructure in large part as is.

Emergency communications in case of disasters can use standard SIP methods and interwork well with the Emergency Telecommunications Services (ETS) existing on the PSTN, as well as in a pure IP environment. SIP signaling can also interwork with the network infrastructure for coordination of network resources, such as PSTN gateway ports or on bandwidth-challenged IP links.

References

[1] "Emergency Call Information in the Domain System" by B. Rosen. Internet Draft, IETF, October 2005.

[2] "Requirements for Emergency Context Resolution with Internet Technologies" H. Schulzrinne and R. Marshall. Internet Draft, IETF, September 2005, work in progress.

[3] IETF GEOPRIV WG: `http://ietf.org/html.charters/geopriv-charter.html`.

[4] "Emergency telephone number" in Wikipedia: `http://en.wikipedia.org/wiki/Emergency_telephone_number`.

[5] "Emergency Services URI for SIP" by H. Schulzrinne. Internet Draft, IETF, October 2005, work in progress.

[6] "Security Threats and Requirements for Emergency Calling" by H. Tschofenig et al. Internet Draft, IETF, December 2005.

[7] "NENA Interim VoIP Architecture for Enhanced 9-1-1 Services (i2)" prepared by the National Emergency Number Association (NENA) VoIP-Packet Technical Committee, December 6, 2005.

[8] "Framework for Supporting Emergency Telecommunications Service (ETS) in IP Telephony" by K. Carlberg et al. RFC 4190, IETF, Nov. 2005.

[9] "Requirements for Resource Priority Mechanisms for SIP" by H. Schulzrinne. RFC 3487, IETF, February 2005.

[10] "Communications Resource Priority for SIP" by H. Schulzrinne and J. Polk. Internet Draft, IETF, July 2005.
[11] "Extending the SIP Reason Header for Preemption Events" by J. Polk. Internet Draft, IETF, September 2005.
[12] IETF working group on Internet Emergency Preparedness: `http://ietf.org/html.charters/ieprep-charter.html`.
[13] "Implementing an ETS for Real Time Communications in the IP Suite" by F. Baker and J. Polk. Internet Draft, August 2005.
[14] "Integration of Resource Management and SIP" by G. Camarillo et al. RFC 3312, IETF, October 2002.
[15] "A Framework for Supporting ETS within a Single Administrative Domain" by K. Carlberg. Internet Draft, December 2005.

CHAPTER 17

Accessibility for the Disabled

This chapter is dedicated to Sigrid and Vinton Cerf, who have a wonderful family and a very successful professional life in spite of Sigrid having been born deaf and Vint suffering all his life from severely impaired hearing.

About Accessibility

Providing real-time communications to hearing- and/or speech-disabled people is both a moral and economic imperative, and is required by regulators across the world. Hearing- and speech-disabled people have often proven to be significant contributors to arts and commerce. Ludwig van Beethoven was deaf at a time when he composed some of his foremost and everlasting classical music. Vint Cerf was not stopped by his impaired hearing from co-inventing the TCP/IP protocol; founding and leading the Internet Society, the IETF, ICANN (Internet Corporation for Assigned Names and Numbers); and working as a key executive at MCI and Google.

The specific needs of the hearing- and speech-disabled for interactive communications can be best met using the Internet and SIP, as we will show in this chapter. Indeed, it is argued in [1] that SIP enables the view that being deaf, hard of hearing, or speech-impaired is no longer a barrier to communications.

Accessibility on Legacy Networks and on the Internet

The legacy telephone network, though it was designed and optimized for 3.1 kHz bandwidth speech, is also used for text communications for the hearing- and/or speech-disabled. Various signaling and display technologies have been introduced in various countries. Some of them are shown in Table 17.1 [2] (Asia and Africa are not included).

Table 17.1 shows that it is effectively impossible to conduct international text telephone communications for the disabled because of the lack of standards discipline in the legacy telecommunications world, where there are countless national, regional, and company-specific "standards." This lack of standards discipline is also manifest in all other areas of telecommunications. If the Internet and SIP did not exist, they would have to be invented.

Table 17.1 Some Current PSTN Textphone Systems

NAME	FEATURES	COUNTRIES
DTMF (Dual Tone Multi-Frequency)	Touch-tones as characters, simplex, 4 char/sec	Netherlands
EDT (European Deaf Telephone)	V.21 modem signaling, simplex, 10 char/sec	Austria, Italy, Germany, Malta, Spain, Switzerland
V.21 Text Telephone	V.21 modem signaling, duplex, 30 char/sec	Norway, Sweden, Finland, Denmark, Ireland, UK, Chech Republic
Minitel	V.23 modem signaling, simplex, 120/7 char/sec for forward/backward	France, Belgium
EIA-825 ("Baudot")	Simplex, 6 char/sec, uppercase only	USA, Canada, Australia, New Zealand
Bell 103 ("ASCII")	Similar to V.21	USA
V.18	V.21 modulation plus ability to adapt to all modulations above. Intended for "harmonization" of the fragmented world of PSTN text telephony	Not taken off in the market except in the UK as TextDirect

Requirements for Accessibility

The requirements to support rich communications for the hearing- or speech-disabled can be divided into generic requirements targeted for SIP and specific needs of the impaired users. We can thus break up the requirements into communication requirements and application requirements.

The generic requirements for SIP based communications for impaired users are:

1. Connection without difficulty in setting up SIP sessions and adding/removing media streams during a session.
2. Users must be able to communicate their abilities and preferences during a session for such features as text, voice, video; simplex or duplex.
3. Redirecting specific media stream to transcoders, for example, for speech-to-text conversion.
4. Roaming or SIP service mobility as discussed in Chapter 15, "SIP Application Level Mobility." A user should, for example, be able to use his or her service profile while visiting an Internet cafe that provides general-purpose PCs for communications.
5. Anonymity: Not revealing the user is speech or hearing impaired, even if transcoding is inserted into the session so as to avoid possible discrimination or prejudice.
6. Inclusive design: Users must be able to connect to other impaired persons that still use legacy protocols and devices via IP-PSTN gateways.
7. Personal resource management: Users must be informed of the media options and their price differences.
8. Confidentiality and security of similar quality as for nonimpaired users.
9. Real-time flow of all media.

The video application requirements for impaired persons have been documented in detail in [3]. Video is used for lip reading and signing in different flavors:

- Sign languages enable the communication of concepts, partial sentences, grammar, and nouns. Rapid hand movements and short blinks of the eyes carry grammatical information.
- Finger spelling is a subset of using a sign language where every letter corresponds to a unique hand position, as shown in Figure 17.1.

Figure 17.1 Example of finger spelling
Courtesy of Omnitor AB

The example in Figure 17.1 is interesting for designers, since it conveys a sense of the size and quality of the video images, and also the frame rate, as will be discussed here.

The frame rate is measured in frames per second (fps).

The most common video resolutions are:

- *Common Interchange Format (CIF)*—325 × 288 pixels
- *Quarter CIF (QCIF)*—176 × 144 pixels
- *Sub QCIF (SQCIF)*—112 × 96 pixels.

The video quality required for sign language is shown in Figure 17.2 expressed as usability (that is, as a function of the temporal resolution or fps, and the image resolution).

Text over IP has its own requirements, which will be discussed next.

Text over IP (ToIP)

Although instant messaging (IM) is considered an interactive text communication capability, it does not meet the needs of the hearing and speech disabled users:

- IM is used with times of seconds to minutes between messages. The Message Composition Indicator, such as "is typing" indicator (see Chapter 13) tells the other party to wait.

- Impaired users require a fast-response real-time text application, whereby, just as with voice, users can follow the thoughts of the other party as they are expressed in typing, and interrupt the other party. This application is called Text over IP (ToIP) or text conversation, and requires short delay times, less stringent than, but similar to, voice.

ToIP is another media that can be part of interactive communication sessions using SIP for signaling, besides IM, voice, video, and other forms. ToIP is a relatively simple application, and it is recommended that it be part of all SIP telephony devices [4]. ToIP is also carried in RTP packets, as described in [5]. The MIME types are called:

- `text/t140` for simple ToIP in an IP-IP environment
- `audio/t140` for use with IP-PSTN gateways, so as to interleave voice and text using the same gateway port to save gateway ports.

The payload of the RTP packets consists of short blocks of text that are defined for backward-compatibility using the ITU-T recommendation T.140 for text transmission.

An example for the RTP packet carrying the `text/red` and `text/t140` payloads with one redundant block is shown in Figure 17.3.

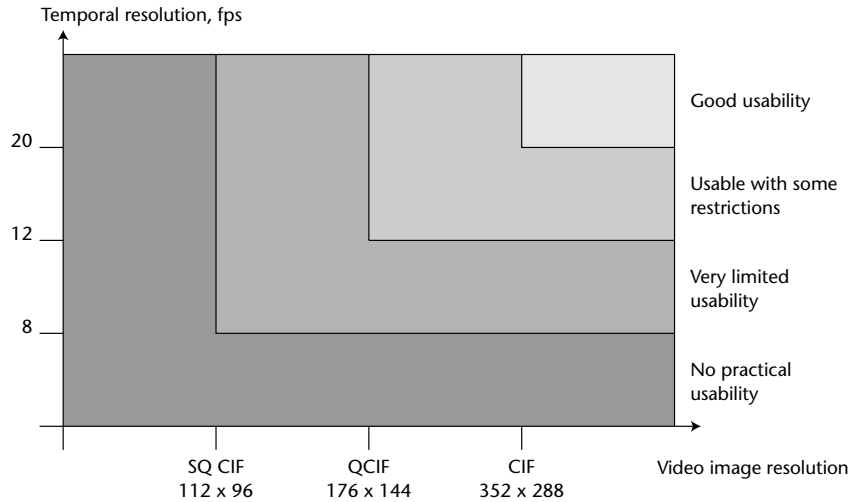


Figure 17.2 Usability as a function of frame rate and image resolution for different video QoS, as required for sign language and lip reading

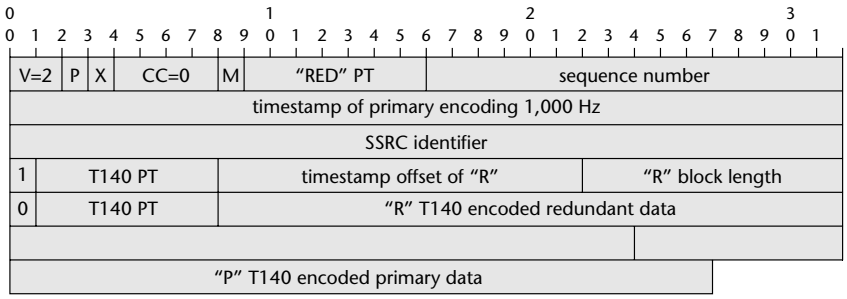


Figure 17.3 RTP packet example for `text/t140` with one redundant block

The text blocks may contain one or more text characters that are UTF-8 encoded. No delimiters are transmitted between the characters in a text block.

The components of the RTP packet for text conversation in the example, shown from the bottom up, are:

- T.140 encoded data: A primary data block "P" carrying the latest entered characters and a redundant data block "R" to have for correction of possible packet loss or errors on poor links.
- Payload type number for each block of text tied to the `text/T140` payload in the SIP session startup (T140 PT).
- The "R" block length.
- Timestamp offset of "R" from the current packet.
- RTP Synchronization Source (SSRC) identifier.
- Time stamp for "P." The clock for `text` timestamps is 1,000 Hz, while the clock for `audio/t140` is 8,000 Hz for compatibility with PSTN gateways.
- Sequence number of the text block. Can be used to detect missing packets and packets out of order.
- Payload type number tied to the redundant `text` payload type in the SIP session setup ("RED" PT).
- M (marker) bit is set to one in the first packet in a sequence following a silent period.

The other RTP fields at the start of the packet are as defined in RFC 3550:

- (CC) is a 4-bit CSRC count of the CSRC (contributing source) identifiers.
- X is an extension bit as defined in RFC 3550.

- (P) is one bit to indicate padding of the payload.
- (V) is a 2 bit indicator of the version for RTP.

An SDP example that describes RTP text transport using port 11000 and having two levels of redundancy is shown here:

```
m=text 11000 RTP/AVP 98 100
a=rtpmap:98 t140/1000
a=rtpmap:100 red/1000
a=fmtp:100 98/98/98
```

Performance Metrics for ToIP

The average speed for typing is considered here to be about 10 characters per second. Using 300 milliseconds as the average time between transmissions and considering three character payloads, the maximum average bit rate is 2,000 bits/s. For adequate QoS, the delay must not be greater than 500 ms, so as to support true interactive text conversation. The delay requirement for QoS for ToIP is, therefore, slightly less stringent than for voice, but not much.

The combination of ToIP with voice and with video for sign language is both very powerful and a natural application for even the lowest-cost PCs. The combination is also called *Total Conversation*. Such an application is shown in Figure 17.4.

Figure 17.4 The Total Conversion application
Courtesy of Omnitor AB

Transcoding Services

Transcoding services are provided by a human or machine acting as a third party to replace one media set with another. A media set can consist of voice, video, and text, depending on the impairments and preferences of the users.

Relay services are a subset of transcoding. Text relay services support the transcoding between voice and interactive text, and are widely deployed on the PSTN. By contrast, video relay services support the transcoding between voice and sign language and are deployed on the Internet.

Transcoding Scenarios

A large variety of transcoding scenarios are possible using the Internet due to the various multimedia possibilities and the capabilities of SIP for setting up point-to-point and multipoint communication sessions. This has been well documented based on significant early implementations of commercial services and research projects reported in RFC 3351 [1]. One such scenario is shown in Figure 17.5 as an example.

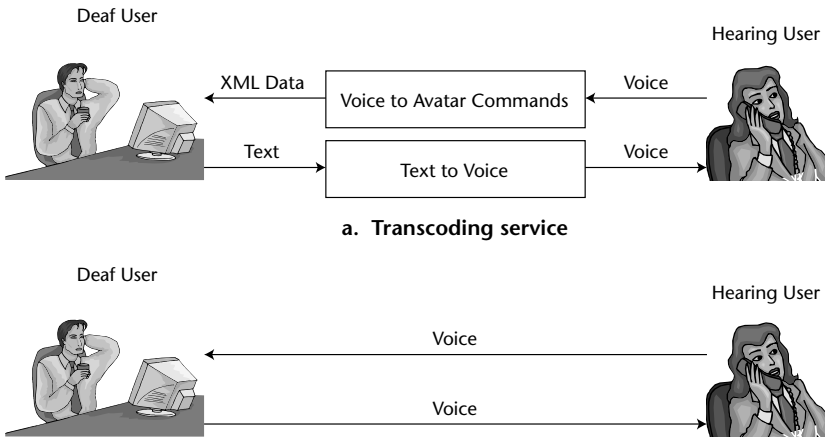


Figure 17.5 Examples for a transcoding scenario (a) network-based and (b) endpoint-based.

Note in Figure 17. 5 that the transcoding function can be located in the end-point PC if no human is required to perform the transcoding function. Early research and prototype PC-based transcoding functions have the following properties:

- Voice is a trivial application on all consumer-grade PCs.
- ToIP applications have very small processing and memory requirements.
- Voice-text conversion and voice recognition are common PC applications.
- Avatars for lip reading are superior to video for the following reasons:
  - Lip reading is difficult when the speaker turns the head.
  - Lips can be obscured by beards and moustaches.
  - Low articulation by the speaker may make lip reading hard to follow.
  - Greater confidentiality, since no human intermediary is involved.

Figure 17.6 shows an Avatar implementation in the Synface [6] project.

Figure 17.6 Avatar implementation on a laptop for a hearing-impaired person.

It is interesting to note that SIP-based communications for the disabled are also compatible with the legacy text phones used on the PSTN side of VoIP gateways. Figure 17.7 shows an example of a text phone designed for use on the PSTN.

The text phone shown in Figure 17.7 is also an interesting milestone for revealing the full richness of Internet communications for the disabled as discussed in this chapter.

RFC 3351 provides solid arguments for the use of SIP for rich communication scenarios and information about early endpoint implementations. As with Internet communications in general, we notice a keen competition between endpoint applications and network-based services along the lines of traditional PSTN-based relay services. Network-based transcoding services will be illustrated in the next section.

Call Control Models for Transcoding Services

Given the existing PSTN-based relay services and the familiarity of impaired users with relay services, it makes sense to maintain such services in the transition period during which communications are migrating to the Internet.

Using SIP call control to support network-based transcoding services [7] is trivial, as can be seen from the example shown in Figure 17.8.

Figure 17.7 Legacy text phone for use on the PSTN

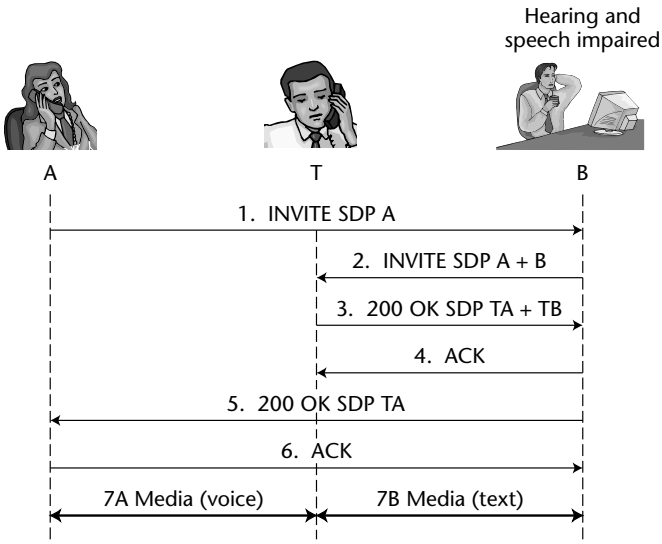


Figure 17.8 Example for relay service invocation by a caller

In the example shown in Figure 17.8, the caller may be on a 2G mobile network and the mobile/2G-Internet gateway is not shown here for simplicity. The called party is deaf and decides to invoke a relay service. SIP third-party call control is used on the IP side of the gateway. The call flow messages in Figure 17.8 is:

1. Caller A sends send an `INVITE` to B and the SDP A data payload indicates its connection data (where it expects the media streams) and its media capabilities.

```
(1) INVITE SDP A

        m=audio 20000 RTP/AVP 0
        c=IN IP4 A.example.com
```

2. Callee B sends an `INVITE` to T with SDP information of both SDP A and its own SDP B (SDP A+B).
3. Transcoder service T replies with `200 OK` and includes the SDP data it wants to use with A and with B (SDP TA+TB).

```
200 OK SDP TA+TB

        m=audio 30000 RTP/AVP 0
        c=IN IP4 T.example.com
        m=text 30002 RTP/AVP 96
        c=IN IP4 T.example.com
        a=rtpmap:96 t140/1000
```

4. B sends an ACK to T indicating the SDP data sent by T are OK.
5. Only now will B reply with a 200 OK message to A and include the SDP data for T (SDP T).

```
200 OK SDP TA

        m=audio 30000 RTP/AVP 0
        c=IN IP4 T.example.com
```

6. A will send an ACK message back to B. A is ready to send media to T.
7. The endpoints A and B are both exchanging media streams with T. There are two bidirectional media streams (a total of four):
   - Audio from A to T
   - Text from T to B
   - Text from B to T
   - Audio from T to A.

Notice in this example that caller A does not need to be aware that callee B is deaf and is using a relay service to receive text. The UA for caller A also does not need to be enabled for SIP third-party call control. Only the relay service and the deaf user need to support it.

The detailed messages for this and other scenarios for network-based transcoding are illustrated in RFC 4117. Call scenarios for relay services with human operators sometimes can support two human agents—one for each sense of transmission so as to provide some degree of privacy since. Each agent is included in only one direction of the conversation. By contrast, using text-to-speech conversion and/or avatars in the endpoints does not expose any private information to other humans in the call.

Summary

Accessibility to communications for deaf, hearing-impaired, and speech-impaired people is a social imperative for human and economic reasons. Existing relay services can support only audio and text. Implementing international communications for impaired users is practically impossible on the PSTN because of the many national and regional incompatible signaling and data formats.

Text over IP can support interactive text conversation and is a media type specifically designed for impaired users.

Video and avatars can support sign languages, finger spelling, and lip reading on low-cost consumer PCs, laptops, and mobile devices for impaired users.

Internet communication based on SIP is ideally suited to provide impaired users all the rich multimedia communications available to everyone else on the Internet. Hearing- and speech-Impaired people can use either endpoint-based applications or network-based transcoding services.

References

[1] "User Requirements for SIP in Support of Deaf, Hard of Hearing and Speech-impaired individuals" by N. Charlton. RFC 3351, IETF, August 2002.

[2] "Human Factors; Duplex Universal Speech and Text Communication" Draft ETSI Guide EG 202 320, 2005-02.

[3] "Application Profile-Sign Language and lip-reading real-time conversations using low bit rate video communications," ITU-T Series H, Supplement 1. Geneva, 05/99. Freely available online at `www.itu.int/home/index.html`.

[4] "SIP Telephony Device Requirements" by H. Sinnreich et al. Internet Draft, IETF, October 2005.

[5] "RTP Payload for Text Conversation" by G. Hellstrom et al. Internet Draft RFC2793bis to update RFC 2793, IETF, January 2005.

[6] The home page for the Swedish Synface project is `www.speech.kth.se/synface/index.htm`.

[7] "Transcoding Services Invocation in SIP Using Third Party Call Control (3pcc)" by G. Camarillo et al. RFC 4117, IETF, June 2005.

CHAPTER

18

Quality of Service for Real-Time Internet Communications

Quality of Service (QoS) for voice is a critical feature for real-time Internet communications. QoS for VoIP is also a much-abused topic, since it was used for a long time by the defenders of the PSTN and the TDM PBX to scare users away from VoIP. QoS is often invoked by network equipment vendors as a reason to buy new network and monitoring equipment—to make the network "VoIP ready." Since SIP is mainly used for VoIP throughout the industry, we will focus in this chapter on a balanced overview on Internet QoS for interactive voice and provide authoritative references for further study on this topic. A complete treatment would require a separate book. As an aid to readers looking for sources on QoS for VoIP, a simple test to characterize the source is:

- Can the author(s) be reached using VoIP in the office and at home?
- Is the document meant to sell some network equipment?

As we will argue in this chapter, it is acceptable for VoIP QoS to be used as an argument to sell *bandwidth*, since the higher speed the broadband connection is, the better VoIP will work. Also, as core Internet speed and broadband access use is increasing, the location for QoS moves from the network to the applications in the endpoints.

MR. QOS VS. MR. BANDWIDTH

The topic of providing bandwidth vs. deploying network equipment for QoS is much debated in technical forums and trade journals [1].

This chapter should help our readers navigate safely through the rocks of commercial pressure to buy QoS hardware, software, and whole QoS network solutions (the more costly, the more enjoyable to the vendors in the QoS industry niche for VoIP).

A short reality check will reveal that all commercial VoIP service providers, including former telephone companies or the giant IM and voice services such as AOL, Google, MSN, Skype, or Yahoo!, work quite well without QoS, since no one can control VoIP calls end-to-end between arbitrary points on the Internet. Also, Skype, Google, and others have proven with massive deployments in the market that quality for voice is mostly an endpoint property, as long as the path over the network does not suffer from plain congestion. Voice traffic is a negligible fraction of the Internet traffic and hardly contributes to network congestion.

The authors have conducted most of their telephone conversations for years over the Internet to enjoy the better-than-PSTN conference quality sound using our computers or SIP desktop phones.

As for IP PBX and IT network vendors arguing the case for expensive QoS equipment, remember that *customers and business partners* will never experience any of the presumed quality if it exists only inside their private network. It is hard to present an adequate business rationale for providing QoS inside private networks only.

As far as the issue of bandwidth versus QoS, QoS in the endpoints and the critical requirements of communications to/from anywhere on the Net should enable the making of informed decisions when investing in quality for voice communications. QoS is, however, not only required for interactive voice communications but also for video, as video becomes more prevalent on both wired and wireless SIP-based communications. 4G wireless networks especially will be able to support interactive video right from the start. It is useful to note that video codecs are more sensitive to packet loss: Synchronization mechanisms are annoyingly visible during resynchronization after large dropouts. In the following material, we will, however, discuss only QoS required for voice, since there is ample experience in the industry with it, while video seems to be an emergent application that so far has not received the same scrutiny as voice.

Voice Quality Metrics

There are three basic categories of quality for voice [2]:

- *Listening quality*—How users rate what they hear during a call. Instances where only the listening quality is critical are presentation/lectures over the network that are temporarily similar to other streaming media applications (such as Internet radio).
- *Conversational quality*—How users rate the ease of conducting an interactive voice conversation. This includes echo and delay that we will discuss here in more detail.
- *Network quality*—The impairments caused by the network are ordered here by severity:
  1. The network is unavailable.
  2. Voice dropouts caused by long packet loss bursts are experienced.
  3. High delay is irritating in interactive conversations.
  4. Delay variations (also called *jitter*) can induce the loss of voice samples in the receive buffer.
  5. High packet loss that is sensed as low speech quality is experienced.
  6. Miscellaneous—Occasional interruption of the call or failed call attempt because of dynamic IP address change (possibly once a day for certain ISPs) or, less often, happens because of route flapping on the Internet.

While these are basic considerations for voice quality of service, the argument is often made that different market segments and different customers may have different requirements for quality. There is, for example, the perception that "business-quality" voice must be better than "consumer-quality" voice. We will leave it to the reader to decide how such distinctions may or may not apply if a consumer calls a business for some service.

Delay Limits for Voice

The ITU-T recommendation G.114 is generally accepted throughout the telephone industry. Following are the values for one-way delay:

- Less than 150 ms for acceptable conversation quality.
- No more than 400 ms for tolerable conversational quality.
- Delay over 400 ms is deemed as unacceptable.

Some wireless SIP based features such as Push to Talk may have slightly different specifications for delay.

Burst vs. Average Packet Loss

The average packet loss figures quoted by most legacy telecom sources are not very meaningful for either data or for voice, and do not accurately reflect the behavior of IP networks and IP applications, since:

- Network congestion or route flaps produce long bursts of packet loss.
- Distributed packet loss is, therefore, not meaningful and is also easier to compensate for when transmitting either data or voice.

Internet voice codecs have a high tolerance for distributed packet loss over time, but long bursts of packet loss cannot be compensated for and translate into loss of speech syllables or even entire words and sentences, similar to that experienced with mobile phones. For this reason, the Extended Report (XR) has been defined for the RTP Control Protocol (RTCP) in RFC 3611 [3].

Acoustics and the Network

Voice is probably the most demanding real-time application for the Internet, though networked games are quite close, or even more demanding, when it comes to delay and packet loss. Figure 18.1 shows the main phenomena [4] at issue for real-time voice communications.

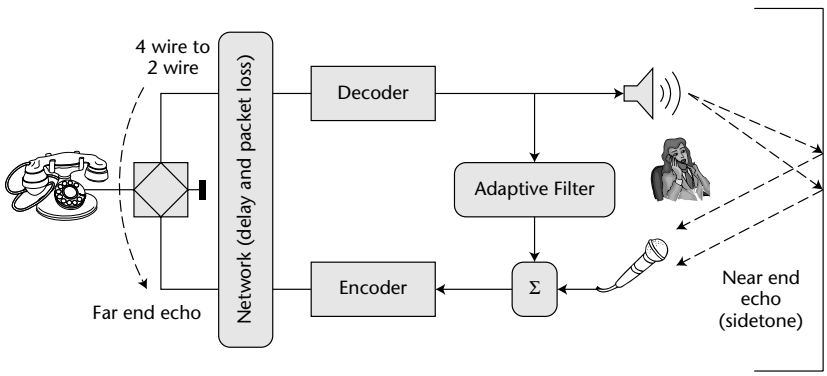


Figure 18.1 Acoustics and the network

The elements that influence the quality of interactive voice shown in Figure 18.1, from the left to the right are:

- *Far end echo* that is due to the feedback from the far end. A user having an analog phone, as shown, may cause the feedback in the "4 wire to 2 wire" element, also called the *hybrid* in analog telephony. In VoIP adapters or phones, the equivalent of the hybrid is a digital signal processing (DSP) application. *Tail length* is the amount of network delay to the left over which echo is controlled in the hybrid. A typical value is 64 ms for a voice sampling rate of 8 kHz. Far end echo is compensated by the echo canceller shown on the right (*adaptive filter*).
- The network can contribute with impairments to quality that will be discussed in the following section on Internet performance.
- The *coder* and *decoder* form the *codec* and serve to convert the digitized voice signal into a format suitable for transmission over the Internet, which is RTP media packets.
- The *near end echo* shown on the right is also called the *sidetone*. A sidetone is manifested most commonly when using multimedia PCs or laptops without a headset. The sidetone and the far end echo can be compensated for by using an adaptive DSP filter for the voice application.
- There are other sources of impairments, such as noise that sometimes has to be locally compensated for. An interesting observation is the fact that digital transmission over the Internet is practically noise-free, and this makes users unsure if the session is still alive. For this reason, many codec packages introduce comfort noise for the reassurance of the user.

Internet Codecs

Most telephony codecs used at present by telephone company-provided VoIP are part of the legacy ITU-T G.7xx series codecs designed for 3.1 kHz audio bandwidth. These were first developed for the now defunct ATM-PSTN gateways and are technically obsolete, with a few exceptions, such as the narrowband G.723.1 narrowband codec. There are more than 25 flavors of ITU-T legacy codecs and, to our knowledge, all but the G.711 codec (which is based on 50-year-old PCM technology) require license fees (which may explain their longevity with the legacy telecom vendors).

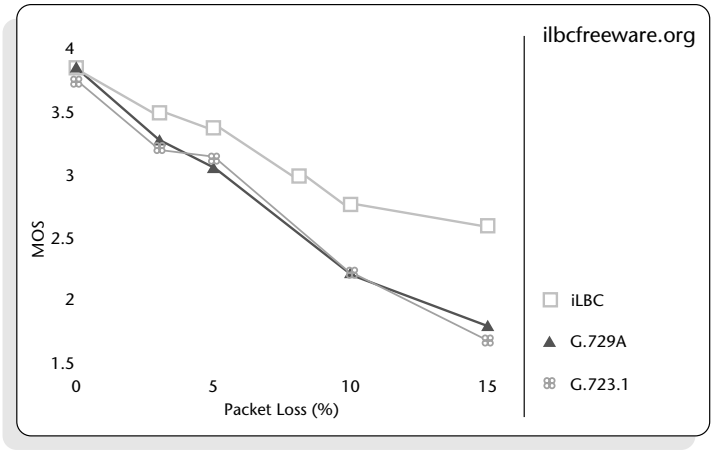
State-of-the-art Internet codecs feature a wide range of audio bandwidths that can deliver better-than-PSTN-quality voice and conference-room-quality sound, while at the same time using less bandwidth and having higher resilience to packet loss.

A comparison of the Internet Low Bit Rate Codec (iLBC) with legacy ITU-T codecs is shown in Figure 18.2 [5]. The iLBC codec is described in [6] and [7].

High-performance Internet codecs have also been developed as open source software, such as the SPEEX codec that uses audio sampling rates of 8 kHz, 16 kHz, and 32 kHz corresponding to audio bandwidths of 4 kHz, 8 kHz, and 16 kHz. The SPEEX codec [8] and [9] features are:

- Free software/open-source software
- Integration of wideband and narrowband in the same bitstream
- Wide range of bit rates available
- Dynamic bit-rate switching and variable bit rate (VBR)
- Voice activity detection (VAD, integrated with VBR)
- Variable complexity

Unfortunately, many VoIP users form their impressions of VoIP quality by the limitations of existing IP-PSTN and IP-PBX gateways that have legacy 3.1 kHz audio codecs.



The tests were performed by Dynstat, Inc., an independant test laboratory.
Score system range: 1= bad, 2 = poor, 3 = fair, 4 = good, 5 = excellent

http://www.ilbcfreeware.org
RFC 3951 : Internet Low Bit Rate Codec (iLBC)
RFC 3935 : RTP Payload for iLBC

Courtesy of Global IP Sound

Figure 18.2 Performance of the Internet Low Bit Rate Codec (iLBC)

Codecs in Wireless Networks and Transcoding

A wide variety of codecs are deployed in 2G and 3G wireless networks, and it is beyond the scope of this book to describe them.

Conversion between the various codecs deployed in wireline, wireless, and IP networks introduces both distortion and delay that is sometimes perceptible. Ideally, broadband wireless IP networks should not constrain the choice of codecs deployed in the endpoints, and there should not be any difference in codecs depending on the type off access, wired or wireless. Most 3G wireless networks have their own codec types, and one more reason to choose 4G wireless networks is not be constrained by the types of codec the SIP UA must use.

Codec Bandwidth

Codec bandwidth can be a consideration on frugal access links. Table 18.1 shows some typical voice codec bandwidths.

The bandwidths shown in Table 18.1 do not show the additional bandwidth consumed by the packet overhead for the encapsulation of the codec payload into RTP and IP packets. The RTP and IP headers can increase the effective bandwidth over the network 2–3 times the codec bandwidth shown here. The overhead for the RTP and IP headers is 40 bytes, and the exact bandwidth over the network is a somewhat more complex function of the frame lengths used for various codecs, typically 10, 20, or 30 ms. Most compressed codecs (this excludes G.711) have effective network bandwidths in the 16–32 kb/s range. As mentioned, the network load from voice is negligible compared to video and various P2P file-sharing applications.

Table 18.1 Typical Codec Bandwidths

CODEC TYPE	CODEC BW IN KB/S
G.711	64
G.729A	8
G.723.1	6.3
iLBC	15.2
SPEEX	4 - 44

The Endpoint Quality for Voice

It can be argued that, in the absence of notable network-induced impairments, the QoS for voice resides in the endpoint. A complex array of technologies is required for high-quality interactive voice communications [10]:

- Network echo cancellation
- Acoustic echo cancellation
- Noise cancellation
- Automatic gain control for transmit and receive level
- Voice activity detection for bandwidth efficient transmission
- Comfort noise generation

As we will show in the next section, the global Internet can support adequate to excellent performance for VoIP, with the exception of some less-developed regions in the world.

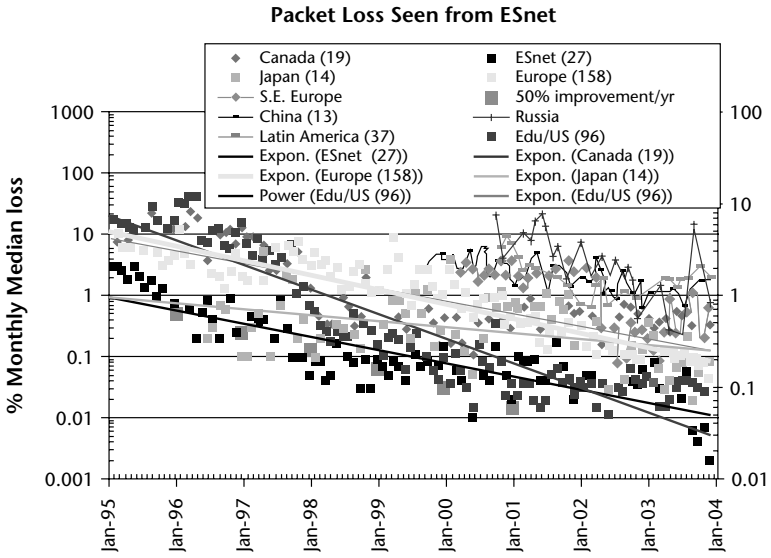
The Internet Performance

As the bandwidth in the Internet core increases, and broadband Internet access is delivering ever higher speeds, so does the Internet performance increase with regard to packet loss and delay.

Figure 18.3 illustrates how the monthly average packet loss over the Internet has decreased over 10 years by an order of magnitude and is at present in the 0.1 percent to 1.00 percent range except in a very few regions of the world. Most global ISPs guarantee average packet loss lower than 0.5 percent but actually deliver average packet loss in the 0.1 percent to 0.2 percent range over their networks. Going back to the codec performance in the presence of packet loss in Figure 18.2, it is obvious that average packet loss in the 1 percent range will not adversely affect codec performance in any way. Packet loss bursts may degrade quality for voice, but we have not seen any published data for Internet burst packet loss.

Figure 18.4 shows the global average delay measured over periods of 24 hours and 30 days, respectively.

These measurements prove that because of the very high-speed routers deployed on the Internet, the average delays are close to the delay of the speed of light in fiber-optic cables.



http://www.slac.stanford.edu/xorg/icfa/icfa-net-paper-jan05/

Figure 18.3 The decrease in average packet loss over the Internet over 10 years

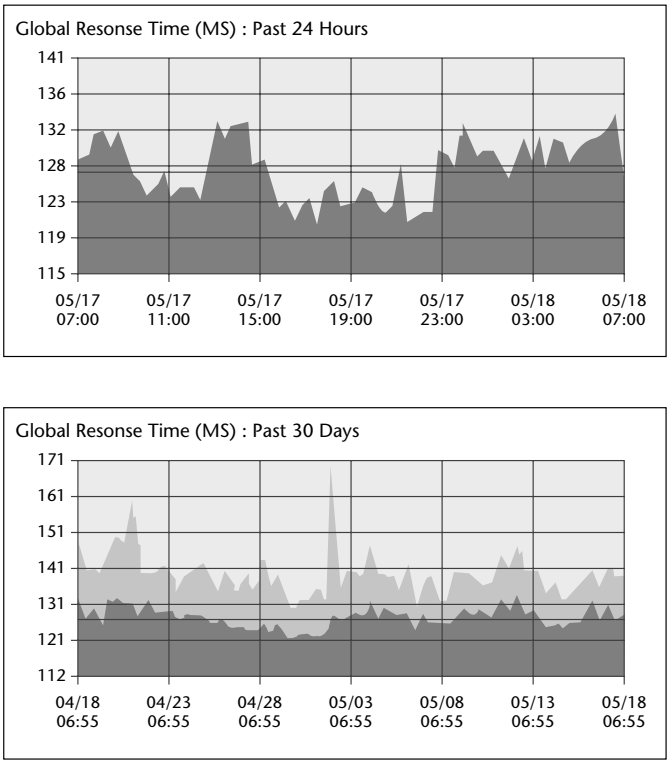
Concerns Regarding Congestion Control

Not all regions in the world enjoy broadband Internet access, and in some countries there is just not enough bandwidth to support acceptable VoIP. The lack of bandwidth has actually led some Internet experts to be concerned about the collapse of traffic if VoIP becomes more widespread in such bandwidth-starved networks [11].

We have not seen, however, any reports of such congestion-based collapse, and the proliferation of broadband and video in most parts of the world makes any noticeable large-scale congestion collapse because of VoIP highly unlikely.

Internet Traffic Statistics: Voice Is Negligible

Several studies show that worldwide Internet traffic is dominated by P2P applications, such as the file sharing of music and video. P2P traffic amounts from 60 to 80 percent of traffic measured on the networks of several ISPs [12], [13], as illustrated in Figure 18.5.



http://www.internettrafficreport.com/main.htm

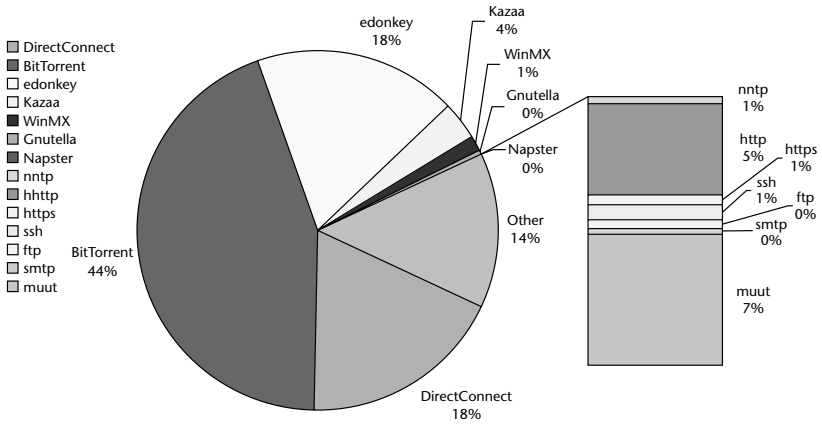
Figure 18.4 Average delays over the Internet

We recommend readers interested in this topic check Web sources at the time of reading this book, since the Internet is a fast-changing environment.

The introduction of IP TV may even further increase the usage on broadband access links for streaming video.

Current measurements of P2P traffic and forecasts for streaming video indicate that voice traffic is (at present and will remain) a negligible component of the overall traffic. Management of VoIP traffic, therefore, makes no sense, except on bandwidth-starved portions of the Net or in private IP networks where bandwidth is a premium (such as on networks using satellite links for remote parts of the world, or for onboard networks of ships and airliners).

The emergence of video such as IP TV on cable and DSL, as well as various video services in wireless networks, may change the mix of traffic types, but only in the sense that more and more bandwidth will be required, in comparison to voice. Voice will, thus, continue to be a beneficiary of the proliferation of broadband.



http://linuxreviews.org/news/2004/11/05_p2p/

Figure 18.5 Sample of P2P and other Internet traffic

A Summary of Internet QoS Technologies

Network quality of service is a very complex topic. Table 18.2 provides a very short summary.

Table 18.2 Network Technologies for Quality of Service

OPTION FOR NETWORK QOS		HIGHLIGHTS AND ISSUES
Network Layer	Best effort	Used on the Internet at present. Requires overprovisioning of bandwidth.
	Differentiated Services (DiffServ or DS) RFC 2475, 3086	Per-hop IP routing behavior defined by the DS Code Points (DSCP) for Expedited Forwarding and Assured Forwarding. Stateless approach with minimal complexity.
	Resource Reservation Protocol (RSVP) RFC 2205	The endpoint (host) requests specific end-to-end reservations along the routing path.

(continued)

Table 18.2 *(continued)*

OPTION FOR NETWORK QOS		HIGHLIGHTS AND ISSUES
Network Layer		RSVP requires state in all nodes in the path.
	Multiprotocol Label Switching (MPLS)	MPLS establishes label-switched paths. MPLS can invoke DS and RSVP. Is one option for intradomain traffic engineering in the IP core, but is not usable on access links where QoS may be a problem.
Link Layer	802.1q and 802.1p	VLAN specifications with a 3-bit priority field that can be mapped to DS.
	802.11e	Can support QoS on wireless LANs using eight traffic classes.
	DOCSIS 1.1	Data Over Cable System specification enhanced to support 802.1q.

Data in Table 18.2 leads to three important conclusions:

1. QoS based on best effort is the only one available for interdomain traffic over the Internet. We will expand later in this chapter on the reasons for the persistence of best-effort interdomain traffic.
2. Differentiated Services (DS) is the only technology for QoS support that is stateless and also has no scalability problems.
3. MPLS is not applicable on access links where most of the QoS issues originate in the first place. MPLS technology is considered a successor of ATM and has inherited, in our opinion, the same mindset leading to similar problems that ATM had. MPLS has also been described as a security risk for the Internet, and as leading to unmanageable routing table complexity for ISPs [14], [15]. (Such concerns by network experts have led to a flood of white papers to the contrary written by router and network marketers.)

QoS on access links can, however, be easily implemented, and many residential and small office routers support setting the DSCP for the access link, in case DS is provided by the ISP.

QoS must also sometimes be implemented in private networks that have a wide geographic reach and have bandwidth-starved links to remote locations.

In "Requirements for SIP Telephony Devices" [16], the following is recommended: SIP devices must support the IPv4 DSCP field for RTP streams per RFC 2597. The DSCP setting must be configurable to conform to the local network policy. SIP telephony devices should:

- Mark RTP packets with the recommended DSCP for expedited forwarding (code point 101110)
- Mark SIP packets with DSCP AF31 (code point 011010)

Similar requirements have been developed for IPv6.

Best Effort Is for the Best Reasons

Following are some fundamental reasons why interdomain QoS is very difficult to implement and explain, in part, why QoS has not been deployed on the Internet [17]:

- There is no scalable way to transfer authority between Internet domains.
  - Each Internet domain is a zone of authority and has its own management.
  - Domain authorities are run most often by commercial competitors.
  - Different domains use different policies.
  - There are no technologies available to protect from misconfigured domains.
- All current interdomain QoS mechanisms create vulnerabilities that can be exploited for theft of service or for DOS attacks.
- Interdomain flows are aggregates, but offending users from other domains must be individually traced.

These reasons explain why the common practice on the Internet is to provision adequate bandwidth between adjacent Layer 2 networks (such as Gigabit Ethernet) of interconnected domains and to monitor the network to avoid congestion. Network loads of 30 to 40 percent are considered adequate for QoS and for congestion avoidance.

Monitoring QoS for Real-Time Communications

The concern with ensuring adequate voice quality within domains has led telephony-minded network operators to deploy probes that monitor packet loss, delay, and even voice-packet-like performance in various locations in the network. This is actually monitoring network performance for one specific application: telephony. Such equipment is sometimes deployed in addition to the traffic-monitoring capabilities of most types of IP routers.

The similar and fractal nature of Internet traffic makes it, however, unlikely that observations from a limited number of probes distributed in a network domain can accurately describe the voice quality perceived by any specific user, at a specific time for a specific endpoint. Recent IETF work that has had the benefit of many inputs from the industry and several prestandard implementations has been aimed at standardizing a better approach:

- To obtain quality reports that are closest to the actual user experience, the only logical placement is in the SIP UA application.
- The extended RTCP reports (RTCP-XR) are used to report the burst error packet loss for one of for all endpoints in the session.
- If the QoS falls below a predetermined threshold, a real-time alarm can be provided for the network administrator.
- If no QoS threshold alarms occur, the QoS data can be stored in the endpoint(s) and periodically reported to a third party, such as a server in the network.
- Fat network pipes in the core with lots of aggregated flows are the only place in the network where probing is meaningful. Here, the similar nature of the traffic fades away and the utilization of probes provides a useful measurement of quality. However, this measurement of quality is mostly an indicator for required capacity increase or rerouting.

The techniques to support this procedure are described in detail in the IETF standards track document on the SIP event package for reporting RTCP-XR events [18]. Note the approach here is not based on network management techniques but on monitoring the quality metrics for applications using SIP events. This technique is also applicable for monitoring other real-time applications running in SIP endpoints. The event package can be used either with the `SUBSCRIBE/NOTIFY` methods or the `PUBLISH` method using the Voice Quality Syntax expressed in BNF. The most recent software implementations have shown the image to be small enough to fit in SIP phones or PC-based SIP UAs.

Summary

The proliferation of broadband and high-speed Internet core networks has moved the placement of QoS for voice from the network to the endpoints.

Endpoint design for high-quality interactive voice includes state-of-the-art Internet codecs, far end and near end echo control, automatic level control, and other voice application design items.

There is a good assortment of network layer QoS and link layer QoS mechanisms available, but they can be applied mostly in intradomain communications, and this is not too interesting.

The simplest and most scalable approach for intradomain QoS are differentiated Services, followed by RSVP, and the least favored by us is MPLS.

By contrast, interdomain QoS is not deployed on the Internet because of an assortment of showstoppers, ranging from differences between domains in ownership, policies and commercial competition. Last but not least, interdomain QoS has to be protected against theft of service and DOS attacks.

The best effort QoS on the Internet is, therefore, well grounded in reality and works well as long as network congestion is avoided by adequate provisioning of bandwidth. Voice uses a negligible fraction of Internet traffic.

Monitoring the voice quality is best done in the SIP endpoints, closest to the end-user experience.

References

[1] "Mr. QoS vs. Mr. Bandwidth" by S. Borthick. *The Business Communication Review*, September 2005, pp 16–17.

[2] "Voice Quality Measurement" Technical Note, Telchemy, Inc., January 2005. `http://telchemy.com/appnotes/TelchemyVoiceQualityMeasurement.pdf`.

[3] "RTP Control Protocol Extended Reports (RTCP XR)" by T. Friedman et al. RFC 3611, IETF, November 2003.

[4] "Network and Acoustic Echo Issues in Voice-Over-Packet Telephony Systems" by P. Sorquist. Global IP Sound AB, white paper, 2002. `www.globalipsound.com`.

[5] The home page for the iLBC freeware is `www.ilbcfreeware.org`.

[6] "Internet Low Bit Rate Codec (iLBC) " by S. Andersen et al. RFC 3951, IETF, December 2004.

[7] "Real-time Transport Protocol (RTP) Payload Format for Internet Low Bit Rate Codec (iLBC) Speech" by A. Duric et al. RFC 3952. IETF, December 2004.

[8] "RTP Payload Format for the Speex Codec" by G. Herlein et al. Internet Draft, IETF, October 2004.
[9] The home page for the open source SPEEX codec is www.speex.org.
[10] "GIPS Voice Quality Enhancement" Datasheet of Global IP Sound A B, 2005. www.globalipsound.com.
[11] "IAB Concerns Regarding Congestion Control for Voice Traffic in the Internet" by S. Floyd and J. Kempf. RFC 3714, IETF, March 2004.
[12] "Peer-to-Peer in 2005" CacheLogic Research. www.cachelogic.com.
[13] "Peer to peer network traffic may account for up to 85% of Internet bandwidth usage." *Linux Review*, November 2004. http://linuxreviews.org/news/2004/11/05_p2p.
[14] "Experts call MPLS bad for the 'Net" by C. M. Marsan. *Network World*, August 6, 2001.
[15] "Warning: MPLS and Rapid Spanning Tree could be hazardous to your network" by J. Duffy. *Network World*, August 10, 2001.
[16] "SIP Telephony Device Requirements and Configuration" by H. Sinnreich et al. Internet Draft, IETF, October 2005, work in progress.
[17] "A Framework for Supporting Emergency Telecommunication Services (ETS) Within a Single Administrative Domain" by K. Carlberg. Internet Draft, IETF, December 2005.
[18] "SIP Service Quality Reporting Event" by A. Pendleton. Internet Draft, IETF, December 2005, work in progress.

CHAPTER 19

SIP Component Services

Applications provided by service providers have some history, and it is remarkable that even in the IP environment, many wireline and wireless service providers have not yet learned from this history. This is the reason why, in this chapter, we will show how value-added services by providers using SIP can be implemented, using application servers in an open, distributed, and loosely coupled architecture that is highly scalable. The application server approach is based on the client-server (CS) model for SIP.

Using application servers in the network is, however, not the only approach. In Chapter 20, "Peer-to-Peer SIP," we will show how applications can also be implemented in peer nodes. In the extreme, the most frequent applications can reside entirely in the endpoints, or, in a mixed environment, some applications can also reside in P2P SIP supernodes, where the architecture described in this chapter will apply.

NOTE **On a historical note, among issues not to forget is that value-added services in the PSTN are implemented using the Intelligent Network (IN) [1] based on central control. The IN is a collection of servers and other resources used to control call setup and to provide voice features, such as announcements, voicemail, and so on. In hindsight, IN services seem rather frugal compared with communications on the Internet. Other architectures, such as H.323 or the so-called "softswitches" based on IP telephony gateway decomposition [2], have similar approaches to the IN for enhanced voice features. We use the term**

"softswitch" in quotes, since it is mostly a marketing concept to designate the combination of the call agent and media converter in IP-PSTN gateways, but also for the central control of MGCP phones or adapters for end users.

A close look at the telephony-oriented approaches will reveal that they provide little more than voice-only features ad nauseam, such as call forwarding, and so on, inspired by the by the PBX, since the telephone companies were hoping to replace private PBX voice networks with carrier-based Centrex voice services.

Master/Slave VoIP Systems

Device control protocols can be found in proprietary IP PBX designs and also in various approaches for VoIP such as Media Gateway Control Protocol (MGCP), Media Gateway Control (MEGACO), and H.248. The decomposition of an IP telephony gateway using a device control protocol between the gateway controller (GC) and the media gateway (MG) is shown in Figure 19.1a.

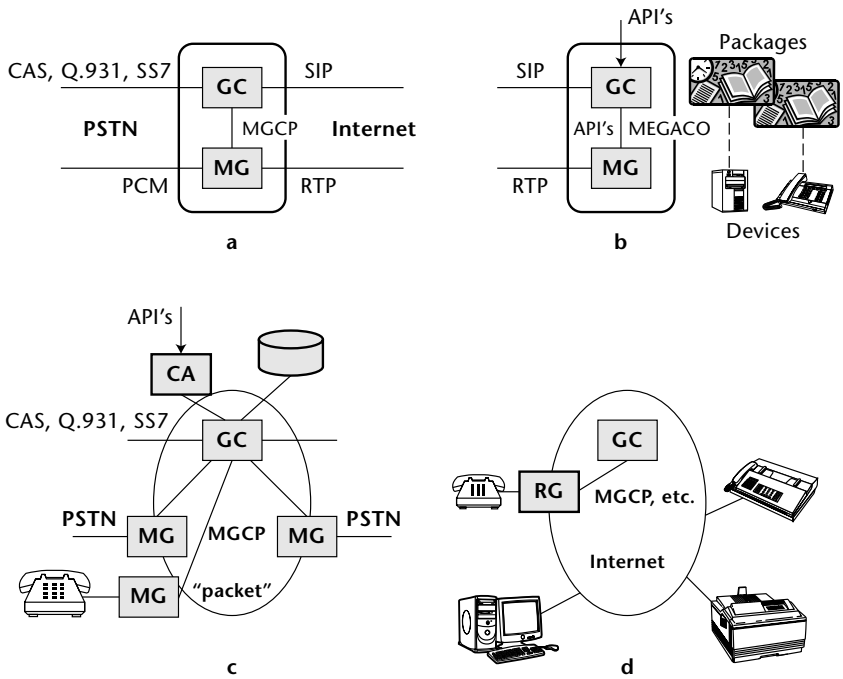


Figure 19.1 Decomposition using master/slave protocols (a) IP telephony gateway, (b) application server, (c) IP telephony gateway network with central call agent (CA), and (d) residential gateway (RG) for telephony

Device control protocols are master/slave protocols where every detail of the device operation is controlled from a central server. Master/slave protocols are also sometimes called "stimulus protocols," since every event or stimulus experienced by the terminal must be relayed to the controller using the protocol. In this model, every event (such as a hook flash), has to be reported to the central controller, and every action of the device has to be controlled (such as how to display a number or a message). All this activity generates a large and mostly unnecessary amount of network traffic between the GC and all the MGs (not shown in Figure 19.1a) compared to the traffic that would be required if the MG would have its own control intelligence.

How does the controller know if the device has a display at all? The answer is, it does not know, unless it has been preconfigured with a so-called *package* (Figure 19.1b) that is written for that particular device (such as, for example, for a specific phone model that may or may not have a display with certain capabilities).

Since the package in the control device has to know every detailed feature of the controlled device, which is also dependent on various product versions, it is practically impossible to have them made by different and possibly competing vendors, in spite of the standard control protocol used between the controller and the device.

In the case of media gateways, packages have to be written and provisioned, depending on the particular circuit switch network signaling of the media gateway such as channel-associated signaling (CAS), Q.931, and SS7 in its various national variants. This forced bundling of the controller with all controlled devices seems to be the right prescription for vendors to lock in their customers, the service providers.

The central control in master/slave protocols is not scalable.

This is in stark contrast to the Internet model, where the implementation of networked devices does not need to be known by other devices to interwork and, even more, where interworking devices have to be developed in an independent manner by competing designers around the world. You can communicate with an Internet-connected device without caring if, at the other end, there is a palmtop computer or a powerful server farm, or some mainframe computers in a data center. Also, when using FTP, e-mail, or browsers, no consideration has to be given how the remote IP device is configured. Device control protocols also have no notion of redirection, and a controlled device cannot refuse a request (unless it reports an error) or offer alternate destinations to honor a request.

New features in the slave devices are useless unless they are also supported by the master. This limits the ability to rapidly develop new services and features, since extensions to the package must be defined. In comparison to SIP, new headers to implement new services and features can be implemented by endpoints only, in most cases, without the knowledge or support in the SIP network. SIP system designers can choose to put only the absolutely minimum required features in SIP servers: user registration and the proxy function.

Master/slave protocols, such as MEGACO, only succeed in reducing infrastructure costs if the simplification of the extremely "dumb" terminals offsets the increased costs of additional protocols and of new intelligent network elements. However, when a requirement is added to the "dumb" terminals to be able to act autonomously under certain circumstances (for example, complete an E911 call when no controller is available), most of the assumed lower-cost benefits of a master/slave protocol will be lost.

IP Telephony Gateways

The earliest implementers of IP telephony gateways used monolithic and highly proprietary approaches for auxiliary functions such as tone announcements and IVR functions or, for example, for credit card number input. Small gateways can be built using application programming interfaces (APIs), depending on the particular product and operating system. However, such monolithic designs proved to be undesirable for both vendors and service providers, because they tried to scale the systems in size and across the network, and to add various new services.

The abundance of services and features in the competitive marketplace led service providers to search for unbundled systems, so as to benefit from products by multiple vendors, specialized to be the best of the breed.

A first attempt to provide unbundled IP telephony gateways was the decomposition of the gateway into a gateway controller (GC) and one or more media gateways (MG), as shown in Figure 19.1a.

The initial MEGACO protocol has been made obsolete in the IETF and replaced by the Gateway Control Protocol is described in [2]. The link between the GC and the MG has undergone numerous developments, starting with APIs and later giving birth to protocols with names such as IP Domain Control (IPDC), Simple Gateway Control Protocol (SGCP), MGCP, MEGACO, and H.248. Initially, the de facto industry standard was the MGCP. The IETF and the ITU have coordinated the development of the protocol, called MEGACO in the IETF and H.248 in the ITU. These standards were developed with some broader aims, such as to accommodate both SIP and H.323, and to be used for the control

of IP gateways to ATM networks and, last but not least, for the control of ATM circuit switches for voice. MEGACO and H.248 are, thus, considerably more complex than MGCP, without offering any more functionality.

All of the preceding protocols have one feature in common: They are master/slave protocols, where an "intelligent" central master controls every action in detail of the "dumb" slave devices, such as media gateways, media servers, and slave telephones.

The gateway controller is also sometimes called a "softswitch." Various designs have started out with the model in Figure 19.1a and have added proprietary APIs for third-party developers to add new services and also APIs to control the MG itself, as shown in Figure 19.1b. Since each system has its own APIs, third-party developers would have to learn all the APIs for all the various proprietary designs. Full-featured multivendor interoperability between the MG and GC is more difficult to achieve, the more APIs there are. Complete interoperability has not been accomplished in the industry, to our knowledge, as of this writing, and there are companies that have found a niche in writing code to for GCs to interoperate with various MGs.

As the number of required services increases, the need for separate service platforms becomes evident. Figure 19.1b shows the decomposition of the service platform between a service controller and media servers using one of the previous master/slave protocols. This decomposition has, however, the well-known drawbacks of central control, such as the following:

- Single point of failure (if there is only one geographic location).
- Proprietary service logic.
- Heavy control traffic between master and slaves leads to very lengthy and complex call flows.
- Details in implementations by vendors and APIs make interoperability unlikely.
- Bundled services inhibit third-party application providers.
- New services are difficult to introduce because of tight coupling of features.
- Integration with Web, e-mail services, presence, and IM is very difficult.

We believe the last item to be the most restrictive for the architecture shown in Figure 19.1b for the master slave approach.

The decomposition using master/slave protocols (such as MGCP or MEGACO/H.248) has constraints for service providers. Figure 19.1c shows a network composed of IP telephony gateways used to bypass the PSTN long-distance and international networks, or to avoid PSTN trunking for PBXs in

enterprise networks. The GC is controlled by a call agent (CA), where the service logic resides, and has access to the necessary database to control call setup.

The central control, the proprietary controller, and the control protocol now have produced a network that is neither the TDM network nor an IP network, but, in effect, a third type of some proprietary network hybrid—the ones that are the most difficult to operate. This new network can provide voice services only. Service providers have enough work cut out for them to manage existing circuit switched networks and the IP network, and need not trouble themselves with the managing a third type of network. Such a PSTN or PBX bypass network cannot support any services that do not exist already on the circuit switched side, thus taking away the main rationale for such third, new networks to operate.

Central control of distributed media gateways, as shown in Figure 19.1c, may be useful, however, in such cases where many smaller IP telephony gateways from an ISP have to interface with the PSTN using Signaling System 7 (SS7) signaling. Since SS7 interconnection points are quite expensive, and no other services than voice are possible anyway over the PSTN side of the call, a central controller combined with an SS7 interconnect point makes good sense. However, ISPs have to be careful not to have any service features provided by the central controller, since such services would be difficult to extend across the rest of the IP network, where multivendor compatibility will be required. This example is an exception to the rule, in our opinion, to avoid central control-type IP telephony gateway networks.

The residential gateway (RG) shown in Figure 19.1d is another example of the use of master/slave protocols such as MGCP and MEGACO/H.248. This time, it is the end user who is deprived of three main benefits available on the Internet:

- Free choice of any other service, as is the case on the Web
- Free choice of any communication application, since all applications reside in the central office of the service provider
- Telephony that is integrated with other services (telephony is segregated from all other Internet applications)

Residential gateways for voice as shown in Figure 19.1d are negating the requirements for equal access to service providers, since competing service providers cannot have access to control the phones or IP-phone adapters behind the residential gateway.

We will show by contrast how these problems can be avoided by using an Internet and Web-centric architecture for the application environment.

The Converged Applications Environment

The converged applications environment is based on the distributed Internet and Web architecture and is not dependent on any proprietary APIs and operating systems for internetworking of multiple servers. It is based on simple SIP and HTTP message, flow only for all control functions. The open architecture is especially well suited for third-party service providers across IP networks or across the Internet. The Application Server Component Architecture for SIP was first introduced in [3].

Figure 19.2 shows the integration of communications with applications and transactions, as is required for e-commerce. The real-time communications part is emphasized here with the main communication servers logically clustered around the capability to exchange SIP and HTTP messages. The various components are loosely coupled, in the sense that once their functions have been invoked by simple call flows, the details of operation are left to each server, without affecting the operation of other servers.

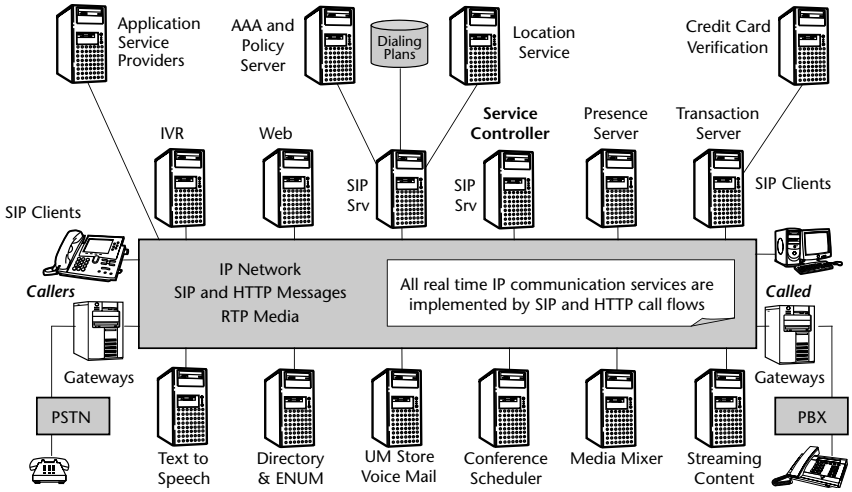


Figure 19.2 Component servers for communications, applications, and transactions

Following are the main types of communication servers:

- General-purpose SIP server (in the center, with database access) acting as registrar, redirect server, and for admission control in conjunction with the AAA and location services, such as databases or ENUM. The redirect server also can implement private dialing plans for enterprise networks.

- Service controller for delivery of services in conjunction with specialized communication servers, as will be shown in the following examples. The service controller uses SIP third-party call control [4] to orchestrate the interaction between the various servers.
- Voice portal using VoiceXML [5] technology for voice control and voice browsing. This also acts as an interactive voice-response (IVR) server.
- Web server for provisioning and control by end users.
- Presence server.
- Text-to-speech server.
- Voice-recognition server.
- Universal messaging (UM) server.
- Conference scheduler.
- Media mixer for audio conferences.
- Content server for streaming multimedia, such as stored presentations, shows, and so on.

In addition to servers for communications, other servers round out the portfolio:

- Transaction server for credit card transactions.
- Application service providers such as productivity software. This allows for the integration of office applications (document editors, spreadsheets, presentations, databases) and personal information managers with real-time communications.

Really interesting applications for outsourcing, however, go beyond the generic services shown here:

- Services of general interest such as travel and weather
- Highly specialized services such as security by voice recognition or Public Key Infrastructure (PKI) systems
- Virtual communities for business and nonprofit organizations

Service providers offering such an open and integrated environment for Web, e-mail, and voice also can be referred to as application infrastructure service providers.

How does it work? Users can provide inputs to the service controller either via the Web servers, the Dual-Tone Multi-Frequency (DTMF) digit collector, or the voice portal using speech recognition with VoiceXML, or simply DTMF input. This allows invoking services using a wide variety of devices, ranging from plain PSTN phones to PCs and palm computers. The user input can be either by voice channels or by using Web pages.

Open protocols are used exclusively. As a consequence, servers can be distributed across the network and can be provided and operated by various parties, using appropriate Internet security procedures, such as some form of secure IP tunneling. All real-time communication servers use only SIP and HTTP to communicate, as will be shown later in this chapter. No APIs are required. This makes the architecture completely open and allows easy outsourcing for specialized or high-performance services, such as unified messaging, instant messaging, or conferencing.

There is a loose coupling between service components. The service controller only invokes various service components by providing call control and leaves the detailed operation to the respective servers.

Dedicated servers also allow the use of application switching in high-traffic service hosting centers such as routing Web, e-mail, and various SIP and RTP communication flows to the appropriate servers.

Figure 19.3 Shows a network based services portal for e-mail, Web and voice.

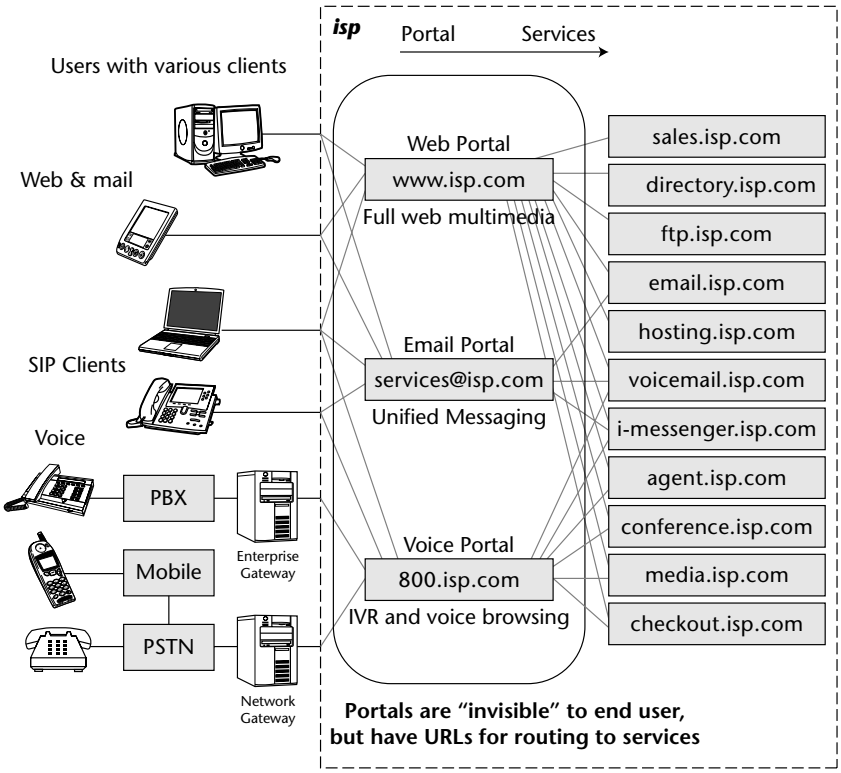


Figure 19.3 Network based services portal.

The architecture is completely distributed. Internet-style alternate servers using DNS for load distribution provide a high degree of reliability. Single points of failure are thus avoided. There is no need to rely on boxes that have the infamous "five nines" of the PSTN.

Two or more levels of authentication are required in this architecture. Users need to authenticate themselves to the controller, and controllers need to authenticate themselves to the various servers, especially if some of the services are outsourced.

In the examples that follow, PSTN-IP VoIP gateways are not shown for clarity in describing the SIP call flows.

The Control of Service Context

The use of the SIP URI described in Chapter 4, "DNS and ENUM," is all that's required to build complex component service systems for large provider networks. A good example would be the options for addressing a voicemail server and the various functions of voicemail [6]. The various voicemail functions are shown in Table 19.1 with three options to design the SIP URIs:

1. The function is the name in the user part of the SIP URI.
2. A voicemail phone number is used separately for each user and each voicemail function.
3. Use the attribute `mode` in the domain part to distinguish the functions for the same SIP URI.

Table 19.1 Options for Service Context for Voicemail Using the SIP URI

URI IDENTITY	EXAMPLE SCHEME 1 EXAMPLE SCHEME 2 EXAMPLE SCHEME 3
Deposit with standard greeting	sip:sub-rjs-deposit@vm.mci.com sip:677283@vm.mci.com sip:rjs@vm.mci.com;mode=deposit
Deposit with on phone greeting	sip:sub-rjs-deposit-busy.vm.mci.com sip:677372@vm.mci.com sip:rjs@vm.mci.com;mode=3991243
Deposit with special greeting	sip:sub-rjs-deposit-sg@vm.mci.com sip:677384@vm.mci.com sip:rjs@vm.mci.com;mode=sg

Table 19.1 *(continued)*

URI IDENTITY	EXAMPLE SCHEME 1 EXAMPLE SCHEME 2 EXAMPLE SCHEME 3
Retrieve - SIP authentication	sip:sub-rjs-retrieve@vm.mci.com sip:677405@vm.mci.com sip:rjs@vm.mci.com;mode=inpin
Retrieve - prompt for PIN in-band	sip:sub-rjs-retrieve-inpin.vm.mci.com sip:677415@vm.mci.com sip:rjs@vm.wcom.com;mode=inpin
Deposit - identitfy target mailbox by To:	sip:deposit@vm.mci.com sip:670001@vm.mci.com sip:deposit@vm.mci.com
Retrieve - identify target mailbox by SIP authentication	sip:retrieve@vm.mci.com sip:670002@vm.mci.com sip:retrieve@vm.mci.com
Deposit - prompt for target mailbox inband	sip:deposit-in@vm.mci.com sip:670003@vm.mci.com sip:deposit@vm.mci.com;mode=inband
Retrieve - prompt for target mailbox and PIN in-band	sip:retrieve-in@vm.mci.com sip:670004@vm.mci.com sip:retrieve@vm.mci.com;mode=inband

Using this design approach, very complex component service systems can be designed by just designing the SIP URI schema in the system.

Another URI parameter of interest is the `cause` for redirecting the call [7]. The `cause` parameters are shown in Table 19.2.

Table 19.2 Cause Values for Redirecting the Call

REDIRECTING REASON	VALUE
Unknown/Unavailable	404
User Busy	486
No Reply	408
Unconditional	302
Deflection during Alerting	487
Deflection immediate response	480
Mobile subscriber not reachable	503

The cause parameters can indicate to the caller or to a service component where to redirect the call and what operations to perform next.

For example, a call that has been redirected will now send an INVITE such as:

```
INVITE sip:voicemail@example.com;\
       target=sip:+15555551002%40example.com;user=phone;\
       cause=486  SIP/2.0
Via: SIP/2.0/TCP 192.0.2.4:5060;branch=z9hG4bK-ik80k7g-2
Via: SIP/2.0/TCP 192.0.2.1:5060;branch=z9hG4bK-74bf9
From: Alice <sip:+15551001@example.com;user=phone>;tag=9fxced76sl
To: sip:+15555551002@example.com;user=phone
Call-ID: c3x842276298220188511
CSeq: 1 INVITE
Max-Forwards: 70
Contact: <sip:alice@192.0.2.1>
Content-Type: application/sdp
Content-Length: *Body length goes here*...

* SDP goes here*
```

The Cause in line three is 486 and indicates the reason is User Busy.

Voicemail

Users can control services either by voice or by using forms on Web servers. We will show in this example how a user can invoke voicemail using the Web server. The simplified call flows are shown in Figure 19.3.

As shown in Figure 19.4, the caller uses a Web page to click on the URL of the called party intended to receive the voice message. The Web server requests in Message 1 the controller to connect the caller with the voicemail server. The controller then connects the user's SIP client with the voicemail server using SIP third-party call control. The call to the message server is accepted in Message 7, and the SDP data from the message server is conveyed in the re-INVITE in Message 8, giving the SIP client the necessary information where to send the audio.

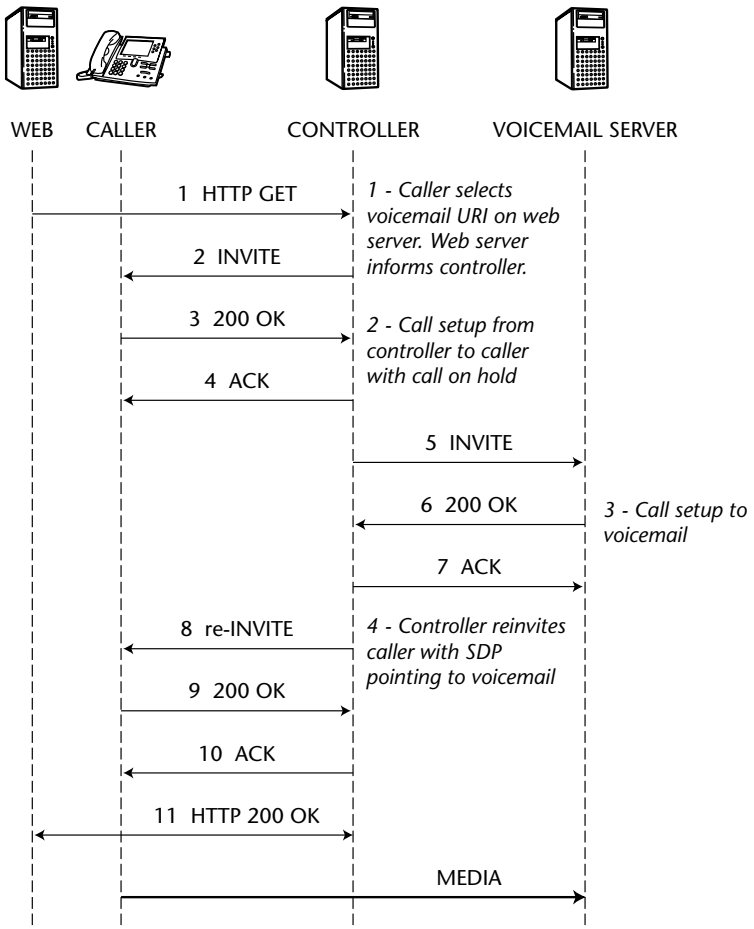


Figure 19.4 Call flow for voicemail

Collecting DTMF Digits

Figure 19.5 shows the basic call flows for the plain collecting of dual-tone multi-frequency digits, without voice recognition. We will discuss how SIP third-party call control is used for this application.

The initial `INVITE` (Message 1 in Figure 19.5) from the caller is directed to the service controller. The Request-URI in the `INVITE` message identifies this service, so various SIP proxy servers in the network (not shown here) know to route the call to the controller.

The controller first forwards the `INVITE` to the DTMF collector (Message 2) with no SDP body. This creates an initial media stream "on hold." The DTMF collector answers with its own SDP body in the reply `200 OK` (Message 3). The controller uses the reply (Message 3) to capture the data in the SDP body from the DTMF collector. It then proxies the call to the desired called party in Message 5 and gets, in return, a `200 OK` (Message 6) in case of success. The response to the caller (Message 7) has the following form:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 100.101.102.103;branch=z9hG4bK7d
To: User A <sip:UserA@here.com>;tag=3422
From: User B <sip:UserB@there.com>;tag=81211
Call-ID: a5-32-43-12@100.101.102.103
CSeq: 1 INVITE
Contact: <sip:UserB@pc.there.com>
Content-Type: application/sdp
Content-Length: ...

v=0
o=UserA 289375749 289375749 IN IP5 110.111.112.113
s=-
c=IN IP4 110.111.112.113
t=0 0
m=audio 5004 RTP/AVP 0
```

After Message 9 in Figure 19.5, the caller and called party can communicate. Possible IP-PSTN VoIP gateways are not shown for clarity of the SIP call flows on the IP side.

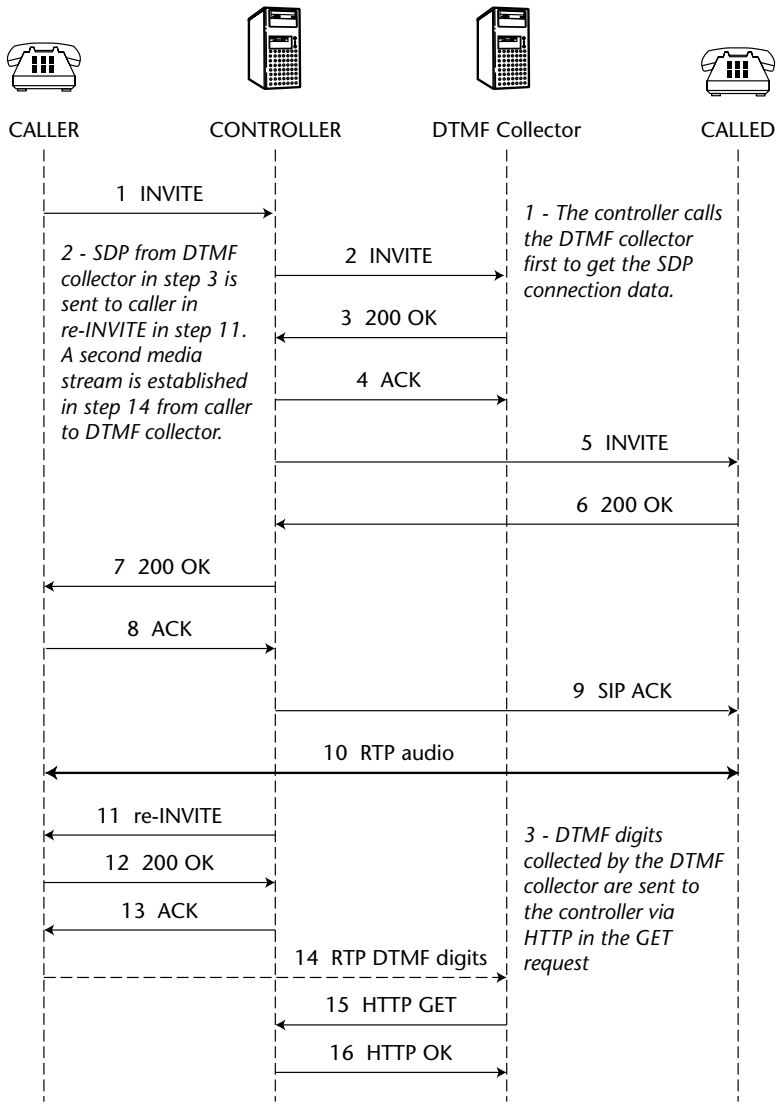


Figure 19.5 Call flow for collecting DTMF digits

The controller then initiates a re-INVITE (Message 11) to instruct the caller's UA in the re-INVITE message where to direct the DTMF media stream using the SDP connection data to the DTMF collector acquired in Message 3. This re-INVITE has the following form:

```
INVITE sip:UserB@there.com SIP/2.0
Via: SIP/2.0/UDP 100.101.102.103;branch=z9hG4bK7d
To: User B <sip:UserB@there.com>
From: User A <sip:UserA@here.com>;tag=19023023
Call-ID: a5-32-43-12-77@100.101.102.103
Max-Forwards: 70
CSeq: 1 INVITE
Contact: s<ip:UserB@pc.here.com>
Content-Type: application/sdp
Content-Length: ...

v=0
o=UserB 289375749 289375749 IN IP5 100.101.102.103
s=-
c=IN IP4 100.101.102.103
t=0 0
m=audio 5004 RTP/AVP 0
m=audio 53000 RTP/AVP 96
c=IN IP4 200.201.202.203
a=rtpmap:96 telephone-event
```

Note that this SDP now has a second media m= line for the DTMF digit transport with a new connection c= line with the IP address of the DTMF digit collector. The caller can now send DTMF digits in mid-call to the digit collector, since it knows the connection data to the DTMF controller.

The called party may instruct the caller to input data using the telephone keypad. The resulting DTMF digits are captured by the DTMF collector and sent to the controller in the HTTP GET message (15).

Plain DTMF service is useful for simple applications such as two-stage dialing, where the user first dials an access number for the respective service, gets a prompt tone, and then dials an identification such as the calling card number. A new dial tone invites the user then to dial the phone number. As we will see, DTMF digits can also be collected by more complex interactive voice response systems.

Interactive Voice Response System

State-of-the-art interactive voice (IVR) systems can be implemented with voice recognition and voice prompts generated using document pages marked up with the Voice Extensible Markup Language (VoiceXML). Figure 19.6 shows the call flow example for IVR service.

The service starts with an IVR exchange to determine the wishes of the caller. The controller, therefore, first proxies the call to the IVR server, so the caller can interact directly with the IVR server. As in the previous example, the initial `INVITE` message (1) from the caller has the Request-URI pointing to the controller for this particular service.

After the establishment of the media stream, the IVR will generate a voice prompt to the caller, along the line of "Welcome to our <*name*> service! Please speak your ID." The answer from the caller is transformed from speech to text and returned in Message 6 of Figure 19.6, HTTP `GET`, to the controller. The next VoiceXML script is sent from the controller in the HTTP `200 OK` (Message 7) to further prompt the caller for information regarding his or her request. After the IVR process comes to an end, the last message HTTP `200 OK` (Message 9) carries an empty VoiceXML script. The call to the IVR is terminated with a `BYE` (Message 10), and the call is forwarded to some other destination with the `INVITE` in Message 12.

Interactive voice response systems based on VoiceXML technology can support several features for voice services:

- Text to speech (synthesized speech)
- Output of audio files
- Voice recognition
- DTMF input
- Recording of spoken input

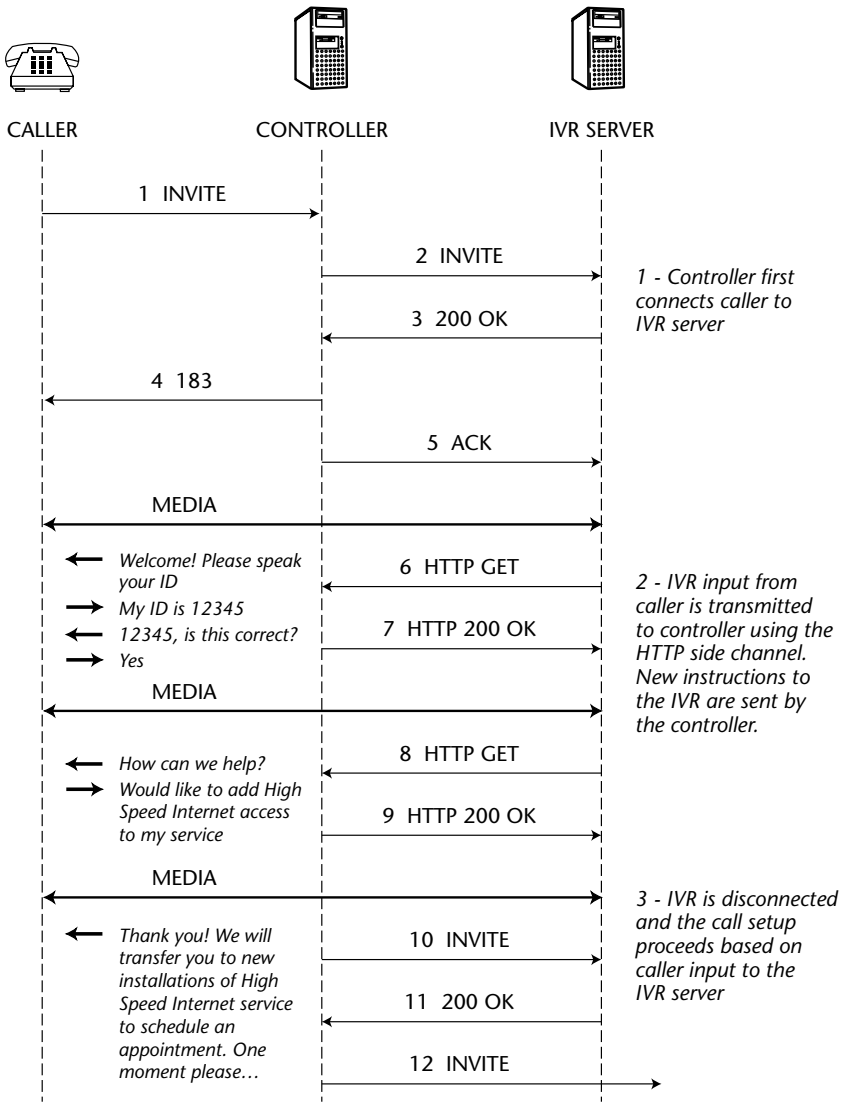


Figure 19.6 Call flow example for interactive voice response service

VoiceXML servers also have some telephony features, such as call transfer and disconnect, but these may not always be necessary in the presence of a service controller as discussed here. The following example reproduces a sample dialog from the VoiceXML specification [5] that shows the power of IVRs using VoiceXML:

```
Computer: Welcome to the weather information service. What state?
Human: Help
Computer Please speak the state for which you want the weather.
Human: Georgia
Computer What City?
Human:  Tblisi
Computer I do not understand what you said. What city?
Human: Macon
Computer: The conditions in Macon, Georgia are sunny and clear at 11
AM...
```

Scheduled Conference Service

A large variety of conference types are possible on the Internet, from spontaneous initiated conferences using presence, to telecom-type scheduled conferences. However, for most types of network-based conferences, a mixing voice bridge is necessary, such as discussed in Chapter 14, "SIP Conferencing." Therefore, it makes sense to separate the conference-scheduling servers from the voice-mixing bridge, since they are so very different in functionality and technology. Figure 19.7 shows an example of the call flow for a scheduled conference using separate scheduling and mixing servers. In this example, we assume that the scheduling server is also the controller.

The conference is scheduled and set up on the Web server, which, in turn, informs the scheduler using an HTTP `POST` message (1). The controller confirms the conference is possible and will be scheduled in the `200 OK` message (2) to the Web server. E-mail or some other means can also be used to inform the users of the scheduled conference.

At the scheduled time, the controller will connect the users successively to the voice-mixing bridge. Only two users, A and B, are shown here for simplicity, since all additional users would have the same call flows for call setup with the conference bridge. Note that an alternative service would be the controller could call the participants A and B and use third-party call control to connect them to the mixing bridge.

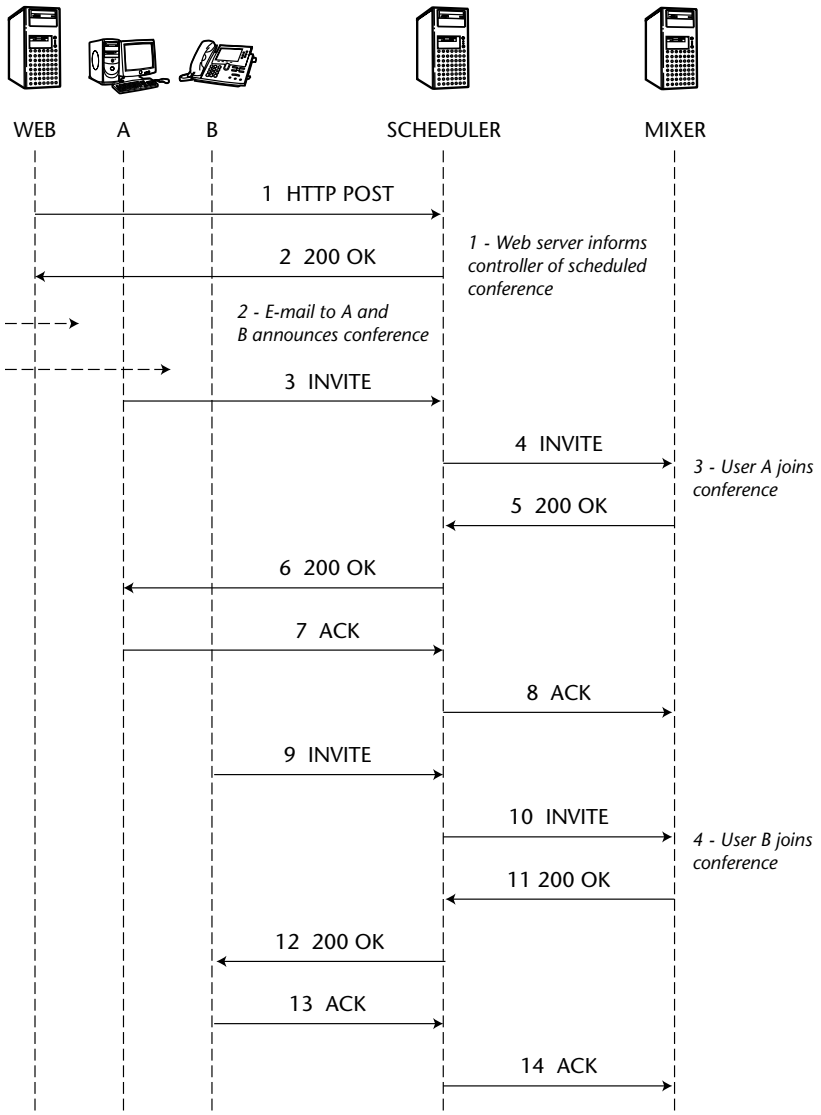


Figure 19.7 Call flows for a scheduled conference.

Summary

IP telephony gateways decomposition using master/slave protocols such as MGCP, MEGCO, or H.248 require specific packages in "softswitches" that are, by all measure, vendor-specific and may change by release versions. Close, proprietary coupling between components have limitations for nontelephony services. They also have the disadvantage of proprietary bundling and introduce added complexity for network operators.

By contrast, the component server architecture allows interaction between large numbers of loosely coupled, specialized servers across the Net. The component server architecture can provide access to all services using the Web, e-mail, and voice, relying only on the basic standard Internet protocols HTTP, SMTP, SIP and RTP. IVR, or VoiceXML service call flows are straightforward using third-party call control to direct incoming calls to the appropriate servers.

References

[1] *McGraw-Hill Series on The Intelligent Network Standards, Their Application to Services*, I. Faynberg, L. Gabuzda, M. Kaplan, and N. Shah. McGraw-Hill, New York, 1997.

[2] "Gateway Control Protocol Version 1" C. Groves et al. RFC 3525. IETF, June 2003.

[3] " An Application Server Component Architecture for SIP" J. Rosenberg at al. Internet Draft, IETF, March 2001.

[4] "Best Current Practices for Third-Party Call Control (3pcc) in SIP" J. Rosenberg et al. RFC 3725, IETF, April 2004.

[5] "Voice Extensible Markup Language (VoiceXMLTM) Version 1.0," The World Wide Web Consortium (W3C), May 2000. `www.w3.org/TR/voicexml`.

[6] "Control of Service Context using SIP Request-URI" B. Campbell and R. Sparks. RFC 3087, IETF, April 2001.

[7] "Session Initiation Protocol (SIP) URIs for Applications such as Voicemail and Interactive Voice Response (IVR)" C. Jennings et al. IETF RFC 4458, April 2006.

CHAPTER

20

Peer-to-Peer SIP

Peer-to-peer (P2P) networks are a more recent innovation on the Internet. As mentioned in Chapter 18, "Quality of Service for Real-Time Internet Communications," P2P traffic dominates the Internet traffic at present. "Peer-to-peer computing could usher in the next generation of the Internet, much as we saw Mosaic usher in the last era" [1].

NOTE **Mosaic was the first popular browser that made the Web accessible to millions of users.**

The history of P2P applications on the Internet started in 1999 with such applications as Napster that used a central index server. Other innovations that came later, such as Kazaa managed to function without any central servers so as to avoid legal and technical problems. Finally, mixed architectures were developed that combined the advantages of the fully distributed and decentralized architecture with some server-like functions—so called "supernodes" in a hybrid P2P mode.

P2P was also adopted for Internet communications. The most famous hybrid P2P network is at present Skype. Skype is the clear leader on VoIP, IM, and presence with, as of this writing, close to 80 million users and more than 5 million users online during busy hours.

The history of P2P networks shows an interesting pattern of innovation on the Internet. Innovators develop extremely popular applications such as file sharing, VoIP, and IM. Skype is a good example. The attention they get from threatened businesses models sparks research in academia that tries to explain and improve on such innovations. Finally, as an understanding of the innovations develops, standards bodies come into play, and even the threatened businesses (such as content development studios and telecommunications providers) start to understand the advantages of P2P networks.

P2P networks are envisaged not only for VoIP but also for other applications, such as file sharing (the origin of modern P2P systems) and media distribution in content distribution networks (CDN), sharing of computing power, application-level multicasting, mobility management, peering between BGP routers, and last, but not least, as a replacement of the DNS under certain circumstances.

NOTE David Bryan has made the remark that "P2P will do to VoIP what VoIP has done to the PSTN." For more in-depth information on P2P in general and P2P SIP specifically, consult David Bryan's Web site at `www.p2psip.org`**.**

Definitions for P2P Networks

This section examines some P2P fundamentals, including the following:

- Overlay networks
- Peer-to-Peer networks
- Distributed Hash Tables (DHTs)

Overlay Networks

An *overlay network* is a computer network built on top of another IP computer network, or on top of the Internet. Figure 20.1 shows an example of an overlay network.

Note that the overlay network resides at the edge of the IP network and is completely ignorant of the underlying IP network, as well as any services residing in the underlying network. The nodes of the overlay network use only the IP addresses from the underlying network. *Both discovery and routing is done on the application layer at the edge only.* No DNS is required for discovery.

Overlay networks have the remarkable characteristic of being self-organizing, as we will illustrate in the section, "The Chord Protocol," later in this chapter.

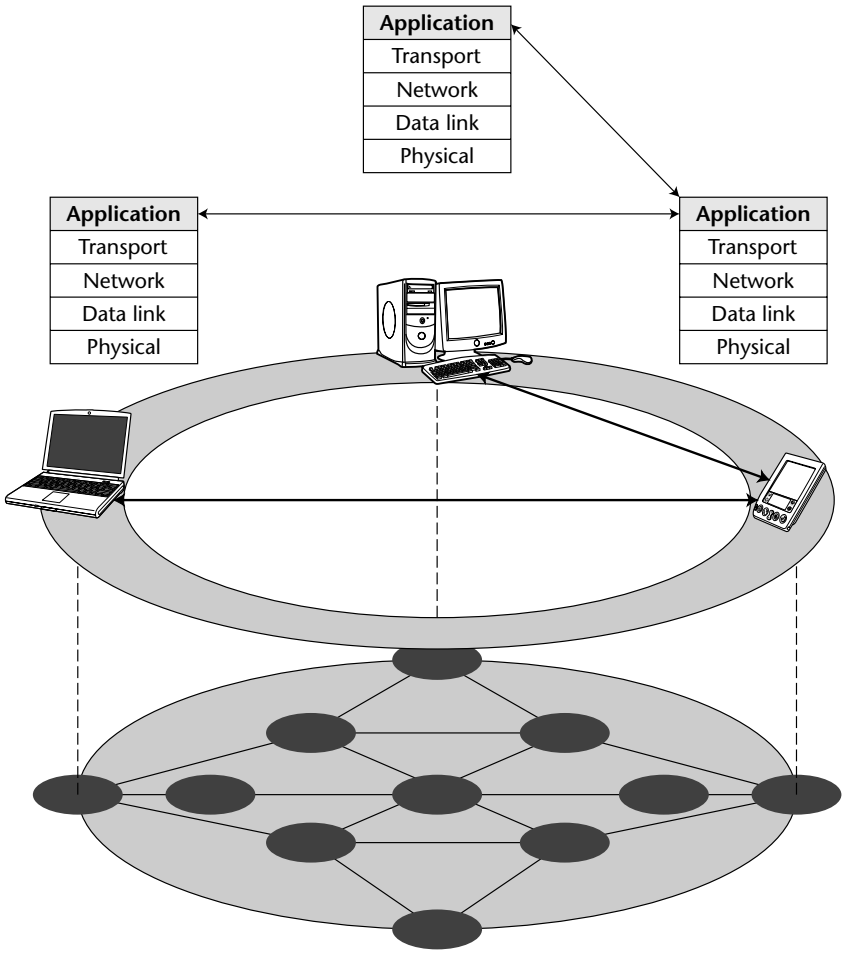


Figure 20.1 Example of an overlay network

Peer-to-Peer Networks

P2P networks share the computing resources available in the network and also the bandwidth of the peer nodes. P2P networks have a number of common characteristics, such as significant autonomy from central servers [2] and intermittent connectivity of most peer nodes at the edge. Peer nodes frequently join and depart from the network.

Not all nodes at the edge must be equal in an overlay network. Some P2P networks use the so-called "supernodes" that have more permanent connectivity, and also possibly more bandwidth than other peers. Supernodes can perform useful functions that are normally associated with servers on the Internet (such as the bootstrapping of nodes that join the overlay network), except that supernodes very closely resemble all other peer nodes. Also, contrary to SIP registrars and proxies, supernodes are also part of the self-organizing P2P network.

A P2P network with supernodes can be considered a two-level P2P network. Recent research has shown that it is possible to build multilevel hierarchical networks to reduce the discovery time in very large P2P networks [3].

P2P networks can scale from small enterprise networks (such as the P2P PBX) to Internet-sized networks with millions of users.

P2P embodies the virtues of the end-to-end principles of the Internet architecture, in spite of its evolution and some later trends that are contrary to the e2e principle [4]. Even the first-ever RFC on host software [5] is based on P2P computing. It is fair to say that P2P computing embodies the best principles of the Internet, as it was originally designed.

Distributed Hash Tables (DHTs)

Since a P2P network is, by definition, highly distributed, any P2P network must be able to store information across the overlay network and retrieve it. One approach from the academic community, known as Distributed Hash Tables (DHTs), has been successfully applied to this problem. DHTs offer the promise of highly scalable, low-latency search and retrieval of data. A mathematical function known as a *hash* is used to turn a variable-length string into a fixed-length number. A good hashing function will have a minimal probability of collisions, that is, two different strings producing the same hash output. Examples of hash functions include MD5 (Message Digest 5) used by SIP in HTTP Digest authentication and SHA-1 (Secure Hash Algorithm 1), as discussed in Chapter 9, "SIP Security."

Nodes in a P2P network can be found by using a key that is similar to an IP address in an IP network, except that the key acts as an identifier for the peer node. The key can be the hash of the content description in file-sharing P2P networks or simply the hash of the IP address in a P2P communication network.

Peer nodes keep a table of a limited number of known neighbor nodes in the form of a hash table for routing at the application layer. The assembly of hash tables in peer nodes is the DHT. *Consistent hashing* is a scheme with the property that the hash table is modified only by the joining or leaving of a small number of neighbor nodes, and not all hash tables on the P2P network need to be updated. Consistent hashing is used by the Chord protocol, as will be shown later in this chapter.

In a DHT, the hash function is used to determine which part of the P2P overlay network is responsible for a particular piece of data. When the data is to be stored in the overlay network, the hash function is used to determine where it is stored. When the data is to be retrieved from the overlay network, the same hash function is used again. All nodes in the P2P network use the same hash function, so they all distribute the network data in the same way. Since the value of the hash is essentially random, the information is evenly distributed over the network.

The hash function is used on the key for the data. For a SIP P2P network, the key might be the AOR URI of a particular user. The data to be stored about this user might be the current registration location, the caller preferences, or voicemail or other stored messages for the user.

If the information stored in the overlay network is public information, such as registration data, then any node in the network can query and retrieve the information. If the information is private, then only a particular node or set of nodes will be able to retrieve or decrypt the information, for example, a voicemail message.

DHTs can be used for a large number of applications, some of them shown in Figure 20.2.

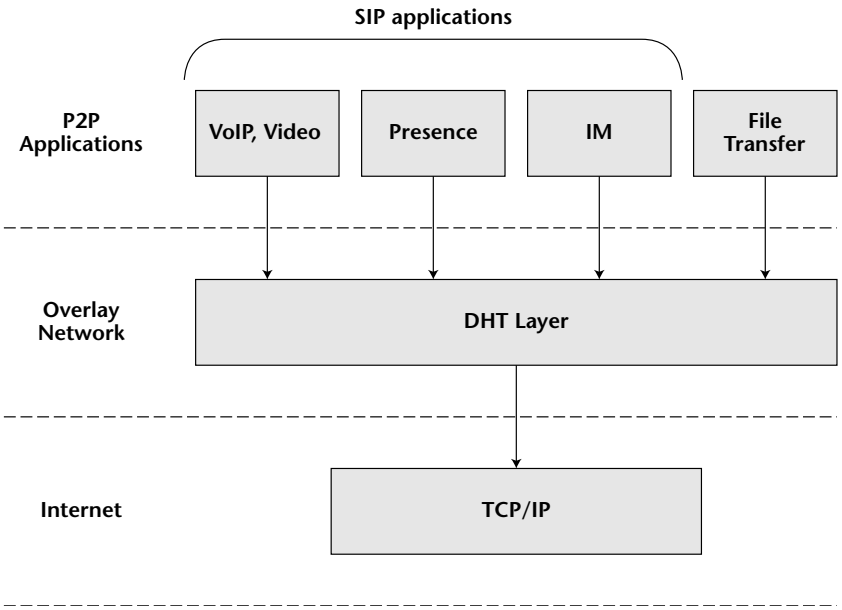


Figure 20.2 P2P network applications

Characteristics of P2P Computing

The importance of P2P computing can be better understood by looking at its characteristics:

- P2P computing is decentralized and, thus, resilient by nature.
- Self-organization is a native property of P2P computing.
- P2P networks have high usability. Their operation and use require no understanding by users and no maintenance by network operations staff.
- Ad hoc networks can be set up with absolute minimal configuration.
- Performance is a function of the bandwidth and computing power of the peer nodes. Performance can be optimized using:
  - Caching
  - Smart routing algorithms (such as in Chord)
- P2P networks are scalable from small home and enterprise networks to global Internet scale.
- Secure P2P networks can be built and are easier to audit for possible vulnerabilities because of limited complexity.
- The cost of ownership is minimal because of the lack of network infrastructure and also because of the lack of operations costs for service infrastructure. Later in this chapter, we will discuss the disruption this causes to the legacy telecom and VoIP industries.

Security of P2P Networks

P2P networks have two broad security aspects:

1. It is harder to attack P2P nodes because of the distributed nature of P2P. Denial of Service (DOS) attacks are most often performed by concentrating the power of many compromised computers on one single target—this approach does not work for P2P. Secure overlay systems (SOS) [6] can be built by using system properties, such as the mobility of peers (harder to impersonate and to find the target) and the large number of targets. Other techniques for secure overlays include secret proxy servers that are known only to a small number of users. The secret servers are actually another network overlay, called the *secure overlay access points*.

2. A significant exposure specific to P2P is malicious peer nodes that could inject false routing information in the peer discovery process. One way to counter this is to store data in parts across a number of nodes, and in multiple copies. This way, collusion of a number randomly selected nodes is required to inject false information into the overlay network.

Conventional security procedures can be applied to P2P networks, including the following:

- Cryptographic key exchange
- Digital digests (hashing)
- Encryption
- Signatures

Other tools developed specifically for P2P computing include the following:

- Sandboxing to protect against malicious code
- Reputation and accountability
- Digital rights management for content distribution

NAT and firewall traversal has long been a well-honed skill mastered by the inventors of various P2P networks to reach the many millions of their users. It is actually reported that blocking P2P in enterprise network is not a trivial job. Skype has been reported to have the agility to change ports on a dynamic base to traverse NATs and firewalls. Techniques similar to ICE [7] for SIP have been developed, although the design details are different.

The Chord Protocol

Chord [8] is an example of a DHT algorithm that has been widely studied. Chord uses a ring architecture, as shown in Figure 20.3. A node joining the ring determines its position in the ring and then inserts itself between two neighbors. A node in a Chord network keeps track of its predecessor node and successor node, and makes periodic checks to ensure that these nodes are still available. Should one of these nodes go away or loose connectivity, the node will determine its new predecessor or successor node. In addition to these two nodes, the node also maintains a "finger table" of other nodes in the ring. The number of nodes in this finger table grows with the logarithm of the total number of nodes in the network, allowing a very large network to be spanned by a small finger table. When searching the overlay network, the finger table allows the node to quickly jump to the part of the ring that is responsible for the data. This allows the Chord ring to scale and grow very large without resulting in very long lookup latency.

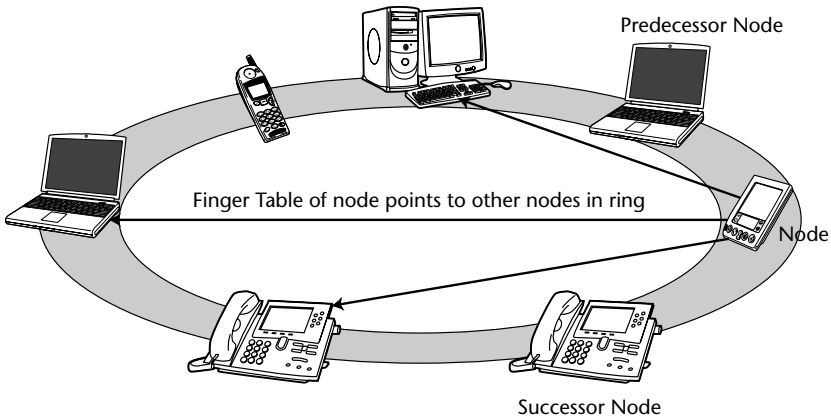


Figure 20.3 Chord Ring

Areas of applicability of Chord to SIP include a distributed location service [9], registration [17], DNS [10], and NAT and firewall relay traversal. These will be discussed in the following sections.

P2P SIP

Client-server (CS) SIP deploys servers (such as SIP registrars, SIP proxies, various servers for component services as described in Chapter 19) for voicemail, third-party call control, IVR, conferencing, and so on, and also STUN and TURN servers for NAT traversal. The SIP standard does not mandate these servers and mentions that they are optional [11]. Once a SIP UA knows the location of the desired parties, all SIP call control can happen in the P2P mode as well, as discussed in [12], where it is shown how PBX and PSTN telephony functions can be implemented using P2P SIP call control.

Standards for P2P SIP discovery and call control have not yet been developed as of this writing, but the principles are fairly well understood, and several P2P SIP implementations have been reported in research and academia, as well as commercial products [13], [14], [15], [16].

P2P SIP can be best understood by a simple example and by comparing it with client-server SIP using the trapezoid model in RFC 3261, as described in Chapter 6, "SIP Overview."

Figure 20.4a shows the discovery and call routing process in the trapezoid model.

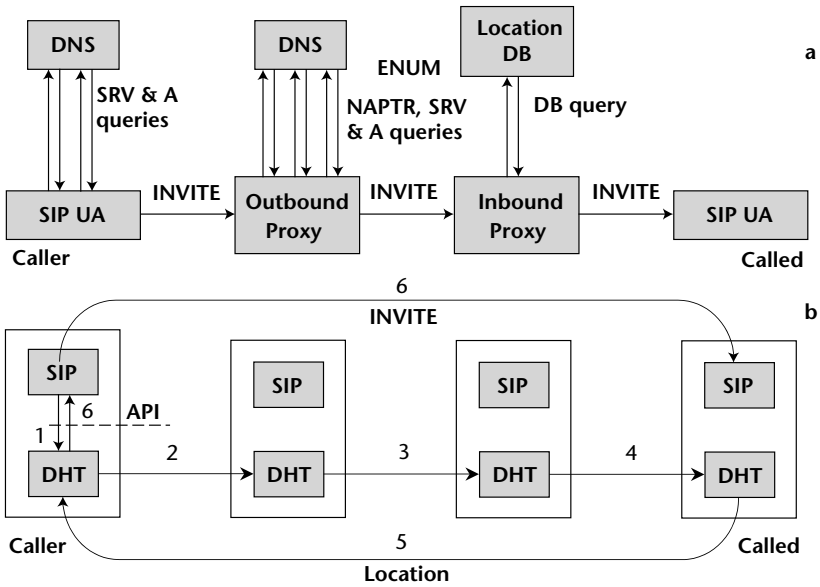


Figure 20.4 Discovery and call routing in (a) the trapezoid model for CS SIP and (b) in the P2P SIP model using DHTs in an overlay network

CS SIP Model

The caller in Figure 20.4a must first determine the outgoing SIP proxy (we don't show the registrar here for simplicity) by issuing first a DNS SRV request to get a list of outgoing SIP proxies. After deciding which proxy in the list to use (for load balancing), a second DNS A record query will return the IP address of the chosen outbound SIP proxy. The SIP UA of the caller will now send an `INVITE` to the outbound proxy.

The outbound proxy will now make three or four DNS queries to find the IP address of the inbound proxy:

1. An ENUM NAPTR query the URI of the SIP service
2. An SRV query to determine a list of incoming SIP proxies
3. An A query to determine the IP address of the incoming SIP proxy

The outbound proxy can now forward the `INVITE` to the inbound proxy.

The inbound proxy will check its own location database (DB) and send the invite to the calling party. This concludes the discovery and routing process in the CS SIP example.

P2P SIP Model

The caller queries the DHT layer (step 1) in the overlay network to determine the location of the peer node for the called party. The DHT layer routes the location query to the target peer node (steps 2, 3, and 4), which will answer and provide its IP address to the SIP UA of the caller (step 5). In step 6, the DHT layer returns the IP address to the SIP layer of the peer node of the caller. The SIP UA of the caller can now send a direct `INVITE` to the called peer node. This concludes the P2P discovery and call setup.

Note in Figure 20.4b that the SIP layer and the DHT layer can be completely independent of each other and emerging IETF standards work allows for future choices of the DHT layer that may not be the Chord protocol, so as to have a flexible architecture that can keep up with current research in P2P networks. The interface between the SP layer and the DHT layer, shown as a dotted line in the peer node of the caller, is an API that may be standardized. Distributed computing on Planet Lab (`http://www.planet-lab.org/`) includes the OpenDHT Layer (`http://opendht.org/`). The OpenDHT runs Bamboo and can be used for P2P SIP Communications.

Use Cases for P2P SIP

Several descriptions for use cases of P2P communications have been made, the latest with the intent to develop requirements for a standards-based approach in the IETF [17]. Such use cases should be considered only examples based on present knowledge, since you cannot predict innovations.

- *Public P2P communication service provider*—Skype is an excellent example of a prestandard public P2P communication service. Another example is `Damaka.com` as a P2P SIP service provider. Service providers must by necessity use a central login server that also performs the authentication, authorization, and accounting (AAA) functions to manage their customer base. Note the AAA server has no role whatsoever in the discovery and call setup for communications. Public P2P communication service providers have the advantage of lowest cost, because of the lack of any VoIP infrastructure, as will be discussed in the next section.
- *Open global P2P communications*—Anyone on the Internet can enjoy rich multimedia communications. The only requirement is to have standards-compliant P2P SIP user devices. The security aspects for global open P2P communication are, however, not clear at present. Endpoint based security such as PGP or ZRTP [18] may be a possible approach.
- *Multimedia consumer devices*—As the number of multimedia consumer devices proliferate in the home, the self-organization of P2P networking is a necessary ingredient for acceptance by consumers. Current protocols such as UPnP have not taken off, despite the critical need.

- *Security-sensitive organization*—Security conscious small businesses or organizations of various types may not feel comfortable using the hosted "Centrex" type of communications that are marketed by telephone companies or carriers. P2P removes the incentive to reduce cost that is invoked for hosted services and allows self-contained, locally resident secure communications.
- *Limited or interrupted Internet connectivity*—Communities can be isolated from the Internet (such as on the battlefield or during natural or man-made disasters). P2P communications will still function, as long as there is a local, most-basic IP network available. Meshed wireless networks can provide the IP infrastructure.
- *Ad hoc groups*—Groups of people that assemble for meetings or events can set up an instant communication network. Here again, meshed wireless networks can provide the IP infrastructure.
- *Serverless PBX*—Serverless IP PBXs were the first commercial application of enterprise P2P communications. We expect serverless multimedia communications for the enterprise to be a successor to the serverless IP PBX that can support only voice. Enterprise systems may require a login server, however, just like public P2P service providers.
- *P2P for self-organizing SIP proxies*—Even CS SIP-based communication systems can reduce their operational costs by deploying a self-organizing P2P cloud of SIP registrars, SIP proxies, and component services, as described in Chapter 19. Self-organizing SIP proxies can also use DNS as described for the commercial SIP Thor product [19]. DNS can be used for self-organizing systems of small scale. A SIP server cloud is a small system, though it can serve millions of users.

Disruption of the VoIP Infrastructure Model

The original concept of the Internet regarding the end-to-end principle and best-effort quality of service has found a triumphant vindication in the market with Skype, by far the dominant VoIP, presence, IM, and video service worldwide. As mentioned, Skype can be rightly considered as a prestandard implementation of P2P Internet communications. An analysis of Skype can be found in [20]. The success of P2P Internet communications can be explained mainly by a number of factors [21]:

- Minimal or no VoIP infrastructure capital cost for such items as:
  - SIP registrar and proxy servers
  - Session border controllers

- Softswitches
- Media servers
- Network elements for QoS
- Policy servers for QoS, for classes of users and applications
- Voice-quality-monitoring network probes
- Network management systems for all of these
- Information Technology (IT) systems for all the above
- Network engineering and integration costs for VoIP.

By contrast, the software applications in the peer nodes actually replace the VoIP, IM, and so on for ßnetwork infrastructure.

- Minimal or no operations costs because of the self-organizing nature of P2P. Payroll in existing telecom operations increases with the number of systems and network elements, and also with the complexity of the infrastructure. In P2P systems, software upgrades for downloading by users replaces the operations cost for VoIP networks.
- Minimal information technology (IT) costs. IT in telecom is often more expensive that the voice network itself that IT has to support. The only IT system for P2P is the customer authentication and logon server, fundamentally not very different from any other e-commerce system.
- No service level agreements with customers and with other connected networks that require a significant legal staff and a business development staff.
- The only residual cost for P2P Internet communications are the gateway services to the PSTN and to mobile 2G and 3G networks. Large costs are incurred in these networks for the expensive accounting systems to support pricing plans, promotions, settlements, and so on.

Summary

The explosion of Internet P2P computing on the Internet has also produced Internet P2P communications. The prestandard P2P system Skype is, by far, the biggest IM and VoIP service provider on a global scale.

P2P overlay networks are self-organizing systems and have significant advantages compared to the customary client-server architecture. P2P communications require no VoIP infrastructure. P2P communications based on SIP can use the Bamboo protocol for Distributed Hash Tables and have most of the

same features as provided by server farms currently deployed in VoIP networks. P2P SIP usage scenarios include applications that cannot easily be implemented using CS SIP.

We explain the reasons for the disruption of the VoIP industry by P2P communications. This area is likely to be explored in bodies such as the IETF in the coming years.

References

[1] "Background and History of peer-to-peer" by CacheLogic, 2005. `http://cachelogic.com/p2p/p2phistory.php`.

[2] "P2P Systems" by K. Ross and D. Rubenstein. Tutorial from NY Polytechnic University and Columbia University, 2005.

[3] "Hierarchical Peer-to-peer Systems" by L. G. Erice et al. Polytechnic University, Brooklyn, NY, and Institute EURECOM Sofia Antipolis, France, 2005.

[4] "The Rise of the Middle and the Future of End-to-End: Reflections on the Evolution of the Internet Architecture" by J. Kempf and R. Austein. RFC 3724. IETF, March 2004.

[5] "Host Software" by S. Crocker. RFC 1. UCLA, April 1969.

[6] "SOS: Secure Overlay Services" by A. Keromytis et al. Columbia University Technical Report EE200415-1. February, 2002.

[7] "Interactive Connectivity Establishment (ICE)" by J. Rosenberg. Internet Draft, IETF, October 2005, work in progress.

[8] "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications" by I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, ACM SIGCOMM 2001, San Diego, CA August 2001, pp. 149–160.

[9] "SIP, P2P, and Internet Communications" by A. Johnston. Internet Draft, IETF, work in progress, March 2005.

[10] "Serving DNS using a Peer-to-Peer Lookup Service" by R. Cox, A. Muthitacharoen, and R. Morris. First International Workshop on Peer-to-Peer Systems (Cambridge, MA, March 2002).

[11] "Requirements for SIP-based Peer-to-Peer Internet Telephony" by S. Baset. Internet Draft, IETF, October 2005.

[12] "A Call Control and Multi-party usage framework for SIP" by R. Mahy et al. Internet Draft, IETF, March 2003. `www.softarmor.com/wgdb/docs/draft-ietf-sipping-cc-framework-02.html`.

[13] "Peer-to-Peer Internet Telephony using SIP" by Kundan Singh and Henning Schulzrinne, Columbia University Technical Report CUCS-044-04, New York, October 2004.

[14] "SOSIMPLE: A SIP/SIMPLE Base P2P VoIP and IM System" by D. Bryan and B. Lowekamp. William and Mary University, Williamsburg, VA, 2005.

[15] Nimcat/Avaya Networks P2P Enterprise nimX Phone System. http://nimcatnetworks.com.
[16] PEERIO Server-free Telephony System. http://peerio.com.
[17] "Use Cases for Peer-to-Peer SIP" by D. A. Bryan et al. Internet Draft, IETF and William and Mary University, Williamsburg, VA, November 2005.
[18] "ZRTP: Extensions to RTP for Diffie-Hellman Key Agreement for SRTP" by P. Zimmermann and A. Johnston. IETF Internet Draft, work in progress, February 2006.
[19] "SIP Thor: Scalable SIP solutions for millions of subscribers." AG Projects, 2006. http://ag-projects.com.
[20] "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol" by S. Baset and H. Schulzrinne. Columbia University, September 2004.
[21] "SIP Beyond VoIP" by H. Sinnreich, A. Johnston and R. Sparks. VON Publishing, 2005.

CHAPTER 21

Conclusions and Future Directions

At the time of the first edition of this book, the SIP standard was RFC 2543, which was replaced in June 2002 by RFC 3261. Numerous other RFCs have been published since with various extensions to SIP. The related working groups (such as SIMPLE for IM and presence, SIPPING for applications, ENUM, AVT, and so on) have used RFC 3261 as a basis for numerous extensions to SIP and various applications.

The growth in IETF standards documents for Internet communications is rather formidable and reflects the work being done to migrate all real-time communications to the Internet. This growth is illustrated in Figure 21.1. The large volume of IETF documents is the price paid for making SIP the universally accepted standard by most wireline and wireless service providers. Most IM service providers use SIP and SIMPLE as well. SIP is the common denominator for all IM companies, since even IM services such as Skype or Google are using SIP to connect to the rest of the world.

The IETF has not been very consequent with upholding a single standard for IM as is the case of SIP for voice and multimedia. RFC 3920 and RFC 3921 describe the XMPP protocol for IM, which is not applicable as well, however, for voice and multimedia, as SIP is.

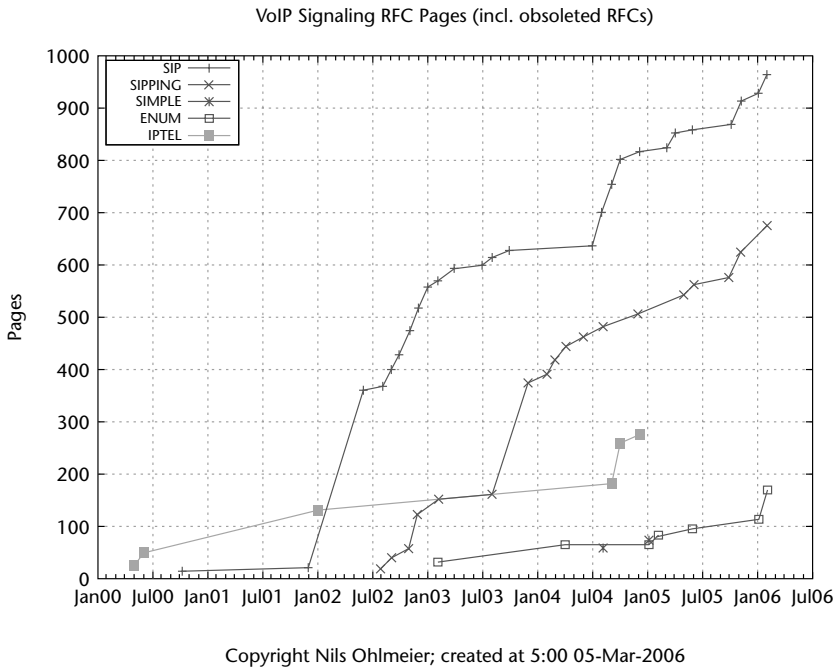


Figure 21.1 The growth of SIP-related IETF standards (http://rfc3261.net).

The core SIP protocol is probably more than 95 percent fully developed and stable, and we may expect only minor extensions in the future that will not affect the SIP core protocol.

What is the SIP core protocol? For simplicity, and for lack of a widely accepted definition, you may assume that the SIP core protocol includes everything described in this book and in the quoted references. If it's not in the quoted references, we assume that it is probably not part of the SIP core.

Many SIP extensions have been proposed and accepted in IETF informational RFCs, to meet the business needs of certain types of service providers (such as cable, telephone companies, and especially 3G mobile providers). Such extensions are part of the so-called P- extensions, where P- stands for "preliminary," "private," or "proprietary." RFC 2427 specifies that, for P- extensions, "it is valid to allow for the development of SIP extensions that . . . are private or proprietary in nature, because a characteristic motivating them is usage that is known not to fit the Internet architecture for SIP." A large number of the IETF documents counted in Figure 21.1 are of this nature.

Short Term Challenges

The short-term challenges for SIP implementations in the market are described in our companion book, *SIP Beyond VoIP* [1]. We provide here only a short summary.

- NAT traversal using ICE as described in Chapter 10, "NAT and Firewall Traversal." The importance of stabilizing the emerging standard for ICE cannot be overemphasized, since without ICE, ubiquitous interdomain SIP-based communications will not be trivial to set up. Part of the NAT traversal challenge is finalizing the work on the Globally Routing User Agent URI (GRUU) [2].
- Abandoning the fixation on telephony, "PSTN over IP" style islands that cannot connect directly over the Internet.
- High-quality voice endpoints with Internet codecs, echo control, and so on. See Chapter 18, "Quality of Service for Real-Time Internet Communications."
- SIMPLE standards based presence and IM.
- Emergency services using the Internet.
- Internet communications for the disabled.
- Identity and security for interdomain communications.

Future Services: The Internet Is the Service

Paul McFedries says [3], "The combination of ubiquity and necessity makes the Net analogous to an atmosphere." In this light, all real-time communications mentioned in this book are just applications living in the "atmosphere" that is the Internet. An example for such services is the interdomain presence service [4] from Tello, which helps professionals to contact their business correspondents in the most convenient and effective way.

It is not practical to enumerate (and even less possible to predict) all the various real communication services and applications on the Internet.

Still to Develop: Peer-to-Peer SIP Standards

The first edition of this book in 2001 did not predict the emergence of P2P and its dominance on the Internet. As discussed in Chapters 6 and 20, P2P SIP has the potential of completely disrupting the VoIP industry (just as Skype has already done) and to make obsolete most business models in the "traditional"

VoIP industry, for VoIP infrastructure vendors and service providers alike. The authors hope to have provided the reader with enough information and references for P2P SIP to start keeping abreast this new field of communication systems that are symmetric in nature, self-organizing, and distributed [5].

Prediction: The Long Road Ahead

Making abstraction of all the technology novelties, the dismemberment of the business models and networks in the telecom world is a huge economic disruption and will not take place without a long regulatory and political evolution, to be digested in the economy of most countries. As often noted by many adults, however, "Watch our children; how they communicate, play, and work using the Net."

Summary

Internet communications based on SIP provide a sheer inexhaustible source of multimedia communications and their integration with personal and business applications, entertainment, information and e-commerce. The replacement of the telephone networks with Internet communications has only just started.

Client-server communications based on SIP are mature, though the industry may need more time to catch up with the standards.

P2P communications and entertainment will be, however, the next disruption on the Internet, although many in the communications and VoIP industry are just becoming aware of P2P.

References

[1] "SIP Beyond VoIP" by H. Sinnreich, A. Johnston, and R. Sparks. VON Publishing LLC, 2005.

[2] "Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)" by J. Rosenberg. Internet Draft, IETF, October 2005, work in progress.

[3] "Cyberspace is Dead–well, the word anyway" *Wired Magazine*, February 2006, p. 39.

[4] "Instant Communication and Collaboration across Businesses, Applications and Devices. `http://tello.com`.

[5] "Survey of Research towards Robust Peer-to-Peer Search Methods" by J. Risson and T. Moors. Technical Report, University of New South Wales, September 2004.

Index

E

N

S

U