



The Essential Guide to
VoIP Implementation and Management



Written by:
John Q. Walker and Jeffrey T. Hicks

The Essential Guide to VoIP Implementation and Management

by John Q. Walker and Jeffrey T. Hicks
NetIQ Corporation

Introductory Chapter: VoIP Basics

| | |
|---|-----------|
| Why We Wrote this Book | 2 |
| Who Should Read this Book..... | 2 |
| How this Book Is Organized..... | 2 |
| Getting Started..... | 3 |
| In the Telephony Community | 4 |
| Telephony Standards..... | 5 |
| How the PSTN Works..... | 5 |
| PSTN Components..... | 6 |
| Understanding Voice Encoding..... | 6 |
| Understanding Switches..... | 7 |
| Understanding PBXs..... | 8 |
| Understanding Signaling..... | 9 |
| Understanding Telephones..... | 9 |
| In the Data Networking Community | 10 |
| How VoIP Works..... | 10 |
| Data Networking Standards..... | 11 |
| VoIP Components..... | 11 |
| Understanding Codecs..... | 11 |
| Understanding TCP/IP Protocols..... | 12 |
| Understanding VoIP Protocols..... | 15 |
| IP Telephony Servers and PBXs..... | 18 |
| VoIP Gateways and Routers..... | 19 |
| IP phones and Softphones..... | 20 |
| For More Information | 21 |
| About the Authors..... | 21 |

Introductory Chapter: VoIP Basics

This introductory chapter of our book, *“The Essential Guide to VoIP Implementation and Management,”* by John Q. Walker and Jeffrey T. Hicks of NetIQ Corporation, explains the audience and purpose of our book, predicts its contents, and discusses the basic terminology and concepts.

We’ll serialize this book, releasing a chapter a month for seven months. A revised, bound edition, to be published in summer 2002, will follow.

Why We Wrote this Book

We want the CIOs and IT managers who choose to deploy Voice over IP (VoIP) in their organizations to look good. There are lots of books with technical details about VoIP, but few explain in plain language how to make it run successfully in an enterprise.

We’ve designed the book to start with fairly simple concepts, with each chapter building on the knowledge gained from previous chapters. We’ve tried to make few assumptions about your background, other than expecting for some basic understanding of the two key terms in the title: “voice” telephone conversations and “IP” data networking.

This isn’t a technical manual about the inner workings of industry standards, such as H.323, G.711, and RTP. However, we will cover the implications of these standards for implementing and managing a VoIP system.

At the end of each chapter we’ve listed some articles, technical books, and Web sites we recommend.

Who Should Read this Book

We’re writing for managers of enterprise IT departments – and for their managers, up to the CIO level. We’re also writing for systems integrators who are embarking on the big effort to deploy voice on an IP network. And we’ve included the folks on the other side of the big knowledge gap we’re trying to bridge—very few professionals in the telephony community know much about the data-networking community, and vice versa.

How this Book Is Organized

Here is what we plan to discuss in each of the chapters:

- **Introduction: VoIP Basics**
 - How the telephony community views this technology
 - How the data networking community views this technology

- **Chapter One: Building a Business Case for VoIP**
 - The top benefits you'll receive from VoIP – and which benefits are more hype than reality
 - How to analyze your VoIP return on investment (ROI)
 - ROI for VoIP management
- **Chapter Two: Planning for VoIP**
 - Why is VoIP management critically important?
 - How to avoid common VoIP pitfalls through proper planning
 - Why most networks aren't ready for a VoIP deployment
 - The critical importance of a thorough testing plan
 - Pilot deployments: how big should you start, and when do you roll out?
 - A roadmap for your VoIP deployment
- **Chapter Three: Do It Yourself or Outsource?**
 - Some options for outsourcing
 - Criteria that determine whether you should roll out VoIP yourself or work with an integrator
 - What to look for in an integrator: important questions to ask
 - A methodology for “doing it yourself” – even if you're a systems integrator
- **Chapter Four: Quality of Service – the key to VoIP Nirvana?**
 - Indications that you need Quality of Service on your network
 - How Quality of Service can help voice and data to coexist peacefully
 - What forms of Quality of Service work best for VoIP?
- **Chapter Five: Ongoing VoIP Management**
 - Core components of a comprehensive VoIP management infrastructure.
 - What parts of the network need to be managed and monitored?
 - Maintaining call quality
 - Optimizing server performance and availability
- **Chapter Six: Establishing VoIP SLAs**
 - Some typical SLAs for VoIP
 - What metrics should an SLA be based upon?
 - What penalties should you build into your SLA?
 - What to look for in an SLA from your service provider

Getting Started

Even the acronym *VoIP* is an example of the rampant jargon you have to master to understand and deploy Voice over IP. There's lots of terminology to cover, from both the telephony and data networking communities. We'll use the right terminology throughout this book, but introduce and explain it in plain English.

First, let's start with some call fundamentals. A telephone call occurs in two stages:

- Setting up the telephone connection between the person making the call (the *caller*) and the person receiving the call (the *callee*).
 - Getting from one telephone to the other, through everything that's in the middle.
 - Committing the resources to that call, so that once you get it, you get to keep it, it's not unexpectedly terminated right in the middle.
 - Taking down the call when it's complete.
 - Billing someone for the call.
- The actual call itself
 - People or computers speak to one another for a certain amount of time.
 - Voice (audio) is translated into a format that can be sent over a network.

Each of these two stages has specialized equipment and a set of rules that guide its operation. Let's take a look at how telephone calls work in the telephony community and in the data-networking community.

In the Telephony Community

Telephony specialists approach communications technology from a background shaped by the traditional telephone network, the Public Switched Telephone Network (PSTN). The telephone service provided by the PSTN is called "plain old telephone service" (POTS). This "plain" type of telephone network we all take for granted uses "circuit-switched" connections, which means that when you make a call, you receive a dedicated circuit, from one telephone to the other, through everything that's in the middle. The typical dedicated circuit through the PSTN has evolved from a physical connection to a logical connection that involves many switches. When you speak into a phone, a microphone creates an analog transmission that's passed on the circuit through the network.

Decades of knowledge, experience, and innovation have allowed the public telephone network to achieve the quality and reliability that it has today. When you pick up a phone, you get a dial tone almost instantly. And when you dial a number, the destination phone starts ringing, usually within a few seconds. Can you even recall the last time your traditional telephone call was dropped by the network? Research shows that because the PSTN is so reliable, people are rarely willing to tolerate reduced-quality or dropped calls, and their tolerance usually comes only with additional convenience, such as the convenience provided by mobile phones.

The level of quality that's expected from the PSTN is sometimes referred to as "five-nines." This term means that the entire network must be available and functional for 99.999% of the time. If you apply this principle over the period of one year:

$$365 \text{ days} * 24 \text{ hours/day} * 60 \text{ minutes/hour} = 525,600 \text{ minutes}$$

"Five-nines" means that the network can be down for a grand total of less than 6 minutes during a year!

Telephony Standards

An international organization that's part of the United Nations, the International Telecommunications Union (ITU) plays a major role in standardizing the technology of the PSTN. Initially providing standards and agreements for connecting telegraph links between countries starting in the 1800s, the ITU has evolved to oversee many areas of standards development within the global telecom industry.

The ITU includes a specific division known as the Telecommunications Standardization Sector, or ITU-T. This division comprises many companies and organizations with interests in telecommunications standards. Once they're grouped into similar functional areas, the ITU-T standards are called recommendations, and they share an assigned letter of the alphabet. Some of the ITU-T recommendations that are relevant to our discussion are:

- G:** Transmission systems and media, digital systems and networks
- H:** Audiovisual and multimedia systems
- P:** Telephone transmission quality, telephone installations, local line networks

The recommendation category letter is typically followed by a period and a number, such as G.711 or H.323. An ITU-T standard recommendation is said to be "In Force" when the standard has been approved by ITU-T membership.

Standards are absolutely crucial to the success of technologies like VoIP. Without standards, your phone call would very likely be dropped when it passed from Vendor A's network to Vendor B's network. Accordingly, many VoIP vendors have drawn on the expertise of the ITU-T and built VoIP products based on well-known standards.

How the PSTN Works

To talk about VoIP technology, it helps to understand a little about how the PSTN works today. Here's what has to happen when someone—the caller—makes a telephone call to someone else—the callee—over the PSTN:

- 1) The caller picks up the telephone handset hears a dial tone.
- 2) A telephone number is entered, specifying the address of the callee.
- 3) Signals are sent through the PSTN to set up a circuit for the call. Capacity and bandwidth are reserved for the call.
- 4) The destination phone rings, indicating to the callee that a call has arrived.
- 5) The callee picks up the telephone handset and begins a conversation. The audio, voice conversation is translated to digital format in the center of the network, and then back to analog at the edge.
- 6) The conversation ends, call billing occurs, the circuit is taken down, and resources are released.

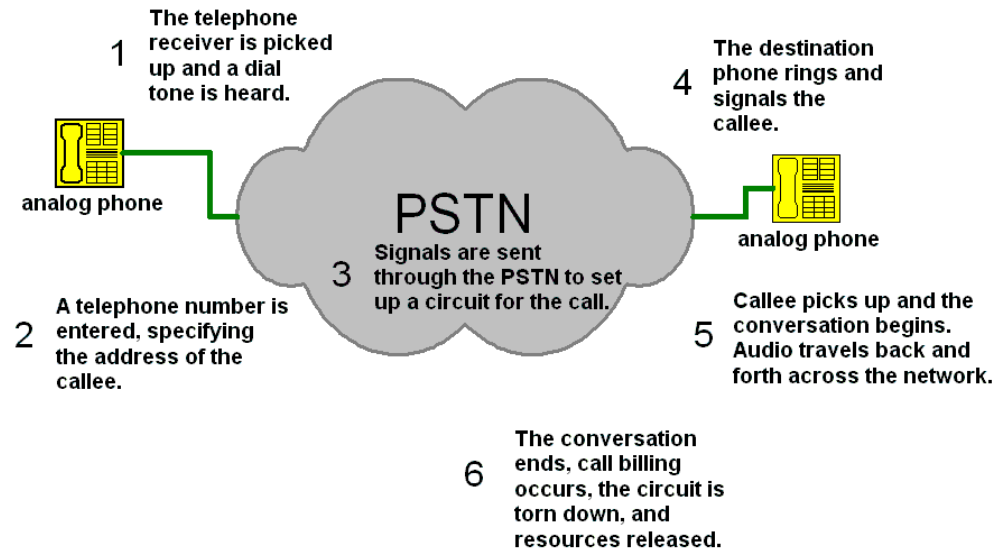


Figure 1. The six steps in a typical telephone call.

These steps must happen correctly and quickly for a telephone call to succeed with high quality. When telephony professionals consider providing the same functionality and reliability on relatively new and unreliable IP networks, you can see where some doubts and skepticism can occur.

PSTN Components

A number of components provide the infrastructure needed for fast and reliable calls on the PSTN. A brief introduction to these components will help in understanding what must be duplicated by VoIP technology to provide the same performance and reliability. Some of the components to be discussed are:

- Voice encoding
- Switches
- Private Branch Exchange (PBX)
- Signaling
- Telephones

Understanding Voice Encoding

When you speak into the mouthpiece of a telephone headset, your audio input is initially sent as an analog transmission over the telephone wiring. When the analog transmission reaches the entry point of the PSTN, it is digitized or converted into digital format – a series of zeros and ones. Once it has been digitized, the encoded voice transmission is transported across the PSTN network to the far edge, where it is converted back again to analog.

The method for converting audio into digital has been standardized. The name of this standard is G.711, and it uses an encoding technique called pulse code modulation (PCM). But, within the G.711 standard, there are two varieties:

- **G.711u:** Also known as μ -law encoding (the Greek letter “mu”), this is used primarily in North America.
- **G.711a:** Also known as a-law encoding, this is used primarily outside North America.

G.711 converts analog audio input into digital output at an output rate of 64000 bits per second, which is commonly referred to as 64 kilobits per second (kbps). A single G.711 voice channel is referred to as “digital signal, level 0,” or DS0. The fact that a DS0 takes up 64 kbps has been used in building links of the PSTN. Thus building a phone network link with a capacity for 24 voice channels would take $24 \times 64 \text{ kbps} = 1.544 \text{ megabits per second (Mbps)}$. A link with this capacity is known as a “trunk level 1” or T1 link.

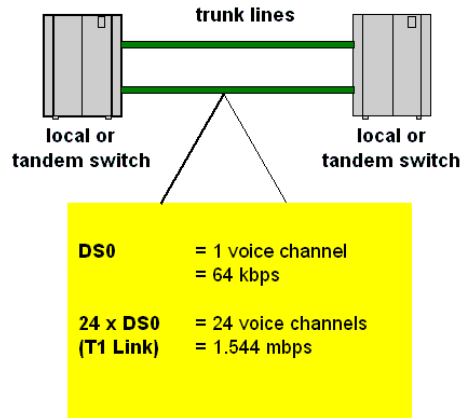


Figure 2. Voice channels in the PSTN.

We’ll encounter the G.711 standard again in our discussion of VoIP networks.

Understanding Switches

Switches are the core component of the PSTN. Switches of various types move call traffic from link to link and provide the circuits and dedicated connections necessary for PSTN calls. The links between switches are usually called trunk lines, whose capacity is usually stated in terms of the number of DS0 channels. Trunk lines use a technology called multiplexing to send multiple voice conversations over the same link.

PSTN switches are often categorized based on their function. However, switches that perform the same kinds of function are often known by multiple names. If you think of connecting a phone in your house or in your company to the PSTN, the first point of entry is a switch called a local switch or local office. This type of switch is also known as a Class 5 switch. The local switch is usually operated by a local telephone company, which is often referred to as a local exchange carrier (LEC). The local switch takes an analog input from the phone connection and digitizes it for transmission through the center of the PSTN. The digitized conversation is transmitted over trunk lines to the next switch in the network.

The next type of switch the digital signal encounters is a tandem switch or tandem office. Tandem switches are usually operated by a long-distance company, or interexchange carrier (IXC). Connected to local switches or other tandem switches to provide a logical, circuit-switched path through the PSTN, tandem switches are sometimes called Class 1, 2, 3, or 4 switches. They carry massive call volumes and are designed to be very scalable and reliable.

In VoIP systems, the IP router is analogous to the switches of the PSTN.

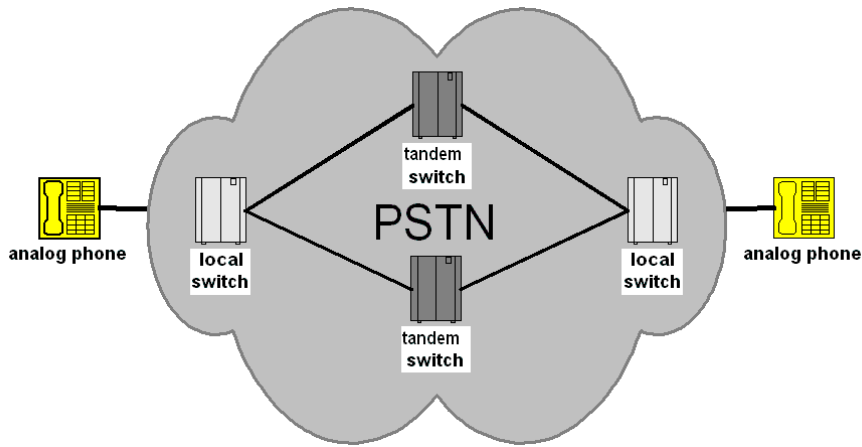


Figure 3. The PSTN, showing local and tandem switches.

Understanding PBXs

A private branch exchange, or PBX, is the foundation for most corporate voice networks. Typically, a corporate telephone network is different from a residential phone system. In a corporate environment, the network has to serve multiple users who need some advanced features, such as caller ID, call transfer, and call forwarding. In addition, the typical corporation would like for its phone system to act like a single network despite the fact that it serves offices in New York, Raleigh, and London.

Residential telephone systems must allocate a separate external phone line for every user. The PBX, on the other hand, allows corporate users to share a limited number of external telephone lines, providing cost savings to the company. It also supports traditional telephone features like call waiting, call conferencing, and call forwarding. Many larger corporations connect PBXs together with “tie lines,” which allow corporate users to make calls to co-workers without placing the call on the PSTN at all. To dial up a user over a tie line, you typically dial a different phone number, based on the tie line extension.

In VoIP systems, an “IP PBX” is analogous to the PBX of the PSTN, providing many of the same functions and features as a traditional PBX.

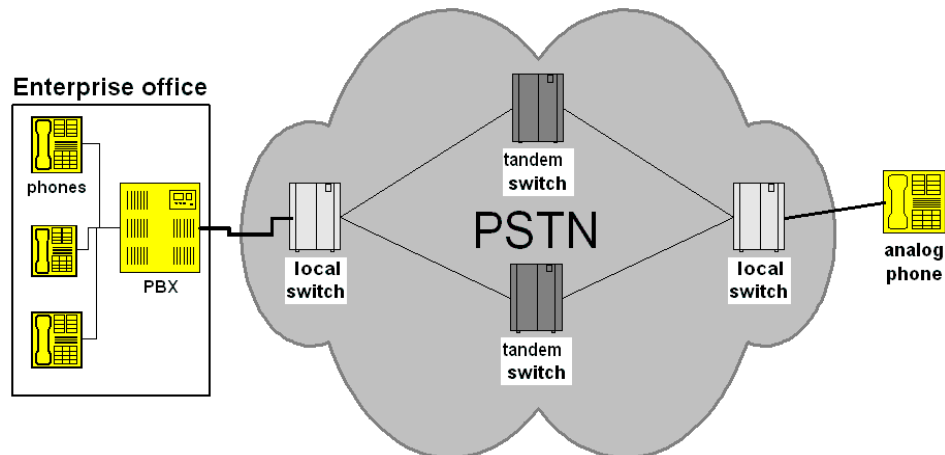


Figure 4. The PSTN, with an enterprise connected via a PBX.

Understanding Signaling

Establishing a telephone call requires several different types of signaling, to inform network devices that a telephone is off the hook, to supply destination information so that the call may be routed properly, and to notify both caller and callee that a call has been placed. A new technology in signaling, known as Signaling System 7 (SS7), is the ITU standard that provides for signaling, call setup, and management for the PSTN calls. Typically, a separate network is used for SS7 flows. Since the data transfer for SS7 does not occur on the same path as the call, it is sometimes referred to as out-of-band.

Two key components make up an SS7 network. The Signal Transfer Point (STP) provides routing through the SS7 network. You could think of these as the IP routers of the SS7 network. The Session Control Point (SCP) provides “800” number lookup and other management features.

When a phone call is made, the signaling protocols get involved to find the route to the callee, establish the connections between switches, and tear down these connections after the call ends. The STPs communicate with the local and tandem switches to reserve capacity between the switches in the path between caller and callee. After the call is completed, the STPs communicate with the switches to release the reserved connections and make them available for other calls.

VoIP systems have a corresponding set of rules for call signaling, which we discuss below.

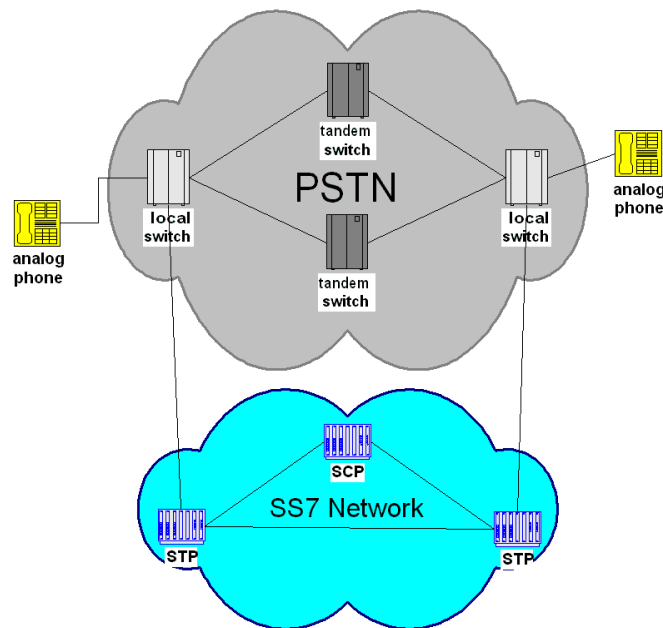


Figure 5. The PSTN uses SS7 networks for signaling.

Understanding Telephones

Telephones that connect to the PSTN traditionally come in two flavors: analog and digital.

- **Analog:** The type of phone that most people have in their homes today. It connects to the PSTN via traditional phone lines and sends an analog transmission – a waveform that varies over time.

- **Digital:** The type of phone that a lot of corporations use. It connects directly to a PBX and sends formatted digital transmissions: ones and zeros.

Nowadays, specialized IP phones can connect to the PSTN as well, but we'll discuss those in a later section and explain how IP phone technology differs from traditional telephones. However, all telephones have some kind of microphone and speakers. The traditional phone has a handset that's held to the ear and mouth during a conversation.

We have not dealt with cellular or mobile telephony technology in this chapter. You can think of the mobile phone network as an additional extension of the PSTN – most mobile calls are carried at least partially over the PSTN. The technology and components that the mobile phone system is built on are beyond the scope of this book.

In the Data Networking Community

Over the years, data-networking engineers have developed precise rules for how a data packet is constructed, and how each side behaves when it sends and receives data packets. These rules are called *protocols*. Although many protocols for data networking have been developed during the past 50 years, since the rise of the Internet, the Internet Protocol, or IP, has become the most important protocol.

IP has proved remarkably scalable and adaptable. That's why IP networking has become ubiquitous, changing the ways we think about transferring data and communicating. Over the past few years, the word “convergence” has drawn a lot of attention and promise to the IP-networking industry. Convergence means taking different types of data—voice, video, and application data—and transferring them over the same IP network.

How VoIP Works

Voice over IP, or VoIP, is simply the transfer of voice conversations as data over an IP network. Unlike traditional circuit-switched calls on the PSTN, in VoIP calls, the telephone connection is “packet-switched.” In a packet-switched environment, multiple computer devices share a single data network. They communicate by sending packets of data to one another, each packet containing addressing information that specifies the source and target computers. The packets within a single transmission can take different paths from end to end across a data network.

With a VoIP call, the call setup portion of the calling sequence has to be simulated—dial tone, ringing, busy signals. The audio portion of the call itself needs to be converted from analog to digital, cut into packets, sent across the network still in packet format, reassembled, and converted from digital back to analog. Codecs at either end do the conversion from analog to digital and back. We'll explain how they work a bit later.

Here's what happens when a call is made using VoIP:

- 1) The caller picks up the telephone handset and hears a dial tone.
- 2) The caller enters a telephone number, which will be mapped to the IP address of the callee.
- 3) Call setup protocols are invoked to locate the callee and send a signal to produce a ring.

- 4) The destination phone rings, indicating to the callee that a call has arrived.
- 5) The callee picks up the telephone handset and begins a two-way conversation. The audio transmission is encoded using a codec and travels over the IP network using a voice streaming protocol.
- 6) The conversation ends, call teardown occurs, and billing is performed.

Data Networking Standards

Just as the ITU has been influential in the creation of standards in the telephony community, the Internet Engineering Task Force (IETF) has led the standardization efforts in the data-networking community. Its particular focus has recently been on IP standards.

New data-networking techniques go through a rigorous trial phase, consisting of study, implementation, and review to verify their stability and robustness. Those that pass these critical examinations are known by their RFC (Request For Comment) number, because the RFC stage is the last step in the adoption of a draft standard as an approved standard.

Each of the components of the Internet Protocol that we discuss here – known by names such as TCP, UDP, and RTP – have one or more corresponding RFCs that describe their operation.

VoIP Components

To transfer voice data on the same network with e-mail and Web traffic, a new and different set of components is required. Some of these components are:

- Codecs
- TCP/IP and VoIP Protocols
- IP telephony servers and PBXs
- VoIP gateways and Routers
- IP phones and softphones

Understanding Codecs

A codec (which stands for “compressor/decompressor” or “coder/decoder”) is the hardware or software that samples analog sound and converts it to digital bits, which it outputs at a predetermined data rate. The codec often performs compression as well, to save bandwidth. There are dozens of available codecs, each with its own characteristics.

Codecs have odd-looking names that correspond to the name of the ITU standard that describes their operation. For example, the codecs named G.711u and G.711a convert from analog to digital and back with relatively high quality. As with most things digital, higher quality implies more bits, so these two codecs use more bandwidth than lower-speed codecs.

Lower-speed codecs, such as G.726, G.729, and those in the G.723.1 family, consume less network bandwidth. However, low-speed codecs impair the quality of the audio much more than high-speed codecs, because they compress the digital transmission with lossy compression – compression that loses some of the original data. Fewer bits are sent, so the receiving side does its best to approximate what the original audio sounded like, but it’s not a high-fidelity recreation.

The table below describes some common VoIP codecs. The middle column in the table shows the rate at which the codec generates its output. The “Packetization Delay” column refers to the delay a codec introduces as it converts from analog to digital and back. We’ll see in later chapters that this fixed amount of delay can affect the quality of the call as perceived by the listeners.

| Codec Name | Nominal Data Rate | Packetization Delay |
|--------------------------|--------------------------|----------------------------|
| G.711u | 64.0 kbps | 1.0 ms |
| G.711a | 64.0 kbps | 1.0 ms |
| G.726-32 | 32.0 kbps | 1.0 ms |
| G.729 | 8.0 kbps | 25.0 ms |
| G.723.1 | 6.3 kbps | 67.5 ms |
| MPMLQ | | |
| G.723.1 ACELP | 5.3 kbps | 67.5 ms |

Figure 6. Common codecs used in VoIP. For each codec, the codec’s data rate is shown, as well as the time needed by the codec to do the analog-to-digital and digital-to-analog conversions.

Codecs use sophisticated techniques for coding and compression. You’ll see names that stretch the limits of your math background, like Multi-Pulse Maximum Likelihood Quantization (MPMLQ) and conjugate structure Algebraic Code Excited Linear Predictive (ACELP) compression. The names tell how the codecs do their job; consider these topics beyond the scope of this book.

Packet loss concealment (PLC) is an additional feature available with the G.711u or G.711a codecs. PLC techniques reduce or mask the effects of data loss during a telephone conversation. PLC does not add delay or have bad side-effects, but it makes the G.711 codecs more expensive to manufacture. Because of its cost, PLC is relatively rare today.

Understanding TCP/IP Protocols

The TCP/IP family of protocols forms the basis of the Internet and most present-day corporate networks. Computer programs send and receive data over an IP network by making program calls to the TCP/IP software, known as the protocol stack, in their local computer. The TCP/IP stack in the local computer exchanges information with the TCP/IP stack in the target computer to accomplish the transfer of data from one side to the other. The information they exchange consists the size of the chunks of data they exchange (the datagram size), the identification associated with each datagram (the datagram header), and what should occur if a datagram is lost or damaged in transit.

It's the Internet Protocol (IP) that determines how datagrams get transferred across an IP network, from the sending program to the receiving program. Datagrams are the units sent and received from end to end by the two sides, and they move in hops, or segments, across a network. Each hop has its own network characteristics; for example, some hops may be fast Ethernet hops, whereas other hops may be slower modem connections. To optimize the performance of the hops, devices on the network may perform datagram fragmentation, cutting large datagrams into smaller pieces, called packets, which need to be reassembled back into the original datagrams once received.

When a datagram arrives at a router or switch in a network, the router or switch decides where the datagram should go in its next hop, and forwards it along. In later chapters we'll come back to this discussion of hops through the network, but for now, suffice it to say that too much time spent going through one or more of the hops can delay the datagrams and add variation in the delay time, making the telephone conversation sound poor.

The sending and receiving application programs communicate by means of a couple of related protocols when they contact their TCP/IP stack.

- **TCP:** When making calls to the Transmission Control Protocol (TCP) interface, the sending program wants to make sure that the receiving program gets everything that is sent – that is, it wants to avoid data being lost, duplicated, or out of order. TCP is known as a connection-oriented protocol because the two sides of the data exchange maintain strong tracking about everything that's sent and received. For example, your browser uses the TCP interface when fetching Web pages – you don't want to see holes or out-of-order pieces of data on the screen, so your browser and the Web server program work together to make sure everything is received intact.
- **UDP:** When using the User Datagram Protocol (UDP), the sending application has no assurance of delivery, and it's willing to deal with that. UDP is called a connection-less protocol, which means that when using this protocol, the two sides don't acknowledge receiving any data to make sure everything arrived intact. Think about a stock ticker running across the bottom of your screen. If a datagram is lost, causing one of the quotes to be lost, it's not catastrophic because another will come along shortly – a stock ticker application is a good example of a program that uses the UDP programming interface to send data.

The application assembles datagrams to contain protocol-specific information. The TCP or UDP portion of an individual datagram is nested inside the IP portion. For example, there's a header that describes how the payload of a UDP datagram is to be decoded. In turn, an IP header containing information such as the network addresses of the sender and the receiver encapsulates that header.

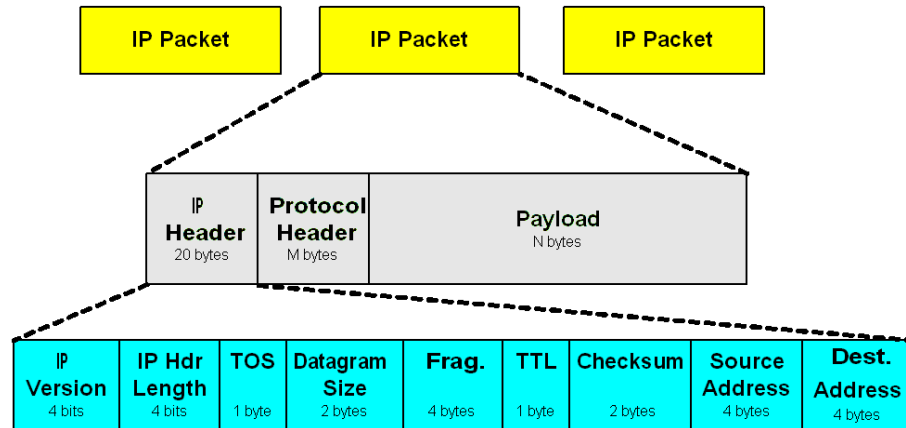


Figure 7. IP packets and their header format.

Whether the protocol is TCP or UDP, there are several standard fields in the header of every IP packet. We'll encounter these fields again in our discussion of VoIP:

TOS (Type Of Service)

The TOS byte can be used to mark the priority of a packet. It's generally set to zero, which means that the devices in the network that examine the packet give it their best effort in delivering it from one side of the network to the other. By setting this byte to a non-zero value, an application can request improved handling for a packet, meaning it's less likely to be dropped or delayed.

This byte is also known as the Differentiated Services (or DiffServ) field.

TTL (Time To Live)

Each time a packet takes one hop in its path across a network, the number in the TTL byte is reduced by one. If a device receives a packet with a zero in its TTL byte, it discards the packet. A TTL of zero means the packet has lived too long (that is, it has taken too many hops), indicating a problem with the network or with the packet. The TTL keeps packets from circling an IP network forever.

Checksum

A checksum is used to detect any changes made to the bits during a transmission. The sending side feeds all the bits it is sending through a sophisticated equation and writes the final result of the equation into the checksum field. The receiving side similarly passes all the bits it receives through the same equation. If its results match the checksum that was sent, the receiving side can be confident no bits were changed (accidentally or maliciously) during the transmission. Otherwise, it should discard the packet it received.

This checksum is used to verify the integrity of the IP header.

Source Address and Destination Address

These are the four-byte IP addresses of the sending and receiving applications. We traditionally write these four bytes in dotted notation, like 199.72.46.202.

The above definitions merely scratch the surface of an extremely complex subject. Because this is obviously a brief primer, we recommend that you seek out some of the many excellent books that explain TCP/IP comprehensively.

Understanding VoIP Protocols

Application programs build their own families of “higher-layer” protocols on top of the lower-layer protocols they use for transport and other tasks. Implementing a VoIP telephone call on a data network involves the call setup—that is, the VoIP equivalent of getting a dial tone, dialing a phone number, getting a ring or a busy signal at the far end, and picking up the phone to answer the call—and then the telephone conversation itself. VoIP protocols are required during both phases:

- Several higher-layer protocols can accomplish call setup and takedown, including H.323, SIP, MGCP, and Megaco. The programs that implement the call setup protocol use TCP and UDP to encapsulate the data exchanged during the call setup and takedown phases.
- The exchange of actual encoded voice data occurs after the call setup (and before the call takedown), using two data flows—one in each direction—to let both participants speak at the same time. Each of these two data flows uses a higher-layer protocol called Real-time Transport Protocol (RTP), which is encapsulated in UDP as it travels over the wire.

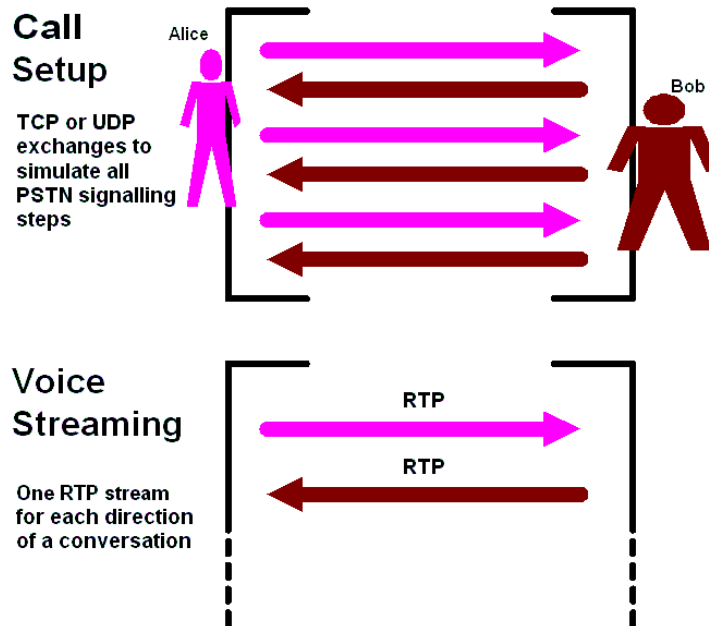


Figure 8. There are two sets of high-level protocols: for call setup and for the conversation.

Let’s look at the call setup and voice streaming protocols in more depth.

Call Setup Protocols

Call setup protocols use TCP and UDP to encapsulate the call setup and takedown phases of a telephone call. They handle functions like the mapping of phone numbers to IP addresses, generating dial tones and busy signals, ringing the callee, and hanging up. There are two families of call setup protocols: one set from the telephony community and the other from the data networking community.

We get the call setup protocols H.323 and MGCP (Media Gateway Control Protocol) from the telephony community by way of the ITU. H.323 is the most widely deployed call setup protocol in use today; a report from Insight Research in January 2001 indicated that 89% of VoIP calls were using H.323. H.323 is actually a family of telephony-based standards for multimedia, including voice and videoconferencing. MGCP is the less flexible version, for use with inexpensive devices like home telephones.

The family of H.323 protocols has been refined for many years, and as a result, it is robust and flexible. But the cost of this robustness is that it has high overhead: a calling session includes lots of handshakes and data exchanged for each function performed.

SIP (Session Initiation Protocol) and Megaco (another acronym for Media Gateway Control Protocol) are lightweight protocols developed by the IETF in the data-networking community. SIP in particular represents typical data-networking logic, which asks, Why use a heavyweight protocol (such as H.323) when a lightweight protocol (such as SIP) will get the job done most of the time? SIP is the current “industry darling”—it is supported by Cisco and Nortel, and Microsoft has recently started shipping SIP client interfaces with its Windows XP operating system.

Although the H.323 family of call setup protocols is predominantly used today, the Insight Research report cited above predicts that the four protocols discussed here, H.323, MGCP, Megaco, and SIP, will each be used in roughly equal proportions within the next few years.

Voice Streaming Protocols

RTP is widely used for streaming audio and video; it is designed for applications that send data in one direction with no acknowledgment. The header of each RTP datagram contains a timestamp, so the application receiving the datagram can reconstruct the timing of the original data. It also contains a sequence number, so the receiving side can deal with missing, duplicate, or out-of-order datagrams.

The two RTP streams, that is, the bi-directional conversation itself, are the important elements in determining call quality of the voice conversations. Let’s look at the composition of the RTP datagrams, which transport the voice datagrams.

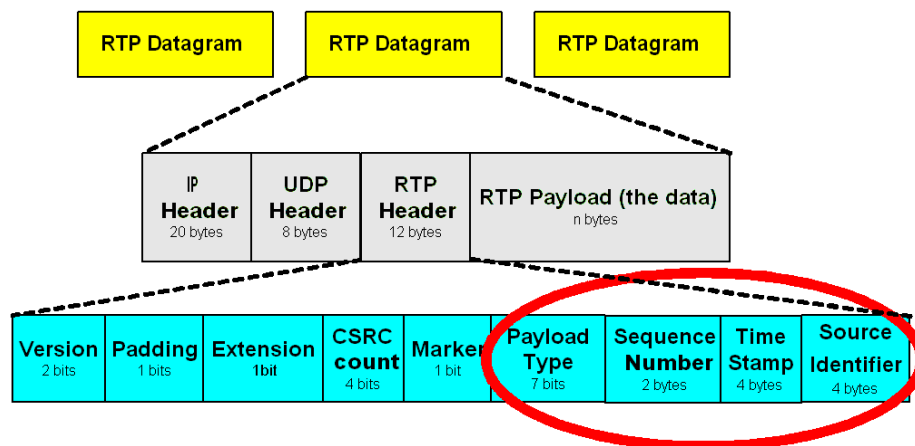


Figure 9. The header used for RTP follows the UDP header in each datagram. The four important fields in the RTP header are described below.

All the fields related to RTP sit inside the UDP payload. So, like UDP, RTP is a connection-less protocol. The software that executes RTP is not commonly part of the TCP/IP protocol stack, so applications are coded to add and recognize an additional 12-byte header in each UDP datagram. The sender fills in each header, which contains four important fields:

RTP Payload Type

Indicates which codec to use. The codec conveys the type of data (such as voice, audio, or video) and how it is encoded.

Sequence Number

Helps the receiving side reassemble the data and to detect lost, out-of-order, and duplicate datagrams.

Timestamp

Used to reconstruct the timing of the original audio or video. Also, helps the receiving side determine consistency or the variation of arrival times, known as jitter.

It's the timestamp that brings real value to RTP. An RTP sender puts a timestamp in each datagram it sends. The receiving side of an RTP application sees when each datagram actually arrives and compares this to the timestamp. If the time between datagram arrivals is the same as when they were sent, there's no variation. However, there could be lots of variation in the arrival times of datagrams depending on network conditions, and the receiving side can easily calculate this jitter.

Source ID

Lets the software at the receiving side distinguish among multiple, simultaneous incoming streams.

The accumulation of headers can add a lot of overhead, depending on the size of the data payload. For example, a typical payload size when using the G.729 codec is 20 bytes, which means that the codec outputs 20-byte chunks of the VoIP call at a predetermined rate specific to that codec. With RTP, two-thirds of the datagram is the header because the total header overhead consists of:

$$\text{RTP (12 bytes)} + \text{UDP (8 bytes)} + \text{IP (20 bytes)} = 40 \text{ bytes}$$

Real bandwidth consumption by VoIP calls is higher than it first appears. The G.729 codec, for example, has a data payload rate of 8 kbps. Its actual bandwidth usage is higher than this, however. When sent at 20 ms intervals, its payload size is 20 bytes per datagram. To this, add the 40 bytes of RTP header (yes, the header is bigger than the payload) and any additional layer 2 headers. For example, Ethernet drivers generally add 18 more bytes. Also, there are two concurrent RTP flows (one in each direction), so double the bandwidth consumption you've calculated so far. The "Combined Bandwidth" column in the table below shows a truer picture of actual bandwidth usage for some common codecs.

Some IP phones let you set the "delay between packets" or "speech packet length," that is, the rate at which the sender delivers datagrams into the network. For example, at 64 kbps, a "20 millisecond speech datagram" implies that the sending side creates a 160-byte datagram payload every 20 ms. There is a simple equation that relates the codec speed, the delay between voice datagrams, and the datagram payload size:

$$\text{Payload size (in bytes)} = \frac{\text{Codec speed (in bits/sec)} \times \text{datagram delay (ms)}}{8 \text{ (bits/byte)} \times 1000 \text{ (ms/sec)}}$$

In this example:

$$160 \text{ bytes} = (64000 \times 20) / 8000$$

For a given data rate, increasing the delay causes the datagrams to get larger, since the datagrams are sent less frequently to transport the same quantity of data. A delay of 30 ms at a data rate of 64 kbps would mean sending 240-byte datagrams.

| Codec | Nominal Data Rate | Packetization Delay | Typical Datagram Size | Combined Bandwidth for 2 Flows |
|------------------|-------------------|---------------------|-----------------------|--------------------------------|
| G.711u | 64.0 kbps | 1.0 ms | 20 ms | 174.40 kbps |
| G.711a | 64.0 kbps | 1.0 ms | 20 ms | 174.40 kbps |
| G.726-32 | 32.0 kbps | 1.0 ms | 20 ms | 110.40 kbps |
| G.729 | 8.0 kbps | 25.0 ms | 20 ms | 62.40 kbps |
| G.723.1 MPMLQ | 6.3 kbps | 67.5 ms | 30 ms | 43.73 kbps |
| G.723.1 ACELP | 5.3 kbps | 67.5 ms | 30 ms | 41.60 kbps |

Figure 10. This rightmost column of this table shows the real bandwidth consumed for each codec in a two-way VoIP telephone conversation.

IP Telephony Servers and PBXs

Many data-networking transactions are based on the concept of client-server computing. Client computers make requests for services to server computers, which perform those services and return the results. You're probably familiar with Web servers, e-mail servers, and database servers.

Adding voice data to IP networks provides yet another set of servers that are designed to provide voice services in new and innovative ways. An IP PBX typically serves as the core IP telephony server. On the PSTN, the PBX is often a "closed-box" system—it provides all the voice functions and features you need, but usually in a proprietary manner. Management of the closed-box platform is left up to the PBX vendor. With VoIP, an IP PBX can be built on a PC platform running on an operating system like Microsoft Windows, Linux, or Sun Solaris. While parts of the IP PBX are inherently proprietary, the platforms can be managed through vendor application programming interfaces (APIs) and through the standard APIs provided by the operating system itself.

An IP PBX provides functions and features like those that a traditional PBX provides. While the standard PBX of the PSTN offers multiple features developed over decades, such as call transfer and call forwarding, IP PBXs are quickly providing the same kinds of features and more.

Other types of IP telephony servers provide new and interesting services. The possibility of unified messaging—the convergence of voice mail and e-mail—can be considered a benefit of a VoIP implementation. Unified messaging servers also run on PC platforms and talk to e-mail servers and IP PBXs to provide message access in a variety of ways.

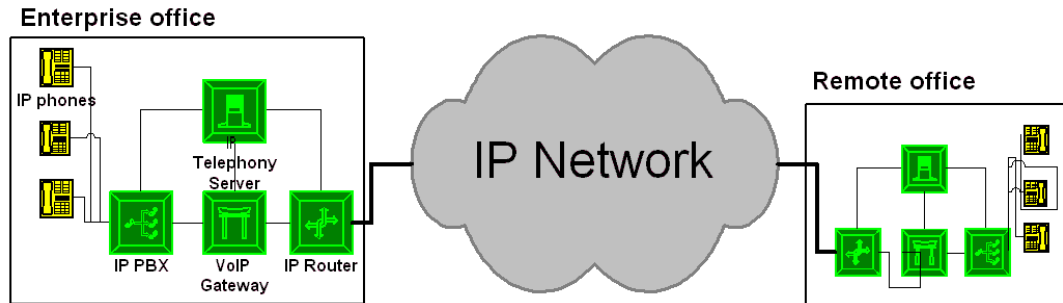


Figure 11. A VoIP network and its typical components.

Another new concept introduced along with IP telephony servers is clustering, in which several of these servers are grouped together in a cluster to offer increased scalability, reliability, and redundancy. Clustered servers function together and can be managed as a unit, providing combined processing power while logically appearing as a single server. Clustering is not available with traditional PBXs in the PSTN.

Gatekeepers are another type of server. Gatekeepers are used by the H.323 protocol to provide admission control features and other management functions for multimedia services.

Video streaming and video conferencing servers also deserve some mention here. While not directly related to VoIP, video servers will eventually take advantage of the converged network infrastructure. Due to its increased bandwidth requirements, video over IP offers a new set of challenges that makes VoIP look easy!

VoIP Gateways and Routers

VoIP gateways and IP routers move RTP voice datagrams through an IP network. VoIP gateways provide a connection between the VoIP network and the PSTN. These devices therefore play a key role in the migration path towards VoIP. There are few totally VoIP phone networks in the world today. It is necessary to connect to the PSTN to place calls to PSTN users. VoIP gateways must talk the SS7 protocol (remember that protocol from the PSTN section?). SS7 is used by the VoIP gateway to signal switches in the PSTN when a phone call is originating from the VoIP network with the callee in the PSTN. VoIP gateways may also provide conversion between different codecs, which is called “transcoding.” If a codec other than G.711, let’s say G.729, is used on the VoIP network, the voice data must be converted to G.711 before being transferred to the PSTN.

In a corporate environment, VoIP gateways can interconnect with traditional PBXs to provide a migration path and allow for staged VoIP deployments. Gateways are typically very smart in terms of the number of protocols that they speak. They have to be, to handle the variety of signaling and data protocols of the VoIP network and PSTN.

By examining the IP packet headers, IP routers make the necessary decisions to move packets to the next router or hop along the path to the destination. Tracing the route of a voice packet through the network can be useful for problem identification and diagnosis; we'll discuss techniques like this in later chapters. However, router technology itself is well-understood and is not discussed in detail in this book.

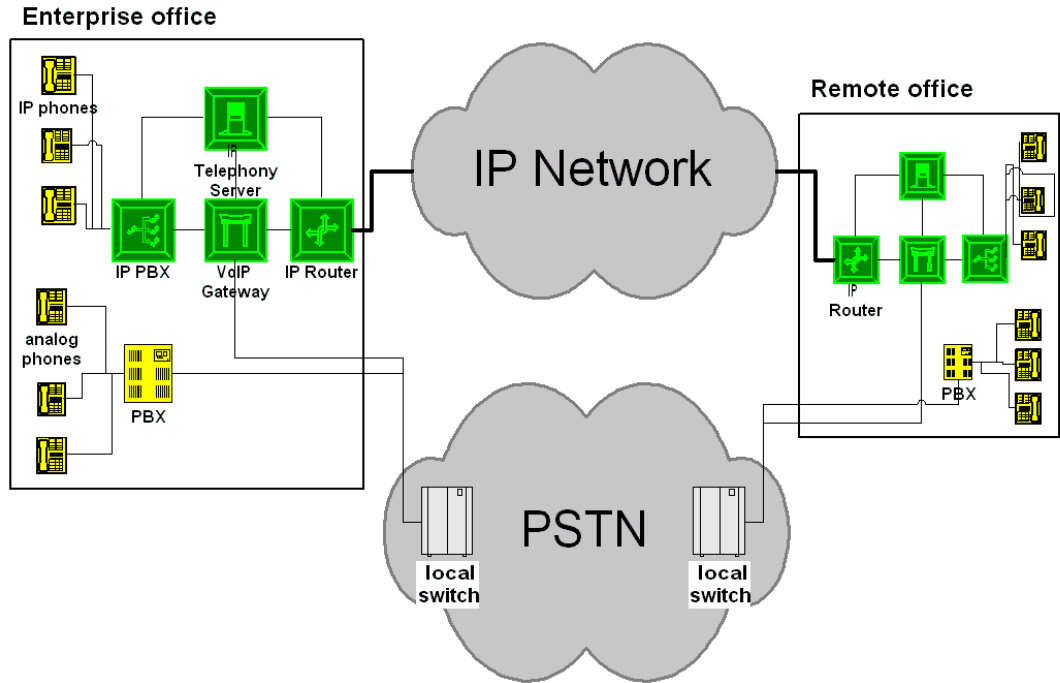


Figure 12. A VoIP network, with its VoIP gateways connected a fallback PSTN.

IP phones and Softphones

To make VoIP work, analog audio must first be converted to digital datagrams. We know this is done by codecs. But, where is the conversion done? Where are the codecs located?

If you're still using older analog telephones, the codecs are located in the IP PBX. Incoming calls are digitized there, before being forwarded onto the IP network.

As an alternative, the codecs can be located in the telephones themselves. These new digital telephones are called IP phones. Rather than having a 4-line telephone connector in the back, they usually have an Ethernet LAN connection. An IP phone makes data connections to an IP telephony server, which does the call setup processing.

And there's yet another choice. Your humble computer can serve in the role of the IP phone on your desk. You plug a headset and microphone into the computer's audio card. The computer's CPU runs the software doing the codec processing, and the computer has a LAN connection into the data network. As with an IP phone, your computer probably relies on an IP telephony server to do call setup processing.

Now that we've discussed the basics of VoIP and its resemblances—and distinctions—from the telephone network we're all accustomed to using, we'll next cover the potential benefits of VoIP, and we'll try to separate out some of the pie-in-the-sky VoIP fantasies we've seen in circulation from the real returns you can expect to gather from your own implementation.

For More Information

Good TCP/IP primers:

1. Comer, D. E. *Internetworking with TCP/IP Vol. I: Principles, Protocols, and Architecture*. Prentice Hall, Englewood Cliffs, NJ, 2000. (ISBN 0-13-018380-6).
2. Stevens, W. R. *TCP/IP Illustrated, Volume 1*. Addison-Wesley, Reading MA, 1994 (ISBN 0-201-63346-9).
3. Walker II, J. Q. and J. T. Hicks. "Protocol ensures safer multimedia delivery," *Network World*, volume 16, number 44, November 1, 1999, page 53.

Good VoIP overviews:

1. Davidson, J. and J. Peters. *Voice over IP Fundamentals*. Cisco Press, Indianapolis, IN, 2000. (ISBN 1-57870-168-6).
2. Newton, Harry. *Newton's Telecom Dictionary*. CMP Books, New York, New York, 2001. (ISBN 1-57820-069-5).
3. *Checklist of VoIP Network Design Tips*, John Q. Walker, NetIQ Corporation, April 2001, www.netiq.com/products/chr/whitepapers.asp.
4. *What You Need to Know before You Deploy VoIP*, Scott Hamilton and Charles Bruno, Tolly Research, April 2, 2001, www.netiq.com/products/chr/whitepapers.asp.

About the Authors

John Q. Walker is a director at NetIQ Corporation. He is one of the founders of Ganymede Software, heading the software development team there before it joined NetIQ in spring 2000. In earlier jobs, he managed teams responsible for designing and developing high-speed networking at IBM. Another job stint involved inventing ways to break security on a new computer system, reporting his methods to the development team. He co-authored a book on portable network programming for McGraw-Hill. Dr. Walker holds a Ph.D. in software engineering; his Masters degree focused on system testing. His e-mail is johnq@netiq.com.

Jeffrey T. Hicks is a Technical Lead at NetIQ Corporation. He has recently led the development teams for the award-winning Chariot and Chariot VoIP Assessor products. He has been active in the development of VoIP testing and management solutions for the past four years. In earlier jobs, he helped develop innovative network communications software products at IBM. Mr. Hicks holds a Masters of Engineering from North Carolina State University and a B.S. in Computer Engineering from Auburn University. His e-mail is jeff.hicks@netiq.com.