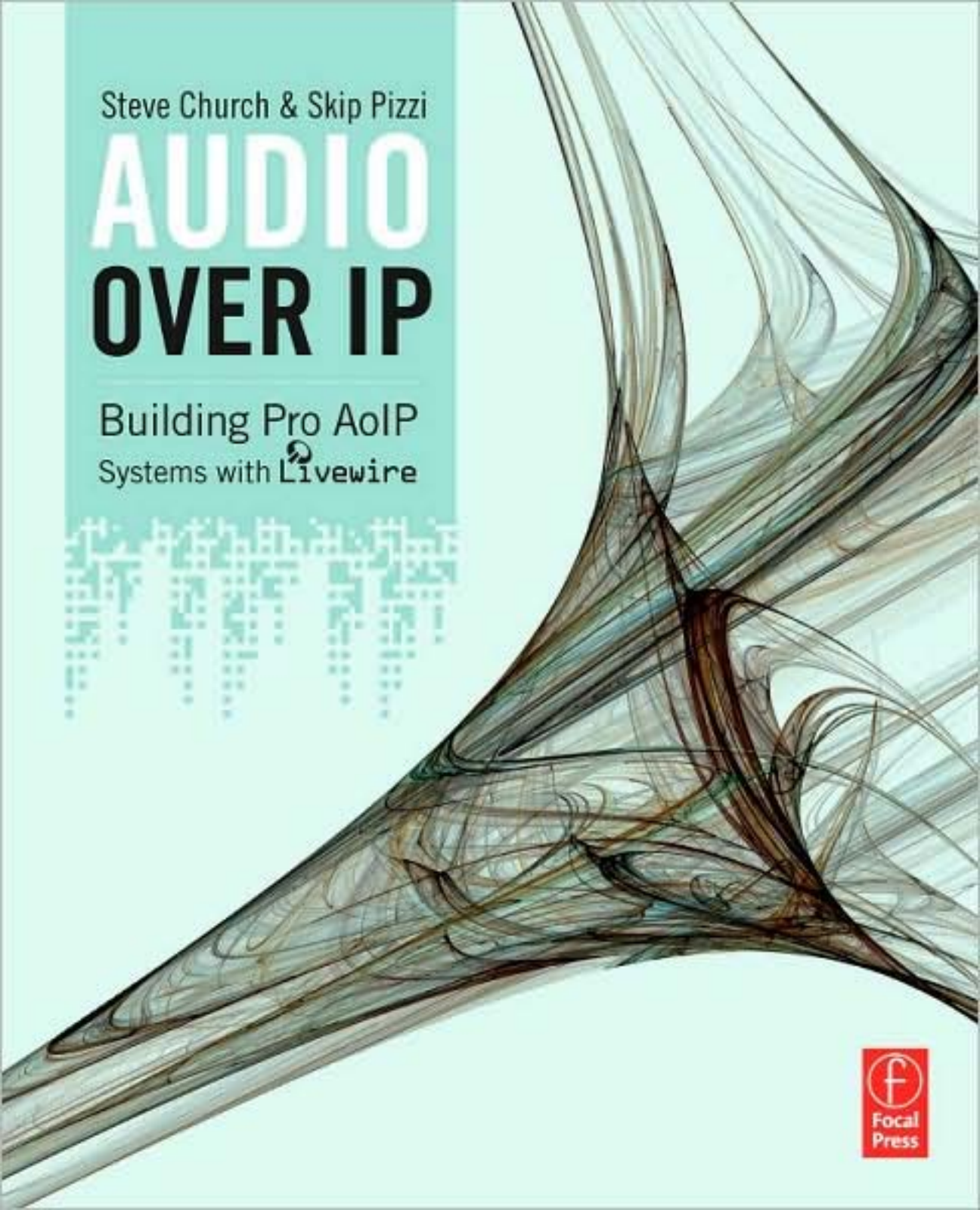
Steve Church & Skip Pizzi

AUDIO OVER IP

Building Pro AoIP Systems with Livewire



Focal Press

Focal Press is an imprint of Elsevier
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA
Linacre House, Jordan Hill, Oxford OX2 8DP, UK

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

Library of Congress Cataloging-in-Publication Data

Pizzi, Skip.
Audio over IP : building pro AoIP systems with Livewire / Skip Pizzi, Steve Church.
p. cm.
ISBN 978-0-240-81244-1
1. Digital audio broadcasting. 2. Netscape Livewire (Computer file) I. Church, Steve. II. Title.
TK6562.D54P59 2010
006.5--dc22 2009029537

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library.

ISBN: 978-0-240-81244-1

For information on all Focal Press publications
visit our website at www.elsevierdirect.com

09 10 11 12 5 4 3 2 1
Printed in the United States of America

Dedications

To my wife and muse, Lana, who, happily, says she missed me while I was writing.
–Steve Church

To my family, who make all of life's lessons worth learning.
–Skip Pizzi

Acknowledgements

Our thanks go to the many amazing people who have contributed to the development of Livewire AoIP technology and its realization, most notably Greg Shay, Michael Dosch, and Maciej Slapka in Cleveland, and Maris Sprancis, Oleg Krylov, Gints Linis, Normunds, Artis, and the rest of the LUMII team in Riga, Latvia. We also are indebted to our editors at Focal Press, Melinda Rankin, Carlin Reagan, Paul Temme, and Angelina Ward, and our colleagues at *Radio World*, Paul McLane and John Casey. To all, we are grateful.

Preface

In 1984, the writer Italo Calvino began composing a series of lectures he never delivered. They were entitled "Six Memos for the Next Millennium," although he only completed five of them before his sudden death in 1985. The lectures were later published in a book of the same name.[1]

His lectures—or "memos," as he preferred—were critiques of literature, considering a myriad of works ranging from Lucretius and Ovid to Joyce and Dostoevsky. Yet a quick look at the lectures' titles shows how they serve as apt metaphors to technology, as well. Their subjects are paragons we associate closely with the digital age that now flourishes in the new era that Calvino addressed from some temporal distance. He named his memos simply:

1. Lightness
2. Quickness
3. Exactitude
4. Visibility
5. Multiplicity
6. Consistency

Engineers will observe how these could easily be taken as high-level design requirements for any proper technology. And in fact, as Calvino wrote these lectures,[2] the Internet as we know it today was being born.[3]

From our contemporary perspective, these concepts still apply well, and also fit nicely into audio engineers' narrower worldview of an idealized digital environment. So they particularly pertain to the subject at hand—audio over Internet Protocol (AoIP) for professional applications.

The agility, speed, accuracy, clarity in design, scalability, and reliability that AoIP systems possess closely mirror the six virtues that Calvino set out. In fact, we could dare to add a seventh, "Efficiency," to complete the set of qualities we ascribe to today's AoIP technologies. Of course, returning to the mundane as we ultimately must, this last attribute translates to cost effectiveness, which is likely the most appealing of all to today's implementers. But it is the other, more fundamental characteristics that combine to enable this more pecuniary advantage.

---

[1] Italo Calvino, *Six Memos for the Next Millennium* (Cambridge, MA: Harvard University Press, 1988).
[2] ARPAnet had just fully converted to TCP/IP in 1983, and the term *Internet* was recognized as the network's official new name at that time.
[3] Christos J. P. Moschovitis, ed., *History of the Internet: A Chronology, 1843 to Present* (Santa Barbara, CA: ABC-CLIO, 1999).

Don't be alarmed—we'll not dwell long on Calvino's literate musings. Rooted as we are in the no-nonsense environs of radio studios and audio production facilities, we'll move quickly to our main goal: to illuminate the practical workings of AoIP. To round your learning, we will provide both the necessary theoretical concepts and hands-on examples of AoIP systems for professional audio and broadcast use.

We begin with a treatment of general AoIP principles, then proceed to how these are realized in one particular family of products—the *Livewire*[4] system. The use of real-world reference points are valuable to understanding, aiding in the transference of purely conceptual information to knowledge that can be acted upon.

The motivation for our choice of Livewire for specific description and concrete examples is twofold: First, it is a standards-based system, making it well suited to the task of illustrating the value of the standardized networking approach. Second, the Livewire system is in wide use around the world. (And, third, it doesn't hurt that we're pretty familiar with it.)

We further believe that our coverage of Livewire as a specific instance of AoIP does not reduce the utility of this book for users or potential users of other AoIP systems. On the contrary, having real examples is vastly preferable to sticking purely to theory. We trust that many of the elements of Livewire we discuss will be easily recognized and made applicable to other systems.

Compare this to a book on web design. If the presentation considered only generic source code and did not describe the actual effects on a particular browser, its usefulness would be greatly reduced.

Thus, the chapters at the beginning and end of this book consider generic AoIP technology, while the ones in the middle focus on Livewire's specific implementation of it. Among these central parts, Chapters 6 and 7 play around the boundaries of AoIP, covering Voice over IP (VoIP) telephony, and audio codecs optimized for the IP environment, respectively. These chapters treat their subjects in a largely non-implementation-specific manner, as well, but one in which the professional audio and the broadcast facility are primarily considered.

Finally, this book is for two groups: those who already have installed AoIP systems and those who are considering it. For the first, this book can serve as a manual with a wealth of information on cabling techniques, equipment maintenance, and other real-world topics. For the second, we trust that upon reading this book, you will understand AoIP at a sufficient level to evaluate if it's right for your facility. In either case, we hope you will find this book a helpful guide along professional audio's new frontier.

[4]The *Livewire* format is a standards-based AoIP system developed by the Cleveland, Ohio-based manufacturer Axia Audio, and supported by a growing number of other audio and broadcast equipment manufacturers. (See the References and Resources section for further information on Livewire products and partners.)

CHAPTER

Introduction to AoIP

1

"IP is like Pac-Man. Eventually, it will eat everything in its way."

—Hossein Eslambolchi, President, AT&T Labs

"Rock and roll is the hamburger that ate the world."

—Peter York[1]

"AoIP eats old-school studio audio technologies for lunch."

—Steve and Skip

The *Internet Protocol*, usually simply called IP, is at the heart of the Internet. IP is the common format used for any kind of data that flows on the Internet and on private extensions of the Internet, such as the local area networks (LANs) employed in enterprise networks and small office/home office (SOHO) networks. Together with Ethernet for transport (cabled or wireless), the rules are set for the entire data networking infrastructure, both hardware and software, which has emerged from a rabble of competitors and has been so broadly embraced over the last quarter century.

IP is now driving a revolution in the field of audio studio design. It promotes a fundamental rethinking of the way signals are distributed and managed throughout the broadcast facility. Since most audio facilities have already been converted to digital, it makes sense to move on to explore the next step in the progression—transitioning to IP—as well.

Given that IP is the *lingua franca* of contemporary data networking, it can provide significant economies of scale for specialized applications such as professional digital audio distribution. This exploits the same process that has made the general-purpose desktop computer an efficient and cost-effective platform for the creation and storage of professional audio content. Audio-over-IP (AoIP) distribution is simply an extension of that thinking and technology, replacing the purpose-built (and relatively expensive) mixers, routers, and switchers that have traditionally been used by

[1]Peter York is a British author, columnist, and broadcaster. It's not clear to us if he was being kind to rock and roll, or hamburgers. It's also not clear if his comment is relevant. But, we are sure that we like it, and that it fits the "eating everything in its path" theme.

Audio Over IP

doi: 10.1016/B978-0-240-81244-1.00001-8

audio studios for managing multiple audio signals as they pass through a production or broadcast facility. IP also allows the full and continuing force of Moore's Law (which states that capacity doubles every two years) to be applied to audio distribution, just as the PC has done for recording and editing. (Anyone remember New England Digital's Synclavier? Popular in the 1980s, this audio recorder/editor/synthesizer was an impressive machine that cost its well-heeled owners over a half-million bucks. Today, a $400 PC offers teenagers much more audio production power.)

Beyond cost effectiveness, however, AoIP offers other important benefits, including:

- Scalability (i.e., the ability to easily accommodate growth and other configuration changes).
- Convenience (i.e., easy and fast installation).
- Tight integration with Voice over IP (VoIP) phone systems, IP codecs, and PC-based applications.
- Smooth incorporation of other services such as associated text and visual content.
- "Future-proofing" (i.e., high likelihood of fitting well into any scenario for future facility requirements).

Putting all these elements together creates a value proposition that is hard to ignore when you are considering options for new facility designs or existing studio upgrades.

Studio audio systems using IP-based technology are now sufficiently mature to allow audio producers and broadcasters to confidently make the transition, providing them with substantial savings while simultaneously positioning them well to accommodate future needs.

1.1 TWO TO TANGO

The broadcast audio studio has a long legacy relationship with the telecommunications world. The earliest audio facilities and standard practices were developed by Bell System and Western Electric engineers in the early 20th century, and the two worlds have never strayed far from each other since.

In particular, broadcast audio has retained a close connection to the telecom environment, since so much of broadcasting's content comes and/or goes from the studio via telco-provided paths. Broadcast equipment designers also have leveraged (and continue to) the massive research and development (R&D) investment made in telecom/datacom technologies.

AT&T's U-Verse service is instructive. It is a consumer telecommunications offering that bundles TV, voice, and Internet, all of which are IP-based. Meanwhile, Alcatel/Lucent, which now owns AT&T's central office equipment business, shows no circuit-switched products on their web site, instead focusing on IP-based central office solutions. AT&T was, of course, the company that invented the circuit-switched paradigm that powered telephony since the 1970s, and served as the inspiration for traditional broadcast routing gear.

U-Verse is an example of an "IP but not Internet" application. The TV and voice services don't need to use IP, but AT&T has decided to consolidate all the services on a common infrastructure, presumably to both save money by leveraging high-volume hardware and to have maximum flexibility via IP's do-anything capability to adapt to whatever the future might bring.

It is not surprising that the next generation of studio audio technology should once again follow a path blazed by telecommunications technologies. AoIP is also "IP but not Internet," leveraging high-volume standard hardware and offering future-proof flexibility.

1.2 ARGUMENTS FOR AoIP

What makes IP so compelling? It's "just a protocol," right? Yes. But a protocol in the data networking context can provide tremendous value to users. At the technology level, it's simply a set of rules: the way data is assembled into packets, how confirmation of reception is communicated, etc. But to users, it means that any conforming equipment is interoperable. And because the IP protocol was designed with generality and extensibility in mind, it enables designers to create novel applications.

Although originally developed for email and file transfers, as the speed of the Internet increased, IP came to be used for media transmission as well, which is now well known as *streaming media*. This development has fundamentally altered the nature of how people use the Internet, and has subsequently had significant impact on all aspects of the media industry as it struggles to cope with the changes it brings and to take advantage of the new opportunities it engenders. Though the Internet's inventors were probably not thinking of streaming when they designed IP, they *were* thinking that keeping the core open and layered would unlock the door to a variety of applications that future creative types might dream up.

Which brings us to AoIP. While they are related, AoIP is not streaming media. Streaming is exemplified by public Internet applications such as YouTube and Pandora. There are no delivery guarantees for these services, and delay can range into tens of seconds.

On the other hand, AoIP is intended to be run exclusively on a controlled local network infrastructure. In some cases, this is just an Ethernet switch. In others, it's a sophisticated system comprised of multiple IP routers and/or Ethernet switches. In all cases, an AoIP system is designed to ensure reliable, low-delay delivery of audio streams suitable for professional applications.

1.2.1 Scalability

Perhaps the most fundamental advantage of AoIP systems over other audio technologies—analog or digital—is the ability of its underlying IP and Ethernet architectures to adapt to change and growth.

For example, a traditional audio environment must have its spatial or imaging format (e.g., mono, stereo, or surround) predetermined, along with the number of simultaneous audio channels it requires (e.g., one, two, or more). An AoIP environment has no such requirement, and can easily adapt to any audio channelization format. This applies to accommodation of any other "layers" in the system as well, such as control-data channels. In traditional architectures, a dedicated path had to be specified for these extra channels (such as RS-422 control data). AoIP systems allow such auxiliary components to be easily and flexibly carried alongside the audio payload.

Similarly, a traditional "crosspoint" audio routing switcher must have its input and output (I/O) configuration fixed in its hardware design. In this way, such a device reflects *circuit switching* and parallel design, whereas AoIP systems implement *packet switching* and serial design. The packetized, serial approach allows great flexibility and responsiveness in accommodating changes in I/O configuration.

Just as telcos have moved away from the circuit-switched paths of their earlier years for similar reasons, studio audio systems can now enjoy the same advantages of scalability and flexibility to implement expansion in any dimension. This comes not a moment too soon, given the competitive pressures coming to bear on broadcasters to accommodate increased content production and expanded audience choice.

1.2.2 Cost Effectiveness

At almost any reasonable size, an IP-based audio system will compare favorably with the cost of a traditional system—both in terms of its hardware and materials pricing, and its installation costs. The reduction in wire alone provides substantial economy.[2] Maintenance expenses for AoIP systems are generally also lower.

These cost differentials increase with the size of the facility, which is why so many larger installations have already moved to IP-based solutions as their needs have called for new technical plants.

1.2.3 Convenience

The small physical footprint, low operating cost, ease of reconfiguration or upgrade, and fast installation of AoIP systems make them extremely convenient for engineering and operations alike at the audio studio facility.

From initial design to implementation to daily operation, IP-based systems make life easier.

[2]Remember that a packet-switched system like AoIP does not require individual wiring paths to each I/O of every device. For example, an audio mixing console or multitrack recording device can have all of its inputs and outputs interfaced to the rest of the facility via a single cable in an AoIP environment.

1.2.4 Smooth Integration with Other IP-Based Systems

VoIP phone systems and IP codecs can be tightly interconnected, creating numerous benefits with regard to both ease of installation and feature enhancement.

1.2.5 Talking the PC's Native Language

A lot of studio audio these days is either being sourced from a PC or being sent to one. IP/Ethernet is the PC's native language, allowing a powerful low-cost interface. Via a single RJ-45 connector, many channels of bit-accurate, high-resolution, bidirectional audio can be connected. Control comes along for the ride.

1.2.6 In the Tech Mainstream

Being in the tech mainstream means that there are a wide variety of learning resources. Books, web sites, and college courses that cover IP and network engineering abound.

Category (Cat) cables, assembly tools, RJ patch cords, jacks, testers, etc. are widely and locally available. Even some Ethernet switches and IP routers are often stocked locally.

1.2.7 Future-proofing

Nothing strikes fear in the heart of the engineer or manager more than making a bad decision on a big-ticket purchase. Moving to an IP-based audio architecture takes a lot of the pressure off, since it offers such flexibility and allows broad ability for reconfiguration down the road. Provisioning for unforeseen changes is much less problematic and cheaper with AoIP than with any predecessor architectures.

Note that the above advantages only fully apply to systems that use *standard* IP in their design. Not all audio systems that use computer networking (over Ethernet and/or on RJ-45 connectors) for interconnection are necessarily "true" AoIP systems. Some systems simply use Ethernet as a physical layer with a proprietary data format above it (e.g., Cobranet), while others may use more IP-like formats but with nonstandard protocol variations.

Some of these nonstandard approaches may have offered some value in the past (such as reduced overhead and latency over standard IP networking), but given the capacity, speed, and performance of a properly configured, standard IP system today, the penalties paid by working in a nonstandard environment generally far outweigh any advantages that such variations might provide, particularly when considered over the long term.

Therefore, this book confines itself to the consideration of fully standardized IP-based systems only, both in its generic AoIP discussions and its specific references to the *Livewire* system (which is an example of such a standards-based AoIP approach).

THE GRAYING OF AES3 For digital audio transport today, AES3 is the main alternative to an Ethernet-based system. Invented in the days of 300-baud modems, AES3 was the first practical answer to connecting digital audio signals. But it's now over 20 years old and is showing its age. Compared to AoIP's computer-friendly, two-way, multichannel-plus-high-speed-data capabilities, AES3 looks pretty feeble with its two-channel and unidirectional constraints.

Then there's the 50-year-old soldered XLR connectors and lack of significant data capacity. AES3 is a low-volume backwater, with no computer or telephone industry R&D driving costs down and technology forward. Your 300-baud modem has been long retired; it's time to progress to the modern world for studio audio connections, too.

1.3 IP-ANYTHING

As the world transitions almost everything to IP, we will likely discover even greater synergies as time goes on. The leveraging of IP as a mechanism to use *generalized* systems and transport paths for various *specific* tasks has undeniable appeal. We've seen U-Verse as a prime example, but this argument is also finding favor in a wide range of other industries, from hotel TV systems to health care. Emerging digital TV transmission systems including the new mobile variants are also favoring an IP distribution model.

For broadcast-industry engineers, familiarity with digital networking technologies, including IP, has become a near-requirement of the job anyway (e.g., it's needed in implementing the online services of a radio station), so why not apply this knowledge to studio audio, too?

It's becoming clear that IP is truly the way of the digital media world, particularly for any industry that values connectedness, agility, and cost effectiveness. In the radio environment, it's not an overstatement to say that AoIP is the future of studio audio signal flow. Arguing otherwise is difficult: There is and will continue to be so much development within the IP environment that it only makes sense to harness the power of that effort, while also allowing Moore's Law to have its ongoing effect on hardware cost reductions. The effects of these very forces are being enjoyed by so many other industries today; why not in professional audio as well?

1.4 WHAT'S THE CATCH?

This is not to say that there aren't some challenges. Primary among these is the latency that the encapsulation process of audio data into IP packets can cause. As

you will see in the chapters ahead, on a controlled local network, this can be made sufficiently small to satisfy pro-audio requirements.

Another issue is a simple one of connector standards. Since AoIP generally travels on copper Ethernet cables, the RJ-45 connector is used for all terminations. Some AoIP system implementers, including Livewire, also use RJ-45 for analog and AES3 digital audio I/O with adapter cables converting to XLRs, phone, RCA, etc. While this minimizes the number of different connector types used in a facility and reduces the physical space required for connector panels, some engineers might not be comfortable with this approach.

The need to accommodate and retain compatibility with analog and AES3 digital audio will remain for some time at any AoIP facility. At the very least, live microphone signals will need to be converted from their native audio format. So until microphones and other audio sources come with native AoIP outputs, interface "nodes" will be needed.

Also note that, at least for the time being, AoIP equipment is not yet fully compatible among various vendors. Thus, settling on a single vendor is going to be necessary for each installation.

Engineers installing and maintaining AoIP systems will have to learn enough IP network engineering to have a basic understanding of the technology (or more, if they are so inclined, which will surely be career-enhancing in these times). This book covers most of what is needed for those basics, and suggests other resources to help you go further.

1.5 IMPLEMENTATION AND INTEGRATION

Given the advantages of scale provided by AoIP systems, it makes sense to make the AoIP domain as large as possible within a given facility. This implies that audio signals in other forms should be converted to IP packets as close to the source as possible.

The best place to do this in most studio configurations is at the studio mixing console(s) and/or the central patch bay (i.e., technical operations center, or TOC). Microphone outputs and signals from other "legacy" audio sources can be immediately converted to digital audio form (if they aren't already) and packetized as IP. Once in the IP domain, these signals can be addressed and routed to any other location on the network. This can include destinations within the confines of a facility via LAN, or anywhere in the world via a gateway to the wide area network (WAN).

Another advantage of this approach is that a mixing console can act as a router. In other words, because any input on the console can have a unique IP address, it can be connected to any AoIP source on the network. (Even more amazing to veteran audio engineers is that this can be accomplished even though the entire console is connected to the network via a single Ethernet cable.) A central switching control unit (typically a PC) can assign these I/O connections, or the mixing console itself can have a control interface for this purpose. In addition, standalone hardware

switch controllers can be distributed around the facility, essentially duplicating the appearance and function of traditional router-control panels.

Certainly, the studio mixing console setup can also be equipped with traditional analog (mic/line) or AES3 inputs as well. Because these sources are converted to IP and placed on the network, they are available to any location in the facility that needs them. (See Figure 1.1.)

Consider also how PC-based audio playout/automation systems can be interfaced to such a system. Rather than their audio outputs being directed through PC sound cards to traditional audio inputs, the automation system can be fitted with an IP driver that provides a software interface between the PC audio and the IP network directly in the AoIP domain. This not only maintains high audio quality, but cuts costs in the automation system since no (or at least fewer) sound cards are required. The IP interface can also carry control data and content metadata as well, eliminating the need for separate data links between devices.

Moreover, a *single* IP driver interface between an automation system and an IP routing architecture can carry many independent audio channels (up to 24 stereo for Livewire), whereas a traditional switching system would require a crosspoint (plus wiring) for each sound card input and output. The combined hardware savings (sound cards + crosspoints + wire + installation) accruing in a large facility is likely to be substantial.

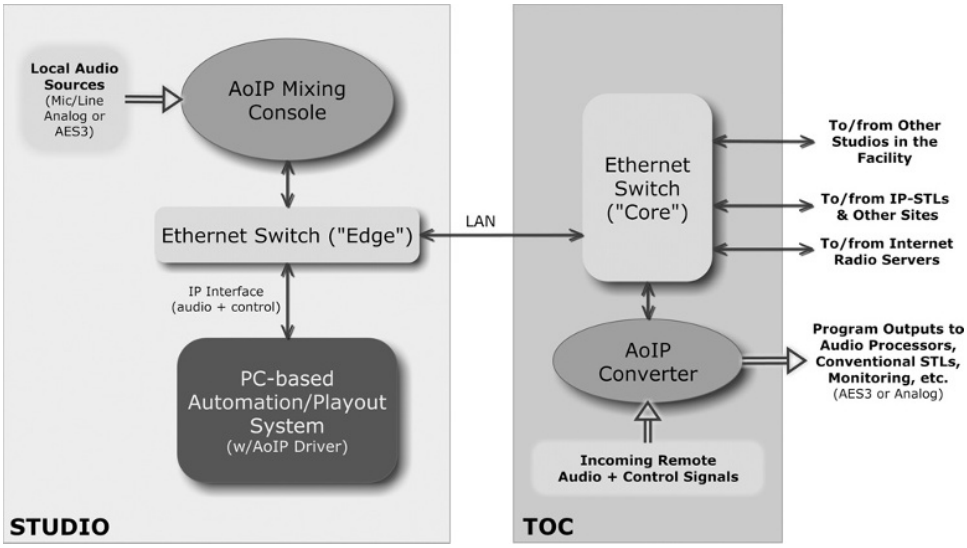


FIGURE 1.1

Conceptual block diagram of a typical AoIP-based broadcast studio facility, showing one studio and a TOC.

As Figure 1.1 indicates, a typical AoIP facility includes multiple Ethernet switches, usually arranged with one large ("core") switch in a central room, and smaller ("edge") switches placed as needed in other rooms around the facility. Such distributed routing intelligence improves performance and also provides redundancy in case of switch failure.

The proliferation of VoIP and other real-time applications via IP have spawned broad implementation of *nonblocking* architecture in Ethernet switches. This approach eliminates data collisions within a switch by ensuring adequate capacity for $n \times n$ connectivity—that is, any input on the switch can always be connected to any output on the switch, under any usage—through the switching fabric. Mission-critical performance is thereby maintained by using Ethernet switches that implement a nonblocking design, and when properly implemented within an AoIP system, switch capacity will never be exceeded.

In some AoIP facilities, the functions of the Ethernet switch can be replaced by an IP *router*. Simply stated, both the Ethernet switch and the IP router perform the same function of getting payload packets to and from their proper locations, but in different ways. Truly standard AoIP systems won't care which is used, however. We discuss the nature and differences of switches and routers in detail in the upcoming chapters, including applications where one or the other may be preferred.

The use of Ethernet switches and IP routers by mission-critical and other high-reliability telecom operations has driven major manufacturers to provide excellent around-the-clock and overnight-replacement support. Note also that as a facility grows, it may need to replace older switches with newer models; the fact that IP and Ethernet are ubiquitous standards means that all upgrades will remain backward compatible. Meanwhile, Moore's Law ensures that as such new hardware becomes available, price/performance ratios will continually improve. It's all good.

The AoIP domain is also extending beyond the studio. Figure 1.1 shows how AoIP is converted back to AES3 (or even analog) for program outputs' connection to conventional studio-to-transmitter links (STLs), but the diagram also indicates that an STL could carry AoIP to the transmitter site (via a WAN or other dedicated link). Whether leased from telco or using a station-operated radio frequency (RF) path, if adequate bandwidth is available, multiple audio channels, control, and metadata can all be carried via IP on the link—bidirectionally, if desired—with minimal latency.

WHITHER THE ETHER? Ethernet is a surprisingly congruent name for a technology initially intended purely for the IT world, but now serving AoIP in broadcast studios. How did that come to be?

Ethernet was named by its inventor, Robert Metcalfe. He had been involved in a radio data network project in Hawaii called ALOHA. The first Ethernet was a bused coax that carried data packets similar to the way ALOHA had sent them over the "ether."

Metcalfe was using the word jokingly. For many years after James Clerk Maxwell's discovery that a wave equation could describe electromagnetic radiation, the aluminiferous

(continued)

FIGURE 1.2

The plaque marks the spot: A small monument to Michelson, Morely, and the ether on the Case Western campus quad.

ENGINEERING HUMOR —cont'd ether was thought to be an omnipresent substance capable of carrying the electromagnetic waves. In 1887, scientists Albert Michelson (the first U.S. Nobel science laureate) and Edward Morely disproved its existence. The ingenious experiment that did so is a cornerstone of modern physics that inspired Einstein's theory of relativity. It was performed at Case Western Reserve University, just down the street from the Telos/Axia head office in Cleveland, Ohio. (See Figure 1.2.)

1.6 AoIP IN USE TODAY

The advantages of AoIP have been well noticed by broadcasters and studio owners around the world. It is fair to say that the engineers designing every new broadcast studio facility built today (and from this point forward) are at least considering the use of an AoIP architecture—and they are increasingly deciding to implement it. Speaking with them afterwards will find almost unanimous agreement that it was the proper choice, and that there's no looking back. In many cases you will also hear that the transition process was far easier than they expected.

The installation of an AoIP system makes many people at the typical enterprise happy, from the chief engineer to the CFO. The total cost of building and operating an AoIP facility is significantly reduced, and yet this can be accomplished without giving up flexibility; in fact it, too, is greatly increased. Operations are often

minimally interrupted as well, due to the small footprint and quick installation of AoIP systems. This is why broadcasters of all stripes, and with budgets large and small, have already moved to AoIP.

In fact, the clientele for this emerging technology almost defies characterization. It includes small independent stations, college radio (including numerous rural and community colleges), ethnic and religious broadcasters, satellite radio services, radio and telecom network operators, content production and broadcast origination sites, and corporate facilities and government agencies, along with some of the largest and most respected stations in the United States and around the world.[3]

Neither is adoption limited to the radio industry. A large and growing variety of professional audio applications are employing AoIP for all of the same reasons that broadcasters have found appealing.

1.7 THE BOTTOM LINE

It's not often that a new technology offers considerable technical improvement, easier installation and maintenance, greatly enhanced flexibility and scalability, *and* reduced cost when compared with its predecessors. Yet these are the attributes of a properly implemented AoIP system.

Broadcasters have always been a cost-conscious lot, and rightly so, but given today's increasingly competitive landscape, efficiencies in capital expenditures and operating costs have become even more critical and desirable.

Meanwhile, it's become quite clear that the radio industry will face substantial change in the near future, and much of it will likely involve quantitative growth in services. More streams, more audio channels, more data, more responsiveness to audience demands, and probably more still, are all on the path that lies ahead for broadcasters. AoIP provides a powerful platform to accommodate these many challenges and opportunities.

[3]IP is a worldwide standard. AoIP is also becoming widely adopted by broadcast and pro-audio facilities across the planet.

CHAPTER

Network Engineering for Audio Engineers

2

You don't need to know most of what's in this chapter to use IP networks for audio. Just as a rock-and-roll roadie can plug XLRs together without knowing anything about op-amps and printed circuit board (PCB) ground planes, you can connect and use IP audio gear without knowing much about packets and queues. But you are reading this book, so either you are the curious sort or you have a need to know what is going on behind the RJ-45. You will be rewarded. You know that fixing tricky problems in the analog world calls for an understanding of the underlying technology; and just the same, an awareness of how data networks function will help you in the AoIP world. Building large and complex systems will require you to have knowledge of how the components interact. All of IT, telephony, and media are going to eventually be built on IP, so now is a great time to get comfortable and proficient with it. Anyway, discovery is fun, right?

We're not going to go so deep that you are overwhelmed with unnecessary details. But there will be enough for you to get a feel for how networks operate, so that you will have a foundation for parsing the lingo you hear from IT folks, and you'll be ready to design AoIP systems, as well as configure them after they are installed. While the topic at hand is general IP/Ethernet data networking, we'll explain a lot of AoIP-specific points along the way that would not be covered in a general networking text.

AoIP systems are built upon standard components, so if you understand data networking generally, you'll be ready for AoIP and the specifics of Livewire. Network engineering is a rich topic, abounding with information and nuance, and it's in constant flux. Fortunately, AoIP uses only a small subset that is easy to learn and use. That is mainly because most of the complexity comes with the IP routing that is the foundation of the Internet. We use only a small piece of that within the local networks that host most AoIP.

Should you want to embark on a journey leading to the top of the IP mountain, bookstores have shelves heaving with networking advice and information. We think

Audio Over IP

doi: 10.1016/B978-0-240-81244-1.00002-X

that this is one of the important advantages of IP for audio. Because of today's data network ubiquity, there are oceans of books and plenty of other learning resources on IP networking in general (although not much that we've found for AoIP specifically, which is why we are moved to write this book). See the References and Resources chapter for some starting points on these deeper networking references. You can probably even find some educational programs in your hometown. If you become really serious, you might explore courses that lead to certified credentials. Cisco has defined a broad range of these, for which testing is conducted by a third party. The Society of Broadcast Engineers (SBE) also offers a certificate in data networking for broadcasting.

2.1 TDM VERSUS IP

Time-division multiplexing (TDM) is a term invented by telephone engineers to describe a system design where a common resource—cable, backplane, radio-frequency spectrum—is divided into channels that are separated by time. This is in contrast to s*pace-division multiplexing* (SDM), which dedicates individual circuits, and *frequency-division multiplexing* (FDM), which separates channels by modulating them into different frequency bands. An example of SDM would be POTS (plain-old telephone service) telephone lines, and an example of FDM would be the analog microwave radios that carried telephone calls in the 1950s and 1960s. Radio broadcasting is, of course, another example of FDM.

TDM was a natural companion to first-generation digitization. Once audio is made into bits, it is quite easy to offset these into timeslots. Simple logic functions comprised of counters and muxes are up to the task. In the early 1960s, engineers had to use the building blocks at their disposal.

The first application of TDM was the T1 line, which was used as a "pair gain" scheme to obtain more channels from existing copper pairs. It was invented in 1962 and remains widely used to this day. It multiplexes 24 channels of 8-bit audio onto two copper pairs, taking advantage of the connection's ability to pass frequencies much higher than the usual 3.4-kHz speech audio. Switching caught-up 20 years later with the introduction of the AT&T 5ESS central office switch in 1982. The line interface part of the 5ESS was comprised of many racks full of card cages holding "circuit packs" that adapted analog POTS lines to a digital backplane where the voice channels were divvied-up into timeslots. Like all TDM equipment, the switching subsection read a full cycle of timeslots into a memory and then wrote them out in a different order. This pattern was followed by other vendors of central office (CO) switches, such as Northern Telecom, Ericsson, Alcatel, and Siemens. Smaller versions were made for PBX applications, as exemplified by the popular Nortel Meridian family.

TELCO TRENDS Alcatel-Lucent is the current heir to AT&T's big-iron hardware division that developed T1 and the 5ESS, and is the inventor of the underlying TDM technology. It is illustrative to note, however, that today Alcatel-Lucent is promoting its IMS (Internet-Protocol Multimedia Subsystem) or its ICS (IP Call Server), which are IP-based technologies, as successors. (In fact, there are *no* TDM products on the Alcatel-Lucent web site at this writing.) The company's customers appear to be following its lead.

Many telcos around the world have announced that they will transition to IP-based systems. Their TDM switches are nearing end of life and they don't want to invest more in a technology that they view as too limiting in the era of the Internet. They want to both reduce their cost for equipment and to be able to offer their clients modern features. Many are already providing or looking forward to providing so-called "triple-play" service: telephony, Internet access, and television over DSL lines. Wireless is in this picture as well. Indeed, the IMS architecture was originally developed by the 3rd Generation Partnership Project (3GPP) for mobile networks. One objective is that users should have a single identity that allows mobile and fixed-line service to interoperate more smoothly.

TDM systems have only audio data within their timeslots. Because there are no signaling or routing instructions in the TDM slots, there needs to be an external mechanism to keep track of where everything is located and to make the needed associations for switching. For the PSTN (public switched telephone network), this is performed by a combination of the logic and storage inside the computers that drive the individual CO switches, and the Signaling System 7 (SS7) protocol that runs between exchanges. The SS7 messaging is carried on data channels independent from those used for speech. In contrast, IP packets "know where they are going" because the destination address is contained within the header of the packet itself. IP routers make all the needed decisions about what to do with the packet based only on the information contained within its header.

In the pro-audio world, AES3 is a TDM transport system. The left and right channels are timeslot multiplexed onto a single cable. MADI (Multichannel Audio Digital Interface) extends the principle to more channels over wider-bandwidth coax cable.

Just as TDM transport led eventually to TDM switching in telephony, first-generation digital pro-audio routers and mixing consoles were also built using TDM technology. The designers of these products borrowed both the "cards-in-a-cage + backplane" and the "timeslots-in-cables" architectures from the telephone industry, and scaled them up to serve the requirements of high-fidelity audio.

2.1.1 Statistical Multiplexing

Statistical multiplexing is the unsung hero of the Internet age. Without it, the Internet would not exist as we know it. Long-haul bandwidth is much more expensive than local area bandwidth. That was the insight of the Internet's creators

that guided many of their design choices. The first Internet was built upon 56 kbps telco data service links. With this paltry speed, there was *always* more demand for bandwidth than was available, and it had to be rationed both fairly and efficiently.

The Internet's designers looked at the switched phone network, and didn't like it much. The engineers who built the PSTN had to build in a lot of expensive bandwidth that was wasted most of the time. Long-distance carriers in the United States love Mother's Day because it motivates lots of revenue-generating calling. (In fact, when a telephone engineer refers to the "Mother's Day Effect," he or she is talking about any event that fills some part of the network to capacity and denies service to many who want it.) Accordingly, the PSTN is designed so that all those doting sons and daughters don't get frustrating busy signals and turn to letter writing. But that means that a lot of precious bandwidth lays unused most of the rest of the year.

The Internet, in contrast, allows multiplexing within and among long-haul links on a packet-by-packet basis, delivering all the available capacity to users at each instant. Thanks to this statistical multiplexing that automatically apportions bandwidth to users, costly long-haul links are used as efficiently as possible.

Imagine if each Web surfer needed to open a 64 kbps channel each time he or she went online. Any time spent reading a page after downloading it would waste all of the channel's bandwidth. Conversely, the surfer's maximum bitrate would be limited to 64 kbps. Would the Web have been practical and successful in this case? You wouldn't have *YouTube*, that's for sure.

2.1.2 IP "Backplane"

AoIP receives no benefit from statistical multiplexing because it needs a fixed and continuous bitrate for each audio stream. That's okay because we are running it over a LAN where bandwidth is plentiful and free. But it invites the question: Why bother with all this IP stuff when we don't receive the main networking benefit, and TDM works just fine, thank you? Well, we've already covered the big-theme reasons for using IP (low cost, common infrastructure, native interface to PCs, in the IT and telephone mainstream, telephone/data/audio integration, etc.), but now we can look at this topic from another angle, with a pure network design perspective.

Think about those circuit packs and backplanes in TDM. They are all proprietary—you can't plug a Siemens circuit pack into a Nortel switch. The same is true in pro-audio—you can't use a card from one vendor's TDM (i.e., AES3) audio router in another's card cage. On the other hand, in an AoIP system, the Ethernet RJ-45 becomes the equivalent of the TDM's backplane, giving the advantage that a wide variety of equipment may interconnect via a standard interface. Also, Ethernet allows the circuit-pack equivalents to be physically distant from the central switch. We can now enclose them in a box and locate them near to the audio inputs and outputs they serve.

2.2 ETHERNET/IP NETWORKS: LAYERING MODEL

You need to be acquainted with the layering concept to know modern data networks. The notion of layers and the open systems they support are central to packet-based network engineering. Because layering is a key to enabling interoperability among multiple vendors and approaches for each function, this design has been a major factor in the growth and operation of the Internet. It's also one of the keys to AoIP generally and Livewire specifically, allowing us to build the audio transport application on top of standard lower layers that were not originally intended for live audio.

The Open Systems Interconnection (OSI) model was developed by ISO (International Standards Organization) and ITU-T (International Telecommunications Union) as a reference paradigm for data networking. Some real applications have been built on it. For example, the integrated services digital network (ISDN) D-channel communication between phones and the telephone network is based on this model. (See Table 2.1.)

But this seven-layer scheme was designed by committee—and it shows, especially when you dig into the details. It has been judged by real-world implementers to be too complicated. As a result, you don't see many products adhering to the details of the standard. But the general idea has well proven its worth. The fundamental principle is that components that work at a particular layer only need to know about and communicate with:

Table 2.1 OSI Reference Model for Network Layering

Layer	Name	Function
7	Application	*Generic Application Functions* File transfer, mail, Web, etc.
6	Presentation	*Data Representation* Independence from local data formats
5	Session	*Process-to-Process Communication* Registration and access control
4	Transport	*End-to-End Communication* Error control, flow control, sequenced delivery
3	Network	*Network-wide Communication (WAN)* Global addressing, routing, fragmentation
2	Data Link	*Local Communication (LAN)* Link addressing, framing
1	Physical	*Physical Channel Access* Line drivers/receivers, encoders/decoders, timing

- The same layer between devices.
- Adjacent layers within a device, using well-defined interfaces.

This is a powerful concept. It means, for example, than an Ethernet switch operating at layers 1 and 2 may be readily exchanged from one model to another without the upper layers noticing any difference. You could just as well upgrade from a 10BASE-T hub to gigabit fiber without having any effect on upper layers. Going further, it would even be possible to change to a different physical network technology entirely. This, in fact, was the main goal of the Internet's design at the outset. When the Internet was first conceived, Ethernet had not yet risen to its current ubiquitous status, and there were plenty of incompatible low-level networking technologies around (ARCNET, Octopus, IBM token ring, StarLAN, DECnet, etc.). The Internet was intended to make all of these invisible to the upper layers so that applications could interoperate.

This works from the top down, as well. You can switch from one web browser to another without anything else in the network needing to change. To use the software engineer's phrase, the differences among the various lower layers have been "abstracted out" to the upper layers.

The top three layers are specified and documented by the IETF (Internet Engineering Task Force), while those pertaining to Ethernet are standardized by the IEEE (The Institute of Electrical and Electronics Engineers).

In Figure 2.1, at the very top is the human user, who wishes to visit a web site. Fortunately, the user has a PC at his or her disposal that is running a web browser. The user enters the text name for the desired site. A name server looks up the IP number address for the site and the browser uses it to request the web page, which the web server at *frog.com* sends to the user. This is simple enough from the top-level perspective. But there is a lot of hidden activity within the layers below, with various standards and technologies coming into play at each.

The PC in the example is interacting with the network at all five layers simultaneously, with each being served by a dedicated piece of the machine:

- *Layer 1*. The physical interface on the network card knows that an active Ethernet link is connected to it and is transmitting and receiving digital bits from the line. It knows nothing of the meaning of the data contained within the bit transitions, not even where frames start and stop.
- *Layer 2*. The logical part of the network card parses Ethernet addresses from the bitstream, so it knows which traffic is intended for itself. It also knows where to send traffic destined for other devices on the local network.
- *Layer 3*. The IP part of the TCP/IP network stack software running in the PC's operating system wraps IP addresses around Ethernet frames to make IP packets. Now the PC is able to send and receive traffic from distant computers on the Internet. When the stack detects that a connection is wanted to a computer that does not exist on the LAN, it sends the traffic to the router, which acts as a gateway to the IP network at the destination side.

- *Layer 4*. The TCP part of the TCP/IP stack software enters the scene. As we'll see, it plays a valuable role in ensuring that data are reliably delivered to applications at the next layer up.
- *Layer 5*. The browser talks the standard HTTP protocol to web servers out on the network to request and retrieve files.

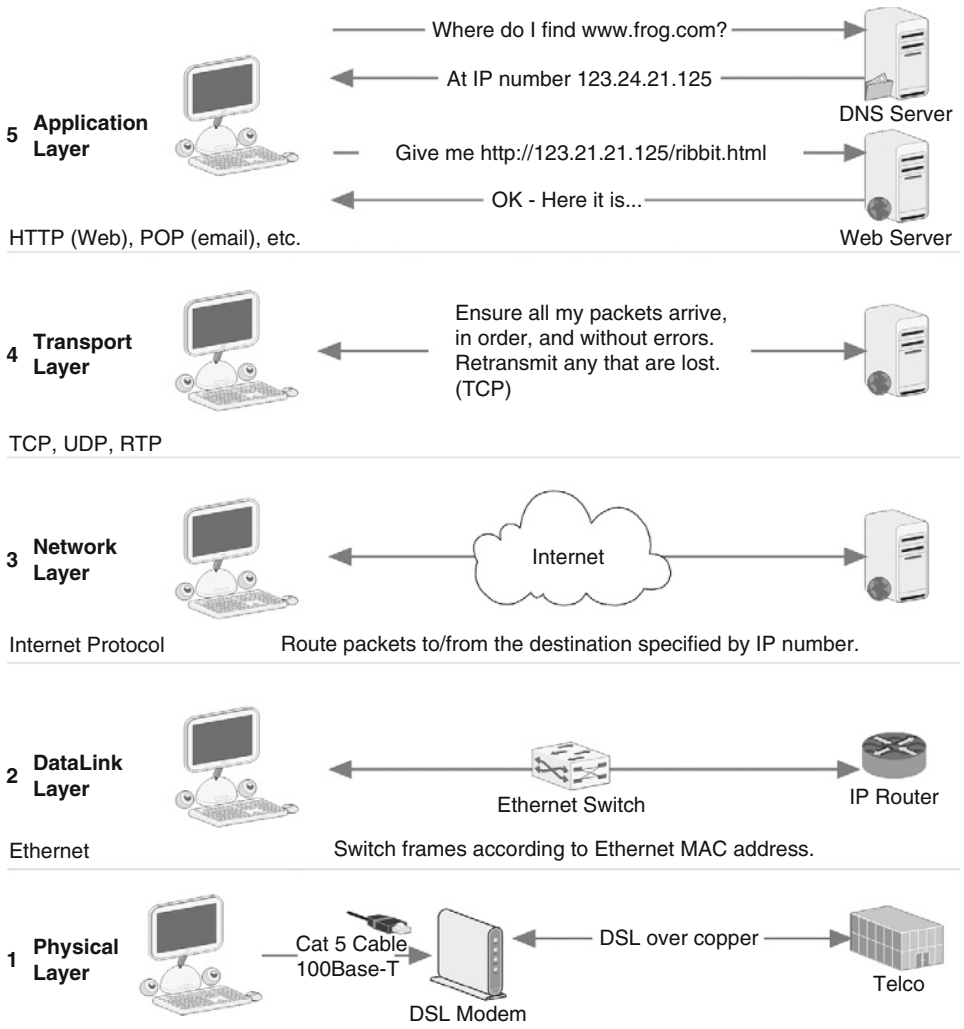


FIGURE 2.1

Here's how the concept of layering has evolved for the real world of today's omnipresent partnership of Ethernet for LANs and IP for WANs.

The browser knows nothing of the Ethernet interface chip, and the chip is blind to the web browser. This is just as the designers intended. Abstraction, isolation, and encapsulation—all elements of an elegant and dependable system design.

Now let's look into each of the layers more deeply.

2.2.1 Layer 1: Physical Interface

This layer is responsible for hardware connectivity. There has been remarkable progress in Ethernet's physical layer over the years, roughly following Moore's Law, which predicts that capacity doubles every two years.[1] We've come a very long way from 10-Mbps bused coax to today's routinely installed gigabit 1000BASE-T networks. With fiber, 10 gigabit is not uncommon, and development is underway for 40 and 100 gigabit. Robert Metcalf, the inventor of Ethernet, says we'll eventually see terabit (1000 gigabit) fiber.

There are WiFi, WiMax, and many other types of Ethernet radio systems. Laser systems are yet another alternative.

All of this illustrates the point that layered architecture encourages innovation. You don't have to change your IP network stack or your email client when your office gets upgraded to a faster network.

ARRIVE ALIVE The original coax-bused Ethernet employed CSMA/CD (carrier sense multiple access with collision detection) to govern how devices shared the cable's capacity. This was retained through the era of passive 10BASE-T hubs. The word *collision* is scary and perhaps makes people think that something bad is happening. It is a reason that Ethernet had picked up a reputation as not being appropriate for real-time media—spread by proponents of alternatives, mainly ATM and Token Ring. There was a time when many people were convinced that anyone needing real-time audio/video would have to install ATM to their desktop.

Thankfully, those days are long over. Starting with 100BASE-T and switched Ethernet, the CSMA/CD functions are disabled. 100BASE-T is full-duplex, with a wire pair dedicated to each direction. Ethernet switches pass traffic only to where it is needed. And modern switches are nonblocking, meaning that the backplane has enough capacity to handle full rate on all the ports at once. Accordingly, there is no sharing, and therefore no reason to use CSMA/CD. In today's Ethernets, AoIP traffic flows smoothly without interruption caused by competing traffic from other devices on the network.

[1] Actually, Moore's Law says nothing about networking, but it might as well. Gordon Moore predicted that the number of transistors on a chip would double every two years, thus either roughly doubling the processing power or halving the cost of computer processing units (CPUs) biennially. (There are many references to this doubling occurring every 18 months, by the way, but Moore insists he didn't say that.) In any case, since network interfaces and digital data radios are made up of chips with transistors, too, Moore's Law indirectly applies to bandwidth growth as well, although telcos and Internet service providers (ISPs) may have something to say about that cost-reduction factor when the networks involved are under their service jurisdictions.

2.2.2 Layer 2: Ethernet and Switching

This layer includes Ethernet's end-station addressing and everything related to it. An Ethernet switch is working at layer 2 because it forwards packets based on Ethernet media access control (MAC) addresses, which are unique ID numbers assigned by the Ethernet-capable equipment manufacturer.

Layer 2 does not ordinarily extend beyond the LAN boundary. To connect to the Internet requires a router. In other words, scaling a layer 2 network means adding layer 3 capabilities.

Officially, the transmission units comprising header and data are called *frames* in layer 2. At layer 3, the correct designation is *packets*. But, since Ethernet frames are almost always carrying IP packets, the word used to describe the combination most often depends on the context or the author's preference. Unless we are referring to layer 2 functions, we usually use "packets" because AoIP audio has the IP header, and because "packets" has become the usual way to describe this sort of network traffic chunk in general parlance. We speak of *packet switching* and *packet networks*, not *frame switching* and *frame networks*.

2.2.3 Layer 3: IP Routing

In addition to Ethernet addresses, each IP packet on a LAN also contains source and destination IP addresses. These are used by routers to forward packets along the most efficient route and to link LANs of different types. When the Internet was invented, there were dozens of LAN technologies in use, and this was an important capability. Now, IP addressing is used both within LANs as a way to access servers from clients, etc., and to connect to Internet resources offsite.

IP in itself is not a complex protocol, but there are numerous capabilities supplied by the other components of the IP suite. The Domain Name System (DNS) removes the burden (to users) of remembering IP addresses by associating them with real names. The Dynamic Host Configuration Protocol (DHCP) eases the administration of IP. Routing protocols such as Open Shortest Path First (OSPF), Routing Information Protocol (RIP), and Border Gateway Protocol (BGP) provide information for layer 3 devices to direct data traffic to the intended destination.

2.2.4 Layer 4: Transport

This layer is the communication path between user applications and the network infrastructure, and defines the method of such communicating. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are well-known examples of elements at the transport layer. TCP is a "connection-oriented" protocol, requiring the establishment of parameters for transmission prior to the exchange of data, and providing error-recovery and rate-control services. UDP leaves these functions to the application.

2.2.5 Layer 5: Application

This layer is generally the only one exposed to users. It includes familiar things like web browsers, audio editors, and email clients, for example. And it's where AoIP devices and software operate.

Application developers decide on the type of layer 4 transport they want to use. For example, database access or Web access require error-free connections and use TCP, while AoIP uses Real-Time Transport Protocol (RTP) layered on top of UDP. (See more about RTP in Section 2.2.9.)

2.2.6 Making Packets

Deep networking engineers care very much about packet construction. The topic will always be covered as a top-level theme in networking textbooks. It will also be an essential part of most Internet standards documents. Why is this detail, though not visible to users, so important? Because by and large, how packets are built defines how the network works and what it can do.

As you might expect, IP packets are constructed in a layered fashion. Figure 2.2 is one representation of the structure for an RTP audio packet. Figure 2.3 examines this structure in more detail, and shows how network engineers usually visualize a packet.

It's not important to know what each of the fields means; the idea is for you to see how a packet is constructed generally. Each of the horizontal gray bars totals 4 bytes. At each layer, devices are operating only with the information contained within the associated header. An Ethernet switch only cares about the layer 2 headers and everything else is just payload. An IP router only "sees" the layer 3 header and doesn't care about the lower-level transport. Applications don't care about headers at all—they just deliver their data to the network and expect to get the data back at the other end. (There are, however, exceptions, such as fancy Ethernet switches that can inspect layer 3 headers for some advanced functions.)

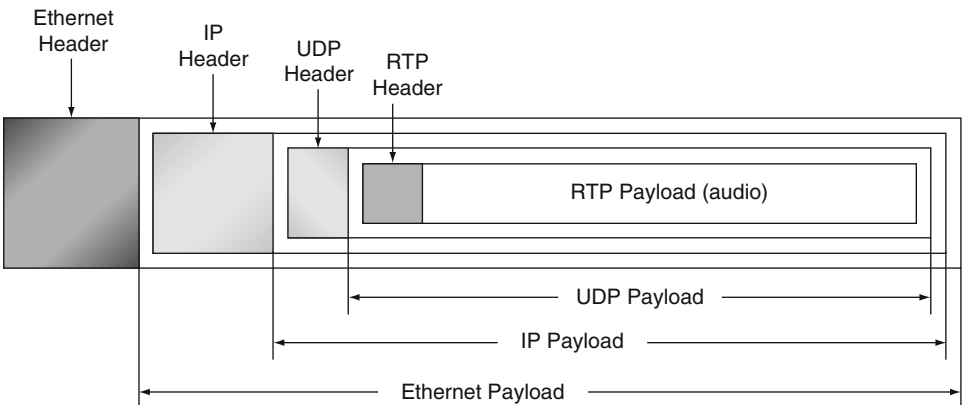


FIGURE 2.2

Layered structure of an RTP packet.

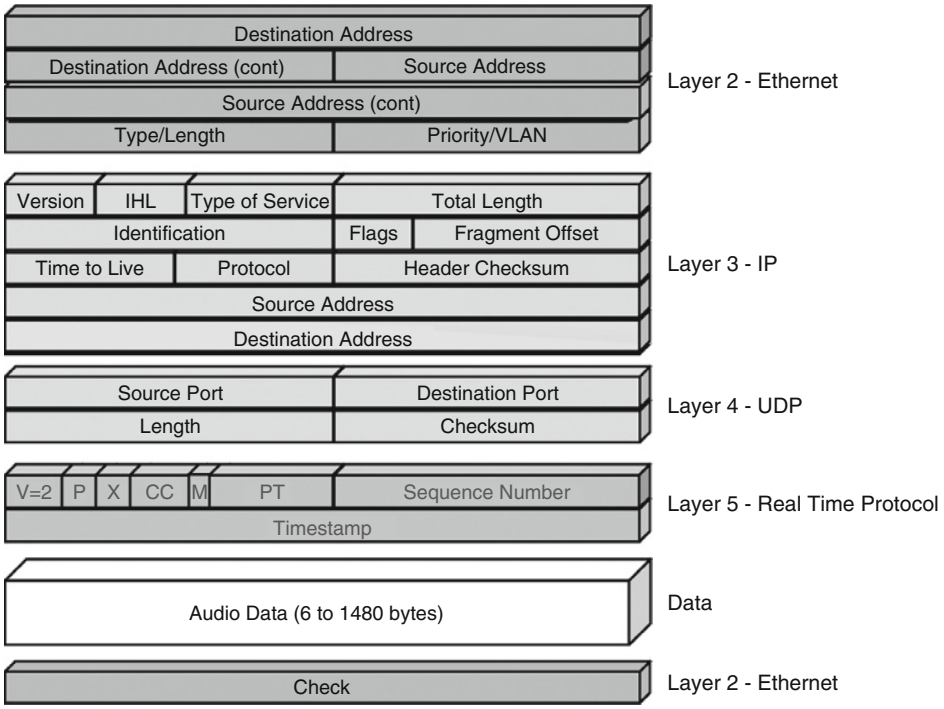


FIGURE 2.3

Detailed view of an RTP packet, showing header contents of each layer.

HOW MANY STATIONS DO YOU HAVE IN YOUR STATION? In careful language, devices that attach to the Internet and have IP addresses are called *hosts*, a name that probably made sense in the early days (they "host" the IP stack and interface). And Ethernet-connected devices are officially called *stations* to keep the radio/ether analogy going.

But what do you call something that is both a host and a station, as almost everything is? *Host* doesn't sound very natural for our audio devices and *station* would be very confusing indeed. Thus, we usually just write *device*, or in the specific case of Livewire, we might write *node*, since that's what we call interface devices in that system.

2.2.7 TCP

Because the acronym TCP/IP is so often written, many people think that the two protocols are necessarily and always joined. This is certainly not so. IP is independent from TCP and may well be used without it. All the same, TCP was invented for the Internet, and is an essential component for its proper operation.

TCP provides two indispensable functions:

- Ensuring reliable reception of data via retransmission of lost packets.
- Controlling transmission rate.

IP routers may drop packets when there is not enough bandwidth on a particular link to transmit them all. Routers also do not guarantee to deliver packets in the same order as they were sent. And there is no protection for bit errors from signal corruption. None of this is a mistake or oversight in the design of the Internet. The inventors knew what they were doing: They wanted the control of any needed correction process to be as close as possible to the endpoints, consistent with the general Internet idea to move as much as possible from the center to the edges.

You need 100 percent reliable transmission for most data files; even a single missed bit could have an unacceptable consequence. TCP gets this done by using a checking and retransmission approach. When a TCP receiver accepts good packets, it sends a positive acknowledgment to the sender. Whenever the sender does not receive this acknowledgment, it assumes there was corrupted or missing data and sends another copy. The receiver holds any data it might already have in its queue until the replacement has arrived. Packets are numbered by the sender so that they can be delivered to the application in correct order. The application always gets good data, but it could be after significant delay.

Transmission rate control is essential for most Internet applications because the bandwidth capacities of the many transmission "pipes" from sender to receiver are almost always different from each other. Further, the available bandwidth to a particular user constantly changes as the demands from the many users sharing the Internet ebb and flow. Think of the old-fashioned case of being at home with a 56k modem connected to your office server via a POTS line. The server and its local network could certainly send data faster than your modem can take it. The same still applies to most devices attached to the Internet today. Meanwhile, in the Internet itself, available bandwidth to any one user is constantly varying. So something needs to slow the sending rate at the server to match both the current network conditions and your modem's ability to receive. That process is performed by TCP through its *flow-control* function. While the details are complicated, the principle is simple: A TCP sender monitors the condition of the buffer at the receiver so it knows how fast the data are arriving and adjusts its transmission rate to maintain the correct average buffer-fill. (See Figure 2.4.)

TCP also has a function called *congestion control*. While this also controls the data transmission rate, it does so with a different mechanism and for a different reason. The retransmission procedure we discussed earlier addresses a *symptom* of network congestion, but not its cause, which is typically too many sources trying to send at too high a rate. To treat the *cause* of congestion, we need to have some way to throttle senders when needed. TCP's congestion control is unusual in that it is a service to the network at large rather than to the individual user. It was conceived as a way to fairly ration network bandwidth to all users. To do this, TCP monitors dropped packets, assuming that lost packets indicate congestion. When a new

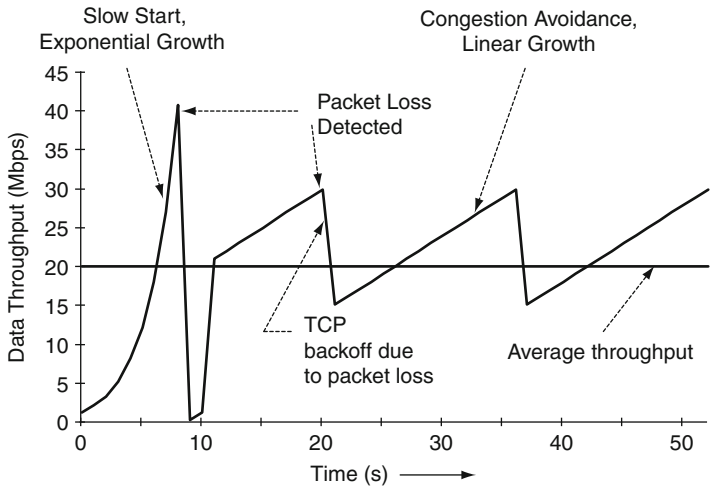


FIGURE 2.4

TCP's transmission rate varies to adapt to network conditions. It aggressively reduces rate when it detects a bandwidth constraint in the network, then slowly increases rate, probing for the limit.

connection is established, a slow-start function causes the rate to start low and ramp up until a lost packet is detected. Then the rate is cut in half and the ramp up begins again. In this way TCP is always probing for the maximum available bandwidth and always adjusting its transmission rate to match. It's really a very slick technique, and one that is well suited to getting the fastest transmission of bursty data over shared links.

TCP is said to be connection oriented because it needs a start-up handshake before communication can start. Following that, a sender and receiver maintain ongoing contact throughout the communication period.

The TCP header adds 20 bytes to the underlying IP header's 20 bytes, creating a combined overhead of 40 bytes.

2.2.8 UDP

UDP assumes that error checking and correction is either not necessary, or it is performed in the application layer, so it therefore avoids the overhead of such processing at the network interface level. Time-sensitive applications often use UDP because dealing with dropped packets (via some sort of error-correction or concealment scheme in the application) is preferable to waiting for delayed packets to arrive (as TCP would do).

UDP has no connection-establishment stage. A UDP sender just blasts the packets out without any regard for the receiver. This means that UDP is suitable for IP multicasting. TCP is not able to do this because IP multicasting doesn't allow senders to

have a one-to-one relationship with receivers. If only one receiver in a multicast transmission is missing a packet, how would TCP deal with sending the replacement? A moment's thought would convince you that rate control also would be impossible in the multicast case. (There have been many schemes invented for so-called "reliable multicast" but they all have trade-offs that make them useful only for certain limited classes of applications.)

The UDP header adds 8 bytes to the underlying IP header's 20 bytes, creating a combined overhead of 28 bytes.

2.2.9 RTP

RTP is layered on top of UDP. AoIP and VoIP are applications carried via RTP/UDP.

Why don't the audio applications use the much more common TCP? The first part of the answer is that it's just not needed. For instance:

- With a LAN's large and reliable bandwidth, TCP's rate-control services are not required.
- Ethernet switches in a properly designed network don't drop packets, so TCP's recovery mechanisms are unnecessary.
- With audio, we have a higher tolerance (compared to most data transmissions) for the very infrequent errors that might crop up, so we don't need TCP's recovery service.

Second, TCP comes at an intolerable cost:

- The show-stopper is that the recovery mechanisms would require an unacceptably long buffer in the receiver when audio needs to be played without pause. Detecting a lost packet, requesting retransmission, then receiving and processing it to insert it into the correct position in the buffer all take time. The buffer would have to be set to a length that can accommodate the worst-case possibility. This is generally many hundreds of milliseconds. For AoIP, delay has to be kept to only a few milliseconds.
- TCP only permits point-to-point connections, not the multicast needed for most AoIP applications.

On the other hand, RTP provides only the few services that AoIP needs: time-stamping, sequence numbering, and identification of the coding method used.

As mentioned, RTP runs on top of UDP. The RTP header needs 12 bytes, the UDP header 8 bytes, and the IP header 20 bytes, thus the total RTP/UDP/IP header overhead is 40 bytes.

2.2.10 Ports

The core IP header has source and destination addresses for the device (host) level, but no way to specify which application *within* a device should be addressed. This is the purpose of ports. Port addresses are specified in a 2-byte field in the UDP and

TCP headers, allowing multiplexing/demultiplexing to as many as 65,536 applications and/or subprocesses.

The port numbers ranging from 0 to 1023 are considered *well-known port numbers*, and are reserved for use by application protocols such as HTTP (which uses port 80).

An important application for ports in audio is for VoIP telephony. We often have servers and gateways that need to process a large number of telephone calls. A unique port number is assigned to each so that they can be properly directed within the device.

2.3 LOCAL AREA NETWORKS

AoIP is normally contained within a LAN, so that is our focus. When audio leaves the safe and secure world of local area networks, it ceases to be AoIP and becomes *streaming media*.

2.3.1 Ethernet Switching

Ethernet switching has caused a revolution in data networking. With switching, each device owns all the bandwidth on its link. No sharing and no collisions. Incoming frames are forwarded only to the nodes that need them.

Despite the power of Ethernet switching, its invention was more akin to falling off a log than sawing one in two. The switch builds up a table of what addresses are attached to what ports, which it does by merely examining the source addresses of sent packets. When frames come in, the switch looks into the table, discovers what port owns the destination, and forwards the data only to that port. In the rare case that no entry exists for an address, the frames are "flooded," or broadcast to all ports, to be sure the intended recipient gets it. If a connection is unplugged or there are no data for a long time, the entry is removed. Pretty simple, eh?

The switching operation described above is for the unicast point-to-point communication that is used for typical traffic such as Web, email, etc. But Ethernet switching supports three communication types:

- *Unicast* means point-to-point, the usual mode for data traffic, as noted.
- *Broadcast* means that a source's packets are sent to *all* receivers.
- *Multicast* means that multiple receivers may "tune in" to the transmission. One source's packets input to the system can be received by any number of output nodes.

Broadcast packets are received by all devices connected to an Ethernet, without distinction or any specific distribution arrangement. The Windows file system, for example, uses broadcasts for a PC to find its partner for a file transfer. A sending device can be 100 percent sure that the intended destination will be found, if it is actively connected to the network. But this comes at a tremendous disadvantage:

Bandwidth is consumed on all links, and all devices have to process the message to determine if it is needed at that location. In effect, broadcasts don't accrue any of the benefits of switching. In a large network, this can be a significant drain on bandwidth and can cause performance to suffer. To avoid this, careful network engineering often breaks up large Ethernets into smaller ones to create isolated "broadcast domains." As we will see, virtual LANs (VLANs) are also a solution. The individual Ethernet segments are then linked together with an IP router, so they appear seamless to users.

DIVIDE AND CONQUER A very rough guideline is that each broadcast domain should have no more than 256 connected devices. A packet sniffer such as Wireshark (see Chapter 8) can be set to filter broadcasts. You can then determine how much bandwidth is being consumed by these broadcasts. If excessive, then the network can be further subdivided.

Multicast is used for Livewire because it lets the network emulate an audio distribution amplifier or router, where an audio source is put on the network once and then can be received by any number of other devices, but only those that need it. With multicast there is no concern for overloading either links or connected devices because the Ethernet switch passes traffic only to ports with devices that have subscribed to a stream. See Section 2.3.5 for more on multicast.

2.3.2 Ethernet Traffic Prioritization

Within a link, we sometimes want to have audio mixed with general data. This happens, for example, when a delivery PC is playing audio and downloading a file at the same time, or when an AoIP device is sending and receiving audio and control messages simultaneously. To be sure audio always flows reliably, AoIP can take advantage of the priority functions that are part of the switched Ethernet system.

Compared to the original, modern Ethernet has an additional 4 bytes of data inserted into the frame's header. One field provides a 3-bit priority flag, which allows designation of eight possible values, as shown in Table 2.2.

Highest-priority packets have first call on the link's bandwidth. If high-priority packets are in the queue and ready to go, the lower-priority ones wait. If there is not enough bandwidth for both, low-priority packets will be dropped, but this is not a problem, as you will soon see.

Figure 2.5 shows only two queues, but the idea is the same for four or eight. Switches used for Livewire must support a minimum of four queue and priority levels. Some low-end switches include no priority support, or may support only two queue levels.

If you have multiple switches in a hierarchical configuration, the priority information is carried automatically to all the switches in the system.

Table 2.2 Ethernet Priority Assignments Used for Livewire Systems

Priority Level	IEEE Recommendation	Livewire Assignment
7	Network control	—
6	Reserved	Livewire audio
5	Voice	VoIP telephone audio
4	Video conferencing	—
3	Call signaling	Livewire control and advertising
2	High-priority data	—
1	Medium-priority data	—
0	Best-effort data	—

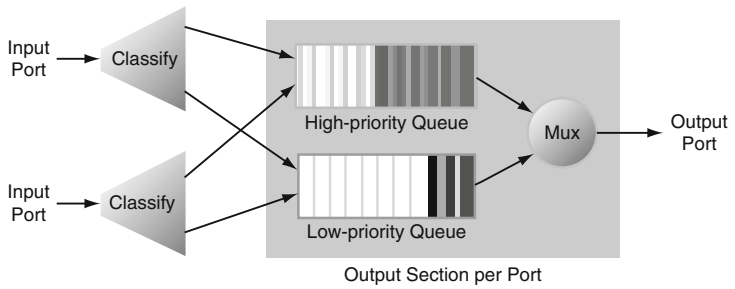


FIGURE 2.5

Ethernet switches that support priorities have two or more output queues.

2.3.3 The Role of TCP for Audio on LANs

TCP was invented for the Internet and is essential in that environment where a user's available bandwidth is variable and IP routers routinely drop packets. But TCP is a key LAN technology as well. While there is plenty of bandwidth on a LAN, multiple fast PCs sending files at full speed to a particular link could still swamp it. Some networked devices can only handle a slow data rate. Thus, TCP's rate adaption is required on LANs.

There is a very low—but not zero—chance that a packet could be damaged or lost, say from someone firing up a mobile phone close to a cable. Our tolerance for this kind of fault is pretty much zero, however. Files must be bit-for-bit accurate so that the office manager's purchase order system doesn't mistakenly order a million boxes of paperclips. So TCP's error-detection and repair service is valuable on LANs.

While TCP is not used to transport audio, it nevertheless plays a role in AoIP. It lets you share high-priority audio with best-effort data on a single network link. Consider this case: A PC is used to host an audio delivery player. The player always is playing AoIP audio into the network. Sometimes it needs to request a file from a server. A fast PC could use all the capacity on a link during the file transfer, competing with the audio stream and causing dropouts. We have a ready answer to this problem, and we have just seen it: prioritization. Audio packets (sent by UDP) are assigned higher priority than general data, so they are never dropped in the switch, but other data packets (sent by TCP) are. That causes TCP to reduce the rate of the file data's transmission to that which can fit in the link's remaining bandwidth after the audio streams are accounted for. TCP automatically finds how much bandwidth it can use, and adjusts its rate naturally to match.

There is another solution to successfully playing an audio file while downloading another. Install two network cards in the PC, one for audio and the other for data. Then each has full call on its link bandwidth. It also provides the possibility to have two completely independent networks, with one for audio and another for general data.

Don't confuse any of this with how audio and data are shared on the overall system. It is the Ethernet switching function that allows the network to be shared, since general data never even get to a port connected to an audio device.

2.3.4 VLANs

The virtual LAN is a technology that came to Ethernet along with switching. It is a way to have "virtually" separate LANs on a single physical network—in other words, multiple networks over one set of wires and routing hardware.

Remember those broadcast packets? They go to all devices, even with an Ethernet switch in the picture. If there are a lot of computers on the network, there could be a lot of traffic generated by these transmissions. VLANs can be used to contain broadcast packets, since they are not propagated outside of their assigned VLAN.

VLANs can also be used for security. If the Livewire network is on a different VLAN than the Internet, a hacker would not be able to gain access to your audio streams or send traffic on your audio network.

In an AoIP network that is shared with general data, VLANs offer protection against a computer having a problem with its network software or interface card. The Ethernet switch can be configured so that the ports to which general computers are connected are not able to forward packets outside of their assigned VLAN, so they can never reach Livewire audio ports. Finally, VLANs protect against the rare case that an Ethernet switch has not yet learned an address and has to flood all ports on the network until it knows the specific destination.

A router must be used to bridge the traffic between VLANs while providing a firewall function.

PHYSICALLY CHALLENGED ROUTERS A router that bridges VLANs is sometimes called a "one-armed" router because it has only one Ethernet port, rather than the usual two or more. There are also "no-armed" routers that are increasingly being incorporated inside Ethernet switches. These provide an internal routing capability that can be used to bridge VLANs without any external boxes.

When the VLAN information embedded in the Ethernet frame is used to direct the switch, this is called a *tagged VLAN operation*. But some devices are not able to do this. In that case, the switch itself has to insert the tag, which is called a *port-based VLAN*. All frames that enter from a particular port are tagged with a certain value, defined by your one-time switch configuration.

There is a special case: Frames tagged with `VLAN=0` are called *priority frames* in the IEEE's 802.1p Ethernet standard. They carry priority information, but not the VLAN ID. The switch will translate to whatever VLAN is default for that port. This is useful if you want to use a port-based VLAN assignment at the switch, rather than tagging from the Livewire device.

Many switches allow a combination of port and tagged VLAN on a given port. To use this approach, you would assign a default VLAN to the port, and frames with either no tag or with `tag=0` will then go to this default VLAN. Tagged frames with a value other than zero would override the default.

It would be possible to use both a port-based and tagged VLAN assignment in a system. For example, you use Livewire node configuration to put all your audio devices onto VLAN 2. But since some PC operating systems don't support tagged VLANs, how would you connect such a PC for configuration and monitoring? Using the port-based assignment, you can set a port to be always VLAN 2 and plug your PC into it.

2.3.5 Ethernet Multicast

AoIP audio is multicast because we want a source to be available to multiple destinations, just like traditional audio distribution using distribution amplifiers (DAs) and audio routing systems.

AoIP sources send their streams to the nearest Ethernet switch using addresses reserved for multicast. These are special "virtual" addresses that are not assigned to any physical port. Audio receivers can listen in with a party-line fashion by sending a request to the switch using the IGMP protocol described in the next section. The request specifies the address for the desired audio stream. Upon receiving the request, the switch begins sending the audio to the port that is connected to the device that made the request. If there is no request for a source, the AoIP stream simply stops inside the switch and no network bandwidth is wasted.

A multicast is flagged in the first bit of the 48-bit address, with a 1 in this position signifying a multicast. That means Ethernet has set aside half of all its addresses for

multicast—enough for 140,737,488,355,328 connections, which should be enough for even the very largest broadcast facility! The designers clearly had big plans for multicast that have not yet been realized.

In the unusual situation that IP routing is used to complement Ethernet switching (such as might be the case in a very large installation), the audio streams are multicast at both layers 2 and 3 using standards-based procedures. Over 8 million unique IP multicast addresses are available. Each IP multicast address is mapped to an Ethernet multicast address according to an IETF standard. (See Chapter 3 for more on Ethernet switching versus IP routing.)

With Livewire, the addresses are automatically and invisibly calculated from much simpler channel numbers. Livewire devices will have a manually configured unicast IP number. But the audio uses only multicast, taking one address for each audio source.

2.3.6 IGMP

For multicasts, we need a way for an audio receiver to tell the switch it wants to listen to a particular channel. Internet Group Management Protocol (IGMP) serves this purpose.

IGMP is part of the IP suite and is a layer 3 function that was designed to communicate with IP routers to control IP multicasts. IP routers include an IGMP querier function, and almost all high-end Ethernet switches include an "IGMP snooping" feature. Audio devices that want to receive a stream send a *join* message to the querier in the router specifying the IP address of the desired source. The switch listens in and turns on the stream.

Knowing that some users will want to use multicast on LANs without involving an IP router, high-end Ethernet switch manufacturers usually include a querier function in their products. A switch with IGMP querier capability will become a querier in the absence of any other querier on the network, so no IP router need be in the picture.

Systems may be built with multiple switches in a tree structure. Usually the core switch will provide the querier for all the devices in a system. But switches at the edge can back up the core, keeping islands alive in the event the core fails. With proper configuration, an edge switch can automatically start being a querier when it detects that the core has stopped working. Then later, the switch would cease being a querier when it detects the core or another querier has begun to operate.

IGMP uses three types of messages to communicate:

- *Query*: A message sent from the querier (multicast router or switch) asking for a response from each device belonging to the multicast group. If a multicast router supporting IGMP is not present, then the switch must assume this function in order to elicit group membership information from the devices on the network.

- *Report (Join)*: A message sent by a device to the querier to indicate that the device wants to be or is a member of a given group indicated in the report message.
- *Leave Group*: A message sent by a device to the querier to indicate that the device wants to stop being a member of a specific multicast group.

An IP multicast packet includes the multicast group address to which the packet belongs. When an audio device connected to a switch port needs to receive multicast traffic from a specific group, it joins the group by sending an IGMP report (join request) to the network. When the switch receives the join request for a specific group, it forwards any multicast traffic it receives for that group through the port on which the join request was received. When the client is ready to leave the multicast group, it sends a leave group message to the network. When the leave group request is detected, the switch will cease transmitting traffic for the designated multicast group through the port on which the leave group request was received (as long as there are no other current members of that group on the port).

The IGMP query message polls devices to confirm that they are still alive and want to continue receiving the multicast. This process removes feeds to devices that have been disconnected or switched off.

An interesting nuance is that the query message specifies a maximum response time for the replies. Responding devices are expected to randomize the time they wait to answer, spreading the traffic evenly up to the response limit. Were this not so, there would be a large burst of traffic as each device responds immediately to the query. Since the Ethernet switches that have to process the messages usually have low-power CPUs for doing so, they could become overloaded. The default is 10 minutes, which should be okay most of the time. By varying this value, you can tune the "burstiness" of the responses, with larger intervals spreading the messages more broadly.

The default time for the querier to send query messages is 125 seconds. This, too, may be tuned to reduce the amount of network traffic.

2.3.7 ARP

There is a need to translate between IP and Ethernet addresses. Consider a server sending data to a machine it knows only by IP address. To communicate, it has to generate an Ethernet frame including the Ethernet destination address corresponding to the desired IP address. To do this, every IP-based device has an Address Resolution Protocol (ARP) module, which takes an IP address as input and delivers the corresponding Ethernet address as output. It maintains a local table with the associations. When it encounters one it doesn't yet know, it broadcasts an ARP query packet to every device on the LAN and the device that owns the specified IP address responds with its Ethernet address. If there is no owner, the packet is presumably intended for an offsite device and is sent to the gateway address of a router. How does the transmitting device find the router's Ethernet address? With ARP, of course.

ARP TABLE Entering arp -a into Windows' command prompt will give you the current list of IP addresses and associated Ethernet addresses—the *ARP table* for that machine.

2.4 WIDE AREA NETWORKS AND THE INTERNET

We said we weren't going to delve much into WANs, since AoIP is intended to be confined to LANs. On the other hand, when we get into VoIP telephony and IP codecs later, we are unavoidably talking about the big wide world of WANs and the Internet, so it will be necessary to touch on the topic here.

By the way, some WANs are, in effect, LANs. For example, you could use an Ethernet radio to extend your studio LAN to the transmitter site. Because the radio works at network layers 1 and 2, there is no IP routing involved. There will also be very good quality of service. So, while the distance is certainly "wide," the two sites are linked in such a way as to effectively comprise a LAN.

2.4.1 The Internet

The original motive for the development of the Internet was to link up the local networks at a few university and military computing centers. Clearly, the designers' goals were modest in light of what their unpretentious project has since become!

But, thankfully, the spirit of the pioneer designers lives on. They were "get on with it" types who preferred to write code and try it out in the real world, rather than engage in lengthy theoretical debates. More important, they wanted to construct the network in a way that was open and extensible, rather than locking it down in a closed and constrained fashion. (Tellingly, Internet standards documents are called RFCs, "Requests for Comments.") There is not much chance the designers had audio/video streaming in mind back when the Internet was getting started, but their approach to the design lets us do it today.

For our purposes, as audio engineers, the Internet has two characteristics:

- It's everywhere.
- It's unreliable.

The first offers tantalizing opportunity; the second, frustration.

Internet service providers (ISPs) are not able to offer any guarantees with regard to quality of service because most of the time they don't control the end-to-end path. It is atypical that both ends of a connection are being served by the same ISP. The common case is that traffic must traverse at least two vendors' networks, with an Internet Exchange Point (IXP) or one or more third-party networks interposed between the two. IXPs are notorious for being overloaded, causing dropped and delayed packets. The third-party networks are often overloaded as well.

VIRTUAL BREADCRUMBS To see the route a connection is taking, you can use the application called *trace route*. On Windows PCs, open the command line window and type **tracert** followed by either a domain name or an IP number. You will soon have a list of all the router nodes involved in the path and information about the delay caused by each.

Economics plays a starring role in shaping the characteristics of the Internet. The Internet is cheap and unmetered precisely *because* it offers no guarantees. As we've seen, the Internet relies on statistical multiplexing, with the expensive long-haul lines that form its backbone being dynamically shared. You might have a 4-megabit DSL line, but that definitely does not mean that you will be assured anything like this data rate end-to-end. It would be prohibitively expensive and impractical to build a network that could handle all subscribers running flat out at their full rate. This would be like the city zoo being sized to have room for everyone in town visiting on the same day. (Perhaps motivated by a particularly compelling Discovery Channel episode?) This is what statistical multiplexing is all about—making assumptions and observations about the nature of typical traffic patterns that can be used to guide a network's design, all within the framework of the ever-present, keep-the-customer-satisfied versus keep-the-cost-down trade-off.

There is nothing wrong with this. Indeed, as we've said, the Internet would not exist without taking advantage of this tactic. But it does mean that you can never count on the Internet as a 100 percent reliable transport for audio. It can reach "good enough" status for some purposes, but only when audio devices are designed for the inescapable unreliable network conditions. We'll meet such devices in Chapter 7.

You can improve your chances of achieving smooth-flowing audio by ensuring that both ends of a transmission are being delivered by a common ISP, thus avoiding the troubles caused by IXPs and third-party-caused bottlenecks. And some ISPs are better than others, surely. Each has its own idea as to where to set the satisfaction versus cost compromise. But to get a guarantee, you will need to arrange some kind of private or "virtual" private network.

2.4.2 Private WANs

Private WANs are more common than you realize. In fact, you probably have at least one in your facility. Do you have a satellite feed from NPR or CBS/Westwood One? This is a private one-way IP network. Are you conveying IP streams from your studio to your transmitter site, such as for your HD Radio exciter, via a radio or telco link? This is another example of a private network.

Private IP networks are traditionally built over telco-leased lines, such as a T1 that is not channelized for voice, ranging up to an OC3 (optical carrier, 155 Mbps) fiber.

2.4.3 VPNs

A private WAN has obvious advantages over a public network like the Internet when it comes to reliability, performance, and security. But maintaining a WAN and paying for leased lines can be expensive.

A virtual private network (VPN) is a private network that uses a public network (usually the Internet) to connect remote sites or users together. Instead of using a dedicated physical connection, a VPN uses "virtual" connections routed through the Internet to connect one private network to another.

Network-based VPNs may be leased ready-to-go from an ISP. The vendor would provide each of your connected sites some kind of interface box that has an Ethernet jack on it. You connect your network to it, and that's pretty much it.

Customer-based VPNs are a lower-cost option. You would buy boxes from a company such as Cisco that perform encryption, firewalling, authentication, and tunneling (encapsulating one packet inside another). These *tunnel interfaces* would connect to the Internet on one side and to your LAN on the other.

The overwhelming majority of current VPN implementations are encrypted VPNs for security purposes. Encrypted VPNs use a **secure channel** or a tunnel for data transmission between VPN sites. This involves the following processes:

- Authenticating the two endpoints of a secure channel so that only authorized users have access to an organization's network.
- Encrypting users' packets and encapsulating them into another packet that seems "normal" to the ISP's network equipment. Therefore, encrypted traffic is absolutely transparent to a provider network and is served the same way as any other traffic.

IPsec (Internet Protocol Security) and SSL (Secure Sockets Layer) are the most popular protocols used nowadays for establishing secure channels. PPTP (Point-to-Point Tunneling Protocol) is also used, although it is less popular, probably because it is a Microsoft proprietary protocol, whereas the first two are IETF standards. All these technologies encapsulate secured data into IP packets.

An important distinction: Secure channels protect an organization's data while they are being transported through public networks, while firewalls protect an organization's data and equipment from external attacks. (See Section 2.4.8 for more on firewalls.) Secure channels also provide some additional protection against external attacks because they do not accept traffic from nonauthenticated users.

For a remote-access VPN that serves individual users, tunneling usually uses PPP (Point-to-Point Protocol). Usually L2TP (Layer 2 Tunneling Protocol) is used to complement the core PPP.

2.4.4 DNS

The Domain Name System (DNS) is a naming system that can be used by any device connected to the Internet. It translates text names meaningful to humans into IP address numbers. Thus, it can be thought of as the Internet's "phone book," translating names such as *www.roamingtigers.com* to a number like 108.72.200.1.

DNS distributes the responsibility of assigning domain names and mapping those names to IP addresses through a system comprised of a large number of servers dispersed throughout the Internet. Authoritative name servers are assigned to be responsible for their particular domains, and in turn can assign other name servers for their subdomains. There are a few (13 at the time of writing) root servers, but few users will contact them directly. This approach has made DNS fault-tolerant. It has also avoided the need for a single central register to be continually updated.

To avoid having to pass a request all the way back to the authoritative server or to a root server, outlying DNS servers along the branches leading to the user will cache (i.e., save in memory) all of the requests that flow through them. This reduces both server load and network traffic, but has the consequence that name changes are not propagated to all users at once. Depending on configuration, a DNS server might only check for updates each day. Network administrators use 48 hours as a rule-of-thumb time for when a new name/number association will be fully propagated. There are also DNS caches within PCs, both in the DNS resolver part of the IP network stack and as a part of a web browser.

DNS lookup takes time, ranging from a few to hundreds of milliseconds depending on the path between the user and the server, the loading of the server, whether there is a nearby cache, etc. Therefore, sometimes it makes sense to bypass DNS and use the IP address directly in order to reduce delay.

DNS also offers something called *host aliasing*. This lets you assign more than one name to a site. For example, the main Telos DNS name is *telos-systems.com*, but we have aliased to *telossystems.com* and *zephyr.com* to help people find us should they try something different than the official name.

DNS has a feature similar to a telco's rotary use of multiple lines. You can assign multiple IP numbers to a name, and DNS will rotate traffic through each. In addition, *virtual hosting* lets a single server host multiple web sites with different names.

Smaller-network users usually just employ the DNS server provided offsite by their ISP. Users with larger networks often have an onsite DNS server to cut lookup time and reduce traffic across the link to the ISP.

The main purpose of DNS is to find domains across the Internet, but it is sometimes used to identify individual machines within a LAN. This is not the usual way machines on LANs are identified to each other. For example, the Windows networked file system has its own naming scheme that it uses to identify participating computers. DNS is needed in the case that people outside of a LAN need to find a particular machine within it. For example, mail.pizzi.com could be how Skip connects to his email server.

2.4.5 DHCP

Dynamic Host Configuration Protocol lets IP devices get configuration information automatically from a server. Using DHCP, a user need not enter an IP number, gateway, network mask, and DNS server values.

Upon connection, the client device broadcasts on the IP subnet to find available servers. When a DHCP server receives a request from a client, it reserves an IP

address for the client and extends an IP lease offer by sending a message to the client. This message contains the client's MAC (physical) address, the IP address that the server is offering, the subnet mask, the lease duration, and the IP address of the DHCP server making the offer. Normally, the client would accept the offer and inform the server that it has done so.

2.4.6 IP Broadcast

IP's broadcast capability is similar to Ethernet's. When an IP device sends a UDP packet with the broadcast destination of 255.255.255.255 (or a subnet broadcast address), all devices will receive it.

2.4.7 IP Multicast

We met Ethernet multicast previously. Multicast at the IP layer works pretty much the same. A source directs a stream to an IP address that is reserved for multicast-only use. Devices that want to tune in do so via the IGMP protocol described earlier in the chapter. The nearest IP router receives the request and initiates a cooperative procedure among all the routers along the way to the source to form a connection path. IGMP is only used from the audio receiving device to the nearest router. Different protocols such as PIM (Protocol Independent Multicast) or DVMRP (Distance Vector Multicast Routing Protocol) are used between the IP routers that form the transmission tree. Note that the audio source device has only to send its stream to an appropriate IP address at the nearest router. If there are no receivers requesting the stream, the stream just stops at that first router. When someone asks to join the group as a listener, the source device knows nothing of this, nor of any of the detail of router trees, etc.

Despite the impressive recent growth of audio and video streaming that could benefit from it, IP Multicast has yet to be widely deployed on the Internet. The charitable view is that this is owing to the difficulty of coordinating it across multiple ISPs, both from a technical and business perspective. A more skeptical take is that ISPs enjoy being able to sell big pipes to the originators of live feeds, who are forced to pay for multiple unicast streams.

Leaving the Internet aside, it would be possible to build a routed LAN or private WAN for AoIP or Internet Protocol television (IPTV) applications. This makes little sense for the typical studio application because it would add a lot of unnecessary complexity over staying with layer 2 for this purpose. But for a very large facility, it offers a more controlled way to distribute multicasts. A common actual application is IPTV service, which could have a very large number of subscribers.

2.4.8 Firewalls

Firewalls protect private networks such as your LAN from unwanted traffic that could be malicious. These can be housed in standalone boxes, but are often combined with the IP router that links a LAN to the Internet.

First-generation firewalls employed simple packet filters. These act by inspecting individual packets. If a packet violates the configured rules, the packet filter will either drop the packet, or reject it and send error responses to the source. This type of packet filtering pays no attention to whether a packet is part of an existing stream of traffic. It stores no information on the connection state. Instead, it filters each packet based only on information contained in the packet itself, most commonly using a combination of the packet's source and destination address, its protocol, and the port number.

The second generation brought us *stateful firewalls*. These are able to look past individual packets to streams or packet series. The term *stateful* comes from the firewall's maintaining records of all connections passing through it. Thus, it is able to determine whether a packet is either the start of a new connection, a part of an existing connection, or an invalid packet. Though there is still a set of static rules in such a firewall, the state of a connection can in itself be one of the criteria that trigger specific rules. This type of firewall can help prevent attacks that exploit existing connections, or certain denial-of-service attacks.

Third-generation firewalls go a step further, having awareness of the application layer. The key benefit of application layer filtering is that it can understand certain applications and protocols (such as web browsing or DNS lookups), and can detect whether an unwanted protocol is being sneaked through on a nonstandard port or being abused in some other harmful way.

2.4.9 NATs

Once exotic, network address translators (NATs) are as common as flies these days. You probably have one at home (Figure 2.6). They are widely used on broadband Internet connections to allow more than one computer on a LAN to share the single IP number given to you by your Internet provider. It costs more to have more IP

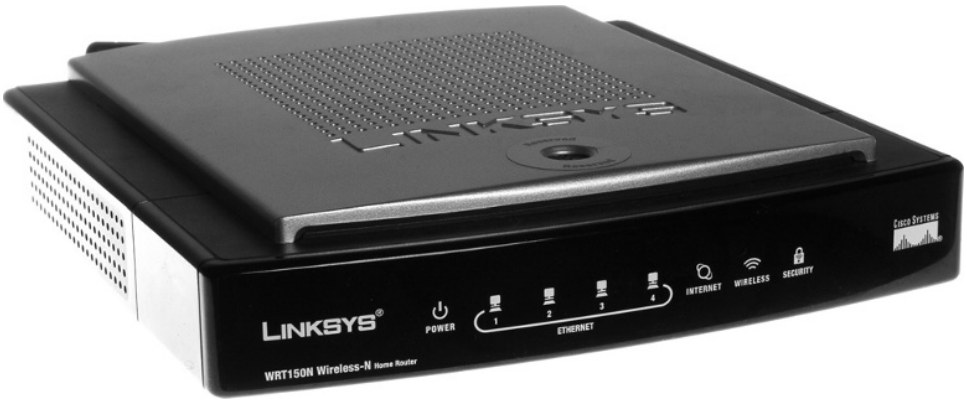


FIGURE 2.6

A home NAT used for sharing a broadband Internet line. It includes an IP router, Ethernet switch, and WiFi.

numbers, usually changing the service category from home to business. So, economics is the main motive for installing a NAT. But security is another essential reason. NATs serve as effective firewalls. Because they hide individual computers inside the LAN, hackers outside the LAN are unable to find and target them.

In addition to address translation, all NATs include basic IP router functionality, including a rudimentary firewall. All connections must originate from a computer on the inside. Since unsolicited incoming traffic can't get through, you have a kind of "double firewall" effect.

Most firewalls and NATs are "symmetric," meaning that only when a packet stream is sent from the inside toward the outside does the NAT/firewall open a return path.

Usually we are happy to have the protective services of firewalls and NATs, but sometimes they cause us trouble. For example, when we want an IP codec to call another that is located inside a firewall, the firewall can block the connection. As we'll explore in Chapter 7, there are ways to deal with this.

Outside of their intended application, we've had success using NATs as simple routers to bridge VLANs and for other purposes. Good ones make fine low-cost routers, as long as the throughput is satisfactory for the specific case.

ADDRESSING THE UNIVERSE It is probably a good thing that home Internet users are widely using NATs because doing so has helped to keep IPv4, the current IP, from running out of addresses. A few years ago, at least one study predicted all IPv4 addresses would be used up by 2008. Without NATs, that prediction might have become reality. In a world where TVs, toasters, and maybe even lightbulbs are going to need an IP address, the depletion of addresses is going to accelerate. Adoption of IPv6 will eventually solve this problem slam-dunk. It increases the address space from 32 bits to 128 bits, which is enough for 100 undecillion devices. Say, you've never heard of *undecillion* before? That's because you've never had the need to count that high. It's equivalent to 10^{38}, which is a very large number, indeed. There are around 10^{28} atoms in the human body, as one reference point.

Another estimate postulates that if the entire surface of the Earth were covered by computers with no space in between them, and each computer was in turn stacked with other networked devices to the height of 10 *billion* apiece, each could have its own unique IP address, and this would only use up about *one-trillionth* of the addresses available. So the IPv6 address space should last for awhile (although computers on other planets may have to wait for IPv7).

2.5 QUALITY OF SERVICE

In the context of IP networks, the phrase "quality of service" (QoS) has a specific meaning, describing the quality of the network with regard to the following:

- Bandwidth
- Dropped packets

- Delay
- Jitter

These are all particularly important for audio applications. Although Web surfing, file transfers, email, and the like are all tolerant to big QoS impairments, audio requires a constant flow of packets, all of which must arrive on time and in proper order. Buffering can correct for most QoS problems, but low-delay audio requires very short buffers.

On LANs, it's no problem to achieve excellent QoS. On private WANs, it's also not much of a problem. On VPNs, however, QoS starts to be an issue. And on the Internet, it is the overriding concern.

Statistical multiplexing has its downside, and we see it here. Because all links that make up the Internet are shared by an unpredictable number of users, with unregulated demands on bandwidth, a user can never be sure what is available to him or her at a given instant.

Let's examine each of the bulleted points above, in turn.

2.5.1 Bandwidth

There has to be enough bandwidth consistently available on the network to support the desired audio transmission bitrate. Including header overhead, for low-delay uncompressed 24-bit/48-kHz audio, this is around 3 Mbps per audio stream. Compression (audio coding) can take this down to less than 100 kbps, but at the cost of delay and audio quality. The network has to be able to convey audio streams at the required rate *consistently*, never falling below the minimum. An *average* bandwidth guarantee is no use to audio applications that need low delay. That's because short receive buffers cannot ride out bandwidth variations.

2.5.2 Dropped Packets

As you've seen, IP routers are allowed to drop packets as a normal part of their operation. This is caused by link overloading, so it depends on a range of factors that are sometimes correctable by careful network engineering when the network is under your control, but are unpredictable and potentially troublesome when the network is being run by someone else.

Audio uses RTP transmission, so there is no lost packet recovery. Any dropped packets are going to result in audio pops and/or dropouts. There is no concealment mechanism for pulse-code modulation (PCM) coding that would cover missing packets.

Recall that low-delay audio cannot withstand the delay that lost-packet retransmission imposes. And other recovery techniques using FEC (forward error correction) also add delay due to the time interleaving that is required to get any significant benefit.

We revisit this topic in Chapter 7. Some codecs are able to effectively conceal 10 percent or even 20 percent random packet loss. But this is no longer low-delay AoIP, by any means.

2.5.3 Delay and Jitter

Delay is caused mostly by the IP routers along the path from the source to the destination. In each router, packets must be inspected and decisions made about what to do with them. This takes time. At minimum, the full packet has to be brought into the router's memory so that the checksum can be verified, creating a baseline delay. Propagation delays in the physical links add to delay, but are usually not the dominant contributor to overall delay.

Jitter is, of course, the variation in delay. Jitter is an important factor because it determines the minimum receive buffer length. The buffer has to be long enough that is it able to catch the latest arriving packet. Any packet that turns up past the buffered time is as good as lost. Jitter is mostly caused by queuing delays. If an outgoing link is busy, packets have to wait around in a buffer until their turn comes.

In AoIP and VoIP equipment, buffers can either be fixed, user-configured, or automatic. On LANs, the buffers can simply be fixed to a low value thanks to the excellent QoS.

VoIP and IP codecs, which have to work on networks with poor QoS, either allow manual configuration of buffer length or have automatic algorithms to set the length dynamically. The latter requires time-stretch/squeeze capability, so it is only practical when the audio is coded. See Chapter 7 for more information.

2.5.4 Service Level Agreements

With dedicated links and MPLS (Multiprotocol Label Switching) service, there usually will be a contract with the provider specifying the terms of its obligations to the customer with regard to quality of service. These are called service level agreements (SLAs). Typically an SLA will include the following points:

- QoS guarantee: delay, specified in milliseconds.
- QoS guarantee: jitter, specified in milliseconds.
- QoS guarantee: packet loss limits, specified as a percentage.
- Non-QoS guarantees, such as network availability, specified as percentage uptime. (For broadcast, this should usually be at least "five nines"—that is, 99.999%.)
- The scope of the service: for example, the specific routes involved.
- The traffic profile of the stream sent into the network: this will be the bandwidth required, including any expected burst.
- Monitoring procedures and reporting.
- Support and troubleshooting procedures, including response time.
- Administrative and legal aspects, such as notice needed for cancellation.

2.5.5 MPLS

Multiprotocol Label Switching is an IP service aimed at customers who need guaranteed QoS, such as for VoIP and video conferencing. MPLS works by prefixing packets with an MPLS header, containing one or more "labels," called a *label stack*. These

MPLS-labeled packets are switched after an efficient label lookup/switch instead of a lookup into the IP routing table. Because routers can see the packets as a stream, reserving a specified bandwidth is possible and usual.

MPLS enables class of service (CoS) tagging and prioritization of network traffic, so administrators may specify which applications should move across the network ahead of others. Do you notice the similarity with the Ethernet prioritization discussed earlier? The result is the same. For example, an ISP could provide you with a DSL line that is shared for both telephony and general data. Streams created by phone calls would get tagged with a high QoS value, while data packets would get a lower tag value. Thus, telephone calls would have first call on the bandwidth. General data would be able to use any bandwidth that remains. The bandwidth available for data would expand and contract dynamically depending on how many phone calls are active at a given moment.

MPLS carriers differ on the number of classes of service they offer and in how these CoS tiers are priced. One of the promises of MPLS is that it can cross vendor boundaries, eventually offering QoS to voice applications in a manner similar to the PSTN.

MPLS is increasingly being used as the basis for virtual private networks.

2.6 IP AND ETHERNET ADDRESSES

IP packets traveling over Ethernet require both IP addresses and Ethernet MAC (physical) addresses.

2.6.1 IP Addresses

IPv4 addresses are four bytes long and are written in "dotted decimal" form, with each byte represented decimally and separated by a period. For example, in the IP address 193.32.216.9, the 193 is the value for the first byte, 32 for the second, etc. Since a byte can hold values from 0 to 255, this is the range for each decimal value.

IP addresses are assigned to your organization by your ISP and parceled out to individual computers by your network administrator. He or she may give you this number to be entered manually, or could opt for DHCP to let your computer get the address automatically from a pool.

The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of IP address space for private Intranets (LANs):

- 10.0.0.0–10.255.255.255
- 172.16.0.0–172.31.255.255
- 192.168.0.0–192.168.255.255

This is why you almost always see gear like home NAT/routers set up with 192.168.0.1 as their default address. Devices on LANs that do not need to be visible

on the Internet almost always use addresses from one of these ranges, most usually the latter. Thus, these same addresses can be used over and over on each LAN without further effect on the available IP address space.

2.6.2 Subnets and the Subnet Mask

Subnets allow a network to be split into different parts internally but still act like a single network to the outside world. The purpose of this is threefold:

- To provide isolated broadcast domains similar to the layer 2 VLANs we described previously, and for just the same reason: Too large a broadcast domain can result in too much traffic in the network and too much CPU time in devices being consumed for filtering packets.
- To allow devices on a LAN to know whether the device they want to contact is local within the LAN or is remote and needs to be contacted via IP routing.
- To reduce the number of entries needed in the routing tables in the IP routers distributed throughout the Internet. With subnets, routers only need to have one entry for the base address, rather than one for each individual IP address. This works in a hierarchically branched fashion, with all of an ISP's subnets being hidden to higher-level routers. Likewise, your ISP doesn't need to know anything about your organization's subnet structure. Without subnets, routing tables would have grown impractically long and the Internet would have been brought to its knees years ago.

As you probably understood from the hint above, there may be multiple levels of subnets, with large subnets being divided into smaller ones. For example, your Internet provider could give your organization a subnet (from the ISP's perspective) with 1024 addresses, which could be split up by your network administrator into four 256-address subnets. This final, smallest subnet level can be referred to as a *broadcast domain*. Traffic within a subnet would be Ethernet switched, while traffic that needs to pass between subnets or out to the Internet would be IP routed.

There are two logical parts to any Internet address: the so-called network prefix, and the individual device address. The subnet mask marks the dividing point in the address between the network part and the device (host) part. The subnet mask is a 32-bit (4-byte) number, just as an IP address is. This has to be entered into an IP device, either manually or automatically via DHCP.

To understand the mechanics of the subnet mask, you need to be thinking in binary numbers because the dotted-decimal representation is obscuring what is going on.

In binary numbers, the only digits available are 0 and 1. The rightmost digit of a binary number represents the amount of ones in the number (either 0 or 1). The next number represents the amount of twos, either 0 or 1; the next number, the amount of fours; etc. Thus, to convert the 8-bit binary number 01111010 to decimal,

we would use the following map. The top row is the "digit weight" and the bottom is the binary number that is being converted.

128	64	32	16	8	4	2	1
0	1	1	1	1	0	1	0

The result of adding the decimal values in this case is $64 + 32 + 16 + 8 + 2 = 122$. So the eight-digit binary number 01111010 is 122 in decimal notation. If you have eight zeroes, the decimal value is obviously zero. If you have eight ones, the decimal value is 255. We're using the map to help you get an understanding of binary representation. For actual calculation, scientific calculators, including the one that comes with Windows, can help you to easily convert binary to decimal. (Experienced network engineers can do it in their head—and probably in their sleep.)

To understand how a subnet mask splits up the IP address into network (subnet) address and device address, you have to convert both the IP address and the subnet mask to binary numbers. Once the IP address and subnet mask have been converted to binary, a logical AND function is performed between the address and subnet mask (which means the resultant value is 1 if both IP and subnet mask value are a 1; otherwise the result is zero). Let's look at an example:

IP Address: 200.122.5.53
Subnet Mask: 255.255.255.0
200.122.5.53: 11001000. 01111010. 00000101. 00110101
255.255.255.0: 11111111. 11111111. 11111111. 00000000
Subnet: 11001000. 01111010. 00000101. 00000000

Converting the binary subnet address to decimal, we get 200.122.5.0. This subnet mask is said to have 24 bits in the subnet field, which leaves 8 bits to define devices. With 8 binary bits, there are 256 possible values (0 through 255). However, there are only 254 of these addresses that can be used for hosts on this subnet because the first and last values are reserved. The first is reserved as the base subnet address and the last is the broadcast address for that subnet. (There are exceptions to this rule, but that is a topic best left to those maintaining routers or studying for certification.) For our example, the usable device addresses in the subnet are 200.122.5.1 to 200.122.5.254. The broadcast address is 200.122.5.255.

You might sometimes hear about class A, B, and C networks. This is an obsolete designation system after the introduction of Classless Interdomain Routing (CIDR) in the early 1990s. But old habits die hard and IT folks often call a subnet with 256 devices (as in our example) a "class C network." Classes A and B designate such large address spaces that we don't hear much about them anymore. CIDR also introduced a new notation system in a form like this: 200.122.5.53/24. The 24 after the slash means that the network prefix portion of the address has 24 bits, leaving 8 bits for the subnet. As you can see, this corresponds to our example.

Now we can answer the question: How does a device know whether another device it wants to contact is on the same or a different LAN? Whenever a computer is instructed to communicate with another device, it "ANDs" its address and the destination address with the subnet mask and compares the result. This happens in the IP stack software that is part of the operating system. If the result is the same, the two devices are on the same LAN and the IP stack will do an ARP lookup to determine the Ethernet MAC address of the network adaptor of the destination device. Once it has the MAC address, communication takes place directly via the Ethernet switch. If, however, the result of the "ANDing" is different, the source device will do an ARP lookup for the MAC address of the configured default gateway, which is an IP router that will pass the traffic to another subnet.

So now you know where all the numbers you have to enter into your computer's network configuration come from, right?

- The *IP number* is either globally unique within the Internet's 4-byte address space, or a private network address served by a NAT.
- The *subnet mask* defines the base address (together with the IP number) and size of your subnet.
- The *default gateway* is the address for the IP router that handles traffic that flows outside of your subnet, which usually means outside of your LAN.
- The *DNS server* address tells your DNS resolver (part of your computer's IP stack software) where to find the DNS server it can use for lookups.

2.6.3 Ethernet Addresses

While IP addresses are either user-configured or set automatically by DHCP, Ethernet MAC addresses are programmed permanently into the network interface by the manufacturer and cannot be changed during the life of the equipment.

You will probably never have to deal with them directly, but who knows? Ethernet addresses are 6 bytes long and are written in dashed-hexadecimal form like this: 5C-66-AB-90-75-B1. (Sometimes colons are used as the separators instead of hyphens.)

Hex notation is just another way to write binary values. Single digits range from 0 to 9; A, B, C, D, E, and F; and byte values from 00 to FF. The value FF means all the bits in a byte are 1s, which is equivalent to decimal 255. While this notation may seem strange at first sight, it comports with how programmers work, since they need to think in powers of two. (Interestingly, IPv6 changes IP address notation to also be hexadecimal, written in a form like this: 2001:db8:1f70::999:de8:7648:6e8.)

> DEADBEEF is a valid hex number. Astound your friends—especially the vegetarians.

There is a unique Ethernet MAC address for each and every network adapter ever made in the world. IEEE handles the allocation among manufacturers, and each manufacturer is responsible to ensure that it makes no two alike within its assigned range.

DIGITAL FRUGALITY We used to feel bad about all those wasted addresses from obsolete and thrown-away network cards. (Steve's Protestant Midwestern United States upbringing is his excuse for this, while Skip is just a closet pack-rat and hates to throw away anything.) Ethernet's 48-bit address space doesn't rise to the literally astronomical magnitude of IPv6's, but the MAC addressing range is still enormous. Supposedly, Ethernet could address each of the Earth's grains of sand. So we can probably all relax.

2.7 NETWORK DIAGRAMS

Reading and making network diagrams is an inescapable aspect of performing network engineering. Figure 2.7 is an example of a diagram. Here we have a simple and quite normal office network for a small business. Perhaps inspired by this book, they have moved to an IP-based telephone system. To support that, they have a VoIP router/server that also includes a T1 gateway to the PSTN. The main router includes an integrated firewall. There are PCs and IP phones, a web server, and a file server—the usual complement for a small office setup. Everything connects via an Ethernet switch. (We assume their email is hosted offsite.)

READING THE RUNES A note about the strange state of networking iconography: The symbol for an IP router is fairly consistent. It's always round with 90-degree crossed lines or arrows on top. Cisco's house standard is thick arrows, and there are variations for routers with integrated firewalls, routers that include VoIP services, etc., but the circle + 90-degree crossed-line theme will always be there. In contrast, Ethernet switch symbols are surprisingly variable given their widespread deployment. This is probably because Cisco's weight in the IP router business puts some discipline into router symbols, while the Ethernet switch market is more scattered. Anyway, the Ethernet switch symbol is *usually* square, with small parallel arrows on the top. If the switch has layer 3 features, there will probably be some crossed lines on the sides. You can pretty much count on this mnemonic, at least: router = round; switch = square.

There are also different tastes in drawing Ethernet-switched network segments. One way is to put a switch icon at the center and radiate lines out from it to all the connected devices. The other is to have a line that acts like a bus, with connected devices tapping onto it. The latter is actually the old default for showing coax-bused Ethernet, but these days a reader would just assume a switched network in all cases. In Figure 2.7 we have taken the best of both ideas. A switch symbol is located near the Ethernet bus to make clear that a switch is involved. This also gives us a place to write the model number, IP address, etc., should we want to.

Diagrams are made for a variety of reasons—to explore ideas, to make presentations, to plan new networks, and to document existing ones. This means that there

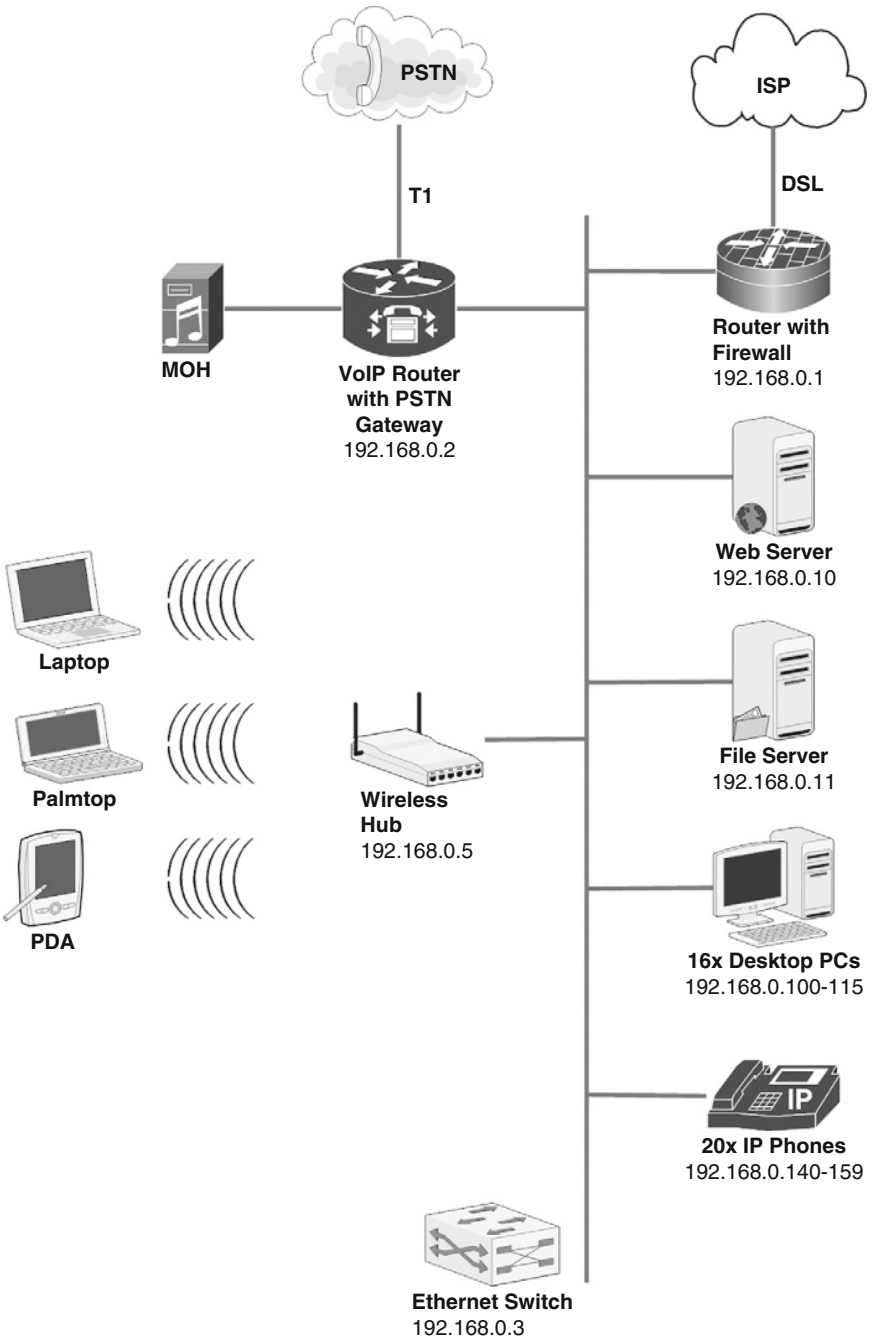


FIGURE 2.7

A typical network diagram. This is depicting a small business office that has a VoIP telephone system with a connection to the PSTN independent of the IP link to an ISP. (MOH = music on hold.)

is no one right way to make them. Indeed, one of the decisions to be made before beginning a drawing project is what level of granularity you need. For example, is a multiswitch Ethernet just a cloud, or do you need to show the individual elements?

Microsoft's Visio is the most popular—indeed the default—drawing application for making network diagrams. It has extensive libraries of symbols and other drawing elements helpful for this work. Networking equipment vendors often make drawings of their products available in Visio library format. Smartdraw is another possibility at about half the cost. It is similar to Visio and also has an extensive library. It can import and export many formats so you have a way to share drawings with others who don't have Smartdraw. Both of these have special features for making diagrams that general-purpose drawing applications such as Adobe Illustrator don't, mainly that lines are automatically attached to objects so that when you move things around the lines stay attached. It's also much easier to sketch something easily and play with ideas by moving objects and even whole sections around.

> Cisco hosts a web page with an extensive library of icons in Visio, EPS, TIFF, BMP, and PowerPoint formats: *http://www.cisco.com/web/about/ac50/ac47/2.html*.

2.8 PRO AUDIO, MEET IP

AoIP is unlike any other application that uses data networking. The name is intended to echo VoIP, but IP telephony has a quite different set of requirements. AoIP needs a delay less than a few milliseconds, while VoIP can tolerate over 100 milliseconds. AoIP is usually operating within a tightly controlled LAN, while VoIP often has to traverse WANs.

AoIP is also pretty dissimilar to streaming audio/video. The latter is a one-way service that can withstand multiple seconds of delay, and its streams are generally transported over the uncontrolled public Internet.

Neither VoIP nor streaming media use multicast, while AoIP uses it almost exclusively.

As we've said, AoIP takes advantage of the open and flexible qualities of both Ethernet and IP, but it uses both in a unique way. This will become more evident in Chapter 4, where we look more closely at the Livewire AoIP system.

CHAPTER

Switching and Routing

3

In Chapter 2, you saw how AoIP systems leverage the intrinsic power of standard Ethernet switches and IP routers. In fact, most AoIP-specific products are essentially peripheral devices, with a lot of the real work of an AoIP system taking place in the generic switches and routers.

So while they are very much "behind the curtain," switches and routers are central to AoIP operations, and thus worthy of additional examination.

3.1 LAYERS AND TERMS

To review, we consider the work done by the devices operating at the data link layer (layer 2) as *switching*, and the most atomic operative units they work with (called *protocol data units* or PDUs) are referred to as *frames*. The communication protocol used at this layer by AoIP is Ethernet (IEEE 802.3), and connected devices at this layer are addressed by MAC (Media Access Control) addresses.[1]

Meanwhile, work done in this area at the network layer (layer 3) is called *routing*, and its PDUs are called *packets*. The communication protocol used at this layer by AoIP is, of course, Internet Protocol (IP), and connected devices at this layer are therefore identified by IP addresses. Table 3.1 summarizes these naming conventions.

So although the old-school audio world used the terms *routing* and *switching* interchangeably, we quite specifically refer to the central devices operating at these different layers in AoIP as *Ethernet switches* and *IP routers*, without exception. Repeat after us: Switching is done at layer 2, routing is performed at layer 3.

[1] "Media" in the networking context refers to the physical interconnection medium (i.e., "wire," or in the case of wireless networking, the RF channel).

Audio Over IP

doi: 10.1016/B978-0-240-81244-1.00003-1

Table 3.1 Naming Conventions Used in This Book for Two "Connective" AoIP Layers

OSI Layer	Layer Name	Protocol	Central Device	PDU	Address Format
Layer 2	Data Link	Ethernet	Switch[a]	Frame	MAC
Layer 3	Network	IP	Router[b]	Packet	IP

[a] *An Ethernet switch is sometimes called a* multiport bridge*, or a "smart bridge." This arises from the fact that in those contexts, the term* bridge *is used for what is more typically called a* hub.

[b] *The term* IP switch *has turned up a lot recently, but don't be confused. This is simply marketing talk generated in the burgeoning world of VoIP, where the "soft" IP environment replaces the old telco hardware TDM "switch" (as in 5ESS). So to maintain apparent congruity with telco lingo, the term* IP switch *is applied to VoIP routing.*

SOMETIMES YOU FEEL LIKE A NUT . . . There is good reason for making the distinction between switching and routing, as this chapter (and others) point out. One fundamental difference, if you haven't recognized it already, is that these two areas define the conceptual boundary in an AoIP system between hardware and software. Or, in more careful terminology, between *physical* and *logical* address space. To wit, the MAC address is essentially "burned into" a device, assigned to the connecting hardware for life, as it were. Everything below this point in the stack is essentially "hardware," or *physical*. On the other hand, IP addresses are assignable, and can be quite flexibly configured and reconfigured as needed. Correspondingly, everything above this point in the stack is essentially "software," or *logical*.

This should give you some indication of the respective values of switching and routing to AoIP. Sometimes the fixed, device-associated assignments of Ethernet are quite desirable, while in other cases, the flexibility of IP addressing is very useful.

As we've also pointed out—but it bears repeating here—networking design allows these two very different capabilities to easily interoperate on the network, flexibly and independently. In fact, AoIP systems typically use both Ethernet switching and IP routing devices together to great advantage—a very sensible arrangement, as it turns out (which you'll read more about later).

When you think about it, even the words chosen here seem appropriate, with "switch" implying something hard-wired or with very limited options (like a railroad switch), whereas "route" connotes a more flexible choice among a variety of paths (like highlighting a roadmap).

Consider also that much of the literature on IP today is devoted to the Internet, where all interconnections and navigation are performed at the network (or "Internet") layer. Thus, the distinction is not so important in that space, although the term *routing* is still generally preferred. In AoIP, we generally do not involve the Internet but remain confined to LANs, so interconnections are important at both layers 2 and 3.

3.2 ETHERNET SWITCH

Almost all AoIP systems (other than a simple point-to-point connection like an IP snake) involve at least one Ethernet switch. Even relatively large AoIP systems may not need any IP router hardware, however. Thus, the switch is of primary interest.

There are many other books that explain the basic and advanced features of Ethernet switching, of course, and most of that information is not necessary to know to accomplish the successful installation and operation of an AoIP system. So what you'll find in this chapter are elements of Ethernet switching (and IP routing) that are specific to AoIP, and helpful to understand for professional audio studio application.

First, remember that Ethernet switches—like a lot of fast computing equipment—can be acoustically noisy due to their requirement for adequate ventilation. This is important to remember when planning a network. Even the edge switches that are often distributed around a facility outside of the TOC should be placed in a noncritical (and preferably well-isolated, but also well-ventilated) acoustic environment. In modern facilities this is, of course, also true of the CPUs for any PCs used in control rooms and studios, which are generally placed in an adjacent or nearby wiring closet or rack room (if not in the TOC), and typically connected to user control surfaces via KVM (Keyboard/Video/Mouse) extenders. So, just make room in those same racks for your AoIP edge switches as well.[2]

Also, like any digital equipment, a stable electrical environment with a properly conditioned AC power source helps to ensure reliable operation. This is particularly important for a facility's core switch, since so many other devices connect to and through it.

3.2.1 Managed Switches

Consider that the most basic function of a switch is to connect networked devices together, but with some intelligence over a "dumb" hub, which simply retransmits all incoming bits to all its connected devices. A switch, on the other hand, inspects the contents of its frames, and determines which to send where based on their addressing.

For AoIP, *managed switches* are usually required. These provide the essential IGMP control of multicast audio streams. They also offer the capability for user configuration and monitoring of the switch's various parameters, and generally include priority management, which is essential when both audio and general data need to share a common link.

Unmanaged switches are common in home and the SOHO environment, and they can be used for some very simple AoIP applications, such as snakes (see Section 5.3.1 in Chapter 5).

[2]As discussed in Chapter 4, some Livewire console products integrate an edge switch into a rack box that includes the console CPU, power supply, and audio I/O. Intended for installation within studios, these are fanless.

3.2.2 Scalability

Because true AoIP systems base their switching infrastructure on standard, off-the-shelf switches, they are relatively easy to reconfigure or grow by changing or adding switches. Such accommodation of growth is an important feature, as any facility engineer knows well (perhaps too well).

The modular and standardized design offers a simple, quick implementation of these changes as well. Over time, a successful facility may find that this "future-proofing" is their AoIP system's greatest asset.

Moore's Law certainly applies to network switches, so when a larger or faster replacement or expansion switch is required, you may find that it costs no more than the previously purchased smaller unit, as is often the case in the off-the-shelf computing world.

Ethernet is also not standing still. Ten-gigabit Ethernet (10-GbE) switches are already available as of this writing, and 100-Gbps performance is probably not far behind.

3.3 IP ROUTER

IP routing is rarely used for AoIP applications, but for very large installations, it can add a lot of power. And routers will almost always be involved for nonaudio applications such as connecting PCs to the Internet or linking VLANs (more on these back in Section 2.3.4). So an understanding of the inner workings of IP routers may prove worthwhile to more advanced AoIP users. For this we recommend sources in the References and Resources chapter.

3.3.1 Roots of the Internet

You may recall that developers of the Internet simplified the canonical seven-layer OSI networking architecture[3] down to a stack consisting of only four layers, as shown in Table 3.2. All of these are well-defined and open standards—there is no proprietary

Table 3.2 Four Layers of the Internet Network Model, with Examples of Each Layer's Protocols

Internet Model Network Layer	Example Protocols
Application	HTTP, RTP, FTP, SMTP, Telnet
Transport	TCP, UDP
Network (or Internet)	IP
Link	Ethernet, WiFi

[3]The Open Systems Interconnection (OSI) reference model, established in the late 1970s by ISO, included application, presentation, session, transport, network, data link, and physical layers, as discussed in Chapter 2.

ownership of any of these core technologies. The link layer is standardized by the IEEE, and the protocol suite used for the upper layers is standardized by the IETF.

The Internet process also includes an addressing protocol for each of its data packets, the *IP address*. Any device attached to an IP network is assigned an IP address. Until recently—that is, using IPv4—an IP address was specified as a numeric string of four 1-byte numbers (or *octets*, since 1 byte is 8 bits), each expressed in decimal form (from 0 to 255) and separated by periods[4] (e.g., 169.10.206.2). This implies that the number of possible addresses in IPv4 is that expressed by a 32-bit number (4×8 bits), meaning that 2^{32}, or approximately 4.3 billion (4.3×10^9), unique IP addresses are available. This may sound like a lot, but many of these are reserved for specific uses.

IP REVS UP Today, the IP world is gradually converting from IPv4 to IPv6, which specifies its IP addresses using 128-bit rather than 32-bit numbers. The numerical expression of IPv6 addresses also differs from IPv4's, in that it generally uses hexadecimal numbers in the form hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh, where each byte (or octet) is represented by a hexadecimal pair of numbers (from 00 to ff, e.g., e7), and each pair of bytes is separated from the next pair by a colon. An example is 30c1:0ab6:0000:0000:0000:8a2e:0370:2f8e. For a while, there may be a lot of zeros in IPv6 addresses, and they can be skipped with the insertion of a double colon, as in this notation of the previous example: 30c1:0ab6::8a2e:0370:2f8e.

One of the primary improvements of IPv6 over IPv4 is its allowance of far more IP addresses. This is a real issue given the expectation that so many devices in the future will require unique IP addresses. IPv6's 128-bit range provides more than 3.4×10^{38} possible addresses, or more than 5000 addresses per square micrometer of the Earth's surface—probably enough to last for awhile.

Nevertheless, it is expected that IPv4 will remain the standard format of the Internet for some time to come, while IPv6 support is gradually deployed worldwide. And for the foreseeable future, AoIP will likely continue to use IPv4 happily, for reasons that will become clear as you read on.

By the way, if you're wondering what happened to IPv5, it was ascribed to a version that was originally intended to be used for connection-based (rather than packet-based) streaming media on the Internet, but work was abandoned on it as streaming media became possible with the development of new protocols over IPv4, which there's much more about later in this book. Anyway, "Whither IPv5?" is always a great trivia question at geeked-out cocktail parties.

IP is used today on many LANs that are not connected to the Internet, but simply run within a facility over Ethernet links. IP's designers must have anticipated this

[4]IPv4 has been in use since 1981, established with the publication by DARPA of the seminal RFC 791 document, generally cited as the original specification for the Internet. Although other protocols preceded it, for most of us, IPv4 is the only version of IP that the Internet has ever used.

because there are a large number of IP addresses that are reserved for non-Internet uses on private networks.[5] A number of IP address ranges are internationally agreed to be reserved for this purpose, the largest contiguous group of which spans from 10.0.0.0 to 10.255.255.255. This group alone provides some 16 million possible addresses that are not accessible from the Internet (routers are programmed to ignore the addresses on incoming Internet traffic), and are only available from within a local network. This also implies that a private IP address has no need to be globally unique, and so these same addresses can be used by any entity on its internal network, thereby conserving the number of IP addresses required worldwide.

Devices assigned such private addresses can still access the Internet if necessary, via an IP router, proxy server, or network address translation (NAT) device.[6] The private address space is useful for studio audio applications, since the devices are not intended to be accessible directly via the public Internet.

When AoIP needs IP routing for audio transmission, it usually makes use of IP multicast on yet another reserved range of addresses. This allows an IP-based infrastructure to work like an audio distribution amplifier or traditional audio router, where one output may be received at any number of inputs.

3.3.2 TCP/IP Suite

The great thing about network models is that (just like standards) there are so many of them. So let's clear up this layering issue once and for all. In Chapter 2 you saw the ISO OSI seven-layer model (how many other palindromic acronyms do you know, by the way?) and how it evolved into the five-layer, real-world variation used in TCP/IP networking today. Meanwhile, we just saw how the Internet actually uses a *four*-layer model, and there are other models with varying numbers of layers as well.

So how many layers are there, really? Well, it depends on your world. Since in this book we are neither engaged in the academia of software design, nor dealing with the Internet per se, we will stick with the five-layer TCP/IP nomenclature, which is just right for the AoIP environment. But in case you'd like to see how these three different maps of the network-layer worldviews intersect, check out Figure 3.1.

3.4 STRADDLING LAYERS

So if both switches and routers examine packet addresses and send them appropriately on their way, why and where would you use one versus the other in AoIP?

[5]These networks use addresses in the private IP address space, as specified in the IETF's RFC 1918, and administered by the Internet Assigned Numbers Authority (IANA).

[6]Note that in IPv6 there will be no private address space or NAT, given the far greater number of globally unique IP addresses it provides.

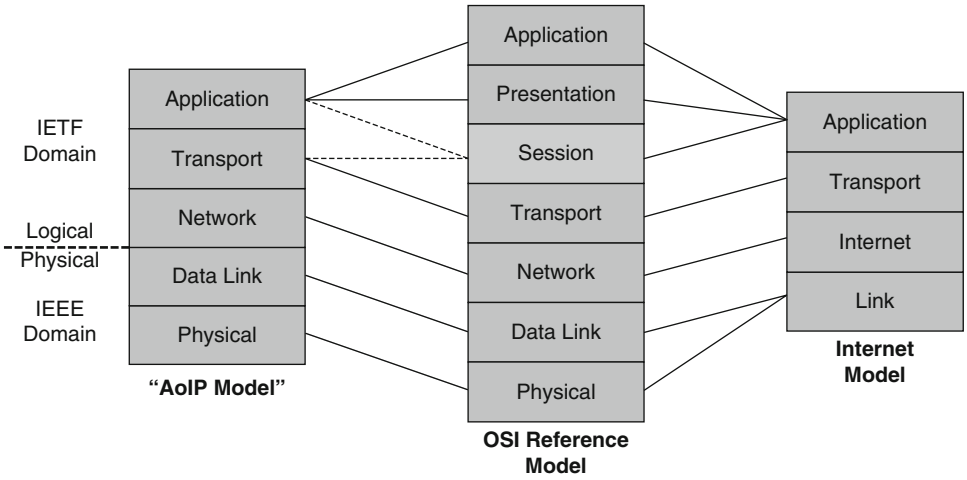


FIGURE 3.1

How the OSI networking model maps to the so-called Internet model and the TCP/IP Protocol Suite model (or "AoIP model" as we name it here, and generally use throughout this book).[7] The dotted line from the OSI session layer indicates that some of its elements map to the TCP/IP Suite's transport layer, while the rest map to the application layer. Note also the boundary shown on the left and the organizations that have jurisdiction in each domain.[8]

First, note that routing is a much more complex operation than switching, because multiple paths from one site to another are the norm at layer 3, and it is the job of the router to find the optimum path, which may well be changing from minute-to-minute. Thus, to keep things simple and cost effective, layer 2 switching is preferred whenever it's adequate, which is most of the time for AoIP systems.

Nevertheless, routers also support multicast and prioritization, just like switches. So it would be possible to have a routed AoIP network on top of a switched one. You'd still need the layer 2 switching because Ethernet would remain as the transport layer, but if the AoIP system fills the IP header with all required information, and does it in a standard way, IP routers could be used interchangeably throughout the AoIP network. Table 3.3 summarizes the two approaches.

[7]Other network models found in the literature range from three to seven layers. The three models shown here are the most commonly referenced today.

[8]Given its focus on the Internet, however, the IETF staunchly supports the four-layer model, which it defines in its RFC 1122. It does occasionally acknowledge other models' layers, however, sometimes even in a cautionary way (as in RFC 3439's Section 3, "Layering Considered Harmful").

Table 3.3 Ethernet Switch and IP Router Compared

	Switch	Router
Layer/Protocol	Layer 2/Ethernet	Layer 3/IP
Function	Determines to which port the addressed node is connected, and switches incoming frame to it	Finds the best route from among many, and forwards packet to next destination along the path
Terminology	Switching	Routing (or Forwarding)
Technology	Simple table lookup in hardware	Complex, dynamic best-route determination in hybrid hardware/software
Specifying Organization	IEEE	IETF
Ports (Physical)	Many, connecting mostly to end nodes	A few, connecting to networks and telco lines
Cost	Low	High (but dropping)

Traditionally, routers did their work with software, while switches had dedicated hardware chips. But now routers are using hardware for packet handling and switches are incorporating some layer 3 functionality, bringing each closer to the other. An increasing number of lower-cost switches have layer 3 features, and these are often a good choice for AoIP applications.

3.5 AUDIO ROUTING CONTROL

While an AoIP system ideally uses standardized technology at layer 3 and below, a proper AoIP system should offer an application that acts as a user interface for routing control, which sits above one or more actual routers and switches to appear as a "virtual router" for the entire facility.

This allows operators to approach the AoIP system just like a traditional audio router, hiding all the details of the underlying network. Under the hood, there might be a number of separate physical devices actually doing the routing work and being controlled by the AoIP application. These "real" devices also can be distributed around the facility, but appear to act as a single, virtual core. This is another value-add of today's commercial AoIP systems. (See Chapters 4 and 5 for a description of the Pathfinder application, which provides this virtual routing function in the Livewire system.)

The control path through the system for an AoIP audio router would thus be:

1. A user requests a route change via a manual command to the AOIP system's router application user interface, which is connected via the network to the routing control server.
2. The routing control server sends commands to AoIP devices, such as audio interface nodes.
3. The AoIP devices send IGMP commands to Ethernet switches (and IP routers, if any).
4. The Ethernet switches (and possibly IP routers) respond by directing the chosen source audio stream through the network to the target AoIP receivers.

Note that such a routing control application is only required when there is a need to emulate sophisticated audio routers. Basic audio channel selection is accomplished by choosing the desired feed directly on AoIP devices such as consoles and interface nodes.

3.6 MULTICASTING

An unusual feature of AoIP is its nearly exclusive use of multicast for audio streams. While both Ethernet and IP offer this capability, it is infrequently used, either on the Internet or in LANs. So this is yet another element that makes AoIP a bit of a special case in its particular application of standardized technologies.

Multicasting allows a directed, point-to-multipoint distribution topology. This means that bits from a single source can be distributed to many destinations, but only to those that need them, thus minimizing unnecessary traffic on the network.

AoIP systems could implement multicasting either at layer 2, layer 3, or both. The multicast implementation used by AoIP is generally carried out at layer 2, using Ethernet. But when a router is in the picture, it uses multicasting as well. Fortunately, the inventors of the IP and Ethernet switch protocols have thought about this and provided the necessary integration.

Remember that the packet routing on a typical AoIP system is quite limited in its scope compared to the Internet, or even to many enterprise LANs. Thus, handling the routing of signals at as low a layer as possible minimizes system complexity (and therefore cost).

Ethernet switching offers plenty of capacity for multicasting, although many enterprise LANs never use it. Yet it's perfect for AoIP, allowing the Ethernet switch to emulate a distribution amplifier (DA) or TDM router's typical capability of multipoint distribution, but without ever running out of outputs.

THE REAL WORLD: SWITCHING OPTIONS COMPARED

As you're probably well aware, AoIP systems provide particular cost effectiveness to professional audio studio facilities. The bulk of the cost differential AoIP achieves comes from switching components and the associated facility wiring required.

To illustrate how this works, what follows is an analysis of traditional audio studio interconnection methods compared with AoIP, both in terms of equipment costs and installation expense.

P2P

Most broadcast engineers remember that it wasn't long ago when all studio interconnection infrastructure was point-to-point. The wiring costs were fairly fixed, and the only real cost differentials came from what equipment you chose to put in the various rooms. Another big-ticket item was the crosspoint switcher, which some facilities avoided by preferring to live with patch bays only.

Over the years, simpler methods of wiring came about, but the design continued to be "console-centric," with all audio sources and destinations wired directly into and out of the audio console. In many cases these connections were brought out to a field of punch blocks or other cable terminations to allow future flexibility.

This concept was extended for connections between studios, which were often made through a central wiring area, like the TOC. The connection between rooms was generally done with multipair audio cables, again often terminated on both ends by punch blocks. The TOC either used manual patch bays and cords, and/or a central routing switcher.

All of this meant that a typical audio feed's path might pass through as many as a dozen "punch-downs" along its journey from source device to transmission output. In addition, the process was extremely labor intensive, and the more elegant and flexible the design, the more work was required (not to mention preparing the requisite documentation). Often, the facility still outgrew its infrastructure all too quickly.

TDM

Eventually, digital (TDM) routers pioneered by telcos started to find their way into professional audio and broadcast facilities. Beyond the easier distributed routing control these systems allowed, they also cut down on the wiring between (or sometimes within) studios, since they allowed multiplexing of multiple signals onto single pathways. Thus, this started the move away from the massive "parallelism" of point-to-point wiring and the transition to serialization.

By the mid-1990s, this approach took the form of a TDM "router plus control surface," where the mixing console could also act as a control point to the central router and mixing "engine" (just as earlier systems had allowed for simple switching-control panels), and it could be connected by serial digital network cabling rather than parallel audio multipairs.

The TDM approach still required traditional wiring breakouts at either end, terminating in multipair and multiple single wiring terminations to most device audio inputs and outputs. The switching hardware was also proprietary and generally fairly expensive, but was

often considered worthwhile both for the flexibility it provided and the installation cost savings (particularly in larger facilities).

AoIP

The next stage in this development is AoIP, which takes the serialization approach further, based on the computer networking that had by that time become common among these and other enterprises. AoIP extends the serialized domain all the way to the individual audio source and destination devices, by allowing all audio equipment I/Os (analog or AES3, plus their control ports) to be connected to terminal devices (or "nodes") of the serial digital network.

Some sources or destinations (e.g., computer-based audio recording and playback systems, processors, STLs, etc,) can even interconnect in the native AoIP format, simplifying their interfacing.

As in the later TDM systems, AoIP mixing consoles are also control surfaces, acting as sophisticated switching controllers, with audio mixing taking place in outboard "mix engines."

Tables 3.4 and 3.5 show some examples of real-world cost comparisons of these three topologies. Note how the greatest differentials exist in the line items associated with wiring and switching, for both equipment cost and installation fees.

Table 3.4 Infrastructure Equipment and Wire Costs for Typical Four-Studio Plus TOC Facility

Materials	P2P	TDM	AoIP
CAT-6 cable or fiber	$0	$200	$600
Multipair audio cable	$2800	$1600	$0
Punch blocks and wiring guides	$1600	$800	$0
Distribution amps	$2400	$0	$0
Central audio router (P2P or TDM) or Ethernet switches (AoIP)	$60,000	$120,000	$18,000
Audio terminal devices	$0	$60,000	$32,300
Audio mixing console/control surface	$76,000	$104,000	$68,000
Audio cables and connectors for studio and TOC equipment	$900	$900	$1200
Total equipment cost for consoles, routing, and wire	$143,700	$287,500	$120,100

Note: *In U.S. dollars.*

Table 3.5 Infrastructure Installation Costs for Typical Four-Studio Plus TOC Facility

Task	P2P Wiring (hours)	TDM Wiring (hours)	AoIP Wiring (hours)
Studio: source/destination equipment to punch blocks or nodes	96	96	32
Studio: console to punch blocks	32	0	0
Studio and Tech Center: multipair cable runs and terminations	48	0	0
Studio and Tech Center: CAT-6 cable terminations	0	16	16
Tech Center: central audio router to punch blocks	32	32	0
Tech Center: source/destination equipment to punch blocks or nodes	24	24	4
Tech Center: distribution amps to/from punch blocks	8	0	0
Programming of audio router or nodes and consoles	4	16	16
Total labor hours	244	184	68
Installation labor expense (at $50/hr)	$12,200	$9200	$3400

Note*: In U.S. dollars.*

CHAPTER

Livewire System

4

Livewire is not only a technology. It is a *solution*, with a wide variety of components and tools made for broadcast and other professional audio applications. Imagine everything you can do with a PC connected to a network: share files, send and receive emails, surf the Web, chat, make VoIP calls, listen to streaming audio, watch YouTube, etc. PCs and networks are designed to be general-purpose enablers. You have a similarly wide range of possibilities for audio applications using Livewire. Yes, it is able to replicate everything that older analog and digital technologies were able to do, but more valuably, it provides a *platform* that lets you go well beyond the limited capabilities of the past.

4.1 WHAT CAN YOU DO WITH IT?

You can build broadcast studios, of course. These can range from basic one-room, one-console operations to sophisticated plants that have dozens of interlinked studios with automated routing switching, monitoring, and other advanced capabilities.

Surely you have noticed that a lot of studio audio is now either coming from or going to PCs these days. On-air automation is the first thing that comes to mind in this respect. But what about the audio editors in the production studio and newsrooms? Altogether, isn't most audio in your facility sourced from or sent to a PC application? So why not interface to them using their native, low-cost, and ubiquitous Ethernet ports? You get a pure, noise-free digital connection with multiple bidirectional channels, and with control coming along for the ride—which is not limited to the simple start/stop that general-purpose input/output (GPIO) provides—you have the rich data path provided by a computer network to play with. You might have to play a CD on occasion and CD players don't have native IP connections (yet), but almost all the other audio devices you need in a studio are available with Livewire ports: telephone interfaces, audio processors, codecs, satellite receivers, delay units, and more.

Audio Over IP

doi: 10.1016/B978-0-240-81244-1.00004-3

The building-block nature of Livewire lets you use it for a variety of applications:

- You could build a routing switcher of almost any size. Audio input/output could be via analog, AES3, or Ethernet. The latter might save a lot of money since no PC sound cards would be needed when those are the connected audio devices, and no A/D or AES interface cards would be needed in the router. Because the router core is based on a commodity computer industry switch rather than low-volume audio frames and cards, cost is low. Using the Pathfinder PC software application, you can build a system with pretty much any kind of manual or automated switching that you need.
- You could build a facility-wide audio distribution system for almost any application such as postproduction houses, theme parks, stadiums, and the like. Since Ethernet/IP is readily scalable, the size can range from a few audio sources to hundreds or thousands. The audio system could hitchhike on an existing network infrastructure, saving the expense of installing and maintaining independent networks. All of Ethernet's potential can be exploited, including connections via copper, fiber, or wireless. Automatic redundancy is possible using well-known computer network techniques. Again, you would have the advantage of direct Ethernet connections to PC-based audio players and recorders. With compressed MPEG gateways, the system could be extended to anywhere an IP link is available.
- You can pass Livewire over any Ethernet link that has good QoS. That means that, for example, studio-to-transmitter links can be made with wireless radio systems. Since these have plenty of bandwidth, multiple bidirectional audio paths are feasible.
- You could connect a PC directly with a Livewire node to make a high-quality multichannel sound card. Because 100BASE-TX Ethernet is transformer balanced and capable of 100 meters length, the node need not be located near the computer. Try that with USB!
- You can make a simple snake by directly connecting two Livewire nodes. Again, all of Ethernet's potential is open: copper or fiber in a variety of formats, etc.

Figure 4.1 shows a studio built with Livewire components. It uses an Element console and its companion PowerStation rack unit as the key pieces.

The PowerStation includes an internal Ethernet switch, which supports direct connection of Livewire and other Ethernet devices. Some of them supply power in the standard PoE (power over Ethernet) format, permitting devices such as VoIP phones and accessory modules to work without needing individual supplies. It also has integrated analog and AES3 audio inputs/outputs. A dual-gigabit uplink can optionally be used to connect to an external Ethernet switch serving as a core linking other studios, audio nodes in a central equipment room, etc.

It would be possible to build an entirely self-contained studio without the external Ethernet switch. The telephone interface device would connect directly to an Ethernet port on the PowerStation.

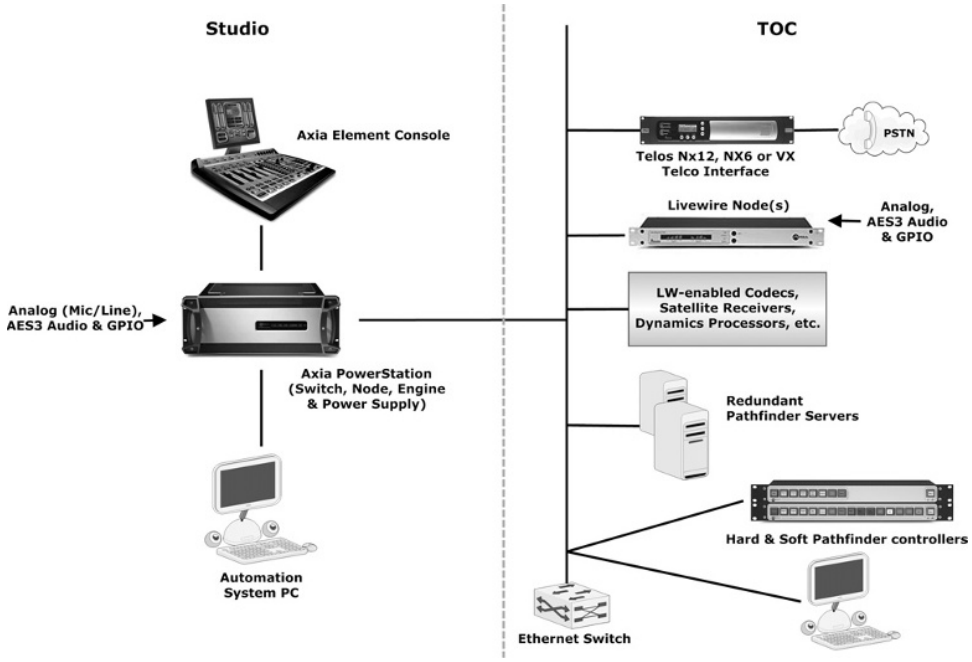


FIGURE 4.1

One among many possibilities for a radio broadcast studio built on Livewire. The automation PC connects directly via Ethernet. The telephone interface, located in a central rack room, connects over Ethernet, as well. This studio is linked to other studios and to common audio sources and destinations via a core switch.

The Ethernet that links the automation system PC to the PowerStation carries multiple audio channels as well as data for control. The same is true for the telephone interface: multiple channels of send and receive audio are transported over the Ethernet along with control for line selection, etc.

4.2 AES3

AES3 and Livewire may comfortably coexist in your facility. You can use Axia interface nodes to connect from one to the other. If you are using a house sync system for AES, Livewire may be synced to that system also. To do this, a Livewire AES interface node would take an AES "black" feed from the AES clock generator or any AES device that is locked to the generator. The node would be configured to recognize that input as the system clock source. An AES-over-IP snake could be built with two or more AES Livewire nodes.

Livewire AES nodes have sample rate converters on all inputs and outputs, but if the rate and clock are matched, the converters are switched off and a bit-transparent transport between the AES and Livewire systems is realized.

4.3 LIVEWIRE SYSTEM COMPONENTS

A Livewire system usually has a mix of PCs with driver software that lets them send and receive Livewire audio streams and hardware audio interface nodes for microphones, loudspeakers, and other devices that are not natively Livewire ready. A small system would have one Ethernet switch, to which everything would connect. A very small system—for example, a two-box snake or a PC sound card replacement—would need no switch. Large systems might have a number of IP routers and Ethernet switches. There could be mixing consoles, intercom systems, phone interfaces, codecs, and other studio audio equipment.

Simple routing is accomplished on the devices themselves, but when sophisticated routing control is needed, a PC-hosted software application will do the work, often in combination with multiple software and hardware control panels.

There is an ever-growing complement of Livewire-enabled software and devices. Here is what is available at the time of writing:

- Axia Windows PC-based audio driver. This looks like a sound card to Windows applications, permitting almost any audio application such as players, editors, etc. to directly connect to the Livewire network. Officially supported partner applications include those from 8BC, AdeuxI, BE (AudioVault), BSI, David, Digispot, Trakt, Enco Systems, Google, MediaTouch, Netia, Paravel Systems, Pristine Systems, RCS Sound Software, Synadyne, WinMedia Software, and Zenon Media.
- Axia Linux audio driver. Provides the same capability to Linux-based PCs via the ALSA sound interface.
- Audioscience PCI interface card. Offloads audio transfer from the PC's processor and provides an isolated Ethernet port for audio. Includes sample-rate conversion and time-stretching/squeezing in DSP hardware.
- Digigram Visiblu driver. Allows software applications written for the Digigram standard to interface with a Livewire network.
- Axia Livewire analog hardware node. Provides 8 $\times$ 8 stereo pro-grade analog inputs and outputs.
- Axia Livewire AES3 hardware node. Provides 8 $\times$ 8 AES3 inputs/outputs with sample-rate conversion.
- Axia microphone node. Eight studio-grade microphone inputs and eight stereo-balanced outputs.
- Axia Livewire router selector node. Provides one input and a selectable output that can access any channel on the network. Designed to look like a traditional x-y router controller. Includes a front-panel headphone jack.

- Axia GPIO node. Provides eight GPIO logic ports for machine control.
- Axia iPlay PC software application. Essentially the router selector node in software. Allows users to play any channel on the network with a PC.
- Axia iProbe software. An application that consolidates control and monitoring of all Axia devices on the network. Useful to check proper operation of the components, upgrade firmware, etc. It also provides stream statistics for troubleshooting and audio monitoring.
- Software Authority Pathfinder. This is a full-up router control software application that has all the features needed for manual or automated control of routing. Going beyond the usual for audio routers, it includes the kind of functionality found in high-end video router applications.
- Axia router control panels. These connect to Pathfinder over IP to provide manual routing control. They come in eight rack-mount variants, with LCD buttons, OLED displays, or film-legendable buttons. Also available for drop-in to Axia's and other studio mixing consoles.
- Axia Element broadcast console. A high-end radio studio modular mixing console.
- Axia IQ mixing console. A more modest mixer (relative to Element).
- Axia intercom system. Used between studios or for any other application where communication is needed. Since the audio is standard Livewire, it may be taken to air just as any other source. Any of the codecs listed below can be used to extend the system over WAN IP links.
- Telos Zephyr ZXS codec. For ISDN and POTS telephone line remotes.
- Telos Z/IP codec. For remotes over IP links. Includes adaptive features that let it work over non-QoS-controlled links such as the public Internet and mobile IP phone services.
- Telos iPort multichannel MPEG codec. Connects 8 $\times$ 8 stereo channels over controlled QoS IP links. Uses state-of-the-art MPEG advanced audio coding (AAC) algorithms to reduce bandwidth. Also possible to use in a 16-channel encode-only mode for creation of streams for public Internet reception.
- Telos Nx12 telephone interface. Up to 12 POTS or ISDN telephone lines are connected with state-of-the-art hybrid and audio-processing functions.
- Telos Nx6. Same as above, but for up to six telephone lines.
- Telos Advanced Hybrid. For only a single POTS telephone line.
- Telos VX multistudio IP-based telephone interface system. Next-generation on-air phone interface system that supports dozens of lines and many studios. Via a gateway, connects to POTS, T1, or ISDN lines, or directly to VoIP telco services with smooth integration to IP private branch exchanges (PBXs) such as those from Cisco, Avaya, Digium (Asterisk), etc.
- Omnia 8x multichannel dynamics processor. Eight channels of high-quality processing in one box. A single Ethernet connects all the inputs and outputs. Use it for processing headphone feeds, Internet and mobile phone streams, satellite uplinks, etc.
- Omnia One on-air audio processor. A low-cost, high-performance digital processor for FM transmission with the respected Omnia sound.

- Omnia 11 FM on-air processor. All the loudness and clarity that millions of floating-point MIPS and clever processor-guru algorithm design can deliver.
- 25-Seven profanity delay and audio time manager.
- International Datacasting satellite receiver. Used by NPR, CBS/Westwood One, and others. An Ethernet port permits direct connection to Livewire networks.
- Radio Systems mixing consoles and StudioHub wiring accessories compatible with Axia analog and AES3 nodes.
- Fraunhofer Institute "content server" encoders for Digital Audio Broadcasting (DAB), Digital Radio Mondiale (DRM), Digital Multimedia Broadcasting (DMB), and Mobile Phone Broadcasting.

Following Gauss' dictum that a good example is worth two books, we'll select some pieces from the list and describe how you can use them. The descriptions here will be brief—just enough to give you a feel for how Livewire devices work. For further exploration, full manuals for the Axia products are available at *www.axiaaudio.com/manuals/default.htm*.

4.3.1 Axia Hardware Interface Nodes

Hardware nodes interface analog and AES3 audio to the Livewire network. They are used to connect microphones, loudspeakers, and other devices that are not available with native Livewire connections. Because hardware nodes provide the essential audio clocking signal, at least one of these must be included in every system.

Analog 8 × 8 Node. Eight balanced inputs and outputs. Software-controlled gain lets you trim-adjust to accommodate different levels. Front panel LED audio level metering.

AES3 8 × 8 Node. Eight AES3 inputs and outputs. An input can be used to sync your Livewire network to your house AES clock, if desired.

Mic + Line Node. Eight microphone inputs with high-grade pre-amps, phantom power, and eight balanced line outputs. The line outputs are able to drive headphones directly.

Configuration and monitoring is via a web interface. Figures 4.2–4.4 are examples from the analog node.

4.3.2 Router Selector Node

The router selector node (Figure 4.5) emulates the user interface and function of traditional x-y-style audio router controllers. It includes an on-board analog and AES3 input and output.

The LCD presents a list of active audio channels, which are selected with the adjacent knob. Programmable "radio buttons" offer immediate access to often-used channels. The router selector node is often used for monitoring and testing in a central equipment room. It can also be installed in production studios and newsrooms as an interface to non-Livewire equipment.

FIGURE 4.2

Axia analog 8 × 8 node.

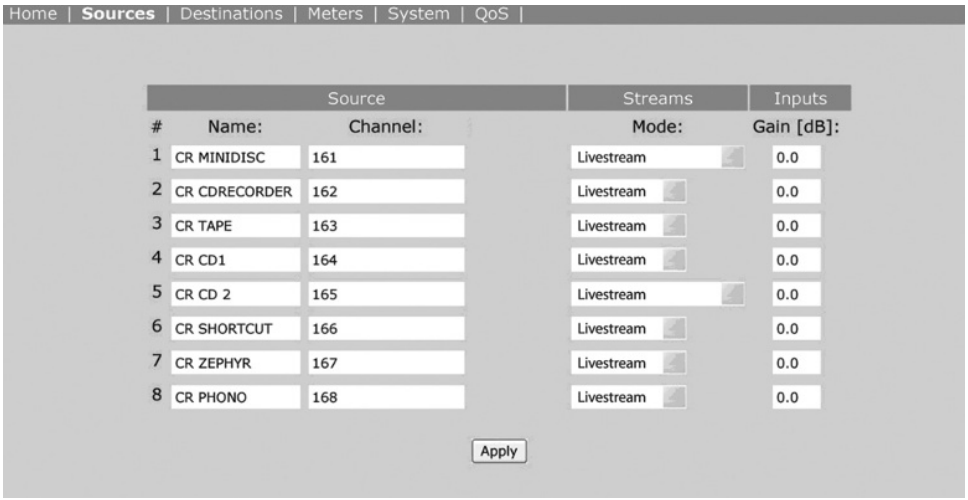


FIGURE 4.3

Source (from the node to the network) text name, channel assignment, mode, and analog gain setting.

4.3.3 GPIO Node

The GPIO interface for parallel "contact closures" has eight DB-15 connectors, each with five inputs and five outputs. It is used for control of CD players, automation systems that don't support network/software interfaces, on-air lights, etc.

For most installations, these are not necessary because the mixing console hardware already has a complement of GPIO interfaces. The GPIO node would be used for expansion, or when there is no console in the system.

4.3.4 Axia Driver for Windows

This is the software interface between PC audio applications and the Livewire network (Figure 4.6). It looks like multiple sound cards to PC applications, supporting 16 inputs and 16 outputs (Figure 4.7). A sample rate converter and a "clock

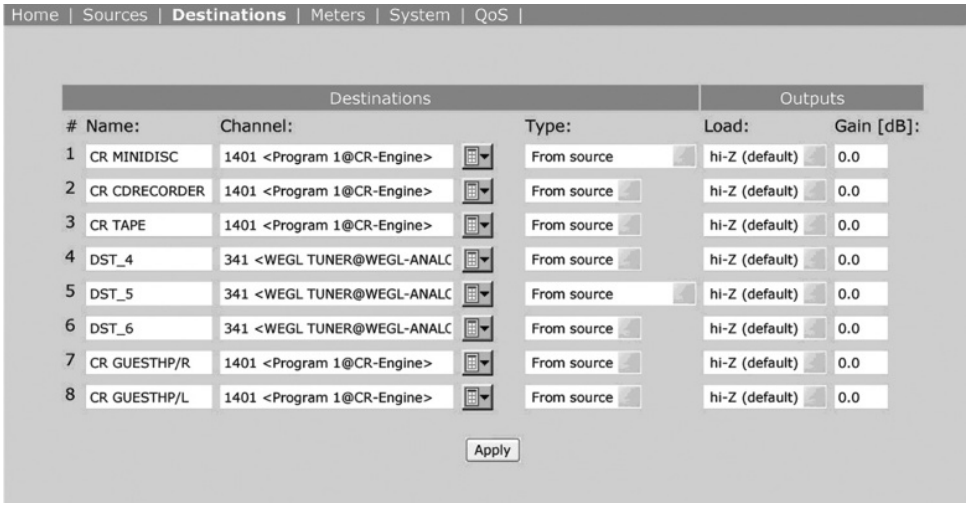


FIGURE 4.4

Destinations (from the network to the node outputs) are selected in similar fashion. Clicking on the icon to the right of the channel number/name brings up a window that lists all the available channels. For Type configuration, you can enter either From Source or To Source. The usual setting is From Source; To Source is for backfeeds such as mix-minus sends to hybrids and codecs.

FIGURE 4.5

Axia router selector node.

generator" that emulates the one from a hardware sound card are included. (When there is a physical sound card, the crystal oscillator that drives the converter chip clock starts a chain of events that causes audio files to be pulled off the hard drive at the rate set by the clock. This needs to be emulated. It's done by performing a software "Phase Lock Loop" on the networked Livewire clock and using that to run the buffer requests between the driver and Windows, which in turn pulls data from the audio application.)

There is a virtual GPIO function to convey button-press-like data from the network to applications. This would be used, for example, for a console fader on a button to start an audio player.

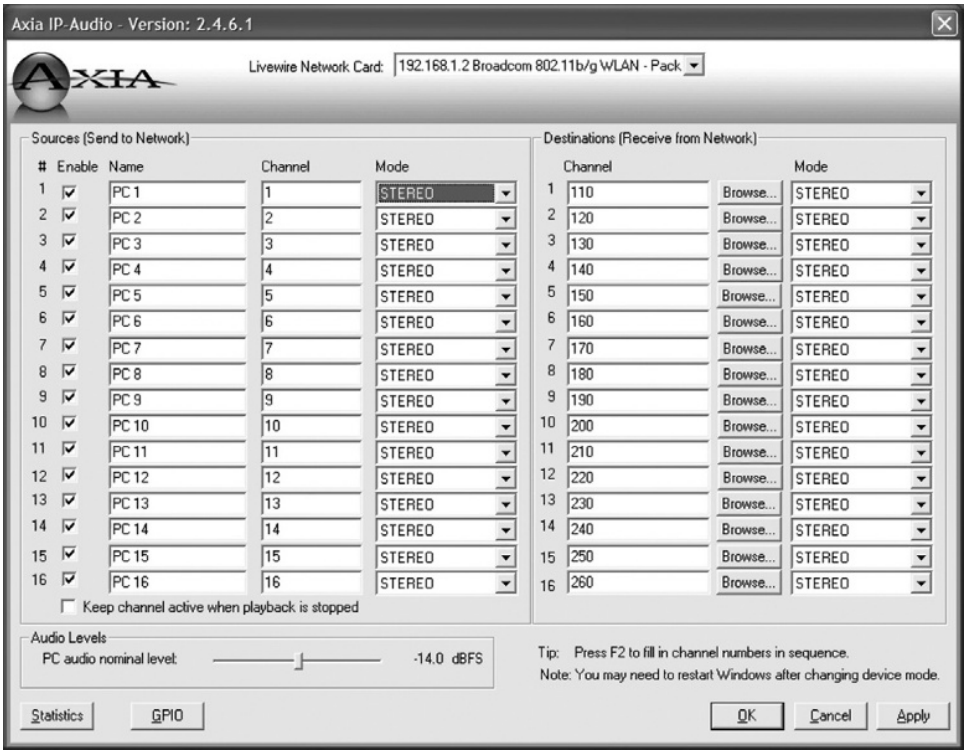


FIGURE 4.6

The Axia Windows audio driver has a setup interface where source and destination audio channels can be specified.

4.3.5 iPlay (PC Router Selector)

iPlay displays and lets users select Livewire streams (Figure 4.8). Where the Axia driver is a software version of a basic hardware node, this is essentially a software version of the router selector. Unlike the driver, iPlay has a user interface for operator selection of channels. Sources are listed and a mouse click chooses the one to take. The list can be configured to have a category filter. The Preview function allows direct listening.

There is a capability similar to the radio buttons on the hardware router selector. Dragging a listed source to one of the buttons allows it to be used to quickly select a desired source. As an alternative, there is a media player interface. Using it, Livewire streams are presented within the player's interface as if they were standard Internet streams. This works with players that can access Internet URLs, such as Microsoft Windows Media, iTunes/Quicktime, and Winamp.

FIGURE 4.7

To PC applications, the Livewire network looks like multiple sound cards.

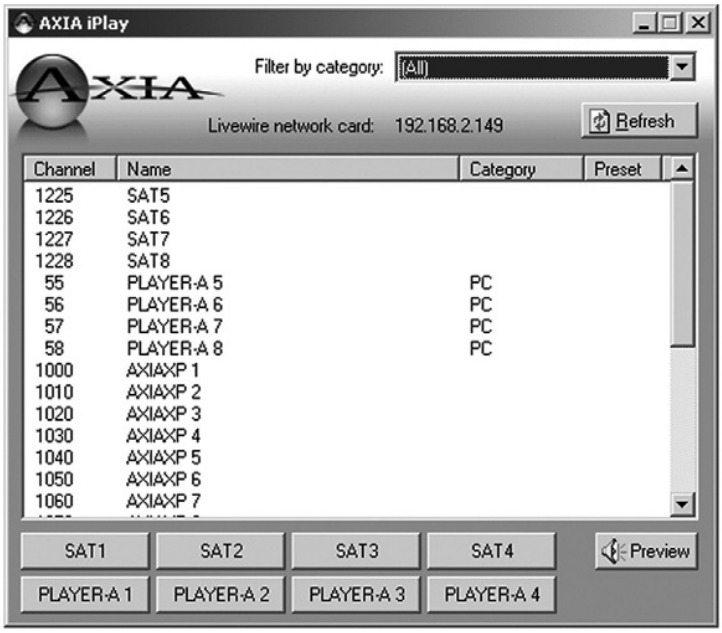


FIGURE 4.8

Axia iPlay application.

4.3.6 Axia Element Mixing Console

The control surface connects to the PowerStation rack unit via a CAN (Controller Area Network) bus cable, which also conveys power (Figure 4.9). The PowerStation performs the mixing and processing functions, with per-channel mix-minus feeds, multiple outputs and monitor feeds, equalization (EQ), microphone and headphone processing, etc. There's plenty of CPU headroom to support future features. It also provides two microphone inputs, four analog line inputs, six analog outputs, and two AES inputs and outputs. There are four GPIO ports. The internal Ethernet switch serves 16 100BASE-TX Ethernet jacks for connecting automation PCs, VoIP phones, accessory modules, Livewire nodes (for audio I/O expansion), and anything else that needs an Ethernet port. Half of these are powered to enable direct connection of PoE (power over Ethernet) devices. For uplinking to a core switch, 2-gigabit copper ports are supplied with support for optional fiber interface modules.

An expansion module adds a redundant power supply and doubles GPIO and audio I/O.

The Element surface and PowerStation may be used to build a one-room, non-networked studio. Still, there is benefit from the Livewire capability: An automation PC equipped with the Axia Livewire driver can be directly connected via Ethernet with multiple audio channels and control being conveyed over one cable, the telephone interface can have a one-jack connection for send/receive audio and control, Ethernet-based accessory modules can be connected, etc.

FIGURE 4.9

Axia Element console with PowerStation engine.

The 2-gigabit uplinks would connect to a core switch in a multistudio installation. It's also possible to connect PowerStations in a ring configuration, without using a core switch.

A web interface is used for Element's configuration. Each Element console can have separate profiles for individual users, allowing store and recall of processing and EQ settings, fader layouts, and other operational preferences.

Element benefits from Livewire's inherent backfeed and GPIO integration. Mix-minus for phones and codecs is automatic and transparent. Every channel has the ability to provide a mix-minus output. When operators select a phone or codec source, the backfeed is automatically generated based on preferences established during profile configuration. There is a single button that selects a Phone Record mode when users need to record phones off-air for later play. A drop-in phone control module can be used to operate Telos' phone interfaces, with the control data flowing over the existing Ethernet and no additional wiring being needed.

4.3.7 Pathfinder Routing Control Software

Software Authority's Pathfinder is a client-server system that provides facility-wide control over any number of supported Livewire devices. The server and client (user interface) applications run on Windows PCs. At the most basic level, Pathfinder controls audio routing via a user interface familiar to users of high-end audio/video routers. But because it has a rich networked connection to all the devices in a system, it can do much more than routing control.

User-operated controllers may be either software applications on PCs or hardware panels. There is a panel designer tool that allows an installer to create custom panels as software applications. The same tool can be used to customize hardware panels that have LCD buttons. Text and icons can be placed on each button along with various color backgrounds, which can change depending on the router state. The action caused by a button press can be configured. Whether software or hardware, controllers communicate with the server over the network via TCP. (See Figure 4.10.)

FIGURE 4.10

Hardware control panels for user interface to Pathfinder in 8- and 16-button versions. LCD button caps let you program text and icon graphics, which can change depending on the state. These are also available in console drop-in forms.

Scenes (presets) can be created and recalled to allow changes to the local studio or to the global network. A virtual patch-bay function provides a graphical way to manage routes.

Because Livewire nodes put audio level information onto the network, Pathfinder controllers are able to display level metering. On the default crosspoint display, green dots indicate the presence of audio. Clicking on these bring up accurate multi-segment meters along with faders that can adjust node gain, virtual-mix gain, and even motorized console faders. (See Figures 4.11–4.13.)

Pathfinder includes a silence detector that allows you to put a "watch" on a particular audio channel. If the audio level falls below a specified threshold for longer than a specified period of time, the system can be made to switch to a backup audio source. This lets you build automatic redundancy into a signal path. If the primary and backup sources and destinations in the silence detector are assigned to different Livewire units and these units are wired to different AC power sources, the signal path can be maintained even in the event of a failure of an interface node or power source.

You can use Pathfinder to make "virtual" routers, which can be subsets of the full system. For example, if a Livewire system has 128 different sources and destinations on the network, but you only want to use a small number of these points in a

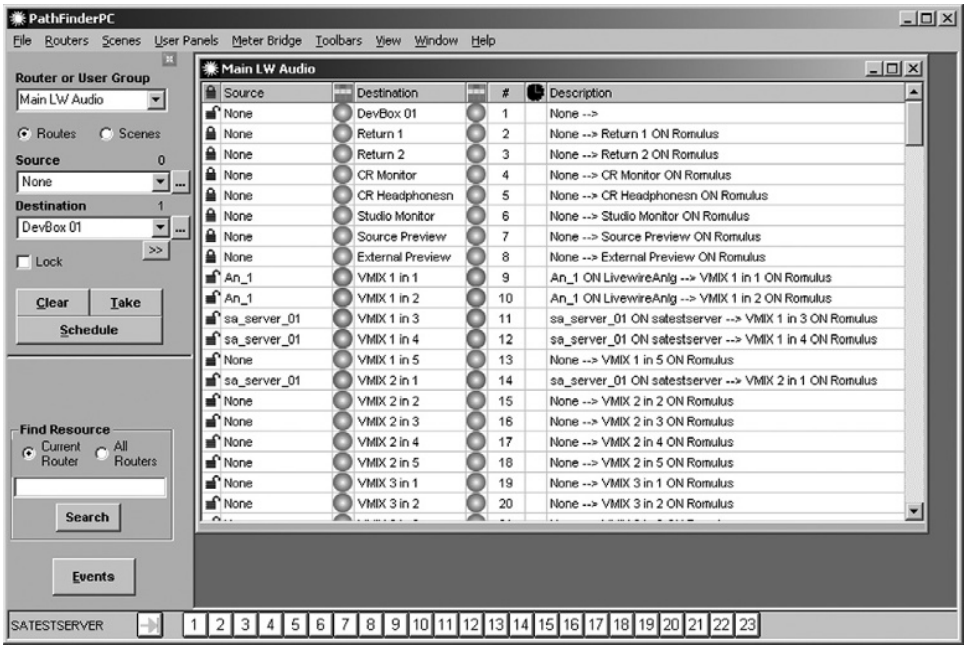


FIGURE 4.11

The Pathfinder default main routing window.

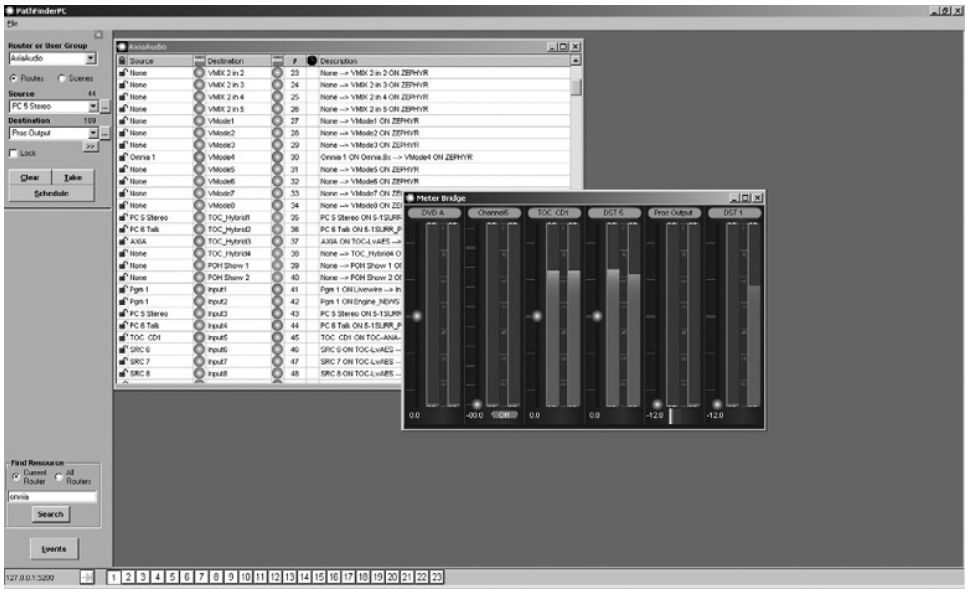


FIGURE 4.12

Clicking on a crosspoint brings up metering and level adjustment options.

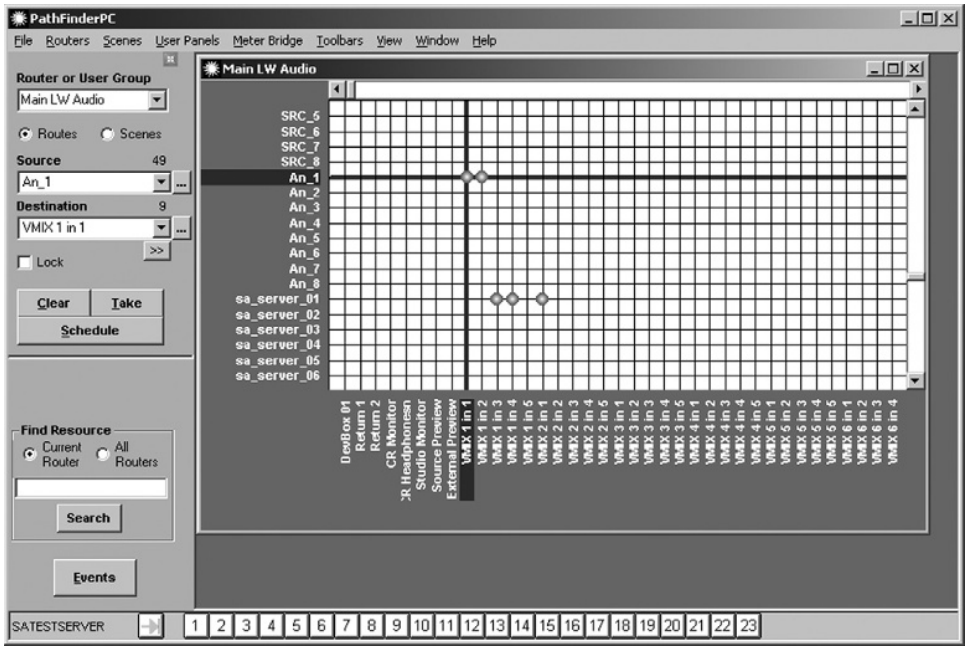


FIGURE 4.13

Alternative routing display with the grid-style format that some users prefer.

particular studio area, you can create a virtual bay that includes only the sources and destinations required by this studio. This virtual router can have its own set of scene changes. The virtual router also allows you to map multiple points to a single virtual point. For example, you can make a virtual source and destination that contains both the audio inputs and outputs for a particular device and also the GPIO points. Thus, when the route connection is made, both audio and GPIO are routed simultaneously.

Pathfinder supports non-Livewire routers including the video routers and machine control routers that are used in television plants. Thus, you can make routing points in the virtual bay that will simultaneously route audio, video, GPIO, and machine control. Pathfinder supports the use of tie lines or gateways between routers. For example, if a system has both an analog video router and an SDI video router, one or several tie lines can be wired through analog to SDI converters between the two routers. Pathfinder will then combine the routing tables and automatically use the tie lines when necessary to get analog sources to the SDI router. This capability allows Livewire terminals to extend an older and already filled router.

Multiple Pathfinder servers can be "clustered" and each can simultaneously monitor the Livewire network, building redundancy into the control system. Since every interface node in a Livewire system is an independent device, there can be a high degree of redundancy and the server can automatically switch audio to a different unit if the usual one fails. With careful planning, you can arrange your system so that the primary and backup audio units are connected to different LAN switches, which are chained together using the standard Ethernet redundancy protocols.

Pathfinder has a timed-event system built into the server, with which you can program events to happen at specified times. Individual routes or scenes can be triggered at a particular time and date or on a rotating schedule on certain days and times of the week. Events can also be created that will monitor a GPIO and initiate a scene change or route whenever a GPIO condition changes. For more sophisticated timed operations, external automation systems can access and manipulate the routing tables provided by the Pathfinder server using the included protocol translator.

There is also a "stacking events" capability that allows an installer to create complex event logic. It provides the power of a scripting language without the need for programming. For each stacking event, you define a list of qualifiers (conditions) and a list of actions. If all the qualifiers are true, the actions are taken. Possible qualifiers are GPIO state changes, audio level triggers, user button presses, time/date range, or other inputs from Livewire devices such as a profile change signaled from a mixing console.

Here's an example: Say we want to create a talkback button. Pressing it causes a talkback microphone to be substituted for the usual program audio that feeds a headphone output. This could either be a "virtual" press on a PC-hosted software panel or a real button on a hardware panel.

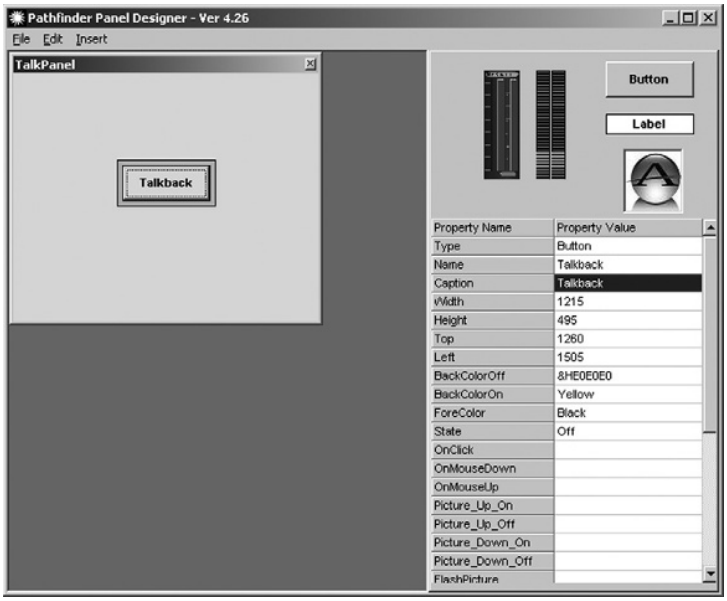


FIGURE 4.14

Making a custom panel with a button designated for talkback.

First, you will need to create the button panel that includes the talkback button using the Panel Designer application. We'll give our panel the name "Talkpanel." Dragging a button onto the panel creates it (Figure 4.14). You can then enter values for the button properties: its text, size, style, and color.

We're going to keep it simple for the example, but you could make this panel with pretty graphics and any number of nice-looking buttons arranged according to taste.

Next we need to build the "stack event" to go along with this panel. To do this, you open the Stack Event Editor tool. From the Toolbox, we dragged the User Button Press icon up to the Qualifiers box. We also dragged the Activate Route icon to both the Actions Qualified box and the Actions Invalid box. This is because we want action both when the button is pressed and when it is not, connecting the talkback upon pressing and restoring the program audio upon release. (See Figure 4.15.)

Clicking on the User Button Press icon brings up a window (not shown) that lets you fill in various properties for the button. At minimum, we need to identify it as "Talkpanel.TalkButton" and set the state to down. This tells the stack event that the qualified action should be triggered upon a press.

Clicking on the Activate Route icon in the Actions Qualified and Actions Invalid boxes will bring up windows to let you enter the details for the route that should be made in each case. The logic flow for the stack event is already designed, but now we need to enter the specifics for each qualifier and action.

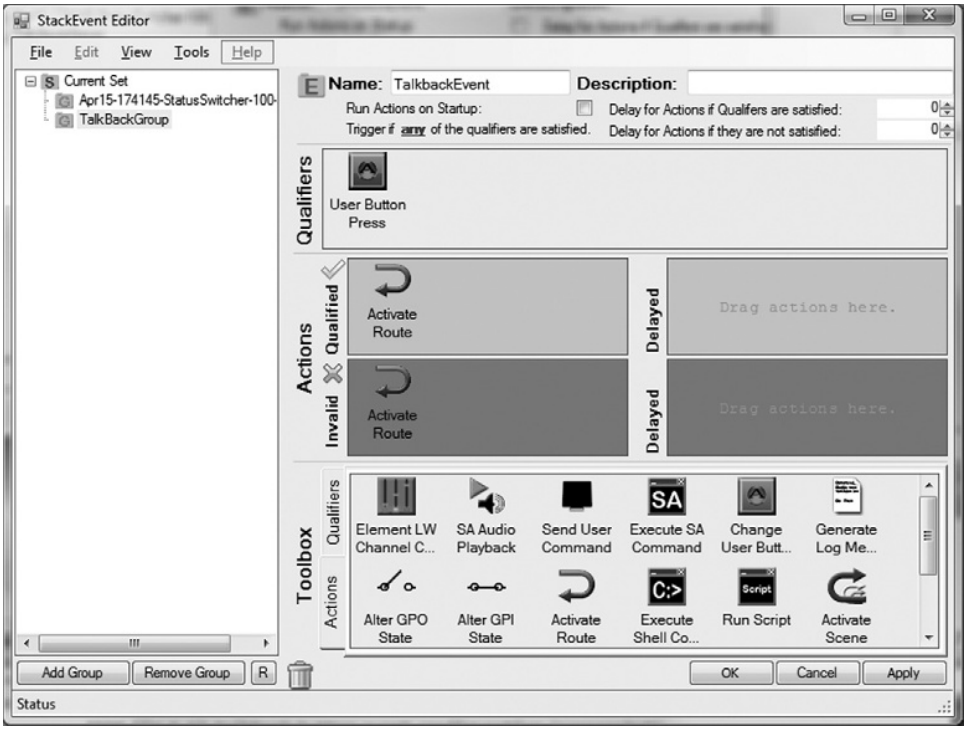


FIGURE 4.15

Talkback button event configuration.

Destination 1 on Router 1 will be the headphone output that we are switching. Pgm 1 will be the usual program feed and Talk-CR will be the talkback microphone. Upon the button press, we want to connect Talk-CR to Destination 1 HP.

Finally, in the Actions Invalid box, clicking the Activate Route icon gives you another window that looks like the one in Figure 4.16, to enter the details for the condition under which the button is released. In this case, we will again choose Destination 1 HP, but set the Source to Pgm 1. This will be the route connection made when the button is released, restoring the usual feed.

Hopefully, this example has given you a sense of what is possible. Notice the other options in the Toolbox? You can create simple or vastly complex logic with a variety of qualifiers and actions to cover any need that should arise.

4.3.8 Axia Intercom System

The Axia intercom system is similar to the ones commonly used in television facilities, but with all the benefits of IP. Since the intercom audio is Livewire, it may be picked up by any device on the network such as a mixing console. Codecs, such as the one in Figure 4.17, can be used to extend the system over WAN links.

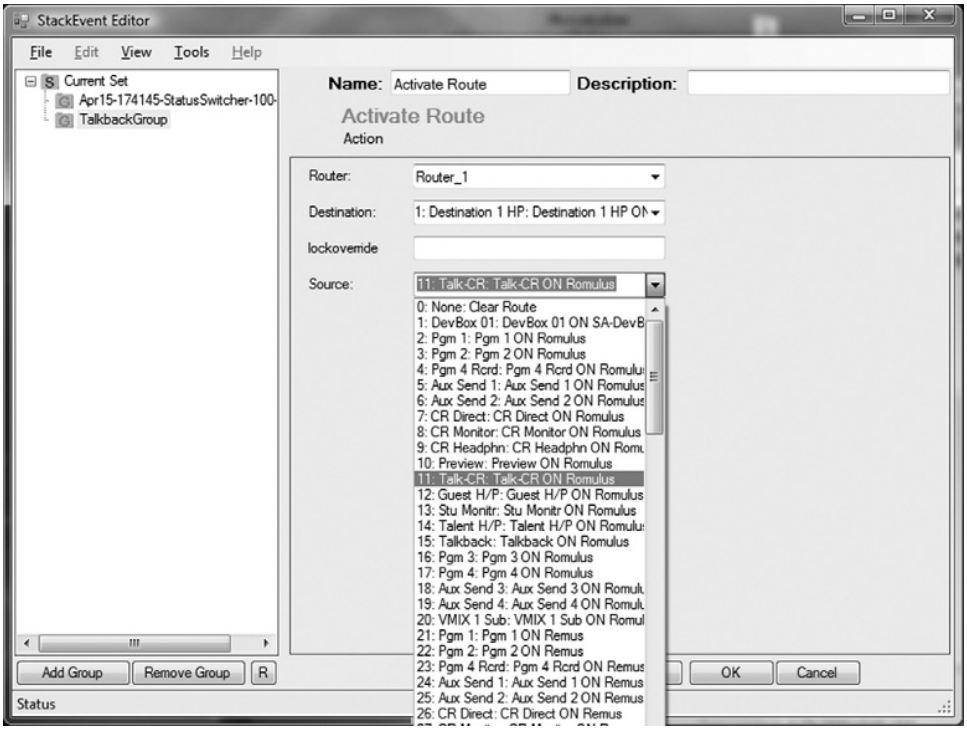


FIGURE 4.16

Talkback button Actions Qualified configuration. When the talkback button is pressed, the source Talk-CR is connected to Destination 1 HP. The source may be either selected from a list as you see here or entered directly.

FIGURE 4.17

Axia intercom panel.

The panels come in 10- and 20-station versions, in both rack-mount and console drop-in formats. They feature an advanced acoustic echo canceller (AEC) that lets you use the console operator's microphone and a loudspeaker. The operator can crank up the volume without having the usual feedback and echo problems that plague intercom systems without such an AEC function.

4.3.9 Telos iPort Codec

Each iPort contains eight stereo MPEG-AAC codecs (Figure 4.18). It can be used to extend Livewire systems over WAN networks that have good QoS. Low cost and simplicity are a result of a single Gigabit Ethernet being used for all Livewire I/O. The same connection can be used for the MPEG-compressed streams, or a second Ethernet can be used in order to provide firewall isolation between the Livewire and WAN networks.

The iPort can be optionally used in a 16-channel encode-only mode. This could provide streams to SHOUTcast-style servers for public Internet consumption or in-house distribution/monitoring applications (Figure 4.19). The iPort's MPEG codecs include AAC, HE-AAC, AAC-LD, and MP3 at a range of bitrates. These are selectable on a per-channel basis.

4.3.10 Telos Nx12 and Nx6 Telephone Interfaces

The Nx12, Telos' first phone system with a native Livewire connection, includes four hybrids and two program-on-hold inputs (Figure 4.20). Making audio connections the usual way would require six input cables and four output connections. With Livewire, a single RJ-45 gets them all at once. With the same connection, all the control is covered as well.

FIGURE 4.18

Telos iPort 8 × 8 MPEG codec.

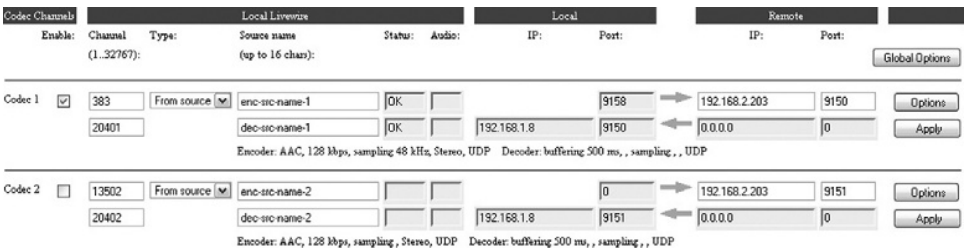


FIGURE 4.19

The configuration web page for the iPort looks similar to those for nodes. It has the same source and destination items, but also includes codec mode settings.

FIGURE 4.20

Telos Nx12 telephone interface.

Telos' VX VoIP-based system is another telephone interface option. It serves multiple studios with dozens of lines. See Chapter 6 for a full discussion of VoIP in the studio environment and a description of the VX approach to phone interfacing.

4.3.11 Omnia 8x Dynamics Processor

The 2U (two rack-unit) box in Figure 4.21 holds eight Omnia processors. It can be used in front of encoders for Internet and satellite feeds, etc. It can also be used as a headphones processor. A single Gigabit Ethernet interfaces all the audio I/O. Configuration is via a web interface. The Omnia One and Omnia 11 on-air processors include Livewire interfaces, as well.

4.3.12 Fraunhofer Institute "Content Server" Encoders

This is a family of encoders that are used to generate streams for DAB, DRM, DMB, and mobile phone broadcasting (Figure 4.22). A single Ethernet connection can interface multiple audio sources directly from a Livewire network. The encoder is incorporated into transmitters from a number of manufacturers.

4.4 CHANNEL NUMBERING AND NAMING

An advantage of having a data network carrying our audio streams is that we can send identifying information on the same cable and system. Receivers can build tables of available audio, and testers can identify specific streams on a cable. In Livewire, we have both a numeric and a text ID for each audio source.

FIGURE 4.21

Omnia 8× dynamics processor.

FIGURE 4.22

Livewire-equipped encoder for DAB, DRM, DMB, and mobile phone applications.

Hardware Livewire devices are configured either using a networked PC's web browser, or with local pushbuttons and displays. PC Livewire nodes have a configuration window that opens when you click on the application icon. Details for each are specific to the product, but the general approach is the same for all audio and GPIO.

4.4.1 Channel Numbers

Channel numbers may range from 1 to 32,767. You assign these to audio sources as you wish. New units are preconfigured from the factory to start with channel 1, thus an 8-channel node will come assigned to channels 1–8. Two new units can be connected to each other with a "crossover cable" (described in Section 5.1.6) for immediate out-of-the-box testing. For your network, you should reserve channels 1–8 for testing and not assign them for routine use. Then, if you plug a new unit into the network before you configure the channels, there will be no problem with conflicts.

In a large system, you will probably want to have a people-friendly naming and numbering system that reflects studio use or location and helps prevent accidental duplication of channel assignments (a big no-no by the way). You have plenty of numbers to use, so you don't have to conserve them. For example, the channels associated with studio 1 could start with 100, studio 2 with 200, etc. There is no requirement that channels be assigned in order or contiguously from a multichannel device.

4.4.2 Text Name

The text name may be up to 24 characters, freely chosen. This is what will appear on the web configuration pages where audio is selected, the router selector node's LCD, mixing console source select lists, etc. A typical name might be "ST1CD2" for Studio 1, CD player 2.

4.4.3 Sources and Destinations

Livewire uses the terminology *source* and *destination* to refer to audio inputs and outputs to and from hardware nodes and other devices. *Input* and *output* is too confusing, since every output is also an input, and vice versa. Therefore, *source* and *destination* unambiguously refer to the signal direction from the Livewire network's perspective.

- *Source.* This is audio sent to the network. It becomes an audio channel that can be accessed by Livewire devices.
- *Destination.* This is an audio output from the network. A Livewire node would deliver this to an analog or AES3 connector. The Axia PC driver could send the audio to an editor. A console mixing engine would use it as an input.

4.4.4 Backfeeds and Mix-Minus

Devices such as telephone hybrids and codecs need audio in both directions. When appropriate, a single channel contains a "to device" (backfeed) audio stream as well as the usual "from device" audio. You can think of the Livewire channel number as something like a telephone number that connects a call with audio in both directions. The advantage of this bundling of the two audio directions is that the association is automatically maintained through routing changes, fader assignment on mixing consoles, and other operations.

Axia consoles automatically generate backfeeds to devices that need them, creating the text name for these in the form To:sourcename. For example, if you have a source called Hybrid 1, the mixer will generate an audio stream named To:Hybrid 1. This audio will be like any other on the network and can be accessed by any device that needs it. Normally, however, it will be consumed by the device the created the original source audio.

This is one of the simplifying benefits of the Livewire approach to studio audio. With either AES or analog, you would need two connections and the association would need to be maintained independently.

You can even use this idea to tie feeds to an announcer's headphones to the channel associated with his or her microphone.

Livewire is naturally ready for intercom system application, where bidirectional audio is the norm.

4.4.5 GPIO

GPIO channels usually share the same channel number as an audio source and the GPIO automatically follows the audio source. A typical situation would be when you have a CD player that needs start control from the mixing console. The console automatically generates the start command and puts it on the channel number you assigned to the audio source. To cause a particular hardware GPIO to output this command as a contact-closure pulse, you configure the GPIO device to listen to

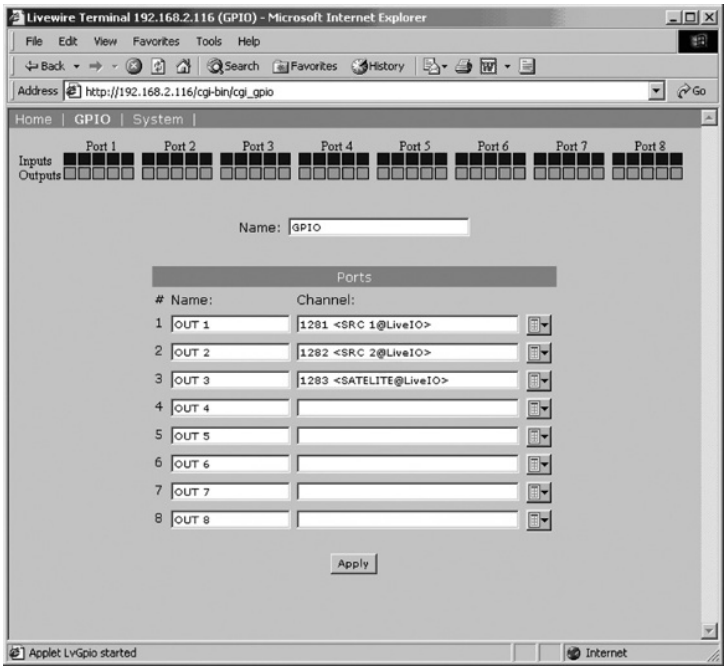


FIGURE 4.23

GPIO configuration and state monitoring. This web page is available on any device that supports GPIO, including dedicated GPIO nodes and console-associated integrated power supply/CPU/engine/GPIO rack units.

this channel. As with the backfeed audio, control follows the audio source to whichever fader is being used. In the other direction, the mixing console automatically looks for GPIO messages on channels that are assigned to faders. (See Figure 4.23.)

Physical connectors on Livewire devices are DB-15s. They interface five inputs plus five outputs. The meaning of those inputs and outputs is source-type dependent. (CR host microphone, studio microphone, control room monitor, line, phone/codec, etc., each have GPIO assignments appropriate to their function.) Appendix A on *http://www.axiaaudio.com/manuals/files/axia_gpio_v2.2_12-2008.pdf* has a rundown. Table 4.1 shows an example, in this case for the GPIO associated with the control room monitor.

While GPIOs are usually linked to audio sources, they may also be independent. In this case, the Livewire system provides a pass-through function where outputs follow inputs—sort of like a GPIO distribution amplifier.

In large systems, you might need a control path for custom GPIO signals. In this case, you can route control from one GPIO port to another without any audio being associated and the console not being involved. We call this a *GPIO snake*. A common application example is cueing to affiliate stations from a radio network head-end.

Table 4.1 Control Room Monitor GPIO Logic and DB-15 Pin Assignments

Name	Pin	Type	Notes
Inputs			
Mute CR command	11	Active low input	Mutes CR speakers and PREVIEW speakers
Dim CR command	12	Active low input	Allows external dimming of CR monitor speakers
Enable EXT PREVIEW command	13	Active low input	Feeds external audio input to PREVIEW
Not used	14	Active low input	
Talk to EXT command	15	Active low input	Turns on Talk to External Audio
Outputs			
CR on-air lamp	1	Open-collector to logic common return	Illuminates when CR monitors are muted
DIM CR lamp	2	Open-collector to logic common return	Illuminates when CR monitors are dimmed
PREVIEW lamp	3	Open-collector to logic common return	Illuminates when PREVIEW is active
Not used	4		
Talk (to CR) active lamp	5	Open-collector to logic common return	Active when a source has activated Talk (to CR)
Power and Common			
Source common	7	Logic common	Connect to ground of source device or to pin 8
Logic common	8	Internal 5-volt return	Can be connected to pin 7 if source is not providing common
Logic plus 5-volt supply	9	Logic supply, individually fused	Can be connected to pin 10 if source is not providing voltage; active only when source has been assigned to channel
Source supply	10	Common for all five inputs	Connect to power supply of source device or to pin 9
Not used	6		

CUE TIP As a provider of programs to affiliate stations, WOR/NY has more than 100 GPIO channels not associated with audio sources that are used for "network cues." A number of network cue switches are installed in every studio. The studio that is actually on-air sends signals to a GPIO node that provides GPIO to a satellite encoder. Pathfinder PC is aware which studio is on-air and makes sure the custom GPIO channel follows the air-chain audio. The GPIO snake channel is associated with audio, but not in the usual way; audio and GPIO routing are controlled in parallel by Pathfinder PC.

GPIOs need not be linked to hardware connectors. Just as audio can flow to and from PC applications with no connection other than Ethernet, GPIOs can be connected by the network and software alone. Automation systems use this to avoid having to use awkward physical GPIO cards and wiring. See Section 4.8 for more on this topic.

4.4.6 V-Mix and V-Mode

Axia engines such as the one used as the backend to the Element console and the platform for the Omnia 8x and iPort include a "virtual mixer" (V-mix) capability. This can be used to create new streams from mixing existing ones. There are eight mixers, each with five inputs. A master summed output is also available. There are eight independent V-mode (virtual mode) channels. V-mode provides the left/right/sum selection that is usually found on console inputs. (See Figure 4.24.)

Control is via web page and Livewire Control Protocol commands sent over the network. For example, the Pathfinder application can manually or automatically make mixing and mode selection adjustments.

WINS, a news station in New York City, uses this to automatically mix background sounds with the announcer microphone. Radio Free Europe in Prague uses the feature extensively for production of its program feeds.

Normally, an engine already in place would provide the mixing as an ancillary function. But if no suitable engine is present, one could be installed specifically for the purpose.

4.5 DELAY

In packet-based systems, delay is an important issue—and keeping it acceptably low is an essential aspect of an IP studio audio system's design. Packetizing audio for network transmission necessarily causes delay, and a careful strategy is required to reduce this to acceptable levels. Internet audio delay is often multiple seconds because the receiving PCs need long buffers to ride out network problems and the delays inherent in multiple-hop router paths. However, with fast Ethernet switching on a local network, it is possible to achieve very low delay. To do this, we must have a synchronization system throughout the network. This also avoids

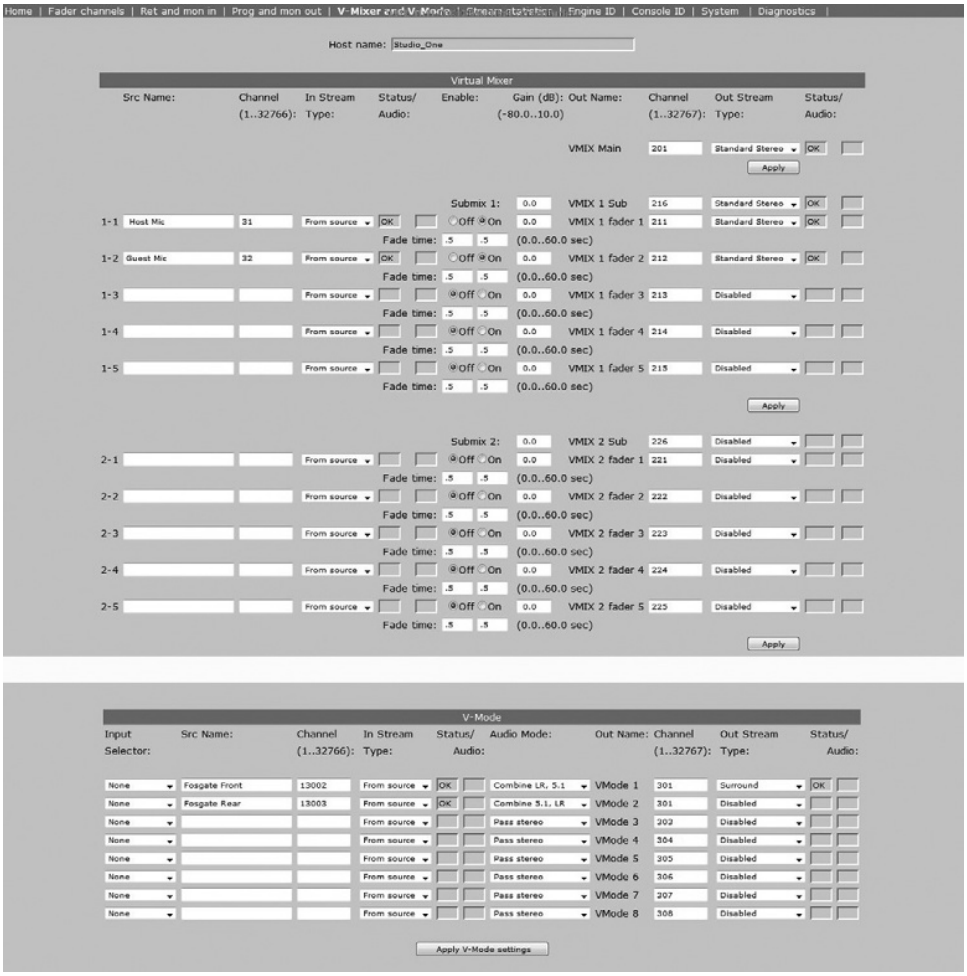


FIGURE 4.24

V-mix and V-mode are available as auxiliary functions inside Livewire engines. This screenshot of the web interface control is used for testing. The actual control would usually be from a software application such as Pathfinder.

sample or packet slips that cause audio dropouts. Internet streaming does not use this technique, so even if it were to have guaranteed reliable bandwidth, you still couldn't achieve the very low delay we need for professional studio applications.

For Livewire, we generate a system-wide synchronization clock that is used by all nodes. Within each node, a carefully designed phase lock loop (PLL) system recovers the synchronization reliably, even in the case of network congestion. Hardware nodes provide this clock, and in each system there is one master node that sends

the clock signal to the network. If it should be disconnected, or stop sending the clock for any reason, another node automatically and seamlessly takes over. (See Section 4.7.3 for more on this topic.)

In broadcast studios we care very much about audio delay in the microphone-to-headphones path for live announcers. Maximum delay must be held to around 10 ms, or else announcers will start to complain of comb-filter or echo problems. We need to consider that this is a total "delay budget" and that multiple links and some processing will often be in the path. So we've decided to have a link delay of around 1 ms end-to-end for anything in this path, allowing us a few links, or maybe a couple of links and a processor.

In our experience, delays to around 10 ms are not a problem. From 10–25 ms announcers are annoyed but can work live; anything above 25–30 ms is unacceptable.

Here is another way to think about delay: Audio traveling 1 foot (0.3 meters) in air takes about 1 ms to go this distance. And another data point: A common professional A-to-D or D-to-A converter has about a 0.75-ms delay.

As is universally the case in engineering, there is a trade-off, otherwise known as the "if you want the rainbow, you gotta put up with the rain" principle. To have low delay in a packet network, we need to send streams with small packets, each containing only a few accumulated samples, and send them at a rapid rate. Bigger packets would be more efficient because there would be fewer of them and they would come at a slower rate, but they would require longer buffers and thus impose more delay. Big packets would also have the advantage that the necessary packet header overhead would be applied to more samples, which would more effectively use network bandwidth.

With Livewire, we enjoy our rainbow and avoid the rain by having different stream types: Livestreams use small and fast packets, while Standard Streams have bigger and slower packets.

Livestreams require dedicated hardware and achieve the required very low delay for microphone-to-headphone paths. PCs with general-purpose operating systems are not able to handle these small packets flying by so quickly, so they use the Standard Streams. The network delay in this case is around 5 ms and the PC's latency is likely to add perhaps 50–100 ms more. Since PCs are playing files and are not in live microphone-to-headphone paths, this is not a problem. Our only concern is how long it takes audio to start after pressing the on button, and for this, delays in the range of Standard Streams are acceptable.

Standard Streams can also be sent from the network to PCs for listening and recording. Again, this small delay is not an issue here, especially given that PC media players have multiple seconds of buffering.

All Axia hardware Livewire nodes can transmit and receive both stream types, determined by a configuration option. Livewire's streams also have a fixed, constant delay, regardless of the system size or anything else. In fact, a source being received at multiple nodes will have a differential delay of less than 5 μs—less than a one-quarter sample at Livewire's 48-kHz rate.

4.6 LEVELS AND METERING

Livewire transparently conveys 24-bit audio words, and is no different from any other digital audio system that has this dynamic range. Nevertheless, there is a lot of confusion regarding digital audio level setting and metering throughout the industry, and with Livewire being somewhat of a clean slate, we have an opportunity to clear up some of this.

It is said that fish have no chance to understand the nature of water. Perhaps the same is true of American broadcast engineers, most of whom have been immersed in VUs all of their working lives (Figure 4.25). Steve recently had a conversation with a European broadcast practitioner that caught him by surprise and set him on a course of discovery. Over a course of fish and chips (or was it tea and crumpets?), the engineer told him that 6-dB console headroom was enough and 9 dB was usual and plenty. Huh? Wouldn't that mean pretty much full-time clipping and distortion? It turns out, no.

4.6.1 Headroom

Volume units (VU) were originally developed in 1939 by Bell Labs and broadcasters CBS and NBC for measuring and standardizing the levels of telephone lines. The instrument used to measure VU was called the volume indicator (VI) meter. Everyone ignores this, of course, and calls it a VU meter. The behavior of VU meters is an official standard, originally defined in ANSI C16.5-1942 and later in the international standard IEC 60268-17. These specify that the meter should take 300 ms to rise 20 dB to the 0 dB mark when a constant sine wave of amplitude 0 VU is first applied. It should also take 300 ms to fall back to −20 dB when the tone is removed. This integration time is quite long relative to audio wavelengths, so the meter effectively incorporates a filter that removes peaks in order to show a long-term average value.

The ratio of peak to root mean square (RMS) for sine waves is 1.414, or approximately 3 dB. This ratio is called the *crest factor*. Voice, sound effects, and music have

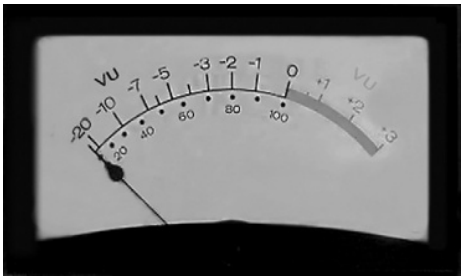


FIGURE 4.25

Classic VU meter. It's ubiquitous, but do we really understand it?

crest factors much larger than sine waves, ranging from around 5–20 dB, with 14 dB being typical. This is the part of the audio that is "hidden" by the VU meter. That is, the meter shows less by this amount than the absolute waveform peak value. That's why consoles with VU meters need 20-dB headroom; generally about 14 dB is used to cover the invisible peaks, and the remaining 6 dB is "true headroom," held in reserve for those moments when the sportscaster suddenly gets wound-up ("GOOOOOOOAAALLL!!!").

So why don't we simply have meters that read the peaks? Perhaps it's an accident of history that tradition maintains. VU meters were designed in the age of mechanical d'Arsonval movements and had to obey the laws of physics. There is just no way those old metal pointers would be able to trace out the audio peaks. Nor would we want them to, as it turns out—the frenetic wiggling would be tiring to watch, indeed. But there is, in fact, another good reason for the VU meter's sluggishness: It does give a reasonably good estimation of the human ear's sense of loudness. Long before today's use of aggressive program audio processing, this meant something to the operators riding manual gain on live broadcasts—and it still does today (with some caveats).

But have you noticed that digital devices like PC-based audio editors usually have peak-reading meters? Their waveform displays are like storage oscilloscopes that can accurately show peaks and the meters are made to correspond to the levels traced by these editing displays. They are usually marked in dBfs—that is, dB down from full scale. And this is how we need to think about audio levels in the *digital* context.

With analog, the numbers on the meter are relative to whatever value we decide to choose for the voltage level on the connection circuit. We nonchalantly misuse the decibel as if it were a voltage, when in fact the dB is the logarithmic ratio of two power levels. A VU meter is actually an AC voltmeter with strange markings. 0 dBu (*not* 0 VU) corresponds to 0.775 V_{RMS}, and the other dB values on the meter are referenced to that. While this voltage may seem an odd choice, when applied to the 600-Ω load used by vintage gear, the power dissipated is 1 mW—a nice, clean point of reference. The modern U.S. practice is that the 0 VU mark on the VU meter corresponds to +4 dBu, or 1.228 V_{RMS}. (The *u* in dBu stands for *unloaded*. This is in contrast to dBm, which assumes the 600-Ω load, and is therefore referenced to 1 mW—a power reference, true to the dB's original derivation. Not so long ago, however, +8 dBm was the norm in the U.S. broadcast and telecom industries, and other countries still use a variety of values today, which we discuss more in section 4.6.3.)

With digital systems, we have an unambiguous and universal anchor—0 dBfs—as the maximum absolute clipping point. That is why DAT tape recorders first abandoned the VU meters that were common on analog tape machines for bargraph meters marked in dBfs, and other digital recording systems thereafter have followed suit.

Turning back to our headroom discussion, let's first consider the analog case. The clip point for most modern studio equipment is +24 dBu. With +4 dBu nominal operating level, we arrive at 20 dB for headroom. If we want the same headroom

in a digital system, we should set our nominal operating level to −20 dBfs. And this is just what U.S. TV and film people usually do, following the SMPTE recommendation RP155.

With all this as background, we're ready to rejoin our delightful lunch companions with the refined accents and milky tea. Americans love their slow meters, but the Brits say that VU means *virtually useless*—and they have a point. For as long as American broadcasters have been staring at VU meters, our British cousins have been gazing into their beloved BBC-style peak programme meters (PPMs). These have a rise time 30 times faster than VU meters and a fallback time of 2.8 seconds. Because of the slow fallback time, they look lazier than a VU, but actually they are much more accurately registering peaks. (The slow release is designed to make it easier for the human eye to register the peak value displayed by the meter.) (See Figure 4.26.)

Now fastidious Euro-engineers would be careful to call the meter a *quasi*-PPM (QPPM) to note that it is not indicating *absolute* peaks. With its 10-ms rise-time filter, the PPM will still miss about 3 dB in "hidden" momentary audio excursions for sounds with the highest crest factors. Thus, British engineers usually set their maximum operating level to −9 dBfs in digital systems, so they have 6 dB for the excited sportscaster reserve. Hmmm . . . isn't that the same as for the VU contingent stateside? Yup.

As those well-known audio engineers Led Zeppelin once sagely informed us, sometimes words have two meanings. Context matters. When Americans speak about their 20-dB headroom, it is in the context of their slow attack–time VU meters (which ignore up to 14 dB in peaks). When Brits pronounce on their "perfectly adequate" 9-dB headroom, they are referring to systems with fast attack–time PPM indicators (which miss about 3 dB of peaks). Both achieve about the same result most of the time (i.e., 6 dB of true operational headroom).

So the "language" of headroom is fluid. The preferences we design into systems on the bench must be adapted by operators who work with real-world audio and somewhat-deceptive level meters. Thus, the designer's headroom of 20 dB and the

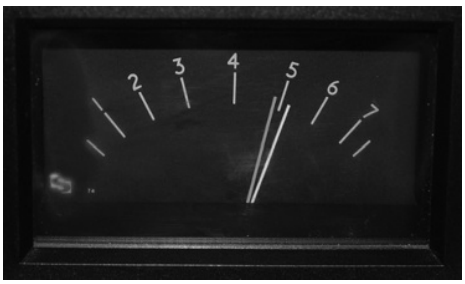


FIGURE 4.26

BBC PPM (dual-pointer version, for stereo display on a single meter). (Photo by Spencer Doane/iStockphoto LP.)

audio mixer's headroom of 6 dB are just different metrics for essentially the same amount of protection from distortion in actual practice.

Given this discrepancy, why not solve it by just using absolute peak meters? At least Euro-broadcasters (who are closer to this point traditionally) might consider such an option, but no. Here is an explanation from a paper on program meters written by the IRT, a research institute attached to the German public broadcasting service:

> *Regarding fast digital "sample programme meter" (SPPM) theoretically no headroom is needed. Those meters are appropriate to control signal peaks with respect to clipping but they are not as suitable as QPPM regarding adequate programme levelling. For example, signals with high proportion of peaks tend to be under-leveled whereas strongly compressed with limited peaks tend to be over-leveled. This can result in grave loudness leaps, which seem to be more intensive than using a QPPM.*[1]

Thus, the PPM seems to provide a good compromise between displaying absolute audio waveform peaks and consistent operational level control.

4.6.2 Alignment

One other important difference between the two meters involves alignment of electrical audio levels and gain structure between devices. Although it is less true in the digital age than it was in the analog audio era, not all devices in an audio system will have identical dynamic range characteristics. It is therefore important to optimally align all devices in the system around an operating-level reference point for optimum advantage, so that no individual device runs out of dynamic range (i.e., clips) before any of the others. This is, of course, the reason that steady-state tones are used to set audio reference levels between connected devices.

For the VU meter, the 0-VU point is used for *both* the setting of such sine-wave operating reference tones and as the usual operator target for maximum audio signal level. On the other hand, the PPM's greater sensitivity to peaks means that the lower crest factor of sine waves will not deflect the meter as much as typical audio signals. So sine-wave alignment reference tones are set at a lower level than the typical audio gain–riding target on PPMs, typically 14–18 dB below the meter's maximum level. (This level is sometimes labeled "TEST" on PPMs.)

So the VU meter retains the singular advantage of having its "nominal level" and "reference level" at the same value (0 VU), marked with an unmistakable scale change from a black line to a big red bar. This certainly adds to its ease of use for operators.

[1] Siegfried Klar and Gerhard Spikofski. On Levelling and Loudness Problems at Television and Radio Broadcast Studios. In: preprint #5538, 112th AES Convention, Munich, 2002.

4.6.3 International Variants

Note also that while the PPM's ballistics and its visual appearance (easy-to-read white pointer and white markings on a black face, in its analog form) are standardized worldwide, its meter-face labeling is not. The BBC-style PPM has only numbers from 1 to 7 (no units) on the meter's marks, with no other labeling to tell operators the normal operating level or the maximum level. Operators are simply trained to set sine-wave alignment tones at 4, and ride gain such that audio levels do not exceed the 6 mark. (To aid novice operators, the BBC's motto, "Nation shall speak peace unto nation," has been adapted to "Nation shall peak six unto nation.")

The European Broadcasting Union (EBU) has opted for a more transparent approach. The EBU digital PPM is labeled with dBfs values, has a "reference level" mark for system alignment at −18 dBfs, and a color transition signaling permitted maximum level at −9 dBfs. Meanwhile, the EBU also had a number of other standard PPM labeling methods back in the analog days (as did another European standards body, DIN), all of which are being gradually replaced by the preferred EBU digital scale.

The new German IRT meter is the same as the EBU digital meter, but is labeled somewhat like a VU with a 0-dB mark and green/red transition at −9 dBfs and a reference level mark at −9 dB, which corresponds to −18 dBfs. Yet another variant is the Nordic N9 meter, which has the word "TEST" marked at −9 dBfs and a compressed scale above this point.

What about the rest of the world? Mostly, Latin Americans and Asians have followed the United States' VU meter approach. Even in Europe, the French, Spanish, and Italian state broadcasters and most commercial broadcasters throughout the continent favor the good-old VU (although often with the meter's 0 VU reference corresponding to 0 dBu rather than +4 dBu). And unlike the PPM, the VU's markings are universal, again simplifying its use for operators who may travel among different locales.

Table 4.2 compares some of the most common audio level metering standards in use worldwide today.

The Nordic, EBU digital, and IRT meters are usually presented as bar graphs today. The IEC IIa and IIb meters are both analog meters that are the same except for their scale markings. The IRT meter is also the same as the EBU digital meter except for the scale marking.

All of these different approaches are paths to a common goal: maximizing signal-to-noise ratio (SNR) on transmission links and recorders while avoiding clipping distortion. European engineers argue that PPMs do this best since operators can see the peaks and thus can ride gain closer to the limit without getting into clipping trouble. While this is true, it doesn't seem to matter that much in practical application. With 24-bit digital paths and >100-dB dynamic range analog converters becoming the norm, the few extra dB accuracy in level setting that PPMs permit is not all that

Table 4.2 Some Standard Audio Program Meter Types Currently in Use Around the World

Specification	A.k.a.	Scale Marks	Reference Level[a]	100% Mark[b]	Attack Time	Decay Time
IEC 60268-10/I	Nordic N9	−36, −30, −24, −18, −12, −6, TEST, +6, +9	TEST = 0 dBu	+6 dBr	80% in 5 ms	20 dB in 1.5 sec (13 dB/sec)
IEC 60268-10/IIa	BBC	1, 2, 3, 4, 5, 6, 7	4 = 0 dBu	6	80% in 10 ms	24 dB in 2.8 sec (8.6 dB/sec)
IEC 60268-10/IIb	EBU	−12, −8, −4, TEST, +4, +8, (unlabeled mark at +9), +12	TEST = 0 dBu	+9 dBr	80% in 10 ms	24 dB in 2.8 sec (8.6 dB/sec)
IEC 60268-18	EBU digital	−60 to 0 dB	−18 dBfs	−9 dBfs	80% in 5 ms	20 dB in 1.7 sec (12 dB/sec)
IEC 60268-18, IRT proposal	IRT digital PPM	−50 to +10 dBr	−9 dBr	0 dBr	80% in 5 ms	20 dB in 1.7 sec (12 dB/sec)
ANSI C 16.5 (IEC 60268-17)	VU meter	−20 to +3 dB	0 VU = +4 dBu	0 VU	99% in 300 ms	20 dB in 300 ms (67 dB/sec)

[a] *Reference level is the alignment point for tones. dBr = dB relative to reference level.*
[b] *100% mark is the gain-riding target for program audio (also called* permitted maximum level *or* nominal level *in various locales).*

consequential. On the other hand, with broadcast program processors automatically adjusting gain over a wide range, the VU's advantage that it approximates the impression of loudness is also becoming irrelevant (though for this reason it may remain useful on studio recording and live-sound mixing boards).

4.6.4 Terminology of Audio Level Metering

Given this divergence, let's pause to gather definitions. As always, agreeing on what words mean is a useful step in achieving understanding.

Alignment level or *reference level* is an anchor metering level (and a corresponding electrical voltage) that exists throughout the system or broadcast chain, which can be used as a guide to adjusting equipment gain controls for optimum system operation.

In the United States, alignment and nominal levels are the same: 0 dB on a VU meter, usually corresponding to an analog sine-wave voltage of +4 dBu (1.23 V_{RMS} = 3.47 V peak-to-peak [p-p]). The clipping level in analog systems is usually +24 dBu, resulting in 20-dB headroom. Carried over to digital systems, the alignment/nominal level is −20 dBfs, as standardized in SMPTE RP155.

For the EBU/IEC PPM, the alignment and peak operating levels are not the same. For analog signals, the "100%" mark on the meter corresponds to 0-dBu nominal level (0.775 V_{RMS} = 1.1 Vp-p). In the digital domain, the nominal level is −9 dBfs, while the alignment level is −18 dBfs. The EBU specifies a PML (see below) of −9 dBfs, thus giving 9-dB headroom.

Nominal level only has meaning for VU meters. It corresponds to the "100%" mark on the meter that an operator uses as his or her target for maximum audio level. The phrase *nominal level* was invented for VU meters to suggest that the value is not "real," but a filtered compromise and approximation.

Permitted maximum level (PML) has meaning in the context of PPMs. This is the level that operators should peak to with program audio. It's often where a color transition occurs on the scale.

Crest factor is the ratio of the peak (crest) value to the RMS value of a waveform. Since it is a ratio, it is a dimensionless quantity, but it is often associated with corresponding decibel values to indicate the peak excursion above RMS values in terms that equate to commonly understood audio levels and metering. (For brevity and operational clarity, crest factor is often quoted only in dB form.) A sine wave has a crest factor of ∼1.4 (the waveform's peak value equals 1.414 times its RMS value), which corresponds to peaks at 3 dB above RMS. Music and sound effects can have a wide crest factor range, varying from ∼4 to 10 (or 12 to 20 dB). Human voice signals can exhibit crest factors of ∼1.5 to 3 (or 4 to 10 dB).

Headroom is the difference between nominal level on VU meters or permitted maximum level on PPMs and the analog clip point or digital full scale, in dB.

The effective headroom in a system is an interaction of the meter scale, the meter ballistics, and the analog reference voltage. As noted, VU meters need up to 18-dB headroom to accommodate audio crest factors, whereas PPMs only require 9 dB. In both cases, there are "invisible peaks" that are masked by the meter filtering, but a VU meter has 13- to 16-dB invisible peaks, while a PPM has only 3–4 dB.

For a closer look, let's analyze both the digital and analog cases, comparing VU meters and PPMs:

- *Digital.* What matters is the PML value, which is almost always −9 dBfs for PPMs. This is the green/red boundary on the meter or the "100%" mark that operators are supposed to use as the peak limit. But this is being read with a QPPM that indicates 10–12 dB higher than a VU on a typical program audio. So this is equivalent to −21 to −19 dBfs as read on a VU—pretty much the same as the VU's usual −20-dBfs nominal peak level.
- *Analog.* For the VU there is 20-dB headroom (+24 − +4 = 20). For the European PPM case, the Livewire analog node audio gain has to be adjusted so that −18 dBfs = 0 dBu. (Offset +6 on the input and −6 on the output from our default setting of −20 dBfs = +4 dBu setting.) This is a 6-dB lower level on the analog circuit for a given digital value, which means 6 dB *more* headroom relative to the VU analog clip point – except that it doesn't matter since the digital limit is only +18 dBu. So because the European operating level is 0 dBu, there is 18-dB headroom, which is only 2 dB less than with VUs.

In both cases, the Europeans are metering more accurately with respect to peak detection, so they probably have a bit more headroom in actual operation. Using a PPM, they cannot be "fooled" by program audio that has a high crest factor, as an operator using a VU meter might be. Nevertheless, most of the time, taking into account both the filtering and specified operating level, the effective headroom for the VU meters and PPMs is approximately the same.

Meter integration time is defined by CCIF as "the minimum period during which a sinusoidal voltage should be applied to the meter for the pointer to reach within 2 dB of the deflection that would have been obtained from a continuous signal." This is roughly equivalent to the attack time.

dBu refers to an RMS voltage level without reference to a particular load (the *u* in *dBu* means *unloaded*). In usual modern practice, this will apply to a low impedance output connected to a high impedance load. 0 dBu is referenced to 0.775 V_{RMS}. Chosen for historical reasons, this is the voltage level at which you get 1 mW of power in a 600-Ω resistor, which used to be the standard reference impedance in professional audio circuits.

dBfs (or dBFS) applies to the audio level referenced to the full-scale value in a digital system. Levels will be expressed as negative values relative to the 0-dBfs clip point.

dBm is a nearly obsolete term from the era when all audio outputs were terminated by 600-Ω loads. 0 dBm is referenced to 1 mW.

4.6.5 Livewire Levels

As we've seen, levels will depend on the meter reference markings and ballistic characteristics, since these are what guide operators to adjust to a particular value with real audio content.

Analog levels also depend on the node level adjustment. Axia Livewire analog nodes have input and output level adjustments calibrated in dB that can be used to change the correspondence between digital values and analog voltage levels. For standard U.S. operation, these should be set so that −20 dBfs = +4 dBu. For standard European operation, these will be set so that −18 dBfs = 0 dBu, which requires an offset of +6 dB on the input and −6 dB on the output from the default U.S. setting. (The end result is that U.S. configurations trade 2 dB of SNR for 2 dB greater protection from clipping relative to most European installations, which is probably a good idea given the different metering typically used in each locale.)

Since metering is what determines the digital level on the network, and software lets us offer options for various standards and preferences, that is just what we do with the Axia consoles that have soft display capability. Were we still in the days of moving coils, we'd have to swap the physical meter to satisfy local requests. And PPMs needed to have electronic help to achieve their fast attack times, so that would have meant yet more physical variation. But software and LCD screens make having options straightforward. We can now show either VU, PPM in its various forms, or absolute peak just as easily as the other. Using a bar graph plus peak-line approach, we can even show VU- or PPM-averaged levels and absolute peak values at the same time. With the Axia Element mixing consoles, users have the option to choose their preferred meter style:

- U.S.-style VU (with extended range)
- EBU digital
- IRT digital PPM
- Nordic N9
- BBC PPM

With all of these, there is an optional true-peak bar that holds and displays the maximum digital value like the ones commonly found on PC audio editors. This has an instant attack time and a nonlinear release time that holds a new peak value for three seconds with no change, then releases at a 20-dB/1.5-sec rate.

4.6.6 Aligning Consoles to PC Audio Applications

Unlike broadcast consoles, PC audio editors generally have true peak-reading meters. Nevertheless, aligning the two with a sine-wave reference tone is straightforward.

- Due again to the crest factors discussed earlier, the level of a sine-wave tone on a console VU meter will be approximately 3 dB lower than what the PC's meter displays. This is a result of the VU's filter causing a near-RMS value to be displayed, which is around 3 dB lower than peak. If the console has a true

peak-reading bar graph meter, it should match the PC's meter on a sine-wave reference-level tone.
- All of the PPM types will correspond to the PC's meter without the 3-dB difference. (You might think that the 5- to 10-ms attack filter on the PPM would round off the peak to a lower value like VU meters do, but this doesn't happen on a sine-wave tone due to the nature of the waveform and interaction between the meter's attack and release times.) Remember, of course, that the PPM's "TEST" level should be used for all reference alignments, and the PC level should be set so that the PC meter reads at the same value (typically −18 dBfs).

If you experience any unexplained discrepancies, check the level setting in the Axia PC driver and any other gain controls that might be in the signal path, such as the Windows mixer.

4.7 DEEP STUFF—HOW THE LIVEWIRE TECHNOLOGY WORKS

4.7.1 Quality of Service

As you've seen, an important concept in a converged network for live media applications is quality of service (QoS). When general data are the only traffic on a network, we only care that the available bandwidth is fairly shared among users and that the data eventually get through. But when studio audio and general data are sharing the same network, we need to take all the required steps to be sure audio flows reliably.

Livewire's method for achieving QoS is system-wide, with each relevant component contributing a part of the whole:

- *Ethernet switch.* Allows an entire link to be owned by each node. Isolates traffic by port.
- *Full-duplex links.* Together with switching, eliminates the need for Ethernet's collision mechanisms, and permits full bandwidth in each direction.
- *Ethernet priority assignment.* Audio is always given priority on a link, even when there is other high-volume, nonaudio traffic.
- *Internet Group Management Protocol (IGMP).* Ensures that multicasts—audio streams—are only propagated to Ethernet switch ports connected to devices that are subscribed to a particular stream.
- *Limiting the number of streams on a link.* Livewire devices have control over both the audio they send and the audio they receive, so they can keep count and limit the number of streams to a number that a link can safely handle.

The result is solid QoS, providing the ability to safely share audio and data on a common network.

4.7.2 Source Advertising

This process allows Livewire devices to dynamically populate the list of audio sources available on a network. Audio source devices advertise their streams to the network on a special multicast address. Receive devices listen to these advertisements and maintain a local directory of available streams. The advertisements are sent when the streams first become available, and at 10-second intervals after that. (Actually, only the data version number is sent every 10 seconds. The full data are advertised only upon their initial entry to the system, and upon any change, or upon explicit requests from those having detected the data version number increase.) If a node's advertisements are not received for three consecutive periods, it will be assumed to have been removed from service. There is also an explicit "stream unavailable" announcement so that devices can get the news more quickly when an audio source is switched off.

Receive devices maintain a local table of available streams and their characteristics, which is updated as any new information arrives. Sources are cleared from local tables when an explicit message is received announcing that a stream is no longer available, or when three consecutive advertisements have been missed.

A receive device may be configured to be permanently connected to particular multicast streams, or users may select audio sources from a list. Configuration options determine if the list displays all available sources or only a filtered subset.

4.7.3 Synchronization

Livewire needs careful system-wide synchronization so it can use the small buffers it requires for its low-latency performance. If Livewire didn't have a distributed way to derive a bit clock, there would eventually be buffer overflow or underflow resulting from the input and output clocks not maintaining exactly the same frequency. This synchronization also keeps the multiple audio sources in the correct time relationship with one other. Were this not the case, there could be phase cancellation problems with a multiple-microphone setup, for example.

A phase lock loop (PLL) in each Livewire device recovers the system clock from a multicast clock packet that is transmitted at a regular interval. At any given time, one Livewire hardware device is the active system clock master. In the event the master develops a fault or is removed from service, the local PLLs in the nodes are able to "ride out" the brief interruption until a new clock master is established, during which time, the smooth flow of audio is maintained.

This PLL and Livewire synchronization in general are essential components that permit the system to work over the range of conditions that it encounters on real-world networks. The PLL is a combination of hardware and software that accommodates packet jitter and loss without disturbing the accurate flow of clocking to converters. It is comprised of a packet detector, a smart filter, and a digitally controlled oscillator. The result is a differential delay of less than 5 μs network-wide, or less than a one-quarter sample at Livewire's 48-kHz rate.

All hardware nodes are capable of being a clock source, and an arbitration scheme ensures that only the one unit with the highest clock master priority is active.

When the clock goes away for three consecutive periods, all nodes begin transmitting clock packets after a delay skewed by their clock master priority status. When a node sees clock packets from a node with a higher priority on the network, it stops its own transmission of clock packets.

You can specify the clock-master priority behavior so that it can be made predictable (Figure 4.27). A particular node can be made to always be the master, with another node set as backup, for example. Each node has a clock master configuration setting that can range from 0 to 7:

- 0 = never (slave only)
- 7 = always (forced master)

The factory default is 3. So all units have equal priority out of the box, and the following is used to break ties (in descending order):

- Lowest Livewire audio transmit base channel
- Lowest IP address
- Lowest Ethernet address

Livewire nodes have an LED-labeled master on their front panel that illuminates when that unit is the clock master.

In order to achieve fast lock, the clock stream is normally transmitted at a high data rate. On LANs, this is a good strategy, but sometimes Livewire is carried over channels that have bandwidth constraints. In this case, a low-rate mode can be selected, which reduces the data rate by a factor of about 10.

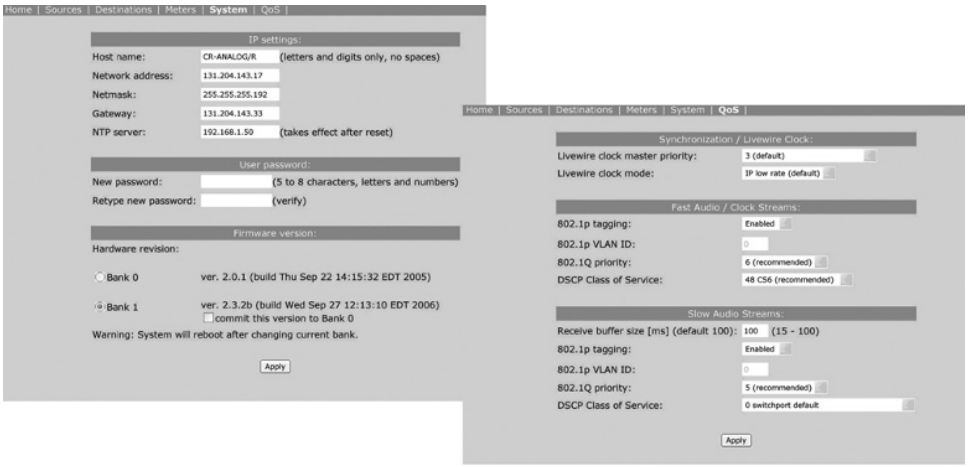


FIGURE 4.27

Web page for configuring the clock master priority in Livewire nodes. This page includes options for Ethernet priority tagging, IP class of service, and other maintenance functions.

There must be at least one hardware node in a system to provide the clock source. Two would be better, to provide redundancy. As you might imagine, it is not good to have more than one priority 7 (forced master) node in a system.

To avoid passing audio through sample-rate converters, the Livewire network can be synchronized to your AES master clock, if you have one. A Livewire AES node provides this function, recovering the clock from an attached AES input, and creating a Livewire sync packet. When this is done, the sample-rate converters in the AES nodes are switched off, and there is bit-for-bit transparency between the two systems. (Indeed, Livewire can be used as an AES-over-IP transport system.) For this to work, the AES node must be the active clock master.

4.7.4 Livewire's Use of Multicast Ethernet and IP Addresses

Livewire uses addresses within the range specified for "organization local scope" used by IANA (the Internet Assigned Numbers Authority). Routers do not propagate traffic on these addresses to the Internet, so they stay contained within LANs. (We also set the "link local" bit and TTL = 1 in the IP header to further ensure that streams stay local.) Since AoIP is used within a single facility on a single switched LAN, this range is appropriate.

The range supports Livewire's 32-k channels, with up to 120 stream types per channel. Livewire only uses four types now, so there is plenty of room for growth.

The motivation for mapping each type to a contiguous block is to allow configuration of switches and routers on a per-type basis by specifying an address range. This direct mapping of channels to addresses also makes sniffing easier: It is simple to know where to look for a particular audio stream.

Livewire channels range from 0 to 32,767. Audio streams are mapped into IP multicast addresses using the channel numbers for the lower 15 bits, as shown in Table 4.3. The multicast addresses in Table 4.4 are used for system functions.

Livewire streams are multicast at both layers 2 and 3. The Livewire channel number is automatically translated to the appropriate addresses at both layers internally. You might want to know the translation algorithm because you or your network engineer might need to check packets with a "sniffer" or Ethernet switch diagnostics.

IP addresses are mapped into an Ethernet MAC layer multicast, according to a de facto standard process for this procedure. This process is as follows:

- Identify the low-order 23 bits of the class D IP address.
- Map those 23 bits into the low-order 23 bits of a MAC address with the fixed high-order 25 bits of the IEEE multicast addressing space prefixed by 01-00-5E.

Example:

- Assume: channel = 80
- Assume: stream type is standard stereo stream
- Then: IP address = 239.192.0.80 (dotted decimal)
- Then: Ethernet MAC Address = 01-00-5e-00-00-50 (dashed hex)

Table 4.3 Multicast Address Mapping for Audio Streams

IP Address	Type
239.192.000.0/15	Livestream and Standard Stereo Streams
239.192.128.0/15	Four addresses are for system functions, others are reserved
239.193.000.0/15	Backfeeds for Standard Stereo Streams
239.193.128.0/15	Reserved
239.194.000.0/15	Reserved
239.194.128.0/15	Reserved
239.195.000.0/15	Backfeeds for Livestreams
239.195.128.0/15	Reserved
239.196.128.0/15	Surround streams
239.251.000.0/15	Reserved
239.251.128.0/15	Reserved

Table 4.4 Multicast Addresses for System Functions

IP Address	Function
239.192.255.1	Livestream clock
239.192.255.2	Standard Stream clock
239.193.255.3	Advertisement channel
239.193.255.4	GPIO (UDP port 2060)

Ethernet addresses are transmitted most-significant byte first, but least-significant bit first within the byte, so in our example it is the 1 in the leftmost MAC address byte 01 that signifies a multicast address.

4.7.5 Livewire Packet Format

As noted previously, there is a fundamental trade-off in the choices a designer must make for audio packet structures: When there are more samples per packet we have more efficiency, translating to more link capacity and less processing power required, but at the expense of longer delay. Good design means finding the best compromise. You've seen already that Livewire gives you two variants to satisfy different requirements: *Standard Streams* and *Livestreams*. The packet structures of

these two stream modes are different, which allows them to be optimized for either high efficiency or low delay.

A description of the two stream types' packet structures follows, but first, let's review some basic delay issues common to all streams. We start with "packet time"—the audio sampling rate and the number of samples that are combined into a packet—then consider other factors, as follows:

- Packet time = 1/sampling rate × samples per packet

Livewire uses a one-packet buffer on the send side and a three-packet buffer at the receive end; adding this to the switch latency, total link delay is therefore defined as follows:

- Link delay = packet time × 4 + switch latency

Standard Streams

Standard Streams use large packets to be efficient with both computer resources and network bandwidth, as shown in Table 4.5. They are usually chosen when PCs are the audio devices. Note that Standard Streams also offer a half-size "variant" format, shown in the last row of Table 4.5.

An Ethernet frame's maximum data length is 1500 bytes, so you can see that we have chosen to pack the Ethernet frame to nearly the maximum possible. There are two reasons for this:

1. The frame rate is the lowest possible to put the least burden on PC receivers.
2. The header overhead is applied to the most data so the proportion of capacity devoted to audio versus overhead is highest.

Table 4.5 Standard Stereo Stream Packet Format

Function	Bytes	Notes
Interpacket delay	12	This is not actually transmitted, but is an Ethernet requirement and must be taken into account for bandwidth calculations
Ethernet header	26	Includes the VLAN/priority fields
IP header	20	Standard
UDP header	8	Standard
RTP header	12	Standard
Audio	1440	240 samples at 48 kHz, 24 bits, stereo
Audio (variant)	720	120 samples at 48 kHz, 24 bits, stereo

Note: *Total bytes per packet = 1440, with core delay = 5 ms (respective values of 720 bytes and 2.5 ms using the variant format).*

Livestreams

Livestreams are specialized for low delay, so we pack only a few audio samples into each packet, as shown in Table 4.6. Because they are smaller, less buffering is needed, and that means the latency is lower. These are usually chosen for anything that is in the live DJ microphone-to-headphone path.

The header load for RTP/UDP/IP is 40 bytes per packet, which takes a significant piece of the network bandwidth, given that the audio payload is only 72 bytes. Fortunately, this is usually of no consequence, since there is plenty of bandwidth on modern LANs.

2 + 5.1 Surround Streams

Livewire inherently carries multiple audio streams and surround mixing is a built-in feature of the Axia Element console and engine, so it is ready for radio and TV surround (see Table 4.7).

Table 4.6 Livestream Packet Format

Function	Bytes	Notes
Interpacket delay	12	This is not actually transmitted, but is an Ethernet requirement and must be taken into account for bandwidth calculations
Ethernet header	26	Includes the VLAN/priority fields
IP header	20	Standard
UDP header	8	Standard
RTP header	12	Standard
Audio	72	12 samples at 48 kHz, 24 bit, stereo

Note: *Total bytes per packet = 72, with core delay = 0.25 ms.*

Table 4.7 Surround Stream Packet Format

Function	Bytes	Notes
Interpacket delay	12	This is not actually transmitted, but is an Ethernet requirement and must be taken into account for bandwidth calculations
Ethernet header	26	Includes the VLAN/priority fields
IP header	20	Standard
UDP header	8	Standard
RTP header	12	Standard
Audio	1440	60 samples at 48 kHz, 24 bit, stereo + 5.1 (eight channels)

Note: *Total bytes per packet = 1440, with core delay = 1.25 ms.*

Surround streams accommodate eight channels, carrying the 5.1 multichannel and a stereo mix version simultaneously. Surround streams carry these eight channels in the following order: front left, front right, center, low-frequency enhancement (LFE), back left, back right, stereo left, and stereo right.

4.7.6 Link Capacity

The speed of the link, the size of the header and payload, and the number of samples that are combined into a packet determine link capacity. The more samples that are combined into a packet, the lower the header overhead, and thus the higher the efficiency and link capacity.

Each Standard Stereo Stream has a bitrate of 2.304 Mbps. A 100-Mbps link can therefore carry 43 such channels at full capacity and a 1000-Mbps link can carry 430 channels.

Each Livestream has a bitrate of 3.776 Mbps. A 100-Mbps link can therefore carry 26 such channels at full capacity and a 1000-Mbps link can carry 260 channels.

In practice, links to hardware nodes will usually carry a mix of Standard Stereo Streams, Livestreams, possibly surround streams, and control data. The biggest node has eight channels, so there is plenty of link capacity to accommodate all the streams. PCs use the more efficient Standard Stereo Streams and maybe only six of them maximum, so again there is plenty of capacity to handle both audio and simultaneous file transfers, etc. Livewire console mix engines connect with 1000-Mbps links, so the sky is the limit there.

Remember that all of the above has been concerned with *per-link* bandwidth. The *overall* system capacity is effectively unlimited with appropriate Ethernet switches.

4.7.7 Network Time Protocol

Network Time Protocol (NTP) is the Internet's standard for conveying time. There are a number of servers on the Internet that users can connect to in order to retrieve accurate time. There are also boxes from manufacturers such as EXE that receive radio time signals and translate them to NTP packets. Livewire does not need NTP, but some peripherals do. For example, Livewire studio mixing surfaces and Omnia processors use NTP to automatically synchronize to the correct time.

4.7.8 Network Standards and Resources

Livewire operates at both Ethernet and IP network layers, taking advantage of appropriate standards-based resources at each layer. Here are the resources in use at the various layers:

Layer 1

- IEEE Ethernet physical

Layer 2

- IEEE Ethernet switching
- IEEE 802.1p/Q prioritization
- IEEE 802.1p multicast management

Layer 3

- IETF IP (Internet Protocol)

Layer 4

- IETF RTP (Real-Time Protocol)
- IETF UDP (User Datagram Protocol)
- IETF TCP (Transport Control Protocol)
- IETF IGMP (Internet Group Management Protocol)

Layer 5

- IETF NTP (Network Time Protocol)
- IETF DNS (Domain Name Service)
- IETF HTTP/WebIETF ICMP Ping
- IETF SAP/SDP (Session Announcement Protocol/Session Description Protocol) (in the Windows PC Livewire Suite application)

4.8 LIVEWIRE ROUTING CONTROL PROTOCOL

As the name suggests, the main purpose of Livewire Routing Control Protocol (LWRP) is changing audio routes. This is achieved by using it over the network to specify the receive addresses at node destination audio ports. It can work for GPIO channels as well.

LWCP is a simple ASCII human-readable protocol. The document describing it is freely available from Axia. LWRP is universally supported by all Livewire devices on TCP port 93. The commands are the same in the audio nodes, GPIO node, Element console, and Axia PC driver.

Automation systems will usually connect to the Axia driver as their single interface point. There is an "LWRP server" inside the driver that connects to any devices in the system that need to be controlled. This server has an auto-discovery mechanism to find the other devices on the network. Automation systems usually provide a generic way of configuring GPIO through TCP. The configuration involves putting the address of the source of GPIO signals (IP:port). During operation, the automation systems send command text strings and looks for responses. For Livewire, the automation system connects to localhost:93 (port 93 on the local PC's TCP interface), which accesses the Axia PC driver's LWRP interface. Then it looks for commands, such as GPI 1 L as a START trigger and GPI 1 H as a STOP trigger, etc.

AUTOMATION INTEGRATION RESOURCES Real examples of LWRP applications are the Axia automation integration manuals for BE AudioVault and ENCO DAD, which are available, respectively, at *http://www.axiaaudio.com/manuals/files/Controlling_AudioVAULT_using_Axia_GPIO.pdf* and *http://www.axiaaudio.com/manuals/files/Controlling_DAD_using_Axia_ GPIO.pdf*.

Some partners provide a GUI for GPIO configuration that is dedicated to Livewire support. One example is from RCS/Prophet NextGen, available at *http://www.axiaaudio.com/manuals/files/Controlling_NexGen_using_Axia_GPIO.pdf*.

The GPIO ports presented by the IP driver or GPIO nodes are indexed by numbers 1 to N, and each can be associated with any Livewire channel. Or, they can be routed to a remote GPIO endpoint identified by the IP:port GPIO snake configuration. Thanks to this abstraction, automation systems can be statically configured to use the locally indexed GPIO ports, while Pathfinder PC can freely change GPIO routes (channel/endpoint assignment). In this way the automation system does not have to know where the GPIO control comes from. The need for this GPIO routing is routine because automation systems typically use different GPIO paths during live-assist operation than they do during full automation.

LWRP also supports transparent passing of custom messages over GPIO channels, providing a capability similar to AES3's ancillary data transmission function. An example application is sending song title text along with an audio channel. Since a single channel number can be used to reference both the audio and the associated data, the two would remain locked together regardless of any routing changes. This is documented at *http://www.axiaaudio.com/manuals/files/Using_Windows_Driver_GPIO.pdf* (see last page).

GPIO signals between nodes are carried over TCP connections, so they can be extended beyond the local network, such as for GPIO snake applications. But console-to-GPIO communication is multicast and UDP. Like the audio streams, it uses channel numbers as addresses. This requires a reliable network with multicasting enabled.

LWRP also provides audio metering and a silence/peak detector. This is how Pathfinder is able to display audio level and respond to silence-detect events.

4.9 LIVEWIRE CONTROL PROTOCOL

Livewire Control Protocol (LWCP) is used when a device needs more sophisticated control than LWRP provides. For example, via LWCP Axia consoles support remote control of channel on/off, motorized fader position, program assignments, selection of profiles, and many other characteristics. An example peripheral device interface is the Telos VX multistudio phone system, which offers LWCP for full control of line selection, dialing, etc.

Axia provides documents for this XML-like protocol for those who want to develop their own control interfaces for products that support it.

To give you an idea of how LWCP looks and works, here is an excerpt from a transaction between a Livewire console telephone controller and the Telos VX IP phone system:

```
# accepting RINGING_IN call and taking it on air
take studio.line#1
event studio next = 0
event studio.line#1 state = ON_AIR, callstate = ACCEPTED, time =
   169040
event studio.line#1 state = ON_AIR, callstate = ESTABLISHED, time = 0

# comment for the line
set studio.line#1 comment = "This is very interesting"
event studio.line#1 comment = "This is very interesting"
get studio.line#1 comment
indi studio.line#1 comment = "This is very interesting"

# seizes line
seize studio.line#2
event studio.line#2 state = SEIZED, callstate = IDLE, hybrid = 0,
   time = null

# calling from line 2 to line 5 placing call on line 2 on-air, hybrid 5
call studio.line#2 number="sip:24@192.168.0.24" hybrid=2
event studio.line#2 state = SEIZED, callstate = CALLING, hybrid = 5,
   time = 0
event studio next = 5
event studio.line#5 state = IDLE, callstate = RINGING_IN, hybrid = 0,
   time = 0
event studio.line#2 state = SEIZED, callstate = RINGING_OUT, hybrid
   = 5, time = 290

# take next call on air
take studio.next
event studio.line#5 state = ON_AIR, callstate = ACCEPTED, hybrid = 5,
   time = 295969
event studio next = 0
event studio.line#2 state = ON_AIR, callstate = ESTABLISHED, hybrid
   = 5, time = 0
event studio.line#5 state = ON_AIR, callstate = ESTABLISHED,
   hybrid = 5, time = 0
```

```
drop studio.line#5
event studio.line#5 state = IDLE, callstate = IDLE, hybrid = 0, time
   = null
```

A NOTE ABOUT PROTOCOL DESIGN There is no question that among network protocols, the Internet has been an impressive success. One of the reasons for this was the approach its designers took—and still use today—when inventing its protocols. These principles are outlined in the IETF RFC 1958 document, and were taken to heart in the design of Livewire. We repeat some of them here in summary form, and in priority order, with comments from Livewire's designers added parenthetically:

- *Make sure it works.* Make prototypes early and test them in the real world before writing a 1000-page standard, finding flaws, then writing version 1.1 of the standard. (Telos and Axia are practical, commercial companies, not academic or governmental organizations. We had two years of extensive lab tests of prototypes in two locations and then real-world field tests at radio stations before locking the core tech down.)
- *Keep it simple.* When in doubt, use the simplest solution. William of Occam stated this principle (Occam's razor) in the 14th century. In modern terms, this means: fight feature creep. If a feature is not absolutely essential, leave it out, especially if the same effect can be achieved by combining other features. (We believe firmly in this principle. We tried very carefully to add nothing unnecessary.)
- *Make clear choices.* If there are several ways of doing the same thing, choose one. Having multiple ways to do something is asking for trouble. Standards often have multiple options or modes or parameters because several powerful parties insist their way is best. Designers should resist this tendency. Just say no. (It was just us—and we did say no. No committees or politics to cause bloating.)
- *Exploit modularity.* This principle leads directly to the idea of having protocol stacks, each of the layers of which is independent of all the other ones. In this way, if circumstances require one module to be changed, the other ones will not be affected. (We built Livewire on all of the available, off-the-shelf lower layers.)
- *Expect heterogeneity.* Different types of hardware, transmission facilities, and applications will occur on any large network. To handle them, the network design must be simple, general, and flexible. (We accommodate both dedicated hardware audio nodes and general-purpose PCs as audio nodes.)
- *Avoid static options and parameters.* If parameters are unavoidable, it is best to have the sender and receiver negotiate a value than defining fixed values. (These were avoidable—we don't have any such negotiated parameters. We do have the receiver selection of stream types, but this is simple one-ended selection.)

A NOTE ABOUT PROTOCOL DESIGN—cont'd

- *Look for a good design, not a perfect one.* Often designers have a good design but it cannot handle some weird special case. Rather than messing up the design, the designers should go with the good design and put the burden of working around it on the people with the strange requirements. (This is our mantra! Make it work, make it solid, build just enough flexibility to get the job done—and no more.)
- *Be strict when sending and tolerant when receiving.* In other words, send only packets that rigorously comply with the standards, but expect incoming packets that may not be fully conformant and try to deal with them. (We always tell our software design engineers to do this. Hopefully they listened.)
- *Think about scalability.* No centralized databases are tolerable. Functions must be distributed as close to the endpoint as possible and load must be spread evenly over the possible resources. (We kept very close to this idea, which is in the fundamental spirit of the Internet. We don't have any central databases or other pieces along these lines. We have a fully distributed system. If one part fails, the others keep going.)
- *Consider performance and cost.* If a network has high costs and there are cheaper variants that get the job done, why gold-plate? (Compare the power and cost of our solution with others. Using simple, off-the-shelf, commodity parts was a guiding principle for our work.)

CHAPTER

Designing and Building with AoIP

5

In this chapter, we progress boldly from the lab to the real world. That means starting with the nitty-gritty nuts-and-bolts of cabling, connectors, and the like. Just as with analog audio, that might be all you need to know to get on with doing a simple AoIP setup. Grab a bag of ready-made RJ cables, get the plugs in the right sockets, do a bit of configuration—and go to lunch.

But building a large system involves making big-picture architecture design choices. Since AoIP is built on general data networking, you have the full range of possibilities from that world open to you in the design of your AoIP facilities. You already know much from the previous chapters, but here we'll show you some concrete examples of how you can build audio plants ranging from a simple audio snake, to a small radio station studio, to a very large audio production facility.

5.1 WIRING THE AoIP FACILITY

5.1.1 Simplification via Cabling

An important goal for AoIP is to simplify installations that would otherwise be burdened with the complexity of the variety of cables, connectors, and wiring styles that have proliferated over the years as new technologies for audio routing and transport have appeared. We'll show you how it is possible to use a common and standard cable and plug approach for everything in your plant. This goal can be achieved even when analog and AES3 components are part of an AoIP installation.

5.1.2 Structured Wiring

When you need to build a big plant, this is the way to go. In the old days, wiring was specific to the task, and often to the vendor. Each telephone, network, and audio had its own cable type and wiring protocols.

Audio Over IP

doi: 10.1016/B978-0-240-81244-1.00005-5

Today things are different. The idea at standards bodies like the Telecommunications Industry Association (TIA) and the Electronic Industries Association (EIA) is to define classes or categories of cables and accessories that can be used for all applications specified for that class. With this approach, you have a vendor-independent way to wire buildings and facilities, so that services from many vendors can be supported over time without replacing cabling and connectors. The name for this concept is *structured wiring*. The model encompasses both specification of the cabling components and the way in which they are installed.

Since structured data network cabling is standardized and widely deployed, outside contractors can install and test your cabling without having any specialist knowledge in broadcasting or pro-audio.

5.1.3 Ethernet for AoIP Systems

AoIP systems primarily use copper cables. By far, the two most common Ethernet transport technologies for AoIP on copper are 100BASE-T and 1000BASE-T (often called "gigabit Ethernet").

100BASE-TX

Although often called 100BASE-T, the current, official name for 100-Mbps Ethernet is 100BASE-TX. It is the baseline for AoIP, using copper cables with RJ-45-style plugs and jacks. It is well matched to connections from Livewire audio nodes to switches since the 100-Mbps bandwidth is plenty for up to 25 send and receive stereo audio channels.

100BASE-TX Ethernet is balanced and transformer coupled, so it has very good resistance to interference, and has no problem with ground loops.

1000BASE-T

Mixing/processing engines serve a lot of audio channels, so a step up in bandwidth is required for them. 1000BASE-T, providing 1-Gbps bandwidth, is the usual choice. PCs are routinely connected this way as well. And switch-to-switch connections are almost always gigabit, using either 1000BASE-T or fiber.

The main reason to move away from the two copper mainstays is if you wanted to mix in some fiber where it made sense to do so, such as for long runs. Note that there are other possibilities as well—dozens of Ethernet media types that have been standardized—but only a few are in wide use. Table 5.1 summarizes the principal options to be considered for AoIP applications.

The length numbers are the official ones and are conservative. The full-duplex operation that AoIP uses permits even longer runs. (In a single-duplex operation, the propagation time for Ethernet's collision-detection mechanism needs to be taken into account, but AoIP's full-duplex operation never has collisions.)

Ten-gigabit Ethernet (10-GbE) is an emerging technology, used mostly to interconnect large IP routers. There are more than 10 standards vying for acceptance. As this is being written, 10GBASE-LR and -ER have the most common usage.

On the horizon are 40- and 100-gigabit Ethernet. Standards are being formulated and some products are coming to market now.

Table 5.1 Common Ethernet Media Types for AoIP

Name	Bandwidth	Cable Type	Length	AoIP Purpose
100BASE-TX	100 Mbps	Two pairs Cat 5 copper (Cat 5e recommended for AoIP to add safety margin)	100 m	Most common Ethernet media; Livewire nodes, PCs
1000BASE-T	Gigabit	Four pairs Cat 5e copper (Cat 6 recommended for AoIP to add safety margin)	100 m	Engine to switch, PCs, switch-to-switch
1000BASE-SX	Gigabit	LED-driven multimode fiber	550 m	Switch-to-switch
100BASE-FX	100 Mbps	Multimode fiber	2 km	Switch-to-switch, audio nodes with external media converters
1000BASE-LX	Gigabit	Single-mode fiber and multimode fiber	2 km 550 m	Switch-to-switch, long runs

5.1.4 Twisted-Pair Cable Categories

Cable categories (usually abbreviated "Cat *n*") are fundamental to the structured wiring concept. The cabling specifications for the various categories are in the TIA/EIA-568-A (and -B) Commercial Building Telecommunications Cabling Standard.

The most significant differences between cables from each category are the number of twists per foot, and the tightness with which the twists and spacing of the pairs to each other are controlled. The wire pairs in a Category 3 (Cat 3) cable usually have two twists per foot and you may not even notice the twists unless you peel back quite a lot of the outer insulation. Cat 5 is twisted around 20 times per foot. Cat 6 has even tighter twisting. Each step up results in better crosstalk performance, with the benefit becoming increasingly important as the data frequency rises.

Here's a rundown:

- *Cat 3*: Pretty much obsolete for data applications, these are used only for voice-grade telephony and Ethernet 10BASE-T.
- *Cat 5*: This designation applies to 100-Ω unshielded twisted-pair cables and associated connecting hardware of which the transmission characteristics are specified up to 100 MHz. Cat 5 cables support Ethernet 100BASE-TX.
- *Cat 5e*: This is enhanced Cat 5 cable. It is quite widely deployed today because it supports both 100BASE-TX and gigabit (1000BASE-T). It is probably the minimum recommended for an AoIP installation.

- *Cat 6*: This provides significantly higher performance than that of Cat 5e. This cable has a plastic pair separator inside that holds the wires in correct relation to each other. For this reason, Cat 6 cables are larger in diameter than Cat 5 cables. Cat 6 is preferred for 1000BASE-T, but not required. Cat 6 is a good idea for new AoIP installations that need to be future-proof, require maximum reliability, and can absorb the cost increment.

A NEW TWIST Belden has a Cat 6 cable called Mediatwist that is particularly interesting for AoIP applications. Rather than being round or flat, this cable has a crescent-shaped cross-section, and the four pairs are each tightly held in molded channels. The two wires in each pair are glued together so that the twist characteristic is fixed and stable regardless of manufacturing tolerances and cable flexing.

Another characteristic of cables that needs to be considered is the insulation material. "Plenum-rated" cables are more stable with changing temperatures due to their use of Teflon rather than PVC insulation. Plenum-rated cables are required in air-handling spaces in order to meet fire regulations. Teflon produces less smoke and heat than PVC in the case of a fire, and does not "support" the spread of flames.

5.1.5 Structure of Structured Wiring

The long cables that go from equipment rooms to connection locations are called horizontal cables. They usually terminate in RJ-45s, either in patch fields or on wall jacks. Patch cords with RJ-45s at each end complete the system, connecting interface nodes and central equipment to the jacks. That's pretty much it—simple, but powerful.

5.1.6 Pin Numbering, Jacks, Cables, and Color Codes

Ethernet uses eight-position/eight-pin modular connectors. TIA/EIA specifies two standards for wiring RJ-45-style cables. The T568A color code is "preferred" by TIA/EIA, but is not so common in the United States for business installations.

The TIA/EIA T568B color code cable specification has the same electrical connections as T568A, but has the green and orange pairs swapped (Figure 5.1). This is also known as the AT&T 258A wiring sequence, and it has been widely installed in the United States. It is also used by the Radio Systems StudioHub+ system for analog and AES connections. Axia recommends it for all new installations.

Either sequence will work just fine if you have it on both ends. In either case, you have a cable with four pairs wired straight through.

Depending on the cable manufacturer, the color conductor of each pair may or may not have a white stripe. The other half of the pair is usually white with a colored stripe, but sometimes can be only white. Both the 568A and 568B formats are shown in Table 5.2.

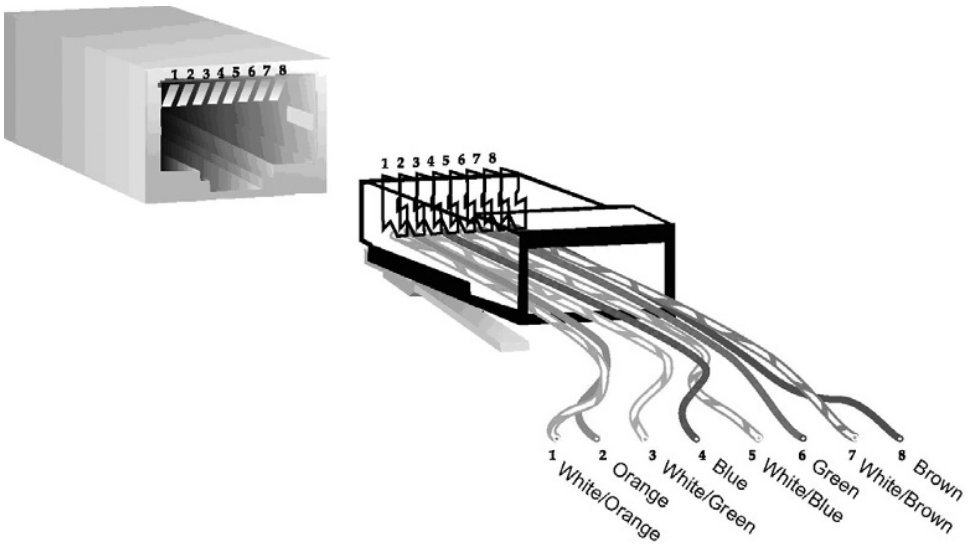


FIGURE 5.1

Wiring an RJ-45 according to the Livewire-preferred TIA/EIA T568B standard.

Table 5.2 T568A and T568B Pin/Pair Assignments from TIA/EIA-568-B Standard

T568A		
Pin	**Function**	**Color**
1	Transmit +	White/green
2	Transmit −	Green
3	Receive +	White/orange
4	N/C	Blue
5	N/C	White/blue
6	Receive −	Orange
7	N/C	White/brown
8	N/C	Brown
Shield	Protective ground	

(*continued*)

Table 5.2 T568A and T568B Pin/Pair Assignments from TIA/EIA-568-B Standard—cont'd

T568B (Preferred for Livewire AoIP)		
Pin	**Function**	**Color**
1	Transmit +	White/orange
2	Transmit −	Orange
3	Receive +	White/green
4	N/C	Blue
5	N/C	White/blue
6	Receive −	Green
7	N/C	White/brown
8	N/C	Brown
Shield	Protective ground	

N/C = No connection

CABLE OF BABEL Something to watch out for: The old telephone USOC wiring code for eight-pin connectors has the pairs in the wrong place, with the wiring in simple one-pair-after-the-other sequence. You'll have a split pair if you mix this sequence with either TIA or EIA formats, and a lot of crosstalk and interference problems will result. You need to be sure that the pairs correspond to Ethernet's requirements.

Why does Ethernet have such a strange wiring sequence, though? Because the center two pins (pins 4 and 5 on the eight-pin format) are where telephone voice circuits are traditionally wired. The designers of the standard originally thought that some people would want to use a single cable for voice and data, so they kept Ethernet clear of the telephone pins. There is also this benefit: If a user plugs a PC's network connection into the phone jack, the network card doesn't get blasted by ringing voltage.

By the way, even though there are two unused pairs in the standard Ethernet four-pair cable, you should not share the cable with any other service, since 100BASE-TX was not designed to withstand additional signals in the cable.

Note also that T568A is sometimes called the "ISDN" (or simply "EIA") standard and T568B is sometimes referred to as the "AT&T" specification.

Finally, on this topic, something really nutty: The overall cabling specifications standard and document from TIA/EIA was called the TIA/EIA-568-A Commercial Building Telecommunications Standard. Within this were the T568A and T568B pin-out standards. Note the dashes and lack of them. Then there is the more recent TIA/EIA-568-B overall standard, which has the same two pin-out standards within. Couldn't these guys have been a bit less confusing?

1000BASE-T Gigabit Copper

1000BASE-T works with Cat 5e or Cat 6 in the same configuration as for 100BASE-TX, but using all four pairs, as shown in Table 5.3. Either the T568A or T568B wiring sequence will work, but all four pairs have to be wired through and working.

There are no separate send and receive pairs for 1000BASE-T. Each pair both sends and receives with a hybrid splitter at the ends to separate the two signal directions, effectively creating four signal paths in each direction.

The signaling rate for 1000BASE-T is the same as for 100BASE-TX, which is why it can be run over the same cable. Nevertheless, 1000BASE-T is more sensitive, owing to the nature of the signal splitter and having twice the number of signals in a four-pair cable. That's why Cat 5e or Cat 6 is recommended. And high-quality factory-made patch cables are pretty much essential for reliable operation.

100BASE-TX Crossover Cable

Sometimes you may want to connect two Livewire nodes directly together without a switch, such as for testing or when you want to make a snake. Or you might want to connect a node directly to a PC for initial configuration, or to be used as a sound card. In this case, the transmit of one device must be connected to the receive of the other. For this, you'll need a *crossover cable*, which is wired as shown in Table 5.4. These are available off-the-shelf.

Most Ethernet switches sense the need for a crossover function and configure their ports automatically to perform the crossover adaptation internally when needed. So for connecting to switches, you probably will not have to use a crossover cable.

1000BASE-T Crossover Cable

You shouldn't ever need a 1000BASE-T crossover cable, since gigabit Ethernet switches and network cards almost always handle this case automatically. Nevertheless, a universal crossover cable (as shown in Table 5.5) can be made or purchased that works for both 100BASE-TX and 1000BASE-T.

Table 5.3 1000Base-T Signal Designations, Pin-out, and Wiring Format (Using T568B)

Pin	Function	Color
1	BI_DA+	White/orange
2	BI_DA−	Orange
3	BI_DB+	White/green
4	BI_DC+	Blue
5	BI_DC−	White/blue
6	BI_DB−	Green
7	BL_DD+	White/brown
8	BL_DD−	Brown

Table 5.4 100BASE-TX Crossover Cable Wiring Scheme		
Pin	**Color**	**Pin**
1	White/green	3
2	Green	6
3	White/orange	1
4	Blue	N/C
5	White/blue	N/C
6	Orange	2
7	White/brown	N/C
8	Brown	N/C

N/C = No connection

Table 5.5 Universal Crossover Cable Wiring Scheme		
Pin	**Color**	**Pin**
1	White/green	3
2	Green	6
3	White/orange	1
4	Blue	7
5	White/blue	8
6	Orange	2
7	White/brown	4
8	Brown	5

5.1.7 Installing RJ-45s

It's possible to build a sophisticated multistudio facility without ever wiring a single RJ-45 plug yourself. You would use modular patch fields or jacks at each end of the long "horizontal" cable with punch-down 110-style connections. Then factory-made patch cords would be used to get from the switch or Livewire node to the patch jack.

Even in this case, though, you could find yourself installing plugs at some point, so here is some advice:

- If you are making an Ethernet patch cord, use stranded conductor cable. Solid conductors are likely to break after a period of usage (from being plugged and unplugged). Note that this applies only to patch cords: Solid cable is best for backbone wiring because it has less loss.

- Be sure you are using plugs designed for the cable type you are using. There are different RJ-45 connectors made for solid and stranded wires.
- Plugs from different manufacturers may have slightly different forms. Be sure your crimp tool fits correctly. In particular, the crimper made by AMP will only work with AMP plugs. Buy a high-quality crimping tool to help prevent problems.
- The outer jacket should be cut back to about 12 mm (0.5 inch) from the wire tips. Check to be sure there are no nicks in the wires' insulation where you cut the jacket. (An appropriate tool can be purchased to permit you to do so rapidly without fear of damaging the inner insulation.)
- Slide all of the conductors all the way into the connector so that they come to a stop at the inside front of the connector shell. Check by looking through the connector front that all the wires are in correct position.
- After crimping, check that the cable strain relief block is properly clamping the outer cable jacket.

When checking the cable either with a tester or a real device, wiggle the cable around near the plug to be sure that the connector still works reliably when stressed.

You'll probably need a couple of tries to get it right the first time, but after some experience, installing RJ-45s will start getting pretty easy. Some connectors include a small carrier that the wires can be fed into first, and then slid into the connector itself. These are recommended because they speed installation and improve alignment accuracy.

5.1.8 Special Care for AoIP Wiring

"Normal" data over Ethernet are usually carried in TCP/IP. As you know from Chapter 2, TCP has a retransmission mechanism that detects errors and corrects them by requesting and obtaining replacement packets when one has been received with a defect. This mechanism can't be used for AoIP. That means it's possible that a network could be apparently okay with computer data because TCP is masking underlying problems, yet the same network could exhibit errors with AoIP traffic. So while paying attention to cabling basics to ensure reliable transmission is always a good idea, it is thus even more important to do so for AoIP networks.

Of particular concern are the prevention of impedance reflections at cable termination points, and stability in the positioning of wires inside cables. Here are some specific recommendations:

- Use the minimum number of terminations and patches that will support your application.
- Make sure that all patch cords, connectors, and other accessories are rated at the same or higher category level as the network infrastructure cable you've installed. Generally, your best bet is to buy premade patch cables, to both save money and time, as well as assure reliability.
- Keep a wire-pair's twist intact as close to any termination point as possible. For Cat 5 cable, pair-twisting should continue to within 1.3 cm (0.5 inch) of termination.

- Maintain the required minimum bending radius. For a four-pair, 0.5-cm (0.2-inch) diameter cable, the minimum bend radius is four times the diameter, or about 2 cm (0.8 inch).
- Minimize jacket twisting and compression. Install cable ties loosely and use Velcro fasteners that leave a little space for the cable bundle to move around. Do not staple the cable to backboards. If you tightly compress the jacket, you will disturb the twists inside the cable and affect the relationship of one pair to another, which could cause crosstalk.
- Avoid stretching of cables so twists are not deformed. The official recommendation is to use less than 25 pounds of pulling pressure.
- Avoid putting any network cables in close proximity to power cables or any equipment that generates significant electromagnetic fields. The official NEC recommendation for Cat 5 UTP is a minimum 50 mm (2 inches) distance from <3 kVA power lines.[1] Proximity to fluorescent lighting fixtures, motors, and transformers should also be avoided.
- The pins on RJ-45 plugs are gold plated, but not all connectors are. For maximum reliability, use connectors with 50-micron gold plating.

5.1.9 Minimizing Pairs in a Cable

Back in the 10BASE-T days, it was usual to have phone-type 25-pair cables carrying data signals. But the standards for Cat 5 and higher call for individual cables for each connection due to the possibility of multiple-disturber, near-end crosstalk, where the many signals in a cable bundle can add up to create combined crosstalk at unacceptable levels.

5.1.10 Patch Panels

Patch panels come in versions for rack or wall mount, and with varying numbers of jacks. Cat 5/5e cables are punched down at the rear into 110-style insulation displacement connectors using a tool very similar to the one that is used with traditional "66 blocks" (Figure 5.2).

Cat 6 cables and their accessories need more care to maintain the twists as close as possible to termination points. With experience, assembling one of these becomes fairly simple and can be done in a minute or two. First the cable is properly stripped, the cable's individual wires are put into the appropriate slots on the terminator, and the wire ends are trimmed. Then the front part of the jack is pushed onto the terminator. Finally, the shell is placed over these pieces and pushed onto them, which draws the wires into the insulation-displacement forks and locks everything together. (See Figures 5.3–5.5.)

[1]The National Electrical Code (NFPA 70), Article 800.133 (2005 NEC) indicates the separation requirements. The separation distance depends on the kVA value and rises to around 10 inches for high-power feeder lines.

FIGURE 5.2

Tool for punching down wires into 110-style connectors.

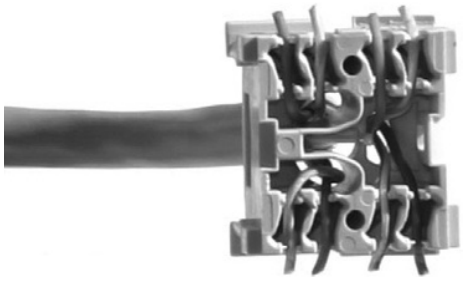


FIGURE 5.3

A Cat 6 jack terminator with wires in place.

5.1.11 Wall Jacks

Modular wall jacks are normally installed so that the pins are at the top of the connector cavity. This helps to protect the contacts from dust accumulation, and from water when baseboards are mopped. When the jack is oriented in this position (i.e., looking into the jack with the contact pins at the top), the pins are numbered 1 to 8 from left to right. Some jacks may not have all pin positions populated, but the numbering would still begin with the first position. For instance, the RJ-11 is a six-position, four-pin jack. Therefore, it has pins 2, 3, 4, and 5 present, and the pins for positions 1 and 6 are usually absent.

The procedure for wiring wall jacks is the same as for patch panels. 110-style IDC connectors terminate the cable. Then these wired-up "Keystone" RJ-45 jacks are pushed into a hole in the wall plate.

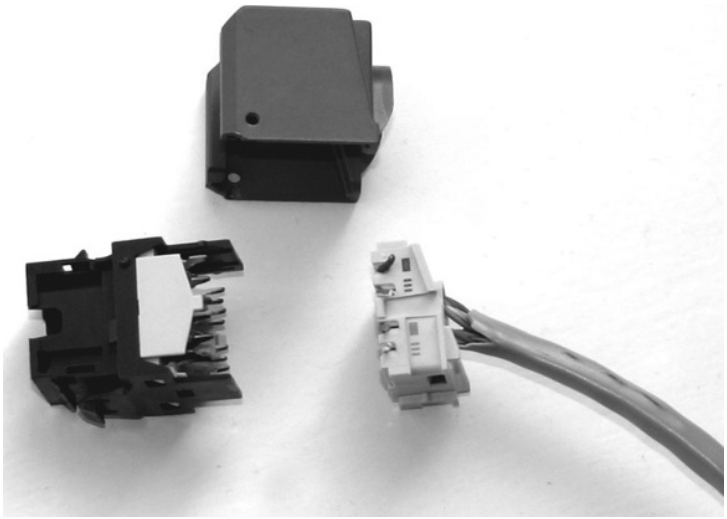


FIGURE 5.4

A complete, disassembled Cat 6 connector.

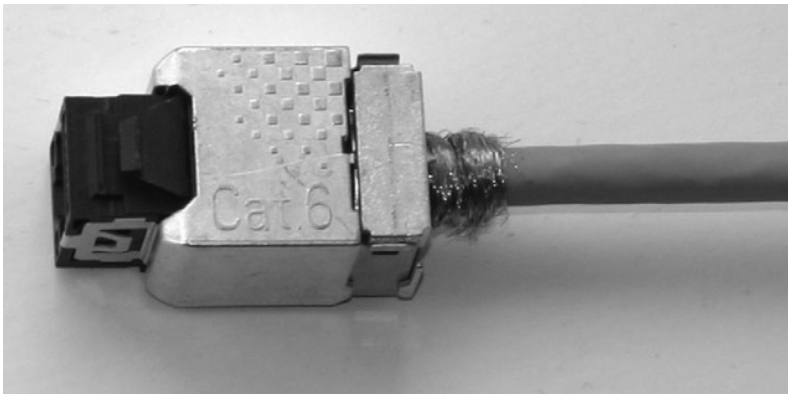


FIGURE 5.5

A high-end Cat 6 cable jack assembly ready for installation into either a rack-mounted patch field or a wall jack. This is a shielded version, so the shell is made from metal to maintain the shield all the way to the edge of the jack.

You should take care not to plug an RJ-11 plug into an RJ-45 jack. It will work to connect the pairs that are supported, but the plastic part on both sides of the plug will push up the outer pins on the jack, and they may not make good connection when the jack is again used for an RJ-45 plug.

5.1.12 Power over Ethernet

With the rise of VoIP running over Ethernet, powering phones over the same cable as the Ethernet data makes sense, duplicating the customary condition with traditional PBXs and their associated phones.

Power over Ethernet (PoE) was invented to satisfy this requirement. There's also no reason why it can't be employed for other things that could benefit from a centralized powering mechanism. One currently widespread application is the powering of distributed WiFi access points. In AoIP, we can use it to power studio accessory modules, phones, intercom panels, and the like. We could also imagine powering small audio interface nodes.

A variety of equipment can supply power to such PoE-enabled devices:

- Many Ethernet switch families include PoE as an integrated option. (These are referred to in PoE parlance as "endspan" sources.)
- Multiport rack-mount PoE injectors are available from a number of vendors.
- "Wall-wart" power supplies can be used to inject PoE at any point along an Ethernet link. They are usually installed locally to the powered device.
- Axia-integrated engines have some number of their Ethernet ports supplied with PoE.

It would not be good for links that don't need to be powered to have PoE power present, however. For that reason, PoE has been designed to use a bit of silicon intelligence at each port. A device that needs PoE must signal this to the power source, and the voltage will only be applied after the signaling has been successfully completed. PoE also offers protection from miswiring and overcurrent faults.

PoE is usually implemented following the specifications in IEEE 802.3af-2003. The powering source provides nominal 48 VDC over two of the four available pairs on Cat 5 cable with a maximum load power of 15.4 W. (Only about 12.95 W is available after cable losses.) 100BASE-TX uses only two of the four pairs in the cable, so two pairs are available for PoE.

Figure 5.6 shows the multiple methods of power distribution that PoE provides over 100BASE-TX links. In 1000BASE-T, only the phantom-power technique is used, since all four pairs are occupied for data transmission.

PoE+ is an emerging enhancement that extends power capacity to 24 W.

5.1.13 Fiber

Fiber optic links can extend the range of Ethernet. Because they are not subject to crosstalk and magnetic interference, they can solve problems that might arise with copper cables. External media converters can be simply plugged into Livewire nodes to convert 100BASE-TX copper connections to 100BASE-FX fiber paths (Figure 5.7).

Modern Ethernet switches often have the option to plug a media converter directly into a special socket so that fiber may easily be used to connect between

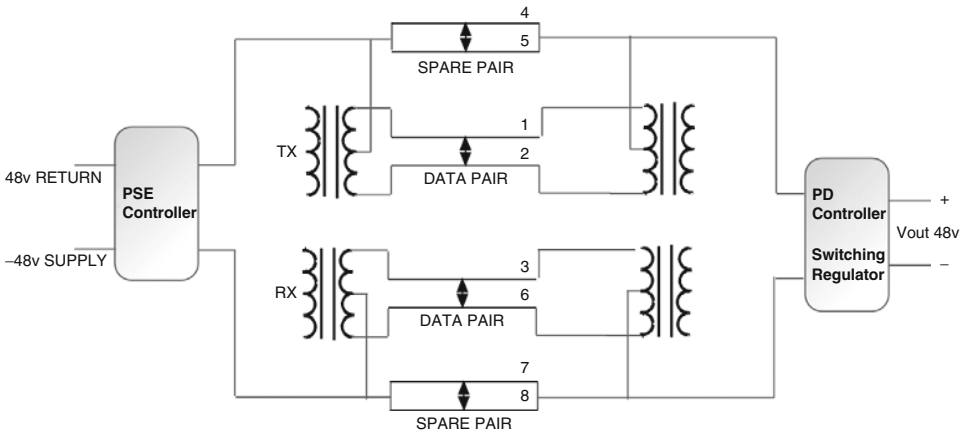


FIGURE 5.6

Schematic of PoE over a 100BASE-TX connection. In a full implementation, power is sourced directly over both spare pairs, as well as via a phantom scheme on data pairs. The powered device can take its feed from any or all pairs. (PSE = power sourcing equipment; PD = powered devices.)

FIGURE 5.7

This unit converts a 100BASE-TX link to a 100-Mbps ST multimode fiber link, for up to 2-km range. Similar units supporting single-mode fiber offer up to 75-km paths.

switches. You can use these ports to make high-capacity backbone links without any external boxes.

The first of these media-flexible interfaces in the 1000BASE-T domain is GBIC (Gigabit Interface Converter), a standard, hot-swappable electrical interface that allows a gigabit Ethernet port to support copper or multiple fiber-mode paths.

A more recent variant, SFP (small form-factor pluggable, also known as "mini-GBIC"), offers similar functionality at a connector density similar to typical electrical-only Ethernet jack fields. (See Figures 5.8 and 5.9.)

FIGURE 5.8

This switch has four uplink ports for use with 1000BASE-T SFP transceiver modules.

FIGURE 5.9

A modern media adapter in the "SFP/mini-GBIC" size—about the same in width and height as a standard RJ-45 jack. With switches that have sockets for them, these can be used to provide on-board optical interfaces in a variety of different flavors, such as 1000BASE-SX or 1000BASE-LX.

LESS IS MORE You'd probably expect something with "multi" in its name to have more capability than the same thing designated "single." But this is not the case with fiber optics: single-mode cables are better and more expensive than multimode cables. These names refer to how light is contained within the fiber, based on the principle of *total internal reflection*, which is how fiber optics (and the rainbow, among other optical phenomena) works. Single-mode fiber-optic strands are smaller, and they more carefully control the light passing through them so it doesn't bounce around as much inside, making them more efficient and permitting longer ranges.

5.1.14 Beyond Cable: Ethernet Radio Links

There are Ethernet radios that offer a lot of bandwidth at surprisingly low cost. Not all units are capable of achieving true Ethernet performance in terms of error rates, however, so some caution is in order. Most of these operate in the unlicensed ISM bands, but with modern spread-spectrum technology and elevated directional antennas (Figure 5.10), interference doesn't look to be much of a problem. Bitrates range to 150 Mbps and distance to 25 miles depending on power level, antenna, and terrain. Licensed radios following the IEEE 802.16 (WiMAX) standard are an option in this category.

There are a number of known successful AoIP installations using these Ethernet IP radios. Application notes and details of these examples are at *www.AxiaAudio.com/stl/*.

For studio-to-transmitter links, remote pickups, and studio-to-studio applications, these radios offer multiple channels of uncompressed, two-way audio transmission, and the ability to multiplex VoIP telephone, remote control, and general data. When audio and general data are mixed, an Ethernet switch provides the prioritization function, assuring that AoIP reliably receives the bandwidth it needs.

FIGURE 5.10

An Ethernet radio antenna on a studio site rooftop.

5.1.15 Analog and AES3 Audio Cabling

Many Livewire products use RJ-45s for balanced line-level analog and AES connections as well as for Ethernet. There are plenty of other connectors already being used for analog audio these days, so you might wonder why this choice was made. The reasons were numerous: cost, density, compatibility, and convenience. RJ-45 sockets and plugs are a lot cheaper than other choices—both for manufacturers of products and for installations at users' facilities. Density is an important advantage: You can get only a few XLRs across the rear panel of a 1U rack box, and you need two of them for each analog stereo connection. The same number of connections provided by a 1U 8 × 8 interface node (using RJ-45s) would require 2U to have the same channel capacity with XLRs for analog connections. (A single RJ-45 can handle balanced analog stereo, and dozens of them fit on a 1U box panel.) With regard to convenience, XLRs and DBs need to be soldered, shells assembled, etc., while RJs are either ready-made or quickly assembled on-site.

Further, because you have a single cable and connector type for everything in your facility, cable and connector inventory is greatly reduced. And as your studios evolve, the same cable that was once used for analog can be used for AES, AoIP packet audio, general Ethernet data, or whatever else might come along.

Using Cat 5 "digital cable" for analog and AES3 signals may seem odd at first, but it does make sense from a technical perspective. The low capacitance and tight twisting requirements necessary for high-speed networks are good for analog and AES audio as well.

The question always comes up: Do these connections need to be shielded? For AES, the answer is clear: UTP cable that works for Ethernet will be fine for the lower-rate AES data. Analog is a trickier question. Shields have almost no effect on the low-frequency magnetic coupling that causes crosstalk from adjacent audio pairs and hum induction from nearby AC wiring. Tight, accurate pair twisting combined with good common mode rejection in the analog interface node inputs is the solution to this potential problem. Tests at (cable manufacturer) Belden with a run-of-the-mill stranded Cat 5 cable concluded that below 20 kHz, average crosstalk between pairs within the cable was around −100 dB with cables of 100 meters, or 328 feet, in length. Belden's higher-end Mediatwist cable had less than −110-dB crosstalk.[2] Foil shields don't begin to have effectiveness until well into the megahertz. So the decision to shield analog connections comes down to your expectation for RF interference and your position on the take-a-chance versus careful-and-conservative continuum. Is the facility in a high RF field? Will staff be using mobile phones very near to the cables? How long are the runs? If you decide that shielding is necessary, shielded versions of Cat 5 or 6 cable ("STP") are available, at a somewhat higher price point.

RJ-45s and Cat 5 cables have been successfully used for nearly two decades, with the many installations based on the Radio Systems StudioHub+ product family serving as real-world demonstrations.

[2]Reported by Steve Lampen, *Radio World*, available at *www.radioworld.com/article/4254*.

Livewire products use the StudioHub+ pin-outs. Since they follow this de facto standard, StudioHub+ wiring components can be used for the analog and AES3 parts of Livewire installations. (See Table 5.6.)

Radio Systems offers an extensive line of single "dongle" and multipair harness cables prewired to connect to a variety of studio gear (Figure 5.11). They also make balanced-to-unbalanced, analog-to-digital, AES-to-S/PDIF, and AES-to-TOSLINK adapters, headphone and distribution amps, splitters, summing pads, etc.

Table 5.6 Livewire/StudioHub+ Standard for Analog and AES3 Wiring on RJ-45s

Pin	Function	Color (T568B Standard)
Shield	Protective Ground	White/slate and slate/white[a]
1	Analog L+/AES+	White/orange
2	Analog L−/AES−	Orange
3	Analog R+	White/green
4	N/C (GND[b])	Blue
5	N/C	White/blue
6	Analog R−	Green
7	N/C (−15VDC[b])	White/brown
8	N/C (+15VDC[b])	Brown

[a] *Optional.*
[b] *Used to power in-line devices such as balanced-to-unbalanced converters.*

FIGURE 5.11

Dongles like these RJ-to-XLR adapters from Radio Systems provide interface from RJ-45 network-style cables to the usual pro-audio connections.

5.1.16 Microphone Connections

Current Livewire gear uses XLRs for microphone inputs. RJs would probably not be sufficiently reliable for such low signal levels. And it has to be said that a microphone cable with an RJ hanging off one end would be pretty weird.

Nevertheless, there are now plenty of microphones aimed at home recordists that have USB interfaces for attachment to PCs running audio applications. AoIP with Ethernet 100BASE-TX would be a much better choice with its 100-meter-length capability and transformer isolation. Perhaps we will eventually see some AoIP microphones?

5.1.17 Unbalanced Connections

Unbalanced analog connections should be avoided for the usual and well-known reasons. If you need to interface Livewire equipment with unbalanced gear, you can take your chances (with short runs) and just use one side in the usual "cheat" way, making sure to always set the same pin "high" throughout. Of course, also remember the lower reference level typically used on unbalanced audio equipment (−10 dBu).

A more careful approach would use a balanced-to-unbalanced amplifier or transformer (such as the device shown in Figure 5.12), located as close as practical to the unbalanced equipment.

Radio Systems' StudioHub+ Matchjack series offers plug-and-play compatibility between the Livewire and unbalanced worlds. These can be powered directly or remotely over spare pairs (see Table 5.6) using a Radio Systems in-line power supply designed for the purpose.

FIGURE 5.12

A Radio Systems' Matchjack interfaces from StudioHub+-format analog audio on RJ-45s to unbalanced-device I/O connections.

5.2 ETHERNET SWITCHES AND IP ROUTERS FOR LIVEWIRE

In Chapter 3 you saw some general AoIP considerations concerning switching and routing. This section drills down a little further in this area on Livewire AoIP per se.

Livewire packets include both the Ethernet and IP headers. This means that Livewire streams may be either switched at layer 2 or routed at layer 3. For most Livewire installations, a managed layer 2 switch or a layer 3 switch that includes the required IGMP querier and snooping functions is recommended. IP routers are able to do layer 2 switching as a subset of their more advanced capabilities, so they may also be used. Switches and routers range from very simple to the exceedingly elaborate. Very large installations may benefit from the advanced monitoring, backup, and other features the IP routers offer. And as the cost of hardware continues along the Moore's Law plunge, IP routers may come to compete in cost with Ethernet switches.

Livewire AoIP Ethernet switch requirements are as follows:

- *Sufficient backplane bandwidth:* It is required to be fully nonblocking to handle all ports at full capacity.
- *Sufficient frame-forwarding rate:* AoIP uses small packets at a fast rate.
- *Correct handling of IEEE 802.1p/Q frame prioritization*: Livewire audio frames must be given priority without too much delay or jitter. The IEEE standard specifies eight levels of priority, but few switches support all the levels. Many support only two or four levels, lumping some of the incoming levels together. We recommend four levels as the minimum for an AoIP system that also needs to support general data traffic.
- *Support for multicast with IGMP control: There must be sufficient filter-entry capacity to cover the total number of audio streams you need*. This is important because when the filter capacity is exhausted, switches forward multicast packets to all ports, subscribed or not, causing a big problem.
- *Support for both port-based and tagged-frame-based VLAN:* The latter is the IEEE 802.1Q standard, which allows the switch to determine priority on a frame-by-frame basis. Port-based VLAN can also be useful: It lets you "hardwire" a particular port for a single VLAN, for example, to be 100 percent sure an office PC can't get onto the AoIP VLAN.
- *Special requirement for VLAN support:* If you want to use a separate VLAN for AoIP, the switch needs to have an IGMP querier working on that VLAN, which also means that you can assign an individual IP number to each VLAN. This is a rare capability among Ethernet switches (it's found on most IP routers, however), and its absence disqualifies many switches from consideration when the feature is required by your installation.
- *Switch management:* Any switch that supports IGMP and multicast will be a "managed switch," which provides at least a web interface for configuration and basic remote monitoring. Fancier switches offer better management, which can sometimes be useful for AoIP.

A regularly updated list of switches that have been tested and approved for use in Livewire installations is provided at *http://www.AxiaAudio.com/switches/*.

5.2.1 Switch Configuration

Because AoIP needs multicast, a switch's out-of-the-box configuration might need to be modified. While this can sometimes be done via the web interface, the command line is often the best way to have the fine-grain control that you probably will need.

Following is an example of a switch configuration for Livewire, in this case for the Cisco Catalyst 3560.

Global QoS Configuration

Enter the following commands:

```
config t
mls qos srr-queue input buffers 50 50
mls qos srr-queue input cos-map queue 1 threshold 1 5
mls qos srr-queue input cos-map queue 2 threshold 1 6 7
mls qos srr-queue output cos-map queue 1 threshold 1 6 7
mls qos srr-queue output cos-map queue 3 threshold 1 5
mls qos
end
```

Configuring Fast and Gigabit Ethernet Ports That Connect to Livewire Devices

Enter the following commands:

```
config t
interface fa0/x
switchport mode access
switchport nonegotiate
switchport voice vlan dot1p
srr-queue bandwidth share 30 20 25 25
srr-queue bandwidth shape 0 0 0 0
priority-queue out
spanning-tree portfast
mls qos trust cos
end
```

- Where "x" is the port # to configure
- Use the RANGE command for multiple port configurations.

Syntax:

```
interface range fa0/x - yy
```

Configuring Ports Connecting to Other CISCO Catalyst Switches

Enter the following commands:

```
config t
interface GigabitEthernet 0/x
switchport mode trunk
srr-queue bandwidth share 30 20 25 25
srr-queue bandwidth shape 0 0 0 0
priority-queue out
mls qos trust cos
end
```

- Where "x" is the port # to configure
- Use the RANGE command for multiple port configurations.

Syntax:

```
interface range fa0/x - yy
```

On a stackable switch use this syntax:

```
fa1/0/x or GigabitEthernet1/0/x
```

Check Your Configuration

To check your switch's configuration, you can enter: `show running-config`. You should see that your entries have been properly accepted by the switch.

Note that there are two different software images that run on Cisco switches: the standard multilayer software image (SMI) or the enhanced multilayer software image (EMI). The latter includes advanced features such as policy-based routing (PBR). EMI comes at a significant cost premium and is not usually required for AoIP. We mention the point because switch configuration is different for each. The example above is for the SMI-version software.

5.3 AoIP APPLICATIONS AND ARCHITECTURES

AoIP installations can range from two-node "snakes," to simple one-studio setups, to large and complex systems with dozens of switches and routers, hundreds of ports, and thousands of audio channels. Fortunately, Ethernet and IP were designed to scale, and therefore so can an AoIP installation. In this section, we'll explore examples that span the size and sophistication spectrum.

5.3.1 AoIP "Snake"

The simplest possible configuration for AoIP is two nodes connected with a crossover cable. If the application is fixed, configuration can be done for each node with a direct connection to a PC before actual operation. Adding a small Ethernet switch would permit online configuration and monitoring. It would also let you add more nodes, perhaps mixing and matching analog and AES3 interfaces. (Indeed, this AoIP approach is about the only way to realize an AES-over-IP link.) (See Figure 5.13.)

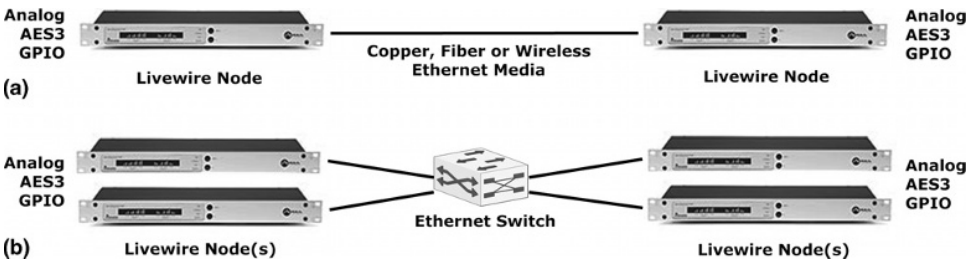


FIGURE 5.13

Livewire can be used as a "snake" in various configurations. (a) A fixed, point-to-point configuration simply connects two Livewire nodes across any Ethernet medium in crossover mode. (b) A dynamically configurable (and potentially point-to-multipoint) arrangement is achieved with the addition of an Ethernet switch. In either case, a mix of analog audio, AES3, and GPIO signals can be transported between or among locations.

Because of the tightly controlled delay characteristics of the Livewire technology, multiple channels, even from different nodes, are synchronized to within a fraction of an audio sample. All of Ethernet's media are open to you: copper, fiber, and wireless.

5.3.2 Networkable PC Sound Card

AoIP is an alternative to internal PC sound cards and USB-based audio interface boxes. In contrast to USB, Ethernet-linked audio interfaces can be located at a distance from the PC. The PC could be in the control room, and the audio I/O in a studio.

In the simplest case, a crossover cable would connect the PC with the interface node. The Axia Livewire driver supports multiple Ethernet cards, so one that is independent from the facility's main network interface could be used just to serve audio nodes.

As always with AoIP, you can add an Ethernet switch to allow multiple interface nodes. The full-up Livewire PC driver handles 24 stereo inputs and an equal number of outputs, so you have plenty of channels available. (See Figure 5.14.)

5.3.3 Studio-to-Transmitter Link

The studio-to-transmitter link (STL) is pretty much the snake we've seen above, but with an Ethernet radio link as the transport media (Figure 5.15). Livewire nodes are used at each end to interface analog and AES3 audio, but you will probably have plenty of extra bandwidth that can be used for other things. HD Radio IP-based transport streams could traverse the same link, as could control/telemetry and VoIP phone extensions. And, of course, unlike traditional STLs, the link is inherently bidirectional.

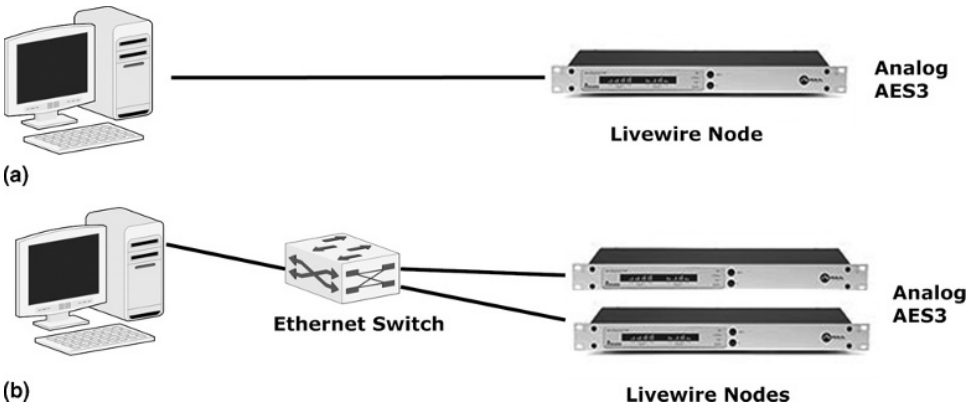


FIGURE 5.14

A Livewire node can act as a remote sound card for a PC, with the audio I/O terminations located anywhere with access to the network. (a) A single node connected to a PC via a crossover cable; (b) a multinode arrangement using a small switch.

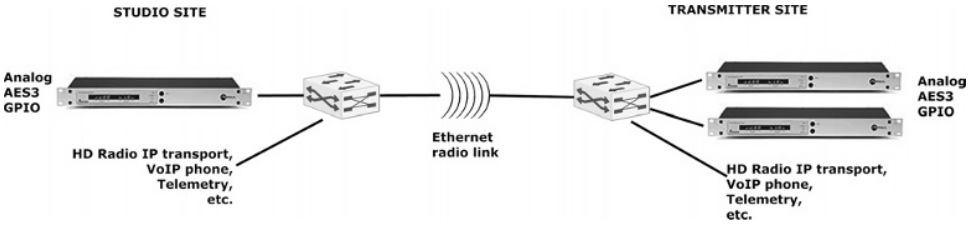


FIGURE 5.15

Livewire nodes can be used to establish a two-way STL (STL + TSL) over an Ethernet radio link.

When there is mixed AoIP and other traffic, the Ethernet switch at each end needs to properly support priority so that audio gets first call on the radio link bandwidth.

Livewire nodes have a "low-rate clock" setting that reduces the bandwidth needed for the clock synchronization data. This configuration option should normally be chosen to reduce bandwidth consumption on the wireless link. (The trade-off is a longer PLL synchronization time when nodes are powered-up or need to change over clock mastership, but this is not an issue in this dedicated, point-to-point application.)

5.3.4 Simple Radio Studio

A single, standalone radio studio can use an integrated Livewire engine that has an internal Ethernet switch, along with local audio and GPIO nodes (Figure 5.16).

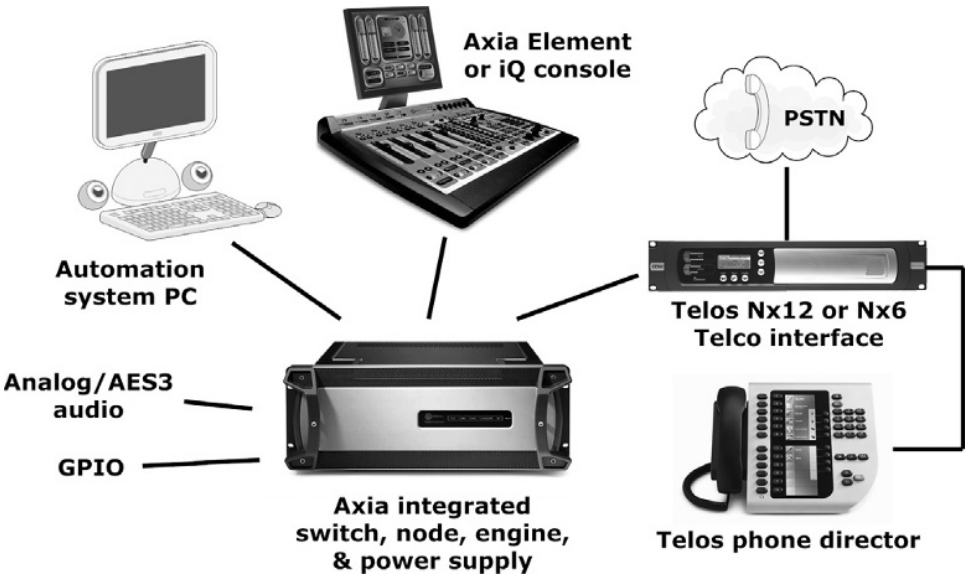


FIGURE 5.16

A basic, single-radio studio can use an integrated Livewire core that includes a mix engine, audio interface nodes, and a small switch. PC audio sources and telephone interfaces connect via IP rather than in the audio domain.

The main advantage from the AoIP approach, even in this very simple case, is that PC-based audio sources connect directly via Ethernet, rather than by sound cards. The telephone interface also benefits, having a single connection for all audio channels and control, which greatly simplifies the usual tangle of cables for the multiple send and mix-minus channels. The networked control means that the console can also have a richer interface to the phone system.

This studio could be easily linked to others by adding a central Ethernet switch and running uplinks to it. Or a daisy-chain approach could connect a few studios without the need for a central switch. Remote audio sources could be interfaced via nodes located outside of the studio and backhauled via Ethernet connection, and that audio can be shared among multiple studios.

5.3.5 Full-Fledged Radio Facility

AoIP is even more advantageous in larger radio facilities. Figure 5.17 shows a typical high-end radio broadcast studio and central resource system. In the studio, it uses the integrated PowerStation core, which incorporates an Ethernet switch, mix engine, audio and GPIO nodes, and power supply. There is no external "edge" switch required in the studio. Analog, AES3, and GPIO connect directly. The automation PC is equipped with the Livewire driver and connects via Ethernet to the

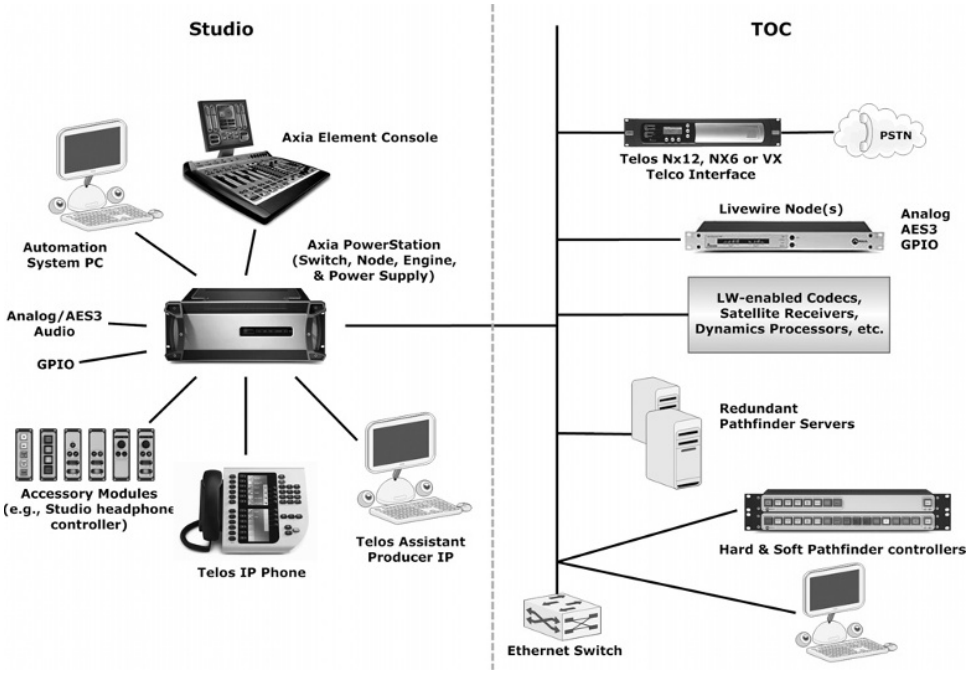


FIGURE 5.17

A typical Livewire radio facility installation, showing one of several studios (at left) and the TOC (at right).

PowerStation. The PowerStation also serves as the Ethernet switch for the VoIP phone system.

There is a central ("core") Ethernet switch in the facility's technical operations center (TOC, or "master control," or simply "rack room") that links additional studios and other devices into the system. A pair of servers provides clustered redundant hosts for the Pathfinder routing controller package, which can be controlled either by hardware panels or PC-hosted client software.

The telephone interface in Figure 5.17 is shown with only a PSTN connection, but this could be the VX system with an SIP connection to a VoIP PBX that serves the entire facility (see Section 6.2). Control and Livewire audio flow over the network to any studio that needs access to the telephone network. A talk-show producer/call-screener desk would also be in the picture for stations requiring it, with a PC connected to the network running another copy of Telos Assistant Producer IP or a similar third-party software application such as those from Broadcast Bionics or NeoSoft.

5.3.6 Livewire "Classic" Radio Studio Setup

Before Axia introduced the PowerStation and iQ integrated engines (in 2008), Livewire AoIP studios were constructed in a more modular fashion (Figure 5.18). Each console had its own power supply/CPU/GPIO unit supported by an independent mixing/processing engine. Each studio usually had its own edge Ethernet switch as well. While a single switch could serve multiple studios, the switch-per-studio design made for "islands of reliability." When there were multiple AoIP-equipped studios in the facility, a core switch in a central rack area connected to each local switch.

The studio's mix engine connects with a 1000BASE-T copper link to one of the two 1000BASE-T SFP ports on the switch with a standard copper transceiver module.

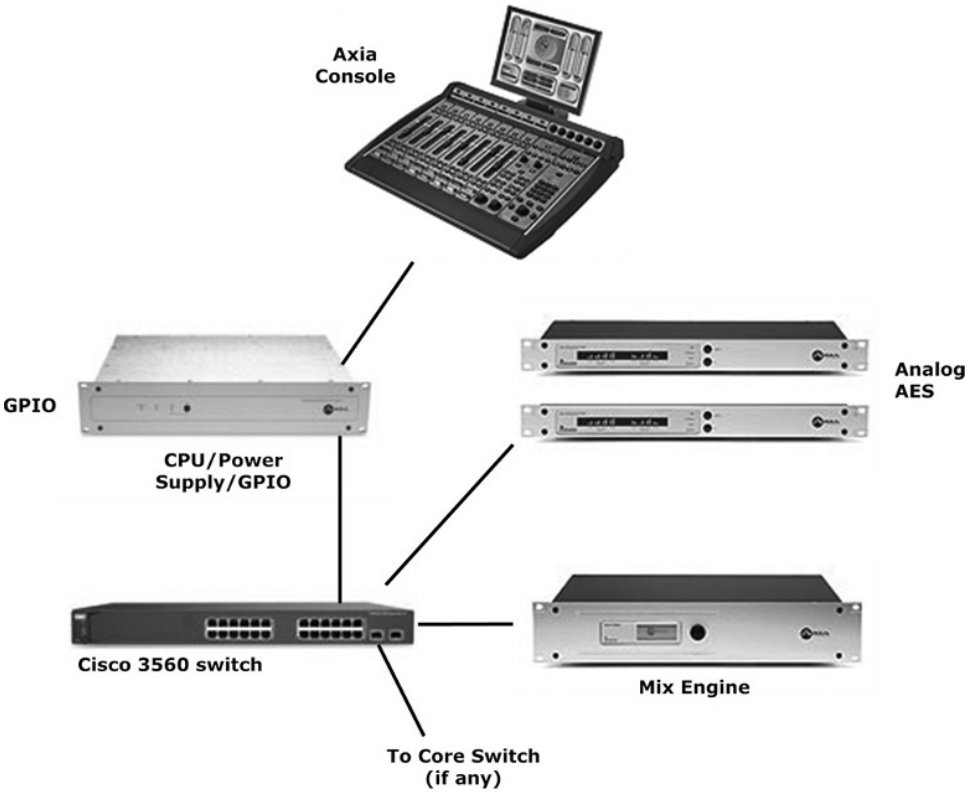


FIGURE 5.18

Prior to the development of a more integrated solution (PowerStation), Livewire radio studio systems utilized this more modular approach.

As noted before, any PCs used in the studio for audio playout and file storage connect directly to the network via the edge switch using Livewire audio driver software (rather than via sound cards and audio connections).

Peripherals such as codecs, telephone systems, and satellite receivers may be connected into the network wherever it is convenient, using audio interface nodes when only analog or AES I/O is available, or directly to the switch when a native AoIP connection is offered.

This modular component approach remains available, and might still make sense for some installations because it lets you have just the mix of interfaces to serve your specific requirements. It also permits you to choose whatever size Ethernet switch you need, providing the availability of a lot of ports, should they be required.

5.3.7 Audio Router

You can make an audio router system with AoIP that will cover almost any capacity requirement, ranging from a one-box 8 × 8 to a feature-laden, facility-wide system with thousands of audio channels (Figure 5.19).

Any number of interface nodes can be used to provide the desired number of audio channels. Managed Ethernet switches range from 1U rack boxes with 24 ports to redundant frames that host modules with hundreds of ports. (The latter usually have IP routing capabilities as well as layer 2 switching.) Multiple switches

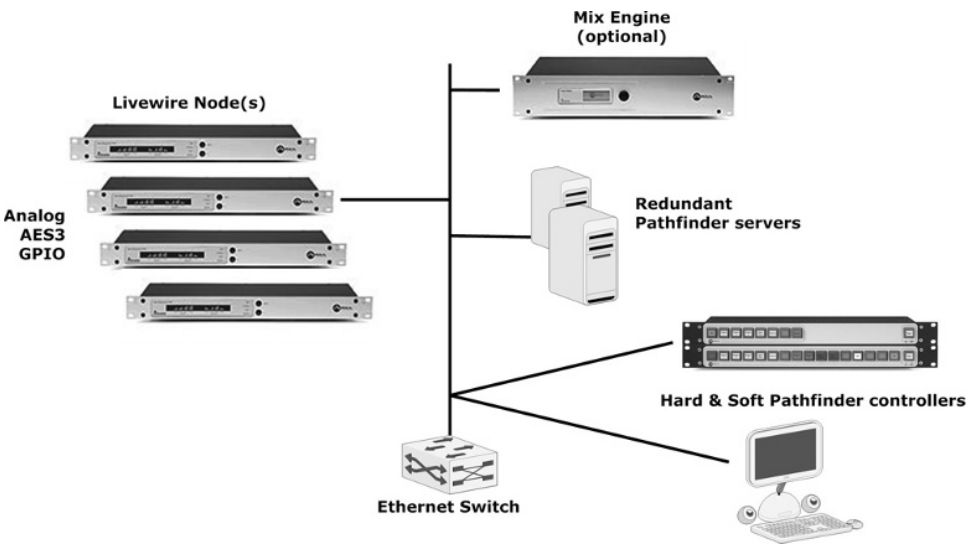


FIGURE 5.19

Livewire can be used as an analog audio, AES3, and/or control-signal router only.

can be linked in a variety of ways to create a "switching cloud" with any number of ports.

Livewire's Pathfinder software application is the glue that lets you control the distributed system as if it were a traditional frame-based audio router. Recall from Chapter 4 that Pathfinder enables a full spectrum of control options. Both software clients and hardware panels are connected into the network via Ethernet. Event triggers can be built using a "stacking events" editor that can respond to a variety of stimuli, including combinations of button presses, timed actions, triggers from GPIOs, networked messages, and silence detection.

There is full support for GPIO, both in the traditional hardware-closure type as well as via network messaging to PCs and other devices that have an Ethernet connection. GPIOs also may be configured to follow audio routing changes.

A mix engine could be included in such a system to add a basic mixing or crossfading capability and stereo channel left/right/sum input selection. Control for these processes also would be delivered over the network and managed by the Pathfinder application.

5.3.8 A 50+ Studio Facility with Redundancy

A good example of a very large AoIP installation is the new plant at Radio Free Europe (RFE) in Prague, one of the largest and most sophisticated audio broadcast facilities in the world. It uses an architecture with a redundant hierarchical structure (Figure 5.20).

The core switch in this facility is actually a high-end IP router, the Cisco Catalyst 6509–enhanced nine-slot chassis with online automatic backup and redundant power supplies. The router's interface modules include 24-port GigE copper and SFP ports hosting GLC-SX-MM 1000BASE-SX fiber connections. Edge switches are Cisco 2960G-24TC-L located near studio clusters. Each edge switch has redundant 1000BASE-SX connections to the core.

RFE makes extensive use of Pathfinder's capabilities to change routes both for audio feeds into studios and to configure program feeds out to the many broadcast transmission sites served by the facility.

5.4 MORE ON ARCHITECTURES

5.4.1 Daisy-Chaining

Core switches are not the only way to link studios and share audio among them. Rather than having a core switch, individual studios may be connected in a daisy-chain fashion.

A gigabit link between switches allows hundreds of audio channels to flow from one group to another. You could have a "circular backbone" with redundant spanning tree links between the switches.

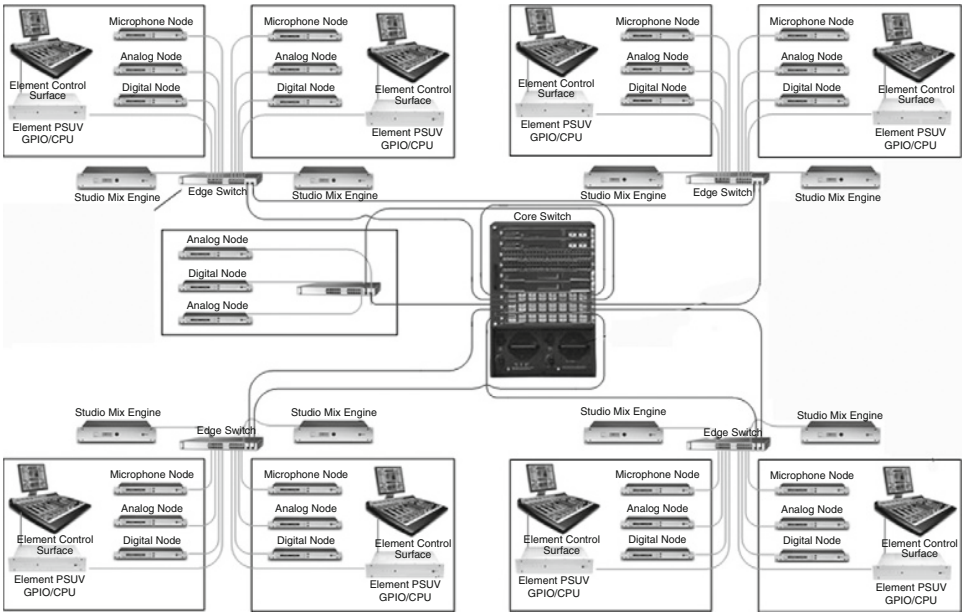


FIGURE 5.20

Block diagram excerpt from RFE's facility in Prague, showing AoIP flow in four of the 50+ studios and the TOC core switch. The core switch connects to edge switches via redundant gigabit fiber links. Edge switches connect to studio equipment via 100BASE-TX or 1000BASE-T.

Peripherals that are used in common, such as codecs, could be plugged to any of the studio switches, or there could be a separate switch daisy-chained-in to pick up such feeds.

5.4.2 Redundancy

Ethernet switching has a built-in scheme for redundancy called *spanning tree*, standardized as IEEE 802.1D. A newer variant is called *fast spanning tree*. Switches that have activated their spanning tree feature can exchange information with each other about the topology of the overall network. You can have redundant backup links that are automatically activated in the event that a main link has failed. Depending on the switch and layout, it could take as little as a second or as much as a half-minute for a redundant link to be connected.

Link aggregation (sometimes called *port trunking*) is another method of achieving redundancy. With spanning tree, even if you have two links between two switches, only one of them will be active at a time. But it's often better to have both active simultaneously because you get twice the bandwidth during normal operation and instantaneous backup should one fail. This link aggregation is standardized in

IEEE 802.3ad. To use it, you usually have to explicitly enable it on your switch. Incidentally, this scheme is supported on some PC network interface cards (intended for servers), so it's not just for switch-to-switch links.

When an IP router is at the core, there are additional options for automatic redundancy that can respond faster than the layer 2 strategies. There is also manual swap-out as a reasonable option. Because RJ-45s are so easy to unplug and replug, and because switches and other Livewire components are much cheaper than traditional alternatives, you can have spare units on the shelf for fast substitution.

Finally, most Ethernet switches and IP routers offer a backup power supply option to address that aspect of redundancy.

5.4.3 Security

You will have 100 percent security if you keep the Livewire system completely isolated from any other network (either LAN or WAN). This is a simple and perfectly good strategy for a worry-free, basic installation.

But there are advantages to sharing with or linking to an office network. First, you can then configure and monitor the system from any connected PC in the facility, and audio can be monitored on any desktop with access. In this case, separate switches or VLANs can be used to provide isolation for the AoIP sector. An IP router passes only the correct packets from one to the other and thus provides a firewall function.

Connection to the Internet brings the advantage that you can monitor and configure your AoIP network from a remote site, but the trade-off is the increased risk from the public connection.

With Livewire nodes and other devices, Web and telnet access are password protected to provide some measure of security. But they do not use exotic secure-access techniques like SSL (secure sockets layer). That's because they should always be behind an effective firewall, which will be an essential component of any AoIP system that has a path to the outside world. A qualified network engineer should be consulted to advise on appropriate firewall and other protection strategies.

Some Axia components that are meant to bridge the AoIP Livewire network with the outside world feature two Ethernet jacks, one for the inside and another for the outside. The iPort MPEG gateway and Telos VX VoIP phone engine are examples of this approach to security isolation.

CHAPTER

VoIP Telephone Systems in the Studio Environment

6

Is it not a bit strange that many computer-laden, all-digital broadcast studios connect to the PSTN (Public Switched Telephone Network) using technology invented in the day of phones with hand-cranks and bulbous "ringers"? Where else in pro-audio do we mix two audio directions on a connection, forcing us to imperfectly pry the two apart in our interface gear? Where else do we use blasts of 100-AC volts for signaling? This becomes even stranger when you consider that the core of the telephone network is also digital, with sophisticated internal signaling systems and independent audio paths. Surely we can do better than this ancient bell-banging and analog audio mash-up stuff in our contemporary telephone interfaces.

That was the idea of ISDN, and it was a good one, as far as it went. The telephone network had begun to transition to digital in the 1960s and by the 1980s, the conversion of the switching and internal transmission was nearly 100 percent complete in many countries. The idea of domestic data communication was just getting underway. Remember bulletin boards and 1200-baud modems? The inventors of ISDN reasoned that if they could pass the full 64-kbps rate of the network to users, they'd be thrilled with the amazing speed! And, with two channels, you could simultaneously talk and look up recipes. YouTube had apparently not entered their imagination.

BACK TO THE FUTURE On the other hand, some phone company forward-thinkers *had* stumbled upon the idea of on-demand music. Steve remembers attending a seminar sponsored by Bellcore in 1987 that introduced a pre-ISDN digital service called Switched-56. Among the futuristic demonstrations was a music service accessed by voice prompts and dual-tone multifrequency (DTMF) key presses. A phone had been outfitted with a "high-fidelity" 7-kHz mono codec and a drive-in theater-class loudspeaker. Steve Jobs had nothing to fear.

While xDSL quickly eclipsed ISDN as the home datacomm technology due to its greatly increased bitrate, broadcasters continue to take advantage of ISDN for

Audio Over IP

doi: 10.1016/B978-0-240-81244-1.00006-7

high-quality remote pickups.[1] Getting access to the bits directly means that codecs other than the usual G.711[2] can be used to achieve much better fidelity. MPEG HE-AAC, for example, offers broadcast-grade stereo using the same channel and bitrate as the phone company uses to provide its old-fashioned mono voice service. While ISDN is still a worthy technology, delivering reliable transmission with low delay, it seems that telcos are phasing it out. While some telcos will still happily sell you ISDN, some have made official end-of-life announcements, and some others seem to react to an ISDN order inquiry as if you had phoned-up the ASPCA[3] to discuss dropping the family dog off at the taxidermist as part of your holiday preparations.

Actually, there are two types of ISDN service: the two-channel BRI (basic rate interface) and the 23-channel (30-channel in Europe) PRI (primary rate interface). The former is the one that seems to have lost its reason for being, and will be the first to go. ISDN-PRI could stay around much longer, as it is widely used to connect large-business PBXs, especially in Europe. But it seems this, too, will eventually be superseded, which brings us to the topic of this chapter. What will eventually replace both POTS (plain-old telephone service) analog lines and ISDN? Voiceover IP (VoIP), of course. Many telcos already offer it, and they are beginning to actively promote it. We'll explore this soon, but let's broaden our focus for now.

There are three distinct possibilities for VoIP in a radio studio or audio production facility:

- Using an IP PBX within a facility for general phone service.
- Using an IP-based studio telephone system for on-air calls.
- Using VoIP to connect to a telco network.

Each of these need to be considered independently, which we shall do shortly, but first let's set the table.

6.1 VoIP IN RADIO STATIONS

A couple of typical radio station scenarios include the following:

- A listener has entered a phone contest by calling on the studio number. He or she has a question that can only be answered by someone in the promotion department, so the DJ wants to transfer the call.

[1] xDSL service (as its name implies) is mostly used by telcos to provide Internet access. This means one end of the xDSL line always terminates at a telco central office (CO), where it is usually routed directly to the public Internet. It is also typically offered for this use with asymmetrical bandwidths, which provide higher speeds *from* the CO to the customer than in the opposite direction, as is generally required by customers for Internet access. Thus, xDSL is not easily applicable to the point-to-point, symmetrical, private applications broadcasters generally require for remote audio backhaul, and for which ISDN is well suited. Unfortunately, the overall demand for that kind of connection is so low that telcos are unlikely to continue offering such service on a broad basis.

[2] The original, universally deployed standard ITU-T codec used for voice-grade telephony, using 8-bit PCM audio sampled at 8 kHz, producing the common 64-kbps data stream mentioned above.

[3] American Society for the Prevention of Cruelty to Animals.

- A newsmaker returns a journalist's call. He or she phones into the station's main number and gets transferred to the newsroom, where the journalist records a comment from the newsmaker. The producer of a talk show hears that the newsmaker is on the line and wants to take the call in the on-air studio for a live chat.

One would think that these would be easy to accomplish, but if you have worked in a radio station over the years, you know that it is anything but. The problem starts with the station's business PBX, where transferring a call is a proposition that involves considerable mental gymnastics. The problem continues with having two telephone systems to contend with, the PBX and the on-air phone system, each with its own operating paradigm. And finally, you have the technical limitation of the interconnection between the systems, which is usually a POTS-line emulation. This means that the studio system has only one way to initiate a transfer—the ancient "switch-hook flash," waiting for a "stutter-tone," and dialing an extension number that needs to be looked up in a printed directory. Sheesh! Why don't we just hire an operator to plug cloth-covered cords into jacks?

Enter SIP (Session Initiation Protocol), which today allows multiple PBXs to coexist in a facility, with a rich signaling path between them. This is one of the benefits of IP-based systems that should have an immediately noticeable payoff in the studio environment. With SIP, it can be simple and routine to hand calls back and forth between the business and on-air systems.

Studio on-air systems are usually connected directly to dedicated telco lines, rather than being routed through the PBX. The main reason for doing this is that passing the audio through the PBX often causes audio degradations. Since the connection is usually analog, there is the conversion to/from digital in the PBX and on-air system. That's a shame, because for two decades, most PBXs have been digital and the on-air systems have been as well, so there is no need for the intermediate analog link. Digital links from studio systems to PBXs have been impossible because the PBX vendors have used differing and proprietary connections to their phones. If the PBX in turn connects to the telco via POTS, that is another unnecessary distortion-and-noise-causing conversion bump in the speech path.

As with most things IP, standardization means we now have a method of hooking things from different vendors together—and a way to overcome problems of the past.

Another scenario:

- An engineer wants to save the station some money and impress the manager. So the engineer negotiates with a telco to provide all the station's phone service on a single ISDN PRI, but then has to find a way to divide the channels between the office system and the studio system. Some new cards in the PBX might do it, but they cost thousands of dollars, making the manager a bit less than fully impressed with the engineer's cost-savings acumen.

Again, IP to the rescue. A gateway can be used with ISDN PRI on the telco side and SIP IP on the PBX side. Both the station's general PBX and the studio on-air

system talk SIP to the gateway. The channels ("lines") can be divided any way that is needed.

SIP is the secret to smooth, interoperable communication among disparate vendors' equipment. Meanwhile, IP avoids unnecessary analog-to-digital (A/D) or digital-to-analog (D/A) conversions and lets the audio pass in pure digital form. Studio systems can now pass audio to/from PBXs without any added distortion or noise.

6.2 SIP

SIP is fast rising to be the big-daddy buzz-acronym among telecom technology acolytes. SIP is how calls are set up over IP connections, so it *is* actually pretty important. Together with helpers like proxy servers and user agents, SIP permits all the familiar telephone-like operations: dialing a number, causing a phone to ring, and hearing ring-back tones or a busy signal. It also enables next-generation capabilities such as finding people and directing calls to them wherever their location, instant messaging (IM), and relaying so-called "presence" (near the phone or not, do-not-disturb, etc.) information. SIP began, rather humbly, as a simple message protocol for setting up connections. But the term has grown to be an umbrella for the family of protocols and tools that have been developed by the IETF to enable VoIP telephony and related services.

By the mid-1990s, audio and video were becoming routine on the Internet. Going beyond email, academic researchers were imagining online audio/video/whiteboard conferences where ideas could be shared live. It became clear that the *Mr. Watson come here, I want to see you!*[4] function had to be done more efficiently than by shouting across the college quad or sending invitation mails. Thus, the IETF's working group MMUSIC (Multiparty Multimedia Session Control) was born. There had already been work within the telephone world that had resulted in an ITU standard, but Internet types didn't much like it. "Too complicated," they averred. "Too limited," they sniffed. "Too *phone company*," they huffed. So off they went to do it the *Internet* way. The document describing SIP was eventually published as proposed standard RFC 2543 in 1999. Work has been ongoing, with the latest version of the specification, at the time of this writing, being RFC 3261.

The SIP message protocol is similar to the Web's HTTP (Hypertext Transfer Protocol) and shares some of its design principles: It is human readable and request-response structured. SIP even shares many HTTP status codes, including the familiar "404 not found."

[4]Yes, this is what Bell really said. Look it up on the Smithsonian's web site. No spilled acid involved.

Here is a typical SIP message:

```
INVITE sip:skip@there.com SIP/2.0
Via: SIP/2.0/UDP 4.3.2.1:5060
To: Skip Pizzi <sip:skip@there.com>
From: Steve Church <sip:stevec@here.com>
Call-ID: 4678995554545@4.3.2.1
CSeq: 1 INVITE
Contact: <sip:stevec@4.3.2.1>
Content-Length: 126
```

This is how Steve's SIP client would signal to Skip's that he wants to connect and speak with him.

SIP works together with several other protocols and is only involved in the signaling portion of a communication session. SIP is a carrier for SDP (Session Description Protocol), which describes the media content of the session (e.g., the codec being used, the bitrate, etc.).

SIP provides the following capabilities:

- Determines the location of the endpoint: SIP supports address resolution, name mapping, and call redirection.
- Determines the media capabilities of the endpoint (i.e., which codecs are available and supported): During a negotiation, SIP determines the best codec that can be used by the parties on the call.
- Determines the availability of the called endpoint: If a call cannot be completed because the target endpoint is unavailable, SIP returns a message indicating this and why.
- Establishes a session between the originating and called endpoints (if the call can be completed).
- Handles the transfer and termination of calls: SIP supports the transfer of calls from one endpoint to another. During a call transfer, SIP establishes a session between the transferee and a new endpoint (specified by the transferring party), and terminates the session between the transferee and the transferring party.

6.2.1 Parts of a SIP System

Like most things based on IP, SIP was designed to be modular. Implementers can pick and choose among the following elements to build the system they need:

- *SIP clients.* Sometimes called *user agents* or *endpoints*. These can be hardware phones or "soft phones" (phone applications running on PCs).
- *Gateways.* When needed, gateways translate between the IP network side and the switched-circuit telco side, providing physical, electrical, signaling, and audio interface.

- *Proxy server.* Receives SIP requests from a client and forwards them on the client's behalf. Basically, a proxy server receives SIP messages and forwards them to the next SIP server in the network. Proxy servers can provide functions such as authentication, authorization, network access control, routing, reliable request retransmission, and security.
- *Redirect server.* Provides the client with information about the next hop or hops that a message should take, and then the client contacts the next-hop server or client directly.
- *Registrar server.* Processes requests from clients for registration of their current location. Registrar servers are often colocated with a redirect or proxy server.

While SIP-enabled endpoints are able to connect directly to each other, SIP servers provide a number of valuable services, including the following:

- Register SIP client devices
- Register individual human users for access to their services
- Perform authentication, authorization, and accounting (when needed)
- Look up the address of the far endpoint
- Support user mobility across networks and devices
- Support for multipoint conferencing
- Support presence information
- Inform users as to call progress
- Communicate requests for QoS to various network elements, such as IP routers

While the various servers just noted could run on individual machines and could even be physically separated by thousands of kilometers, in usual practice they are often just software elements of an application running on a single machine. As we'll see, many small IP PBXs include the gateway as well, making a one-box solution that includes everything needed for a small-office installation.

An example of a SIP server being used in the broadcast world comes from the Telos Z/IP codec family, which uses a Telos-developed, enhanced SIP server, called (naturally) the Z/IP server. The server is provided as a service on the Internet, but may also be installed by users who prefer to maintain their own. In addition to basic SIP functions (registration, address lookup), the Z/IP server offers additional services:

- Allows display and dialing by simple text name. Keeps a database of names and performs DNS/IP lookup upon a dialing request from an endpoint codec.
- Maintains group lists created by users. Upon entering a group name and password, the list is displayed on endpoint codecs so that users aren't burdened with having to enter or upload lists manually.
- Provides geolocation services by associating IP numbers with physical location. Allows the display of a routing map on the codec LCD display.
- Upon request, keeps a record of network performance in order to assist in troubleshooting problems caused by QoS impairments.

Many products that support SIP for its standards-based interconnection capability do not have an internal architecture corresponding to the SIP specifications, so you would not see these SIP server components listed therein; or they might be included, but not labeled according to the standard names bulleted above. Instead, these functions would just be provided as part of the system "black box." Cisco and Microsoft VoIP products fall into this category, for example.

6.2.2 Addressing

SIP addresses, also called SIP URIs (uniform resource identifiers), are in the form sip: user@host. The user portion of the address can be a text name or a telephone number, and the host portion can be a domain name or an IP address. The address resolution process normally begins with a URI and ends with a username at an IP address. Just as with email, the sender needs no information about the physical location or IP address of the receiver. This is one of the powerful features of SIP: It automatically implements portability and mobility.

Examples of valid SIP addresses are as follows. The usual form is an email address prefixed by "sip:":

```
sip:joesmith@company.com
```

You can call a PBX telephone at a business (in this case, extension phone 123 at Telos Systems) this way:

```
sip:123@telos-systems.com
```

If you don't have a name or extension, you might want to contact the receptionist:

```
sip:receptionist@telos-systems.com
```

Here's an internal machine-to-machine message, such as from an on-air phone system to a PBX or gateway to initiate a PSTN call:

```
sip:12162417225@168.123.23.1
```

Note that in this case, an IP number is provided to identify the concrete machine that is to receive the message. You usually don't want to use DNS (Domain Name System) for this because it takes unnecessary time for the lookup step, and because there may well not be a DNS name associated with a machine being used as a telephone server.

To assist readability, SIP lets you use +, −, and . phone-number separators. It removes them prior to processing:

```
sip:+1-216-241-7225@telos-systems.com
```

As you can see, SIP bridges the telephone and Internet worlds. Both Web-type and PSTN telephone number addresses are possible, and users on either network can reach those on the other.

Often, address resolution involves multiple steps and SIP message hops. A DNS server, a SIP proxy server, and a SIP redirect server might all be involved in a single name resolution, for example.

A few other points of interest regarding SIP: Some servers associated with SIP systems can accept unformatted text names, but this is not part of the standard.

URIs are not URLs (universal resource locators). URIs are independent of the location of the named object. Email addresses are an example of URIs. In SIP, a request URI is defined to indicate the name of the destination for the SIP request (INVITE, REGISTER, etc.). URLs describe the location of a resource available on the Internet. For example, *http://www.telos-systems.com* is the URL for a Web home page. It is resolved by DNS to a concrete IP address.

PSTN telephone numbers are sometimes called E.164 numbers, a designation applied in an ITU-T standard that describes the format of telephone numbers to be used worldwide. ENUM (*E*.164 *NU*mber *M*apping) is the Internet service used to look up the URI associated with a particular E.164 telephone number. It's part of the DNS system. SIP can use ENUM to locate the VoIP system associated with a telephone number that accepts incoming calls.

6.2.3 How SIP Works

As we've seen, SIP is a simple, text-based protocol. It uses requests and responses to arrange for communication among the various components in the network, and ultimately to establish a connection between two or more endpoints (Figure 6.1).

But such a direct connection is uncommon. Almost always, there are SIP servers of various kinds in the picture. When a user initiates a call, a SIP request is sent to a SIP server (either a proxy or a redirect server). The request includes the address of the caller and the address of the intended called party. In more sophisticated scenarios, users register with a registrar server using their assigned SIP addresses. The registrar server provides this information to the location server on request.

From time to time, a SIP user might move between end systems. The location of the user can be dynamically registered with the SIP server. Because the end user can be logged in at more than one station, and because the location server can sometimes have inaccurate information, it might return more than one address for the end user. If the request is coming through a SIP proxy server, the proxy server tries each of the returned addresses until it locates the end user. If the request is coming through a SIP redirect server, the redirect server forwards all the addresses to the caller in the contact header field of the invitation response.

When communicating through a proxy server, the caller sends an INVITE request to the proxy server and then the proxy server determines the path and forwards the request to the called party.

Since we usually need to reach phones that are connected to the PSTN, gateways will be involved in real-world systems (Figure 6.2). These translate SIP signaling to the PSTN's requirements: loop current, DTMF, and ring detect for POTS lines; setup messages for ISDN; etc.

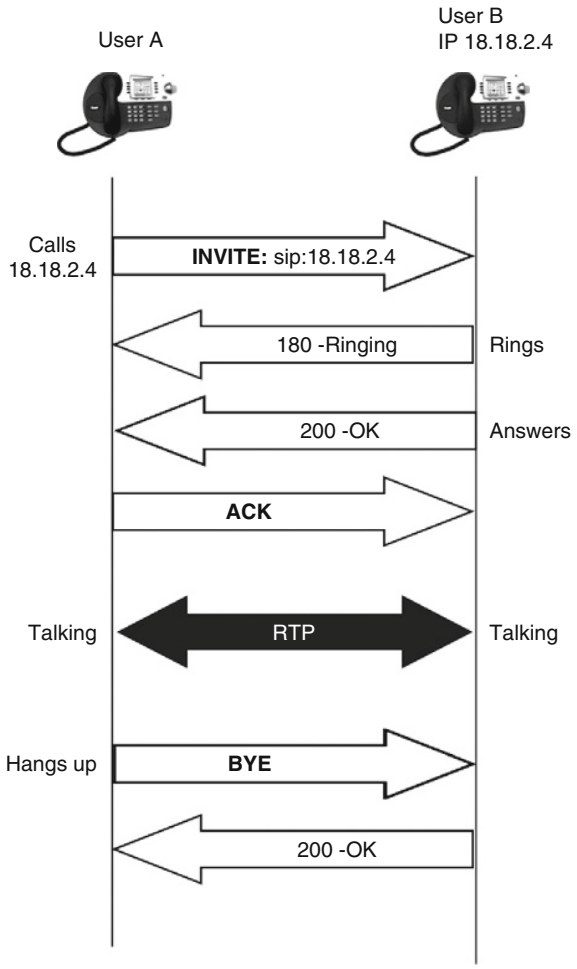


FIGURE 6.1

SIP at its simplest: An IP phone calls another directly.

SIP messages may be carried by UDP or TCP. SIP has its own built-in reliability mechanisms, so it doesn't need TCP's reliability services. Most SIP devices such as phones and PC clients use UDP for transmission of SIP messages. PBXs on LANs almost always use UDP because LANs don't drop packets and there is no need to incur the overhead of TCP. *Transport Layer Security* (TLS) protocol is sometimes used to encrypt SIP messages. TLS runs on top of TCP. (This is the protocol used with HTTP to make the secure HTTPS used for secure Web transactions.)

Not shown in our transaction examples is the media negotiation that is part of the INVITE/200 OK/ACK sequence. Through this process, endpoints decide which

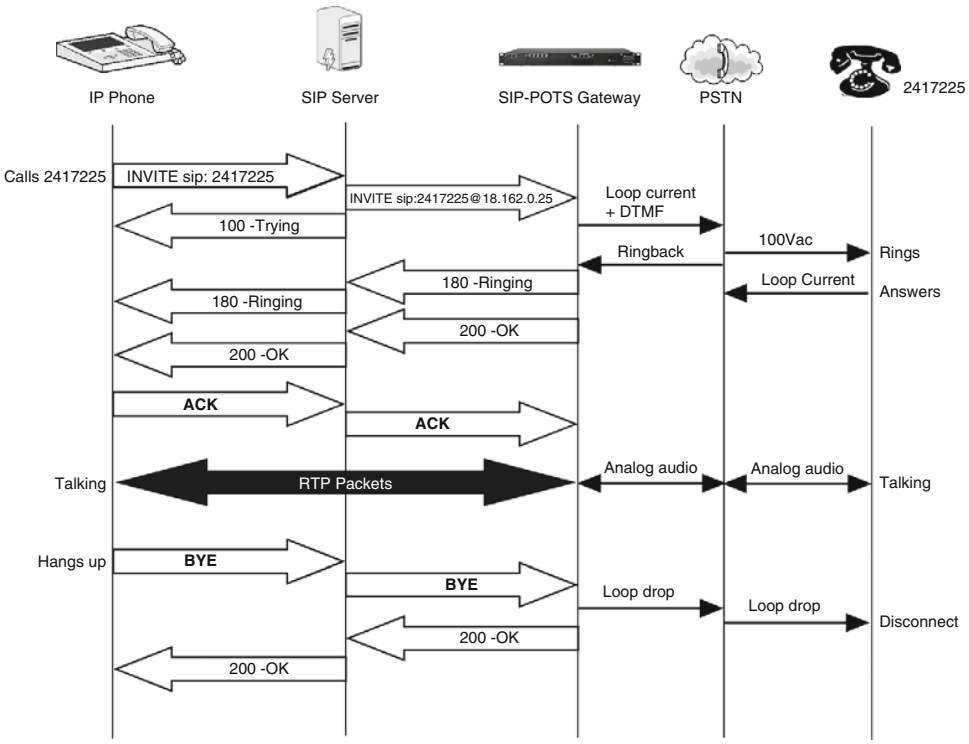


FIGURE 6.2

A SIP call setup to the PSTN via a SIP server and a gateway to POTS lines.

codec to use. The SDP defined by RFC 2327 is the way codecs are offered and (hopefully) accepted by the other end. Usually, the caller sends an SDP message along with its INVITE, listing the codecs it is prepared to use. The far end chooses one of them and tells the caller which it prefers in the 200 OK response. The caller can let the far end propose a codec by not sending an SDP message in its INVITE. It is possible that the two endpoints have no codec in common and the connection is unable to proceed, but systems are designed so that this does not happen. For example, almost all phones, gateways, and SIP telco services have G.711 as a supported codec, so this is an insurance policy that two endpoints will find common ground. Within a PBX system, designers usually choose one codec as a standard for the system and stick with it for all connections. For example, the Telos VX studio system uses an 8-kHz sampling rate, 16-bit uncompressed PCM internally for all calls that connect to the PSTN.

6.2.4 The State of SIP and Its Future

None of the PBXs described in this chapter use SIP as it was intended by its developers. None use SIP servers at their core. All use their own rough equivalents,

designed independently. So what went awry? The SIP schemers were certainly far-ahead thinkers, who wanted their protocol to support rich media, mobility, portability, sophisticated endpoints, etc. The problem seems to have been a lack of a certain "practicality." For example, consumer PC-to-PC VoIP products needed to solve the problem of firewall and NAT traversal, which has been addressed quite slowly within the SIP working groups. Meanwhile, Skype's developers solved it quickly and effectively. Then there is the problem of supporting all the features a vendor wants to employ to differentiate its product. It's unsurprisingly faster to just implement it your own way rather than waiting for the idea to make its way through a committee, who might well not see things your way.

At Telos, we faced this problem in the design of the VX studio system. We needed a lot of things specific to the studio environment that are not supported in SIP's structures. So we designed our own protocols for use within the boundaries of our system. But we use SIP at the border of the system to connect with other vendors' products, and eventually to the Telco network. This is just the strategy Cisco, Microsoft, and almost all PBX vendors have followed.

And now *this* process is emerging as SIP's great value. It's the glue that ties systems together. Studio systems can talk with PBXs for the first time. PBXs can talk with each other. And eventually they will all be able to talk to telco networks, smoothly and fluently. SIP's inventors got something right.

6.3 IAX AS A SIP ALTERNATIVE

SIP is not the only game in town. The Inter-Asterisk eXchange (IAX) protocol is an alternative to SIP for interconnections between both VoIP servers and for client-server communication.

IAX2 (as the current version is named) uses a single UDP data stream (usually on port 4569) to communicate between endpoints, both for signaling and data. The voice traffic is transmitted in-band, in contrast to SIP, which uses an out-of-band RTP stream for audio. IAX2 supports multiplexing channels over a single link. When trunking, data from multiple calls are merged into a single set of packets, meaning that one IP datagram can deliver control and audio for more than one call, reducing the effective IP overhead without creating additional latency.

As IAX's name indicates, it was invented by the Asterisk[5] people as a way to trunk calls between one Asterisk server and another. It has escaped from Asterisk and is now supported in a variety of soft switches and by a few VoIP carriers. Its main advantage is its bandwidth efficiency and simpler firewall configuration, since all traffic flows through a single port.

[5]Asterisk was one of the first software-based PBX implementations (1999), of which there are now many. Its "hybrid" open-source or proprietary model has attracted a large number of developers to the system. There is more about Asterisk later in this chapter.

6.4 CODECS

Among the benefits of VoIP is the opportunity to use a variety of codecs, depending on the nature of the transport network and the needs of the application. Table 6.1 provides a list of commonly used VoIP codecs.

The packet size given in the table is the default and some equipment lets you change it, depending on what the codec allows. For example, G.711 is often set to 10 ms in order to reduce latency. As we've seen, there is a trade-off: The smaller packet size results in more IP header overhead and thus lower bandwidth efficiency. Smaller packets also consume more processing power in the equipment.

The rates given in the table for the MPEG codecs are the target rates. Useful rates range from 32–96 kbps for AAC-LD and 24–96 kbps for AAC-ELD.

Table 6.1 Codecs Used by VoIP Systems

	Audio B/W	Bitrate	Packet Size	Bitrate after Packetization	Notes
G.711 u-law	3.4 kHz	64 kbps	20 ms	88 kbps	U.S. PSTN standard
G.711 A-law	3.4 kHz	64 kbps	20 ms	88 kbps	European PSTN standard
G.729a/b	3.4 kHz	8 kbps	20 ms	32 kbps	Common lo-fi VoIP codec
G.723.1	3.4 kHz	5.3 or 6.3 kbps	30 ms	22.3 kbps	Very low-rate codec
G.726	3.4 kHz	16–32 kbps	20 ms	40–56 kbps	Better quality than G.729
G.722	7 kHz	48/56/64 kbps	20 ms	88 kbps (at 64 kbps)	Wideband; simple low-delay ADPCM codec; now in Cisco phones
G.722.1 Annex C	14 kHz	24/32/48 kbps	20 ms	40/48/64 kbps	Wideband; also called Siren14; invented by Polycom, and used in its video conferencing systems
G.722.2/ AMR-WB	7 kHz	6.6–23.85 kbps	Variable	Unknown	Wideband; ITU mobile phone standard

Table 6.1 Codecs Used by VoIP Systems—cont'd

	Audio B/W	Bitrate	Packet Size	Bitrate after Packetization	Notes
G711.1	7 kHz	64/80/96 kpbs	Variable	Unknown	Wideband extension to G.711
G.729.1	4 kHz/ 7 kHz	8–32 kbps	Variable	Unknown	Newer, scalable version of G.729; at higher rates, becomes a wideband codec
iLBC	3.4 kHz	15.2 or 13.33 kbps	20 or 30 ms	Unknown	Proprietary codec invented by Global IP Sound; available in some Cisco phones
RTAudio	3.4/ 7 kHz	8.8/18 kbps	20 ms	39.6/58 kbps (includes FEC)	Microsoft proprietary codec, used in the Office Communications product family; has both narrow and wideband modes
MPEG AAC-LD	20 kHz	48–64 kbps	10 ms	96 kbps (at 64 kbps)	Full-fidelity; used in some IP video conferencing products
MPEG AAC-ELD	20 kHz	32–64 kbps	20 ms	48 kbps (at 32 kbps)/96 kbps (at 64 kbps)	Newest codec in the AAC family; full fidelity for music and voice at low rates; used in broadcast codecs.

Note: *Audio B/W values shown are actually the upper limits of the codecs' audio-frequency response.*

CARBON-BASED (AUDIO) LIFE FORMS The PSTN uses the G.711 codec. Its audio-frequency response is limited to 3.4 kHz. A modern reader might ask, "Why was such a low fidelity considered to be satisfactory for digital telephony?" Microphones, loudspeakers, and earphones have all had much better fidelity for many decades. Indeed, when the G.711 codec was standardized in the 1960s, FM radio was just getting going with its much superior 15-kHz bandwidth. Mostly the coding choice resulted from the legacy of the carbon button

(continued)

CARBON-BASED (AUDIO) LIFE FORMS—cont'd microphones that were ubiquitous in telephones throughout most of the early history of telephony. (Edison invented the carbon microphone and licensed it to the Bell System, which had only an impractical liquid-based microphone in its own portfolio.) This microphone has a nonflat response curve with a 5-dB peak at 2 kHz and hefty roll-offs below 300 Hz and above 3 kHz. They were mostly abandoned for all but telephone use in the late 1920s, but were standard in phones up to the 1980s. (See Figure 6.3.)

Old analog long-distance lines also had a lot of high-frequency attenuation owing to capacitive effects. When microwave radios were introduced to long-distance telephony, a decision had to be made as to what frequency range to accommodate in their FDM (frequency division multiplex) scheme. There was, as always, a trade-off: More frequency response meant fewer channels. Since the microphones weren't producing much in the way of high frequencies, why bother carrying them over the radio links? Thus, the radios were designed with narrow 4-kHz carrier spacing. When the first digital T-Carrier systems where invented, it must have seemed perfectly natural to stay with the 3 kHz or so audio bandwidth enshrined in the microwave link technology. A sampling rate of 8 kHz with 8-bit (compressed) depth had a nice symmetry and delivered a satisfactory 4-kHz Nyquist response limit,[6] so on with the show.

FIGURE 6.3

A Western Electric carbon button microphone, invented by Edison and used in phones up to the 1980s. What was the reason for telephony's long-standing lo-fi standard? Rapping the handset on a table was said to loosen the granules and improve fidelity. (*Source:* Wikipedia under GNU Free Documentation License. Photo by Gary Ashton.)

[6]The Nyquist frequency is half the sampling rate, and is the theoretical audio bandwidth limit. At 8 kHz, the Nyquist frequency is 4 kHz. The relatively primitive filters used in G.711 codecs need a 600-Hz transition band, resulting in the 3.4-kHz actual bandwidth.

df/d$ Interestingly, in a 1938 paper in the *Bell System Technical Review*, an AT&T engineer, A. H. Inglis, wrote, "Frequency limitation is essentially an economic one, subject to change as conditions change." It has taken some time, but Mr. Inglis is being proven correct. A few years into the 21st century, conditions have changed, and we're finally able to escape the long-lasting drag of Edison's 1878 invention.

Early implementations of VoIP have perhaps given it a bad reputation with regard to audio quality. Some of this was due to network QoS problems, but much can be laid at the feet of the codecs that were chosen. The widely used G.729 has a bitrate of 8 kbps, and just the audio fidelity you would expect from such a low rate. Thankfully, modern VoIP PBXs have settled generally on at least G.711 for internal calls, and many are now moving up to G.722 wideband codecs, or even better. Telcos providing "business-class" IP services are almost always offering G.711 as their baseline codec, rather than the old low-grade G.729 standby. Users of popular VoIP soft phones such as Skype have undoubtedly noticed that audio quality is better than on usual phones, at least in terms of frequency response. Calls from mobile phones that are eventually upgraded to the G.722.2/AMR-WB codec will suffer a downgrade in fidelity when they are carried over the PSTN, but should retain their wideband quality when passed to landlines via IP. All of this means that VoIP is likely to be the beneficiary of a reputation makeover in the coming years, with users coming to associate VoIP with *premium* quality.

6.5 PACKETIZATION AND RTP

Real-Time Transport Protocol (RTP) is the standard for both VoIP and AoIP networks. RTP runs over UDP IP packets. As we've seen, with UDP there is no mechanism for recovering lost packets. VoIP uses UDP rather than TCP because the latter would increase delay, owing to the need for long receive buffers to cover the time that might be required to receive any lost and retransmitted packets. On LANs, where there is no packet loss, UDP's lack of packet-recovery is no problem. When using UDP on wide area networks, however, dropped packets have to be addressed within the audio codec, which must have concealment to reduce audibility of lost audio samples. This is particularly a requirement for wireless and public Internet applications, where packet loss is a routine occurrence. Another solution is to have guaranteed QoS on any WAN IP links. This is possible on private or "virtual" private networks such as the VPNs that link corporate headquarters to branch offices. As we'll see, it's becoming possible to order IP telephone service from telco providers with QoS guarantees.

The RTP header is 12 bytes in its usual form. This is added to the UDP (8 bytes) and IP (20 bytes) headers to make a total header length of 40 bytes. The codec output is broken into segments and put into the IP packets following the headers. Some codecs are frame based and thus have an inherent packet-ready format. For example, G.729 has a 10-ms frame, which could be placed one-to-one inside IP packets. But usually two

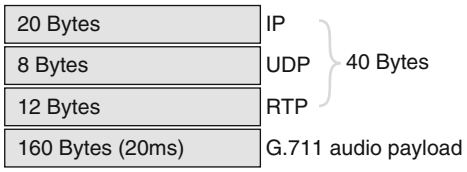


FIGURE 6.4

IP/UDP/RTP header plus 20 ms of G.711 audio payload per packet is an efficiency-versus-delay trade-off. Each codec type may make a different trade-off.

frames are put into one packet to improve efficiency. The MPEG codecs have longer frame lengths and are usually packetized one-to-one. Codecs such as the G.711 companded-PCM and G.722 ADPCM work on a sample-by-sample basis and have no inherent frames, so they may be packetized at any desired boundary. Usually 20 ms is chosen as a compromise between delay and efficiency, but sometimes 10 or 30 ms is used when either lower delay or higher efficiency is preferred, respectively. (See Figure 6.4.)

For example, using G.711, there are 80 bytes of data produced for each 10 ms of audio. A 40-byte header on an 80-byte payload is not out of the question, but is not particularly efficient. That's why the VoIP world has settled on 20-ms packets. This means there are 40 bytes in the header and 160 bytes in the audio payload—a reasonable compromise.

On LANs, we have so much cheap bandwidth that we can afford to "throw it away." This is just what we do for studio-grade AoIP systems, where the header can be even larger than the audio payload. Our goal there is very low delay—much lower than the target for VoIP systems. In VoIP systems that run only over LANs, we can similarly decide to let low delay take priority over efficiency and operate with smaller packets. For example, the Telos VX studio IP phone system uses 10-ms packets for G.711-coded telephone audio.

On the other hand, VoIP systems that run over WANs must use bits more efficiently. In private networks, bandwidth is expensive. On the uncontrolled public Internet, you have a higher chance for unbroken conversation if the bitrate is kept to a reasonable level. For all these reasons, header compression is sometimes used. The robust header compression (ROHC) specified in IETF RFC 3095 seems to be an increasingly deployed method, especially for wireless VoIP. There are currently two ROHC profiles defined for the compression of IP/UDP/RTP traffic: the original definition in RFC 3095 and the more recently published RFC 5225.

6.6 DELAY

Delay is primarily a function of packet size and jitter. Clearly, the longer the packet, the more time it takes to gather up the audio samples, and the more the delay. Jitter plays a more subtle role. It determines how many packets have to be buffered in the receiver. The buffer has to be long enough to cover the latest arriving packets. In VoIP systems, the buffer is often a user configuration item, which is set by

experience. A value is chosen that results in few packets falling over the buffer time limit. On LANs, there is no significant jitter and the buffer can be as little as two packets. As usual, the public Internet is the most challenging situation, not only because there can be long delays, but because the delay is so variable. For this reason, adaptive buffers combined with effective concealment in the codec is the best strategy to ensure uninterrupted audio. We discuss more on this topic in Chapter 7.

One effect of delay is echo—a talker's voice being returned back to him via some kind of leakage along the transmission path. The usual cause is a poor hybrid at the interface of the digital and analog circuits at the far end of a path that includes a POTS line. Another source of leakage is mechanical coupling between the earpiece and microphone in the far-end telephone handset. Yet another is acoustic coupling when a loudspeaker phone is used at the far end. Such a phone needs to have either a ducker or an acoustic echo canceller that can be counted on to maintain many tens of decibels of send-to-receive isolation. Because VoIP has more delay than analog or circuit-switched digital speech paths, the demand put on the system for low leakage is higher.

Generally, VoIP system designers expect to achieve at least 35- to 45-dB ERL (echo return loss) and thus target 150 ms as the maximum roundtrip delay. (Subjectively, the longer the delay of echoed speech, the more noticeable and annoying it is, so keeping the delay short also reduces the requirement for high ERL–see Figure 6.5.) IP PBX systems designed for operation on LANs would have much lower delay, perhaps in the 50-ms range. Echo is not the only reason to keep delay as low as possible;

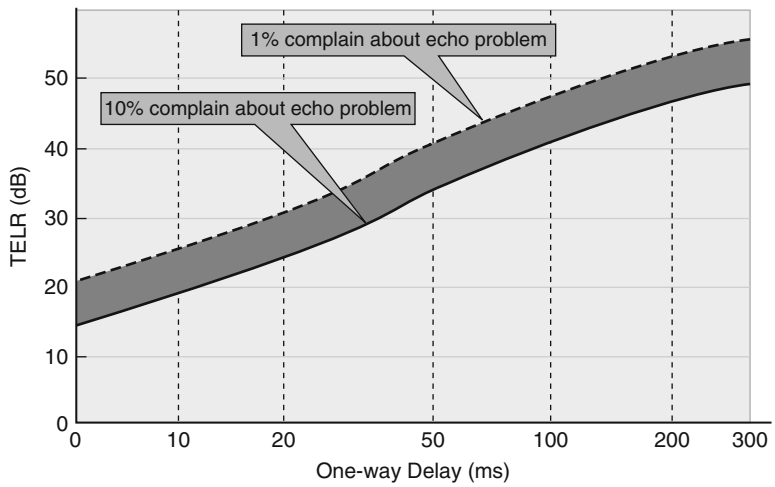


FIGURE 6.5

Reaction to echo delay is a function of both amplitude and time (from ITU G.168). TELR = Talker Echo Loudness Rating, the difference between the perceived volume of the caller's own voice in real time and that of its echo, measured in dB. Like an S/N ratio, a higher TELR indicates a quieter echo. "Annoyance contours" shown indicate that area below the 10% curve (solid line) are unacceptable, and area above the 1% curve (dotted line) are ideal.

the natural flow of conversation depends on delay not being too high, as well. The 150-ms VoIP target has also been found to be adequate for this aspect.

6.7 IP PBX

If reported statistics can be believed, 80 percent of all new PBX lines installed worldwide in 2008 were VoIP. Plenty of older TDM (time-domain multiplex) PBXs are still in service and will remain so for years, but VoIP is clearly on the rise. There is probably an IP PBX in your future.

A typical system would be comprised of call management software, gateways to/from the IP network and POTS or ISDN telco connections, and IP phones (Figure 6.6). Call management software could run on a PC or on specialized, dedicated hardware. Application servers provide any needed additional functions, such as voicemail. In some systems, these are integrated into the call management software, or at least run on the same machine.

If the connection to the telco is via SIP, there would be no need for a gateway; the local IP network would connect to the telco's IP network. Presumably, a firewall would protect the local network from anything destructive that might enter via the telco connection. Configuration and management is via a web browser pointed at one or more system elements.

Most vendors use proprietary communications protocols between their call management application and their telephone sets. They say that this is needed to support the features on the phones, such as displays and soft buttons. Most also support a basic variant of standard SIP, allowing third-party SIP phones and other endpoints to be attached to the system.

There are generally two paths to adding third-party devices to SIP PBXs. One is to emulate a telephone set. This makes sense at first blush, but it may not be as simple as it seems. The reason is that telephone-like devices will probably need to be *registered*—that is, to let the main unit know the device is there via a special SIP message. Implementation of this differs across products (the ghost of proprietary strikes yet again), making it somewhat troublesome. In contrast, *SIP trunking* is generally simple, straightforward, and preferred. Fortunately, almost all IP PBXs support that.

6.7.1 Cisco

According to published reports, in 2008 Cisco became the leading vendor of PBXs worldwide, besting the long-standing leaders, Nortel[7] and Avaya. Being primarily an IP router company, Cisco's PBXs are IP-only.

[7]In January 2009, those who follow events in the telephone world were shocked to learn that Nortel had declared bankruptcy. The company, more than 110 years old and with 32,000 employees, has been a giant in the central office and PBX equipment business. The cause was almost certainly that the company had come under pressure from IP PBXs.

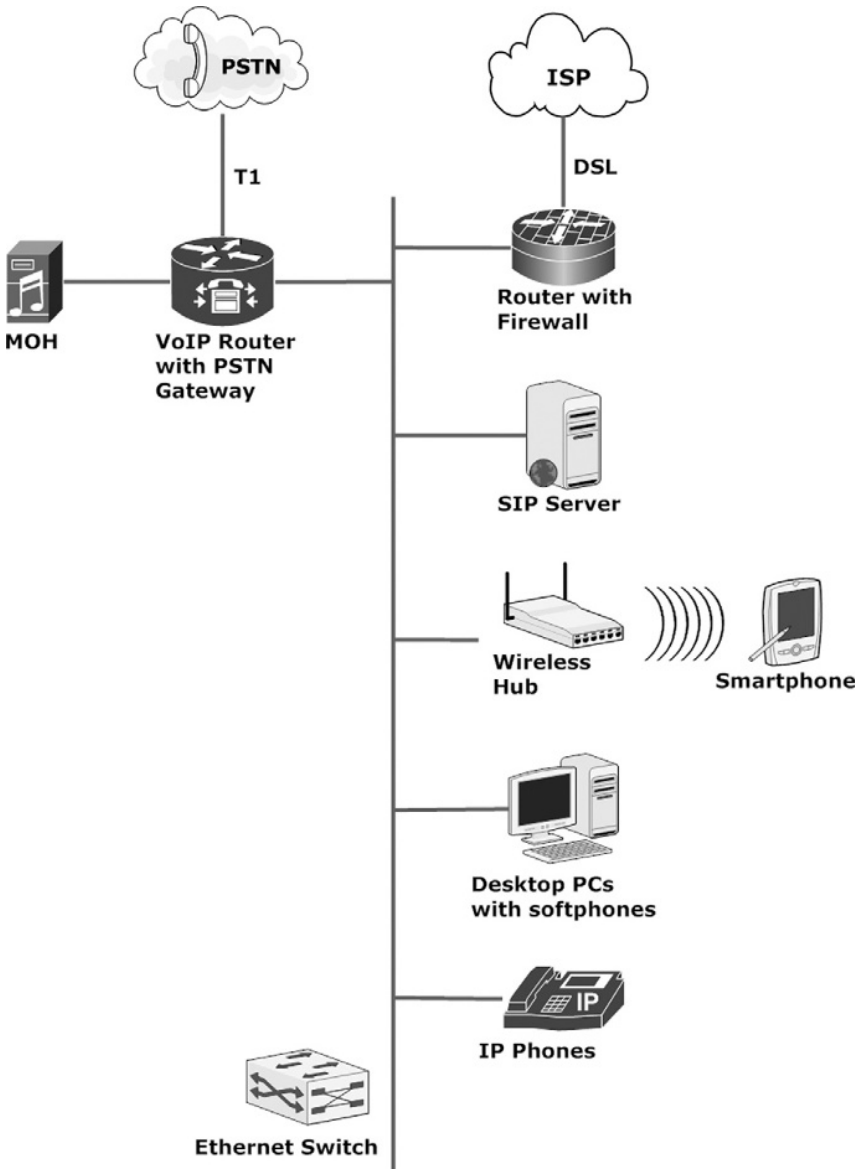


FIGURE 6.6

A simple SIP-based VoIP system. The VoIP router could be Cisco's Call Manager Express (with its included PSTN gateway), Asterisk, or another integrated product. Calls could also be passed to an IP network directly over the firewalled DSL connection. Clients can be PC soft phones, hardware IP phones, or even radio-linked smart phones. For a basic system, the SIP server would not be needed, but one could be used to provide services such as interactive voice response. An existing TDM PBX could be integrated via a gateway.

Cisco's critics say that its equipment and service contracts are too expensive, that its phones should run SIP (they do now, actually), and that only experts with years of training can figure out how to install and configure its arcane systems. But its commanding market share suggests it must be doing something right. Perhaps some of Cisco's success can be attributed to its dominance of the IP router business, with all of the IT types who have had Cisco training and who are experienced and comfortable with the company's products.

Anyway, the phones look good, and some have a big, color display. See, for example, Figure 6.7.

Cisco makes two product families, Cisco Unified Communications Manager for large installations and Cisco Unified Communications Manager Express for small ones.

Cisco Unified Communications Manager

Cisco Unified Communications Manager is intended for large companies, university campuses, and hotels. A system would usually include a PC-based software application and Cisco IP telephones.

Unified Communications Manager is a soft switch. It is software that runs under Windows or Linux on a PC server. The application acts as the registry, the SIP proxy, and the telephony feature server. But there is no on-board interface to telco

FIGURE 6.7

A Cisco 7971 IP phone.

circuits. Indeed, once the call is signaled and set up, the voice or video media stream is peer-to-peer, not passing through the server. (This is in contrast to some traditional IP-enabled PBXs that stream the audio through the PBX continually.) That is how Cisco's approach allows a single server to support thousands of users. The audio switching is accomplished by the Ethernet switch. Since modern Ethernet switches are nonblocking, supporting as much bandwidth as there are ports, from an audio transport perspective there is no limit to the number of phones in a system.

If a connection to traditional PSTN lines is required, such as to POTS or ISDN, a gateway is used to bridge the telco and IP worlds. The device would typically have ISDN PRI or T1 on the telco side and, of course, SIP and IP on the Cisco side.

Applications such as voicemail can be added to a soft switch–based system by attaching additional servers to the network. This illustrates one of the important benefits of VoIP: The system bus in an old-fashioned PBX is proprietary and contained within the box, so there is no chance to attach something to it. With VoIP, the "system bus" is as accessible as the nearest Ethernet port. In Cisco's case, starting with version 6.0, Unified Communications Manager Business Edition (CUCMBE, a.k.a. "Cucumber") places Cisco Unified Communications Manager and Cisco Unity Connections (voicemail) on the same server.

Cisco Unified Communications Manager Express

Cisco Unified Communications Manager Express is for small- to medium-size businesses. The telephone software runs on an Integrated Services Router, such as the 2800 family, which provide routing, VPN, and firewall functions. These are 1 or 2U rack-mount boxes that have slots for a variety of interface modules. The cost and size of the particular box within the family determines how many and what type of slots are available. See Figure 6.8 for an example.

Normally, there would be some kind of module installed to interface to an IP WAN network for data traffic, and another set to connect to the PSTN, which could

(a)

(b)

FIGURE 6.8

Cisco 2800 Integrated Services Router, (a) front and (b) rear panels. (Photos by Steve Church.)

be via POTS, ISDN, or T1/E1. The software can support up to 250 users, and includes modern PBX features like auto-attendant and voicemail. It connects to Cisco IP phones using both SIP and Cisco's Skinny Client Control Protocol (SCCP). The core software is Cisco's proprietary Internet Operating System (IOS). SIP support for both telephones and trunking is possible, so third-party devices can be quite easily attached.

In its latest-generation phones, Cisco has the G.722 codec that broadcasters know well, which permits audio bandwidth up to 7 kHz, which Cisco refers to as "wideband." This looks to be a trend in VoIP systems. Phones from Polycom, Avaya, Snom, Linksys, Mitel, and Grandstream all now support G.722. Calls that are made within a local network will be higher fidelity than we've been used to. And calls from among branch offices and headquarters that are linked by IP networks can also be wideband. Only when calls must traverse the PSTN will they be constrained to the old-fashioned 3.4-kHz frequency response imposed by telephone engineers in the 1960s. But we have to ask: Why don't they keep going? While G.722 is certainly better than standard phone quality, broadcasters have long ago abandoned it to the much higher fidelity of MPEG codecs, which use the same bitrate. When will VoIP vendors make the discovery?

6.7.2 Microsoft

As you would expect, Microsoft takes a mostly software approach to VoIP, rather than the hardware phones that Cisco and the other IP PBX vendors expect you will use. Microsoft offers a soft phone application called Microsoft Office Communicator 2007 (Figure 6.9). Communications Server 2007 is the backend, which works in combination with a Microsoft SQL server that provides the database services the systems needs. Office Communicator may also be linked to a Microsoft Exchange server for nonvoice operations such as email.

Knowing that some users will prefer something like a traditional hardware phone, Microsoft has developed the Office Communicator Phone Edition software package, which it licenses to companies that want to build compatible phones. In most respects, the Communicator Phone Edition phone is a physical version of the software Communicator, with a similar user interface and comparable functionality. Like the software Communicator, the hardware phone allows you to place a call either by using a numeric touchpad or by clicking one of the contacts. The LG-Nortel IP Phone 8540 is an example of one manufacturer's approach to such a phone (Figure 6.10). It comes with a 5.7-inch 320 × 240 pixel, color touchscreen LCD display. The operating system is Windows CE 5.0. The 8540 provides security via a built-in biometric fingerprint reader used to authenticate the user of the phone, ensuring that only authorized users are granted access to personal and corporate information such as voicemail, call logs, contact lists, and calendars. Network connectivity is through an integrated two-port Ethernet 10/100 BASE-T switch with RJ-45 connector ports. This allows for a direct connection to the network for the phone as well as the collocated PC.

FIGURE 6.9

Microsoft Office Communicator application. A voice call is started by clicking on the phone handset icon to the right of a contact's name.

FIGURE 6.10

LG-Nortel 8540 IP phone, designed to work with Microsoft Communications Server 2007.

FIGURE 6.11

Polycom CX200 desktop USB phone.

If you are using the Communicator soft phone application, you will normally connect a headset or handset to the host PC's sound card. There are a number of products that could be used, including computer keyboards that have a built-in handset. Another way to get a more traditional telephone-like feel would be to use something like the Polycom CX200 desktop phone (Figure 6.11). This connects to the PC via USB and includes keys for controlling volume, speakerphone mode, and use of an external headset. LED indicators provide the user with call forwarding status and message waiting indication.

Microsoft uses a proprietary signaling protocol between the Communicator clients and server. The server uses SIP to the outside world, however that means it can connect to TDM PBXs and PSTN lines through gateways, and to SIP-enabled PBXs directly. Microsoft offers something it calls a "mediation server" as part of its package. This translates between codec types that might not be supported in the gateway.

The main idea underlying Microsoft's tactic is that most people in offices have PCs on their desks, which make quite good phones. The PC's rich interface is a useful improvement over the usual office phone. In fact, there are many benefits to having the phone user interface on the PC's screen. Contacts can be shared with email and instant-messaging applications. You could start a conversation in email, escalate to IM, and then decide to make a voice or even a video call with but a few clicks of a mouse. This is described by the term *Unified Communications*, which is certain to become buzz-speak among corporate IT types within the next years.

We've grown so used to telephone numbers as a way to reach people that it seems almost natural. But, is it? As mobile phones and PC applications like Skype have shown, it's much friendlier to find people by name. PC applications do that

better than phones, but phones are catching up. Big color screens make phone interfaces more PC-like and remove some of their disadvantages.

With its VoIP system, Microsoft has also embraced wideband audio. Calls within the system are connected using a proprietary Microsoft codec called RTAudio.

6.7.3 Asterisk

With Asterisk, telephony services are implemented as software running on a PC. There is a free open-source version of the software and a paid version, which comes with support.

Asterisk demonstrates that telephony can escape the province of exotic and costly hardware. Telephone switching, voicemail, etc. can be just another service that runs on a standard-issue PC. Of course, specialized knowledge is required to configure and manage these systems. Yet we've seen many occasions where once rarified services—email, for example—have become commonplace. The same could happen with telephony as IT professionals become acquainted with systems like Asterisk.

Asterisk is not a soft switch, despite being downloadable software. Audio streams traverse through the box rather than being passed peer-to-peer via the Ethernet switch.

Asterisk supports on-board connections to conventional circuit-switched telephone lines and devices via PCI expansion cards sold by the developer's affiliated company, Digium, and others.

Asterisk PBXs are very often self-installed on PC hardware, but there are also "appliance" versions such as the Trixbox (Figure 6.12) that come more-or-less ready to use, with preinstalled and configured software. These are usually x86 based, but there are some available that are built on the Analog Devices Blackfin processor, benefiting from the lower cost and lower heat/power consumption of that CPU.

FIGURE 6.12

Trixbox is a ready-to-go PBX based on the open-source Asterisk software.

SIP-compatible telephones that can work with Asterisk are available from a number of vendors. Even Cisco's phones can be set to a generic SIP mode that will work. There are also plenty of PC-based soft phones.

6.7.4 Others

We've looked at a representative sampling of IP PBXs: one hardware, one software, and one Linux-based open source. But there are a number of other companies that offer IP PBXs in a variety of styles and with differing philosophies—among them, Nortel, Avaya, Mitel, Siemens, Alcatel, Shoreline, Spherecom, and Vertical. What matters from the perspective of a radio station or audio production facility is that your PBX should let you unrestrictedly connect devices using the SIP standard for call control and RTP for the audio streams. And the PBX should support the codecs you want to use without any unacceptable transcoding or anything else that would cause audio problems.

6.8 GATEWAYS

Gateways provide the bridge between the SIP/IP network and the telco network. Traditionally, that meant POTS, T1, or ISDN on the telco side. But modern gateways can be used to link to telco-hosted SIP services. In this case, the gateway becomes an IP firewall and router. Many units now have both IP and circuit-switched connections on the telco side. The IP connection can be used for both Internet data and voice.

Gateways provide all or a subset of the following:

- Physical translation between IP and circuit-switched telco networks (e.g., RJ-45 Ethernet to multiple RJ-11s for POTS connections)
- Signaling translation from SIP to the telco's format
- Call-progress tone generation and detection (i.e., busy, dialtone, etc.)
- DTMF tone generation and detection on both the IP and telco sides
- Caller ID detection on the telco side
- Audio transcoding between codecs
- Line echo cancellation (digital hybrid)
- IP router functions
- PBX-like services (not really a gateway function)

AudioCodes Mediant 1000 is a typical, modern, full-featured SIP gateway (Figure 6.13). It has six card slots that can accept interface modules for a mix from 1 to 4 E1/T1/J1 spans, from 4 to 20 ISDN-BRI lines, and from 4 to 24 analog FXS/FXO interfaces. Voice routing capabilities along with SIP-to-SIP mediation let the system be used as a basic standalone PBX. It can connect to a telco via SIP/IP and have a circuit-switched connection as a backup. It also serves as an IP router. It can transcode between G.711, G.726, G.723.1, G.729A, GSM-FR, iLBC, and G.711.1 codecs.

FIGURE 6.13

AudioCodes Mediant 1000 gateway, with a two-port T1/E1 and a four-port FXO module installed.

The Mediant 1000 MSBG version adds IP routing capabilities and a number of wideband codecs: G.722, AMR-WB, and Microsoft's RTAudio.

AudioCodes also makes lower-cost and much simpler desktop-style gateways in a nonmodular format. These come preconfigured from 2 to 24 POTS interfaces. (See Figure 6.14.)

The Cisco Integrated Services Routers, such as those in the 2800 family we have already seen, can be used as gateways.

PCs hosting Asterisk PBX make good gateways as well. Digium makes a variety of PCI-slot telco interface cards for POTS (Figure 6.15) and T1/E1. Some third-party vendors make cards for ISDN. This is not an off-the-shelf solution, and installing and configuring the cards and software is not for the faint-at-heart.

6.8.1 Overview of Circuit-Switched Interfaces

When you order or configure a gateway, you need to know what kinds of interfaces you will be using to connect to the telco network.

FXS/FXO

These are designations for the two ends of a standard analog POTS line. Most often these lines are used for basic home telephones. But they can also be used to link a PBX with a telco central office.

FIGURE 6.14

AudioCodes Mediapack gateway, supporting four FXO connections.

FIGURE 6.15

High-density POTS FXO interface card for Asterisk.

- An FXS (foreign exchange station) interface emulates a circuit supplied by a telco central office. An FXS supplies talk battery and detects an off-hook condition. It generates 100 VAC for a ringing indication. It provides dialtone and other call progress signals such as ring back and busy. It responds to DTMF (dual-tone multifrequency) tones for dialing and may send caller ID information in modem-encoded audio.
- A telephone, and anything that looks like a telephone, is an FXO (foreign exchange office) device. An FXO device signals an off-hook condition by drawing loop current. It responds to ringing voltage. It provides dialing, either by old-fashioned pulsed loop interruption or by DTMF. It may detect caller ID.

An interesting limitation of FXS/FXO interfaces is that signaling from the FXS that a call has ended is sometimes not signaled, and the type of signaling varies around the world. In the United States, most telco central offices interrupt the loop current when the call has ended, but some do not. And many PBXs do not. Eventually, dialtone will return, though, and this can be used as a disconnect signal, but there will be a many-second delay. This could cause *glare*, the condition where there is confusion between the CO and the PBX as to whether the line is free. A call could ring-in just when the PBX attempts to access the line for an outgoing call. In many other countries, an audio tone is sent on the line to indicate the end of the call. This lack of *disconnect supervision* results from the idea that there was no reason for a central office to hang up a phone by remote. This was a physical, human action

performed by someone who knew that the conversation had ended and reacted by replacing the receiver back in its cradle. The design was never intended to work for machine-to-machine connections.

Actually, there are two types of off-hook signaling. On a *loop-start* line, when the phone goes off hook, the circuit is closed, and the central office detects the change in current. This is the common residential format. *Ground-start* signaling is a small modification to the scheme to permit disconnect supervision and remove the possibility of glare. In an idle circuit, the central office provides −48 v on the ring wire and an open on the tip wire. From the PBX side, ring is grounded first, then the central office circuit must respond by grounding tip. The PBX senses this, releases its ground, and maintains the connection by drawing loop current. When the conversation is finished and the line is to be cleared from the CO side, the CO removes the load across the pair, and the PBX accepts this as indicating a disconnect request.

E&M Trunks

E&M trunks use two extra wires for signaling, the so-called *ear* and *mouth* connections. These solve the problems with glare and disconnect supervision. This scheme is nearly obsolete, but occasionally E&M interface cards are used to connect music-on-hold (MOH) to VoIP PBXs. They are convenient for this purpose because the audio path is transformer isolated and there is no need to supply talk battery from the MOH source.

T1/E1

These are basic digital interfaces to the switched voice network, and are widely used. This is especially so in the United States, where T1 is nearly standard for large PBXs. (In Europe, ISDN PRI is more widely employed.) T1 transports up to 24 voice channels, while E1 supports as many as 32. T1s are common in the United States and Japan, while E1s are provided by telcos in most of the rest of the world. In addition to the audio, these digital circuits also carry basic signaling in channel-associated signaling (CAS) bits. This signaling emulates loop start, ground start, or E&M, depending on the configuration.

T1s can also be used for IP connections. In this case, usually an entire T1's 1.544-kbps capacity is used as a transparent pipe from the local IP router to the ISP's equipment. The phrase *channelized T1* is sometimes used to distinguish a T1 that is intended for circuit-switched voice application. A *fractional T1* is a service that uses a portion of the line's full capacity. It is sometimes possible to order a T1 that is divided into a channelized portion and a data-transparent part for IP connectivity.

ISDN-PRI

ISDN-PRI (Integrated Services Digital Network Primary Rate Interface) uses the same underlying circuits as T1s and E1s. Over a T1, 23 speech channels are offered, while

an E1 provides 30. One or two of the channels are reserved for signaling communications. This out-of-band protocol transmission allows transfer of information such as calling number, codec type, clearing causes, and such. (Strangely, though, T1 sends caller ID data via modem encoding it into the speech channel.) The speech paths are called *B* (*bearer*) channels, while the signaling is carried in *D* (*data*) channels. Almost all large VoIP gateways and PBXs support ISDN-PRI lines. The signaling in the United States is a slightly different protocol than that used in Europe and other parts of the world. Your gateway will need to be set to match the protocol on your line. Normally in the United States this would be NI-1 (National ISDN-1), while Europe would use the Euro ISDN standard.

You might hear the term *QSIG* in the context of ISDN-PRI gateways. This is a signaling protocol that is yet more sophisticated than ISDN's usual Q.931 protocol, and is layered on top of it. With the ascendancy of SIP, QSIG looks to be yet another valiant attempt falling by the wayside.

ISDN-BRI

ISDN-BRI (Basic Rate Interface) lines offer two B channels, supported by one D channel. (As noted above, B channels carry audio payload "speech", while D channels carry signaling; this is sometimes referred to as a "2B+D" configuration.) These were intended as a residential replacement for POTS lines or for small businesses. One application envisaged by its inventors was to allow a simultaneous voice call and data connection. With DSL providing much higher data rates, ISDN-BRIs are moving ever closer to obsolescence. Nevertheless, when they are available, they could be useful for small installations that need only a few lines. Most VoIP gateways have cards to interface with these lines. As with PRI lines, take care to set the gateway's configuration to match the type of signaling that your line uses.

A ROSE BY ANY OTHER NAME The word *line* is becoming troublesome. Back when a POTS line was associated with a single telephone number, the word had a clear meaning. When you ordered 10 lines, you received 10 physical pairs and had 10 telephone numbers. (Well, okay, rollover rotary service exposed only one number to the public, but the others were still there in the background.) ISDN was the first step on the road to a verbal ghost town. We engineers began to speak of a BRI "line" with two voice channels. But radio producers and hosts still communicated with each other as if channels were lines, saying things like, "The tree-hugger is on line 3. Do you want to take him now?" Station *engineers* knew that meant BRI line 2, channel 1, but operational staff typically had no clue. Now we are faced with SIP trunking and other IP-based services, where a single "line" (which might be connected via one, two, or more physical copper pairs; an optical cable; or a wireless link) can carry any number of voice channels. Operators, blissfully unaware of all this, will undoubtedly continue with their conditioned habit, referring to a certain caller as being on a particular line. Thus, is the word *line* destined to join *dial* in a peculiar departure from original meaning?

6.9 USING VoIP TO CONNECT TO THE TELCO NETWORK

While it remains a niche at this writing, SIP trunking is growing in support from both PBX vendors and carriers. Over time, this will almost certainly appreciably reduce the use of the older POTS and T1 trunking. Eventually, it may eliminate the need for it completely.

Whether the gateway to the PSTN is at your physical location or at another site should make no difference, as long as the IP path between you and the gateway has guaranteed QoS with sufficient bandwidth to support the maximum number of active connections you expect to have. In the case that the IP link is to be used for both telephony and data, the system must be designed so that phone calls have priority. In order to ensure this, there must be only one IP service vendor between you and the PSTN, and this vendor must guarantee QoS in a properly written service level agreement. Any time that IP service crosses from one vendor to another, all bets are off as to both the probability of achieving consistent good quality and having any chance of getting problems resolved.

The other thing to look out for is what codec will be used. For calls that ultimately are carried by the PSTN, only the native G.711 codec is acceptable for studio applications. Anything else would involve transcoding and an unacceptable reduction in fidelity. This would be especially audible when mobile phone calls are involved. These already have poor quality due to their low-rate 14.4-kbps codec. Passing those calls through G.711 within the PSTN and then yet another codec on the way to your studio over an IP link is asking for aural trouble.

Finally, you need to be sure that your equipment and the carrier's gear can properly communicate. While SIP is a standard, vendors often enhance it with extensions that are not universally supported.

One development that could help is a project called SIPconnect, undertaken by SIP Forum, a consortium of SIP vendors. The SIPconnect Interface Specification was launched by Cbeyond Communications in 2004, with support from Avaya, BroadSoft, Centrepoint Technologies, Cisco, and Mitel. It attempts to detail the interconnection specifications between IP PBXs and VoIP service provider networks. It specifies a reference architecture, required protocols and features, and implementation rules. It calls for the G.711 codec to be provided on all equipment and services.

IP + PSTN = QoS One of the ironies of the current state of IP telephony is that with care, calls that pass through the PSTN can have guaranteed quality of service, while those that stay within the IP cloud usually don't. Within a single provider's network, all is usually well, but as soon as a call crosses multiple vendors' networks, QoS evaporates. This is because the Internet exchange points (IXPs), where the different ISPs' networks interconnect with each other, are regularly overloaded, causing packet loss and jitter. When a call transits via the PSTN with a single-vendor IP connection at each end (though they may be different vendors), IXPs are replaced by the PSTN. In effect, the PSTN serves as a peculiar kind of IXP. This solves the QoS problem, but introduces another one: enforced low-fidelity due to the PSTN's G.711 codec and its 3.4-kHz audio bandwidth limit.

Would it make sense for a radio station or audio production facility to change their telephone service to SIP-based IP? As this is being written, the answer is not clear. There is no inherent reason that properly engineered IP trunks would provide anything other than a reliable high-quality service. In practice, *caveat emptor*.

6.9.1 MPLS

Multiprotocol label switching is an emerging IP service aimed at customers that need guaranteed QoS, such as for VoIP. MPLS works by prefixing packets with an MPLS header, which contains one or more "labels," called a *label stack*. These MPLS-labeled packets are switched after an efficient label lookup/switch instead of a lookup into the IP routing table.

MPLS enables class of service (CoS) tagging and prioritization of network traffic, so administrators can specify which applications should move across the network ahead of others. This function makes an MPLS network useful to firms that need to ensure the performance of low-latency applications such as VoIP. Carriers supporting MPLS differ on the number of classes of service they offer and in how these CoS tiers are priced.

One of the promises of MPLS is that it can cross vendor boundaries, eventually offering QoS to voice applications in a manner similar to the PSTN.

6.9.2 IP Centrex and Hosted PBX Services

Just as it shouldn't matter whether the PSTN gateway is on your premises or not, it also shouldn't matter where your IP PBX is located. This is the principle that enables IP Centrex services or hosted PBX services. These locate the hardware at the service provider's site and remove the need for phone system equipment at your location. In a full-fledged installation of this type, you would have only IP phones at your site, which would be plugged into an Ethernet switch, which would connect to the Internet via a router. The main advantage is that someone else is responsible for installation and maintenance of the backend equipment. It might also be that a vendor of these services has invented a suite of applications that would be difficult to replicate at individual business sites.

THE REAL WORLD: SIP TRUNKING ORDERING EXPERIMENTS

We tried to order an SIP trunking service at a few representative sites around the world, wanting to see if the idea was practical from both technology and cost perspectives. Following is what we learned in mid-2009.

Cleveland, Ohio

AT&T suggested we install its service called IP Flexible Reach. AT&T provides a T1 line that can be used for both telephony and Internet access. G.729a/b, G.726, and G.711 codecs are supported. One codec is specified at the time of ordering the service and is used for all calls.

AT&T claims you can expect 17 simultaneous calls with G.711 and 50 with G.729. A "managed router" (from the Tenor family, made by Quintum) is installed by AT&T on-site, which handles prioritization between the voice and data traffic. (It can also convert to POTS, T1, or PRI ISDN, should your equipment require it.) The AT&T technician we talked with said that quality should be perfectly good for on-air applications since all traffic stays within AT&T's QoS-controlled network before it hits the PSTN. The cost was quoted as $597 for the data circuit plus $60 per pair of numbers per month. AT&T bundles 300 minutes of long distance per number/month, with $.04 per minute for anything over. QoS is guaranteed in a service level agreement.

XO Communications is another firm that offers SIP trunking. It proposed installing a line with a 1.5-Mbps rate, supporting up to 16 G.711 calls. Bandwidth not being used for VoIP could provide Internet service. The company provides the on-site router and claims to manage the circuit all the way through to the PSTN. The cost was $490/month, which includes the line, PSTN numbers, free local calls, and 2000 U.S. long-distance minutes. Additional minutes were quoted at $.04 each. There are bigger pipes available, up to 10 Mbps, which could support more VoIP channels or more Internet bandwidth.

Munich, Germany

The firm Arcor is an ISP that offers SIP trunking service. It sells a 4-Mbps DSL line for €239/month ($338) plus €79 ($112) for the QoS guarantee required for high-quality VoIP. This line supports a maximum of 27 calls using the G.711 codec. Bandwidth that is not being used for VoIP can be diverted to general Internet access. Each number into the PSTN costs €2.95 ($4.20), and there is the usual per-minute charge for both local and international calls.

The company also offers a hosted IP PBX service. The customer buys only SIP phones and registers them with Arcor's SIP server. The price is €9.95 ($14) per user per month, including a flat rate for calls within Germany.

Riga, Latvia

Telos has a regional office in this Baltic country, so we tried here, too. Lattelekom is the main telco. The company told us it offers a hosted PBX IP service from a Siemens Hi-Path switch located in its central office. A business-class QoS-guaranteed DSL or optical connection that would support 25 voice channels was around €80 ($113) per month. Each phone port cost about €10 ($14) per month. Lattelekom said the service was popular with companies such as banks that have a lot of branch offices. We were told the system could possibly be used for SIP trunking, but that it had so far only been offered for use with "certified" phones, presumably meaning from Siemens.

The company explained that it would soon be installing an Alcatel/Lucent IMS switch, which would eventually replace its TDM equipment. When this was ready, SIP trunking would, in fact, be the routine way for clients to connect. This included even simple home subscribers, who would get a DSL box with both Ethernet and POTS ports. Cost was expected to be the same as existing POTS and ISDN service, around €10 ($14) per month per voice channel plus the cost of the DSL or optical line. Interestingly, the Lattelekom people told us that they had decided on a policy of not investing in any more TDM equipment because the "tech is now obsolete." The company intends to transition the entire network to IP over the next few years.

THE REAL WORLD: ADVENTURES IN VoIP AT COMMERCIAL RECORDING

Commercial Recording Studios is an audio recording and video production facility that serves the advertising community in Cleveland, Ohio. The facility includes four primary audio production suites, two Final Cut video editing suites, and a battery of HD equipment for video shooting.

Dan Bays is the director of engineering. He tells the story of their experience with VoIP telephone service and how it has evolved over the years.

Twelve years ago, we began a service to deliver spots to radio stations via the Internet. We developed the software for this process and installed a system with automatic redundancy in our web servers, power systems, and connections to the Internet. In order to help defray some of the costs of these systems and Internet connections, we also began hosting web sites and email for a number of other companies around Cleveland. So we had developed a lot of experience in IT and IP.

In 2003, we noticed that we had a cutting-edge facility, except our phone equipment. We were still running a 20-year-old Merlin system with 25 phones. We were paying for 10 analog phone lines and 6 ISDN lines. The ISDN lines were used with codecs for sessions with other studios across the country. Four of the analog phone lines were dedicated to servers that sent faxes to radio stations as part of our delivery system. We had another four analog phone lines for a radio call-in show that a client was doing from our facility. We knew we needed to take our phone system into the 21st century and thought this would be an opportunity to reduce our line charges as well. We believed IP telephony was the way of the future, so we began discussions with our telco, which proposed a Cisco IP phone system.

The telco sold us on the ability to roll all of our analog phone lines and the ISDNs into a single T1 that would be terminated in the Cisco. This would give us big savings on line charges, which theoretically would pay for the system over a couple of years. When the install took place, we discovered that the telco really did not have a way of moving the ISDN lines into the T1 bundle and making them work with the codecs. Another challenge to the expected savings was when we realized Cisco wanted $1000+ per year for a service contract to keep the system up to date. We chose to reject this after the first year. It was a phone system. We needed to be able to make calls and receive calls. If our old Merlin system lasted 20 years without significant upgrade, why did we need anything more for a newer system? The new system was great, but we were not enjoying the savings we expected.

By early 2008, we realized that we were paying for a full T1 that provided 24 simultaneous call paths, but we were never using more than 6 to 8 at any time. We also had our dual T1s into the Internet, giving us a 3-Mbps data pipe. Most of the engineers in our facility were using Vonage at home and had experienced the cost benefits of VoIP over the Internet. Our VP started asking whether there was a way to use this technology for our business phone system. In addition, we had a producer who spent part of the year living in Italy and was looking for a way to have calls to/from the Cleveland

area without the crazy international phone charges. I set up a VPN into our phone system and tried using one of our IP phones to work across it. We tested it from several of our engineers' houses in the Cleveland area, and it worked perfectly. The remote phone became an extension on the system just as if it were inside our building. Unfortunately, when our producer took it to Italy, we discovered problems with his ISP were preventing the VPN from working properly. Seems they were blocking something and we were unable to determine the cause of the problem. Despite the Italy problem, we were convinced that IP was the way to go and we started investigating it to replace the T1.

It happens that we hired a software programmer around this time whose previous job was installing VoIP systems. He wanted to try installing the Asterisk open-source phone server, connected to an Internet SIP provider. We built a test server and migrated one of the Cisco phones over to SIP. It appeared to work well. I spent a lot of time implementing and testing QoS on our network so that all phone traffic was DSCP tagged, and being sure those tags were passed and honored through switches, firewall, and the router to the Internet. Our Internet provider had no QoS, so once traffic left our place, QoS would not be guaranteed, but at least internally VoIP traffic was protected. In the spring of 2008, we spent a weekend migrating all of our Cisco phones over to SIP software, and getting them to run on an Asterisk Trixbox. Our SIP provider was Binfone out of New York. Then we moved all of our phone numbers over and had SBC shut off our old phone T1. Our phone traffic was now running via IP on the same 3-Mbps data link as our Internet traffic.

This worked remarkably well. Over the course of a week we might have one or two bad calls, but across the board things were good. We were saving not only the $700 T1 charge, but were seeing hugely reduced charges for local and long distance minutes. We were convinced that we had a winning solution. We did have to bring in a few analog phone lines for faxes because they are not reliable over IP for some reason.

Then in the summer of 2008, the contract for our data T1s came up for renewal. By this time, our telco had purchased a major long-distance carrier. Our sales rep informed us that they were phasing out the old equipment and that we would get better performance and reduced latency if we let them put in a new pair of T1s directly to the major carrier's backbone.

Upon cutting over to the new lines, we had big problems. General Internet data was fine, but our call quality was terrible. At least 25 percent of our calls experienced garbled sound, dropouts, and doubling. It was clear there was some fundamental flaw in running VoIP over the new service. In fact, in the time before our old T1s were taken offline, we could move our phone server between the old lines and the new lines and hear the change in quality. For the next six months, I spent hundreds of hours testing, troubleshooting, fine-tuning, and working through layers of tech support trying to solve this problem. We got copper pairs fixed, reserved bandwidth on our router for VoIP traffic, paid to have QoS installed on our T1 on the telco's side, had Binfone do latency and jitter testing on the network between them and us, had the telco do "intrusive"

testing on our lines, reduced the bitrate to 16 kbps per call, etc. Nothing helped, and each day there would be a new tech support guy from the phone company who was unfamiliar with the history, so back to square-one. Nobody could explain why the old line worked, but the new one didn't. Our staff started using their mobile phones for client conversations because they didn't trust the installed system.

We considered going with a SIP service from AT&T where they control the network through to the point where it connects into the PSTN, but that would have required a dedicated T1 and would have been expensive, essentially eliminating the cost savings.

Finally, as a last resort before plugging POTS lines into our nice new SIP phone server, we asked our cable TV provider to drop us a business-class cable modem for $150 a month. We purchased 2 Mbps down and 768 kbps up. Then we moved only our phone system to it—and all of the problems went away! The phones have been wonderful since. Even the phone in Italy now works.

The moral of the story:

- *VoIP over the Internet is inexpensive and actually works well—when you have a good ISP. Be prepared to switch providers if you are not happy on the first go. Chances are you can find a provider that will make it work. It's much less painful and time intensive to switch providers than to try to get one to troubleshoot a problem.*
- *VoIP should be fine when one vendor owns the path all the way into the PSTN, but we haven't tried that because it pretty much wipes out the savings.*
- *We've never had a problem with the part of the system that runs on the in-house LAN. The IP PBX concept is fine. The problems only occurred when we went over the "bleeding edge" trying to use VoIP over a WAN and the Internet.*

6.10 SKYPE

Because Skype is a popular VoIP service provider, and its system works pretty well, we should say a few words about it here. Skype's technology is a bit of an enigma. We know this much: It is certainly not SIP based, so Skype will not interoperate with other VoIP applications (although it probably uses SIP internally for its SkypeOut and SkypeIn interfaces to PSTN gateways). For a time it was using the iSAC codec from the company Global IP Sound (now Global IP Solutions), then was using an in-house-developed codec with the name SVOPC (Sinusoidal Voice Over Packet Codec). That's a wideband codec with a 16-kHz sampling rate, and thus around 7-kHz audio bandwidth. A new codec called SILK was introduced in early 2009 in the Skype 4.0 release. It has two modes: 16-kHz sample rate with 8-kHz audio bandwidth, and 24-kHz sample rate with 12-kHz audio bandwidth. It is apparently able to shift between the two modes depending on network conditions.

Audio streams are encrypted and do not use RTP. Indeed, it seems Skype attempts to obfuscate its streams, perhaps in order to keep firewalls from discovering their presence.

Skype was developed by a group of engineers in Estonia who had developed the KaZaA peer-to-peer file-sharing system. Presumably, Skype uses some technology that was invented during that time. For example, it is generally believed that the user database is stored in a distributed fashion within users' computers, rather than in a central database.

One of Skype's interesting features is its ability to circumvent firewalls and NATs, a topic we cover in Chapter 7. Apparently, Skype does this in a particularly stealthy way, which is effective across a wide variety of conditions, but that gives pause to corporate IP managers concerned about security.

Should Skype's popularity continue, radio broadcast studio systems will have to find a way to elegantly interface to it, perhaps using some kind of server acting as a gateway. (Indeed, several U.S. television talk shows and news services are already using Skype videoconferencing to backhaul some of their remote guests and reporters.) Sure, you could put a PC running Skype in the studio and feed the sound card output to the audio mixer, but how would calls get screened by a producer? And do you really want yet another PC, display, and keyboard in the studio just to take Skype calls?

6.11 STUDIO ON-AIR SYSTEMS

Here is where the icing meets the cake. With an IP audio studio infrastructure, an IP-based on-air telephone system is a sweet topping. Remember the radio station scenarios described at the start of this chapter? As you can now appreciate, passing calls between your office PBX and on-air system is easily accomplished with IP-based systems. Need to share telco service across both your business offices and studios? Need to share lines among studios? Check, and check. But you profit in many other ways:

- A single on-air phone system server can supply all the studios in your facility with rich telephone capability.
- A common wiring and Ethernet switch infrastructure serves both your studio audio and telecom needs.
- On-air call director controllers can be sophisticated devices owing to their connection over IP.
- Call-screening software running on PCs connect over the same network, and can include integrated soft phones, thus smoothing operations and saving the money that would otherwise be spent on hardware phones.
- Mixing-console control surfaces can incorporate phone system controllers that need no additional connection; their signaling just rides on the network

connection already there. Rich status information can be displayed either on the phone control module or the console's main screen.

- Recording and playback of DJ plus telephone conversations are simplified. PC-based editors send and receive audio directly over the network using their native Ethernet connections.

Since IP-based studio telephone systems run on LANs, there is no concern with QoS. We can be confident that all packets will arrive, and that they will do so quickly and with very little jitter. We have plenty of bandwidth at nearly zero cost. (Were this not the case, we wouldn't be running our studio-grade audio over them, and you wouldn't be reading this book.) Thus, in studio/LAN applications we experience all the benefits of IP with none of the trouble. IP gives us a low-cost and universal way to connect gear from multiple vendors. It lets us take advantage of the economies of scale that result from plugging into the IT world. It lets PCs talk using their native language. And it provides a rich communication path for speech at a range of quality levels alongside all the control data we need to support advanced user interfaces. Via IP, studio telephone systems interface smoothly with telco services and PBXs on the one side, and IP-based studio equipment on the other.

Because VoIP is able to benefit from high-quality codecs, which can be automatically selected on a call-by-call basis, there is a good chance that the future will bring us much higher-fidelity on-air calls. Even mobile phone calls might get a lot better if the AMR-WB codec starts getting traction. And even without wideband codecs, the pure digital interface that IP offers between the studio audio and telephone systems helps to improve quality.

6.11.1 Line Echo Cancellation

This is the classic function performed by broadcast digital hybrids. When POTS lines are used in studios, the send and receive audio need to be isolated as much as possible. In a studio application, a hybrid interface needs particularly good send-to-receive isolation. When too much of the send audio leaks through the hybrid and appears in the receive audio signal fed to the telephone input on the mixing console, there will be a number of unwanted effects:

- Distortion of the host's voice. The telephone line will change the phase of the send audio before it returns, with varying shifts at different frequencies. The host audio will suffer degradation as the original and leakage audio are mixed at the console and combine in- and out-of-phase at various frequencies. When this occurs, the announcer sounds either hollow or tinny as the phase addition and cancellation affects different frequencies.
- Audio feedback can result from the acoustic coupling created when callers must be heard in the studio on an open loudspeaker.

- When lines are conferenced and the gain around the loop of the multiple hybrids is greater than unity, feedback singing will be audible.
- If the leakage is very high, operators will not be able to control the relative levels of the local host audio and the caller since the console telephone fader will affect both signals.

As we touched on earlier, these impairments can occur even when a digital telco line is being used, owing to coupling at the far end. IP to PSTN gateways or the equivalent function within an IP PBX should always have a line echo canceller (LEC) as part of its suite of adaptation functions. But it is not always the case that the LEC rises to "broadcast quality." For that reason, an on-air system attached to a VoIP system may need to have an additional "helper" LEC.

6.11.2 Audio Processing

The digital hybrid interfaces employed in studios often include audio-processing functions, which may include some or all of the following:

- AGC, on both the input (studio audio send) and output (telephone receive audio) paths.
- Audio response shaping on the send audio to improve intelligibility. Without filtering, high-fidelity studio microphones put too much low-frequency energy into the telephone line.
- Automatic multiband EQ on the telephone audio to compensate for the wide variety of telephone sets in the field, as well as effects from different codecs and other impairments in speech paths.
- A filter to remove low-frequency hum and noise on the telephone audio.
- A ducker to dynamically lower the volume of the telephone audio when the host speaks. This serves both an aesthetic and a technical purpose. As to the first, many talk hosts prefer to have control over the conversation and the ducking helps them to achieve that. As to the technical benefit, a ducker improves the effective, or apparent, send–receive isolation, compensating for deficiencies in the performance of the core hybrid.

6.11.3 Acoustic Echo Cancellation

A common annoyance in radio studio operations is the feedback that results from using a loudspeaker in the studio to listen to telephone calls (typically done to avoid talk show guests in the studio from having to wear headphones). This comes from the acoustic coupling of the audio from the loudspeaker into the studio microphone. Ducking helps by reducing the gain "around the loop." But it compromises full-duplex operation and can make it difficult for the caller to hear the host. We just said that the ducker is often used for aesthetic or production reasons, but

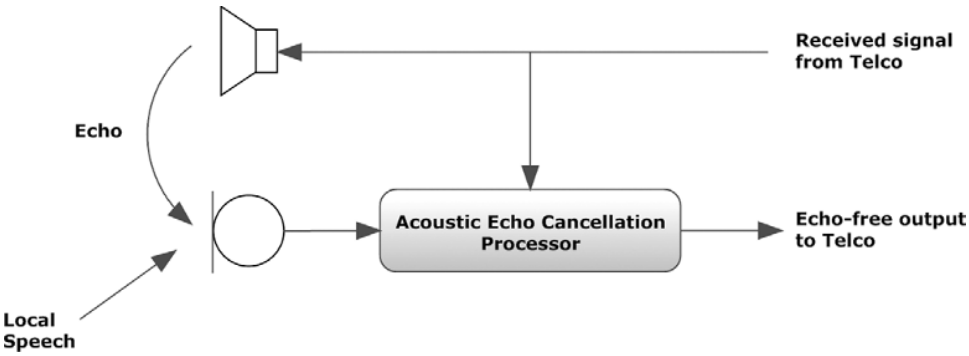


FIGURE 6.16

An AEC is required for smooth full-duplex conversation when a loudspeaker is used for hearing telephone audio in the studio.

this only requires gain reduction in the *receive* path. In order for a ducker to help with feedback, it also would need complementary gain reduction in the send path. There is also the fundamental problem that any echo from the studio would be heard by the caller. When the caller talks, his voice bounces around the studio and gets sent back. Due to the time dispersion caused by the room reflections and the roundtrip transmission delay, this acoustic echo is very distracting to the caller.

Acoustic echo canceling (AEC) is the answer (Figure 6.16). The audio at the studio microphone consists of the host's voice combined with the unwanted telephone audio that is delivered to the room via the loudspeaker. AEC removes the loudspeaker audio, leaving only the host in the send audio returned to the caller. AEC has been used in high-end audio and video conferencing systems for many years. High-end broadcast hybrids and on-air systems such as the Telos Delta, 2 × 12, and N × 12 have included a limited form of AEC. But only recently has AEC technology advanced to the stage where it is truly effective. Thankfully, it comes just when the additional delay of mobile and VoIP connections make it nearly essential. The good news has been a result of both breakthroughs in the design of AEC algorithms and the ever-increasing power and lower cost of processor chips.

These latest-generation cancellers are a miracle. You can have callers amplified to ear-splitting volume on the speakers in the studio, and very little of the caller audio makes it back to the other end. AECs work with up to 20-kHz audio bandwidth, so they are ready for the wideband VoIP codecs now coming online. And they solve a longstanding problem: Older "time-domain" AECs depended on the acoustic path remaining fixed, and could quickly degenerate into feedback when a microphone was slightly moved. The "frequency-domain" technology used by the new generation of AEC equipment can dynamically adapt to moving microphones and other echo path changes.

This new AEC technology is particularly useful for TV studio applications where it can be impractical to have talk show guests using headphones or even earbuds. These programs also like to use roving hosts with handheld microphones. Today's high-performance AEC technology lets talent move around and allows everyone in the room to listen to phone calls on foldback loudspeakers.

6.11.4 Application Example: Telos VX System

The Telos VX studio telephone system takes full advantage of IP to deliver clear benefits to broadcast studio operations (Figure 6.17). It is a compelling example of IP's ability to support a variety of attractive features while lowering costs. It also simplifies the technical side of studio telephone interfacing, perhaps making the harried engineer's life a bit easier.

The system consists of the following components:

- A main server that interfaces to telco lines provides SIP signaling, performs line-state control, takes care of audio switching, supplies a suite of audio-processing functions, and supports studio controllers.
- Desktop controllers: These are similar to IP phones, but designed specifically for studio use.
- Drop-in console controllers: These communicate via IP to the main server.
- Assistant Producer VX software: This is a Windows PC application (including a soft phone) that producers can use to screen calls, record calls for later play on-air, assign caller priorities and share them with on-air talent, associate notes with calls that can be read by the talent, and exchange instant messages.

A single server can be a foundation for a large multistudio facility that needs dozens of telephone lines. A single Ethernet RJ-45 carries Livewire audio and control to all of the mixing consoles and controllers that need to access the telephone system. Another RJ-45 transports SIP control and telephone audio. (All can be combined on a single jack with a configuration option, but the two-jack approach creates a rock-solid firewall between the studio and telephone systems, should that be required.)

When a SIP PBX or gateway is already installed for the business side of the facility, the VX can connect to that as well, allowing lines to be shared between the two systems. If there is no SIP interface in place, a gateway can be employed to join the studio system to POTS, T1/E1, or ISDN PRI/BRI telco lines. The brave may consider a direct connection to telco SIP trunking services.

The system provides a simple set of basic IVR (interactive voice response) features. An audio message with a "legal notice" can be played to callers automatically. Callers can have the option to leave a message rather than talking live. The message files can then be accessed for editing a subsequent play on-air. (This one feature could change the character of call-in shows. Busy people often don't want to wait for their turn to be in conversation with a talk host, so they don't bother to call. Giving callers a "leave a message" option could well expand the range of callers to shows, adding those who would otherwise not call.)

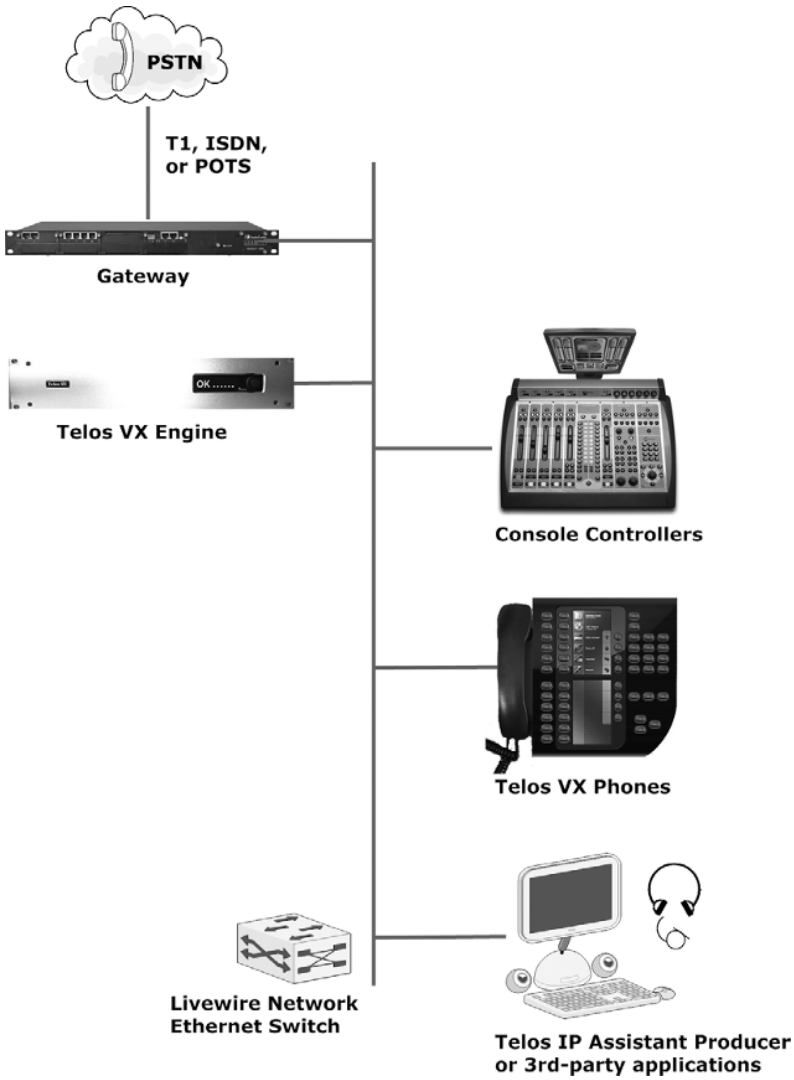


FIGURE 6.17

A studio on-air system can be easily integrated into AoIP and VoIP infrastructures, smoothly bridging the pro-audio and telephony worlds. All telephone, audio, and control is via IP/Ethernet. A gateway interfaces to the PSTN. Controllers can be hardware directors, console drop-in modules, or PC-based soft phones.

Because of the vast processing power of modern CPUs, audio processing is richer and of higher quality than previously possible. The usual suite of processing familiar to users of hybrids is provided: AGC, ducking, and auto EQ. But there is also a highly effective AEC per studio. Air talent will be surprised at how loud they can crank the preview speaker and not worry about feedback. TV producers are going to love

getting enough volume on studio foldback monitors, and not having to bother with earphones (for listening to phone calls, anyway). Finally, in this system, there is a LEC (digital hybrid) for each caller, permitting feedback-free and clear coupling between callers during multiparty conferencing.

Because IP is not restricted to the G.711 (3.4-kHz bandwidth) codec, the VX system is ready for wideband higher-fidelity calling. Imagine the day when calls from AMR-WB-equipped mobile phones sound *better* than today's landline calls!

The system also allows operators to select and distribute phone line outputs to up to eight faders for easy management of conferencing, with independent control of levels. In addition, "VIP" lines (i.e., lines reserved for interview guests or remote talent) can be dedicated to faders, bypassing the selector buttons.

IP-connected consoles can have a rich interface to the VX system (Figure 6.18.) Line selectors can be tightly integrated. For VIP lines assigned directly to faders,

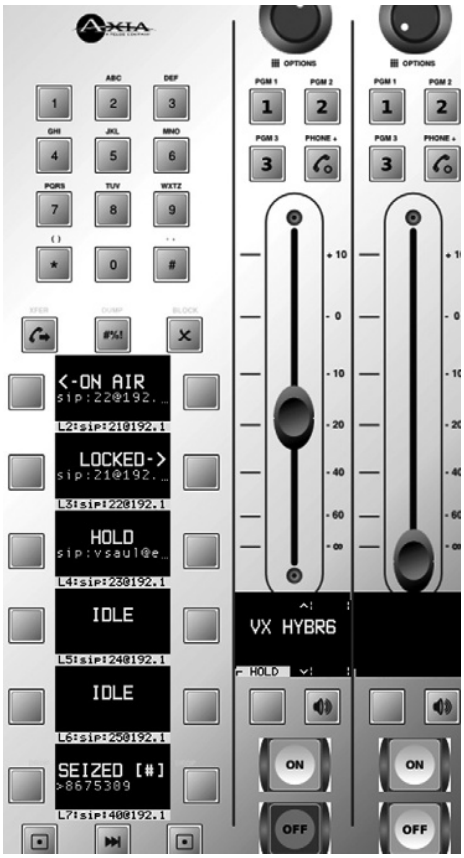


FIGURE 6.18

Axia iQ mixing console telephone control module. The IP network offers tight integration with a single RJ-45 connection.

icons can show ringing, held, etc. line status. Caller ID can be displayed on the console's main screen.

Owing to IP's ability to transport high-volume data along with the audio, VX Directors make a significant leap over previous generation controllers (Figure 6.19). In the premium Director, two large, color LCDs display not only line status in a more nuanced way than was before possible, but also any notes entered by a producer, how long a call has been waiting, etc. With its rich interface, the new IP Director gives talent a straightforward way to select and assign calls to faders.

Another major improvement is the Assistant Producer VX software application (Figure 6.20). It provides the usual call-screening functions for call-in talk shows, but with a number of enhancements enabled by the IP nature of the system. The integrated soft phone uncomplicates the producer's life, since the PC interface is used for all operations, including answering and making calls, assigning priority, writing notes, etc. It also reduces cost, since no multiline hardware phone needs to be installed.

Because Livewire audio is available at the PC via the Livewire Driver software, the producer can readily record calls for later play. These can be edited with a PC application running on the same machine. When a file has been produced, it can be sent to the on-air studio over the network. Thus, the one Ethernet cable is used for all of the following:

FIGURE 6.19

Telos VX Desktop Director.

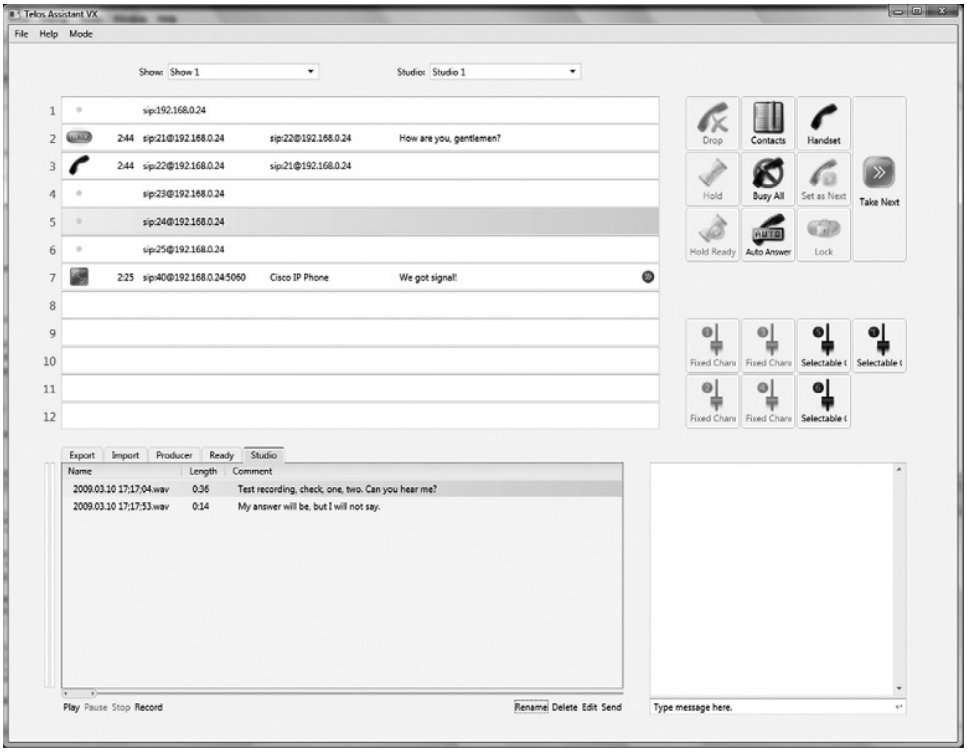


FIGURE 6.20

Telos Assistant Producer VX PC application. It provides the usual call management and communication between the talent and producer, but also features an integrated recorder, player, editor, audio file manager, and soft phone. All audio and control is via the network.

- Telephone audio for the soft phone.
- Livewire audio for the recording of calls.
- Transfer of recorded call files from the producer to the studio.
- Data messages between the PC and the VX main box for line selection, etc.
- Data messages such as call notes and IM between the producer and on-air studio.
- Database lookup of caller information, such as how many times they have called, the quality of their contribution, whether they have won any contests, etc.
- Web browsing, email, etc.

This is the power of IP made real. To accomplish this level of functionality with older technologies would have been impractical.

6.12 APPLICATION EXAMPLES: TELOS Nx6 AND Nx12

If a facility doesn't need the sophistication or multistudio capability of the Telos VX system, smaller Nx6 or Nx12 phone interfaces also have a Livewire interface. They are intended for single-studio application, and include on-board connections for either POTS or ISDN BRI telco lines.

They support both Livewire audio and control over IP, so are also a good fit to IP-based studio installations. Indeed, there is a version of the Nx6 aimed at Livewire-equipped studios, which has no connections for traditional analog or AES audio, saving the expense of these when they are not needed.

6.13 TRANSFORMATIVE TECH

When the connection from a station's listeners to its studios eventually evolves to become IP based from end-to-end, as it is nearly certain to do, things could change appreciably. We've already touched on the possibility for higher fidelity. But that's only the immediate and obvious next step. With an unconstrained pathway for data along with voice, a talk show's producer or host could text chat with a caller prior to his or her being on-air, for example. There could be an automatic updating of the time a caller is planned to be on-air. There could be instant voting. A window could be kept open on a listeners' PC or smart phone that would deliver supplementary text or visual information. This has the potential to revolutionize the nature of the relationship between listeners and programs, making the connection more transparent and open—closer to the style of Internet applications such as Facebook that supply users the appeal of rich interaction.

Peering yet further into the future, should video streaming catch on, we might want both to see and hear callers via an Internet link.

The first stage of the application of any new technology is to replicate the function of what came before, but once the new platform is in place, creative people invent new and unexpected ways to use it. IP is a powerful and amazing enabler that has already showed us plenty of surprising things; it's inevitable that more is on the way.

CHAPTER

IP Codecs

7

This book is dedicated primarily to studio-grade audio systems on LANs. We did take a detour in Chapter 6 to VoIP telephony, but it was in the context of interfacing IP studio systems to the outside world, which almost always involves some kind of telephone connection. Having edged out of the studio already, however, we've decided to keep going—a little bit further, anyway. So in this chapter, we take a look at IP codecs. These are devices and systems that are used for remote broadcasts, interstudio links, or to construct audio networks spanning large distances. Some are close cousins to VoIP phones and systems, using signaling and IP audio transport in a nearly identical fashion. What sets them apart is that they are able to achieve broadcast audio quality. The most sophisticated are optimized for reliable operation over the less-than-perfect conditions of public Internet and mobile phone links.

First, let's consider how we got to where we are today. Most readers will recall how ISDN has served broadcasters well for remote backhaul applications. Indeed, it was a small-scale revolution when it first appeared in the early 1990s. At long last, the digital nature of the telephone network was revealed to mere mortal end users. For the first time, the dial-up network could be used for high-fidelity remotes. Compared to the equalized analog "broadcast loops," which were the only high-fidelity telephone service provided previously, ISDN was a wonder. While ISDN is still a perfectly good technology, it does have some drawbacks, however. The main one is that usage is billed by the minute. Another is that installation of the line at the remote side usually has a multiweek lead time and incurs a significant setup charge. As we pointed out in Chapter 6, telcos had originally envisioned ISDN as the way Internet would be delivered to the masses. But DSL now fills that role. Broadcasters remain one of the few users of ISDN basic-rate service, but they don't provide enough business for telcos to justify the expense of maintaining the infrastructure.

So we now turn to the omnipresent Internet and other IP networks for an alternative. At many remote sites, there is an existing IP connection that can be hijacked for an ad hoc broadcast. Increasingly, high-speed IP links via mobile phone networks also offer the chance to connect from almost anywhere within large cities. International connections are no problem, and the per-minute charge is gone.

Audio Over IP

doi: 10.1016/B978-0-240-81244-1.00007-9

7.1 THE CHALLENGE

On LANs, a packet's life is easy. There's plenty of bandwidth, no significant jitter, and nothing is dropped. Thus, the success of studio-grade AoIP. For VoIP, we've encouraged you to be sure that you have a link with high-grade QoS when you need to leave the secure comfort of your LAN. But broadcast IP codecs are purpose-made for the tougher wide-world.

On the Internet and mobile links, there are no QoS guarantees. To avoid audible defects, equipment for audio transmission over IP networks must be designed to cope with these conditions. Specifically (as Chapter 2 describes), the challenges are as follows:

Jitter. For an IP audio stream, the time-span difference between the earliest arriving packet and the latest (relative to their respectively correct arrival times) is known as jitter. In other words, jitter expresses the error range in arrival times experienced across a given set of packets. On the Internet, this ranges from tens to hundreds of milliseconds. (In contrast, ISDN has no significant jitter.) A long buffer in the receiver, which meets or exceeds the maximum expected jitter time, can be used to even the flow and deliver consistent, continuous audio. But this comes with a cost: Audio is delayed. Since codecs at remotes are often used in two-way fashion, this delay can cause trouble for the talent, who find it difficult to speak naturally. And there is always the possibility that a packet will arrive later than the maximum buffer time. Thus, a fixed buffer has only limited utility.

Packet loss. Dropped packets are a normal condition on the Internet. It was designed so that any router node that becomes overloaded can deliberately drop some percentage of the packets flowing through it. An IP codec needs to address this condition so that audio flows smoothly despite the network's imperfection.

No bandwidth guarantee. Finally, we must cope with the fact that there is no guarantee for bandwidth. That's why IP codecs have efficient audio coding algorithms at their heart. These are carefully designed to transport high-fidelity audio at the lowest possible bitrate.

Yet even a very efficient codec is difficult to optimize for these conditions. If we fix the codec bitrate to some high value to get good fidelity, we might find that the network can't provide enough bandwidth to support that rate, causing audio interruptions. On the other hand, if we decide to be conservative and set the codec bitrate to some lower value so we can be more confident that we avoid dropouts, we sacrifice audio quality. Therefore, even better than an efficient codec is one that can adapt to network conditions dynamically.

The strategies for dealing with these issues depend on what you need the codec to do. In the main, whether your application can stand a lot of delay or needs to maintain low delay will determine the nature of the solution.

7.2 WHEN DELAY DOESN'T MATTER

There are many applications for IP codecs where delay doesn't much matter. An example would be a broadcast of a concert or sports program that doesn't involve interaction. Assuming there is a two-way connection, both the jitter and lost-packet problems can be solved by having a long buffer in the receiver and some kind of retransmission method to recover any lost packets.

TCP provides just such a reliability service. Since it is ubiquitous, it's a natural solution. When a receiver detects a lost packet, it requests the source to resend, so all packets eventually arrive to the audio decoder. But the cost is a multisecond delay, since the receive buffer has to be long enough to accommodate the time it takes for the replacement packet to arrive. You've probably experienced the effect of this buffering because streaming audio on the Internet often uses TCP. You click on a web link or play button, and wait.

TCP's flow control algorithms are also a problem, since they could needlessly restrict bandwidth.

An alternative to TCP is the retransmission method described in RFC 4588, *RTP Retransmission*. Support of the extended RTP profile as defined in RFC 4585 is necessary for implementing this retransmission method. This gives a compromise between the low latency of RTP and the full recovery/reliability offered by TCP. It adds latency because the receive buffer needs to be long enough to allow the recovered packet to be slotted into the stream when it eventually arrives. But unlike TCP's "wait-forever" policy, the latency is limited to some specified time value. And it avoids the problems with TCP's flow and rate control mechanisms. It also has the interesting property that the recovered packets are sent on an independent stream, which means that a receiver can choose to ignore them and just listen to the main RTP stream.

This scheme is not likely to be satisfactory for low-latency bidirectional applications. As the authors of the RFC say: "In the case of multimedia streaming, the user can tolerate an initial latency as part of the session setup and thus an end-to-end delay of several seconds may be acceptable. RTP retransmission as defined in this document is targeted at such applications."

THE INFINITE BUFFER File transfer offers a way to get audio from one place to another over the Internet with no concern whatsoever for jitter and dropped packets. TCP is always used to ensure reliability for such transfers, so the audio data always arrive intact. With some stretch, this can be thought of as an IP codec with a *very* long receive buffer. And as we said, with a long enough buffer, all problems disappear.

7.3 WHEN DELAY DOES MATTER: ACHIEVING LOW DELAY

When delay does matter, retransmission is not going to be useful. A broadcast IP codec intended for bidirectional application needs to be optimized for the purpose. An integrated system that pulls together a suite of appropriate elements is required, as follows.

7.3.1 Audio Coding

The MPEG AAC-family codecs have always had quite good error-concealment techniques, but these had been optimized for the bit errors found on nonpacketized transmission paths. Recent work has expanded the concealment technology so that it can work effectively with packet loss as well. It's a clever technique. The codec keeps an ongoing measure of the spectral shape of the audio. This is easy because the codec already must have a time-to-frequency domain transform as part of its perceptual coding functions. When a packet loss is detected, a synthetic replacement is created by using the spectral values to filter white noise. To the ear, this sounds very much like the original. The amplitude is tuned at each end of the packet to match the preceding and subsequent packets so there is no audible "pop" from the splice. It turns out that this can be very effective, indeed. As much as 20 percent random packet loss can be inaudibly concealed.

MPEG AAC is also an efficient codec, and the addition of spectral band replication (SBR) makes it the most efficient within the MPEG family. AAC with SBR is officially called HE-AAC (high-efficiency AAC), but is also known commercially as AAC+. The downside is it has quite long delay—around 150 ms. That would mean 300 ms for a roundtrip, plus yet more for the IP packetization and buffering processes—too much for interactive two-way conversation. AAC-LD (low delay) comes to the rescue. It has around 50-ms throughput delay, so is much better on that count. But it has 30 percent less bit efficiency than plain AAC. Since SBR adds approximately the same 30 percent in efficiency to AAC, if we could combine that with AAC-LD, we would have a low-delay codec with the coding power of plain AAC. And that is just what the new AAC-ELD (enhanced low delay) codec does. It has reasonably good fidelity down to 24 kbps, and excellent fidelity when used at 64 kbps and above. At 128 kbps, it is regarded as indistinguishable from the original.

AAC-ELD's wide bitrate range is a good match to the needs of IP networks, since they vary so widely. A mobile phone connection might be limited to perhaps 40 kbps, while dedicated links could be sized as desired, supporting codec rates of 256 kbps, 384 kbps, or more. An international Internet connection might support a 64- to 96-kbps rate.

There are other low-delay codecs, most of which come from the VoIP telephony world. Though a few have pretty good fidelity, they are often aimed at speech and don't sound all that good on music. They are, for the most part, based on an analysis of human speech production. In contrast, the MPEG family of codecs exploits

"perceptual" techniques, which are based on how the ear perceives general sounds. Naturally, these perform better with nonspeech audio material such as music or speech mixed with background sounds. There are some hybrid codecs that have a blend of both.

Overall, there are plenty of interesting codec options out there (see Table 6.1 in Chapter 6 for a taste), but none rival the MPEG AAC family's combination of good performance on general audio material and wide implementation across many vendors' products.

7.3.2 Transport

We have to accept the network as it is and apply a strategy that gets audio reliably from one end to the other. There are three options, only one of which proves to be satisfactory for low-delay interactive applications.

- *TCP* solves the lost packet problem via retransmission, but this imposes a delay penalty, as discussed earlier. We can't use this for low-delay applications.
- *Forward error correction* (FEC) has been proposed as another way to deal with packet loss. The principle is simple: Both the original and some form of copy of the packets are sent on the network. If one is lost, hopefully the copy was not and the receiver can use it as a replacement. The structure of the original-copy sequence is organized to maximize the chances for successful recovery. You don't want to just put the original and a copy adjacent to each other, since that increases the odds that both will be lost. A minimum 2×2 FEC requires the buffering of four packets, while a more reliable 5×5 FEC would require a 25-packet buffer. The latter has more time spread, so is better able to cover losses. But now, unfortunately, we are back to significant delay: In the 5×5 case, as much as 600 ms with the typical packet size. As well, FECs cause streams to occupy more bandwidth, and a network that is losing packets is one that is probably already near its limit, so adding to the bandwidth requirement is just as likely to create a problem as to solve one. There may be some cases where FECs make sense, but they are generally not useful for audio on the public Internet.
- The third option is to use the same RTP and UDP that provide transport to VoIP and Livewire. Combine this with *concealment* and the *advanced adaptive functions* described next, and we have the best possible approach to low-delay, high-fidelity audio over today's public Internet.

7.3.3 Adaptive Receive Buffer

Unless we have a guaranteed QoS network, it is not possible to predict jitter. Each packet is subjected to different network load conditions, which affect its transit time. In fact, each packet could take a different route. For uninterrupted audio, a buffer in the receiver must accommodate the longest packet-transit delay the

network presents. If a packet arrives outside of the buffer time, it's as good as lost. On the other hand, a long buffer translates to a long audio throughput delay, as mentioned previously. So we want to optimize the buffer for the conditions that actually exist. And we want this to vary as needed to adapt to changing network conditions.

But how do we detect the network condition? Recall that TCP adjusts its flow rate when packet loss is detected, so this is a long established way for attached equipment to respond to the network. TCP is constantly probing the network for the fastest supported speed by increasing the rate until loss is detected, then backing off. Although we are not using TCP for audio transport here, we can borrow exactly this idea for our receive buffer adjustment. We start a new connection with an average-length buffer. If packet loss is detected, we expand the buffer length. Unless there is an extreme case, the effect of the lost packets is not heard because the codec conceals it. Meanwhile, the buffer algorithm constantly but slowly pushes to reduce the buffer length. This feedback loop causes the buffer to automatically adjust to the optimum length for current network conditions. On networks with low-jitter connections, the buffer will be made small to minimize delay. But when the jitter is high, the buffer is made long to ensure that there are no audio dropouts.

This buffer-length adjustment requires that time be squeezed or stretched, and this must be accomplished inaudibly. Fortunately, this is possible, as many audio editors with this feature demonstrate. (Most broadcasters will be familiar with profanity delay units that also have time squeeze/stretch processing, and these generally handle a far wider range of time processing than the buffer adjustments we are talking about here.)

7.3.4 Adaptive Codec Bitrate

An important feature of the AAC-ELD codec is that it can be made to "gearshift" its bitrate without making audible glitches. Coupled with a bandwidth-sensing algorithm, a broadcast codec can automatically adapt its bitrate to the available network bandwidth. The algorithm constantly probes the network for the maximum rate that can be carried and sets the codec to this rate. When the network has high capacity, audio is as high fidelity as possible. When the network has limited bandwidth, the codec adjusts to a lower bitrate to ensure that audio gets through.

7.3.5 Putting It Together

Thus, an ideal, low-delay, bidirectional IP codec system would have the following characteristics:

- Effective, inaudible packet-loss concealment. (Retransmission imposes too much delay, so packet loss has to be addressed only with concealment.)
- Adaptive receive buffer, with the necessary time squeeze/stretch capability. (Jitter is changing over time, and the buffer needs to adapt.)

- An efficient codec, to achieve maximum audio fidelity from the lowest bitrate. (The lower the rate, the better the chance for audio getting through, especially on mobile networks.)
- Adaptive codec bitrate, dynamically accommodating network conditions.

This package is a state-of-the-art solution to the challenges posed by difficult IP networks. Working together, these components complement each other to deliver high-quality audio with low delay.

7.4 DELAY AND ECHO

The ISDN we're used to has almost zero delay, so the delay in ISDN codecs comes from the audio coding process itself, not the network. The popular MPEG AAC and Layer 3 (MP3) codecs each have around 150-ms delay. This is too much for the roundtrip, so the usual practice is to use G.722 for the return path, trading off lower delay for lower fidelity. G.722 has around 20-ms delay, so a connection with AAC one way and G.722 the other would have around 170-ms roundtrip delay over ISDN.

The AAC-ELD codec has around 60-ms delay. Used on both directions, that would result in 120-ms roundtrip delay from the coding process. On a good IP network connection, there could be around 50- to 100-ms delay from packetization and buffering, making the total delay around 170–220 ms. That is acceptable for two-way conversation, but it is pushing the limit. It's about the same as the delay in mobile phone conversations, which people have become accustomed to, so we can expect talent to be reasonably satisfied.

Remember that graph in the last chapter (Figure 6.5) that shows how both delay and amplitude contribute to annoyance from echo in telephony? Same story here. In a perfect system, there will be no delayed return of the talent's voice to his or her earphones. Consider the case where talent is on location and an IP codec path is used for backhaul. A local mixer at the remote site sends the talent microphone audio directly to headphones, and a mix-minus at the studio blocks the codec feed from returning, entirely avoiding the codec delay. But there are unintended causes of delayed talent audio making its way back. One is audio leakage from studio headphones at the station end, caused by high monitoring volume and imperfect headphone isolation. Since we know that annoyance from echo is a function of both delay time and amplitude, anything that can reduce the amplitude is going to help. Don't use supra-aural ("open-air") headphones, for example.

7.5 CALL SETUP: SAY HELLO TO SIP (AGAIN)

Broadcast IP codecs can be thought of as special-purpose VoIP phones. We usually want our IP codecs to work like their ISDN equivalents, with a dialing function to find and connect to the destination codec. The same Session Initiation Protocol (SIP)

we met in Chapter 6 in the context of VoIP can be used to provide this service to IP codecs. It is yet another example of how we can profitably hitch onto work done for the IT world.

While it is possible to have SIP connect two units with no other component, it is common to use a SIP server installed somewhere on the network to help get around firewalls, provide "group list" features to related users, and support a relocation service so that a destination can be found regardless of which IP number it is connected to.

Just as for VoIP telephony, SIP serves as a carrier for Session Description Protocol (SDP). This signals between the two ends what codecs are available at each, and allows the system to negotiate the optimum codec among those available. Codec users finally have what they have been waiting for—no need to know or set the coding method before attempting a connection. Just "dial" and let the system figure it out.

We are not obligated to use SIP, however. For nailed-up connections over dedicated links, we could just stick with entering IP numbers directly. The transmitter specifies the receiver's IP address and audio flows. For applications that don't need to use the "call" paradigm, the simple IP address approach works perfectly well.

7.6 SIP SERVERS

A system using SIP requires proxy and registrar servers to function as a practical service. Although two SIP endpoints can communicate without any other SIP infrastructure, this approach wouldn't replicate the dialing-style connection that the public switched telephone network offers. SIP provides a signaling and call setup protocol for IP-based communications that can support a superset of the call processing functions and features present in the PSTN: dialing a number, causing a phone to ring, and hearing ring-back tones or a busy signal.

7.6.1 Firewalls and NATs

As we saw in Chapter 2, when a network address translator (NAT) is in the IP path, all connections must originate from a device on the inside. This means that any codec inside a NAT would be both invisible and unreachable by another codec on the other side. Firewalls have the same effect. What can we do about this?

A simple solution is to put the studio codec directly on the public Internet, making it visible to codecs at remote sites even when they are behind NAT/firewalls. But it would not be possible to make a call *from* the studio *to* the remote codec (unless the latter is also directly on the public Internet, which is often not possible to achieve by the remote crew), and having anything on the Internet without firewall protection is inviting problems anyway. So we need to think about alternatives.

Consider that Web traffic certainly moves both ways past these NAT/firewall devices. This happens because the NAT or firewall is usually "symmetric," meaning

that when a packet stream is sent from the inside toward the outside, the NAT/firewall opens a return path for some period of time. Placed outside any firewalls, a SIP server can both receive and make calls to codecs located inside firewalls. Codecs register with the server, which then takes advantage of the open return path through the NAT/firewall to send an acknowledgment. The server sends additional messages periodically to keep the path open. When one codec wants to connect with another, it contacts the SIP server, rather than the other codec directly. The server knows where to find the other codec and has an open path to it (from prior registration), so the server can signal that a connection is being requested. Each codec can now send messages and audio streams to the other, thereby opening a direct return path through its NAT/firewall that the other can use. (See Figure 7.1.)

With highly restrictive NATs and firewalls, the SIP server can act as a relay for the setup messages. In extreme cases, the server may even have to relay the audio stream.

7.6.2 Telos Z/IP Server

The Telos Z/IP server was designed to support the Z/IP codec. It is similar to a SIP server, but is specialized for broadcast codec application. It provides the following functions:

- *Directory services.* Allows for easy discovery of other devices. Users control the visibility of their devices in the directory. A device may be (1) visible to all, (2) visible only to the group it belongs to, or (3) not visible in the directory. A device always belongs to a group (by default it belongs to the "public" group). The user may create a group at any time as long as the group name is not already in use. By giving others the group password you allow them to

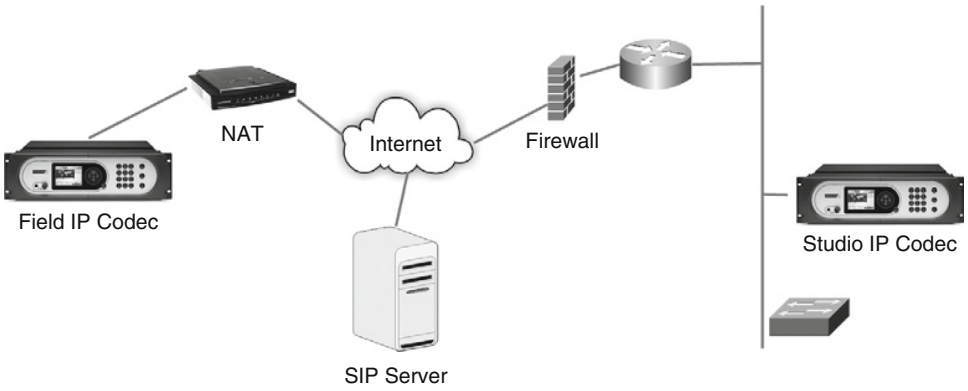


FIGURE 7.1

A SIP server placed on the network outside of firewalls and NAT routers is the secret to IP codecs being able to reach each other.

add their device to the same group. This also allows them to view devices that are visible only to the group.

- *Presence services.* Allows users to view the connection state of their "buddies."
- *NAT traversal services.* Allows a device to discover its public address and NAT type, if any. Keeps a connection open to devices that are not usually reachable behind the NAT to allow incoming calls to get through. For more restrictive NAT types, the server relays the signaling information to assist the connection establishment. For the most restrictive NAT types, the server can also function as a media relay.
- *Geolocation services.* Allows a device to determine its geographic location and that of the destination device, as well as the path taken between them (visual traceroute).
- *QoS data collection.* The server keeps track of call QoS data reported by devices. This information can provide some insight about the packet-drop patterns, bitrates achieved, etc.

SIP servers can be public or private, and the same is possible with the Z/IP server. Telos operates a public server that Z/IP clients can use without having to support their own, or private servers can be installed by Z/IP owners.

7.7 NETWORKS

One benefit of IP is that there are so many ways to use it. From local networks to satellites, you have many options from which to choose.

7.7.1 Public Internet

The Internet has the great twin advantages of near-ubiquity and low cost. One can find an IP connection almost anywhere and simply plug (or wirelessly connect) into it without waiting for installation—or, in many cases, without even paying for it. There are no service guarantees, so you take your chances. But with the adaptive codec technology, the Internet becomes a reasonable proposition for many broadcast applications. There have already been successful remote broadcasts from airplanes. International hookups are as easy as local ones.

A broadcast codec intended to be used over the Internet can benefit from having an integrated traceroute and network-conditions graphing capability, so that you can see the cause of problems when they occur.

7.7.2 Dedicated Links

For studio-to-transmitter links and other high-reliability point-to-point applications, it is often possible to order telco IP links that have guaranteed performance. Packet loss, jitter, delay, and bandwidth are specified in a service level agreement.

7.7.3 MPLS Service

MPLS services are attractive to broadcasters for IP codec application since they offer a good cost/performance compromise—more expensive than nonguaranteed public Internet service, but less costly than dedicated links or ISDN.

7.7.4 Mobile IP Services

Mobile IP services with fast-enough uplink speeds can be used for remotes. In the CDMA world, EVDO Rev A is fast enough for us to use, and is widely deployed in the United States. The uplink has a maximum rate of 1.8 Mbps, but under normal conditions users experience a rate of approximately 500–700 kbps. EVDO Rev B promises yet faster speeds. The bandwidth is shared and there is a possibility of oversubscription and thus packet loss or insufficient bandwidth, but some trials with IP codecs have been successful. In Europe, GSM is not generally fast enough in the uplink direction, though a fast service called High-Speed Uplink Packet Access (HSUPA, also known as Enhanced Uplink [EUL]) is just coming online, offering up to 5.76-Mbps bandwidth. Both EVDO and HSUPA have QoS capability in their technology specifications, but it is not clear if or to what extent mobile providers will pass this to their customers.

Access to these services is usually via a PC card or USB radio device. These are designed to be used with laptop PCs, but can just as well be plugged into broadcast codecs that include the appropriate connector and software. An external box can be used to interface the PC card to the codec via a wired Ethernet connection. Finally, an idea that is growing more prevalent is to use a device that turns such a mobile IP connection into a personal WiFi "hotspot." Since laptops and smart phones almost universally support WiFi, this is an easy way to get an Ethernet connection to them. The broadcast IP codec would need a WiFi interface for this to work (see Section 7.7.7).

7.7.5 Ethernet Radios

There are plenty of radio systems on the market that can be used for Ethernet links. Most work on the license-free ISM bands at 2.4, 5.2, and 5.6 GHz. With high-gain antennas and a line-of-sight path, these can have range up to many tens of miles/kilometers. Bitrates are in the tens of megabits, so there is much more bandwidth than is typically needed for audio applications.

7.7.6 WiMax

These are Ethernet radios that conform to a standard, assuring interoperability. Unlike the usual Ethernet radio that is used for point-to-point operation, these permit multiple sites to share a common channel and infrastructure. Since the channels are shared, there would be the possibility of oversubscription and contention for

bandwidth. Perhaps WiMax vendors and providers will introduce some form of priority mechanism to offer guaranteed quality of service.

7.7.7 WiFi

IP codecs usually work over WiFi radio links without trouble. These would normally be only a part of the total IP path, however, perhaps extending an available DSL connection to the required location at a remote site. Again, the bandwidth is in the tens of megabits, much more than is usually needed.

7.7.8 Satellites

Many satellite services are now IP based and can be used for both point-to-point and point-to-multipoint links. While satellites are certainly exotic compared to other IP connection methods, from the perspective of the terminal equipment, they look about the same as any other link.

7.8 EBU N/ACIP STANDARD

Led by Mathias Coinchon of Radio France and Lars Jonsson of Swedish Radio, the European Broadcast Union established a collaborative process that produced a standard for broadcast IP codecs. The N/ACIP (Norm/Audio Contribution over IP) *Requirements for Interoperability* document was published in April 2008 as EBU Tech 3326.

To be compliant, each manufacturer must support a core set of functional components. A vendor is free to add its own enhancements as additions to the required core, but these would not be guaranteed to work with other vendors' products.

The standard specifies:

- *Transport protocols:* To be used on top of IP, including port definition and packet-loss recovery mechanisms.
- *Audio coding algorithms:* Three codec categories are specified: mandatory, recommended, and optional.
- *Audio frame encapsulation:* Definition of framing and encapsulation of the codec frames into transport-layer IP packets.
- *Signaling:* Defines connection setup and termination procedure, and signals parameters for the receiver, such as the audio coding to be used.

The standard recognizes three types of audio contribution:

- Unidirectional with no return channel (e.g., satellite links)
- Bidirectional where the return audio is narrowband and used for purposes of cueing only (e.g., concerts, sporting events—latency is not an issue)
- Bidirectional with bidirectional broadband audio (e.g., interviews with remote guests, discussion programs with distributed talent—latency is an issue)

7.8.1 Transport Protocols

The underlying network for N/ACIP is always to be IPv4. RTP/UDP is specified as the required transport. Unicast is mandatory, and it is recommended that IP multicast is supported as an option.

Lost-packet recovery by using retransmission according to RFC 4588 may be used as an option. Support of the extended RTP profile as defined in RFC 4585 is necessary for implementing this retransmission method.

TCP may be used as another option for retransmission recovery. Again, latency is an issue, and the N/ACIP document includes this caveat: Congestion avoidance mechanisms of TCP may lead to problems with continuous streams on networks with packet loss and long roundtrip delay. Transmission overhead is higher with TCP.

The framing should be done according to RFC 4571, *RTP over Connection Oriented Transport*.

At this writing, forward error correction is "currently a work in progress" according to the N/ACIP document. It mentions that an option that is being considered is the one described in RFC 5109, *RTP Payload Format for Generic Forward Error Correction.*

7.8.2 Audio Coding

The standard specifies the following codecs:

Mandatory

- G.711 A-law and u-law
- G.722 at 64 kbps
- MPEG-1/2 layer 2 at 32–384 kbps
- PCM linear at 12/16/20/24 bits and 32/48-kHz sampling rate (optional for portable units; 12 bits is optional for all)

Recommended

- MPEG-4 AAC
- MPEG-4 AAC-LD
- MPEG-1/2 layer 3 at 32–320 kbps

Optional

- MPEG-4 HE-AACv2
- Enhanced APT-x
- Dolby AC-3
- AMR-WB+

7.8.3 Signaling

SIP must be used as the signaling method for bidirectional links. SDP (Session Description Protocol) is used to list the available codecs. Codec negotiation should use the model described in RFC 3264, *An Offer/Answer Model with the Session Description Protocol*. For the IP multicast case, SAPv1 is used for session announcement.

By default, a call is established with both directions using the same codec. It is recommended that using different codecs in each direction be supported, however. It also should be possible to change codecs during an already established connection.

7.8.4 What's Missing from N/ACIP?

None of the required codecs have effective concealment mechanisms, so they must be used either with a QoS-guaranteed network or with some form of retransmission. Packet loss would otherwise cause annoying pops and/or audio dropouts. Retransmission is not mandatory, so a compliant device might not be useful for bidirectional applications where low latency is necessary. And since retransmission causes latency to rise unacceptably for bidirectional applications, a device that includes it might still not be satisfactory. The nonmandatory but recommended MPEG AAC codecs do have good concealment, so they could be the solution when a device includes them.

SIP servers are not mentioned in the standard, though they or something providing similar functionality are needed for most real-world applications. In particular, traversing firewalls and NATs is an unavoidable necessity when one or both codecs are behind one of these.

Broadcast codec users usually want to have data along with their audio. At its simplest, a few GPIOs allow an operator in the field to start an audio player back at the studio. More sophisticated applications might include extending the station automation system or on-air telephone production software interface. This needn't involve the codec at all, since IP links have no problem with being shared. There are some times when the codec device might be good to have in the loop, though, such as when tight time synchronization is needed, or when the codec has been able to punch through a firewall (thanks to having access to a SIP server), while standard IP data traffic remains blocked. Recognizing this, the N/ACIP people have begun a discussion (in January 2009) about updating the standard to include a way to implement this capability.

CO-OPETITION One sun-dappled fall morning, a small group drawn from around the world found themselves doing something unexpected. While drinking coffee and eating sweet rolls at a guesthouse among the swaying trees on the Case Western University campus in Cleveland, these competitors who would normally be eying each other suspiciously were having a friendly chat. It was early in the 1990s and ISDN codecs were just making their entrance. One broadcast codec vendor, who was not in attendance, had chosen to implement MPEG-1 audio layer 2 in a nonstandard way (even naming it something else), and the others had decided to counter by ensuring compatibility among their products. They had the support of the developers of the MPEG coding algorithms, who were also represented at the meeting. The result was constructive: Everyone present agreed to use common protocols so that products from all would work smoothly together. They also agreed to arrange testing to confirm that their engineers had got the implementations right. The plan decided that day in Cleveland has endured and continues to benefit the broadcast community right up to the present.

7.9 LIVEWIRE-ENABLED IP CODECS

7.9.1 Telos Z/IP

The Zephyr/IP is an N/ACIP-compliant IP codec family. In addition to the required G.711, G.722, PCM, and MPEG layer 2 codecs, it also includes MPEG AAC, HE-AAC, and the new AAC-ELD codecs. (See Figure 7.2.)

FIGURE 7.2

N/ACIP-compliant Telos Z/IP codec. It includes adaptive features for operation over non-QoS networks.

FIGURE 7.3

Telos iPort eight-channel codec.

If the AAC-ELD codec is used with an adaptive receive buffer and automatically adjusted bitrate, the low-delay strategies described above are fully implemented.

The Z/IP includes a Livewire interface so that it can connect directly to AoIP-based studios that use this technology.

7.9.2 Telos iPort

The Telos iPort is attractive for studio facilities that are built on Livewire. It can be used to link two locations—a studio and transmitter site, for example—that need up to eight bidirectional stereo audio paths between them. From a user's perspective, the audio channels from the remote location look like they are local.

It can be used for any application where MPEG encoding and/or decoding is needed for transmission over IP channels, such as VPNs, satellite links, Ethernet radio systems, and telco- or ISP-provided QoS-controlled IP services. It includes the MPEG AAC, HE-AAC, HE-AACv2, AAC-LD, and layer 3 (MP3) codecs. (See Figure 7.3.)

The iPort has only two Ethernet jacks for all audio and control: one for the Livewire I/O and another for the WAN connection. Here, again, we see the advantage of IP audio. Eight analog bidirectional stereo channels would need 32 XLR connectors and the A/D and D/A converters to drive them. AES3 would cut the number of XLRs to 16—an improvement, but certainly not as efficient and low cost as a single RJ-45.

It's possible to set the iPort to a 16-stereo-channel encode-only mode. This would normally be used with a server such as SHOUTcast or Steamcast to encode for Internet streaming, broadcasting to mobile phones, and audio distribution systems.

"THE REAL WORLD": KIWI IP

"The challenge was to build a 256 × 256 audio routing network to serve 25 remote stations across an IP WAN with the possibility to route any source from any location to any destination. Without IP, this would have been crazy expensive. And probably racks full of stuff. With IP, we delivered a compact, impressive system at a remarkably low cost."

—Igor Zukina, Director of Engineering, Streamcom, New Zealand.

This project is an excellent example of how IP can simplify systems and reduce cost (see Figure 7.4). Igor describes the system:

PungaNET *is a real-time national audio distribution and contribution network that connects 25 radio stations across New Zealand. Each station has eight audio inputs and outputs that can be cross-connected system-wide in any needed configuration. The network also provides TCP/IP connectivity for standard office applications such as file sharing, VoIP, and Internet access. A public web site provides Internet streaming of programs for all member stations. The system was commissioned at the beginning of 2009.*

Routing and management (booking and scheduling) is provided by a central management system, which is a software application with a web user interface that is accessible across the network. Audio equipment is provided by Telos/Axia and the central switching equipment by New Zealand–based XI-Audio. Cisco routers are used at the network edges.

A requirement for the system design was that it be based on a standard private IP networking service available from Telecom New Zealand. This is a product made to provide secure IP service for businesses between their office locations. These IP networks are much lower cost than the old synchronous circuits that were used in the past to interconnect audio codecs. The IP network can provide QoS, with various levels being available at a range of price points. We chose an "interactive class network," which is a product sitting between the cheapest "business data" and the most expensive "real-time" class-of-service networks. The central site has a 60-Mbps circuit and each station has a symmetrical 2-Mbps circuit. The latency of the data network from the sites to the center is around 10–15 ms, although for one of the most remote rural sites it is 45 ms. Network jitter is constantly monitored and is around 5 ms. No lost packets have been detected so far.

Telecom New Zealand has a basic network configuration that they offer at a good price, but when you request custom configurations, the cost skyrockets. For PungaNET we required multiple levels of queue priorities, which would be custom. We had in the past ordered custom-configured circuits for a number of customers, and these become a support nightmare. For example, the special settings were lost during every major telco network maintenance cycle. This may not be a problem with all providers, but it is something to consider during system design: Keep it simple where you have no direct control. To avoid this problem, we decided to install our own IP routers at the network edges, and use a single class-of-service network from Telecom New Zealand. This provides us with full packet tagging and queue management under our own control.

With 2-Mbps links at the stations and the need for eight stereo channels, audio coding is required. We use a few different codecs:

- *Audio distribution: Discrete stereo AAC-LC at 128 kbps*
- *Audio contribution: Discrete stereo AAC-LD at 128 kbps*

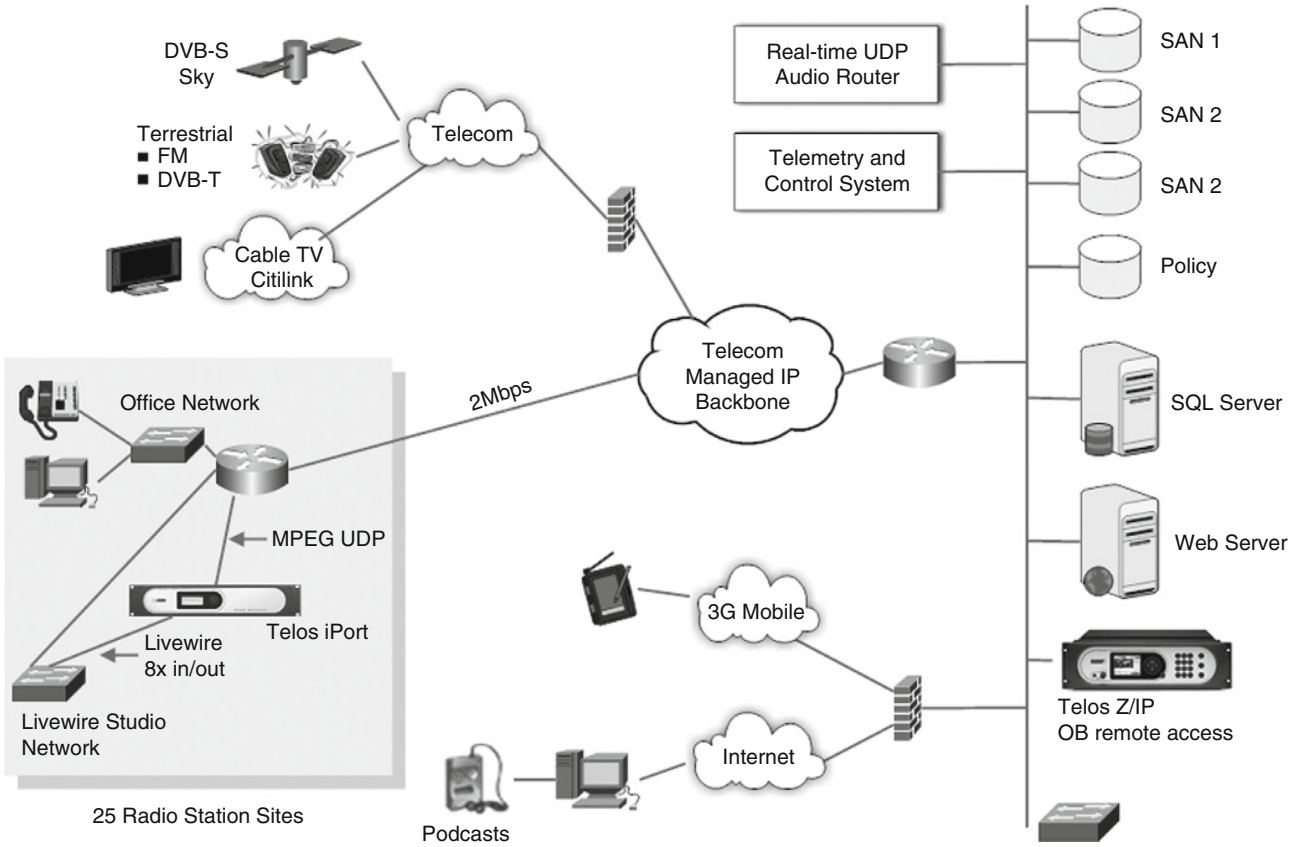


FIGURE 7.4

PungaNET system diagram.

- *Intercom: Mono AAC-LD a 64 kbps*
- *Internet stream: Parametric stereo HE-AACv2 at 28 kbps*

The MPEG AAC family offers a range of bitrate and delay trade-offs. For on-air audio, we went with standard AAC for its high fidelity. For the intercom system, low audio latency is more important than fidelity, so we chose AAC-LD for that. For Internet streaming, we use HE-AAC for its very high bitrate efficiency.

We didn't want to use IP multicast in PungaNET's WAN due to our desire to stay with a standard telco IP product. Therefore, we contracted with local firm XI-Audio to make a unicast UDP audio stream router that gives the same result but keeps the routing entirely within our hands. The router runs on off-the-shelf PC hardware, so cost was much lower than a traditional audio router would have been. It is located at the central site.

Each station has a Telos iPort, which encodes and decodes eight channels of audio. The iPort outputs are UDP/IP streams. Each of these is mapped to a port on the UDP router. The router receives the stream and sends it to the required destination, or copies of the stream to multiple destinations, if needed. The outputs from the router are mapped to the iPort decoder inputs at each station. The router has 256 inputs and 256 outputs, all carried over one RJ-45. From a user perspective, it works just like a traditional audio crosspoint router.

The UDP switch is controlled by Axia's Pathfinder router control software, which provides a user interface and automatic management of route changes.

GPIOs are provided by Axia devices at the stations. These are switched along with the audio in the router and managed by Pathfinder.

PungaNET provides scheduling of live-to-air content. A master control-like facility is required, which can provide cross-mix transition from local audio to network audio, from one network program to another, and from the network back to local. This mixing is performed by a V-mix instance (supplementary audio mixer function) with the station iPorts.

The central router management (CRM) system provides a web-based management tool. Users at each station can log on to the CRM over the network to manage and schedule network and local resources. The system's database and business logic provides a large set of resource scheduling and controlling mechanisms. Users can share studio resources to selected members of the network, configure intercom routes, or simply schedule on-air programming. Program providers can publish their schedules to the program guide.

There is a Pathfinder server at each station. The CRM system interrogates each individual Pathfinder at all the remote locations and collects information about all audio and GPIO devices, ports, and Pathfinder router organizations.

When a user schedules channels and resources in the CRM system, event scripts are created and loaded to each Pathfinder scripting engine for execution. Schedule events are downloaded to each Pathfinder server 48 hours in advance so that a fault or maintenance of the CRM system will not affect system routing.

All stations have Axia Livewire-based studios. The Telos iPort has a direct RJ-45 connection to the studio Livewire network that conveys all eight in/out audio channels. Thus, there is no analog or AES3 anywhere in the system.

7.10 CONVERGENCE

This is becoming a familiar story, isn't it? With broadcast codecs migrating to IP and telephony moving to VoIP, it seems some kind of convergence is inevitable. Both technologies use SIP/SDP, so a particular call can be specified to use either a telephone-grade codec to interwork with the public switched voice network, or a high-fidelity one. Mobile phones might well use VoIP in the future, as some proposals within that industry are suggesting.

Just as some broadcasters discovered that sharing an ISDN PRI or T1 among both telephones and codecs was a way to get lower cost, there are going to be opportunities to share telco or ISP-delivered IP services.

Within studios, we'll soon have broadcast codecs, on-air telephone systems, and routing/mixing all in the IP domain. Things are probably going to get simpler, cheaper, and more interesting.

A DIFFERENT KIND OF CODEC Once our distant ancestors developed language, humans could benefit from the experiences of others. But the bandwidth of speech is so low compared to the rate of human sensory data that it required huge compression and decompression at each end of the communication channel. This process of describing and interpreting was enabled by detailed world models that everyone carried in their heads and "mapped" their experiences to.

Because these world models vary from person to person, the coding is lossy, and misunderstandings are inevitable. But the imprecision also makes words more intimate and personal. If some words particularly resonate with you, it's because they are natively supported in the way you view the world.

From appreciation of this condition came Alfred Korzybski's famous premise: *The map is not the territory*. His point, of course, was that an empirical description, however precise, could not truly recreate the human experience of the actual environment, but it could provide a useful presentation of its structure.

On the other hand, it seems the underlying grammar of language—the *deep structure*—is pretty much hardwired among humans. (This insight was one of Noam Chomsky's major contributions to linguistics.)

The result is that as we process words, we intrinsically assess the trustworthiness of their source. We can learn not to believe everything we hear, or to distrust certain people. Much of this comes from a misalignment of people's maps. Think about your last

A DIFFERENT KIND OF CODEC—cont'd conversation with the general manager. Perhaps we need some kind of codec negotiation procedure at the start of each conversational interaction with partners with whom we haven't already established the necessary common ground?

This poses a special problem for authors of books such as this one. Since we don't know you, we can only imagine what your experience, and thus your maps, might be. We can only hope the coding loss in our communication channel with you is low.

CHAPTER

Troubleshooting 8

Ethernet is a mature technology with millions of installations and years of proven reliable service. You are not very likely to have problems in the core networking technology if you follow basic wiring, network layout, and configuration principles. Nevertheless, the real-time audio application is a special case, and specific knowledge can help. We'll cover some general Ethernet and IP troubleshooting topics here, as well as those expressly oriented to AoIP and Livewire.

8.1 PREVENTION

Before we get into troubleshooting, however, it should be noted that the best way to avoid downtime is to build the network well in the first place. Use high-grade cables, good-quality factory-made patch cords, etc. Be careful with the punch-down and plug installation. If you make your own patch cords, they should be built with stranded wire cables. Solid conductors are likely to crack when flexed a lot, usually right at the RJ-45 plug, causing intermittents and bit errors. Be sure you have the right plug for the cable you are using: An RJ-45 plug designed for stranded wire will cut through a solid conductor.

8.2 BASIC NETWORK TESTS AND DIAGNOSTICS

8.2.1 Link Test

A layer 2 test, the link test checks the connection between the switch and a designated network device on the same LAN. During the link test, IEEE 802.2 test packets are sent to the designated network device in the same VLAN or broadcast domain. The remote device must be able to respond with an 802.2 Test Response Packet. Most switches support this test via a Web or command line interface.

Audio Over IP

doi: 10.1016/B978-0-240-81244-1.00008-0

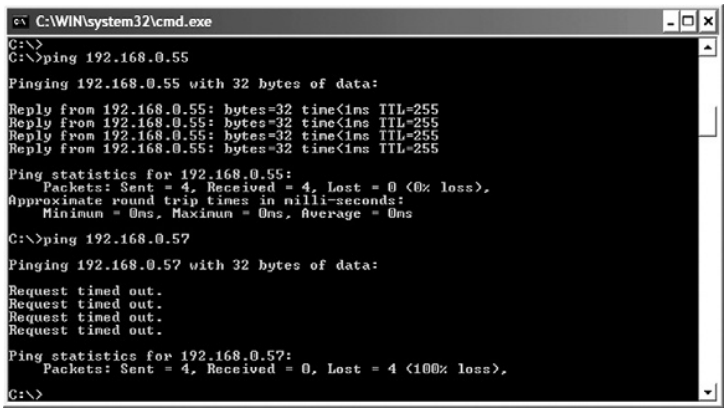


FIGURE 8.1

Accessible from the Windows command prompt, ping is a basic test of IP connectivity. In this example, the first ping was successful and the second was unsuccessful.

8.2.2 Ping and TraceRoute

A layer 3 test, ping is a simple and effective way to check basic "reachability" of an IP-enabled device (Figure 8.1). Ping sends a test packet to a device and waits for an echo response. A Windows PC can do this within the command prompt window. Enter ping x.x.x.x, where x.x.x.x is the IP address. If you get the echo, the basic connectivity (including layers 1, 2, and 3) is okay. You can substitute the device's domain name for the IP address if a DNS server is available on the network.

TraceRoute (often shortened to tracert) is a more elaborate version of ping, showing the path packets are taking through the network and an estimate of delay at each routing node.

8.2.3 Rate and Duplex Modes

Ethernet links can be 10, 100, or 1000 Mbps, full or half duplex. Most devices support a range of modes. For Livewire, you always want the maximum rate both sides can support and full duplex. With both the switch and the connected device set to Auto, both sides will automatically negotiate to the correct modes.

THE CASE OF THE DUPLICITOUS DUPLEX Beware of this common problem: You can configure Ethernet ports for specific modes manually, *but you should not do this*. The main problem is with the duplex setting. If you set this manually to full duplex on a connected device, the switch—in compliance with a flawed IEEE standard—will most frustratingly and counterintuitively set itself to half duplex, which will cause audio breaks and other troubles. Livewire equipment is always set to the Auto mode by default, so this problem will not arise with those devices. But look out for this with other equipment, such as PCs. Unless there is a well-reasoned purpose to do otherwise, switch ports should always be configured to Auto.

8.3 ETHERNET SWITCH DIAGNOSTICS AND CONFIGURATION

8.3.1 Link, Activity, Duplex, and Rate LEDs

The *link* LEDs on the front panels of Ethernet switches and other devices tell you if the cable is connected and whether low-level connectivity is good. Checking the link LED is always a good first stop in an Ethernet troubleshooting situation.

The *activity* LEDs (usually amber or green) will be on continuously when any Livewire audio streams are present on the link. That is because the logic that drives the LED extends the on time so that you can see it with normal traffic. Livewire packets are traversing the network at such a fast rate that the LED never has a chance to turn off.

The *duplex* and *rate* LEDs, when present, should indicate the correct values. An engine that expects to have 1000-Mbps link capacity will not be happy with 100 kbps. And full-duplex is always required.

8.3.2 Advanced Switch Diagnostics

High-end Ethernet switches provide diagnostic tools. Most offer a web interface as an easy-to-use first point of departure (Figure 8.2). Many also have a command-prompt telnet interface for more sophisticated configuration and monitoring than the web GUI provides.

Checking port bandwidth utilization is another good troubleshooting item. This allows you to see if you have a working link and if audio packets are properly flowing. If the utilization is too high, there is probably something wrong with the switch's multicast configuration, which is causing packets that should not be destined to a particular port to be sent there. Bandwidth consumption for a Livewire Livestream is around 3 Mbps. Simple arithmetic will give you the number of active streams flowing through a port.

On some switches, Simple Network Management Protocol (SNMP) can offer a yet deeper look. SNMP and remote monitoring (RMON) are part of the TCP/IP Internet suite. (RMON is built on SNMP, so they are closely related.) They offer a way to probe and monitor network equipment operation in a vendor-independent way. For example, an Ethernet port has a standard way of communicating its status that is supposed to be used by all products with these ports. Almost all sophisticated Ethernet switches offer these, and they are useful tools to monitor traffic, check operation, etc. To use SNMP and RMON, you will need a software application that presents the information. Hewlett-Packard's *OpenView*, for example, can do this. (H-P ships a simpler version called *TopTools* with many of its switches.) The Management Information Base (MIB) is used to organize information within SNMP.

8.3.3 Switch Configuration

Proper switch configuration is essential to correct operation of the system. Because Livewire uses multicast, an Ethernet switch's default out-of-the-box configuration

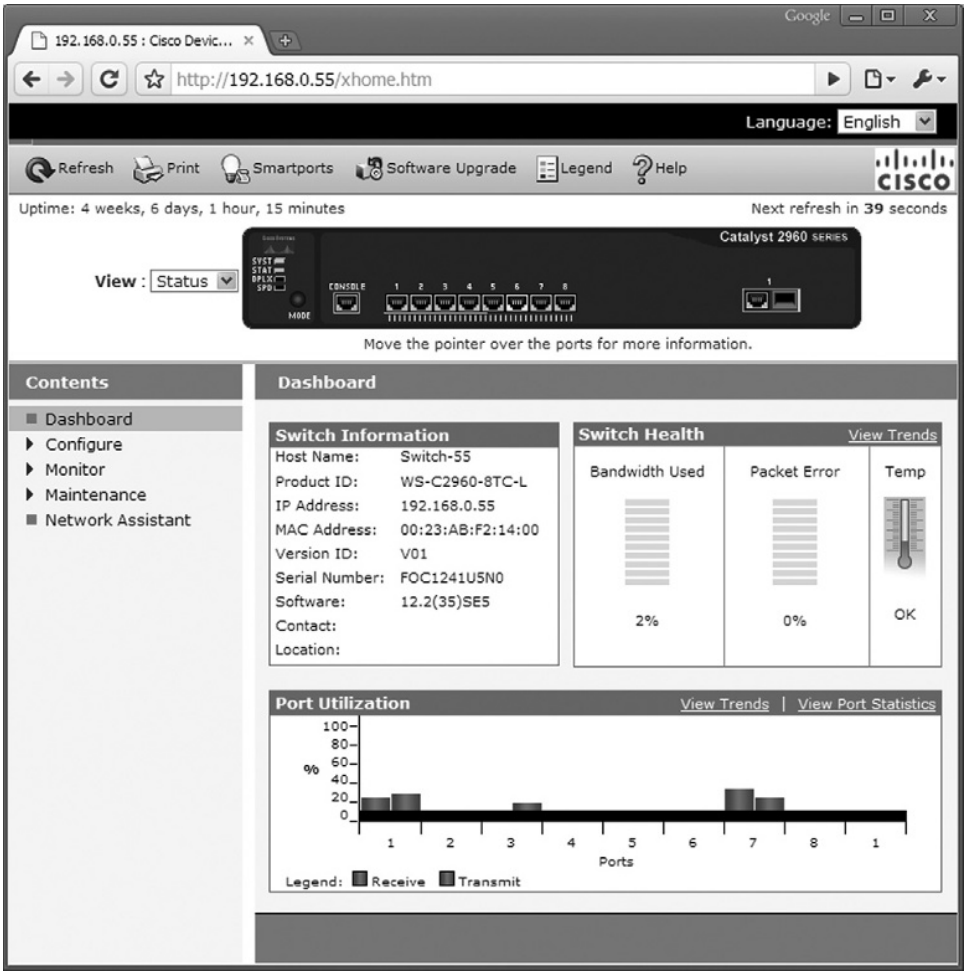


FIGURE 8.2

Managed switches provide a web page to check their status. Here is a typical one, showing general status and port bandwidth utilization.

might not work. IGMP must be switched on, VLAN parameters set if you are using them, etc. This topic is well covered in Chapters 3 and 5, so we won't go into detail here. We've included this reminder here, though, because in our experience this is the second most common cause of trouble (after cable-related problems).

For switches that have been tested by Axia, there are "precooked" configurations posted on the company web site. See *http://www.axiaaudio.com/manuals/default.htm*

and scroll down to the "Ethernet Switch and Adapter Documents" section to check your configuration against the one recommended for a particular switch.

8.4 CABLE TESTERS

"It's the cable. It's always the %&#@!* cable!" said Steve's first boss. This was in the analog audio days, and about half the time, he was right. That percentage is probably a bit *higher* in Ethernet systems. Indeed, a number of surveys have put the "network medium" to blame 70–80 percent of the time. This refers to the cables, connectors, and hardware components that make up the signal-carrying portion of the installation.

Wiggling and unplug–plug operations are legitimate and effective troubleshooting methods. But there are plenty of cable testers to help you perform more elaborate checks. These range from simple conductivity testers to sophisticated units that test cables for adherence to the TIA/EIA standards, detect breaks with a time-domain reflectometer (TDR), and more.

The testers shown here represent something of the range available. At the top of the line, the Agilent Framescope 350 and the Fluke DSP-4000 family can certify that your cable meets the designated category requirements with regard to crosstalk, attenuation, etc., and can perform a number of other sophisticated tests. (See Figures 8.3–8.6.)

8.5 SNIFFERS

Sniffers are software applications that run on PCs, which can listen in on the packets flowing on an Ethernet link. High-end Ethernet switches include a *port-mirroring* function that lets you designate a monitoring port that mirrors (copies) traffic on any other port you select. The PC running the sniffer connects to the monitoring port.

PC-based sniffers are quite good for checking such things as the advertising channel. With a fast PC and a good network card, even the small and frequent Livestream audio packets can be successfully captured.

WireShark is the current PC-based sniffer champion, available on both Windows and Linux (Figure 8.7). It is free and available at *www.wireshark.org*. It has all the features you need for basic work. You can set filters to limit the capture to the specific data you need to see, make timestamps, start capture on various trigger conditions, and more. Data can be recorded to a file for later analysis.

There are hardware-probe-based sniffers that have no problem with fast packet rates, don't need Ethernet switch port mirroring, and provide nanosecond-accurate timestamps. These are expensive and made mostly for development-lab applications. While it's not very likely that you will ever need to use one of these, it's good to know that they exist.

FIGURE 8.3

Agilent Framescope 350.

8.6 LIVEWIRE COMPONENTS

Most Livewire components feature diagnostic tools that can help you to sort out various conditions that may occur:

- Livewire nodes have a built-in loop-back testing procedure that measures audio noise and distortion. The web interface lets you check a number of conditions. For example, there is a web page that lets you see audio levels for assigned sources and destinations.
- The router selector node is a useful device for listing available audio streams and listening to them. Since it has one source-channel send, it can be used as a test audio injector.

FIGURE 8.4

Fluke DSP-4000. The adapter at the top can be changed to allow the unit to work with both copper and fiber cable types.

- The Livewire PC driver lets you open a diagnostic window that tells you about the system clock and audio stream status.
- The iPlay application gives similar information about stream status with an operator-level user interface that permits easy selection of the audio channel to be monitored.
- Pathfinder has an accurate audio metering capability that can be used to check any stream on the network.
- The PC-based iProbe software application can check stream status with regard to packet jitter, drop, and correct arrival sequence. It can play audio to the PC's sound output. It can also check system-wide to ensure that correct firmware versions are installed, etc.
- The engines associated with mixing consoles have stream monitoring and reporting capabilities that can help you to find problems with lost packets, jitter, and sequencing errors.

FIGURE 8.5

Fluke's MicroScanner Pro is a simpler model that checks for wiring errors. It can also tell you the distance to a break with a TDR and do tone trace with an optional remote unit.

8.6.1 Livewire Node Network Status LEDs

On Livewire nodes, four LEDs indicate the status of the Livewire and Ethernet connections, as well as valuable information about clock synchronization:

LINK. When illuminated continuously, this LED represents the presence of a working Ethernet link to the switch. It does not, however, indicate the quality of the connection. If no Ethernet link is present, this LED will blink slowly.

LIVEWIRE. This LED indicates that the connected Ethernet segment has Livewire traffic present. If the LINK LED is illuminated, and the LIVEWIRE LED fails to illuminate, there are either no other Livewire devices connected or there is a problem with the Ethernet switch such that it is not passing traffic through to the port to which the Livewire node is connected.

MASTER. Livewire employs a master/slave clocking system. At any one time, only a single device can be the clock master. The system has the ability to automatically change to a different clock master should the current master become

FIGURE 8.6

ByteBrothers basic wiring tester and tone line-finder.

disconnected, or otherwise become inoperable. This happens transparently, without any audio glitches. If the MASTER LED is lit on a node, it indicates that this node is currently acting as the master for the system.

SYNC. If sync packets are being properly received by the Livewire node, this LED will be continuously illuminated. When a node is first connected, the LED will flash slowly, indicating that the PLL is working to achieve lock. Normally, this condition would continue for a few seconds, then the SYNC LED stays lit continuously.

The SYNC LED indicates the receipt of clock data from another Livewire node serving as the system master. As noted above, illumination of the MASTER LED indicates that this node is acting as the master clock source for the network. Therefore, during correct operation, either the SYNC or the MASTER LED must be illuminated. If both or neither LED illuminates, something is amiss.

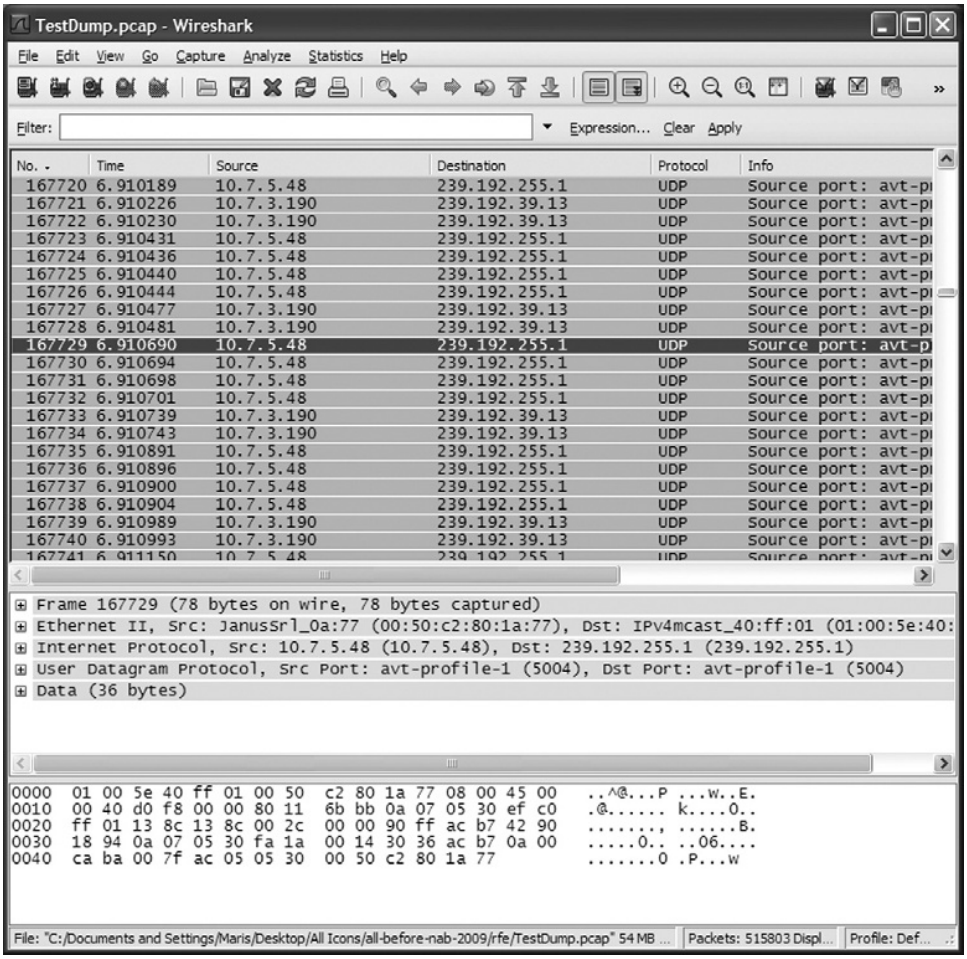


FIGURE 8.7

WireShark in action. At no cost, it turns a PC into a pretty good packet sniffer.

8.6.2 Checking Audio Levels

Livewire nodes have a web page showing level metering for both sources and destinations (Figure 8.8).

8.7 LOGGING

Logging is a powerful troubleshooting tool. Many Livewire components have a capability to record logs to internal nonvolatile memory. These may be recovered via the web interface to explore problem causes. It's also usually possible to configure

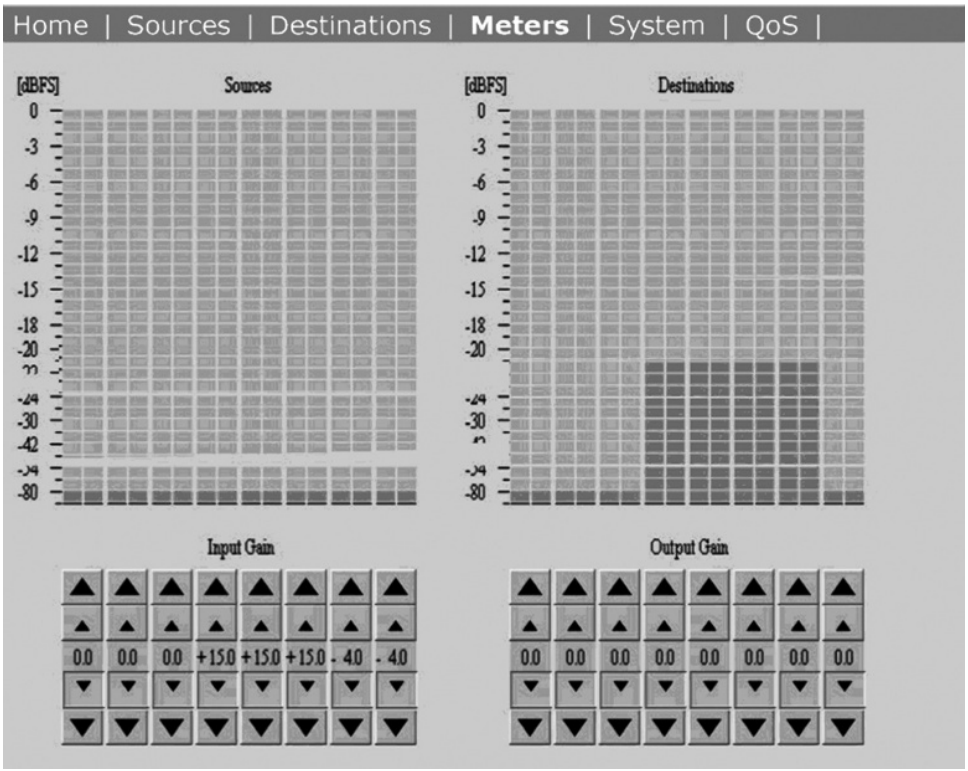


FIGURE 8.8

Livewire node audio meter web page.

logging so that it goes to a standard Linux-style syslog server installed somewhere on the network.

8.7.1 Element Console

For our first logging example, let's look at the Axia *Element* console (Figures 8.9 and 8.10). Here's an excerpt from an Element console log:

```
# Element log file
# created on Tue, 07 Apr 2009 09:58:03
# part 1
[i] Tue, 07 Apr 2009 09:58:03 - slog: logging started, log buffer
   size: 100 entries
[i] Tue, 07 Apr 2009 09:58:03 - libusbcan: Initializing CAN-bus:
   [GPIO-CAN-libUSB]
[i] Tue, 07 Apr 2009 09:58:08 - lwrd: new connection 1 from
   127.0.0.1:32772
```

```
[i] Tue, 07 Apr 2009 09:58:08 - mman: module 1:0: activated
[i] Tue, 07 Apr 2009 09:58:08 - mman: module 3:1: activated
[i] Tue, 07 Apr 2009 09:58:08 - mman: module 4:0: activated
[i] Tue, 07 Apr 2009 09:58:08 - mman: module 9:1: activated
[i] Tue, 07 Apr 2009 09:58:08 - mman: module B:0: activated
[i] Tue, 07 Apr 2009 09:58:08 - DBG: Phone channel 1: '192.168.0.99',
   'Telos', 'telos'
[i] Wed, 08 Apr 2009 16:27:57 - acl: Set Local Time exits
[i] Wed, 08 Apr 2009 16:36:50 - mman: module B:0: activated
[!] Wed, 08 Apr 2009 16:37:01 - mman: module B:0: not responding
   (71, 68; 3)
[i] Wed, 08 Apr 2009 16:37:01 - mman: module B:0: disconnected
[i] Wed, 08 Apr 2009 16:38:44 - mman: module B:0: activated
[!] Wed, 08 Apr 2009 16:38:48 - mman: module B:0: not responding
   (46, 43; 3)
[i] Wed, 08 Apr 2009 16:38:48 - mman: module B:0: disconnected
[i] Wed, 08 Apr 2009 16:38:51 - mman: module D:0: activated
[!] Wed, 08 Apr 2009 16:38:56 - mman: module D:0: not responding
   (54, 51; 3)
[i] Wed, 08 Apr 2009 16:38:56 - mman: module D:0: disconnected
[i] Wed, 08 Apr 2009 16:39:29 - mman: module D:0: activated
```

From the log, you can see that there is an intermittent problem with the connection to some of the surface fader modules. Perhaps the cable is to blame?

8.7.2 Pathfinder PC

For our next example, let's move on to the Pathfinder PC application. *PathfinderPC Server* includes a logging engine that captures information not only from itself, but from devices across the entire network. Logging may be configured to write data to text files on the PathfinderPC Server and/or to a centralized server using the standard syslog format. The logging is customizable, and you can choose from among 100 different system events that can be captured. All events of the selected types can be captured system-wide or only from specific devices. The log message types range from device connection and failure information to route changes, GPIO changes, and software and hardware button pushes, to the actual messages sent and received from the devices.

Each log message contains nine fields of information:

- Millisecond Counter
- Date of Message
- Time of Message
- Server IP Address
- Message Severity
- Message ID Number

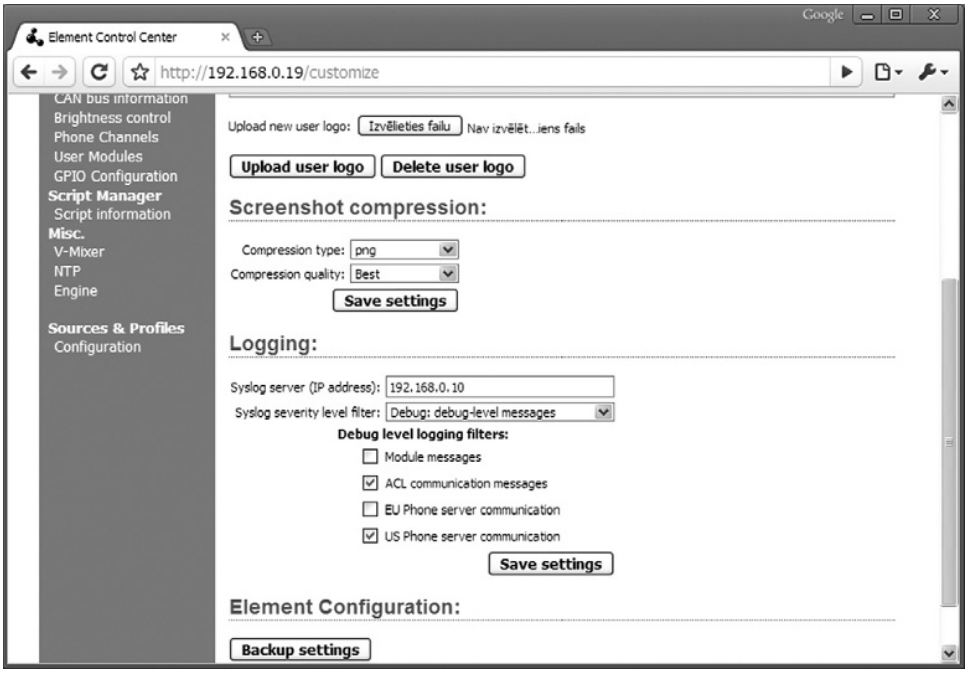


FIGURE 8.9

Element console logging configuration page. You can decide the detail level of the log with the dropdown menu and which kinds of communication are logged with the checkboxes. For example, you can log the connection between a phone control drop-in module and the phone system. It is also possible to remotely download a screenshot of the console's user display.

- Message Source
- Remote Device
- Message Data

For example:

```
11216362
4/16/2009
2:52:31 PM
172.16.1.13
6
1005
Router:2:satestserver
TCP:172.16.1.254:93
Connected
```

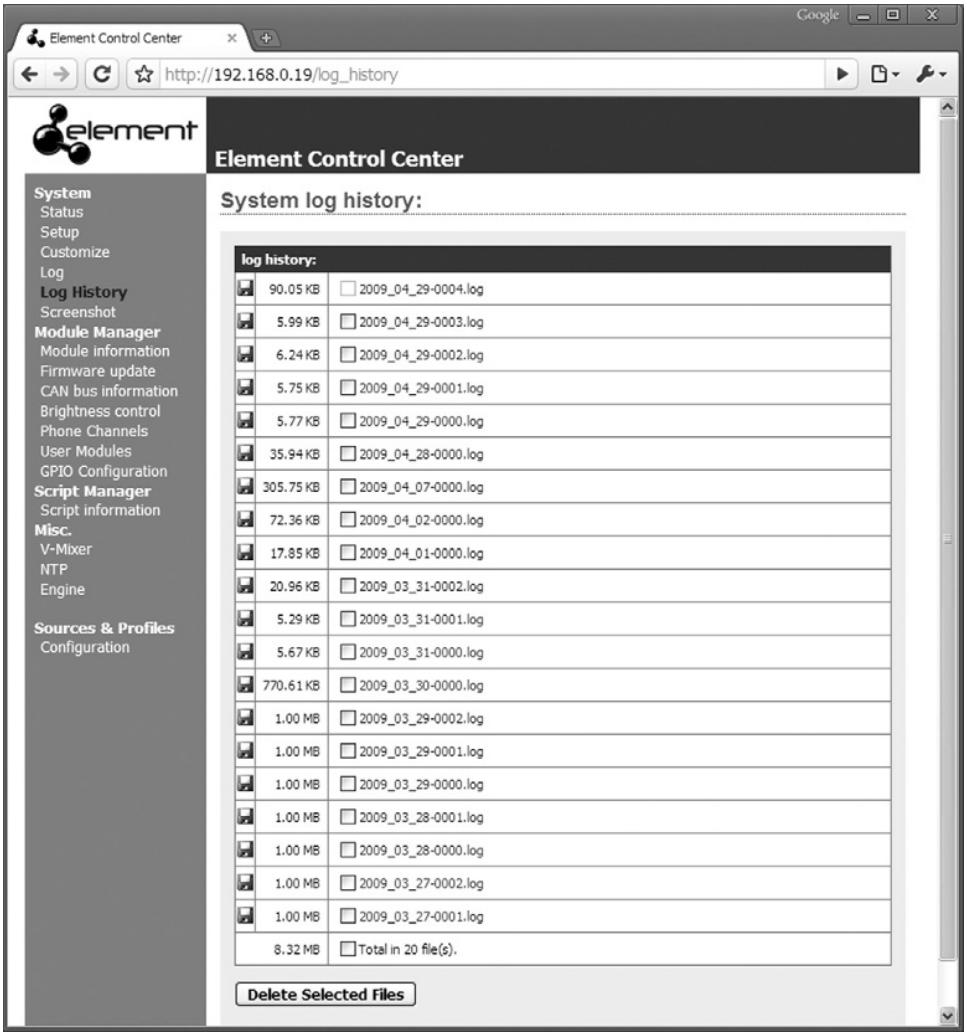


FIGURE 8.10

This screen allows you to choose and download the logs you want to see.

In this example we first see the logging time, date, and millisecond counter. The server IP address reporting the event is 172.16.1.13, and the message severity is 6, which means informational. The message type is 1005, which means it is a connection message. The Pathfinder PC server object that sourced the message is router number 2. The remote device involved is at 172.16.1.254 on port 93. The actual message is *Connected*, indicating that Pathfinder has successfully connected to the device.

Here's another example:

```
275519267
5/6/2009
9:42:21 AM
172.16.1.254
6
3004
Router:2:Main_LW_GPIO
GPO Changed
2 18 lhhhL
```

This message shows a GPIO change on the system. The message description reports the GPIO router number and I/O number in Pathfinder as well as the actual pin status of the port.

Here are a few real-world problems that have been solved with Pathfinder logs:

- One installation had a sophisticated system for network affiliate cuing. Part of the ID system's trigger was a GPIO. It seemed the system was getting multiple IDs when it should have only been getting one. By analyzing the log file, we discovered that the GPI was happening twice at the specific time. The problem turned out to be that the facility still had an old ID system wired-in as well as the new one, and both were triggering the GPIO closure.
- A user-panel button on a console was designed to trigger an action in the system. Repeated triggers were being generated, rather than the single one that was expected. By analyzing the log files, the station engineer was able to see repeated pushes on the button and track it down to a (bored?) board operator that was tapping on the button in time with music without realizing what he or she was doing to the system.
- One of the most common conditions that logs can help with is determining if a network affiliate station received a closure from a satellite-receiver relay. Because Pathfinder is connected to all of the Livewire equipment on the network, it can log the closure coming from the satellite receiver at the GPIO node. In the case of a missed event, this allows you to easily determine if the problem occurred at your end or at the network program-originator's side.

IP CSI This unfortunate event must take the prize for the use of logging. A public station volunteer who was apparently unhappy about some direction given to his program poured gasoline on the studio console and lit it on fire one evening. Pathfinder logs helped the police and station personnel determine the exact time of the start of the fire. Interestingly, because the console was simply a controller and isolated from the mix engine, audio remained on the air throughout the event.

CODA Tucked back here for those of you who made it this far is a philosophical rumination that has nothing—and everything—to do with the topic at hand.

As a young man first studying electronics, I found it both fascinating and disturbing that mathematics should have such an accurate correspondence with the behavior of chunks of metal, dialectric insulators, and all the other real-world physical pieces that make up our equipment.

Just why should

$$f = \frac{1}{2\pi\sqrt{LC}}$$

tell us reliably the resonant frequency of a tank circuit? Or indeed, why does even $P = IE$ compute the output power of a transmitter? As the wondering phase of youth passed, the matter was forgotten and the formulas were simply accepted as a working tool. But occasionally, the nagging question reappears to my now more experienced, but no more enlightened, mind.

It turns out that this is a topic that occupies the bright minds of very serious guys, and I happened upon a lecture recently by one of them that has us thinking about this theme anew. Just why are we given such a tremendous gift? Without math's power, would radio have ever been invented (or discovered)? Would anything electronic have been?

Astrophysicist and science philosopher Mario Livio of the Hubble Space Telescope Science Institute has written an entire book on the topic. In *Is God a Mathematician?* the central theme revolves around the question: Is mathematics a human invention, or is it describing the structure of the universe that we are merely, and gradually, discovering? (Roger Penrose's *Road to Reality* treads similar ground.)

Attempts at answers run the gamut. On one side, physicist Eugene Wigner, in a seminal 1960 essay, pointed out the "unreasonable effectiveness" of mathematical theorems: the astounding ability of math to predict unimagined results. Wigner was following a trail blazed by Einstein, and Einstein's general theory of relativity remains one of the best examples: His predictions about how gravity can cause ripples in space–time were recently confirmed by measuring radio waves from a set of high-energy stars called double pulsars, using technology unknown in Einstein's day. Doesn't this indicate that the mathematical structure of the world is out there waiting to be discovered?

On the other hand, math does run into limits, and chaos theory suggests that it may never be possible to predict the weather or the stock market. Cognitive neuroscientists have pointed to basic mathematical constructs in the human brain, suggesting that we impose numbers and forms on the world, not vice versa. Further, mathematics is less stable than it appears to us in grade school. At the higher reaches, there is constant upheaval and debate. If the "truths" discovered through mathematics are in flux, doesn't that indicate they are a product of human study and manipulation, rather than something fixed and eternal?

Livio takes a middle position, contending that math may be both invented and discovered. He notes the "symmetries" of the universe: The immutable laws of math and physics

CODA—cont'd that make a hydrogen atom, for instance, behave in the same way on Earth as it acts 10 billion light years away. Is this a sign of universal structure, as teased apart with the help of math? No, he says, it is more likely a sign that "to some extent, scientists have selected what problems to work on based on those problems being amenable to a mathematical treatment." He goes on to say that we probably invent the initial concepts (the postulates and axioms), but discover the relationships between them (the theorems).

So, what of our tank circuits? Why is mathematics so wonderfully satisfactory as a way to model them? Partly because radio's inventors chose a project that was amenable to mathematics. Radio waves were predicted by Maxwell's equations, and radio equipment designers found math that helped them with their designs. If we hadn't math, we wouldn't have radio in the first place, and thus there would be no need for the discovery of math that predicts tank circuit operation. Unfortunately, that doesn't resolve the fundamental "Why?" question in a conclusive way. It's a big question, and there simply is no absolute answer.

The fact that we are able to ask such questions is, in itself, an amazing thing. Working as I do each day with remarkable people doing creative design, I wonder at how the human brain, presumably adapted by evolution for eating and procreation, is able to comprehend the intricate logic of software code and DSP algorithms—or musical composition, for that matter.

Our power to make sense of the world begins with the amazing ability of the brain to detect patterns. Recognizing familiar patterns in unfamiliar situations is the beginning of reasoning by analogy, and therefore of abstract thought. Our ancestors, living as hunter–gatherers in the wilderness, must have been very good at figuring out the behavior patterns of the animals they needed for food and those they needed to avoid. They also needed to understand their own interwoven behavior patterns in order to be able to work together in cooperative groups.

Remarkably, it would turn out that the human brain that was good at these things could also recognize and generate abstract patterns in language and mathematics. Finding patterns and generating novel variations gives us pleasure. (So, yes, we ultimately do come to some connection with the main theme of the chapter.) Whether in music, poetry, dance, physics, or engineering, pattern recognition is at the heart of aesthetic enjoyment.

As we become more sophisticated, we learn to recognize ever-more subtle patterns. This leads language, logic, and the tools that amplify them to expand in a kind of runaway, self-reinforcing feedback loop. Like a transmitter "going into business for itself," once the initial conditions are met and the oscillations start, there is nothing you can do to stop them but to cut the power. Though there is not much immediate survival advantage to designing DSP software or building space telescopes, there is now no breaking the positive feedback loop that was begun when language and logic did confer an advantage. For the clever people who are drawn to do these things, there is an intrinsic pleasure that rewards their work.

In addition to language, the use of tools is a defining human characteristic. Math is among the most powerful of these, since it allows us to amplify our meager inherent logical reasoning capabilities, and helps to keep us from fooling ourselves. Again the positive feedback: We shape our tools, and they in turn shape us. Seems we humans are on quite a roll.

—Steve

(continued)

CODA—cont'd Abstracting even further, this reasoning may also apply to the *content* of much of our work in pro-audio and radio: music itself. In this case, the argument juxtaposes math versus art, rather than science (although musical instruments are technical devices, too). This part of the discussion goes back to at least as far as Pythagoras, whose "music of the spheres" showed how consonant—or "pleasing"—musicals intervals also fell into integral arithmetic ratios (of vibrating string length, for example).

The Pythagoreans took it so far as to then reject as "evil" any natural phenomenon that didn't (or they couldn't) resolve into integer values. They thus came to revile anything associated with a calculation that resulted in a remainder. Talk about scientific terrorism. It's a cautionary tale to those who would assess undeserved importance on their tools, and possess the hubris to disavow any limits to their current understanding.

Mario Livio considers such Pythagorean specifics in another of his works, *The Golden Ratio*, which he begins with a quote from Lord Kelvin: "When you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind." Livio takes this premise as a challenge for the remainder of the book, where he considers the relationship of aesthetics and mathematics throughout history—a fascinating and, at times, astounding account.

Closer to home, perhaps because both the acoustic and the electronic forms of energy we're considering here share the phenomenon of the periodic waveform (albeit in different mediums), we find that they each lend themselves well to mathematic computation. Or, once again, is that just an easy construct for us to empirically grasp far loftier physical and cognitive processes? Who or what is really in charge here?

Of course, the answers to these questions are essentially indeterminable in any empirical way. That infinite quest may be what keeps us moving forward.

—Skip

CHAPTER

FAQs

9

The body of a book is like a lecture, while a Q&A is like a conversation, and sometimes the latter is more interesting. Since AoIP is still a new technology—and a big departure from what many audio practitioners are used to—we thought a Q&A might help ease the path to enlightenment. Here are some questions about AoIP (and Livewire), and their answers, grouped into general topic areas.

9.1 RELIABILITY

Is AoIP really reliable for mission-critical applications like broadcasting?

AoIP uses the same technology that underlies VoIP telephony. Over half of the Fortune 100 companies now use VoIP, and VoIP PBX systems now outsell the traditional systems by a wide margin. With these new systems, telephones plug into a standard Ethernet/IP network. Contrast this with traditional PBX phone gear—proprietary devices that required purchase of phone sets and parts exclusively from the company that built the mainframe. You were locked into a single vendor, because the technology that ran the mainframe was owned by the company that made the gear—similar to TDM audio routers.

IP is growing as a universal transport for almost any kind of signal. You see it now in television studios, business teleconferencing, government communications, banking, and beyond. It's hardly unproven, even for applications specific to audio and broadcast studio infrastructure. There are plenty of people using it successfully today, under high-reliability requirements.

Can the same IP network be used for general data functions as well as audio?

Yes, should you choose to do so. Ethernet switches naturally isolate traffic. You can even use the same link for both audio and data, since the audio is prioritized. This will often be the case whenever a multipurpose device like a PC is connected to that network. You will sometimes want to download files, receive email, etc. on that PC, as well as use it for AoIP applications.

Audio Over IP

doi: 10.1016/B978-0-240-81244-1.00009-2

The choice of switch is important in this context, however, so you should only use units that have been tested and verified for such usage.[1] You could also set up two separate networks and link them as described immediately below.

To be clear, however, this means that the AoIP network can be used for purposes other than exclusively AoIP, but it does not advocate the mixing of an enterprise's AoIP and commercial business functions on the same network, or any other direct Internet connection on the AoIP network that might expose it to hacking.

Can the AoIP network be a completely separate entity?

Yes, and there are several ways to do it:

1. Have a completely separate and physically isolated network for Livewire. Take advantage of Ethernet on each, but don't combine any Internet or business functions with AoIP.
2. Have two physical networks as in (1), but link them with an IP router. Correctly configured, the router provides an adequate security barrier, but allows crossover traffic when appropriate.
3. Share a single physical network for audio and business functions, but isolate AoIP to its own VLAN. Again, an IP router could be used to link the two networks.

Do AoIP networks have any single points of failure? Is there a central "brain" that can take the whole system down if it fails?

A properly designed AoIP network should be distributed, with no single central device that can bring the entire system down. Ethernet networks can be designed any number of ways, including those that are fully redundant and self-healing. Larger AoIP facilities are typically built with edge switches serving each studio, connected to redundant core switches. In this way, each studio is able to operate independently, and the entire facility also has backup.

How can AoIP promise live audio over Ethernet? Won't it drop out?

No, and AoIP systems would not be acceptable for professional use if this were the case. With Ethernet switching, each device owns all of the bandwidth on a link so there is no possibility of contention or audio loss. If a node needs both audio and data, such as a PC running an audio editor and a web browser, audio is prioritized and has precedence. Thousands of hours of lab testing with careful logging of packet transmission and the many AoIP systems already in the field have proven this.

Consider that the backplane of a modern Ethernet switch can handle full-duplex traffic on all its ports simultaneously without any packet loss. If all AoIP links are designed so that they never exceed any port's capacity, the switch's capacity will never be exceeded.

[1]Switches that have been verified for successful use with Livewire are listed at *http://www.axiaaudio.com/switches*.

By the way, this concern probably arises from many years ago when Ethernet used a shared coax cable (10BASE-2). In rare cases two devices would grab the bus simultaneously. When this happened, one would back off and send a few milliseconds later. These were the famous collisions. But with today's switched Ethernet, there is no shared bus—each device completely owns its own full-duplex link. There are never collisions or lost packets as a result of network congestion; it's physically impossible.

But the Internet is a packet network and the quality is not very good for audio.

Yes, Internet bandwidth is not guaranteed, so there can be problems when there is not enough. But this is where AoIP and the Internet are worlds apart. With AoIP you completely own and control all the pieces of the network, and there is more than enough bandwidth on a modern, switched Ethernet LAN, so performance is fully reliable.

Does AoIP use any compression (perceptual coding)? Isn't codec cascading a concern?

AoIP is not compressed. In fact, with AES I/O, AoIP can offer a bit-transparent transport. When audio must leave the LAN to traverse a WAN such as the Internet, a gateway can perform compression, but this is an optional extension to an AoIP system.

9.2 COST EFFECTIVENESS

AoIP is so much less expensive than traditional architectures. What does it leave out?

Nothing. AoIP's cost savings compared to traditional routers are achieved by using standard, off-the-shelf switching hardware rather than custom-built solutions. It's a lot less expensive to use a mass-produced Ethernet switch available from any network vendor than it is to construct a customized crosspoint routing switcher, with its cards, frame, and peripherals. This is the same principle that has driven the audio and broadcast industries to use PCs for production and playout—they are a lot cheaper and more powerful than any industry-specific machine could be.

Another way AoIP can save money is by eliminating PC sound cards. Professional, multiple-output sound cards are expensive. Instead, AoIP PC drivers are used, which look just like a sound card to the computer's operating system, but use the PC's network interface card (NIC) for streaming audio in and out of applications. This approach can provide many more simultaneous channels of stereo I/O for the money than pro-audio sound cards. Further, eliminating the sound cards also reduces the cost of the I/O infrastructure in the switching network. With a traditional router, PC audio must be brought in through a router input card or console module; bringing multiple channels of audio into the system in this manner—from workstations or digital delivery systems—can significantly increase the overall cost of the routing system. Using AoIP PC drivers eliminates this cost not to mention the potential audio-quality improvements from avoiding any A/D or D/A conversions that some sound cards or routers might still require.

9.3 LATENCY

What about the audio delay through the system?

The most demanding application in this respect is live audio monitoring, such as when air talent hears his or her own microphone's output in headphones. In this case, 10 ms is the limit before noticeable problems occur. Proper AoIP system design avoids this, and in fact keeps delay much lower than this (e.g., <1 ms per link), so a number of links can be successfully cascaded. To put this in perspective, a normal professional A/D or D/A converter has about 0.75-ms delay.

Are there any problems with delay of control commands over the network? Other systems using TCP/IP reportedly have problems in this respect.

AoIP control latency also should be very low, ideally <50 ms for hardware GPIO closures.

Does latency increase whenever you add inputs?

No, latency is fixed at a low value regardless of the channel count. You can run an AoIP system with a thousand channels and the latency will be the same as for a single stereo stream. Indeed, the delay can be made so consistent that audio channel-to-channel phase shift is held to less than a one-quarter sample. LAN switches have negligible delay, and the variation in delay that they might create is smoothed by buffers in the AoIP receivers. The delay is, in fact, fixed by the packet size and the buffer length.

9.4 INTERCONNECTION

Is Cat 6 cable required to connect everything? That could get pricey.

A typical AoIP system will only require Cat 6 cable for its most heavily trafficked network segments, such as between mixing engines and switches, and between switches. All other equipment can generally be connected with common, inexpensive Cat 5e cable.

What about using Cat 5 outside the IP/Ethernet domain? Can it be used for AES3 digital audio paths?

A study conducted by the BBC Research and Development Department concluded that Cat 5 shielded twisted audio pair cable offered the highest performance of all the cables tested. These tests included coaxial cables and special cables specifically designed for digital audio, yet Cat 5 cables were preferred for their consistent performance, and their flexibility to support other signal formats.

Cat 5 cables are engineered for data rates up to 100 Mbps to support networks such as 100BASE-TX. Since AES3 signals have a bandwidth of only up to 3 Mb/sec (depending on sample rate), AES3's requirements are well within the Cat 5's

guaranteed performance parameters. Dependable, error-free transmission is possible on Cat 5 at lengths up to 920 meters (over ½ mile). Cat 5 cables perform well for AES3 because they are engineered to have characteristic impedance of 110 Ω and extremely low capacitance (~12 pF/ft). This yields low signal reflection and excellent high-frequency response, and thus produces very low error rates.

How about using Cat 5 for analog audio paths?

This also works well. Again, the low capacitance needed for high-speed data's wide bandwidths yield exceptionally flat analog audio-frequency response, even over very long cable lengths. The tight and consistent twists produce good common-mode rejection (reducing hum and crosstalk). As Steve Lampen, Multimedia Technology Manager for Belden Wire & Cable writes, "Digital cables make the absolute best analog cables. You can go farther with flatter frequency response than with any cable designed for analog."[2]

Is UTP adequate for analog or should shielded cable be used?

Generally UTP works well, but in high-RF environments shielding can help.

Is there any crosstalk between the pairs within the Cat 5 cable?

As long as your circuits are balanced, there is almost no left/right crosstalk inside the cable. With a balanced input circuit that has 50-dB CMRR (common-mode rejection ratio), separation will be greater than 90 dB.

So, must all the analog and digital audio signals be balanced?

Generally, yes, or crosstalk performance will degrade. AES3 digital audio signals are always balanced and require no conditioning. Unbalanced analog connections are discouraged, but if necessary should be used for short paths only, and even then using separate cables for left and right channels if you care about stereo crosstalk. A better solution is to make use of available adapters (from RCA to balanced RJ-45) as close to the source device as possible.

Is there a practical limit to the size of an AoIP installation?

An AoIP facility can generally have as many studios and audio channels as its Ethernet switching fabric can support. Switches come in all sizes, some with hundreds of ports, and multiple switches can be cascaded to expand the number of ports.

What about for smaller facilities? Isn't AoIP pretty sophisticated for a simple setup?

Look at how Ethernet is used in data networks: You see everything from a single PC connected to a printer, to a few PCs in a small office tied to the Internet and a couple of printers, to huge campus networks with thousands of nodes—all successfully

[2]Stephen H. Lampen, "The Axia Guide to Choosing Category Cable," available at *http://www.axiaaudio.com/tech/cable*.

using the same basic network topology. This is one of the reasons Ethernet works well for AoIP: It is easily scalable (technically and financially) from the smallest facilities to the biggest imaginable. In fact, small facilities may benefit the most because they might gain substantial new routing capability at a very modest cost.

What about hooking up over the Internet? Can you plug a port from the AoIP switch into an Internet router? Could this replace ISDN backhaul?

As the Internet becomes ever more ubiquitous and bandwidth more plentiful, arguments toward using it for audio transmission become more convincing. A gateway device could perform any necessary conversions between an AoIP system's uncompressed PCM and a lower-rate bitstream suitable for Internet transmission, using a suitable codec. The main problem to overcome is the Internet's lack of any QoS guarantees. A "net storm" that starves bandwidth and momentarily drops audio might not be a big deal to a kid at home surfing the web, but it sure wouldn't be good for an important on-air feed, for example. Private networks with reserved capacity are one answer. Another could be the "resource reservation" and "differentiated services" technologies that are making their way toward possible eventual implementation by ISPs—at a cost, of course.

In theory, RSVP, Diffserv, MPLS, IPv6, and other emerging technologies will in due course offer us reliable audio transmission via the Internet. Given the relatively slow pace of new technology adoption at the core of the public Internet, however (where not much has changed in the past decade, other than bandwidth capacity), and the problems with scaling the lab work to the real world, perhaps the following observation applies: Sometimes there is a gap between theory and practice; but the gap between theory and practice *in theory* is not as large as the gap between theory and practice *in practice*.

Therefore, the wait for ISPs to offer QoS guarantees at reasonable prices is likely to be long. And when they do, transmission delay is still probably going to be an issue for live interactive broadcasts. So, it looks as if ISDN is going to remain the most reliable option for remote hookups for awhile.

Nevertheless, if you can live with the uncertainty, and/or the Internet is your only viable backhaul option, emerging IP codecs and the commercial products/services based on them that have become available can be used (and in many cases, directly interfaced with AoIP systems) for the best possible application of the Internet to pro-audio interconnections today.

9.5 LIVEWIRE-SPECIFIC ISSUES

How do analog and AES sources become part of a Livewire network?

This is achieved via Livewire nodes. These come in variants for analog microphone, analog line level, and AES3 applications. The analog nodes include high-quality microphone preamplifiers and ADCs, while the AES3 interface uses a direct bit-to-bit

procedure with no conversion of any kind. You can also sync a Livewire system to a facility's AES master clock.

What is the best audio format to use with Livewire?

Livewire doesn't care what format your audio files are stored in. During the playout process, your playout software will uncompress any compressed-format files (MP3, MP2, apt-x, etc.) and present them to the Livewire PC driver you've installed on the PCs. From that point Livewire carries all audio in uncompressed 48-kHz/24-bit PCM. You decide what other audio coding, if any, you want to use on storage and transmission systems that you connect (via native IP, AES3, or analog audio) to the Livewire network.

Are both logic and audio routed together in Livewire?

Yes, and IP is great for transporting data, of course. Logic commands from external devices like CD players, DAT machines, etc. enter the Livewire network using GPIO nodes. The logic data are then "bound" to the audio stream and routed with it to whatever destination it proceeds.

A growing number of devices are equipped with native Livewire interfaces (e.g., codecs, telephone interfaces, audio processors, delay lines, and satellite receivers). These can supply both audio and control logic directly from the device to the Ethernet switch over a single Cat 5e connection, further simplifying in-studio wiring and making Livewire's audio-plus-logic routing even more convenient.

Is the Livewire format open to other vendors to make gear for direct interface?

Yes. Software vendors for PCs can use the Livewire driver to make their applications directly compatible. Most of the popular audio automation/playout systems already have done so, along with other broadcast equipment manufacturers. See *http://axiaaudio.com/partners/partner_products.htm* for a current list of these products.

Does Livewire support optical audio links?

Livewire is fully compatible with copper and fiber connection types. A typical configuration uses edge switches dedicated to studios, with 100BASE-TX copper connecting nodes, engines, control surfaces, etc. to the switches. A gigabit fiber backbone connects the edge switches to each other (typically at a core switch) in order to share audio among the studios.

What Ethernet rates does Livewire hardware support?

Livewire nodes connect with 100BASE-TX links. PC NICs can use 100 Mbps or 1000 Mbps, copper or fiber. Livewire processing engines use 1000BASE-T. Switch-to-switch links may be via any supported Ethernet media. You can also connect nodes via fiber using media converters should you need to do so, such as for extended-range snake applications.

What happens if someone accidentally unplugs a Livewire Ethernet cable?

Livewire nodes "advertise" the presence of their audio streams to the entire Livewire network. If someone unplugs a node, the sources attached to it will be offline, but when the node is plugged back in, it will advertise that the audio streams are available again. Within 10 seconds, all destinations that need those sources will be back up and running.

Additionally, among the many features of Livewire's Pathfinder routing-control application is a silence-detect function that can be programmed to switch to an alternate feed if one stops working for any reason.

How do you deal with mix-minus feeds in Livewire?

With so many of today's radio programs relying heavily on phones and remotes, this is an important feature, and it's a source of frequent misunderstanding by operators. Livewire mixing consoles generate mix-minuses automatically, on-the-fly, without any user intervention.

Here's how it works: When a caller is on-air, he or she hears the main program audio, minus himself or herself—a regular mix-minus return feed. When callers are standing by off the air, however, they hear a special "offline" phone mix that can contain audio sources preselected in a configuration process. These might include prefader audio from the host mic, other phone callers, etc.

Best of all, mix-minus settings for audio sources such as phone hybrids and remote codecs are assigned to the source itself, not the console fader, and retained across any routing. So when a source that needs a mix-minus is loaded onto *any* console fader anywhere in the facility, the proper mix-minus settings are automatically loaded, too. When another source is loaded onto that same fader later that doesn't need mix-minus, the mix-minus return feed is cleared.

At the physical level, mix-minus is easy, too. Livewire carries audio in both directions, so one RJ-45 covers everything.

How do contact closures get in and out of the network?

Via GPIO connections, which are included on mixing console power supplies (with 40 GPIOs), and available on separate 1RU GPIO interfaces (with five GPIOs).

As AoIP continues its penetration, however, more control functions will likely move from "dumb" contact closures to smarter network transactions. For example, an audio playout system that now uses a closure for a Start command could just take a packet over a network for this function. Beyond such replacement of today's closure-based functions, you could also have "now playing" text or other information flowing between systems. Satellite receivers could send program information and requests for specific local tasks, not just a "start something" closure, as another example. The ability to increase the sophistication of control communications via the AoIP network is another exciting prospect for future enhancement that Livewire provides.

Is Livewire compatible with program-associated data (PAD) for radio broadcasting?

Yes, and this is an existing example of the extended connectivity IP networking offers. Devices that generate PAD today (such as PC audio playout systems) plug into

Livewire via network connections and the information they supply is sent along with its associated audio. Any destination devices that need PAD can also plug into the network and retrieve it. This means that you can send audio and PAD together, without incurring extra costs for separate audio and data networks.

This seems like a lot of IP to keep track of. What administration tools does Livewire have?

All Livewire devices have a web browser control and monitoring capability. Keep the IP addresses in a "favorites list" and you can easily check them. Or make your own web page with all the links. An additional tool is the iProbe Network Management Console. This is an intelligent network maintenance and diagnostics tool that makes managing, updating, and remote controlling a Livewire network easy and intuitive.

Can RS-232 data go through a Livewire network?

Yes, using third-party devices, such as those from Lantronics, serial data can go anywhere across the network and be used where it's needed.

References and Resources

References

Preface

Calvino, I. (1988). *Six Memos for the Next Millennium*. Cambridge, MA: Harvard University Press.

Chapter 1: Introduction to AoIP

Moschovitis, C. J. P. (Eds). (1999). *History of the Internet: A Chronology, 1843 to Present*. Santa Barbara, CA: ABC-CLIO.

Naughton, J. (2001). *A Brief History of the Future*. Woodstock, NY: Overlook Press.

Pizzi, S. (2002). IP Technology. Supplement to *Radio World Newspaper*.

Pizzi, S. (2004). IP2: Digital Networks and Beyond. Supplement to *Radio World Newspaper*.

Pizzi, S. (2007). Audio Contribution and Distribution Channels. In E. A. Williams (Ed.), *NAB Engineering Handbook* (10th ed., pp. 645–661). Boston: Focal Press.

Pizzi, S. (2008). Using IP for Broadcast Studio Audio (Axia White Paper, February 2008). Supplement to *Radio World Newspaper*.

Chapter 2: Network Engineering for Audio Engineers

Kurose, J. F., & Ross, K. W. (2008). *Computer Networking: A Top-Down Approach* (4th ed.). Upper Saddle River, NJ: Addison-Wesley.

Tannenbaum, A. (2002). *Computer Networks* (4th ed.). Upper Saddle River, NJ: Pearson Education/ Prentice-Hall.

Chapter 3: Switching and Routing

Institute of Electrical and Electronics Engineers. (2008). *IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements, Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, Parts 1–5*. (IEEE 802.3-2008). New York: IEEE.

Spurgeon, C. E. (2000). *Ethernet: The Definitive Guide*. Sebastopol, CA: O'Reilly Media.

Chapter 4: Livewire System

Axia Audio. (2007). *Introduction to Livewire, Version 2.1*. Cleveland, OH: Axia Audio. Available at: http://axiaaudio.com/manuals/files/IntroToLivewire2.1.pdf

Church, S. (2003). Ethernet for Studio Audio Systems. In *Proceedings of the NAB Broadcast Engineering Conference*, Las Vegas, NV.

Dosch, M. (2003). A Network-Enabled Radio Console Architecture. In *Proceedings of the NAB Broadcast Engineering Conference*, Las Vegas, NV.

Klar, S., & Spikofski, G. (2002). *On Levelling and Loudness Problems at Television and Radio Broadcast Studios* (Preprint 5538). Munich, Germany: 112th AES Convention.

Chapter 5: Designing and Building with AoIP

Abruzzino, J. (1998). *Communications Cabling* (2nd ed.). Chantilly, VA: CNC Press.

Telecommunications Industry Association. (2001). *Commercial Building Telecommunications Cabling Standard* (TIA/EIA-568-B series). Arlington, VA: TIA.

Chapter 6: VoIP Telephone Systems in the Studio Environment

Alexander, J., Pearce, C., Smith, A., & Whetten, D. (2006). *Cisco CallManager Fundamentals* (2nd ed). Indianapolis: Cisco Press.

Au, D., Choi, B., Haridas, R., Hattingh, C., Koulagi, R., Tasker, M., & Xia, L. (2005). *Cisco IP Communications Express: CallManager Express with Cisco Unity Express*. Indianapolis: Cisco Press.

Camarillo, G. (2002). *SIP Demystified*. New York: McGraw-Hill.

Church, S., & Taylor, R. (2007). Telephone Network Interfacing. In E. A. Williams (Ed.), *NAB Engineering Handbook* (10th ed, pp. 609–644). Boston: Focal Press.

Davidson, J., & Peters, J. (2000). *Voiceover IP Fundamentals*. Indianapolis: Cisco Press.

Handley, M., & Jacobson, V. (1998). *SDP: Session Description Protocol* (RFC 2327). IETF Network Working Group. Available at: http://www.ietf.org/rfc/rfc2327.txt

Handley, M., Schulzrinne, H., Schooler, E., & Rosenberg, J. (1990). *SIP: Session Initiation Protocol* (RFC 2543). IETF Network Working Group. Available at: http://www.ietf.org/rfc/rfc2543.txt

Hersent, O., Gurle, D., & Petit, J. P. (1999). *IP Telephony: Packet-Based Multimedia Communications Systems*. Addison-Wesley: Reading, MA.

Inglis, A. H. (1938). Transmission Features of the New Telephone Sets. *Bell System Technical Journal*, *17*, 358–380.

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., & Schooler, E. (2002). *SIP: Session Initiation Protocol ('SIP v2')* (RFC 3261). IETF Network Working Group. Available at: http://www.ietf.org/rfc/rfc3261.txt

Schulzrinne, H. (1996). *RTP Profile for Audio and Video Conferences with Minimal Control* (RFC 1890). IETF Audio-Video Transport Working Group. Available at: http://www.ietf.org/rfc/rfc1890.txt

Schulzrinne, H., Casner, S., Frederick, R., & Jacobson, V. (1996). *RTP: A Transport Protocol for Real-Time Applications* (RFC 1889). IETF Audio-Video Transport Working Group. Available at: http://www.ietf.org/rfc/rfc1889.txt

Sinnreich, H., & Johnston, A. B. (2001). *Internet Communications Using SIP*. New York: John Wiley & Sons.

Chapter 7: IP Codecs

Church, S. (2008). Advanced Tech for IP Remotes. In *Proceedings of the NAB Broadcast Engineering Conference*, Las Vegas, NV.

European Broadcasting Union, Network Management Committee, Audio Contribution over IP Group (EBU N/ACIP). (2008). *Audio Contribution over IP: Requirements for Interoperability* (EBU Tech 3326). Geneva: EBU. Available at: http://tech.ebu.ch/docs/tech/tech3326.pdf

Ott, J., Wenger, S., Sato, N., Burmeister, C., & Rey, J. (2006). *Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)* (RFC 4585). IETF Network Working Group. Available at: http://www.ietf.org/rfc/rfc4585

Rey, J., Leon, D., Miyazaki, A., Varsa, V., & Hakenberg, R. (2006). *RTP Retransmission Payload Format* (RFC 4588). IETF Network Working Group. Available at: http://www.ietf.org/rfc/rfc4588

Chapter 8: Troubleshooting

Livio, M. (2002). *The Golden Ratio*. New York: Broadway Books.

Livio, M. (2009). *Is God a Mathematician?* New York: Simon & Schuster.

Chapter 9: FAQs

Kirby, D. G. (1995). Twisted-Pair Cables for AES/EBU Digital Audio Signals. *Journal of the Audio Engineering Society*, *43*(3), 137–146.

Lampen, S. H. (2005). *The Axia Guide to Choosing Category Cable* (Axia White Paper). Cleveland, OH: Axia Audio. Available at: http://www.axiaaudio.com/tech/cable/Axia_Cable_Guide.pdf

Other Resources

Livewire Products and Partners

Axia Audio. http://www.AxiaAudio.com

Livewire partner companies. http://axiaaudio.com/partners/default.htm

Livewire-approved Ethernet switches. http://www.axiaaudio.com/switches

Livewire-enabled products. http://axiaaudio.com/partners/partner_products.htm[1]

Radio Systems. http://www.studiohub.com

VoIP

Audiocodes. http://www.audiocodes.com

Cisco Systems. http://www.cisco.com

Digium (Asterisk). http://www.digium.com

LG-Nortel. http://www.lg-nortel.com

Microsoft. http://www.microsoft.com/voip

Quintum. http://www.quintum.com

[1]This is a current list of products that offer direct IP interface in the Livewire format with full interoperability. The products include audio processors, satellite receivers, telephone interfaces, audio delays, mixing consoles, and other peripheral products.

Siemens. http://www.siemens.com/hipath

SIP Forum. http://www.sipforum.org

Telos Systems. http://www.telos-systems.com

Trixbox. http://www.trixbox.org

Useful Products and Supplies

Cable and Network Testers

Agilent. http://www.agilent.com

ByteBrothers. http://www.bytebrothers.com

Fluke. http://www.flukenetworks.com

JDS Uniphase. http://www.jdsu.com/products.html

Cable and Supplies

AMP (RJ plugs and tools). http://www.amp.com

Belden Cable. http://www.belden.com

Corning Cable Systems (fiber). http://www.corning.com/cablesystems

Hubbell Premise Wiring. http://www.hubbell-premise.com

Panduit. http://www.panduit.com

Siemon. http://www.siemon.com

Cabling Information and Standards

Cabling Business Magazine. http://www.cablingbusiness.com

Cabling-Design.com. http://www.cabling-design.com

IHS Standards Store. http://www.global.ihs.com

Telecommunications Industry Association. http://www.tiaonline.org

Ethernet Radio Equipment

Dragonwave. http://www.dragonwaveinc.com

Exalt Communications. http://www.exaltcom.com

Mikrotik. http://www.mikrotik.com

Motorola Point-To-Point. http://www.motorolaptp.com

Ethernet Switches

Cisco Systems. http://www.cisco.com

Hewlett-Packard. http://www.procurve.com

Glossary of Acronyms

The following list defines the acronyms found in this book. In cases where a term is not a true acronym or abbreviation (e.g., XLR), it is otherwise defined. Corporate or commercial product acronyms are generally omitted.

1000BASE-LX 1000-Mbps Ethernet (gigabit) on single-mode fiber

1000BASE-SX 1000-Mbps Ethernet (gigabit) on multimode fiber (short wavelength)

1000BASE-T 1000-Mbps (also called gigabit) Ethernet on twisted pair

100BASE-FX 100-Mbps Ethernet on multimode fiber (long wavelength)

100BASE-T Original name for a family of 100Mbps Ethernet standards, most of which are now obsolete, and replaced by 100BASE-TX.

100BASE-TX 100-Mbps Ethernet on twisted pair

10BASE-2 10-Mbps Ethernet on coaxial cable (RG-58A/U or similar), now obsolete.

10BASE-T 10-Mbps Ethernet on twisted pair

10/100BASE-T Designation for equipment that supports both 10BASE-T and 100BASE-TX

10 GbE 10-Gbps Ethernet

1U One rack-unit (1.75 in [44.45 mm] height in a 19 in [48 cm] wide equipment rack)

2U Two rack-units (3.5 in [88.9 mm] height in a 19 in [48 cm] wide equipment rack); continues as ***n*U**, where rack height equals $n \times 1.75$ in (44.45 mm)

3GPP 3rd Generation Partnership Project

5ESS Widely deployed TDM telco CO switch

AAC Advanced Audio Coding

AAC+ Commercial name for HE-AAC standard

AAC-ELD AAC Enhanced Low Delay

AAC-LD AAC Low Delay

A/D Analog-to-Digital

ADC Analog-to-Digital Converter

ADPCM Adaptive Differential Pulse-Code Modulation

AEC Acoustic Echo Canceller

AES Audio Engineering Society

AES/EBU See *AES3*

AES3 Audio Engineering Society standard for serial digital audio transmission, defined most recently by AES3-2003 (also called AES/EBU).

AGC Automatic Gain Control

AMR-WB Adaptive Multirate, Wideband (audio codec standardized as ITU-T G.722.2)

AMR-WB+ Adaptive Multirate, Wideband Plus

AoIP Audio over IP

ARCNET Attached Resource Computer Network

ARP Address Resolution Protocol

ASCII American Standard Code for Information Interchange ("AS-key")

BGP Border Gateway Protocol

BRI Basic Rate Interface (suffix to ISDN)

CAN Controller Area Network

CAS Channel-Associated Signaling

Cat 5 Category 5 cable

Cat 5e Category 5 extended cable

Cat 6 Category 6 cable

CDMA Code-Division Multiple Access

CE Consumer Electronics; Chief Engineer

CEA Consumer Electronics Association

CO (Telco) Central Office

CoS Class of Service

CPU Central Processing Unit

CR Control Room

CSMA/CD Carrier Sense Multiple Access with Collision Detection

D/A Digital-to-Analog

DA Distribution Amplifier

DAB Digital Audio Broadcasting

DAC Digital-to-Analog Converter

DARPA Defense Advanced Research Projects Agency

DAT Digital Audio Tape

DB Connector type used in many computer interfaces, typically in the form DB-*n*, where *n* is the number of pins in the connector

dB Decibel

dBfs Decibels below digital full scale (also dBFS)

dBm Decibels referenced to 1 milliwatt (0 dBm = 1 mW)

dBr Decibels relative to reference level

dBu Decibels unloaded (0 dBu = 0.775 V)

DHCP Dynamic Host Configuration Protocol

Diffserv Differentiated Services

DIN Deutsches Institut für Normung (German Institute for Standards)

DMB Digital Multimedia Broadcasting

DNS Domain Name System

DRM Digital Rights Management; Digital Radio Mondial

DSCP Differentiated Services Code Point

DSL Digital Subscriber Link

DTMF Dual-Tone Multifrequency ("TouchTone")

DVMRP Distance Vector Multicast Routing Protocol

E&M Ear and Mouth

E1 European telco TDM data communication circuit

EBU European Broadcasting Union

EIA Electronic Industries Association

EMI Enhanced Multilayer Software Image (in Cisco Ethernet switches)

ENUM E.164 Number Mapping

EQ Equalization

ERL Echo Return Loss

EUL Enhanced Uplink

EVDO Evolution Data Optimized (also noted as EV-DO)

FDDI Fiber-Distributed Data Interface

FDM Frequency-Division Multiplexing

FEC Forward Error Correction

FTP File Transfer Protocol

FXO Foreign Exchange Office

FXS Foreign Exchange Station

G.7*nn* (n = any number) Nomenclature used by ITU-T for standard audio codecs (see Table 6.1 for examples and detail)

GBIC Gigabit Interface Converter

Gbps Gigabits per second

GPIO General-Purpose Input/Output

GSM Global System for Mobile Communications

GSM-FR GSM Full Rate

GUI Graphical User Interface

HE-AAC v2 High-Efficiency Advanced Audio Coding, version 2 (includes Parametric Stereo)

HE-AAC High-Efficiency Advanced Audio Coding (also known as AAC+)

HP Headphone

HSUPA High-Speed Uplink Packet Access

HTTP Hypertext Transport Protocol

HTTPS Secure Hypertext Transport Protocol

I/O Input/Output

IANA Internet Assigned Numbers Authority

IAX Inter-Asterisk Exchange (IAX2 is current version)

ICANN Internet Corporation for Assigned Names and Numbers

ICMP Internet Control Message Protocol

IEC International Electrotechnical Commission

IEEE Institute of Electrical and Electronics Engineers (also called I-triple-E)

IETF Internet Engineering Task Force

IGMP Internet Group Management Protocol

iLBC Internet Low Bitrate Codec

IM Instant Messaging

IP Internet Protocol

IPsec Internet Protocol Security

IPTV Internet Protocol Television

IPv4 Internet Protocol, version 4

IPv6 Internet Protocol, version 6

IRT Institut für Rundfunktechnik (Institute for Radio Technology)

ISDN Integrated Services Digital Network

ISDN-BRI ISDN Basic Rate Interface

ISDN-PRI ISDN Primary Rate Interface

ISM Industrial, Scientific, and Medical (radio bands reserved for unlicensed use by these industries)

ISO International Standards Organization

ISP Internet Service Provider

IT Information Technology

ITU International Telecommunications Union

ITU-R International Telecommunications Union, Radiocommunications (RF) branch

ITU-T International Telecommunications Union, Telecom branch

IVR Interactive Voice Response

IXP Internet Exchange Point

J1 Japanese telco TDM data communication circuit (same as *T1*)

kbps Kilobits per second

KVM Keyboard/Video/Mouse

L2TP Layer 2 Tunneling Protocol

LAN Local Area Network

LCD Liquid Crystal Display

LEC Line Echo Canceller

LED Light-Emitting Diode

LF Low Frequency

LFE Low-Frequency Enhancement

LW Livewire

LWCP Livewire Control Protocol

LWRP Livewire Routing Control Protocol

MAC Media Access Control

MADI Multichannel Audio Digital Interface

Mbps Megabits per second

MIPS Millions of Instructions Per Second

MMUSIC Multiparty Multimedia Session Control

MOH Music on Hold

MP3 MPEG-1 Audio layer 3 codec

MPEG Moving Picture Experts Group

MPLS Multiprotocol Label Switching

ms milliseconds

N/ACIP Norm/Audio Contribution over IP

NAT Network Address Translation

NI-1 National ISDN-1

NIC Network Interface Card

NTP Network Time Protocol

OC Optical Carrier (often followed by a number indicating bandwidth capacity, e.g. OC3, OC12, etc.)

OS Operating System

OSI Open Systems Interconnection

OSPF Open Shortest Path First

PBR Policy-Based Routing

PBX Private Branch Exchange (in-house business telephone system)

PC Personal Computer

PCB Printed Circuit Board

PCI Peripheral Component Interconnect

PCM Pulse-Code Modulation

PD Powered Device (in PoE context)

PDU Protocol Data Unit

PIM Protocol Independent Multicast

PLL Phase Lock Loop

PML Permitted Maximum Level

PoE Power over Ethernet

PoE+ Enhancement to PoE, providing higher power capacity

POTS Plain-Old Telephone Service (vernacular reference to PSTN)

PPM Peak Program Meter

PPP Point-to-Point Protocol

PPTP Point-to-Point Tunneling Protocol

PRI Primary Rate Interface (suffix to ISDN)

PSE Power Sourcing Equipment (in PoE context)

PSTN Public Switched Telephone Network

QoS Quality of Service

QPPM Quasi-Peak Program Meter

QSIG Q Signaling

RFC Request for Comment (IETF Standards prefix)

RIP Routing Information Protocol

RJ-11 Standard connector for POTS cables

RJ-45 Standard connector used in Ethernet network cabling

RMON Remote Monitoring

RMS Root Mean Square

ROHC Robust Header Compression

RS-422 EIA data communications standard (equivalent to ITU-T Recommendation V.11.A)

RTP Real-Time Transport Protocol

SAP Session Announcement Protocol

SBE Society of Broadcast Engineers

SBR Spectral Band Replication

SDI Serial Digital Interface

SDM Space-Division Multiplexing

SDP Session Description Protocol

SFP Small Form-Factor Pluggable (also called Mini-GBIC)

SIP Session Initiation Protocol

SLA Service Level Agreement

SMI Multilayer Software Image (in Cisco Ethernet switches)

SMPTE Society of Motion Picture and Television Engineers

SMTP Simple Mail Transport Protocol

SNMP Simple Network Management Protocol

SNR Signal-to-Noise Ratio

SOHO Small Office/Home Office

SS7 Signaling System 7

SSL Secure Sockets Layer

STP Shielded Twisted Pair

STL Studio-to-Transmitter Link

T1 N. American telco TDM data communication circuit (also called DS1)

T568A Wiring standard for RJ-45 connectors

T568B Wiring standard for RJ-45 connectors (preferred over T568A)

TCP Transmission Control Protocol

TDM Time-Division Multiplexing

TELNET Teletype Network (Internet remote virtual terminal control interface)

TELR Talker Echo Loudness Rating

TIA Telephone Industry Association

TLS Transport Layer Security

TOC Technical Operations Center

TSL Transmitter-to-Studio Link

UDP User Datagram Protocol

UGC User-Generated Content

URI Uniform Resource Identifier

URL Uniform Resource Locator

USB Universal Serial Bus

USOC Universal Service Order Code (pronounced "YOU-sock")

UTP Unshielded Twisted Pair

VDC Volts, Direct Current

VI Volume Indicator (commonly called "VU Meter")

VLAN Virtual Local Area Network

VoIP Voiceover IP

VPN Virtual Private Network

VU Volume Unit

W3C World Wide Web Consortium

WAN Wide Area Network

WiFi Set of wireless data networking standards defined in IEEE 802.11

WiMAX Wireless WAN standard defined in IEEE 802.16

xDSL Digital Subscriber Link (the "x" indicates any of a number of variations, such as ADSL [Asymmetrical DSL], HDSL [High-Speed DSL], etc.)

XLR Cable connector type widely used in professional audio for analog and AES3 signal paths

Index

Note: Page numbers followed by *'f'* indicate figures, *'t'* indicate tables.

L

M

N

O

P

Q

R

S

T

U

V

W

Z