A network-enabled radio console architecture

Michael Dosch
Telos System, Cleveland Ohio USA

catfish@telos-systems.com

Introduction

Radio broadcast audio mixing consoles have remained relatively unchanged for more than twenty years. Originally, source equipment connected to stand-alone mixing consoles with discrete analog signals. Later, the preferred method of interconnection became AES/EBU digital. More recently, high-end broadcast consoles have begun to offer proprietary centralized mixing and routing engines which make possible the sharing of sources between studios.

Using modern computer networking equipment, it is now possible to build robust Networks capable of transporting digital media signals throughout a complete studio facility. This paper describes various console models, outlines the advantages offered by a studio Network and explains how future broadcast equipment— most notably mixing consoles— will need to change in order to fully exploit these advantages.

Sources are different now

The audio mixing console has long been the central processing and control device of the radio studio. Despite a trend toward digital processing, the basic architecture of the console has not changed in more than twenty years. Audio source equipment feeds the console analog or AES/EBU audio. The user mixes live and recorded elements and the outputs of the console feed the transmission chain and other destinations. This approach is heavily dependent on the user to push the right buttons at the right times so as to deliver the appropriate content. And sharing sources between studios is difficult. The stand-alone console is ideal for dedicated studios that can be set up for a certain show type and left unchanged.

Newer console designs have begun to offer integrated routing switchers using proprietary centralized mixing/routing time division multiplex (TDM) engine cores. These systems offer significant advantages over stand-alone console designs. Because all studio sources are connected to a central core engine, it is possible for sources to be shared by multiple studios. Further, because the mixing and routing is performed centrally, the studio console interface is a flexible control surface that can be reconfigured in software to accommodate changing show types, shared resources and the instant recall of user preferences and settings. The centralized mixing/routing engine approach reduces costs when compared to stand-alone mixing consoles due to reduced wiring costs and a consolidation of expensive components.

While these advancements offer benefits to the modern radio plant, even the most advanced consoles of today seem to ignore the now central role played by the personal computer (PC). Most broadcasters are using PC's to replace many other studio functions—particularly audio source equipment. Gone are the days of playing from CD, carts, vinyl, cassette and reel tape in a typical broadcast. Most program audio is now recorded, edited and played out of a PC system.

While consoles remain much the same, the PC has quietly taken center stage in today's radio studio. Traditional consoles handle PC audio the same as any discrete source, hindering potential intercommunication that might enhance accuracy and efficiency. Instead of using analog or AES/EBU audio as the interconnection standard, we believe broadcast audio systems of the future will use networked Ethernet to provide a much more flexible and cost-effective alternative to console systems used today.

Why Ethernet?

With traditional consoles, the PC uses sound cards to feed analog or digital audio to the console. In a complex studio, it may be desired to play many audio elements from the PC simultaneously. While modern PC's are capable of playing multiple simultaneous audio streams, the sound cards can often be a limiting factor. With Ethernet, the PC does not need sound cards. Rather, it passes the audio directly to the network via a standard network interface connection (NIC), eliminating the expense and compatibility issues associated with sound cards.

Ethernet can be carried over standard computer networking devices such as switches, hubs and routers. These networks can be easily scaled from small single-studio installations all the way up to the most advanced consolidated multi-station, multi-studio facilities.

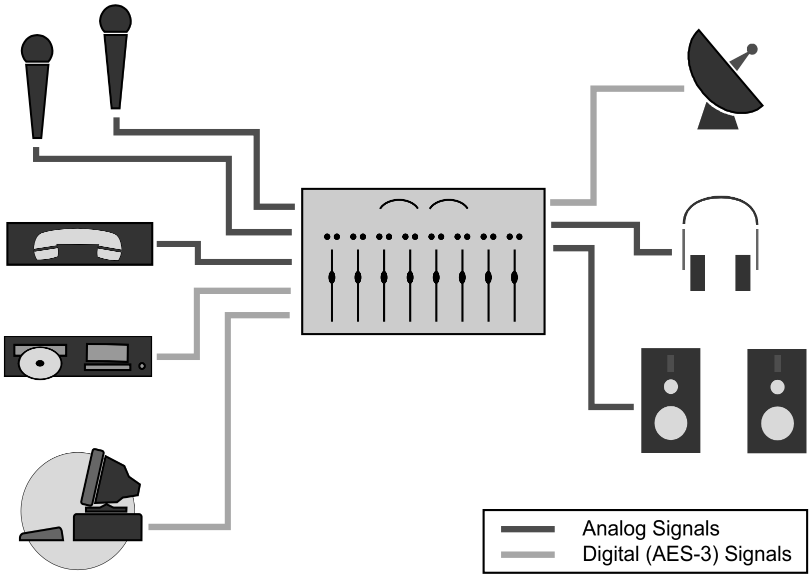


Figure 1. Simplified diagram of a traditional radio studio using discrete audio connections.

More importantly, Ethernet is *information rich* meaning that associated data can travel the same path as the audio. As broadcasters continue to embrace digital audio broadcast (DAB), there will be a need to convey content-related data to the transmission chain. An Ethernet will carry both audio and associated data on a single connection to any destination.

An Ethernet provides device-independent flexibility. Sources and destinations are network resources, as are mixing engines, storage devices, processors, and other types of peripherals. Because of this, an Ethernet is easy to install and maintain. Once a device has been connected to the network, it is now an available resource to be used as the engineer wishes. Sharing devices across studios on a permanent or temporary basis will no longer require wiring changes.

What's wrong with the way it is?

Discrete analog and digital connections to a broadcast console have worked well for years. Some might say this approach is not broken and shouldn't be fixed. Indeed, there are some very sophisticated facilities running some of the most complex shows on stand-alone consoles from PR&E, Wheatstone and others. But we must carefully consider the changing technology of radio, particularly the capabilities of DAB, as we evaluate whether the old model will be able to meet future needs. Figure 1 shows a (greatly) simplified diagram of a typical radio studio connected to a traditional console. Analog and digital sources are connected to the console where they are mixed and routed to the program output. The operator chooses the sources— including computer audio— and selects levels to produce a live show.

Even though the PC is providing most of the recorded audio and the playout software can log what is played, it is quite possible for user errors downstream to render the log useless. For example, the PC may have played a spot while the console fader was down, the channel not assigned to program, another source was being played simultaneously encroaching on the spot, etc.

Because sources are tied to the console, they are not easily shared by other studios. And with only analog or AES/EBU connections, any provisions for program associated data will need to be made separate from the console, complicating the system design. For example, how does the system know if a source is feeding the program chain or is simply being auditioned locally?

Despite its limitations, this is by far the most popular radio console model in use today, and is quite satisfactory for many applications.

The more sophisticated designs of today, provide a centralized mixing/routing engine as depicted in Figure

2. The central engine core performs all the switching, mixing and console processing for a group of studios. Sources can be shared across studios. For very large plants, multiple engines can be ganged together and some have special provisions for dealing with localized studio sources.

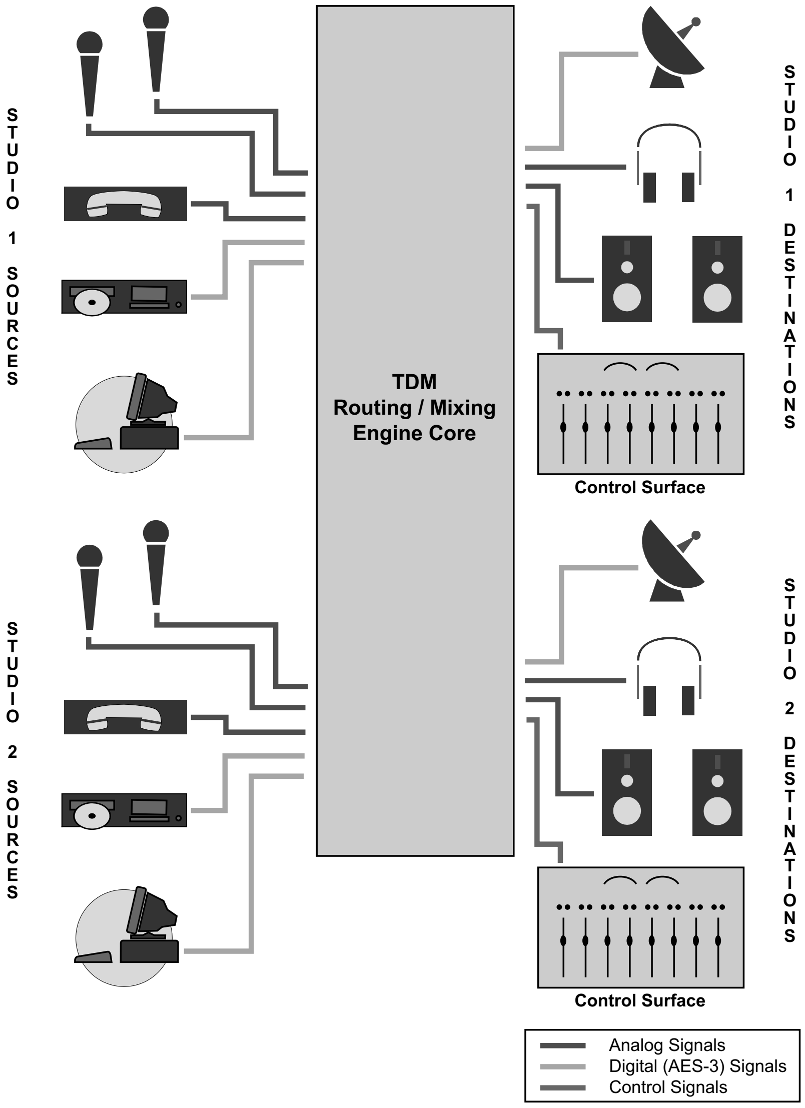


Figure 2. Simplified diagram of a centralized router/ mixing engine core.

Control surfaces provide the user interface, but perform no actual audio processing. The user interacts with a surface much the same as they would an actual console, but rather than changing the audio directly, their input is captured and fed to the central engine core to change levels, switch signals, etc. Control surfaces can be reconfigured quickly to accommodate different shows or user preferences.

This approach provides much more flexibility than the stand-alone console model previously described. Wiring costs are greatly reduced and studios can be more efficiently utilized. Perhaps the most significant benefit is the seamless integration between routing and mixing functions. Each input channel can select from a range of available sources. Outputs and monitor preferences are stored for instant recall when launching a show. Yet with all this integration, all of the audio is *still* treated as discrete. This is especially limiting for the PC which must use sound cards to convert its audio to analog or AES/EBU streams before feeding it into the engine core. An Ethernet audio network can provide all of the benefits of the centralized core approach while adding a wide range of new capabilities.

Why use computer technology?

The computer industry has advanced the state-of-the-art in computer networking, routing and switching systems. It is now possible to transport digital media signals reliably over controlled Ethernet audio networks with guaranteed quality of service (QoS).

Studio audio in the broadcast plant is especially demanding. It is not enough that the network be capable of reliably delivering audio packets. The delivery method must provide for synchronization, absolutely no information loss, and extremely low delay (latency).

By carefully specifying the network components, system design and transport protocol, it is possible to build a low-latency, no-loss, synchronized Ethernet audio network using a combination of commonly available Ethernet and PC components and some purpose-built broadcast pieces.

Additionally, because the underlying network is Ethernet, PC's can connect directly to the network without any translating hardware. Ethernet cables, plugs, tools, testers, hubs, and Ethernet adapters are ubiquitous and inexpensive. By building the studio infrastructure using these elements, broadcasters are able to access advanced technology with costs driven lower by the high volumes of the mainstream computer networking markets.

What about traffic?

An Ethernet audio network must manage traffic more intelligently than the typical office LAN which routinely drops packets and uses TCP/IP to throttle the speed of the source to deal with variable network congestion. While this method works fine for web browsing, email and print jobs, the penalty for this method of delivering audio is very high latency due to large buffers, audio drop-outs, or both.

The best way to solve this problem that we have found is to use switching Ethernet hubs to prioritize audio streams for reliable transmission and to control the flow of traffic. In an ideal system, high-priority audio can be conveyed over the same Ethernet segments as standard TCP/IP or UDP/IP control or file transfer data.

The switching Ethernet hub is ideally suited for an audio network. In Figure 4, six workstations are connected to a switching hub using 100BT Ethernet segments. Each segment is capable of carrying 24 inputs and 24 outputs (linear PCM, stereo 48kHz sampling rate, 20 bit resolution) simultaneously. So in the simple example shown, this network provides a 144 by 144 cross-point matrix. The switching Ethernet hub performs two vital functions for the Ethernet audio network.

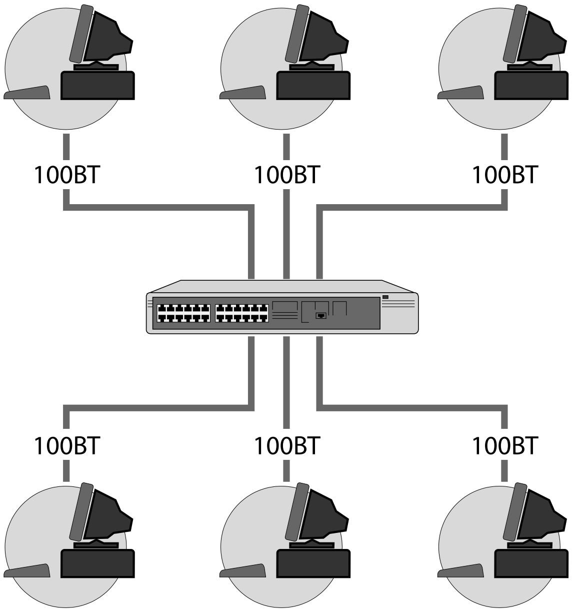


Figure 4. A switching hub is used to prioritize and manage network traffic flow.

First, it divides the network into independent Ethernet segments, each capable of carrying a full payload of traffic. It does this by sending only those packets intended for a particular segment. With a properly written protocol and careful system design, it is possible to completely eliminate network congestion and contention.

By contrast, a standard (non switching) hub will cause the connected devices to share bandwidth; relying on the connected devices to ignore the unnecessary packets. Without a switching hub, the six workstations above would share a single 100BT network connection limiting the matrix to 24x24 total inputs to outputs at best.

Second, the switching hub prioritizes the data. This feature is what allows the Ethernet audio network to also carry lower-priority associated data without concern that these additional packets will affect the delivery of time-critical audio packets. In fact, it is possible to set multiple levels of priority for maximum reliability and efficiency.

For example, in a broadcast studio, we could set live elements like microphones to the highest priority, computer audio sources to medium priority, and logic signals and PAD to low priority. By prioritizing traffic this way, it is possible to deliver live audio with minimal latency and still allow other traffic on the same net. With switching hubs and a well designed protocol and system, a broadcast-capable Ethernet audio network is possible.

Why is low latency so important?

The traditional console model provides for very low input to output delay. This is a critical requirement for a live-format broadcast console in which the announcers will typically monitor their own voices in headphones.

Studies have shown that total mic to headphone delay in excess of 30ms will cause live monitoring to become distracting if not impossible. Delays between 15 and 30ms produce an annoying comb effect. Ideally, a console system would have much lower latency, perhaps less than 10ms total.

A 10ms latency budget disqualifies most network methodologies, even those which purport to offer low-latency delivery. The problem is that even the low-latency protocols— even those intended for media use— will add at least 5ms per network hop. Multiple network hops are required in even the simplest systems.

To gain acceptance by broadcasters, networked audio systems will need to provide latency performance in the range of 1ms per network hop. The other system components will also need to be designed for speed. It does little good to have an ultra-fast network, only to have huge buffers in the mix engine adding tens of milliseconds to the round-trip delay.

The essential components of a network-centric radio console are shown in Figure 5. Each component will add some delay to the overall chain. The good news is that with careful design and some clever application of technology, it is possible to build an Ethernet audio network capable of delivering real-time signals with minimal latency. In fact, it is possible to build an entire studio network with port-to-port throughput times that can rival the traditional console.

So analog sources are networked?

Every source and every destination should be made available to the network as a resource. Every microphone, tape machine, satellite feed or CD player used in the broadcast plant needs to be connected to the network.

In order to be useful, an Ethernet audio network will need to have provisions for converting analog feeds to packets and back again. Professional-grade A/D/A conversion would logically be bundled together with the adapters. It would also be beneficial to have network-addressable GPIO interfaces to start and stop sources and to provide remote control capabilities.

What about digital sources?

Again, an Ethernet audio network must be able to interface with discrete digital sources and destinations. Because AES-3 is a universally-accepted standard for transporting linear PCM audio, translation between AES/EBU and network would be required for certain devices.

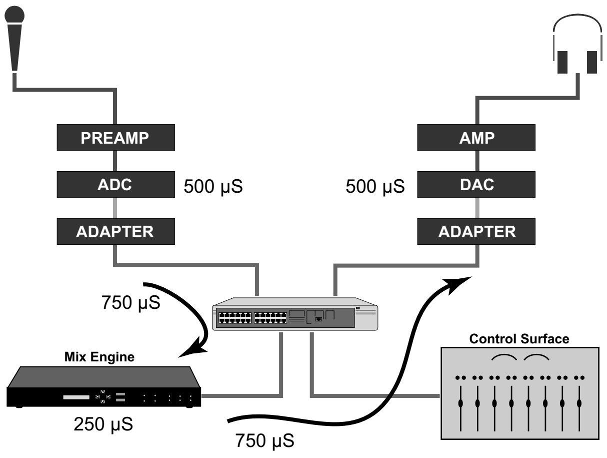


Figure 5. For radio systems, total delay must be well under 10ms for live audio signals.

In an ideal future, every device would be equipped with an Ethernet adapter and would be capable of transmitting and receiving properly formatted packets directly. We believe that the benefits of Ethernet will drive many broadcast equipment manufacturers to replace or supplement their AES-3 digital connections with network ready Ethernet jacks in future designs.

And to reiterate an earlier point: most recorded audio in the modern broadcast plant originates in the PC. IP allows the PC to speak directly to the network through its NIC— no sound cards required.

Is this scalable?

The overall bandwidth of a switched network scales with the size of the network (more bandwidth is added as the network grows). This means that bandwidth does not limit the number of channels that can be supported network-wide. There is virtually no limit to how large or complex a network can be built using this approach.

What may be surprising though is how cost-effective an Ethernet audio network can be for small, simpler installations. Even a one or two studio facility will benefit from the ability to share sources, direct connect to PC's, transport associated data and wire everything with inexpensive Ethernet cables.

Studio systems can be built as stand-alone clusters, each with its own central switching Ethernet hub. Interconnecting multiple studios can be accomplished via one of the switched Ethernet segments. Although 100BT Ethernet is ideal for local shared sources, some broadcasters may wish to connect the studios together using a 1000BT copper or fiber link.

Where is the cross-point switcher?

Perhaps one of the more interesting attributes of the Ethernet audio network is its ability to provide the functions of a cross-point audio switcher— without any additional cost. In the networked audio system, every audio source and every audio destination is available on the network, eliminating any need for a dedicated cross point audio switcher.

Some larger facilities use expensive, proprietary cross-point audio switchers to share sources and reconfigure destinations. These traditional routing switchers can easily cost more than $50,000US for a typical plant. And while these routers are competent at routing analog or AES/EBU discrete signals, an Ethernet audio network is superior for most modern radio plants with mixed analog, digital and computer-generated signals.

Figure 6 shows a simplified cross-point switching example. Analog and digital sources are converted to digital and interfaced to the network as high-priority multicast streams, available to all interested destinations. Connections are made by simply having the destination (output) terminal adapter request a source stream. This could be done locally with a simple user interface on the terminal itself or with a configuration application.

Any audio workstations on the network can "direct connect" via Ethernet; no sound cards required. Audio from the workstations will be IP-standard, medium priority, and can feed the same destinations as the high-priority live streams. This system is much more flexible than the traditional audio cross-point switcher at only a fraction of the cost. And unlike the proprietary cross-point switchers which are prohibitively expensive for the smaller station, an Ethernet audio network is cost-effective for very small systems— as small as only a few devices— while being able to scale up to meet the needs of the largest facilities.

Some facilities may choose to use audio networks to simply replace the function of the cross-point routing switcher, connecting to traditional consoles and source equipment. Even in this application, the network approach offers key benefits over traditional approaches.

Modern broadcast plants have a mixture of local and centralized sources and destinations. CD players, microphones, headphones and speakers are mostly local to the studio while audio servers, satellite receivers, transmission feeds are usually in the central terminal room.

The traditional cross-point switcher is often a central resource. Studio sources and destinations must be connected back to the central device. This can be done with either multiple discrete audio cables or some type of proprietary studio connector interface device. Both approaches add cost to the already-expensive cross-point audio switcher.

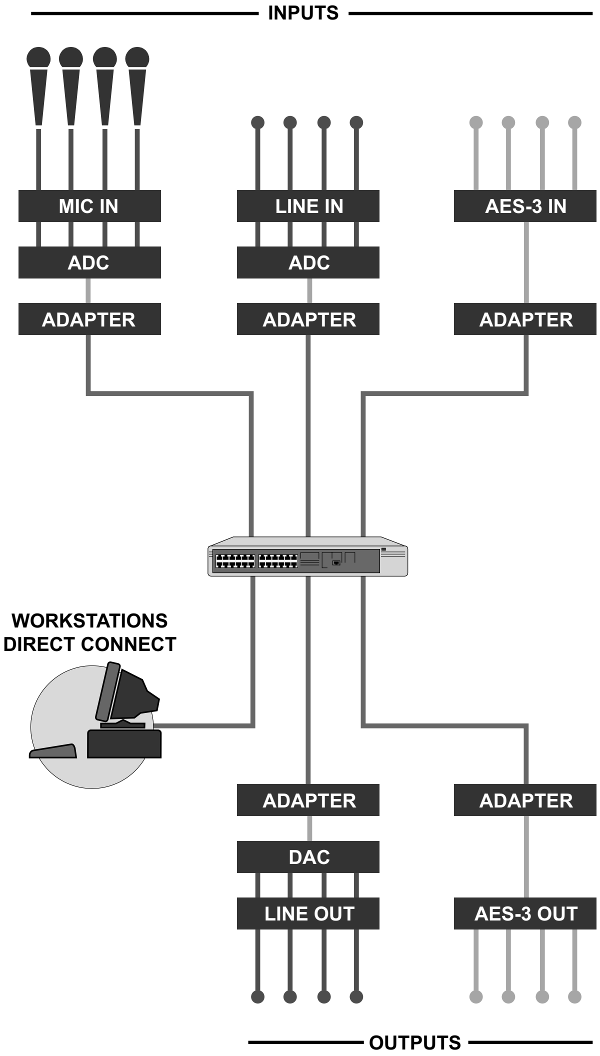


Figure 6. An Ethernet Audio network configured as a crosspoint audio switcher.

The networked audio approach allows conversion terminals to reside near their sources and destinations. Terminals can be located both in studios and in central rack rooms. Switches can also be distributed around the facility or centralized. Even workstations can be central, local or both. Everything is connected together with standard low-cost Cat-5 cabling.

While the Ethernet audio network makes an excellent replacement for the traditional cross-point switcher, much more is possible once we establish the network infrastructure. In particular, if we are to add a device to manage the mixing and routing of signals on the network, we can also replace the traditional console.

A PC-based mixing engine?

Having established that all of a facility's sources and destinations can be networked, let's now address the need for mixing and processing. Ideally, a mixing engine would be attached to the network and would receive the desired streams and would perform any mixing and signal processing necessary and send the result to the appropriate destinations.

Most of today's digital mixing console engines— both stand-alone and the centralized router/engine types— are based on proprietary DSP architectures. While these designs are satisfactory for the discrete audio studio of the past, the networked approach makes possible a different architecture, one based on the power of the modern PC motherboard.

A Pentium-4 equipped motherboard is an amazingly powerful device, with processing power comparable to large multi-DSP proprietary embedded systems. In fact, the PC engine is much better suited for mixing in a network-centric facility than proprietary engines. All the connections into and out of the engine are made via Ethernet.

Of course, most PC motherboards are burdened with slow, general purpose operating systems and inefficient applications. To make an effective mix engine, the PC must be optimized for this purpose, with an efficient and reliable operating system capable of handling real-time processing tasks (such as real-time Linux) and tight, efficient application code. In order to keep the overall system latency under our 10ms maximum, the engine will need to receive, mix, process and distribute live streams within a millisecond or two. Although challenging, this too is possible with careful design. Needless to say, this PC engine must be dedicated to perform the engine functions exclusively.

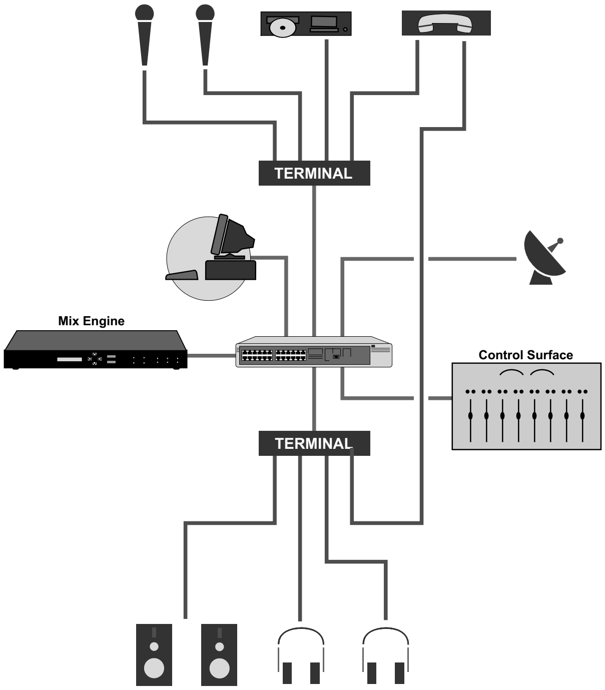


Figure 7. A PC-based mixing engine completely replaces traditional console functions.

In the network-centric architecture, the mixing engine is an available resource just like the sources and destinations themselves. It costs only a fraction of what a proprietary mixing engine would, again taking advantage of computer industry volumes to make technology more accessible. The low cost and wide availability of the PC-motherboard makes this engine architecture much easier to acquire, maintain and upgrade than traditional approaches.

A simplified studio mixing system is shown in Figure 7. Analog and digital discrete sources are converted to digital live (high-priority) streams and fed to the network. The mixing engine sweetens and mixes these streams and feeds the result to the appropriate output destinations, based on a configuration template and live-input from a control surface or user application.

A single P4 engine is capable of supporting a very complex studio setup, with 24 or more active sources, multiple program outputs, monitor outputs, mix-minus outputs, auxiliary sends, talkback paths, etc. Amazingly, this PC-engine can outperform the very largest multi-bus, multi-channel, stand-alone consoles used in radio today.

Due to the tremendous amount of latent power in the P4 motherboard, the PC-based mixing engine is capable of adapting to a wide range of situations without any hardware changes. One studio setup might have a dozen or more live sources, each with independent mix-minus output requirements. Another setup might use 6 or 8 computer-sourced IP streams and several different control surfaces. The PC-based mixing engine adapts to the needs of the studio instantly and effortlessly.

Further, it is possible to integrate external functions into the engine. Many consoles will use external effects devices, equalization, profanity delays, headphone dynamics processing, and other specialized functions. A PC-based mixing engine can assign resources to provide these and other functions that might otherwise require dedicated equipment.

The engines can be located in the studios or in the terminal rooms, stand-alone or shared. The networked broadcast plant requires an entirely new way of thinking about systems architecture, but once our minds are open to the possibilities, it is easy to see how powerful and flexible tomorrow's systems will be.

Is all this really possible?

The concepts described here are more than interesting theory. Telos has in fact developed a studio audio transport system called *Livewire,* a suite of audio networking tools which will forever change the way we connect and use studio audio equipment.

The *Livewire* network uses a common Ethernet to carry audio streams and any associated data or control between devices, studios and facilities. At its heart *Livewire* uses Ethernet switches to isolate links, manage traffic and ensure fully reliable transmission.

Livewire assigns the highest priority to live audio streams (called Livestreams) for delivery in less than 1ms per network hop, while also providng an IP-Standard medium-delay mode for connection to PC's. It distributes a clock signal over the Ethernet for precise synchronization and low delay.

The *Livewire* system includes translation terminals for microphone audio, line-level analog audio, and AES/EBU audio for connection to traditional equipment. These terminals provide the synchronization and advertise the availability of connected sources to the rest of the network and can be located physically near their associated gear.

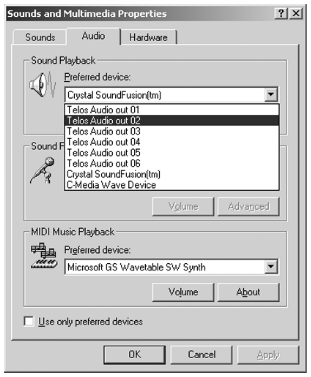
A specialized Routing Controller terminal provides a list of available streams which can be scrolled and selected or instantly accessed via softkeys. It connects to Livewire and provides convenient audio input and output ports.

The Livewire system provides a unique way of handling audio from PC's using a software driver that causes the network to look like a sound card to the PC application. Equipped with this driver, the application will pass audio to and from the network seamlessly.

A PC-based Engine running Linux and a highly-tuned application mixes and processes Livewire streams while adding less than 1ms of throughput delay. The Engine adapts to changing studio requirements and has sufficient processing headroom to allow for "accessory" features like built-in headphone dynamics processing and channel equalization that might require add-on devices in a traditional studio system.

Telos offers control surfaces to provide the tangible user interface (UI) for the board operator, with intuitive controls and displays designed for the fast-paced live format radio show. These surfaces communicate to the Engine and other devices over the Livewire.

Livewire terminals interface to analog and AES-3 gear.



Sound Playback

Preferred device:

Crystal SoundFusion(tm)

Telos Audio out 01
Telos Audio out 02
Telos Audio out 03
Telos Audio out 04
Telos Audio out 05
Telos Audio out 06
Crystal SoundFusion(tm)
C-Media Wave Device

Sound R

MIDI Music Playback

Preferred device:

Microsoft GS Wavetable SW Synth

□ Use only preferred devices

A software driver makes Livewire look like a sound card to PC's.

Control surfaces communicate over the same network.

A PC-based Engine mixes and processes Livewire streams.

Putting it all together

Shown on this page is an example studio system using Livewire components. In this example, the studio has a large number of active local sources. Each microphone has an independent monitor feed which enables the host to talk to each guest's headphones privately.

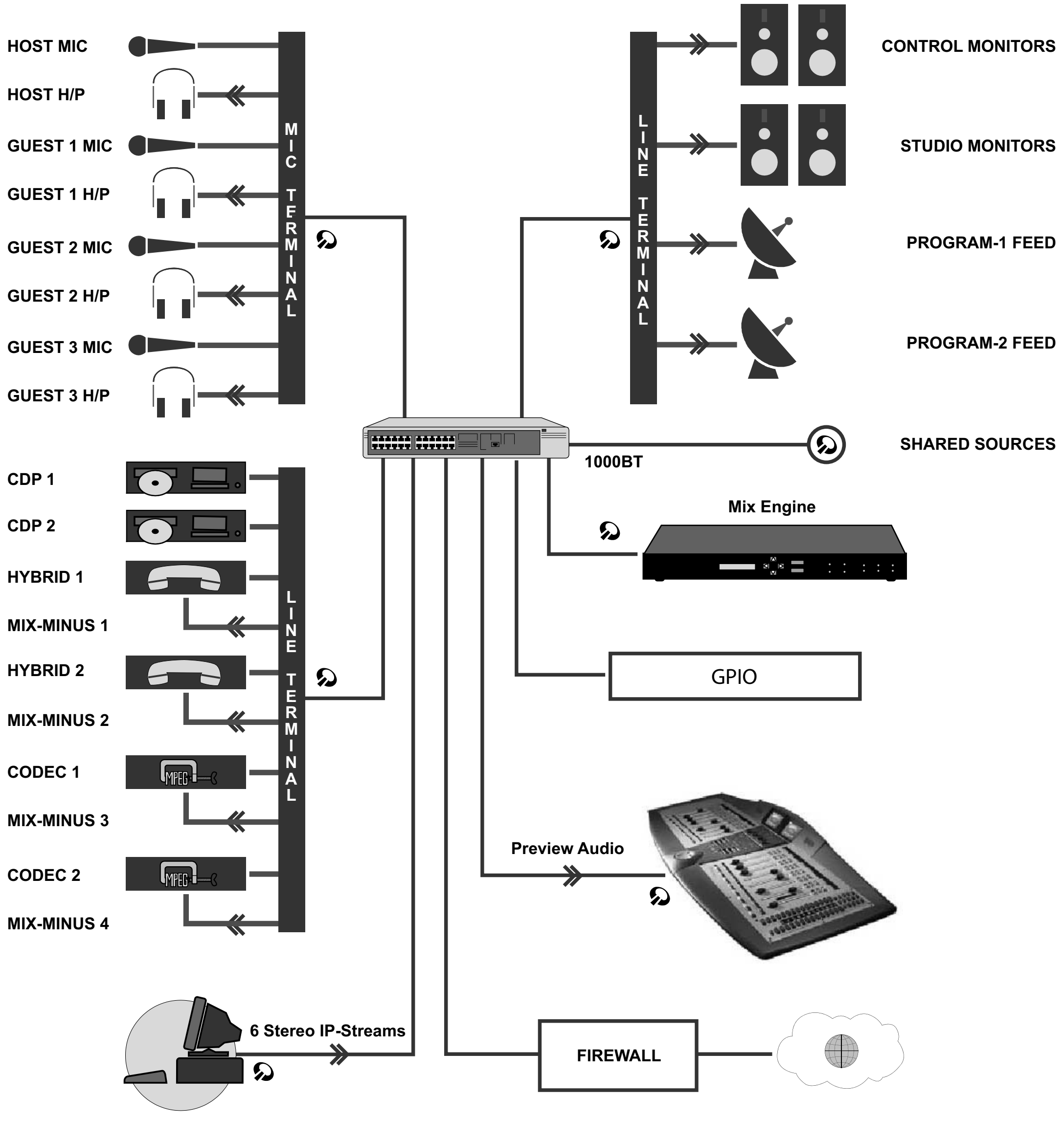
The phone and codec sources each have associated mix-minus outputs. In fact, due to the Engine's ability to assign resources as required, it is possible to have a mix-minus output for every assigned source. And the management of mix-minus outputs is handled completely within the Engine automatically, finally making hybrids and codecs as easy to use as CD players.

The audio delivery software is directly feeding the network with 6 simultaneous stereo audio sources. Additionally, the Ethernet switch is linked to other studios and centralized sources and is also making these local sources available to other interested studios.

In this example, the traffic is light and the local Engine working well below its capacity. There are 10 local Livestream sources, 6 IP-Audio sources and 13 local destinations. Any program associated data is carried through the network along with the audio data and can be delivered to interested devices by simply connecting them to an unused switch port.

A GPIO terminal is shown which provides for remote control and contact closure commands for microphones and discrete peripherals.

In this drawing, we even show a firewall-protected internet connection. The idea of allowing internet traffic onto a critical audio network would be terrifying were it not for the traffic management features of the Ethernet switching hub. Because of the priority placed on Livestreams over IP-Standard audio streams over everything else, Livewire ensures that even on a busy network, audio comes first.



The Future

Some will be uncomfortable with the idea of computer networking technology for audio delivery. Proprietary embedded systems may feel more industrial and secure. What's more, we have all had our share of bad experiences with computers and networks. We groan at the thought of "rebooting" our consoles. For good reason of course.

In order to be accepted by broadcasters, Livewire— or any other audio networking approach for that matter— absolutely must provide the highest level of reliable operation. This is our programming we're talking about. The office printer can be off line for an hour while we hunt down the IT expert. The station audio must be uninterrupted.

We believe that the future will clearly prove, despite some initial apprehension, that studios built around audio networks will provide high reliability, cost efficiency and greatly enhanced studio operations. Once networking begins to gain acceptance, we should see other significant changes.

We described here a console engine which hangs on the network intercepting streams, mixing, processing and presenting the result back to the network for interested destinations. It is easy to imagine future broadcast products equipped with Ethernet audio connections to be addressed and shared throughout a facility.

And as Moore's law continues to drive PC MIPS up and prices down, the network-enabled radio Engine will very soon have excess capacity that could be tapped for alternative tasks. Software plug-in products to do voice processing, program delay or even codec or hybrid functions may eventually replace the need for stand-alone broadcast gear.

Broadcast technology has always been driven forward by advancement in the communications and computer industries. PC's replaced broadcast carts. Digital Signal Processing replaced analog functions. And each technology advance brought with it new standards of performance and new operating possibilities.

We can now apply computer networking to the broadcast plant in ways never before possible. Discrete point-to-point wiring and TDM mainframe-type engine cores will soon seem antiquated once broadcasters begin to experience the benefits of the networked audio plant.