The ACN Future is Here

How a new protocol may dramatically change the way we work

By: John Huntington

In the June 2004 issue of *L&SA*, I wrote about the sophisticated, network-based entertainment control protocol, ACN (or Advanced Control Network), which was being developed by the Control Protocol Working Group (CPWG) of the Entertainment Services and Technology Association (ESTA). In the article, I laid out some possible applications for ACN, and also urged the industry to handle the ACN rollout carefully. That rollout is now underway, as the standard was finally completed late last year, released with a new (and more future-proof) name: E1.17-2006—Architecture for Control Networks.

I've been following the entertainment control market for more than 20 years, and ACN is by far the most complex protocol I have ever attempted to understand and explain. However, that should not discourage you from using ACN. The complexity exists in order to make the system as simple and as plug-and-play as possible for users. This time, I'm writing to urge the industry to adopt and implement ACN as widely as possible. Why? Because ACN really could radically improve the way we interact with show devices, opening the door to a new generation of storytelling tools and techniques. To use the oft-overused phrase, ACN may represent a paradigm shift for our industry.

A paradigm shift—really?

With ACN, the term "paradigm shift" isn't necessarily hyperbole. Just imagine a few of the possibilities that it could create:

For the designer:

- Use moving lights that precisely and automatically track the position of moving scenery.
- Allow a performer's actions to determine which of a range of designer-preselected colors is displayed on stage.
- Watch a touring show's console live over the Internet.
- Create a moving light hot backup that automatically takes over for another in case of a failure.
- Use a video clip or a lighting fade that dynamically tracks the tempo of accompanying music.
- Connect multiple consoles from different manufacturers that can independently control the same dimmer rack.
- Allow a sound image to track a moving light (or vice versa).

For the technician:

- Configure every device in your system using only names that make sense to you (no more DMX universes or slots).
- "Discover" a huge lighting system in less than a minute.
- Load and configure a list of clips into your video server directly from your console, while simultaneously loading clip media into the server over the same network connection.
- Allow the factory to diagnose a problem in your system over the Internet.
- Let your moving lights inform you of a stuck color wheel or a blown lamp before you bring the light up onstage.
- Create a 100% full tracking console backup, with automatic switchover, using just a network connection.

Of course, all of these things would be possible with custom systems or maybe even those sourced from a single manufacturer. ACN, however, makes possible sophisticated cross-manufacturer and cross-discipline interoperability, creating better market conditions to support these kinds of advancements on a wide scale.

What is ACN, anyway?

The market that spurred the development of ACN was one we have seen time and time again: Multiple, non-interoperable control approaches from a number of manufacturers vying in the marketplace, with users crying out for systems that can talk to each other. In this case, the non-interoperable systems were Ethernet-based lighting data transmission systems from several lighting manufacturers, none of which could communicate with the others. Interoperability, as we have seen with protocols like DMX and MIDI, generally causes markets to grow, and interoperability, is, in fact, ACN's number-one design goal. While ACN's development was certainly driven by lighting manufacturers, potential applications for it extend into many other markets. The development task group, originally headed by Steve Carlson, of High Speed Design, and later by Dan Antonuk, of ETC, wisely made every effort to ensure that ACN can be applied as broadly as possible. That effort apparently paid off, since Shure, a leading audio manufacturer, recently embraced ACN as a control platform.

The ACN standard offers an open and sophisticated infrastructure for communication with, and control of, virtually any device connected via a network. (ACN is network-independent, but full-duplex, switched Ethernet is, of course, the current platform of choice.) ACN "controllers" configure, monitor, and control "devices," by "getting" and "setting" "properties" over the network. While ACN can certainly carry DMX-style data, it is very different from a simple point-to-point protocol like DMX, or even from previous DMX-over-Ethernet approaches, because it offers the system designer a sophisticated, plug-and-play, scalable network infrastructure with key features not offered by Ethernet alone (more on this later).

In addition, one of ACN's paradigm-shifting benefits is that each type of component can be controlled in a way optimized for that device. For instance, a moving light might present its pan-and-tilt axis properties in degrees, with three decimal places, while simultaneously accepting XYZ target coordinates for the fixture, presented in meters relative to a zero point onstage. The same luminaire might also present a list of the names of installed gobos, while allocating a simple on-off setting for its shutter, and a percentage with one decimal place for its dimmer. A video server, on the other hand, might present a table full of its available clip names and disk directories, with its current clip time code presented as a continuously updating time value. The server's Alpha channel might be presented as an integer value, while a rotation parameter could be allocated in degrees accurate to two decimal places, and its current operating temperature presented in degrees Celsius. In ACN, control for each device is completely bi-directional and completely customize-able, and, therefore, is optimized; no longer does anyone have to shoehorn sophisticated control parameters into a number of unidirectional eight-bit slots (as in DMX or RDM), or seven-bit integers (as in MIDI).

Another key development of ACN is "discovery." A protocol like DMX uses simple point-to-point broadcast technology, so, with a DMX-based system, electricians have to set the receiving "address" of every device in a system, figure out to which DMX "universe" each device should be physically connected, resolve any conflicts, and patch and assign all those units. This can require an enormous effort and some incredibly tedious paperwork, especially on larger shows, where multiple consoles are often required simply to get enough DMX outputs. MIDI and other similar protocols work the same way—all the data is broadcast, and it's up to the system designer to manage the channels, individual connections, etc. ACN, on the other hand, automatically "discovers" any number of devices connected to the same network, and a console can then present them to the user for configuration and patching. If the network is properly administrated, this speedy process works the same for two connected devices or 2,000.

It's working now

ACN is so new that it's difficult to get your hands on actual ACN gear, so I convinced *L&SA* to fly me out to ETC's headquarters in Middleton, Wisconsin, to see real ACN products in action. My hosts were Dan Antonuk, network products development manager, who has chaired the ESTA ACN task group for the last five years; Steve Terry, vice president of research and development; and Fred Foster, ETC's CEO and founder.

For ETC, ACN is not some bright idea for the future; it's a functioning, important part of the company's product line today. "For a number of years," says Terry, "it has been a priority at ETC to achieve industry-wide interoperability for networked products, in much the same way that DMX512 did with consoles, dimmers, and moving lights in the '90s. That's why ETC has committed time and resources to the ACN effort, which is now bearing fruit due to the efforts of many companies over many years. It's not altruistic on our part—we believe that ACN will be the commercial future of our industry."

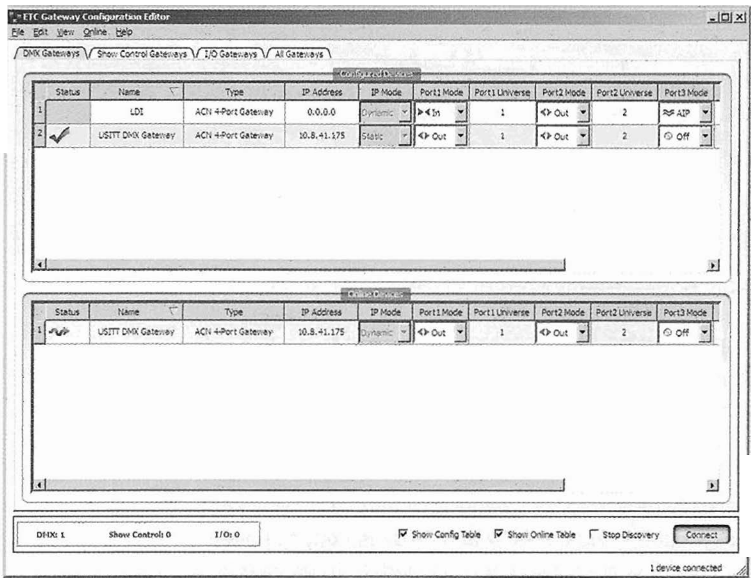
And ETC has put its money where its mouth is: While touring the shop, I saw more than 200 ACN-to-DMX gateways undergoing burn-in testing before being shipped out to a (very large) job.

ACN nodes being tested.

These DMX gateways were connected for testing using standard full-duplex Ethernet switches and Cat 5 cables, and, using "streaming ACN," all the nodes were getting pounded with worst-case signals (all channels fading simultaneously), and not a glitch was visible in the LED output panels used for testing. This type of application, bridging the worlds of ACN and DMX, is likely to be a common first application of ACN, and, to facilitate this, under development as of this writing within the ESTA CPWG is E1.31: Lightweight Streaming Protocol for Transport of DMX512 using ACN. This protocol will likely include the DMX concepts of universes, slots, and addresses, as its whole purpose is to transport DMX data over the ACN network in a simple and easy-to-use way.

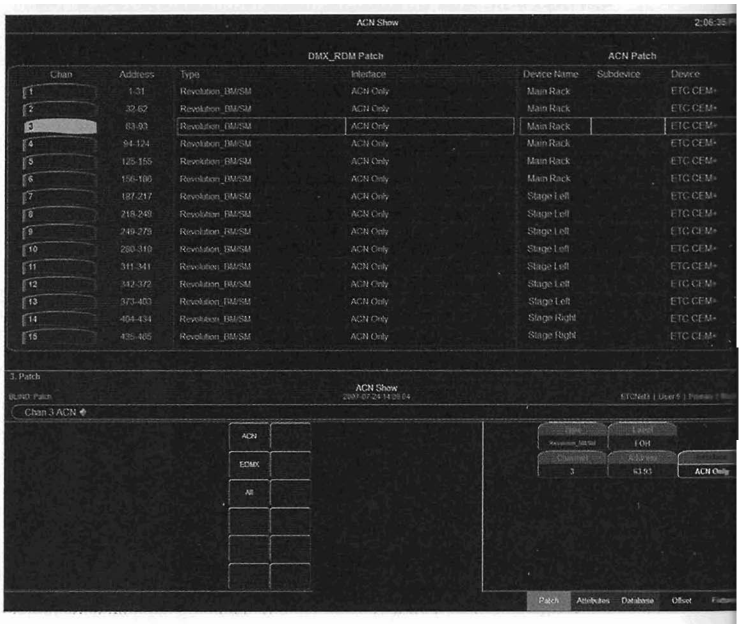
ETC's Eos consoles were designed with ACN architecture from the ground up, and Antonuk showed me some of these consoles working with ACN as well. These consoles can not only speak and listen to external ACN (I saw them communicating with the DMX gateways), but they are also using ACN today as a protocol backbone for internal operations. For example, multiple Eos consoles (which were designed with ACN in mind) can be connected together over the network. Each console can be configured to operate the system independently and simultaneously (although, to avoid conflicts, only one console is actually talking to the connected devices at any given time, and the other consoles are speaking to that master console). If the master console goes down, a full, complete tracking and hot backup machine can take over within a few hundred milliseconds. And all of this is accomplished using ACN as the communications backbone, so the only connection is a standard Cat 5 cable.

While show networks should be kept as closed and stripped-down as possible, ACN was designed to be well-behaved and interoperable over any sort of network. In fact, Antonuk demonstrated device discovery via ACN over the ETC corporate network—he found several gateways in offices spread throughout the ETC R&D department. (Entertainment technicians better start learning about routers and firewalls now!) Those devices are displayed using a screen like this:



ACN discovery.

At this point, ETC has a working ACN-to-DMX gateway (200 or so of which are shown in the previous photo), an I/O gateway (for relays, analog inputs, and RS-232), a show control gateway that handles SMPTE and MIDI, and, this fall, will have a full ACN dimmer rack. In the future, of course, we hope to be able to automatically discover a complete range of devices from a wide variety of manufacturers. Setting up a network full of those devices, technicians would likely configure some sort of name into the device (probably using a web-based interface on a laptop), and plug it in into the network. The system would then connect to the device, assign it an IP address automatically—and then it would be presented to the user for patching directly into control channels on the console in the usual way.



A patch screen.

Fixture parameter libraries are already important in the DMX world, but they will be even more important with the sophisticated control functionality possible with ACN. The developers of ACN anticipate that getting a new device to work with your console will be as easy as going on the web and downloading a new file to install onto it, but time will tell about that. In the long run, developers hope that controllers will have enough intelligence to even map out and control a new device it has never seen before. This is certainly possible technically, but, as of today, it's not clear whether or not the market will support this kind of development.

Under the hood

Now that you've seen what ACN is capable of, and what one company is doing with it today, let's look at how it does it. Remember, while it's good to understand protocol details, ACN is designed to make things easier for the user. So, if you get bogged down, please skip ahead to the "ACN Future" section!

Acronym soup

Network engineers are seemingly in love with acronyms, but these abbreviations are a necessary evil with something as complex as ACN: The main acronyms (you might as well learn them now) are CID, DDL, DMP, SDT, RLP, UDP/IP, and IEEE 802.3. Let's take a quick look at each; keep in mind as you work through this section that the protocols relate as follows:

COMPONENT
(console or target device)

Component IDentifier (CID)

Device Description Language (DDL)

Device Management Protocol (DMP)

Session Data Transport (SDT)

User Datagram Protocol (UDP)

Internet Protocol (IP)

Network (e.g. Ethernet)

Component IDentifiers (CID)

Every unit connected to an ACN network needs a unique address, known in ACN as a Component IDentifier (CID). Given past experience with limited address space (i.e. IPv4), the task group wisely used a 128-bit number for this CID, allowing an enormous possible range. Devices will typically have a single CID, but it's also possible for one physical device to use more than one; for example, if you run multiple control applications on the same computer, each application may use its own CID.

Device Description Language (DDL)

The Device Description Language (DDL) defines the interface between the device's functionality and the control abstraction of ACN. DDL is dealt with in systems as a file, written in eXtensible Markup Language (XML), which describes a device's properties in a tree-like structure for use by controllers over the ACN network. Properties in the DDL are associated with "behaviors," which explain additional information about the property, such as what the property actually does, a name for the property to be used by the controller, etc. The DDL structure also gives different controllers the ability to pick and choose which level of control they want to deal with; a simple controller might just get the address of the properties and some other, simple information, while a sophisticated controller might use the entire DDL file.

Device Management Protocol (DMP)

While the Device Description Language (DDL) contains a list of properties associated with a particular device, the Device Management Protocol (DMP) is the part of ACN where the "getting" and "setting" of the properties takes place. Other functions include the handling of parameters that might be continuously (or irregularly) updated, through the use of "events" and "subscriptions." The DMP only understands how to work with the properties; it doesn't understand what they mean.

Session Data Transport (SDT) Protocol

The Session Data Transport (SDT) protocol solves some of the key design challenges of ACN, offering communications far more sophisticated than are available in a simple protocol like DMX. Let's look at two key differences between DMX and ACN, and how SDT addresses these issues. First, a simple protocol like DMX repeats its data over and over, whether levels are changing or not. A continuously updated approach like this wastes bandwidth, but has the advantage of overwriting any data that gets corrupted along the way, or not delivered, and this approach works well for a variety of straightforward applications. However, with a protocol suite as complex as ACN, and the huge amount of data ACN networks are likely to carry, a DMX-like repetitive-data approach could swamp the network with unneeded redundant information. Instead, what is needed in modern systems is "reliable" transmission, meaning that each control packet is guaranteed to reach its destination (or the system will be alerted if a failure takes place).

Secondly, since DMX data is transmitted on a simple, point-to-point bus connection, it is broadcast out to every connected device, and it's the responsibility of

In the long run, the real power of ACN can only be realized if we jettison the archaic concept of DMX "slots" and "universes" and move into more open, sophisticated control.

each receiver of the data to decide whether or not to act on it. If three devices must be controlled by the same data, they all listen to the same data slots on the DMX line; if, at other times, they need to be controlled independently, multiple copies of the data are sent. In ACN, what's preferable is the ability to "multicast" the data, giving multiple devices a way to simultaneously receive the same stream of data from a single sender, maximizing network efficiency.

Standard approaches based on networking protocols like TCP/IP (see below) could solve some of these problems, since the TCP suite offers reliable transmission, and a wide variety of multicasting approaches are available. However, TCP was generally designed to manage the connection of devices in pairs, not in groups, and TCP is optimized to handle streams of data, rather than the short, burst-like messages we are likely to see in ACN. Of course, it's quite possible with TCP to simply set up dozens or hundreds of connections, and still get the all-important reliable delivery, but this could make for an extremely slow network, or one that needed extremely careful management and design, since TCP is designed more with reliability in mind than performance.

For full ACN functionality, new devices are likely going to have to be developed from the ground up with ACN in mind, and manufacturers will only do that if they have pressure from the marketplace.

The task group looked far and wide for a protocol that could offer both reliability and multicasting, but couldn't find anything that would work within the demands of ACN. So it invented from scratch the Session Data Transport (SDT) protocol, which offers reliable multicasting; guarantees to layers above (i.e. DMP) that packets will be delivered to multiple receivers; and, if packets arrive out of order, the correct packet sequence can be reconstructed. SDT does this through the use of "sessions" (hence the name), through which data can flow to and from a device bi-directionally. SDT also integrates some other key, low-level functionality of ACN, such as packing data into Protocol Data Units (PDU) that are outside our scope here.

Root Layer Protocol (RLP)

The Root Layer Protocol (RLP) is the interface between ACN's protocols (SDT, DMP, etc.) and the lower layer network transport protocols, like UDP/IP. RLP has been designed separately to ensure maximum network independence. ACN includes a series of "Profiles for Interoperability" (EPI), and one of the EPIs addresses ACN over UDP: EPI 17. ACN Root Layer Protocol Operation on UDP. If a designer wanted to send ACN over a different network, a new RLP could be created.

UDP/IP

UDP stands for User Datagram Protocol, and IP, of course, is the Internet Protocol, and this pair of protocols (actually part of the TCP/IP suite) will likely be the most widely used ACN transport mechanism. UDP is the Transmission Control Protocol (TCP)'s simpler, less complex cousin. It offers "unreliable," connection-less delivery, but this works in ACN since the reliable, ordered, multicast data delivery is handled by the Session Data Transport (SDT) protocol. UDP has another advantage for control purposes in that it is generallv faster than TCP. since it sends datagrams right way, while TCP needs additional time to manage connections, and might also delay an individual packet to pack it properly onto a stream.

IEEE 802.3 (Ethernet)

IEEE 802.3, commonly known as Ethernet, is likely going to be ACN's main physical network.

More on discovery

ACN discovery uses the Service Location Protocol (SLP), an open protocol developed under the Internet Engineering Task Force (IETF) to locate "services" on a network. Get ready for some more acronyms—SLP has three types of objects which ACN uses: Service Agents (SA), User Agents (UA), and Directory Agents (DA). In ACN, Service Agents are controlled devices (a moving light, a dimmer rack, a video server, a sound matrix) that offer their "service" to the network; User Agents are controllers that are looking for the Service Agents. Optional (but very useful) Directory Agents are servers that store information on known Service Agents and how they can be reached. The services themselves are indicated using the Uniform Resource Identifier (URI), and might take the form of something like service:acn.esta.

When an ACN network of connected devices is discovered, the controller (User Agent) first looks to see if there is a Directory Agent (DA) available. If it finds one, it will unicast all further service requests to the DA. If not, it multicasts a request for the service type acn.esta. Any connected device that can offer an ACN "service" then responds with its unique Component IDentifier (CID). If the controller (UA) recognizes that (unique) CID as one it has dealt with before, it can simply recall that configuration and control information from that previous session, and use that. If not, it issues an attribute request to that device, which responds with a DMP message called csl-esta.dmp This response contains information about the device, such as the DCID, the two human-readable text strings, and other information.

If the controller recognizes the device as a type it understands, it can move on to another device; otherwise, it could request the full DDL (Device Description Language) file to get all the available control parameters, and then present this information to the user for further action. When the controller has worked through all the networked devices in this manner, all this information could be presented to the user, who would then take the discovered devices and patch them to appropriate control channels or other control methods.

More on identifiers and addresses

ACN uses a number of "Universally Unique IDentifiers" (UUIDs), and the large number of different types of ID numbers that are spread throughout the ACN standard can be quite confusing. However, the basic idea is simple: ACN protocols create or use a series of unique identifiers that cannot occur anywhere else on the network, and the protocols can create these UUIDs without any sort of global coordination or consultation with any sort of registration authority. ACN uses several techniques for generating these UUIDs, but the techniques they use to do this are mostly outside our scope here.

As detailed above, each connected ACN device has a Component IDentifier (CID), which is one UUID. Other IDs and addresses include the IP Address, the Component Names, and the Device Class IDentifier.

IP addressing

Each host on an IP network needs a unique IP address. Many show networks, as of this writing, require manual configuration of IP addresses, but, since ACN is designed to be as self-configuring as possible, EPI13: Allocation of Internet Protocol Version 4 Addresses to ACN Hosts mandates automatic configuration of IP addresses through the Dynamic Host Configuration Protocol (DHCP), and several related protocols.

Fixed Component Type Name (FCTN)/User Assignable Component Name (UACN)

Two text strings are included in ACN for human identification of a particular device: The Fixed Component Type Name (FCTN) and the User Assignable Component Name (UACN). The FCTN would typically be set by the manufacturer, and include the manufacturer name and a general model type (i.e., "Zircon Designs Woo Flash"). The UACN is assigned by the user, and would likely include some informational text useful to them (i.e. "Unit #666, Truss # 13").

Device Class IDentifier (DCID)

The confusingly named Device Class IDentifier (DCID) should really be called "Device Name" or something similar, since the DCID is used by a manufacturer to indicate to the ACN network a particular equipment model or type of device. For instance, 20 moving lights of the same brand and type should present 20 of the same DCIDs to a network; of course, the CIDs and other IDs will be unique. Consoles and other controllers can use this information for configuration, associating certain types of control libraries with the DCID; only one DCID is allowed per CID.

Identifier/address summary

Let's add a column to our simplified protocol summary table to show how this lays out (including some other addresses handled by the DDL and other protocols):

PROTOCOL OR UNIT COMPONENT (console or target device)	ADDRESS / ID TYPE
Component IDentifier (CID)	Component IDentifier (CID)
Device Description Language (DDL)	Device Class IDentifier (DCID)
Device Management Protocol (DMP)	Protocol ID (set by ESTA for each protocol)
Session Data Transport (SDT)	Session Number
User Datagram Protocol (UDP)	Port Number
Internet Protocol (IP)	IP Address
Network (e.g. Ethernet)	MAC Address

Of course, in UDP/IP over Ethernet, the network itself deals only with IP addresses and MAC addresses; ACN's CIDs, DCIDs, etc. are all just part of the data payload carried over the MAC frame.

Further developments

As of this writing, ACN has been released, and, while it hasn't yet been offered yet in many products, there are already several initiatives underway to extend and refine ACN, and promote its use. Several protocol extensions are under development by ESTA, and another key effort is "OpenACN," led by Dr. Philip Nye of Engineering Arts, who was an important member of the ACN task group from the beginning. Dr. Nye is looking for funding to develop a complete open-source implementation of ACN. More details can be found on: http://www.openacn.org/

An ACN future

I'm really excited about the potential of ACN, because it could open up a whole new world of control possibilities. For example, let's say I want to build a moving-head video projection fixture that is "aware" of its position in the room, and have it be able to aim itself at any spot onstage at any elevation in XYZ coordinates (instead of pan and tilt angles). In my fixture's DDL, I could establish basic and traditional "legacy" functionally, basing some common features (like brightness) on established practices; I might even have a "dumb" mode, where my fixture responds to pan-and-tilt angles like a traditional moving light. But, with ACN, I could create an additional control mode where when my fixture works allows the fixture to be aimed at a target on the X/Y/Z coordinate system, and be controlled in this sophisticated way by any console that could figure out my device properties.

Terry: "ETC has committed time and resources to the ACN effort. which is now bearing fruit due to the efforts of many companies over many years. It's not altruistic on our part—we believe that ACN will be the commercial future of our industry."

And that's just step one. What if I want to have a scenic automation controller report a scenic unit's position and have my new fixture track it, projecting its image on a moving platform or screen as the scenic unit moves onstage? In the ACN future, I could plop a show controller into my system, have it interpret the positional information from the scenery controller (perhaps broadcast in simple UDP ASCII format) and output XYZ target scenic position for my moving video head fixture in ACN. My video fixture can now, using all open industry standard control methods, accept scenic position data from the show controller/scenic system and track that point in 3-D space, while accepting image and intensity information from the "lighting" console. Think about where all this can lead, and you can start to see why I think ACN could change everything. With it, the future is really limited only by your imagination.

The ACN future needs your help

Using ACN as an open, network-based DMX replacement is certainly an important transitional step, but, in the long run, the real power of ACN can only be realized if we jettison the archaic concept of DMX "slots" and "universes" and move into more open, sophisticated control. This may take some time, because, while ACN makes life very easy for users, it is complex to implement for manufacturers, and needs a fair amount of processing horsepower in every device. While computer power gets ever cheaper, it will still take a major investment by a wide range of manufacturers to get ACN off the ground. Before my trip to Wisconsin, I hadn't really thought through how significant the changes to a console would have to be to fully embrace ACN, and this may be why we didn't see a flood of ACN products on release of the finished standard. Think about it: In a DMX-only world, a console simply has to store and manipulate an array of relatively simple eight-bit numbers. Now, with ACN, the console has to have some knowledge of the meaning of each parameter of each controlled device, and each parameter, of course, could be of a completely different type, format, and structure. That means that, for full ACN functionality, new devices are likely going to have to be developed from the ground up with ACN in mind, and manufacturers will only do that if they have pressure from the marketplace.

For several years after the MIDI Show Control (MSC) standard was developed, I used to end every show control workshop at LDI by urging the attendees to go out and beat up every manufacturer on the trade show floor, telling them to implement MSC in their consoles instead of developing their own, non-compatible MIDI implementations. Eventually, the market spoke (certainly well beyond my own impact) and MSC is now standard on most lighting consoles for interconnection and remote triggering. Right now, we are at a similar and important juncture with ACN; we have to get past a tipping point where enough interoperable ACN products are spec'd and installed for it all to gain momentum. At that point, everyone would be used to using networks on their shows (that's coming with or without ACN), would figure out solutions to the key implementation problems, and would then be in a position to truly exploit the full potential of something as sophisticated as ACN. I'm calling on you to write/call/fax/e-mail your favorite manufacturer's rep, or beat them up on the floor of PLASA, InfoComm, or LDI, and tell them why you want ACN now. The future depends on you.

John Huntington is a professor of Entertainment Technology at NYC College of Technology, and is author of *Control Systems For Live Entertainment*, the third edition of which was just released (parts of this article were excerpted from the new edition with permission). Mr. Huntington can be contacted and more information on the book is available through **www.controlgeek.net**