Why WiFi?

February 3, 2018 Version

Lee Hollaar is an Emeritus Professor of Computer Science at the University of Utah and technical advisor to Pioneer Theatre Company. He has been programming computers since 1964 and has been designing and building computer networks since 1978. He developed and taught one of the first courses in computer networking in 1982, teaching it fifteen times before he retired and revising it as technology and protocols changed.

There have been some questions about why I am using standard WiFi (IEEE 802.11) for communications to the wireless DMX devices I've designed for Pioneer Theatre Company. After all, one company selling similar products warns "Could you imagine waiting several seconds (or longer) to execute an important cue on a show? ... It must work, it must work with imperceptible latency, and it must work every time."

Those fears bring back my memories of the early network battles, where companies promoting technologies such as token rings (IEEE 802.5) raised the same point about the new-fangled Ethernet. It wasn't suitable for real-time applications, because the time it took to transfer a message was indeterminate. Time has shown that, for most applications, that's just a hypothetical and not real problem.

The reason that I use WiFi is simple. It provides the service that I needed at the lowest cost. Pioneer Theatre already had a private WiFi network installed that covered the stage area, used for remote control of its lighting console. It was already using sACN on an Ethernet from its console to control a VISCA camera interface, allowing a Sony camera to be controlled as if it were a moving light, and that sACN was also on the WiFi network. No new hardware or connections to the console were necessary to have DMX over WiFi.

Because of its ubiquity, WiFi routers and access points are inexpensive and can be purchased from places like Best Buy, Staples or Office Depot, Amazon, even Home Depot or Costco or Walmart. That's handy if you have a problem and need a new router quickly. And if there are problems with signal coverage, which is unlikely given the small area and openness of a stage, you can get range extenders to fill in any dead areas.

Of course, a little common sense can reduce the chance of problems using WiFi. If possible, configure the WiFi router or access point to operate on its own channel and not share a channel with a busy network. Use the lighting network just for lighting by blocking connection to the Internet in the router, maybe allowing only special things that need Internet access.

But this isn't as much a problem as one would guess. While Pioneer's lighting WiFi network is on a different channel than the one used by the sound people, it shares its channels with campus WiFi in the building. But because there isn't heavy WiFi use during a show and the lighting WiFi has as stronger signal on stage, there hasn't been any problems.

The biggest reason I used WiFi is that the microcontroller I'm using, the ESP8266, comes with WiFi hardware (it was originally designed to be the WiFi module in a PC, but people found that it could be used as a general-purpose microprocessor), so nothing else is needed. The Arduino software development for the 8266 includes all the software you need to easily use that hardware, including support for the IP multicasting used by sACN. Just give the WiFi library the SSID and password for the lighting WiFi network, say a few magic words, and I'm connected.

The first microcontroller that I used cost about $20 from Adafruit. We now buy WeMos D1 mini's for about $5 each! The microcontroller and a single inexpensive chip gives me the hardware for an sACN-to-DMX gateway. WS2801 and WS2812 addressable LED strings just require connection to the microcontroller, and analog LED strips only require a FET per color.

Finally, there are a number of free or inexpensive tools available to troubleshoot problems with sACN over WiFi. I use WiFi Analyzer on my Android phone to make sure there is a good signal where it's needed, although there hasn't been a problem on Pioneer's stage because of where the WiFi access points are located. sACNView running on a portable with a WiFi connection lets me see the actual DMX being transmitted and whether there are two or more sources simultaneously trying to send to the same universe. (That happens when the console and its backup aren't properly synchronizing or there are two consoles connected to the lighting network.)

That, plus the diagnostic messages from my programs as they start up, gives me just about everything I need to troubleshoot a DMX-over-WiFi connectivity problem.

As for a networking topology that must work every time, at the hardware level there isn't such a beast. Instead, protocols are developed that provide the desired service in actual operating conditions.

And sACN for DMX does just that for our applications. Because the DMX information is repeated as often as 40 times a second, if a packet containing the slot values for a universe doesn't make it to the receiver for any reason, it will be repeated in about 25 milliseconds, far faster than can be seen for the types of LED cues we are handling.

The unreliable IP multicast protocol using by sACN is perhaps the best alternative for DMX over WiFi/Ethernet. Using a different multicast address for each universe means that my program doesn't even see DMX information for universes other than the one I'm interested in. Repeating the DMX information solves the reliability problem better than an acknowledgment-based protocol that could become constipated unless every active receiver sent back its acknowledgment promptly. And packets only have to be sent once to be received by everybody interested in the universe.

If the signal can't be received because of a hard wireless problem, it's easy to determine because there is an LED on the microcontroller board that I turn on only when I'm receiving DMX information. And that same wireless problem could likely affect the higher-priced wireless DMX alternative, since it operated in the same radio band as WiFi.

There is only one instance where I've seen a problem with using WiFi to transmit information to the 8266 microcontroller, and that wasn't with sACN/DMX but with using Open Sound Control (OSC) to trigger an action. Unlike DMX, OSC only transmits its control packet once and doesn't get an acknowledgment of its reception (unless you are using TCP instead of UDP for the OSC message, which senders like QLab can't do), so if a message doesn't get through for any reason, it's lost.

But my solution to that is simple. I've written an intermediate handler than can run on a PC or a Mac that receives the messages over a reliable software link on the computer running the OSC sender, and then uses a simple light-weight protocol to keep sending the message until it get through.

But, as I said, that's not needed for DMX over sACN over WiFi, because the lighting console repeatedly sends the universe slots so if you miss one packet, you'll get new information almost immediately.

Also it's nice to be able to use the same WiFi network not just for DMX, but also other protocols such as OSC.