What does RDM mean for the rest of us?

BY **MILTON DAVIS**

How RDM can be put to good use right now.

WHILE EXHIBITING AT VARIOUS TRADE SHOWS, I have talked with hundreds of professionals in the entertainment industry. Although the *ANSI E1.20—2006 Entertainment Technology-RDM- Remote Device Management over USITT DMX512 Networks* standard was released in 2006, even today I often get a "deer in the headlights" look when I bring up this protocol with professional users. There have been numerous demonstrations of the new technology and panel discussions covering its technical details. The RDM forums at www.rdmprotocol.org/forums/ are also a useful source of information regarding the standard and its implementation. That's great for the propeller-heads amongst us (okay, I'm in that category, too), but what about everyone else? In real and concrete terms, what does RDM do and why should you care?

Let's look at this question by first examining how our needs have changed over time. DMX512 has been in use since the early nineties and it still has a lot going for it. The protocol is in wide use and new equipment is constantly being introduced with DMX512 compatibility. The original purpose for DMX512 was to get dimmer level information from a console to dimmers. With traditional DMX the information only goes one direction: out from the console to the dimmers. This was a great improvement over the prior one-wire-per-dimmer analog controls available up to that point. The concept of controlling the intensity of 512 dimmers over a single small cable and over relatively long distances was revolutionary at the time.

The key thing here is that we were only controlling dimmers at that time and the number of dimmers was relatively small. Troubleshooting was simple and the devices we were controlling were also simple. If a fixture didn't turn on, you grabbed your test lamp and checked the breaker and dimmer output as you walked by the racks. If it was hot there, you either had a bad cable or a bad lamp. There was no need for the dimmer to report back to tell you anything about itself. All information was "pushed" from the user in a single direction: out from the console to the dimmers. The user keyed in a command on a console. The console sent new DMX levels out to the dimmer. The dimmer changed the voltage to the lamp. The user saw the result of keying in the command.

The concept of controlling the intensity of 512 dimmers over a single small cable and over relatively long distances was revolutionary at the time.

The same scenario takes place today, but the quantity of devices being controlled has increased and types of devices has expanded far beyond simple dimmers with incandescent lamps. Today, we have moving lights, foggers, LED-based fixtures, and a wide array of special effects devices. Troubleshooting tasks become far more complex in these situations. There are opportunities for numerous errors in simply setting the DMX address and operating modes on a moving light. The use of DIP switches makes for a non-intuitive user interface. The values on the switches may be offset by 1 address on some products. If multiple universes of DMX are involved, there is also the chance of incorrectly figuring the number of channels to offset into the second and subsequent DMX data streams.

As noted earlier, traditional DMX512 communications move in only one direction. The core difference between DMX and RDM is the fact that RDM allows for two way (bi-directional) communication while maintaining compatibility with standard DMX512 signals. Between standard DMX-style data packets, a console can send commands or ask a device for information and the device can respond back to the console regarding the command or request. This one difference creates the opportunity for a rich operational experience.

Imagine making a phone call that can only carry your voice out. Anyone with a cell phone has certainly had this experience more than once. The person on the other end of the line can hear you, but cannot respond. You could give the listener information and make requests for action, but you would have no idea whether they received the information and acted on it. If the phone line could allow the listener to just say "OK" or make a clicking sound you would at least have some assurance that they are present. If you allow the listener to reply with complete sentences, you could get a lot of information. For example, you might say "What time is it?" and the listener could reply with the information. From this simple request, we can determine a number of things after the response:

- If there is no response, we can determine that the listener is not there.
- If there is any response, we know that the listener is present.
- If the response is something like 2:30 p.m., we know that the listener understands the question.
- If we know the time to be 2:30 p.m., we can infer that the listener is in our time zone and that their watch is set correctly.

Let's apply this same phone call metaphor to a lighting console and a moving light. Using current DMX512 technology, the lighting console sends DMX data telling the moving light to do things. If things work as the operator expects, all is good. There is no ability for the light to tell the console that it's OK or that it received its commands at all. Since things are working, does it really matter? But let's say that the operator uses the console to tell the light to turn on and move to a particular position and nothing happens. At this point, the operator must question a number of things including:

- Was the proper command entered?
- Is the fixture patched correctly?
- Is the fixture getting power?
- Is the DMX line connected?
- Is the fixture's address set correctly?
- Is the fixture broken?
- If the fixture is broken, what is wrong with it?
- Is some gear between the console and the fixture broken?

From here, the operator-turned-troubleshooter essentially only knows that something is wrong. In a large rig, this can be the beginning of a huge waste of time to eventually discover a minor fault.

If we have the same situation with an RDM-enabled system, let's see what might happen. The failure described is unlikely to take place at all with an RDM-enabled fixture and console because the operator would already know about the issue preventing proper operation. When the fixture was first hung, powered, and provided with data it would be recognized by the console. This immediately tells us that the fixture has power and that its data line is connected. The console can set the fixture's DMX address remotely so that error can be eliminated. If the fixture monitors its functions, it can provide information about its lamp, motors, et cetera. So, if the failure mode described were to take place, the fixture might, for example, already be informing the operator that its locking mechanisms are engaged. The operator knows that there is a problem and what needs to be done to quickly correct it.

Another use for RDM is its ability to remotely configure a device. The most common and useful example is setting a DMX address without the need to visit the device in person. This feature alone makes RDM worth the effort. How many times have you

discovered that your carefully pre-addressed moving lights have been hung in the wrong places? With RDM, you can leave the rig in place at full trim height and simply set the DMX addresses from the ground.

With RDM, you can leave the rig in place at full trim height and simply set the DMX addresses from the ground.

These examples are great on paper, but consider the products available today. A visit to ESTA's manufacturer ID listing www.esta.org/tsp/working_groups/CP/mfctrIDs.php shows that there is a huge number of companies who are designing and building products which will support RDM. Most of those shown have products on the market today. During ESTA's Plugfest events, we have seen products in every stage from early prototype to full production. While the Plugfests have been oriented toward engineering issues, they have also demonstrated that there are real products available today with extensive RDM implementations.

A note of caution needs to be brought up here. Deploying the RDM protocol in a fixed or touring rig is quite easy. Your existing cables (even the technically non-compliant ones with 3 pin connectors) and non-RDM end devices (fixtures, dimmers, etc.) will work normally while connected directly to RDM enabled devices. However, this co-existence will only work well if the non-RDM enabled devices are truly compliant with the DMX512 protocol. There are devices (even new ones) which will work with DMX512, but will misbehave if exposed to RDM data. These devices are not DMX512 compliant.

Fortunately, a number of manufacturers have provided devices with "safe" or "stripped" DMX outputs. These products remove the RDM information and send only generic DMX512 packets that will work with non-compliant gear. This may force you to run two cables to a location; one with RDM capability and the other without. As time passes, the need for this type of accommodation should diminish.

Another thing to keep in mind when evaluating RDM-enabled gear is to determine how fully the device implements the RDM protocol. There are only a few RDM commands which are required by the standard. Support for most commands is optional. This allows the manufacturer to determine the appropriate commands for their device based on the user's needs and on the resources available. For example, supporting a motor current report on a fixed position lighting fixture would make no sense. There are many RDM commands which might be nice to support in a product, but the manufacturer has chosen to not put the appropriate sensors in the product. Devices like temperature sensors, current sensors, and clocks are common examples. It might be nice to know the temperature in a device, but the RDM standard does not require that a device can support temperature reporting. It is up to the user to request a list of supported RDM commands from the manufacturer and to determine if the list is adequate for the application being considered.

Distribution gear (e.g., DMX splitters) will need to be re-evaluated when RDM is deployed. Standard DMX splitters will not work with RDM. Since traditional DMX only sends data in one direction, splitters are designed for that application. New bi-directional splitters will be needed to allow the console and end devices to see each other. The change from standard DMX splitters to RDM enabled devices is typically only a matter of replacing one piece of gear with another in a properly installed existing system.

DMX512 is not dead by any means. By adding the RDM protocol to our tool kit, we gain the ability to extend the life of our existing gear and to add functionality as new gear is purchased. DMX and RDM mix well together over existing cabling thus keeping the cost of retrofitting low. The benefits of RDM-enabled equipment available today allow us to enhance control, configuration, and troubleshooting efforts. As you do your next load in, think about how much easier it might be if the fixtures and dimmers could tell you about their status and if you could do something as simple as setting the DMX address on a device without leaving the chair at the console. ■

Milton Davis, a 30 year veteran of the entertainment lighting industry, is employed by Doug Fleenor Design, Inc. where he designs new products and provides telephone support in a bi-directional mode of communication. He is a member of ESTA's Control Protocols Working Group contributing to the DMX512-A and RDM protocol standards. Milton has worked as a lighting designer, master electrician, field service manager, quality control manager, product developer, and software engineer. He is an ETCP Certified Entertainment Electrician and Recognized Trainer.