In-depth technology for radio engineers

radioworld.com | August 11 2021 | $5.00

RADIOWORLD engineering extra

A DIY mail transfer agent

Old-school origination clients need a little help.

A nod to Dick Sequerra

Robert Richer remembers.

Virtualize your audio processing

A checklist for high-density virtual deployment.

Network monitoring on a budget

Paessler PRTG lets you monitor 100 devices for free.

FUTURE

RADIOWORLD
engineering extra

Vol. 45 No. 20 | August 11 2021
www.radioworld.com

FOLLOW US
www.twitter.com/radioworld_news
www.facebook.com/RadioWorldMagazine

CONTENT
Managing Director, Content & Editor in Chief Paul J. McLane, paul.mclane@futurenet.com, 845-414-6105
Senior Content Producer — Technology Brett Moss, brett.moss@futurenet.com
Technical Advisors Thomas R. McGinley, Doug Irwin
Technical Editor, RW Engineering Extra W.C. "Cris" Alexander
Contributors: Susan Ashworth, John Bisset, James Careless, Ken Deutsch, Mark Durenberger, Charles Fitch, Travis Gilmour, Donna Halper, Craig Johnston, Alan Jurison, Paul Kaminski, John Kean, Peter King, Larry Langford, Mark Lapidus, Jim Peck, Mark Persons, Stephen M. Poole, James O'Neal, Rich Rarey, Jeremy Ruck, John Schneider, Dan Slentz, Randy Stine, Tom Vernon, Jennifer Waits, Chris Wygal
Production Manager Nicole Schilling
Managing Design Director Nicole Cobban
Senior Design Directors Lisa McIntosh and Will Shum

ADVERTISING SALES
Senior Business Director & Publisher, Radio World
John Casey, john.casey@futurenet.com, 845-678-3839
Publisher, Radio World International
Raffaella Calabrese, raffaella.calabrese@futurenet.com, +39-320-891-1938

SUBSCRIBER CUSTOMER SERVICE
To subscribe, change your address, or check on your current account status, go to www.radioworld.com and click on Subscribe, email futureplc@computerfulfillment.com, call 888-266-5828, or write P.O. Box 1051, Lowell, MA 01853.
Licensing/Reprints/Permissions
Radio World is available for licensing. Contact the Licensing team to discuss partnership opportunities.
Head of Print Licensing Rachel Shaw licensing@futurenet.com

MANAGEMENT
Senior Vice President, B2B Rick Stamberger
Vice President, Sales & Publishing, B2B Aaron Kern
Vice President, B2B Tech Group Carmel King
Vice President, Sales, B2B Tech Group Adam Goldstein
Head of Production US & UK Mark Constance
Head of Design Rodney Dive

FUTURE US, INC.
11 West 42nd Street, 15th Floor, New York, NY 10036

Connectiv
ipso. Regulated

Radio World (ISSN: 0274-8541) is published bi-weekly with additional issues in February, April, June, August, October and December by Future US, Inc., 11 West 42nd Street, 15th Floor, New York, NY 10036-8002. Phone: (703) 852-4600, Fax: (703) 852-4583. Periodicals postage rates are paid at New York, NY and additional mailing offices. POSTMASTER: Send address changes to Radio World, PO Box 1051, Lowell, MA 01853.

FUTURE
Connectors.
Creators.
Experience
Makers.

Future plc is a public company quoted on the London Stock Exchange (symbol: FUTR)
www.futureplc.com

Chief executive Zillah Byng-Thorne
Non-executive chairman Richard Huntingford
Chief financial officer Rachel Addison

Tel +44 (0)1225 442 244

Broadcasting — a virtual world?

More and more, radio's infrastructure is moving to the cloud

Cris Alexander
CPBE, AMD, DRB
Director of engineering, Crawford Broadcasting

Comment
on this or any story. Email *rweetech@gmail.com.*

Webster identifies the word as an adjective and gives the definition as *being so in effect, although not in actual fact or name.*

We hear the word virtual a lot in today's culture. Virtual reality has moved from hard-core gaming to the more mainstream with headsets and adaptors that can turn our smart phones into VR engines.

We ask our virtual assistants for directions or to change the channel on the TV. You can get a virtual pet or send someone a virtual gift card.

Since the start of the pandemic, virtual meetings have become everyday events. I have participated in countless virtual training sessions, virtual conferences and even a virtual NAB conference earlier this year.

So it would seem that we now live in a virtual world, and perhaps some do. But not me. I'm more of a brick-and-mortar kind of guy — I like real things that I can see and touch, things that exist in the real world and not just in lines of code and ones and zeros.

That's not to say, however, that I don't find many of those virtual things useful. I very much do, and I appreciate them.

For example, in addition to some of the things I listed above, I use a very capable flight simulator, which you might categorize as a virtual airplane, to keep my flying skills sharp between real-world flights. Now that has value.

So you might say that I'm warming to the idea of a virtual world.

In our biz

In recent years, and especially since the pandemic came along, broadcast equipment manufacturers have been coming out with all kinds of virtual devices.

THIS ISSUE

Top
Virtual consoles like this Axia IQs are now a real-world thing.

Above
Apps like Wheatstone's ScreenBuilder can pull information from many locations, inside and outside a facility, including from virtual consoles, to produce a "one-stop" display of important data.

Some of these things exist only in cyberspace, and some have real-world components.

One thing these virtual devices have in common is that they all have real-world application. We're not talking about video games here. We're talking about top-tier products that perform critical functions in the broadcast infrastructure.

In that sense, they depart from Webster's definition in that they do exist in actual fact ... just not in the traditional physical form.

For some time now, AoIP has been making inroads into broadcast infrastructure. I have been involved with this personally and can tell you that all those virtual connections beat the heck out of punch blocks and switchers.

Our cable ladders are almost empty, now with just small bundles of Cat-6 wires rather than huge bundles of audio and AES cables. Most any engineer who has been blessed with AoIP infrastructure will likely sing its praises, especially the ease with which "wiring" changes can be made, often from the comfort of his or her home or office.

A number of manufacturers have come out with virtual mixers, virtual boards, virtual control surfaces or whatever name you want to give them.

I suspect that the impetus for most, which likely predated the pandemic, was for remote operation or facility consolidation. But when COVID came along, a whole new need for that kind of thing presented itself. People were working from home, far from the faders, switches and knobs that they would normally manipulate with skilled hands.

Dreaming big

In recent days, I have read of PPM watermark insertion that will soon occur "in the cloud." That, apparently, is one of those hybrid applications I mentioned where some real-world hardware is involved.

It is intriguing, and I'm all for eliminating a piece of equipment in the air chain along with its connectors and potential points of failure.

And of course we have seen audio playout or digital media systems moving into the cloud in recent years. That's a little scary for me; I like everything to be under our own roof where I can lay hands on the infrastructure and be responsible for its safety and security. But undeniably, that's the way it's all going.

I think the days of the on-premises file server based digital media system are numbered. Maybe, just maybe, that's a good thing. I'm adjusting to the idea.

Our phone vendor tells me that equipment manufacturers have warned that their on-premise system assembly lines will, in the next couple of years, be no more, and all our telco service will be cloud-based. Letters I get regularly from carriers confirm this —"traditional" telco services are going virtual, and we had better be prepared to make the switch.

Now, we're seeing virtual audio processors. In this issue, you will read about Omnia's enterprise audio processor, which exists in cyberspace. This is another hybrid virtualization, as real-world I/O has to exist, at least for the time being. But how long will it be before our exciters could simply connect to the network with no real world I/O at all besides the obligatory RJ-45 connector? That's an exciting prospect.

Last issue, I left you with the way-out-there idea of high-power RF over IP. I'm pretty sure that's not going to happen, *ever*.

But what about a virtualized transmitter? That would be a hybrid application for sure, and if you think about it, many of the components are already in place in some production transmitters: single-board computer, web interface, SNMP control and monitoring.

You would need real-world power amplifiers, combiners, low-pass filters and control/monitoring hardware for sure, but could we move exciter, control and diagnostic functions completely outside the transmitter cabinet and into the virtual world?

Is that something that could happen someday as we continue to virtualize broadcast infrastructure? It's certainly something to consider.

Personally, I'm hoping for a virtual phasor and virtual ATUs. Now that would be cool. But seriously, folks, it's not out of the realm of possibility. Way back in 2001 and again in 2006, Mario Heib presented papers at the NAB Broadcast Engineering Conference proposing a digital phasor with variable phase/variable output power amplifiers at the tower bases. I can see virtualizing every part of such a system other than the actual power amplifiers and matching networks.

We're headed for a point where cutting-edge facilities will have mostly empty equipment racks. I think that is something to get excited about.

Writer

Todd Dixon

Engineer

Crawford-Birmingham

Share your brain

Got an idea for a great technical story with tips or best practices? Email Cris Alexander at *rweetech@gmail.com*.

Network monitoring on a budget

Paessler PRTG lets me monitor 100 devices for free

One of the great things about having equipment with network capability is the ease with which you can get to that equipment and configure it, and the dramatic increase in the amount of diagnostic and performance data you can receive from it.

Of course, with so much data at your fingertips, the hard part for broadcast engineers is trying to find a way to filter that information into something that is concise and meaningful and that allows them see their operation at a glance.

The best way to do that is by using the Simple Network Management Protocol, or SNMP.

For the past year or so, I've been trying to wrap my gray matter around SNMP. "Real" IT folks have been using it for decades to keep track of network speeds, operating systems, hard drive capacities and a host of other metrics around their data centers.

It may have just been me, but I wasn't finding real how-to-do-it kind of articles. It seemed most writers just wanted to keep the secret to themselves and dazzle you with sharp terminology.

After several failed attempts, I found a solution that made sense to me. It's called Paessler PRTG network monitoring, and it allows me to monitor all of the devices I have set up (up to 100 for free).

It also gives me a mobile device app that provides notifications when something is not behaving correctly on our network.

In preparation

Before we set up any solution, let's talk about how SNMP works.

Device manufacturers that want their equipment to talk on a network develop a Management Information Base (MIB) file. The file is a reference database of structured definitions and Object Identifiers (OIDs) that numerically represent any kind of data possibility that you may want to pull from the equipment.

SNMP allows polling of equipment, but also has traps designed into the protocol as well so that certain datum can initiate messages back to the monitoring software (i.e. a threshold has been surpassed).

I won't address traps here, but it is important to know that they exist and can be used as part of your solution as well.

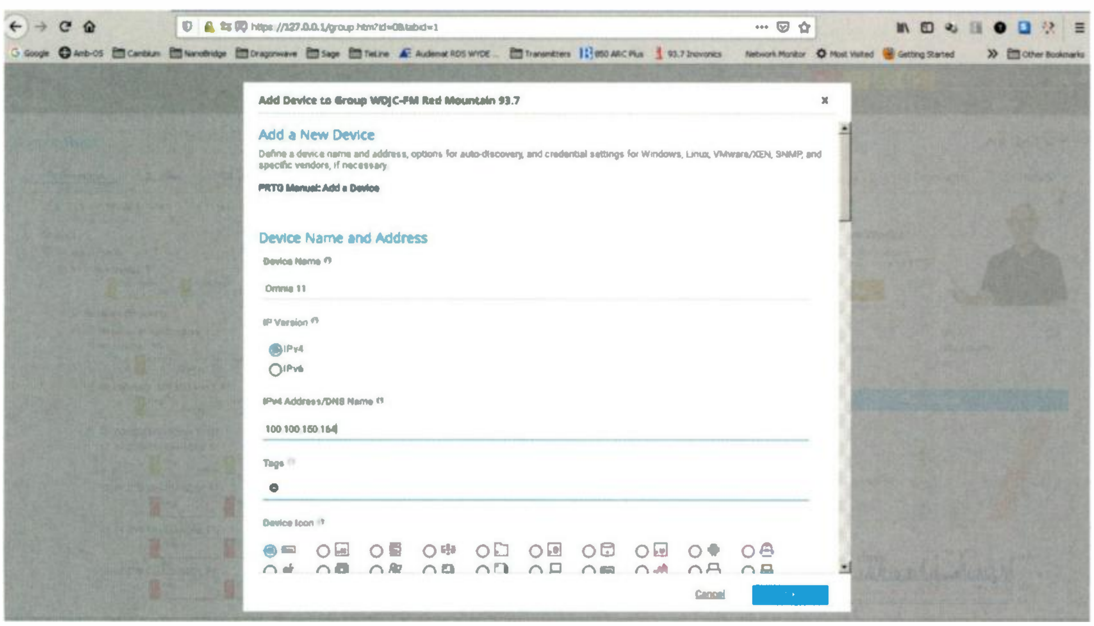
So not only will we need the monitoring software, but we need some software to help us sift through all the OIDs available and only use those that we really want information about.

In preparation, we'll need to download the necessary software. Both the Paessler software (*www.paessler.com/prtg*) for network monitoring and the Ireasoning MIB Browser (*www.ireasoning.com/downloadmibbrowserfree.php*) for parsing MIB files are freely available.

The other downloads that will be necessary at some point are the MIB files for your equipment. You can generally download the files from the manufacturer, but pay attention to the hardware models, as an MIB may vary from one device to another.

The Paessler software requires a Windows machine and sets up the network monitoring on that machine. It almost goes without saying that the machine should be a part of the network that you are trying to monitor.

Below
Fig. 1: Adding a device in PRTG.



Remote access with web enabled products

WEB ENABLED

POWERFUL & AGILE

SOUND PROCESSING

All-digital audio processing for AM / FM airchain, general overload protection, and production.

NEW 719N DAVID IV

FM/HD Radio™ Broadcast Processor

WEB ENABLED

NOW WITH REMOTE WEB INTERFACE

- Streamlined 4th generation DSP-based audio processor.
- 5-bands of dynamic range compression & "Graphic EQ".
- 25 Factory presets and 20 customizable presets.
- Multilingual front panel in English, Spanish, & Portugues.

JUSTIN 808

FM/HD Radio™ Alignment Processor

WEB ENABLED

- Unique single box solution to maintain time alignment between analog FM & HD1, with precise alignment to within 23 microseconds (±1 sample). 100% automatic.
- Web interface for remote control, metering, etc. SNMP support.
- Extensive data logging with graphic display. SMS/email alarm notifications.

NOVIA 272

FM Audio Processor | Stereo Gen | Dynamic RDS

WEB ENABLED

- A member of the compact 1/2-rack sized, all-digital DSP-based 3 band NOVIA family of processors.
- Models are available for FM, AM, & Dual Mode Stereo.
- 5-bands of dynamic range compression & "Graphic EQ".
- Analog, AES-digital, Streaming IN/OUT. Simple set-up with 10 factory presets and 10 customizable presets.

INOMINI 223

Multimode Audio Processor

- An audio processing powerhouse in a small package.
- Versatile DSP-based design is user programmable to serve multiple broadcast applications: NRSC, AM, FM, SCA, & TIS.
- Monoaural.

www.inovonicsbroadcast.com | sales@inovonicsbroadcast.com | 831-458-0552

WHY INOVONICS? Quality Solutions. Competitive Prices. Quick to install. Easy to Program. Three-year Factory Warranty. Quality after sales service.

VIEW OUR PROCESSORS ONLINE

INOVONICS BROADCAST

Note that network monitoring isn't a resource intense activity on simpler networks, so an older machine will do fine for this purpose.

Setting up

PRTG configures a web server as well so that you have access to the information gathered locally on the network. But to harness its real power, you'll want to forward a network port (any port, like 23333 as an example) on your router or firewall to that local machine's web server (secure https-port 443) so that you can have access to that data from the outside world via another machine or the mobile app.

Installing PRTG is pretty straightforward. After choosing a strong password, you can immediately begin putting in device information.

Even without any knowledge of MIBs and OIDs, you can begin to configure PRTG to send network pings to devices so you know that they are online on your network.

Simply choose to add a device, fill in the appropriate name and IP information, and choose an icon that makes sense to you. See Fig. 1 on page 6. Once that is done, you'll see it show up in your "Network Infrastructure" list under the name you gave it.

Next, choose to add a "sensor," like a network ping. It might seem daunting, but once you add a device you can also choose "auto discovery" for the device.

Of course, it will find a number of things that you may not want to monitor, and you may quickly push the 100-sensor limit of the free software. Any sensors in the auto discovery that you don't find useful can be removed with a right click of the mouse and choosing to delete that monitor information.

I have found it useful in PRTG to go ahead and create groups by location and then to add devices inside of that group. As a for-instance, we have four tower sites on our network, so I have identified each in a group and then added the related transmitter and devices such as STL IP radios and codecs inside of that group. In this way, my notifications tell me immediately which site is having an issue.

"Even without any knowledge of MIBs and OIDs, you can begin to configure PRTG to send network pings to devices so you know that they are online on your network."

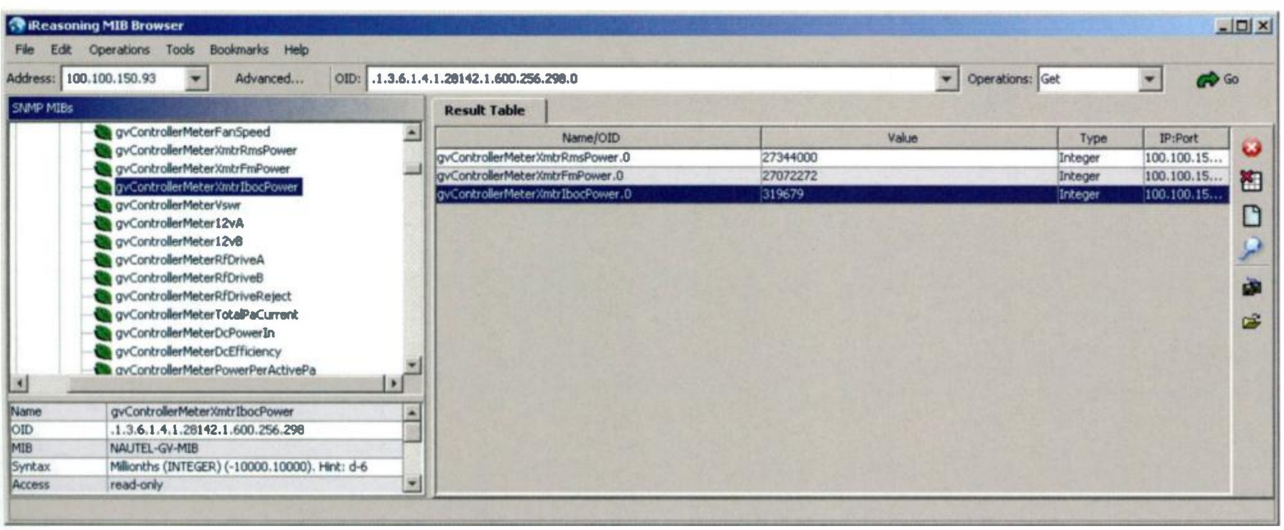
The auto discovery isn't going to find all of the information that a broadcast engineer really wants to know, such as a transmitter's forward and reflected power indications, STL path RSL voltage levels or any of a host of other polled information. Before we really get started adding those parameters, you'll need to jot down some SNMP information that is available for each monitored device.

SNMP requires network ports 161 and 162 to be open (they generally are), a "Read Community" string and a "Write Community" string (usually "public" or "community"), and you'll need to know the SNMP version the device is set up to use (normally V1, V2c or V3, each increasing in security level). We'll need all of this information in the Ireasoning MIB browser.

Once the MIB browser is installed, go into the "File" menu and import the MIB files that you've downloaded from each vendor. See Fig. 2. You'll see a directory structure populate in the window that contains the data options available for the equipment that you have.

With that done, we'll enter the IP address of the device we want to get data from, and then click the "advanced" button to make sure our read and write community strings are accurate and the SNMP version is correct.

Below
Fig. 2: Selecting the parameters to monitor.



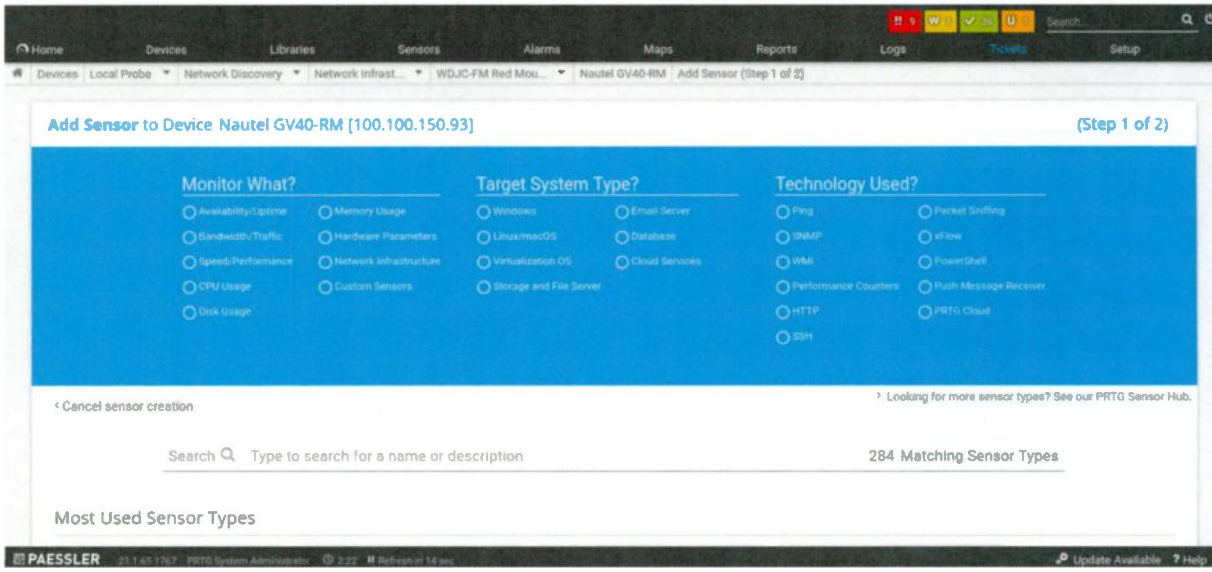
Next, we'll browse the MIB folder structure to find the OID (data) that we need.

Now that we've found the OID we're looking for, we simply click the "Go" button on the right. The browser will query the device for the data and display it in the window on the right. The most important part of that whole operation is getting the

OID number that you need to use in PRTG to poll that data automatically.

If you are feeling really brave and want to see just how much information is really available to you via SNMP from that device, click the down arrow in front of the "Go" button and choose "walk" and then click "Go." Then go refresh your coffee.

Obviously you won't need every piece of data that you'll see from that operation, but isn't it nice to know that you could have it if you did?



Above
Fig. 3: Choosing the sensor type.

In Fig. 2 on page 8, you can see that I've imported our Nautel GV40 MIB file and have found the OID numbers for the transmitter's RMS forward power, its FM power and the IBOC power.

Once we have the actual OID number for each piece of data, we can hop back into PRTG and add a sensor to the equipment we have already added. When we choose to "Add Sensor," we're taken to a page where we're asked what type of sensor. For our purposes, our answer will almost always be to click SNMP and then choose the SNMP Custom. See Fig. 3 above.

From there, similar to adding a device, we can name the sensor and put the OID number, the value, and unit. As you can see in the MIB browser image in Fig. 2, the value you get may need to multiplied or divided the value by a multiple of 10 for it to be correct, which you're given the option to do in this sensor screen.

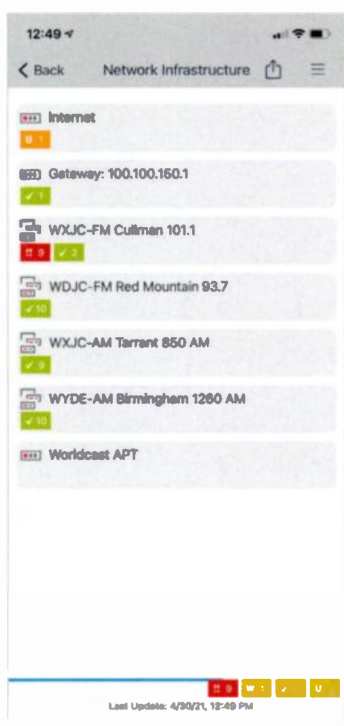
Once the sensor is created, it will take a minute or two to get its value, and you'll be able to see if everything looks satisfactory. If you are anything like me, you may need to go back and edit several times to get the value to display the way you want it to.

Going forward, it is simply a matter of repeating this procedure until you have all the equipment you want to monitor in place and grouped together the way you want them.

Is it worth the effort?

Obviously, you can devote a whole lot of time to this, and you might begin to question if it is worth the effort.

There are two powerful aspects to installing something like this that you might consider. The first is that if you choose to port forward through your firewall to the web page that PRTG establishes, you can have web access to the page from anywhere and can evaluate the data in that way or via your cellphone and the PRTG app available in either Android or IOS store (including Apple watch) that allow you to get notifications when any sensor that you have setup doesn't respond. See Fig. 4 below for an example.



Below
Fig. 4: The PRTG app provides a wealth of information at a glance.

At a glance, can see a load of valuable information about network speed via pings, where the failure point is in a STL path via receive dBm levels, and your transmitter's forward and reflected power indications. During an emergency, you'll immediately be able to direct your attention (and vehicle) to the location of the cause of the problem.

The other power comes through the graphical histories that PRTG can provide for each device. You can see historical values for each sensor. If you want to know exactly when you went off the air and when things were restored, you can check any of the histories to find out.

You'll also have notifications and emails that you can reference. PRTG really helps to aggregate all of this data so that you can manage the increasing amount of detail that is necessary to keep stations running at their best.

So far, here in our market, we're only using about 47 sensors to monitor four of our five transmitter sites and their associated IP radio STL paths (one site doesn't have network data access). I plan to add our audio codecs for each location into the system as well, which will put us at around 65 sensors.

Certainly, there will be users who need to manage bigger systems, and 100 sensors might have them pushing the limit of the sites they could manage. There are also a number of cases where an engineer might never use all of the data options that the free version of PRTG offers them.

Whatever size operation you find yourself managing, the best time to begin harnessing the power of using SNMP network accessible devices is today, and you can get into it on no budget at all using PRTG network monitoring and the Ireasoning MIB browser.

APEX is the most powerful and flexible radio automation system in the world, the ultimate solution for any application. Whether you are a small Internet station, or a large multi-station group, APEX has all you need, at an affordable price. Join the thousands who have chosen Arrakis for their On air automation.

Easily supports One to Ten stations...

APEX can be just a single PC or easily supports up to 10 on Air PC studios with all Air, scheduling, and reporting. Record your voice tracks, change your schedules, or manage all of your audio libraries from anywhere that you have high speed internet. Have audio files that you want to go to only one station? It'll do that. Or maybe you want a handful of files to go to 3 of your 10 stations. No problem. Also, custom user profiles meet any size station's needs. The live (on-air) screen gives you complete control of how it looks. Resize, and map any module to your exact preference, and then save a special user profile for every on air personality. Easy to use and yet scales to meet your needs.

Just add a Windows 10 PC and stir...

For only $125 per month, APEX comes with all software for On Air, scheduling, and reporting. It comes with the hardware for sound cards, logic, and audio switching. Just add a current model Windows 10 PC and you can be on air in minutes. And... all software updates and telephone customer service is included. So for the cost of most monthly service contracts you can be on air with APEX.

www.arrakis-systems.com


IT'S ALL IN WHEATNET-IP
THE INTELLIGENT NETWORK
GSX: Ready To Go
-80.0
5:18:59 PM
GSX
FOCAL

Writer

Paul Kriegler

Director, Processing Sales US/ Canada

Telos Alliance

More info

Questions about this article? Email Paul Kriegler at *paul.kriegler@telosalliance.com* or Mary Ann Seidler at *maseidler@telosalliance.com.*

Above
Fig. 1: Omnia Enterprise 9s interface showing Omnia.9 display metering with processing activity, I/O loudness readings, and frequency analysis for one station. Up to eight are available from this interface and NFremote.

Virtualizing your audio processing: A checklist

Achieve high-density, virtual deployment of audio processing

As director of Omnia radio processing sales for Telos Alliance, I get to talk to some very IT-savvy engineers who have either moved their audio processing into virtual environments or are flowcharting future deployment.

From these conversations, I created a checklist of some things you should consider before virtualizing your audio processing.

Before we dive into the list, however, we should answer the most often-asked questions: "What is this virtualization business? And why is it useful?"

Virtualization

Virtualization is the concept of taking a server and, with the help of virtualization software, partitioning it so that it appears as several virtual servers running their own operating system, allowing the user to employ the server for several different applications instead of just one.

As applied to audio processing, imagine running hundreds of FM, HD or stream-processing instances, delivering content where needed and backed up on-premise or off-premise (cloud).

Why virtualize?

There are a number of good arguments for virtualization. Among them:

- Less dedicated broadcast hardware. Saves on IT costs and electricity, too.
- Backup, disaster recovery, redundancy and uptime. Think multiple servers to load balance or for failover, keeping your infrastructure more available and allowing for multiple backups or cloud deployment.
- Local studio rule changes. Send FM composite direct to transmitter sites over redundant paths using Omnia's μMPX codec (MoIP or Multiplex over IP).
- Combine all efforts for FM-RDS, HD and streaming! Run multiple instances on one server. Smaller footprint, easier to manage, provides consistency.
- Spin up or turn off instances as needed, operate on-premise or in the cloud. Need a holiday stream?
- Sign-on a new station or format? Changes like these can be handled quickly.
- Future-friendly. With software, we can add new features as standards or operating systems change.

11 THINGS
YOU DIDN'T KNOW ABOUT GV

- **NEW** future-ready embedded compute engine with 10X more processing power
- **NEW** efficient power supplies for reduced cost of ownership
- Cool operation for long life & reliability
- Field-serviceable RF power modules
- More power FETS for more power headroom
- Full cabinet design for easy service
- Positive pressure airflow for better heat transfer
- Cold start to full power in seconds
- ON-Air fan replacement
- Outstanding reliability based on 2,000+ installs
- HD Spectrum/Efficiency Optimizer for pristine signal quality & high efficiency

PLUS

FOR A LIMITED TIME: FREE 2nd back-up Exciter*

Contact your Nautel Rep today

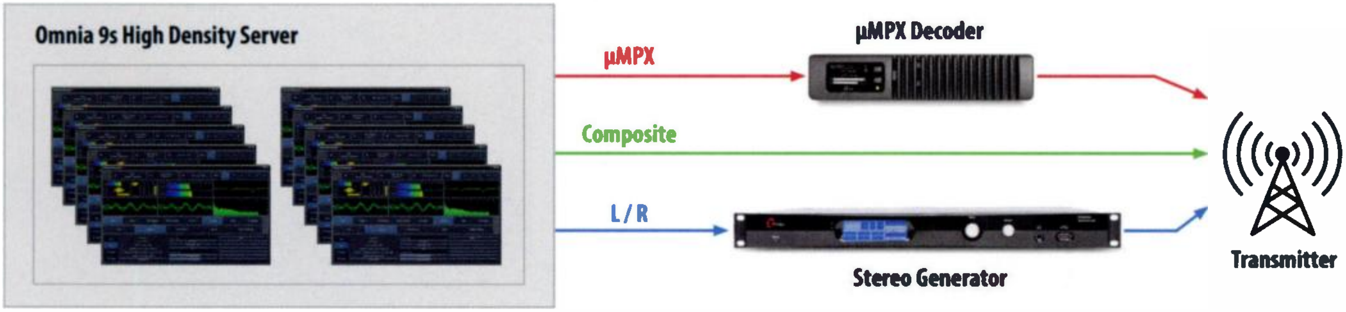
877-628-8350 or nautel.com/11things

Worry-Free Transmission

Because Radio Matters

* • Limited time offer
• Offer cannot be combined with other promotions or discounts
• Worldwide

nautel



Above
Fig. 2: Two ways to generate composite with Omnia Enterprise 9s on a server.

Just over a year ago, Omnia released its first virtual audio processing ecosystem, Omnia Enterprise 9s, which is designed for high-density or virtual deployment. It's based on the sound, performance, features and interface of our Omnia.9 hardware audio processing platform.

Pictured in Fig. 1 is the interface of the Omnia Enterprise 9s, which, if you are familiar with an Omnia.9, is identical. However, look closely and you will see one major difference: options for Stations 1 through Station 8. That's separate processing paths for eight radio stations, all running on one server.

Checklist to deploy processing virtually

1 **Bandwidth** — Not everyone has access to fiber. With the Omnia µMPX codec in the Enterprise 9s, you don't need that kind of bandwidth to send the entire FM baseband with RDS and stereo pilot across a link. With µMPX, it passes at a low bandwidth of 400 kbps without degradation of the audio baseband or transcoding artifacts. FLAC can be sent at 800 kbps.

2 **Secondary ISP** — Consider a secondary ISP to carry your content or composite signal in the event of main failure. You'll find two NICs on the rear of the Omnia MPX Node for redundancy. For that matter, Starlink (from Elon Musk) may soon become an option for transmitter sites way past "the end of the internet."

3 **Appropriate Server, CPU or Host** — Assuming you are running a virtualized environment for your audio processing. In our cloud testing of Omnia Enterprise 9s, moving from one physical server to another requires little or no "hit" or downtime. For users deploying one instance of Enterprise 9: each FM and µMPX output requires four CPU cores and 500MB total RAM.

4 **Handling I/O** — In the Omnia Enterprise 9S environment, here's what is available today:

Input — Livewire, AES67 and Windows drivers. Stream receiver.

Output — For true MPX out, including pilot and RDS, you will need to license the µMPX codec for FM transmission over some type of IP link with Omnia MPX Node hardware deployed at the transmitter site (Fig. 2).

In that figure you also see an Omnia.9sg deployed (optional). The Omnia Enterprise 9s software can encode a lossless FLAC stream, which the 9sg can use to make composite at the transmitter site.

5 **Handling EAS and PPM** — EAS: If your studio or program feed is generated elsewhere, current solutions include having your EAS unit at the transmitter, on its own separate local audio processor.
Watermarking: Stations in PPM metered markets that want to virtualize their airchain will need a way to properly watermark their signal. Full market testing has rolled out in a couple of large markets for MRC accreditation with Nielsen. While not released yet, this initiative will put the watermark where it belongs, inside the audio processor. (Our Linear Acoustic TV processors have had Nielsen watermarking for some time.) This also eliminates maintaining a hardware watermark encoder.

6 **Appropriate IT Department** — Do I have the appropriate IT talent to maintain the IP paths to where audio, or composite, is delivered?

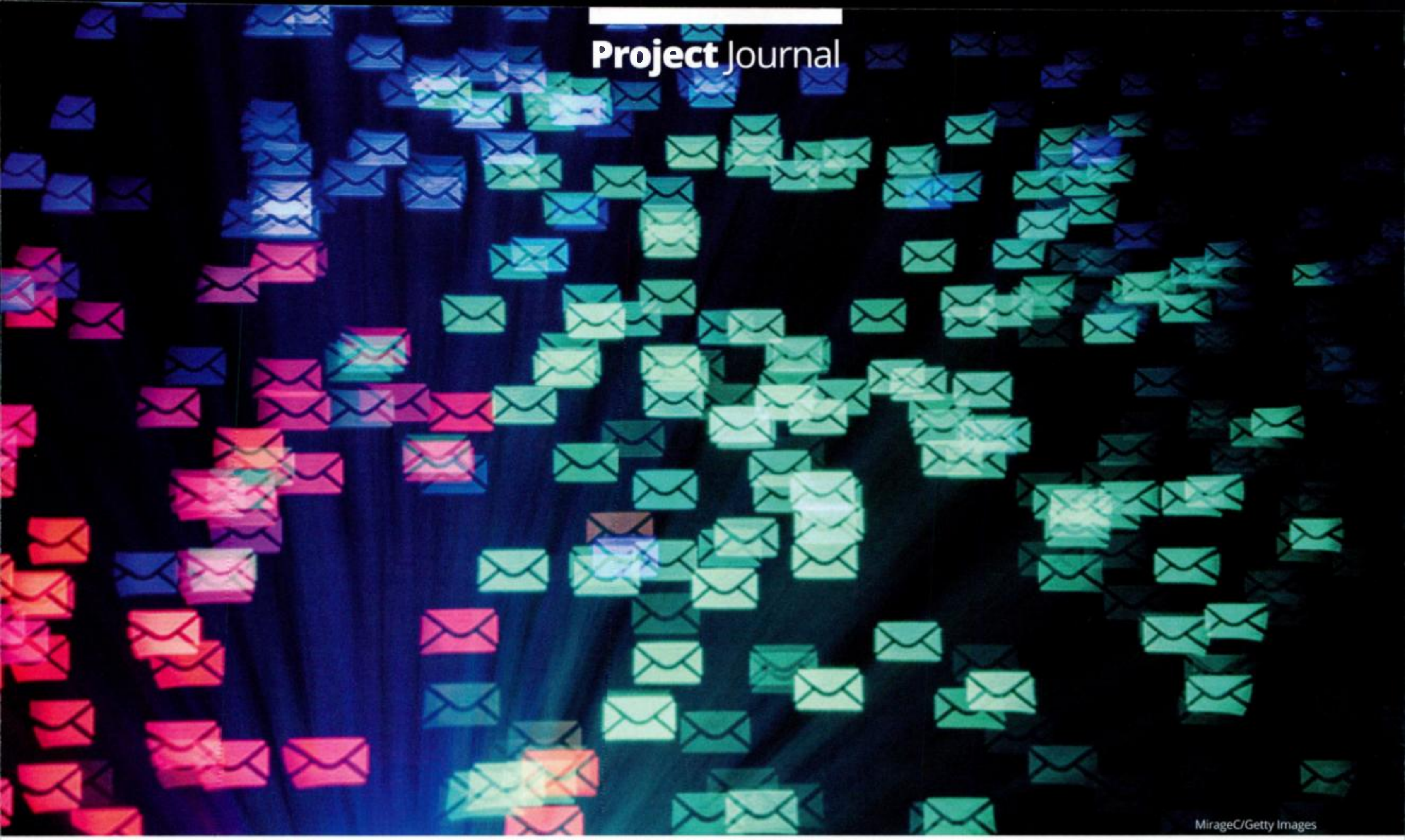
7 **Can I get this in a container?** — This is on our roadmap.

To help illustrate redundant paths to the transmitter site, Fig. 2 shows two ways to generate composite with Omnia Enterprise 9s on a server, feeding:

- µMPX Codec over IP to Omnia MPX Node: Omnia MPX Node encoder and decoders come with support for NET1 and NET2 ethernet jacks, for full redundancy using two IP paths.

- Omnia Enterprise 9s to Stereo Generator: The other path shown would send FM out from the Omnia Enterprise 9s as audio to an Omnia.9sg. FLAG encoding/decoding is another option to form your own STL.

It's fascinating to see stations using this Omnia Enterprise 9s product in so many different ways and workflows. It's a credit to the ingenious radio engineers in the industry that no two deployments have, so far, remotely been alike.


MirageC/Getty Images

Writer

Frank McCoy
Chief engineer of Salem Communications' Chicago cluster and president of the consulting firm FM and Co.

A DIY mail transfer agent

Old-school email origination clients need a little help with authentication

Quite a few devices in my broadcast technical environment report alerting information by email. I and others have come to depend on these alerts as what my Ops Manager Eric Thomas refers to as "the rhythm of life" as far as the radio station is concerned: pattern changes, tower lights on, like that.

In many cases these alerts come from old-school email origination clients without the ability to authenticate themselves.

These devices need a mail transfer agent or MTA that simply trusts and forwards, something you would never expose to the open internet. In addition, providers like Gmail and vtext have a variety of ways to detect and reject emails that are suspicious. It'd be nice if my mail was accepted without complaint or being bounced.

Like many firms of size, our company email has been SaaS outsourced to Microsoft 365, which requires meaningful email client authentication. The various broadcast devices I need to serve would never play nicely with an MS or Gmail directly. The system I want must live inside the private address, protective firewall environment of the corporate IT department. And it would be nice if it didn't cost much.

This article takes you through the build process step-by-step for a send-only MTA. All it does is route alerts from your various non-authenticating devices to real-world email providers on the outside. It will not and should not receive mail.

Almost anything will work

To begin the project, I found an old mid-tower Dell box in the recycling bin at the office. So my core system was free. So far so good. My application is low stress and almost any x86 castoff will work.

I updated the BIOS using the Dell tool and installed a 250-gigabyte flash drive. It could easily have been smaller, but it's what I had. The box already sported a mighty 4 gigabytes of RAM. When this machine shipped it had Windows 7 Pro on it. It still had the hologram license sticker. Vintage 2013, Intel Core i3. Old, unsuited for office use, but more than good enough for this.

The next step was to load Ubuntu Linux 20.04 Desktop on it. Download a copy at *ubuntu.com*. The 2.7 GB download is an .iso file for which you'll use your favorite DVD burner application to copy onto writable disk media or a thumb drive. Connect the box to a network segment that provides DHCP. Boot your boneyard PC from the media you created and install. If you see the choice, elect the "minimal" install. This skips the usual office desktop applications which aren't needed for this.

Write down your username and password choices (because if you don't, you'll be repeating the install). Then open a terminal with CTRL-ALT-T key combination and make your first use of "sudo" which means, more or less, "super user do." From the terminal you can invoke administrative privileges this way, needed for installation and configuration.

> **"The various broadcast devices I need to serve would never play nicely with an MS or Gmail directly."**

From now on, things in this article that you are expected to type will be bold, like... **sudo apt upgrade**, which will install the latest patches in case they aren't in the .iso you downloaded. You might as well start from the latest and greatest. Everything is lower-case and yes, case matters. Do not type any punctuation marks. Answer "Y" when prompted and watch the process.

Next is the utility for remote terminal access... **sudo apt install ssh**, giving you the same terminal functions by IP address using an app like putty.exe. Check from another machine to confirm access via SSH. You can discover the machine's DHCP assigned address with... **sudo ifconfig** which will probably complain that you need net-tools. Might as well add those using... **sudo apt install net-tools**.

You now have a generic Ubuntu Linux install. And for the record, this could probably be a Raspberry Pi* just as easily as the scrap pile Dell I am using.

DNS stuff

The process beyond this point requires two prerequisites:

1 You need a real registered host.domain hostname for your machine. Maybe you have a domain (and access to DNS records to assign its address and other attributes), or for a few bucks, just buy a name. If you buy one, make sure the name registrar provides DNS hosting. Most do for small operations like this.

2 You'll need an actual registered (static) IP address, at least initially, to facilitate the creation of certificates and mail encryption keys, required for delivery to name brand mail service providers. Don't worry, your MTA will retreat to the safety of a private address (i.e., 10.x.x.x, 172.16.x.x, 192.168.x.x) inside a proxy once it's set up. But to get the needed badge of validity, it must stand on the street corner of the real internet for a few minutes.

Start with the prerequisites by choosing a name for your newly configured box. I suggest mail.your_domain_name.com. Obviously, replace "your_domain_name" with your domain name. If you chose some other name during OS install, don't panic. Use the inbuilt nano Linux text editor to change it: **sudo nano /etc/hostname**. Replace the text as appropriate with the new hostname. Do the same with /etc/hosts: **sudo nano /etc/hosts**.

Commands are along the bottom edge of the little nano text editor. CTRL-O to write, RETURN, then reboot.

> **"Any mails from your devices, pointed to the private address the server has, will be forwarded to whatever mail service is indicated."**

Using the config tool provided by your domain DNS host, create an A record using the full hostname you gave the mail server. Associate it with the public, registered IP address you will use temporarily. Create an MX record pointing to the same hostname.

Keep in mind that DNS records may take a bit of time to propagate across the internet. I suggest leaving the records you create permanently in place, even though they won't point to the eventual resulting system anymore. Some mail providers check to see if your hostname is a real box.

Now let's install the MTA software: **sudo apt install mailutils**.

When you eventually get a choice dialogue box, choose "Internet Site," which is the default. Then tab to OK. If it asks for a system mail name, it's the same name you gave the server you are installing on.

To do what we want, Postfix will require a bit more configuration. Proceed to the text file /etc/postfix/main.cf: **sudo nano /etc/postfix/main.cf**.

You'll need to locate a few lines of the file for editing, so this would be a good time to test out the CTRL-W command in nano to find, first: mydestination = $myhostname, your_domain_name, localhost.com, localhost.

Substitute your domain name as indicated. $myhostname is a variable set earlier in the file by a "myhostname =" statement. Might be good to check that info is correct, but Postfix usually gets this right on install.

Then find... mynetworks = and change it to read mynetworks = 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16 to allow all the ranges of non-routable private addresses. These are the mail originating devices allowed to use your MTA for outbound messages.

The example config I show gets the whole universe of private IPv4 addresses. If you wish, you can restrict down to the individual device by using a complete address and /32 as the CIDR mask. Or anything in between. This is your gatekeeper. Let paranoia be your guide.

At this point save your changes to main.cf and restart postfix to allow the software to read the changes: **sudo systemctl restart postfix**.

Testing, testing...

Let's test your server from the terminal (CTRL-ALT-T) with... echo "This is a test of email" | mail -s " This is the subject line of the test" your_email_address.

Some receiving domains will probably accept the mail. Outlook.com (MS 365) might, but it'll probably show up in junk or clutter. Vtext.com (Verizon email to text conversion) probably will as well and display in cellular text. Google Gmail won't accept it at all; it'll get bounced. So there is more to do. Time to enable encryption, certificate validation and transport layer security (TLS).

Meanwhile, take a look at the logging that Postfix provides, as it will tell you exactly what happened in the interaction with any outside service. Linux provides a nice utility to look at the last few lines of a log file and even monitor it as lines are added: **sudo tail -f /var/log/mail.log** (note: your install may name this log file differently, but it will be in the folder /var/log). Also notice that the -f argument makes the utility display new lines as they are appended in real time. You can have more than one terminal instance open at a time, allowing you to watch what happens with the gmails of the world in real time.

Next up is the authentication piece that real email services will be looking for from your MTA.

Let's get the needed credentials from the free Let's Encrypt folks. Consider making a contribution to their efforts. **sudo apt install certbot**. As always, press "Y."

Then **sudo ufw allow 80**. This tells the UFW firewall to open a port for HTTP. The command should answer with confirmation that it opened the port. Certbot's process will reach out and "Let's Encrypt" will reach back over port 80 to confirm the host presence at the address claimed, using the DNS records you created. But certificate information doesn't include IP address directly, so a later move to a private address inside the firewall shouldn't break anything.

Now let's create a certificate that identifies this server as within the domain you've placed it in and provide a public key to go with the private key that'll be stored on this host. **sudo certbot certonly --standalone --rsa-key-size 4096 --agree-tos --preferred-challenges http -d your_domain_name** (those are double dashes, by the way).

When complete, you should see this:

Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator standalone, Installer None
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for `your_domain`
Waiting for verification...
Cleaning up challenges

IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at:
/etc/letsencrypt/live/your_domain/fullchain.pem
Your key file has been saved at:
/etc/letsencrypt/live/your_domain/privkey.pem
Your cert will expire on [year-month-day please take note!]. To obtain a new or tweaked version of this certificate in the future, simply run certbot again. To non-interactively renew *all* of your certificates, run "certbot renew"
- If you like Certbot, please consider supporting our work by:

Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
Donating to EFF: https://eff.org/donate-le

If it doesn't work, check your DNS configuration, or maybe just wait a while and try again. When it completes, pay attention to certificate expiration.

Now let's tell our server where the keys are kept. Back to the postfix config file we go:
sudo nano /etc/postfix/main.cf.
Under # TLS parameters find:

smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key

and change it to:

smtpd_tls_cert_file=/etc/letsencrypt/live/your_domain_name/fullchain.pem
smtpd_tls_key_file=/etc/letsencrypt/live/your_domain_name/privkey.pem

Finally, to further satisfy Gmail, add the line: "always_add_missing_headers = yes".

Save, exit nano then restart postfix: **sudo systemctl restart postfix**.

If all went as described (it should have), then you have a mail transfer agent. Any mails from your devices, pointed to the private address the server has, will be forwarded to whatever mail service is indicated.

Finally, you can disconnect from the public internet (the registered static IP), bring this server inside the firewall and give it a private address.

Use the desktop "settings" tool to change the IP. So long as port 25 is open through the firewall, you are all set. Just point your various reporting devices to that same private address and watch the alerts flow.

> **"It doesn't take a lot to create a functioning MTA out of equipment already on hand."**

If it doesn't work, check your DNS configuration or maybe just wait a while and try again. When it completes, pay attention to certificate expiration. In my case, using GoDaddy DNS, I had to assign the IP address to my machine that is associated with the your_domain_name DNS entry. Your DNS provider may work differently. The error messages that certbot sends are helpful for Googling.

Now let's tell our server where the keys are kept. Back to the postfix config file we go.

 # Raspberry Pi differences:

1 Enable SSH terminal access using the raspi-config tool
2 net-tools are already installed in the Raspi Lite .iso
3 Use raspi-config to change the hostname
4 mailutils does not include postfix when pulled from the raspi repositories. **sudo apt install postfix**

It doesn't take a lot to create a functioning MTA out of equipment already on hand. Having that functionality will ensure that important status and warning messages get through, which will make the engineer's job easier and provide for faster responses to anomalies.

Readers Forum

Air Chain by Dick Sequerra

Cris, your delightful reminiscing on changes in audio processing ("Air Chains Then and Now," June 16, 2021 RWEE) brought back memories of when I was hired by GAF Corp. in 1976 to rebuild and manage New York's Classical WNCN, 104.3 FM.

We started with a blank sheet, and the first thing GAF's chairman did was to retain the services of the legendary Dick Sequerra to handle all aspects of the audio chain.

The chairman was an audiophile, and had one of the amazing Sequerra FM1 tuners in his home. Of course, we had one in our main studio as an off-air monitor, too.

As our studios and offices were located in a building on Sixth Avenue, and the building's foundation went straight down to the Sixth Avenue subway, the studios were built as rooms within rooms, with the inner rooms mounted on rubber dampers that resembled hockey pucks.

Sequerra hired WFMT Chicago CE Al Antlitz to design and fabricate the boards used in each studio. Their performance was far superior to anything available in the marketplace and greatly added to the very clean sound that he wanted to achieve.

Sequerra hoped to broadcast the way that the BBC did with its Classical channel, requiring the announcers (presenters) to ride gain, with no processing. AFTRA, the announcers' union, mandated that wasn't about to happen, so we ended up with the then-new Orban processors, which really were superior to anything else on the market.

As part of his passion for excellence, Sequerra went into the bowels of Manhattan and rebuilt the telco amplifiers going from our studios to the Empire State Building, unknown to New York Telephone, of course.

After the new Gates transmitter was delivered to our space in the Empire State Building, Sequerra spent three days tearing the transmitter apart and rebuilding it to his specs. The performance was just spectacular! My memory is not absolutely perfect here, but as I recall, with the Orban bypassed, the lines were clean to something in the -90 dB area.

There is no doubt that Dick Sequerra's passion and expertise made WNCN the best-engineered FM station in the country. The programming team was led by a Julliard graduate, and the combination made WNCN a leader in the country's Classical music FM stations.

Thanks for the memories.

Robert E. Richer
International Media Consulting
Farmington, Conn.

TECH MART